

Learn RouterOS

By Dennis M Burgess



Learn RouterOS

By Dennis M Burgess



Learn RouterOS

By Dennis M Burgess

Copyright and Trademarks

All trademarks and copyrights are held by the respective copyright holder.

Copyright © 2009 by Dennis Burgess

All rights reserved. No part of this book may be reproduced, stored, or transmitted by any means—whether auditory, graphic, mechanical, or electronic—without written permission of both publisher and author, except in the case of brief excerpts used in critical articles and reviews. Unauthorized reproduction of any part of this work is illegal and is punishable by law.

ISBN: 978-0-557-09271-0

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, 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 the knowledge and 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 do RouterOS work for your business, or company. You will have 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 give you the direction on what features you need to use, and why. If you need to know what exactly a feature or command does, you will need 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 ways of configuring 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 them! Some features are packed with comments and suggestions on how to use that feature along with other features, and why to use such features.

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 through 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 even 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 needing a method to control bandwidth for subscribers, built their first RouterOS x86 systems.

After realizing the power and performance of RouterOS, as 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

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, 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 a purpose to help Wireless ISPs as well as provide high-quality consulting services for RouterOS systems. In the USA the options for RouterOS consulting services were very limited to small home businesses, and technician level admins trying to help out businesses and ISPs with RouterOS. I formed Link Technologies, Inc to give these businesses 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.

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

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

Link Technologies, Inc
PO Box 96
House Springs, MO 63051
<http://www.linktechs.net>
Support@linktechs.net
314-735-0270

Table of Contents

[Introduction](#)

[Who should use this book](#)

[About the Author](#)

[Link Technologies, Inc](#)

[What is RouterOS?](#)

[How this Book is organized](#)

[RouterOS Hardware](#)

[RouterBoard Devices](#)

[Solar Power and RouterBoards](#)

[X86 Based RouterOS Systems](#)

[Supported x86 Hardware](#)

[RouterOS Licensing](#)

[Extended Frequency Licenses](#)

[Installation](#)

[Using NetInstall on RouterBoard Products](#)

[DOM / Flash Card / Hard Disk Installation via NetInstall](#)

[Ways to Lose your RouterOS License](#)

[Accessing RouterOS](#)

[What are all of the methods of accessing a RouterOS System?](#)

[Default User and Password](#)

[Using Neighborhood Viewer](#)

[Using Telnet](#)

[SSH – Secure Shell Access](#)

[WebBox](#)

[Interfaces and IP addresses](#)

[Wireless Interfaces](#)

[Registration Table](#)

[Routing](#)

[System Options](#)

[Basic Firewall](#)

[Simple Queues](#)

[PPPoE Client](#)

[Access List](#)

[DHCP Server](#)

[Upgrades](#)

[Using WinBox](#)

[WinBox Menus](#)

[WinBox Interface Options](#)

[Managing RouterOS](#)

[User Defaults](#)

[User Management](#)

[Adding/Removing/Changing Local Users](#)

[RouterOS User Groups](#)

[Active Users](#)

[SSH Keys](#)

[AAA Settings – Radius RouterOS Users](#)

[RouterOS Services](#)

[FTP Service](#)

[API Service](#)

[SSH / Telnet Services](#)

[WWW Service / WWW-SSL Service](#)

[WinBox Service](#)

[Working with Files](#)

[Backup / Restore](#)

[Creating Editable Text Backup Files](#)

[Importing Scripts](#)

[Logging](#)

[Setting Logging Rules](#)

[Basic RouterOS Setup](#)

[Configuring IP Addresses](#)

[Common IP Information](#)

[24 bit Block or a /8 Prefix](#)

[20 bit Block or a / 12 Prefix](#)

[16 bit Block or a / 16 Prefix](#)

[Default Routes](#)

[DNS Caching/ Service](#)

[DHCP-Client](#)

[DHCP-Server](#)

[DHCP Server Wizard](#)

[IP Pools](#)

[Masquerading - NAT](#)

[Configuration of basic Masquerading](#)

[Home Router](#)

[Home Router Walkthrough](#)

[Verify that we obtained an IP address](#)

[Common Wireless Configurations](#)

[Bridged Access Point Configuration](#)

[CPE – Client Premise Equipment Configuration](#)

[Bridged Client](#)

[How to Use Pseudobridge Mode](#)

[Routed / NAT CPE](#)

[RouterOS Features](#)

[IP Features](#)

[Interface ARP – Address Resolution Protocol Settings](#)

[ARP List / Table](#)

[Static Routing](#)

[Routing and Routes](#)

[Checking Gateways](#)

[Using Distances](#)

[ECMP – Equal Cost Multiple Path](#)

[Policy Based Routing](#)

[Routing Policies](#)

[Using Mangle to Route Traffic](#)

[Firewall Features](#)

[Traffic Identification](#)

[Understanding Connection States](#)

[Packet Flow in RouterOS](#)

[Chains](#)

[Input Chain](#)

[Output Chain](#)

[Forward Chain](#)

[Other Chains](#)

[Jumping to Chains](#)

[Returning from Chains](#)

[Address Lists](#)

[How to Match Data](#)

[Connection Bytes](#)

[Built-In Peer to Peer Filtering](#)

[Layer 7 Filters](#)

[Connection Limiting](#)

[Port Scan Detection](#)

[Ingress Priority / TOS / DSCP](#)

[Random](#)

[Limit / DST Limit](#)

[Nth](#)

[Time](#)

[Firewall Actions](#)

[Protecting Your Router](#)

[Protecting Networks](#)

[Common Firewall Options](#)

[SPAM Prevention](#)

[Brute Force Attacks](#)

[DOS/POD Attacks](#)

[Firewalling Examples – Using Multiple Rules to do what YOU want!](#)

[Using Mangle](#)

[Chains](#)

[Using Marks](#)

[Packet Marks](#)

[Routing Marks](#)

[Connection Marks](#)

[Change TOS Bit / DSCP](#)

[Change MSS](#)

[Clear DF](#)

[Set Priority](#)

[Strip IPv4 Options](#)

[Performing Network Address Translation](#)

[Chains](#)

[Masquerading](#)

[PPPoE Client and other types of Tunnels and Masquerading](#)

[Inbound NAT](#)

[Outbound NAT](#)

[Performing a One-to-One NAT – Assigning a Public IP to a Private](#)

[Selective Port Forwarding](#)

[Inbound NAT with DHCP Public IP Address](#)

[Redirect](#)

[Interfaces](#)

[Ethernet](#)

[Switch Controls](#)

[Ethernet Speed and Negotiation / MDI-X](#)

[Virtual Ethernet Interfaces](#)

[Bridge Interfaces](#)

[Bridge Ports](#)

[Bridge Settings / Using IP Firewall](#)

[Virtual LAN \(VLANs\)](#)

[VLAN Configuration](#)

[Bonding](#)

[MESH](#)

[Switches and MESH](#)

[VRRP](#)

[Tunnels](#)

[EoIP](#)

[Bridging an EoIP Tunnel](#)

[IPIP](#)

[PPP System](#)

[PPP Secrets](#)

[PPP Profiles](#)

[PPP Active Connections](#)

[PPP Server](#)

[PPP Client](#)

[L2TP/PPTP Servers](#)

[Windows PPTP VPN Users](#)

[L2TP/PPTP Server Interfaces](#)

[L2TP/PPTP Client](#)

[Bridging PPTP](#)

[PPPoE Server](#)

[PPPoE Server Interfaces](#)

[PPPoE Server, Dynamic Routing and / 32 Subnets!](#)

[PPPoE Client](#)

[OpenVPN](#)

[OpenVPN Server](#)

[OpenVPN Server Interface](#)

[OpenVPN Client](#)

[IPSec](#)

[IKE Domain](#)

[Choosing a Tunnel Type](#)

[Wireless and RouterOS](#)

[WIC – Wireless Interface Cards](#)

[Basic Configuration of Wireless Interface Cards](#)

[Wireless Tools](#)

[Air/Data Rates and Performance](#)

[Access Point Time](#)

[Bands](#)

[Wireless Operational Modes](#)

[AP-Bridge \(P2MP Access Point\) Mode](#)

[WDS-Slave Mode](#)

[Bridge \(P2P Access Point\) Mode](#)

[Station \(Wireless Client\) Modes](#)

[Security Profiles \(Securing your Wireless Connection\)](#)

[MAC Authentication](#)

[WEP \(Wired Equivalent Privacy\)](#)

[WPA / WPA2](#)

[Access Lists](#)

[Registration Table](#)

[Connection Lists](#)

[Area / Area Prefixes](#)

[Virtual Access Points](#)

[N-Streme](#)

[N-Streme Dual](#)

[Using WDS \(Wireless Distribution System\)](#)

[WDS Bridged Wireless Link](#)

[Static WDS Bridges](#)

[WDS Bridged Access Points](#)

[WDS Bridged Access Points - Dual Radios](#)

[WDS and 802.11n](#)

[Wireless Link Optimization / Best Practices](#)

[Keep it Simple First](#)

[Hardware Selection](#)

[Antenna coax and selection](#)

[Antenna Alignment](#)

[Find Possible Interference](#)

[Signal Issues](#)

[Secure your Link and Testing](#)

[Minimize Rate Flapping](#)

[Using Nstreme](#)

[Troubleshooting Wireless Links](#)

[Low Signal](#)

[Wandering/Fluctuating Signal](#)

[Bad CCQ](#)

[Traffic Control](#)

[Identifying Queue Data](#)

[Hierarchical Token Bucket – HTB](#)

[HTB Packet Flow](#)

[HTB Queue Tree Structure](#)

[HTB and Rate Limiting](#)

[Queue Types](#)

[FIFO Queues](#)

[RED Queues](#)

[SFQ Queues](#)

[PCQ Queues](#)

[Using PCQ](#)

[Queue Trees](#)

[Simple Queues](#)

[Limiting Total Throughput for IP or Subnet](#)

[Bursting](#)

[Creating Queue Priorities with Parents](#)

[Ensuring Bandwidth Allocations – VoIP](#)

[Creating Advanced Queues](#)

[Double Queuing](#)

[Large Transfer Queues](#)

[Setting Multiple PCQ Rates](#)

[*Using Multiple Data Packages and PCQ*](#)

[Controlling P2P \(Peer-to-Peer\) Traffic](#)

[Limiting / Changing P2P and the Consequences](#)

[Hotspots](#)

[Wireless and Hotspots](#)

[Paid Hotspots](#)

[Free Hotspots](#)

[RouterOS and Hotspots](#)

[Definitions](#)

[Setup of a Hotspot Interface in RouterOS](#)

[Configuration of Servers and Server Profiles](#)

[Hotspots with Radius](#)

[Internal Hotspot User Management](#)

[Using IP Bindings](#)

[Creating Walled Garden Entries](#)

[Viewing Hotspot Hosts and Active Users](#)

[Running multiple-subnets behind a hotspot interface](#)

[Running Dynamic Routing \(RIP/OSPF\) Behind a Hotspot Interface](#)

[Radius Client](#)

[Multiple Radius Servers](#)

[Troubleshooting Radius Client Issues](#)

[Nuts and Bolts](#)

[Accounting](#)

[DHCP Relaying](#)

[Neighbors](#)

[M3P – MikroTik Packet Packing Protocol](#)

[Pools](#)

[Socks](#)

[Clock](#)

[NTP](#)

[Client](#)

[Server](#)

[System Identity](#)

[Logging](#)

[Reset Configuration](#)

[Scripting](#)

[Scheduler](#)

[Auto Upgrades](#)

[Watchdog](#)

[Bandwidth Test Server](#)

[Bandwidth Test Client](#)

[E-Mail System](#)

[Using Fetch Commands](#)

[Graphing](#)

[Packet Sniffer](#)

[Streaming Packet Sniffer Data](#)

[TFTP Server](#)

[Traffic-Flow](#)

[UPnP](#)

[IP Scan](#)

[Web Proxy](#)

[Web Proxy Access List](#)

[Cache and Direct Web Proxy Tabs](#)

[Transparent Web Caching](#)

[Store System](#)

[MetaRouters](#)

[Dynamic Routing](#)

[If Installed vs Always](#)

[RIP](#)

[OSPF](#)

[Changing Path Costs](#)

[OSPF Full Duplex Links](#)

[BGP](#)

[Instances](#)

[Peers](#)

[Networks](#)

[Aggregates](#)

[Routing Filters](#)

[The Dude NMS](#)

[Installation](#)

[Windows Installation](#)

[RouterOS Installation](#)

[Dude Agents](#)

[Installation of a Dude Agent](#)

[Dude Layout](#)

[Running a Server](#)

[Resetting Configuration](#)

[Menus and Options](#)

[Server Configuration](#)

[Configuration of Dude Servers](#)

[Dude Agents](#)

[Dudes Syslog Server](#)

[Dude Discovery Services](#)

[Admins](#)

[Charts](#)

[Devices](#)

[Device Options](#)

[Device Appearance](#)

[Files](#)

[Transferring Files within Dude](#)

[Links](#)

[Link Speed Setting](#)

[Logs](#)

[Network Maps](#)

[Map Settings](#)

[Adding Devices to your Maps](#)

[Working with Devices](#)

[Upgrades](#)

[Creating Links](#)

[Creating and Linking to Submaps](#)

[Notifications](#)

[Outages](#)

[Probes](#)

[Tools](#)

[User Manager](#)

[Hardware / License Requirements](#)

[Installation of User Manager](#)

[Configuration of User Manager](#)

[First Time Access](#)

[Understanding Concepts and Definitions](#)

[Basic Configuration Settings](#)

[User Sign-Ups](#)

[User Sign-In Page](#)

[Active Sessions](#)

[Vouchers](#)

[Command Line Interface](#)

[Quick Reference Guide](#)

[NetInstall of RouterBoard Products](#)

[NetInstall your Flash / DOM / Hard Disk](#)

[Creating a Active/Backup Bridged Auto-Fail Link](#)

[Setup Transparent Web Proxy System](#)

[Redirect Non-Paying Customer](#)

[Per Connection Load Balancing](#)

[Create a Private VPN](#)

[Appendix](#)

[Features Only Available via Command Line Interface](#)

[Index](#)

What is RouterOS?

Simply put it is an infinitely configurable routing software package¹. 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 give 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 Addresses and data traffic. In the world of IP routing, there is not much that RouterOS cannot do! Many routers and network devices will let you do certain functions. One device may be a PPPoE Server/Concentrator. Another device may control bandwidth and the way the data flows across your network. Then 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 professionals to understand each device.

RouterOS does 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 will take a look at a cost saving system that has the same reliability and performance that they are used to in more expensive hardware. In some cases, RouterOS devices and software can be less than one-quarter of the cost of similar capable device, and have more features than those more expensive devices.

How this Book is organized.

There are two sections to this book. The first section will teach you all about the features that RouterOS offers, and how they relate to different types of networks. You will learn about the feature, what it does and how it can help your network. The second track is a quick configuration guide. This lets you understand the components of the features, and puts them into an example for you.

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 Ethernet ports, wireless slots, memory configurations, and design types. RouterBoards can cost under \$49 USD, and up to several hundred depending on the hardware. These devices 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. The RouterBoard Crossroads platform is a micro Access-Point or CPE, Client Premise Equipment. These units are low cost, and include a built in 400mw 802.11 b/g wireless radio card. This radio also is FCC certified with a number of antennas. This board works great as an indoor access point or a client radio.

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 above, includes three Ethernet and Mini-PCI Slots. There are two versions though, a standard 433 and a 433AH. The AH includes an ultra high power CPU, at 680MHz, and the added Micro-SD card. The standard 433 does not have the Micro-SD card slot, and has a lower speed processor clocked at 300MHz.

Other versions include a mini-router, or RouterBoard 450 including five Ethernet ports, and a 493 Multiport Router. This unit includes nine Ethernet ports and three M-PCI slots. They make the 493 in both standard and AH versions, with the AH having the faster CPU just like the 433AH. However the 493AH does not include the Micro-SD card slot. Mikrotik also has come out with a dual radio board, the 411AR, giving you the high power CPU and an integrated b/g radio card, but also gives you a radio card slot for future expansion.

The RouterBoard 600 is considered an Extreme Performance 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 are looking to run 802.11N you will typically need to use this type of board as the 802.11N protocol allows for greater than 100 Megabit UDP throughput. Without the Gige interfaces, you will have a hardware limit at your Ethernet port.

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 1333MHz. 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.

These RouterBoards all contain an on-board NAND. NAND is basically Flash Memory, just like your USB Stick or Compact flash card. This is on-board a 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>.

Solar Power and RouterBoards

I have had quite a few requests on how to use RouterBoards with Solar systems. So I wanted to give you a few pointers. The key is power consumption, the newer RouterBoards, specifically the 400 series, is the most common boards used for solar powered sites. Most sites are powered by battery arrays at ether 12, 24, or 48 volt. 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 is some voltage drop, you can do a web search on how to calculate this. If you are not doing a long Ethernet run, then 12 volt 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.

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 12 volt source, you will quickly drop below 10Volts and the RouterBoards will stop running. If you wanted to use 48v, the RouterBoard will not take that voltage 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 and we have seen the voltage spike higher than what the RouterBoards are designed for so they power off to prevent overvoltage. So I like to run a bit lower than 24 volt 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 deployed and have never had to mess with it other 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 though, are based on a number of interfaces and with multi-port Ethernet cards and wireless cards on the market as well as available through MikroTik; you can make an x86 RouterOS system with little effort and at little 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 other companies out there as well; a simple Internet search will provide a number of results, which sell completed x86 systems with performance and reliability in mind. These systems are designed to use multiple bus channels, and high quality hardware to deliver the peace of mind.



One such manufacturer is Link Technologies, Inc. Their PowerRouter series of

devices gives you out of the box, ready to run RouterOS Systems. They are designed for high performance RouterOS routing taking into account bus speed limitations, and even adding multi-core processors to 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. They offer this model in both AC and DC versions

They also create an ultra high-end system, called the PowerRouter 2200 series. These systems can run up to Dual Quad Core Xeon processors, and can deliver up to 22 GigE Interfaces, including SFP interfaces that you can use Fiber modules with. These also sport dual hot-swappable power supplies as well.

Supported x86 Hardware

It's important to note that RouterOS does not use "drivers" in the same respect 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 not like this. RouterOS contains all of the drivers that you will need right out of the main installation. MikroTik though, chooses based on popularity, usability, as well as what is in the latest Linux kernel to base what drivers to include with the installation package.

With that said 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 T1/E1 interfaces, Mini-PCI and PCI adaptors, 3G or cellular data card, 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 pre-designed system that is supported and tested with RouterOS, I would suggest purchasing these. The cost on these is typically minimal vs. the cost of router failures due to hardware failure. I have seen this many times, customers wondering why their system does not constantly run. I actually asked one customer what kind of hardware, 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 as well as designed for a long lifespan. Servers are built with higher grade components, power supplies, and better network cards typically, and this is why they tend to last longer. Same with your RouterOS x86 device. 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, 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 you use to install RouterOS on. You can install RouterOS on USB sticks, SATA and IDE Hard Drives, Disk on Modules or DOMs², as well as compact flash cards. You can move the storage device from one system to another, but not from one storage device to another. So you can move your compact flash card from one x86 system to another x86 system. You cannot move the license from the existing compact flash card to another. If you need a larger compact flash card, then you will have to purchase another license.

What is my Software ID? The software ID is the ID number associated with your RouterOS installation. It uses the hardware, disk information as well as other methods to generate software ID Key. This key is then used to generate a license upon 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 etc, **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 feature³. Some may have special paperwork for you to fill out to obtain this license feature. However, if you have a license or can run 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. Please see your reseller or distributor for costs associated with this license feature.

Installation

Installation methods will depend on what hardware you are using. RouterOS can be installed on many different devices. These would include x86 computers, or RouterBoard Products. RouterBoards typically come with not only the RouterOS software already loaded, but has a license 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 a IDE or SATA DOM, or possibly a USB stick or other form of flash card. Compact flash cards would be included with this. You can though, use three other methods. NetInstall using a bootable network interface card or NIC is one method. Using a Floppy is another, as well as 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. Note that RouterBoards should come with an 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 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. 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 powering on is slim. You may even need to look around 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 your 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
- 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, would 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 etherl 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, we should have already downloaded all necessary files.

Fourth, ensure that your PC does not have any firewalls turned on or active and 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 is 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 *"pres 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*”. You will need 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
```

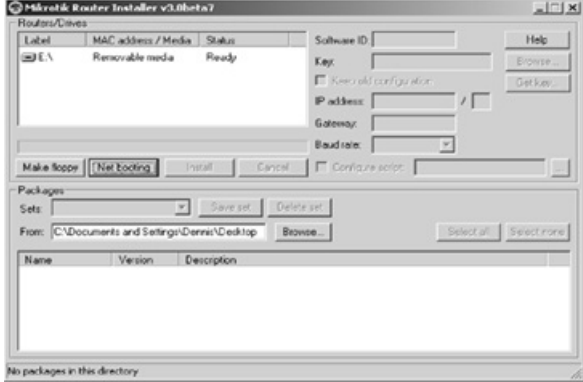
```
  p - boot protocol
```

```
  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.



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:



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. From 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 installation
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:
F5ZJ23uJmf7rJbMTIiNoBLuySVaaa1FwgQDdutWLWNKildr2ze3rKX
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.

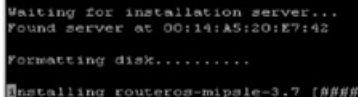


Note that we now have a device, typically labeled *nstreme*, along with its MAC Address. This is the RouterBoard, and it's waiting for installation. We then use the

browse button under the packages section and find the location where 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, as 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, 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, generates the SSH Keys, and starts the RouterOS Services. At this point, you have a working RouterOS system!

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 form 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 with 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.



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 its 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 have those options, including your baud 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!

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!

Accessing RouterOS

RouterOS is not your normal Router. Typical 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 layer2? 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 am able to access all other RouterOS devices without IPs and without configuration remotely!

The most common way though, 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 routers IP address. This will bring up a configuration page, which you can download WinBox at. I would suggest though, getting the latest version via Mikrotik's webpage.

Just like the Net Install Utility, WinBox will function at either Layer2 or Layer3. So 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.

What are all of the methods of accessing a RouterOS System?

- 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

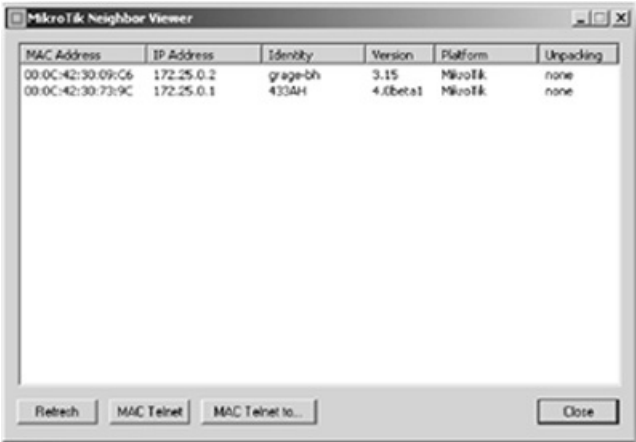
Default User and Password

RouterOS will default with 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. Upon selecting one of these, you have the option to open a MAC telnet session with it. This opens the terminal program that 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.



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, will allow you to MAC Telnet into your router.

```
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

      XXXX      XXXX      XXXX      TTTTTTTTTT      XXXX
      XXXXX     XXXXX     XXXX      TTTTTTTTTT      XXXX
      XXX XXXXX XXXX III XXX XXXX SSSSSS 000000      TTT      III XXX XXX
      XXX XX XXXX III XXXXX SSS SSS 000 000      TTT      III XXXXX
      XXX XXXX III XXX XXXX SSSSSS 000 000      TTT      III XXX XXX
      XXX XXXX III XXX XXX SSS SSS 000000      TTT      III XXX XXX

Mikrotik RouterOS 4.0beta1 (c) 1999-2000      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 an 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. Upon loading your RouterOS device, you will note that it generates SSH security keys. These keys are used to the secure connection. This means that text that is transmitted or received by your SSH client is encrypted, and not sent in clear text.

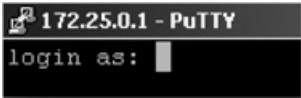
SSH though does run on the IP layer, so you will need to have Layer 3 connectivity to your Router. There are a number of FREE SSH clients that you can use. Putty is one of them, as well as OpenSSH, and other applications. We will show the Putty application here.



As you can see there is 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 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 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, so that you don't get this message again if you wish. Typically you would hit yes to cache the key. If you hit no, you will continue connecting but it will not cache they key.



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 use a webpage for basic configuration. To get to this page, you will need layer3 connectivity or IP connectivity to your Router. Your PC must be on the same subnet as the RouterOS system.

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

MikroTik webbox 4.10.0.1 login

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](#).

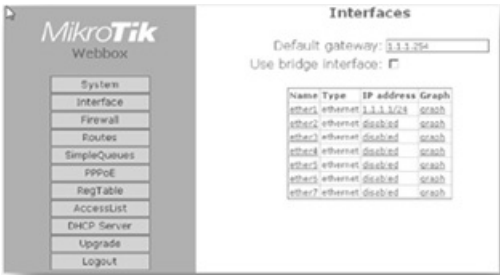
License
Mikrotik, RouterOS and the Mikrotik logo are registered trademarks of Mikrotik's SIA. Please read the [license](#).

As you can see, there are a number of options here. If you click on the WinBox image, you can download, 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 our “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. The WinBox section we will go into all of the feature usages in detail.

Interfaces and IP addresses



Once logged into the WebBox you will be presented with a number of options. On the left side, you will have your 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.

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

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 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.

Configuration for ether2

☒ Disabled
☐ Obtain an IP address automatically (DHCP)
☐ Configure an IP address manually

OKCancel

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.

☐ Disabled

☐ Obtain an IP address automatically (DHCP)

☒ Configure an IP address manually

Address:

Netmask:

OK Cancel

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 be 255.255.255.0. Once you have entered this information, go ahead and press OK.

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

Note in the example on the left, you have an IP with your static IPs. Here you can also click on [graph](#) to view the interface graphing if you have this enabled.

[Wireless Interfaces](#)

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

WebBox wireless interfaces will show wireless interfaces, with a type of wireless. We can select the type wireless to pull up the basic wireless information to configure your wireless interface.

Wireless interface (wlan3-2.4)

ssid:

Mode:

ap-bridge

Band:

2.4GHz-b/g

Frequency:

2.432GHz

Authenticate by default:

☒

Forward by default:

☒

You can see the wireless interface settings. You can configure the basic options of your Wireless interface here. You can setup your SSID, Mode, 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.

Security

☐ None

☒ WiFi Protected Access (WPA)

Pre-shared key (8 - 64 characters):

Group key update:

OK

Cancel

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.

Registration Table

The RegTable 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

Default gateway:

Routes

Destination	Gateway	
0.0.0.0/0	1.1.1.254	disable edit remove

You can also specify the default gateway for your RouterOS system, right here by typing the gateway right 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.

Add New Route

Destination:

Netmask:

Gateway:

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

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

[System Options](#)

System



The screenshot shows the 'System' configuration window in RouterOS WinBox. It contains the following fields and options:

- ID:** A text box containing 'www.mikrotikrouter.co'.
- Version:** Displays '3.19'.
- System:** A button labeled 'RESET!'.
- Do:** A button labeled 'reboot'.
- Change:** A button labeled 'password'.
- Refresh:** A text box containing '1s' with a dropdown arrow on the right.

Under the system option on the left side, you have options to setup the system ID; this is the identity of the RouterOS system. It will also display your version, allow you to reboot your RouterOS device. You can also change your user's password from this screen.

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. Be careful with this as it will wipe out your configuration as well.

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 perform NAT out the public interface. These enter specific commands into RouterOS to perform these actions.

Firewall

Public interface: ether1 ▾

Protect router: ☐

Protect customer: ☐

NAT: ☐

Apply

[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, as well as 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
inet	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:	<input type="text"/>
Out-Limit:	<input type="text"/>
In-Limit:	<input type="text"/>
Target-IP:	<input type="text"/>
Interface:	<input type="text" value="all"/>
Time:	<input type="text" value="00:00:00"/> - <input type="text" value="23:59:59"/>
Days:	sun <input checked="" type="checkbox"/> mon <input checked="" type="checkbox"/> tue <input checked="" type="checkbox"/> wed <input checked="" type="checkbox"/> thu <input checked="" type="checkbox"/> fri <input checked="" type="checkbox"/> sat <input checked="" type="checkbox"/>

PPPoE Client

PPPoE client is disabled

Enabled:	<input type="checkbox"/>
User:	<input type="text"/>
Password:	<input type="text"/>
Interface:	<input type="text" value="ether1"/>

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 as well as the username and password.

Access List

The RouterOS Access list specifies what interface and what MACs can either

Authenticate or Forward. Your interface defaults will apply if you do not have the MAC address in the access list. This is your 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 MAC on multiple interfaces; one could not allow the client to register, and another would, etc.

Access List <u>add</u>				
MAC-Address	Authenticate	Forward	Interface	
00:1E:58-B4-2A:FD	yes	yes	cameranet	delete edit remove

Change Access List Entry

MAC-Address:

00:1E:58-B4:2A:FD

Interface:

cameranet

Authenticate:

yes

Forward:

yes

OK

Cancel

DHCP Server

WebBox has options to specify basic DHCP server information. You can enable the DHCP Server; specify the range and gateway to hand out, as well as the DNS servers to use. You will need to specify the proper interface as well.

DHCP Server is on

Enabled: ☒

Address range: -

Gateway:

Primary DNS Server:

Secondary DNS Server:

Interface:

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

Leases <input type="button" value="Add"/>					
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

Upgrade

L:\MT Versions\routers

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

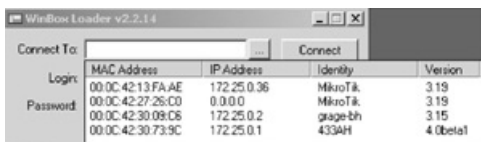
Filename	
routeros-powerpc-3.19.npk	<input type="button" value="remove"/>
<input type="button" value="upgrade"/> <input type="button" value="downgrade"/>	

[Using WinBox](#)

RouterOS has a great utility that comes free of charge, which allows you to have a graphical interface for RouterOS. 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 of your RouterOS system. This page will allow you to download a version of WinBox. I do recommend that you visit their website for the latest version though. The webpage will deliver the latest version, if you have the latest RouterOS version on your router.



WinBox uses either the Router's MAC address or an IP to connect. 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 pressing this, WinBox will use the MNDP packets sent out from RouterOS devices on the local network, Layer2, and display them for your selection.

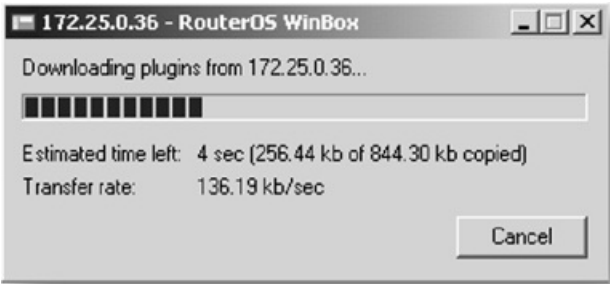


Inside the WinBox Display screen, you have several pieces of information, including the MAC address of your RouterOS device, the Identity and the IP on the interface closest to your PC. You also will receive the Version of RouterOS as well.

If you click on the MAC address, it will place the 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.



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.

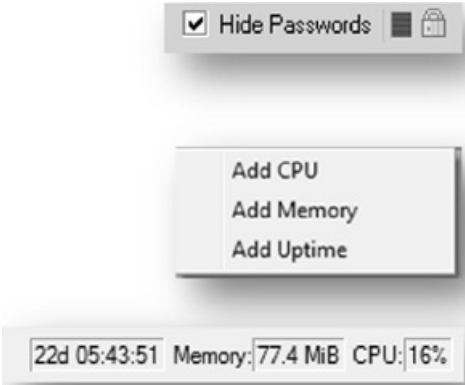


At the top of WinBox in the title bar, you will have a number of details. The username@IP or MAC address of the RouterOS device will be listed at the top. Next, the system identity is displayed, then the WinBox title, along with the current RouterOS version number and what RouterBoard or system the RouterOS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



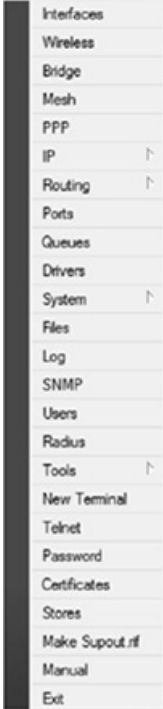
The two arrows on the left side of the screen are undo and redo command buttons. On the right side, we have options to hide passwords, a small green box that shows the CPU load, and a lock to show if we are logged in securely to the RouterOS or not. This is determined inside the WinBox Application before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



WinBox Menus

WinBox is organized into different menus, that allow you to access each of the RouterOS features from. For instance, the interface menu will give you access to the interface options, and settings, while the IP menu choice, will give 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.



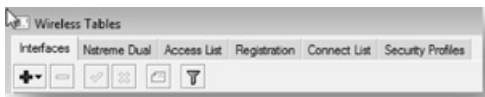
RouterOS organizes its features inside a Directory-Structure like system. Each object on the WinBox menu, has other sub-menus. For instance, if you click on System, you will get the menu to the right. Notice, that we can setup parameters such as clock settings, view system resources and even reboot or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

Identity
Clock
NTP Client
Resources
License
Packages
Auto Upgrade
Logging
History
Console
Scripts
Scheduler
Watchdog
Health
Reboot
Shutdown

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 that each represent 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 interface tab. MikroTik just thought it would be

better to have a tab approach for these items vs having a listing like in the system command. In the command line, you would simply type, *wireless interfaces*.



Below and on the next pages I have created a layout for winbox menus so that you will know where each menu item is. Keep in mind as well though, that this is for version 3.25+ as there was a major menu change to accomidate small resolution laptops!

Interfaces



Wireless



Bridge



Mesh



PPP





Queues

Files

Log

Radius

Tools

Simple Queues

Interface Queues

Queue Tree

Queue Type

BTtest Server

Bandwidth Test

Email

Flood Ping

Graphing

IP Scan

MAC Server

Netwatch

Packet Sniffer

Ping

Ping Speed

Telnet

Torch

Traceroute

Traffic Monitor

New
Terminal

Make
SupOut

Manual

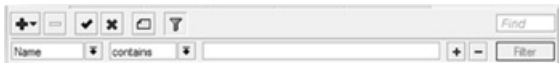
Exit

WinBox Interface Options


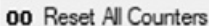
Inside each of these tabbed sections, and all throughout RouterOS, you will find these icons. The left most icon is an ADD icon. If you have a small down arrow, then there are other options than just ADD. In the wireless section, you can add Virtual APs or WDS links, in the Ethernet section; you have the ability to add VLANS or other types of interfaces that pertain to the associated section. You also will see these 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 and X, are 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 a well, by selecting if it contains, does not contain, is or is not as well. Then 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.

A rectangular button with a light gray border and a slightly darker gray background. It contains the text "00 Reset Counters" in a black, sans-serif font. The "00" is bold.A rectangular button with a light gray border and a slightly darker gray background. It contains the text "00 Reset All Counters" in a black, sans-serif font. The "00" is bold.

You will also find sections in RouterOS that contain counter resets. In some sections, such as Firewall rules etc, you will have counters that count packets or bytes. If you select an object, you can reset that individual objects 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 each section. The find will find the selected text and highlight it in the object window below to help you locate objects with certain text.

A rectangular input field with a light gray border. It contains the text "Find" in a light gray, italicized, sans-serif font.A rectangular input field with a light gray border. It contains the text "all" in a black, sans-serif font.

Managing RouterOS

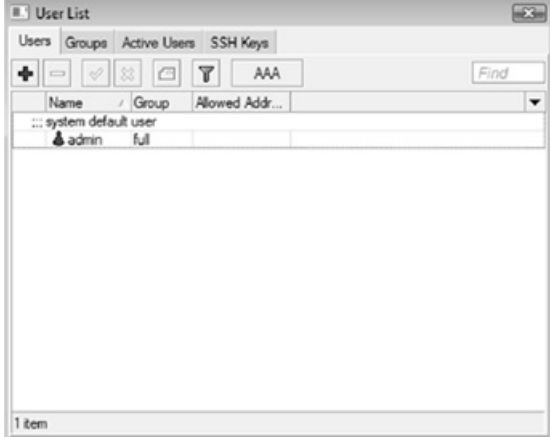
In this section we will cover how to manage your RouterOS installation. This will include managing user's access to your router, controlling basic services that your RouterOS offers, and managing the logging that your RouterOS system generates. This is sometimes a full time job if you have quite a few 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 Defaults

By default RouterOS will install with a user called Admin and have no password. This user will be in the Full User Group, giving you full access to the router.

User Management

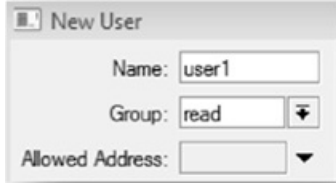
RouterOS has a built in user management system, this is located under the Users section of RouterOS.



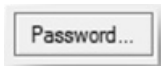
In the user section, you will have a number of tabs, just like the rest of RouterOS. These tabs include the list of users, the groups of users, current active users, and any SSH Keys that you generate.

[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.

A screenshot of a 'New User' dialog box. It has a title bar with a small icon and the text 'New User'. Inside, there are three fields: 'Name:' with the text 'user1' entered; 'Group:' with a dropdown menu showing 'read'; and 'Allowed Address:' with an empty text box and a dropdown arrow to its right.

By 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.



With 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, vs. someone getting the admin username and password, and logging in without your knowledge.

[RouterOS User Groups](#)

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.

Users	Groups	Active Users	SSH Keys
			<input type="text" value="Find"/>
	Name	Policies	
S	full	local telnet ssh ftp reboot read write policy test winbox passw...	
S	read	local telnet ssh reboot read test winbox password web sniff	
S	write	local telnet ssh reboot read write test winbox password web ...	

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.

New Group

OK

Cancel

Apply

Comment

Copy

Remove

Policies

☐ local
 ☐ telnet

☐ ssh
 ☐ ftp

☐ reboot
 ☐ read

☐ write
 ☐ policy




☐ test
 ☐ winbox

☐ password
 ☐ web

☐ sniff

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.

Users Groups Active Users SSH Keys					
					<input type="text" value="Find"/>
	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	

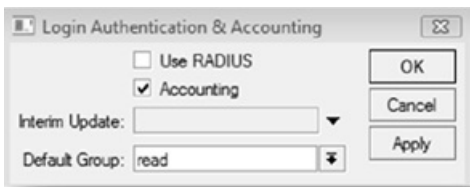
[SSH Keys](#)

SSH keys are used to authenticate sessions without using a username/password. By importing a DSA key here, and your SSH session having the corresponding key. You will import keys here. 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.

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.



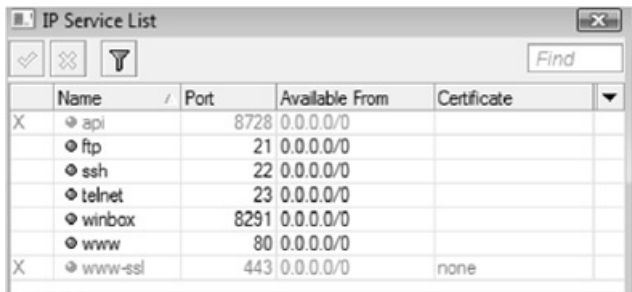
[AAA Settings – Radius RouterOS Users](#)



With the AAA system, you can set your RouterOS to use a Radius server to allow users to login. With this, you can have a centralized radius system for router management. The users that you have in the radius 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.

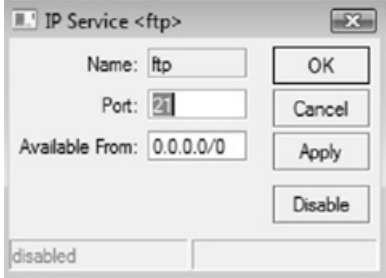
RouterOS Services

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



	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	

By going to IP → Services, you will be able to turn on and off these services as well as change ports, and change from what IPs they are available from. Each one of these objects can be turned off or on, by disabling or enabling. 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.



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.

[FTP Service](#)

FTP is used to allow the transfer of files to and from RouterOS. There are other ways as well to transfer files and 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.

[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 Availability 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 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 already. This will allow you to run SSL on the web server.

WinBox Service

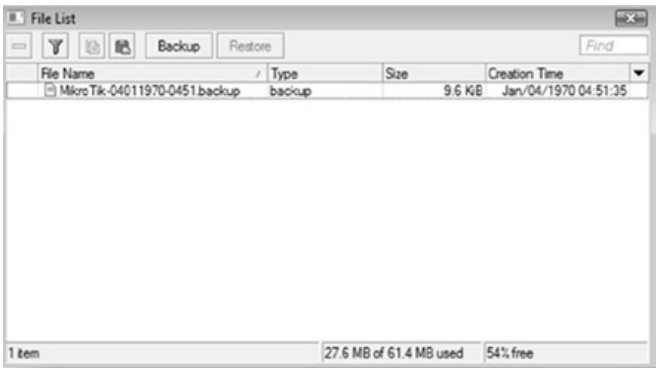
WinBox by default runs on port 8291. Inside the IP services system, you can change this port as well as change where it is available. Normally, I would secure this with Firewall rules, like other items. I typically though, leave it on the default port.

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 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 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.

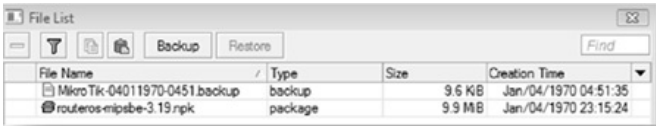


As you can see, you have information, such as how many items that you have inside your file system, as well as space information on the bottom of the window. Here you can select an object and delete it by using the minus button at the top.

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, etc, right into the file list window!



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.



The backup file that we created by clicking the backup button, we can now simply, click and drag onto our desktop or file folder. It will then download from RouterOS.

Something to note about downloading and uploading files in RouterOS through WinBox, you will typically need an IP or layer 3 connection. Sometimes the Layer2 connection can be a bit flaky, and unreliable. I have seen where it will stall, stop and hang on some computers. I would recommend putting an IP on your Router, then connecting with the IP address through WinBox and then upload your files.

[Backup / Restore](#)

In the File List window, you also have backup and restore options. Backing up RouterOS is simple as clicking the backup button. When you click the backup button, you will see that there is a .backup file created. This is your 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 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 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.

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=yes \
comment="" 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=ether1
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=ether2 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=ether3 speed=100Mbps
/interface vlan
add arp=enabled comment="" disabled=no interface=ether2 mtu=1500
name=\
vlan100.2 vlan-id=100
add arp=enabled comment="" disabled=no interface=ether3 mtu=1500
name=\ vlan100.3 vlan-id=100
/interface wireless security-profiles
set default authentication-types="" eap-methods=passthrough group-ciphers="" \
group-key-update=5m interim-update=0s mode=none name=default \
radius-eap-accounting=no radius-mac-accounting=no \
radius-mac-authentication=no radius-mac-caching=disabled \
radius-mac-format=XX:XX:XX:XX:XX:XX radius-mac-mode=as-username \
static-algo-0=none static-algo-1=none static-algo-2=none static-algo-3=
```



```
none static-key-0="" static-key-1="" static-key-2="" static-key-3="" \  
static-sta-private-algo=none static-sta-private-key="" \  
static-transmit-key=key-0 supplicant-identity=MikroTik tls-certificate=  
none tls-mode=no-certificates unicast-ciphers="" wpa-pre-shared-key="" \  
wpa2-pre-shared-key=""
```

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.

Importing Scripts

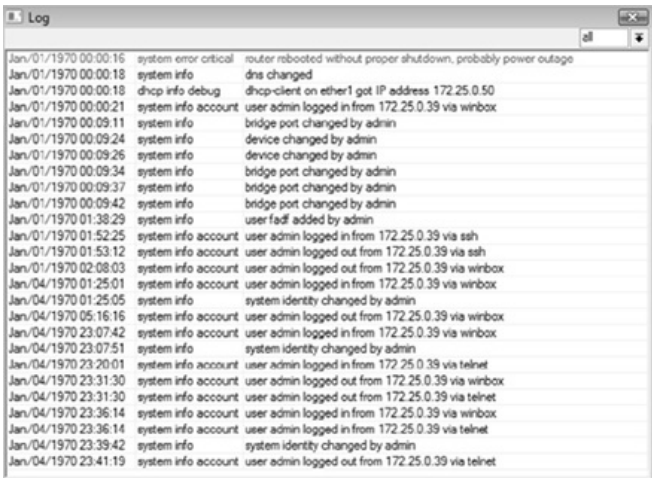
Once you get real good at reading and doing command-line interfaces, you can start creating scripts, or RSC files that you can bring right into RouterOS. You will need to create this file, and of course test and test again. Once you have it just the way you want it, then go ahead and upload the file. 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.

```
[admin@LearnRouterOS] > import export  
Opening script file export.rsc  
  
Script file loaded and executed successfully
```

Logging

Just like with other Routing systems, you have logging capabilities. You will use this to review access to the router, 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.



The screenshot shows the WinBox 'Log' window. It has a title bar with 'Log' and standard window controls. Below the title bar is a search bar with the text 'all' and a dropdown arrow. The main area is a table of log entries. Each entry consists of a timestamp, a log level/type, and a description of the event.

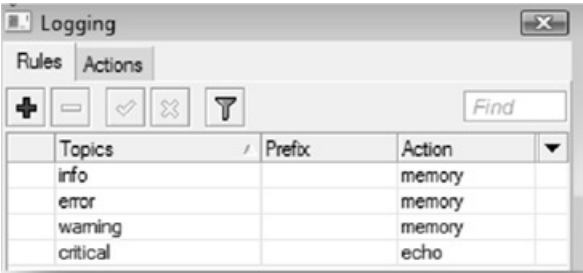
Timestamp	Log Level/Type	Description
Jan/01/1970 00:00:16	system error critical	router rebooted without proper shutdown, probably power outage
Jan/01/1970 00:00:18	system info	dns changed
Jan/01/1970 00:00:18	dhcp info debug	dhcp-client on ether1 got IP address 172.25.0.50
Jan/01/1970 00:00:21	system 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

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. I would think that these are the minimum that I would have on a Router. The ones that I would use normally are Radius and hotspot logging

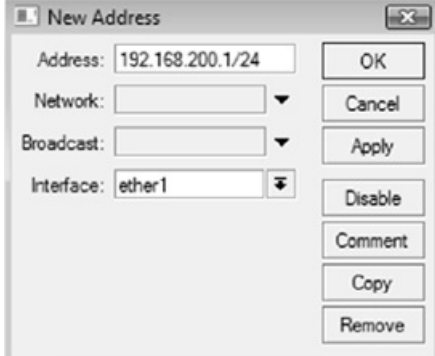
Basic RouterOS Setup

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 these, you will need to know where you can find them and how to configure them.

Configuring IP Addresses

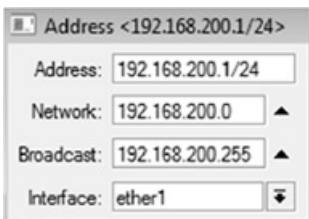
We are dealing with a Router right? Well then we will need some IP addresses to go on our Router. Now, we are not going to get into talking about sub netting and TCP/IP right here in this book, but we are going to at least get you on the Internet with some basic IP information.

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 and the third is what interface



A screenshot of a 'New Address' dialog box. It has a title bar with a close button. The dialog contains four input fields on the left: 'Address' with the value '192.168.200.1/24', 'Network' (empty), 'Broadcast' (empty), and 'Interface' with the value 'ether1'. Each field has a dropdown arrow to its right. On the right side of the dialog, there are six buttons stacked vertically: 'OK', 'Cancel', 'Apply', 'Disable', 'Comment', and 'Remove'.

you wish to place that IP on.



A screenshot of an 'Address' dialog box. The title bar shows the title 'Address' followed by the current address '<192.168.200.1/24>'. The dialog contains four input fields: 'Address' with '192.168.200.1/24', 'Network' with '192.168.200.0', 'Broadcast' with '192.168.200.255', and 'Interface' with 'ether1'. The 'Network' and 'Broadcast' fields have upward-pointing triangle buttons to their right, while the 'Interface' field has a dropdown arrow. There are no buttons on the right side of this dialog.

I get plenty of comments on 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 accustomed 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 as putting in all of those 255's. Using the Triple-255-dot-zero is called the Dotted-decimal Notation method. Another method, that RouterOS uses, is the

CIDR, or Classless Inter-Domain Routing method. This method uses the /24 to notate how many subnet mask bits are on. If you convert 255.255.255, to decimal, and count the ones, you will get 24, hence, 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 placed in 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 these fields for you. I do recommend that you allow RouterOS to do this for you as it will prevent human error issues normally.

[Common IP Information](#)

I wanted to do a quick review, as 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 IANA Reserved Private Network ranges are as follows:

[24 bit Block or a /8 Prefix](#)

10.0.0.0 through 10.255.255.255 – 16, 777,216 Total IPs

[20 bit Block or a /12 Prefix](#)

172.16.0.0 through 172.31.255.255 – 1,048,576 Total IPs

[16 bit Block or a /16 Prefix](#)

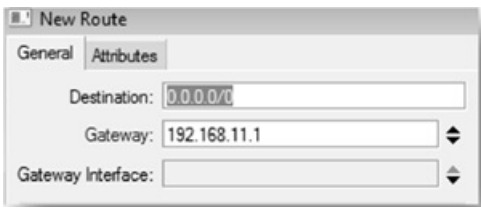
192.168.0.0 through 192.168.255.255 – 65, 536 Total IPs

These blocks are set aside just for private network use. 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. When you use a private address, these IPs 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.

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 otherwise 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 give you access to the routing table, and allow you to click the plus sign and create a new route.



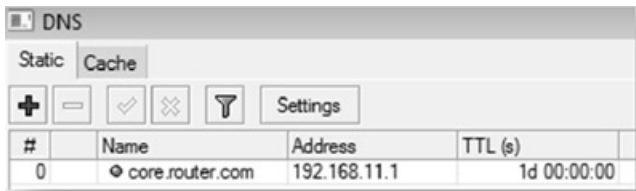
Above, you will see the destination of all zeros, or 0.0.0.0/0. This means all networks with any subnet mask. What we call a default route. You will need to enter the default gateway address for your network under the gateway setting

[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 DNS servers; 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 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 is that as your DNS lookup hits your RouterOS and it does have the information that you need in cache, your DNS lookups are a few milliseconds vs. 30-50 milliseconds just for the round trip time up to the next public DNS server.

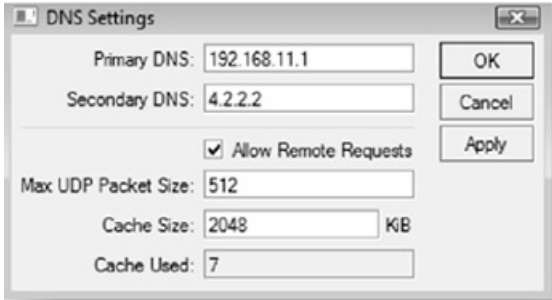
You will access your DNS system, by clicking on IP → DNS.



The screenshot shows the RouterOS DNS configuration window. The 'Cache' tab is selected. Below the tabs are several icons: a plus sign, a minus sign, a checkmark, an 'X', a funnel, and a 'Settings' button. Below these icons is a table with the following data:

#	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.



DNS Settings

Primary DNS: 192.168.11.1

Secondary DNS: 4.2.2.2

☒ Allow Remote Requests

Max UDP Packet Size: 512

Cache Size: 2048 KB

Cache Used: 7

OK

Cancel

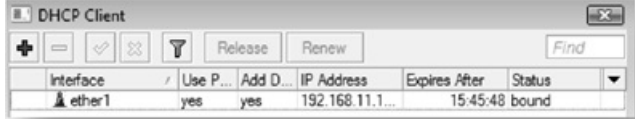
Apply

Note above, we have both primary and secondary DNS server that we can enter. Also, more importantly, is a check box to *Allow Remote Requests*. This check box will make your RouterOS act and respond to remote DNS services. If you don't check this box, the DNS settings here will be strictly for your RouterOS services and usages, vs. other customers or clients.

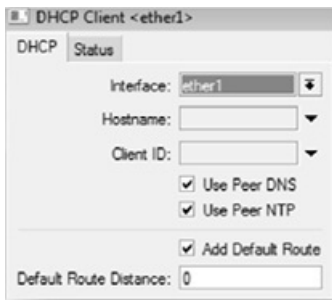
[DHCP-Client](#)

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 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 this 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.



Above you will see the DHCP client, running on Ether1. On the right you will see the options that you have to select when you add a DHCP-Client. The main item that you will need to select is what 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 get the Peer DNS, NTP and default route. We are also not going to make any changes to the route distances here as well.



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.

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

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

[DHCP-Server](#)

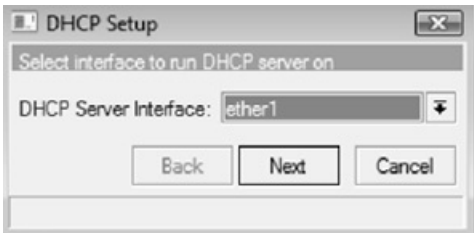
Just like the above, RouterOS has the ability to become a DHCP-Server, handing out IP configurations for client usage. You can have multiple DHCP-Servers on different interfaces handing out different IP scopes for you, as well as have DHCP-Clients running on other interfaces. You can only have one DHCP 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.

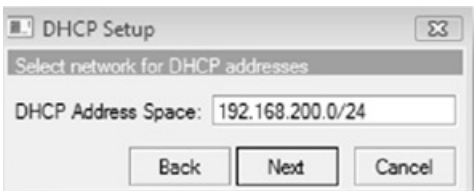


To access the DHCP-Server menu, you will click IP → DHCP-Server. DHCP-Servers are not complicated to setup, but there are a number of functions and pieces of information that must be obtained and setup for them to work. Due to this fact, RouterOS has created a wonderful DHCP Setup button that we can 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.

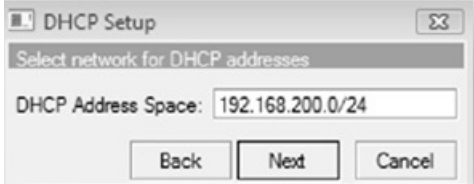
[DHCP Server Wizard](#)



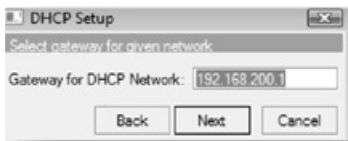
Let's run through the wizard so that you understand all of the information and questions that RouterOS asks during the setup wizard. Start by clicking the DHCP Setup button. Then it will ask what interface you wish to run the DHCP-Server on. Remember, DHCP Servers run on an interface. You typically will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



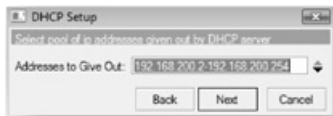
The next step will ask you for your DHCP Address space. This typically will be filled in for you if you have your IP already on the interface that you selected. This is basically the subnet that the DHCP Server will run on.



Upon clicking on Next, you will have the option for 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.



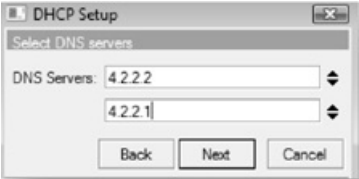
Next we will have our Addresses to give out. This is a pool of IPs that will be given to your clients as they request them. By default, RouterOS will say all of the IPs in the subnet other than the IP of your router. In this case, our router is 192.168.200.1, so it is defaulted to giving out 192.168.200.2 through 192.168.200.254.



If you are running a business network you may need to have some IPs that are 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 this to something like 100-200 as the range.

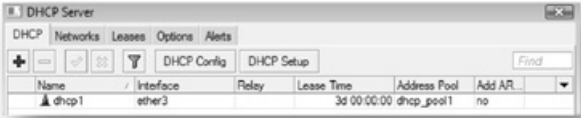
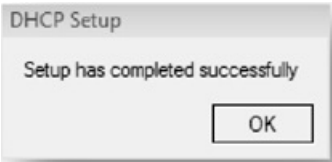
The next section is the DNS server setup. As we said in the DHCP-Client section, we can hand out the DNS servers that we wish our clients to use. Here we can enter the DNS servers to hand out. It could be the Local IP of our MikroTik, so we could put in 192.168.200.1 as our primary DNS server and add 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 may perform another DHCP request, and very well may get the same IP. However, if there is a break in time after the lease time is up and the computer does another DHCP request, then that IP may have went back into the pool of addresses and been handed out to another DHCP Client.



There are a number of thoughts to the lease time. Typically DHCP traffic is minimal, so more 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 will wish to put this 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 high lease time. I would always side on a lower lease time than a higher one, as the worst it can do have the DHCP client renew their lease. This does not generate much traffic and doesn't affect clients.



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 Lease time and what pool of IP addresses it will use. You also have options here to select to add ARPs for the leases that you have, as well as the ability to use Radius.

DHCP Server <dhcp1>

Name:

Interface:

Relay:

Lease Time:

Address Pool:

Src. Address:

Delay Threshold:

Authoritative:

☒ Bootp Support

☐ Add ARP For Leases

☐ Always Broadcast

☐ Use RADIUS

OK

Cancel

Apply

Disable

Copy

Remove

DHCP Server

DHCP Networks Leases Options Alerts

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 have all of the network settings. As you can see by 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.

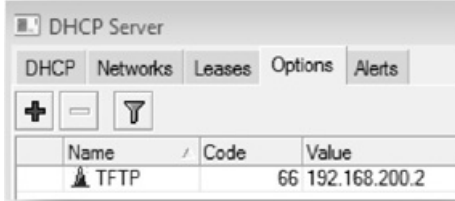
The image shows a configuration window titled "DHCP Network <192.168.200.0/24>". It contains several input fields for network configuration:

- Address:** 192.168.200.0/24
- Gateway:** 192.168.200.1
- Netmask:** (empty)
- DNS Servers:** 4.2.2.2
- DNS Domain:** (empty)
- WINS Servers:** (empty)
- NTP Servers:** (empty)
- DHCP Options:** (empty)

Each field has a small icon to its right: a double-headed arrow for Address, Gateway, DNS Servers, NTP Servers, and DHCP Options; and a downward-pointing arrow for Netmask and DNS Domain. The WINS Servers field has a double-headed arrow.

Double clicking on the DHCP Network object, will allow 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 say that this Network has this DHCP Option, in this case the TFTP name would show up in the DHCP Options section. You can specify several DHCP options as needed.



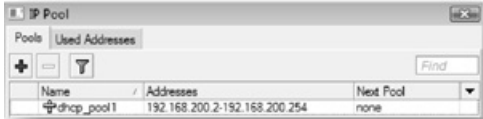
The DHCP Setup Wizard does quite a few things, real quick, let's review them here.

- What does the DHCP Server Setup Create?
- DHCP Server Interface
- What interface to run On, What Lease time to Use, What IP Pool to use
- DHCP Network Settings
- What Gateway to hand out, What DNS Server and Other DHCP Network Options
- IP Pool
- Creation of a pool of IP addresses to hand out.

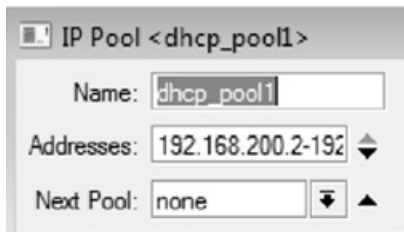
IP Pools

The wizard also creates IP pools, these are pools of IP addresses that your RouterOS system will allow you to assign out of. This is automatically created for you, but you should know where it is created.

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



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 range that it gives out. There is also an option for the Next Pool. This option says what pool to go to once this pool is out of IP addresses.



Masquerading - NAT

The NAT or Network Address Translation system inside RouterOS is very advanced. What we are going to focus on is just one function called Masquerading. What 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. So we will refer to them as separate items.

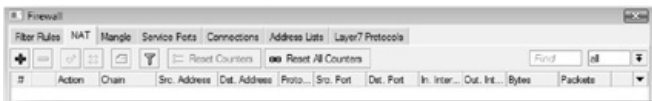
We don't need to go into the interworking knowledge of how Masquerading is accomplished, however, it is important to know that, from the Internet's perspective, looking back at your 100 clients, what we will see is just that single public IP address that we were assigned. All of the traffic will be coming from that 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 individual device behind that Masquerade. It hides those private addresses, and because of that, you can't directly connect them.

Configuration of basic Masquerading

IP	└	Addresses
Routing	└	Routes
Ports		Pool
Queues		ARP
Drivers		Firewall

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.

We are going to need to create a basic Masquerade. We will assume our Internet connection is on ether1, and our private network is on Ether2. 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**



New NAT Rule

General Advanced Extra Action Statistics

Chain:

Src. Address:

Dst. Address:

Protocol:

Src. Port:

Dst. Port:

Any. Port:

In. Interface:

Out. Interface: ☒

Packet Mark:

Connection Mark:

Routing Mark:

Connection Type:

NAT Rule <>

General Advanced Extra Action Statistics

Action:

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 ether1, we are going to setup this rule to match on our Out Interface using ether1. We use a chain of srcnat. 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 Ether1, 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

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 network, to gain access to the Internet through a single Internet connection.

- **What We Know**

- ☐ DHCP Internet connection
- ☐ Several computers for our home network
- ☐ Internet Connection is on Ether1
- ☐ Private computers will be on Ether2

- **Features we will need to use**

- ☐ DHCP-Client
 - To get the IP information from our Internet provider
- ☐ DHCP-Server
 - To assign private addresses to our computers inside our network
- ☐ Masquerading
 - To translate the many private IPs on our computers inside our network to the single public that we will receive from our provider.

- **Here are the steps we will take:**

- ☐ Login to RouterOS
- ☐ Set your Private IP on ether2.
- ☐ Setup DHCP-Client to run on ether1

- ☐ This will obtain your Default Route and DNS information
- ☐ Setup DHCP-Server on ether2.
 - DNS information will be filled in since we obtained it from our DHCP-Client
- ☐ Default Gateway will be our router
- ☐ Setup Masquerading out the Ether1 Interface
- Create rule, out ether1, action Masquerade

[Home Router Walkthrough](#)

Step 1: Login to your Router

Step 2: Set your Private IP on ether2. We will use 192.168.200.1/24 for your private range. Click IP → Addresses → Plus Sign. Add IP address to ether2.

The image shows two overlapping network configuration windows. The top window, titled 'New Address', has four fields: 'Address' with the value '192.168.200.1/24', 'Network' (empty), 'Broadcast' (empty), and 'Interface' with the value 'ether2'. The bottom window, titled 'New DHCP Client', has two tabs: 'DHCP' (selected) and 'Status'. It contains several fields and checkboxes: 'Interface' with 'ether1', 'Hostname' (empty), 'Client ID' (empty), 'Use Peer DNS' (checked), 'Use Peer NTP' (checked), 'Add Default Route' (checked), and 'Default Route Distance' with the value '0'.

New Address

Address: 192.168.200.1/24

Network:

Broadcast:

Interface: ether2

New DHCP Client

DHCP **Status**

Interface: ether1

Hostname:

Client ID:

☒ Use Peer DNS

☒ Use Peer NTP

☒ Add Default Route

Default Route Distance: 0

Step 3: Setup DHCP-Client on Ether1. Click IP → DHCP-Client → Plus Sign. Select interface ether1. We will use all of the peer information as well as the default route from our provider, so leave these checked.

[Verify that we obtained an IP address](#)



Step 4: Setup DHCP-Server on ether2. Click IP → DHCP-Server → DHCP Setup Button. Follow the DHCP Setup Wizard. Select Ether2 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 Ether2. Then leave the defaults 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 Ether1. Select IP → Firewall → NAT tab → Click the Plus sign. The chain will be SRCNAT, out interface will need to be set to ether1, and then click on the action tab. Drop down the actions and select masquerading



Once this is done, you can now plug a computer or device into ether2, 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 your RouterOS. Some of the material in this section, we will talk on briefly, but in the RouterOS Features section, we will go in to much more depth. We will do 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, find the best channel for your usage, as well as setup the band that you wish to operate in. Then create a SSID that you wish to have. 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 NAT and DHCP, well, simply put, if you are allowing your access point to be a simple bridge, I would have those features on

your router below your access point vs. on your access point. There is no need to have more services on it. 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.

CPE – Client Premise Equipment **Configuration**

CPEs are mainly used at a subscriber's home for WISPs or Wireless ISPs. These configurations may not be the best for your business however the bridged client and access point can be used to create a layer 2 back haul between buildings or towers. For WISPs, I will always recommend the routing/NAT setup for a CPE vs. client bridges. The reason is that it keeps the clients from connecting gear on your network; you will be providing them a private subnet that is just for them. If they plug in something incorrectly, creating a bridging loop, or have a virus that has 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.

First, create your bridge group on your client system. Add both the Ethernet and the wireless interface to the bridge group. Even though adding the wireless client is not typically necessary, I do anyways. Next, configure your wireless interface for a mode of station-wds. This mode will form a station relationship along with WDS to your access point. Setup 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 in order for the unit to register to the AP. You will also need to configure the WDS settings; I would recommend dynamic WDS along with adding the bridge group you first created as the default WDS bridge group.

Once you create this and your CPE associates, you should see a WDS interface with the MAC of your CPE on your access point. Also, in the bridge group you should see a WDS interface dynamically created on your CPE as well. This will bridge the Ethernet 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 spec.

[Routed / NAT CPE](#)

The best way to configure a CPE is to have a Routed or NAT mode. These really are two different modes. Routing allow you to place a publicly or privately routable IP address that can be routed through your network. This typically is the best way, however, for customers that do not need publicly or privately routed subnets, most residential and/or business connections, 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!

To do this, you will need to configure your CPE's wireless interface in station mode, just like any other client device. Then 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 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!

RouterOS Features

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”. An example is that 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.

IP Features

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 I get all of the time. When I bridge, putting IP addresses are easy and things work, why should I route? Let me answer this question with two comments. One, bridging and IP addressing are very easy to manage, and run. However, it is not a matter of if it will fail, 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 you should route? The Internet is routed for a reason. Failures cause topology changes, and just like the Internet you should have routed traffic be able to fail over to other connections and links to make them redundant. Bridging will allow some redundancy, however, typically at the expense of turning OFF links. Preventing a bridging loop ends up disabling ports, entire links are wasted. With routing you can have some traffic, go over a primary connection, and other traffic go over another, so you actually use the hardware 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 500 devices on the network. ARP entries should be limited to just what is needed to communicate. This also has another benefit, and that is to be able to handle ARP and Broadcast storms. You are limiting the size of your broadcast domain. Due to this, you also limit the effects of these types of issues to a much smaller area. Backbone connections should not be affected by this type of traffic as they do not need ARPs etc to go across them from customer routers, CPEs or other devices. Read the VLAN section as well as it will talk about how VLANs do not keep traffic separate on physical networks.

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 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, layer 2. This translation is held in the ARP list. See the following section for more information about the ARP list.

Depending on the interface that you have, you typically will have only a few ARP options. The main options, that you have is 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. So, if 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 of a MAC of an IP, it will send an ARP request out to find the MAC for an IP.

So what are the uses for the other modes? Some administrators will use the ARP Disabled mode as a form of security. With this in the disabled mode, you will not send out ARP requests, or reply to it. Because of this, you will have to add ARP entries in your ARP list manually. On the other device, you will have to do this as well. This requires manual ARP entries on both devices, but works quite well.

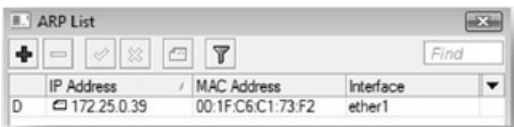
With that said, some devices don't have manual ARP tables and you can't make a manual ARP Entry. The Reply-Only ARP mode, will allow for this. In this mode, your RouterOS will reply to ARP requests, but will not send out ARP requests. So your remote device will send out the ARP request, and your RouterOS 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.

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 does work. What happens is the interface that you have setup for Proxy-ARP, takes any ARP requests it receives and forwards them on other interfaces. If another device on the other interfaces has this IP, that ARP request is replied to. However, the MT translates this. The interface with Proxy-ARP will say it has the IP, but then translate it to the other interfaces MAC. So think of it as a Masquerading of MAC and IPs at layer 2.

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

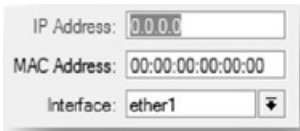
On a professional recommendation, I typically stay away from this method; see the NATing section for information on how to NAT public IP addresses vs. using Proxy-ARP. However in a pinch, this will work. If at all possible, I would rather have routed subnets. When you have trouble on your network, look 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. Below is an example of the ARP list. Note that we have a D next to the entry in this unit. That means this was a dynamically created entry. We get dynamic entries by your Mikrotik sends out ARP requests and receive the reply from the device.



IP Address: 0.0.0.0

MAC Address: 00:00:00:00:00:00

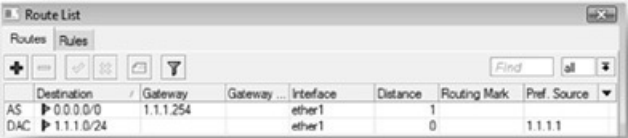
Interface: ether1

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

MikroTik also gives you tools such as Ping, MAC Ping, etc, to help you with 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.


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 Routes that 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

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 a few types of routes that I wish to cover a bit more as well. 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. 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!

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.



The image shows a 'New Route' configuration window with two tabs: 'General' and 'Attributes'. The 'General' tab is active. The fields are as follows:

- Destination: 0.0.0.0/0
- Gateway: (empty text box)
- Gateway Interface: (empty dropdown menu)
- Interface: (empty text box)
- Check Gateway: (empty dropdown menu)
- Type: unicast
- Distance: (empty dropdown menu)
- Scope: 30
- Target Scope: 10
- Routing Mark: (empty dropdown menu)
- Pref. Source: (empty dropdown menu)

In v3 of RouterOS we also have 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 is only one other IP

address to use in the /30 subnet. Your router will have one, and the other will be used as the Gateway. This interface routing makes it very easy to route out just by using an interface name vs. having to know the IP address.

Checking Gateways

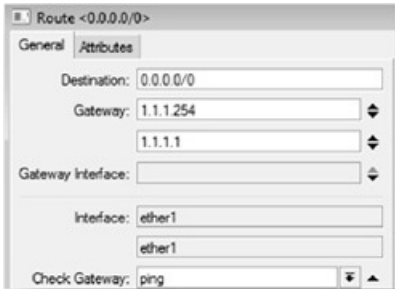
Check Gateway give you the ability to verify that the gateway is available. It has two options, ping or ARP. If the router does not have an ARP for the gateway, it will consider it unavailable. This makes the route turn Blue, it would be active, but in this case, the gateway is not on-line or otherwise unavailable. In most cases you would use the ping option, 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 entry.

Using Distances

The distance is also useful to us. This is the distance that your static route has, or its cost. If you use an example of two different Internet connections, and one has a distance of 2 and one has a distance of 1, the route with a distance of 1 will be preferred over the route with a distance of 2. Simple right, the one with the lower distance is preferred. The route with the higher distance will be blue. It is a valid route, but there is another route that is preferred, just due to its lower 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. This system basically says that two gateways are of the same cost. In the example to the left, you will see that we have specified multiple gateways. Because of this, we will balance our connections going out between both gateways.



Note that I used the word balance connections out the gateways, this will not balance bandwidth. It does this by matching source/destination IPs up. Once computer 1 establishes a connection, that source/destination IP information will stay on one interface until that connection is done. 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 other 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 what IP address you are coming from to ensure encryption. This will break HTTPS, so make sure you don't do this on that type of traffic.

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

By 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 show this.

If you have an unbalanced connection, i.e. one gateway has 2 Meg vs. the second having 1 Meg, you can list the same gateway multiple times. If you had this setup you would place the gateway with the 2meg service in the list twice, and the other 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 wish to control how traffic flows. The basics on this, is that you will have a policy rule, that will match data in some way, that then says what routing table to use. With this, you can send data from one customer through an anti-virus/anti-spyware box and other customers directly out to the Internet. I have used this as well in enterprise applications where I want remote site users to go to the main site for Internet access vs. 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. Second, you have to give that traffic a place to go. In this case, we use a separate routing table. This table is distinguished by a routing mark. The last thing you need is a routing rule, which basically says, if it is the data that you have identified then use this table.

So the things you need are: Traffic Identification, another routing table, and a policy.

Routing Policies

There are two ways to identify your traffic; one is doing 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. Also, the 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.



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

New Route

General **Attributes**

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 use 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 call 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 adding the normal destination for a default route, and setup a proper gateway. Below we changed our Routing mark to the alt_table.

Route List

Routes Rules

Find alt_table

	Destination	/	Gateway	Gateway ...	Interface	Distance	Routing Mark	Pref. Source	
AS	0.0.0.0/0		172.25.0.1		ether1		1 alt_table		

Above we see this rule added, note, that we click 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.

Route List

Routes Rules

Find all

	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 is possible, due to the fact that the first one has a routing mark that we are using to move it to another routing table.

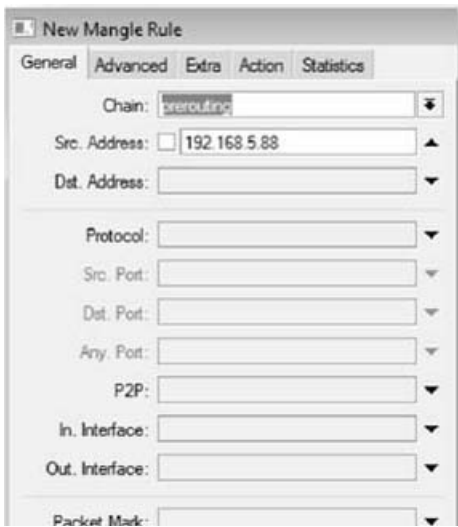
QUICKTIP: When using multiple routing tables, secondary tables (tables that are not the main table) will failover to the main table if there is not a matching route. However, if you use a default route in your secondary table, it will never fail to the main table.

[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.



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



Connection Mark: ▼

Routing Mark: ▼

Connection Type: ▼

Connection State: ▼

New Mangle Rule

General Advanced Extra Action Statistics

Action: ▼

New Routing Mark: ▼

☐ Passthrough

To do this, click on the action tab, and select Mark Routing as the action. You can then define a 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.

Firewall Features

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. It is used to not only prevent unauthorized access to your router, and your network, but also can be used to prevent unwanted or unnecessary data from flowing around in 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 get to the desired result. If I had one thing to say about firewalling it 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 one will be 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 amount 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 protocol, 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 watch for just cars on this highway, going out of our city, on 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 say, I want this car, with this flag on this road, going to this highway on this lane. Not only though, can you identify through set rules like this, but also setup methods to match only a percentage of traffic, connection counts, and may other methods as well. We also have a tool called Torch; we will cover this in the tools section, which will help you identify the traffic as well.

Once you have identified this 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 your Mangle system.

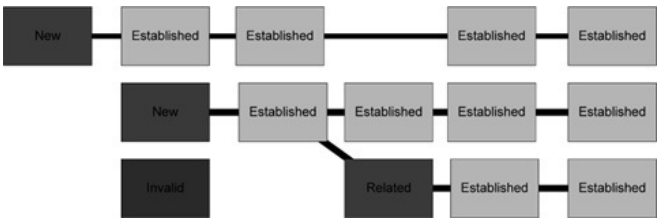
Understanding Connection States

You need to understand connection states for your firewalling as well. There are four types of connection states in RouterOS, Invalid, New, Established and related. Normally, when a connection is established between point A and B, it goes through two connection types. One is a 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 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.



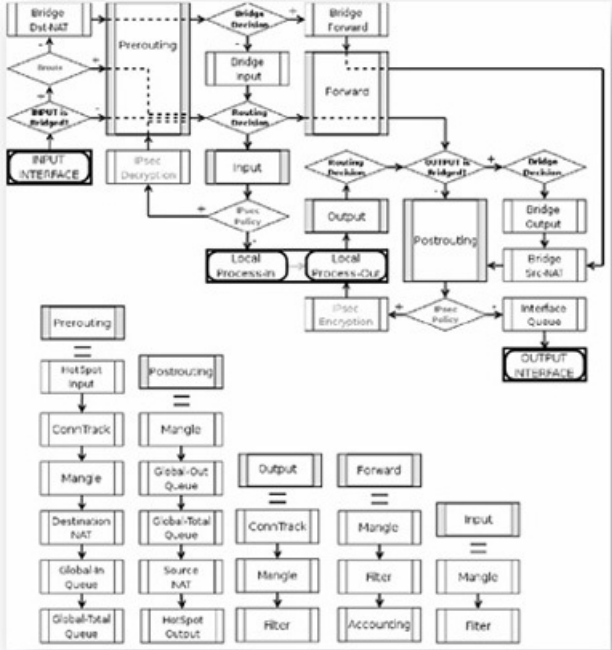
This is important to understand with the RouterOS firewall as it gives you the ability to understand how connections are created. Invalid connections are typically hacker attempts! Also, instead of processing on every packet that passes through your router, you can process on the new connection states only. 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 these 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?

Do you want 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 the amount of CPU usage that you use, as well as still accomplish 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 RouterOS packet flow diagram. You will need this to understand how packets flow.



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 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. With this, you can manage each portion of your RouterOS firewall simpler 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 to 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 your 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 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 name 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, though mentioned at the beginning of the session, you can also help 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 your three built-in chains, input, output and forward. For organization and other reasons, you build other chains that you create names for. For you to use these 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.



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 say jump to our web_server chain.

General Advanced Extra Action Statistics

Action:

Jump Target:

We have created a rule, and told it to jump to our web_server chain. Note that we have added the 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.

General Advanced Extra Action Statistics

Chain:

Src. Address:

Dst. Address:

Protocol: ☐ 6 (tcp)

Src. Port:

Dst. Port: ☐ 80

General Advanced Extra Action Statistics

Action:

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.

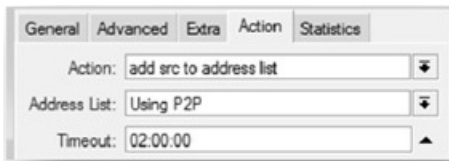
An example of jumping and needing to return may be something like a connection limit chain. You jump to the connection limiting chain to apply a few connection limiting rules, just as an organizational method, at the bottom of your connection limit chain, you can apply a rule called return. This returns 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 in which you jumped from.

Address Lists

Address lists are an extremely powerful feature of RouterOS. The address lists gives you the ability to provide a list of addresses, a single address, 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 these lists. To access your 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, they are added to the address list of your choosing for a specified time.



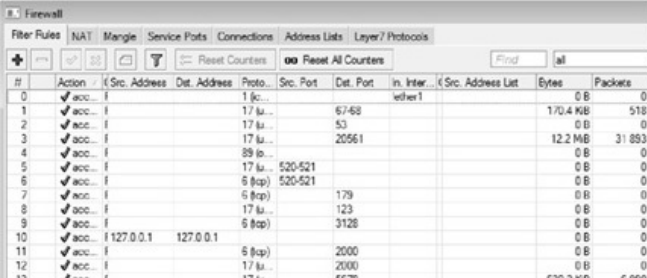
An example of using this feature is to have an address list dynamically get created as users use Peer to Peer applications. This then will create an address list dynamically giving you the list of users that are using Peer to Peer applications. The timeout

value in your firewall rule will determine how long they 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 example is to create lists of possible hackers. The methodology that you will use is detected, add, block. First you detect that a hack attempt. You do this by placing firewall rules that identify this type of traffic, and once identified, you will add their IP to the address list. Once this is done, now you have an IP address list that you can create another rule. I typically will then say, if you are on this address list, even if you 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. So once you are listed, you can't get past my firewall until your timeout value has been reached, and you are off the list.

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. So make sure you process them from the top down. I typically will accept at the top and drop at the bottom.

A screenshot of the RouterOS Firewall configuration window. The 'Filter Rules' tab is selected. The table shows 13 rules. Rule 0 is the default rule, 'action: accept', 'protocol: all', 'in. interface: ether1', 'bytes: 0 B', 'packets: 0'. Rules 1-13 are 'action: accept', 'protocol: all', 'in. interface: ether1', 'bytes: 0 B', 'packets: 0'. Rule 10 has 'src. address: 127.0.0.1' and 'dst. address: 127.0.0.1'. Rule 11 has 'protocol: 6 (tcp)'. Rule 12 has 'port: 2000'. Rule 13 has 'port: 5678'.

#	Action	Src. Address	Dst. Address	Proto...	Src. Port	Dst. Port	In. Inter...	Src. Address List	Bytes	Packets
0	✓ acc...			1 (c...			ether1		0 B	0
1	✓ acc...			17 (u...		67-68			170.4 KB	518
2	✓ acc...			17 (u...		53			0 B	0
3	✓ acc...			17 (u...		20561			12.2 MB	31 853
4	✓ acc...			89 (o...					0 B	0
5	✓ acc...			17 (u...	520-521				0 B	0
6	✓ acc...			6 (tcp)	520-521				0 B	0
7	✓ acc...			6 (tcp)		179			0 B	0
8	✓ acc...			17 (u...		123			0 B	0
9	✓ acc...			6 (tcp)		3128			0 B	0
10	✓ acc...	127.0.0.1	127.0.0.1						0 B	0
11	✓ acc...			6 (tcp)		2000			0 B	0
12	✓ acc...			17 (u...		2000			0 B	0
13	✓ acc...			17 (u...		5678			630.2 KB	6 990

With these rules you have the basic IP matching ability's right on the general tab. This will give you the ability to match based on source and destination addresses, protocol as well as source, destination ports, and 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 marks and firewall rules.

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 under the advanced tab, or you can say NOT this address list, by checking the ! box.

GeneralAdvancedExtraActionStatistics

Chain: input

Src. Address:

Dst. Address:

Protocol:

Src. Port:

Dst. Port:

Any. Port:

P2P:

In. Interface:

Out. Interface:

Packet Mark:

Connection Mark:

Routing Mark:

Connection Type: ☐ http

Connection State: new

GeneralAdvancedExtraActionStatistics

Src. Address List:

Dst. Address List:

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. If 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 what you want!

[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 a connections transferred amount of data.



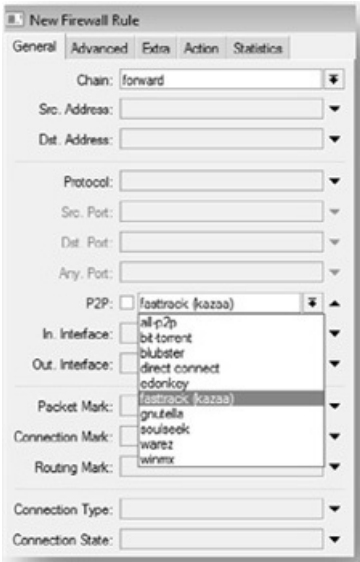
The image shows a screenshot of a firewall rule configuration window. It has three input fields with labels and dropdown arrows. The first field is labeled 'Content:' and is empty. The second field is labeled 'Connection Bytes:' and contains the text '1048576-0'. The third field is labeled 'Src. MAC Address:' and is empty. The 'Connection Bytes' field has a small upward-pointing arrow on its right side, while the others have downward-pointing arrows.

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

Now that you have this rule, you can do something with it. I sometimes will do a connection or packet mark, with a special rule, that puts it into an extended download queue. Everyone can fight over so much bandwidth in one queue for these extended downloads. Normally though, it would not be 1 Meg, it would be something like 200+ Meg for most of my configurations, but it is a preference.

[Built-In Peer to Peer Filtering](#)

RouterOS has a great following with ISPs (Internet service providers) and WISPs (Wireless Internet Service Providers). In many cases P2P or Peer to Peer applications are disruptive to some services, creating many packets per second and using up quite a bit of Access Point Time. RouterOS gives you the ability to match your firewall rules to built-in P2P Filters. These filters are extremely optimized Layer7 filters. They are constantly being updated by RouterOS, due to this; the latest version will give you better matching vs. older versions.



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, and do

whatever you wish to. You can apply other options to do connection limiting on this, or even drop your traffic. In the address list section, we mention that you can also use this filter to add P2P users to an Address List, and then base other rules off that.

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

Layer 7 Filters

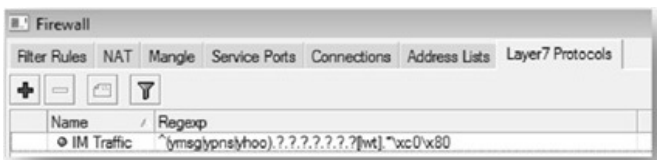
Normally, when you identify traffic, you are using port, protocol, IP addresses, etc. However, some applications use common ports that are used for other types of data. Some instant messenger applications will use TCP port 80 to connect with IM servers. TCP port 80 is more commonly used for HTTP traffic. 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. This header contains your IP addresses, port numbers as well as options like TOS etc. This is less than 2% of the data of many packets. Due to this, we can process rules very quickly. However, when we start doing Layer 7 or application layer filtering we now start looking at the entire packet. Therefore the amount of data we process 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.

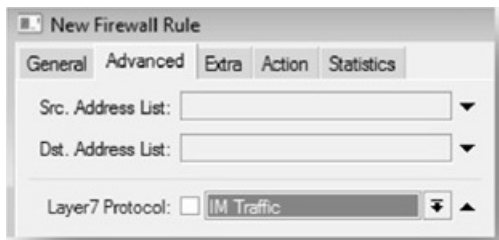
```
^(ymsg|ypns|yhoo).??.??.?  
?.?.?[lwt].*\xc0\x80
```

If we use the IM or instant messaging traffic that we talked about earlier, we can match data based on a layer 7 filter that defines what the packet must contain. If it does contain that, it will match that filter. The note to the right shows 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 IP → Firewall → Layer 7 Protocols Tab.



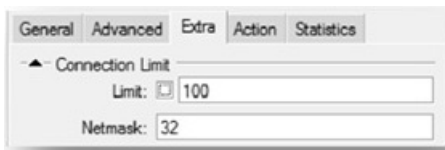
In the above example, we have defined a Layer 7 Protocol and given it a name. This name can be anything but you will use it in your firewall rule. When you create that firewall rule, you will use the advanced tab, and select the Layer 7 Protocol that you created in the Layer 7 Protocol tab. Once you do this, you can then define an action based on that protocol.



With this process, you will consume quite a bit more CPU time, as you are processing the entire packet. You will need to do your own testing, but assuming 2-3 times more CPU for the same amount of data if 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 your 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.



To use this feature, you will need to select the TCP Protocol, 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 my network, so that I apply this to only my network vs. the entire Internet. Once this is done, we can apply the limits. The limit field is for the number of connections. If you wished to limit your customers to 100 connections, you would enter that number in the limit field. The netmask field defines in what size of a 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.

I get lots of questions on what to set this number 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, as they are paying a premium for their connection as well as 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 per workstation, you typically will not have issues.

Port Scan Detection

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.

▲ PSD

Weight Threshold: 21

Delay Threshold: 00:00:03

Low Port Weight: 3

High Port Weight: 1

Action:	<input type="text" value="add src to address list"/>	▼
Address List:	<input type="text" value="portscanner"/>	▼
Timeout:	<input type="text" value="8d 08:00:00"/>	▲

For this to work, you then have a delay threshold and a weight threshold. What this basically does, is says, if you scan ports, and the total weight of the scanned ports within the delay threshold would result in a match to the rule. Remember, this is a rule, so you are matching data based on the rule. Once the data is matched, you have to do something with it. I will typically place an action to add the source IP address to an address list, typically port scanners or some other easily recognizable name. Then I typically will place another rule that will drop all traffic from the port scanner address list.

[Ingress Priority / TOS / DSCP](#)

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

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

[Random](#)

Using random can be fun. I will use the example that when me and my wife are not

getting along. I use the random command to drop 30% of her web traffic and trust me, I get a response in about two minutes from enabling that rule. I don't know what is worse, actually using that rule, or the fact that I just enable and disable when I need it!


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

One more thing that you can do is randomly match data. I have used this to randomly match traffic and add some IP addresses to one address list, and anything that makes it past the random rule, gets added to another address list. Then I can load balance using those lists. Usually we will have these lists timeout after a number of hours so that users can possibly be moved onto another Internet connection next time they move data.

[Limit / DST Limit](#)

This function allows you to effectively limit packet rates based on time. Blocking DOS or 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 count option, which is the maximum average packet rate. This is measured in pps or packets per second. You can though, change the time variable, so you can say, per second, or per minute as needed. option, and this is how many to allow in a burst.



▲ Limit

Rate: 1 /sec

Burst: 5

You also have a burst

The thing to keep in mind with the limit value is that it does not match data until it goes over the rate. Once it goes over the rate, then it matches data. This rule also is global, so unless you specify other options in your rule, it doesn't matter what the IP, or ports are.

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 use the DST limit feature, you can limit to something like 10 pps per destination IP and port. So an individual computer can have 40 pps, but only 10 pps per port.

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.

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 get their time.

For x86 applications, you won't have to worry about this.

Simply specify the start time and the duration. Then specify on what day or days it applies for with the check boxes. I typically use this 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 that time frame, the first rule will match and take the correct action. However, if that rule does not match due to it being out of the time matching time, then it will fail over to the second rule.



Firewall Actions

Inside your firewall there are many different actions. Some just accept data, some deny it, and others can change it. 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. What this does is “accept” or allow data traffic. By default RouterOS is a “allow 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.

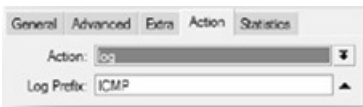
Drop

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 never received it.

I commonly use drop with a connection limiting rule. Once we have xx number of connections, then the connection limit rule starts to match data. If I accepted the data, that means data over the connection limits I had specified would be accepted and continue on, nothing would really change and the rule doesn’t really stop connections. By making that rule a deny rule, once the user goes over that connection limit, it will start dropping or denying connections above the connection limit.

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 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.



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 to a Syslog system etc. 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 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 will stop the ICMP packets, and then supply a reject message back. Once you select the reject action, you can then specify with what reject message to respond with.

Action:

Reject With:

icmp network unreachable

icmp admin prohibited

icmp host prohibited

icmp host unreachable

icmp net prohibited

icmp network unreachable

icmp port unreachable

icmp protocol unreachable

tcp reset

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, it doesn't open a connection, and then drops everything else. To the hacker, the TCP connection is open, and there is no response to close connection packets. This keeps these connections open on the hackers system, and consumes resources etc. The end result, the hacker has lots 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. 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 others
- Input Chain: Log other data that makes it past the other rules.
- Input Chain: Drop that other data

The basics to this list, is 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 is the hard part. RouterOS offers so many options, which building your firewall may seem like a daunting task. Overwhelming options and abilities, and as a router administrator putting it all together is hard work.

First off, we want to prevent the unwanted traffic that we don't need on our network. TCP based connections are a place to start. We will block invalid connections, but allow those established and related. This will keep us from processing a bunch of data and focus our efforts to blocking the initial creation of the connection. This also will keep our CPU time down. If you wish 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, we will wish to prevent some data that is common from 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. First off, they don't want this, even if they don't know they don't. This is a very big hole for hackers, viruses etc, to get into, so I typically will block these ports as they should not traverse my networks. Another set is TCP/UDP 445, this has the same usage of the NetBIOS traffic and I think should be blocked.

```
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 floating around 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.

In our input chains, we also limited ICMP packets and TCP SYN packets when it comes into our Router. 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. This may or may not prevent issues, and may cause issues, but it is common to do.

[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.

Mail servers commonly use port 25 for mail, so identifying the 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 5

connections, and then adds the source IP to an address list will allow you to identify the user. Set this rule up for an hour or so timeout on the list. A second rule would then block all port 25 outbound access based on that address list. With this method, once a computer is infected and sending out massive e-mails, it will be placed on the list, and all SMTP traffic will be blocked for an hour. After that, if they continue to attempt, 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 dealing with 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 processing this is to change the first rule, and add an “accepted” mail server list. You create an address list that is for approved mail servers. This and in your first rule that adds customers to an address list, you exclude the approved mail server list by selecting not on this list. If they are on the list, the rule will never match, so you don’t have to worry about them getting on the list.



Brute Force Attacks

These attacks send the dictionary at a SSH, telnet or FTP Server trying to find a word that will let them in. One of the ways I have found to 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 attempts before it blocks an IP for a set amount of time.

To do this, you will simply create a rule that says if it is a new connection on one of these ports; add them to a stage 1 list. Normally, most actual users will not go past this, and this stage 1 list only keeps the IP there for maybe a minute or two. With telnet, you will get several attempts to type in the correct username and password, but if you type the wrong ones, 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 your 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. Now you have made a third connection attempt, and you are already on the Stage 2 list! Now we add your IP to a stage 3 list. This time though, you are added to that list 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, as you could 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, so on and so forth. The idea behind this is that we have several stages, to let someone that may be valid to login. Once they have gone over this 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 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 are 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, having so many connections open that the server is overwhelmed.

First off we need to identify these types of connections. These are simple as 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 went over this limit, we can then add the remote IP to an address list, that we can then block from.

The second attack that is common is called a POD or Ping of Death attack. This attack has a whole bunch of computers around the world, ping you or an IP behind your router with large packets. The sheer number of ICMP packets coming in typically overwhelms the bandwidth that is available, as well as the number of requests per second uses up CPU power to respond too. Both a DOS and POD attack, having a small Internet connection, under 50-100 Meg typically will result in slow service, or performance as the connection is consumed 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 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 are used to clicking a box to enable a firewall, or selecting a link to turn on VoIP QoS. These types of clickable configurations don't match data correctly, don't guarantee QoS, and in general are a shotgun approach vs. what RouterOS can do. Getting back to that hard part is that you have to make it do what you want! Sometimes it's not as simple as a single check box or enabling a feature, but building firewall rules to do something followed by three more then if statements. One eventually will block that hacker, but you have to make it do it!

Be careful of prebuilt Firewall Scripts. Make sure you know exactly what that script does PRIOR to its installation in your network. Why; in many cases, that firewalling script does things that may adversely affect your network. Things that might be fine to block on the original creators network, but maybe not yours!

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, giving 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 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 shoots TCP port 23 packets over to our SSH Attack chain.

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 to understand on how to match data. The key to your mangle again, is matching data. All of the data matching

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

Prerouting

This is the most common location for your mangle rules. This 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 determining what route to take. 99% of your mangle rules will go here.

Postrouting

This 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 possibility is if you are changing the TOS

bit of the packet.

Input

The input chain in mangle is the same as the input chain in the firewall system. These 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, again, is just like the forward chain in the firewall system. This chain processes on packets that are

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 use marks for. Policy based routing, along with traffic management, and queues are just a few. Each of these RouterOS features will use 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 any way. 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 may 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 them. 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. This will allow you to apply different routing tables to this type of traffic.

Unlike the actual routing rules section, in mangle, you can apply routing rules to packets. This means you can use all of the abilities of RouterOS 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. It also gives you the ability to match based on routing marks as well.

[Connection Marks](#)

Connection Marks are using 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. This 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, vs. indirectly with a connection mark. If you are having CPU issues with the RouterOS device, either you are pushing more data or simply have lots of rules, then you can start using the connection mark to help your router and CPU. I will say though that 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 will allow you to identify traffic on some routers and put a bit number on it for further identification across yours or on other networks. I will typically use this to match data, changing

the TOS bit, then I match and prioritize data across backbone routers, by only using the TOS bits vs. matching data again by some other method. This is one of the few actions that work well on your postrouting chain.

Change MSS

This allows you to change your MSS or Maximum Segment Size field of your IP header. This 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 above the number of bytes in the MTU or Maximum Transmission Unit.

The typical usage for this 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 postrouting 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 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 it 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*.

Performing Network Address Translation

Network Address Translation or NAT, is a very useful feature in many Routers. But unlike many other routers, RouterOS offers a full featured NAT system. To 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 privates' vs. 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 dstnat 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, where the outside will contain public IP addresses.

To access the NAT system in RouterOS you will click on IP --> Firewall --> NAT tab.

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 srcnat or a dstnat chain. Srcnat 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. Dstnat 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 a single IP address that you received from your Internet provider.

New NAT Rule

General

Advanced

Extra

Action

Statistics

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 tid-bits though 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

Inbound NAT or dstnat 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.

The image contains two screenshots of the Mikrotik WinBox NAT rule configuration interface. The top screenshot shows the 'General' tab with the 'Chain' dropdown set to 'dstnat', 'Src. Address' empty, and 'Dst. Address' set to 'public' with an unchecked checkbox. The bottom screenshot shows the 'Action' tab with 'Action' set to 'dst-nat', 'To Addresses' set to 'private address', and 'To Ports' empty.

General | Advanced | Extra | Action | Statistics

Chain: ▾

Src. Address:

Dst. Address: ☐ ▲

General | Advanced | Extra | Action | Statistics

Action: ▾

To Addresses: ▲

To Ports: ▼

To create your basic NAT, think of data that is coming in via a public IP address. This data we need to get to a private IP address. This private could be customer IP, or a server. In this case, you will create a DSTNAT 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](#)

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 srcnat, or source NAT. This feature is very basic. It says, if data comes from *xyz* private IP address, then perform srcnat 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.

The image displays two screenshots of the RouterOS NAT rule configuration interface. The top screenshot shows the 'General' tab with the following settings: Chain: srcnat, Src. Address: Private IP Address, and Dst. Address: (empty). The bottom screenshot shows the 'Action' tab with the following settings: Action: src-nat, To Addresses: Public IP Address, and To Ports: (empty).

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

A method of checking this feature is to have the private IP computer or router, 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 srcnat rule, you should be able to reload that webpage to 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 srcnat 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 functionality 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.

General Advanced Extra Action Statistics

Chain: dstnat

Src. Address:

Dst. Address: ☐ Public IP Address

General Advanced Extra Action Statistics

Action: dst-nat

To Addresses: Private IP Address

To Ports:

General Advanced Extra Action Statistics

Chain: srcnat

Src. Address: ☐ Private IP Address

Dst. Address:

General Advanced Extra Action Statistics

Action: src-nat

To Addresses: Public IP Address

To Ports:

Above I have provided screenshots of the rules that are required. Remember, your

dstnat is inbound, so your DST address field will be your public, and that dst-nats to your private. Your srcnat 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 have to have the srcnat rule to send data from your private

GeneralAdvancedExtraActionStatistics

Chain: srcnat

Src. Address: ☐ Private IP Address

Dst. Address:

GeneralAdvancedExtraActionStatistics

Action: src-nat

To Addresses: Public IP Address

To Ports:

IP out the public. So create your srcnat 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.

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.

The image contains two screenshots of the Mikrotik WinBox interface, specifically the Firewall Rule configuration window. The top screenshot shows the 'General' tab with the following settings: Chain is 'dstnat', Src. Address is empty, Dst. Address is 'Public IP Address', Protocol is '6 (tcp)', Src. Port is empty, and Dst. Port is '80'. The bottom screenshot shows the 'Action' tab with the following settings: Action is 'dst-nat', To Addresses is 'Private IP Address', and To Ports is '81'. Both screenshots show the 'General' and 'Statistics' tabs as inactive.

General Advanced Extra Action Statistics

Chain:

Src. Address:

Dst. Address: ☐

Protocol: ☐

Src. Port:

Dst. Port: ☐

General Advanced Extra Action Statistics

Action:

To Addresses:

To Ports:

You can also change the port during the translation on the inside or private network as well. Once you specify the To Address 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 each 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 dstnat 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 dstnat 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 dstnat 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.

General

Advanced

Extra

Action

Statistics

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

dstnat 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.

General	Advanced	Extra	Action	Statistics
Chain: <input type="text" value="dstnat"/>				
Src. Address: <input type="checkbox"/> <input type="text" value="Private Range"/>				
Dst. Address: <input type="text"/>				
Protocol: <input type="checkbox"/> <input type="text" value="udp"/>				
Src. Port: <input type="text"/>				
Dst. Port: <input type="checkbox"/> <input type="text" value="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!

General

Advanced

Extra

Action

Statistics

Action:

redirect

⌵

To Ports:

53

▲

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 what 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, as well as 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.

Interface <ether1>

General

Ethernet

Status

Traffic

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

OK

Cancel

Apply

Disable

Comment

Torch

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 ether1, 2 etc, and then adding more description to the name, vs. 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 happen 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 vs. 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 Unit¹, 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.

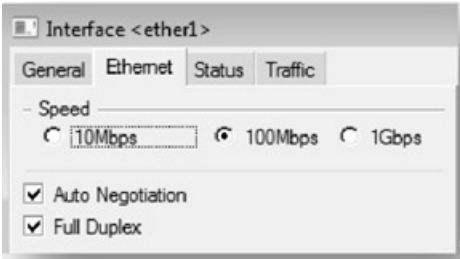
[Switch Controls](#)

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 on-board the hardware. 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 100meg 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 vs. using the CPU and software bridging.

[Ethernet Speed and Negotiation / MDI-X](#)

RouterOS supports full Auto-Negotiation on its Ethernet Interfaces. Most network and computer related devices will support this as well. This will allow the Ethernet ports to negotiate how fast they can communicate. The options are 10Mbps, 100Mbps or 1Gbps. RouterOS also now support 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 and both send and receive data at the same time. This is default for most Ethernet connections. If you have a device that only runs at 10Mbps Half-Duplex, then you would want to manually configure your interface, by checking the 10Mbps option, and uncheck both Auto Negotiation as well as Full Duplex. This will tell RouterOS to ONLY run the selected Ethernet port in this mode only.

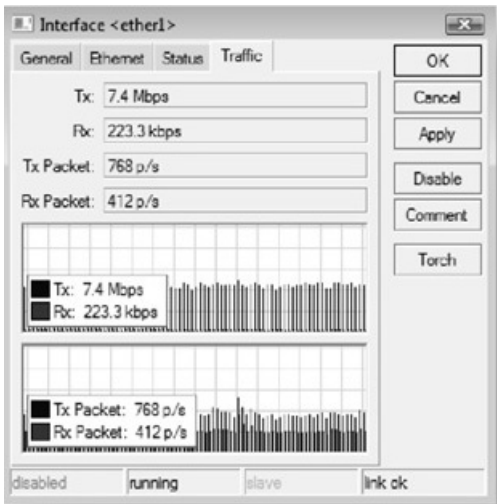


In the Ethernet status window, you will see the current status of your Ethernet port. In this example, we have performed auto-negotiation. We have a link rate of 100Mbps, and are running Full Duplex.

Note at the bottom of the window, we have a few other indicators. If the interface was disabled, the disabled text would be black vs. 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 your Ethernet interface has a link indicator.

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

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!

Interface <vif1>

General Traffic

Name: vif1

Type: Virtual Ethernet

MTU: 1500

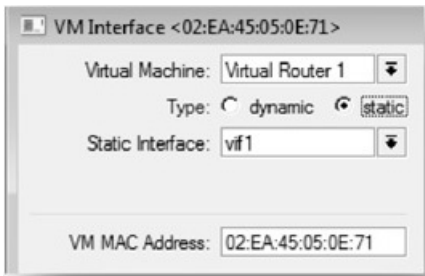
MAC Address: 02:A8:B9:1A:75:C3

ARP: enabled

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 Meta Routers, you will create your Meta Router using the normal procedure. Once

created, then you can assign interfaces. Click on MetaROUTER → Interfaces tab. This will allow you to assign interfaces to your virtual routers.



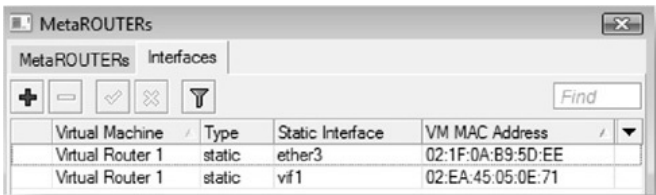
VM Interface <02:EA:45:05:0E:71>

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 ether3, 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 ether3 interface of our physical router, and the second, is our Virtual Ethernet Interface interconnecting both our physical router and our virtual router.

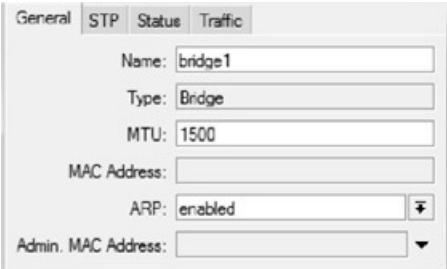
This interface, just like every other one in RouterOS, can be masqueraded, firewalled, and otherwise controlled just like if it was a real interface.

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.




The image shows a screenshot of the MikroTik WinBox configuration window for a Bridge interface. The window has four tabs: General, STP, Status, and Traffic. The 'General' tab is selected. The configuration fields are as follows:

Field	Value
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.

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. This gives your data a single path, that single cable, between the two switches. Now, add another cable. What occurs is packets enter on one switch, go out through the first cable, 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. It just 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.

 **Interface <bridge1>**

General

STP

Status

Traffic

Protocol Mode: ☒ none ☐ stp ☐ rstp

Priority: hex

Max Message Age:

Forward Dealy:

Transmit Hold Count:

Ageing Time:

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 or 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 it would, it disables the port, so you can't use that port in your network. If it's a wireless link, that is turned off, one of the bad things with using STP or RSTP.

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, vs. 30-50 seconds for a change with STP. 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.

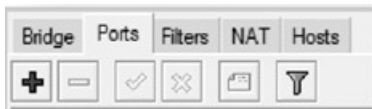
A few things to note, if the port is in forwarding status, which means data is flowing across that port. Disabled status means that it has been disabled due to loop detection. Listening means that it is trying to figure out if it can bring that port to a forwarding status without creating a loop. Backup 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 this. Below is a bride that is running STP.

	Name	Type	Tx	Rx	Tx Pac...	Rx Pac...	MAC Address	Protoco...
R	11bridge1	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.

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.

Bridge Port <vlan100.2>

General Status

Interface:

Bridge:

Priority: hex

Path Cost:

Horizon:

Edge:

Point To Point:

External FDB:

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 Mbits, and right along side of it, you have a low cost 30+ Mbits wireless link. Of course you will wish to use the higher capacity link normally, but with Spanning Tree, it does not detect what 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

OK

Cancel

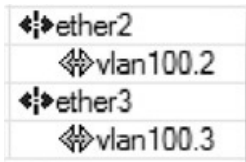
Apply

Virtual LAN (VLANs)

RouterOS Supports Virtual LANs, or VLANs. These are used to separate traffic inside an individual Ethernet segment. This will reduce the number of devices in a broadcast domain, however, 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 that 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 do that!

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.



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
 vlan 100.2	bridge 1
 vlan 100.3	bridge 1

Something to note with this type of setup, is that you can use different VLAN IDs. So if VLAN 100 is on ether1, you could bridge VLAN 200 on ether2, or bridge VLAN 100 on ether1 with VLAN 300 on ether1. You can also add firewall rules based on your bridged interface.

VLAN Configuration

VLAN configuration is super simple. All we need to know what the VLAN ID that you wish to run. If you wish to run VLAN 100 on ether2 then that is the extent of the configuration that you will need.

New Interface

General Traffic

Name:

Type:

MTU:

MAC Address:

ARP:

VLAN ID:

Interface:


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. Then 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 ether1, 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.

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.



The screenshot shows a 'New Interface' dialog box with three tabs: 'General', 'Bonding', and 'Traffic'. The 'Bonding' tab is selected. It contains the following fields:

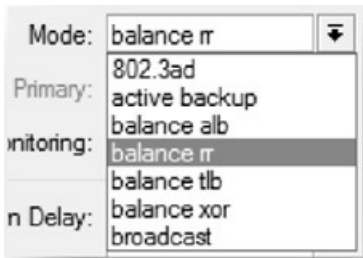
- Slaves:** A list of interfaces. 'ether2' is in the first box, and 'ether3' is in the second box. Each box has a dropdown arrow and a plus/minus icon.
- Mode:** A dropdown menu set to 'balance rr'.
- Primary:** A dropdown menu set to 'none'.
- Link Monitoring:** A dropdown menu set to 'none'.
- Down Delay:** A text box with '0' and a unit dropdown set to 'ms'.
- Up Delay:** A text box with '0' and a unit dropdown set to 'ms'.
- LACP Rate:** A text box with '30 s' and a dropdown arrow.

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 6 GigE interfaces on PowerRouter 732 units. After doing this, and using Jumbo Frames, bandwidth tests showed 5.9 Gigabit 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 does not work the best if the connections have different latencies.



The *802.3ad mode* is the IEEE dynamic link aggregation standard mode. This mode the interfaces will be aggregated in a group where each slave shares the same speed. This mode would be typically what you would use to increase overall speed into a switch, as long as it supports the IEEE 802.3ad standard. This will provide 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 Gig vs. a single Ethernet cable. This mode I have used 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 vs. using active-backup. It does work however there are better ways of providing a failover from a primary to secondary connection. I have used this in replacement of STP though.

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 the 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* 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, 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 ether1 is my primary connection. This mode works quite well even if you don't have balanced links. So your primary connection may be high-end giving hundreds of megabits 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.

Mode:

active backup

▼

Primary:

ether1

▼

Link Monitoring:

none

▼

Down Delay:

arp
mii type 1
mii type 2

Up Delay:

none

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 3 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. This is a hardware and driver requirement that must be met for these modes to function. Most min-Gbics and other such devices must 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.

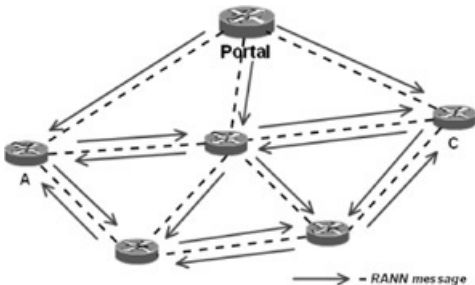
A few notes about bonding Most of these methods require the latency of the connections to be similar, as well as the speed of the connections. If you are trying to balance across different types of connections, I would suggest using another

method in some cases. Trial and Error sometimes will help you with this, to setup and test across your links to see how it 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 vs. 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!

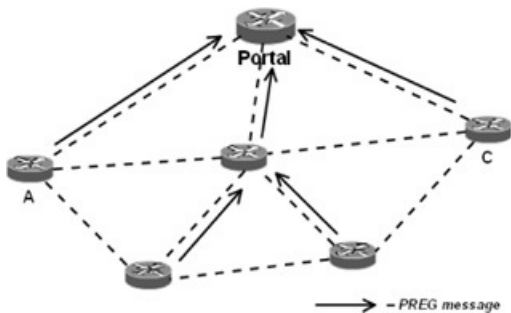
MESH

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 though; 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

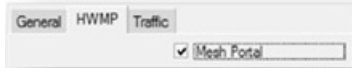
To access the MESH configuration, you will simply click on the MESH button on the left side of WinBox. To use the MESH configuration in RouterOS, the first thing you need to think of is that this is configured as a bridge. Look back up in the bridge system if you need to. You are going to create a mesh 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 that the portal is 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 anything without having a good strong signal to form a WDS bridge with my devices. So, by doing this, I eliminate low signal transmissions, and other devices that simply will not form a good quality link to!



So once you start creating your 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.

Now, 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. This will allow you to set it up 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 label devices as outsiders if they are not part of the mesh network. These may be 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 though. 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 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 it's only for loop prevention. It 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 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 become the master router, assuming that 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 typically when I thought 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.

The image shows a configuration window titled "New Interface". It has four tabs: "General", "VRRP", "Scripts", and "Traffic". The "VRRP" tab is selected. The configuration fields are as follows:

- Interface: A dropdown menu showing "ether1".
- VRID: A text input field containing the number "1".
- Priority: A text input field containing the number "100".
- Interval: A text input field containing the number "1".
- Preemption Mode: A checkbox that is checked.
- Authentication: A section with three radio buttons: "none" (selected), "simple", and "ah".
- Password: A text input field with a dropdown arrow on the right.

So 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, and you will also need to setup a priority if you wish to have one router to be primary and 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.

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 of 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.

So since you created a VRRP interface, you will need a virtual IP. Well this IP is going to 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 ether1. Your backup router would be 172.25.0.2/24. Then you would configure your VRRP IP, the virtual IP. This will be placed on the VRRP interface, and the IP address would be something like 172.25.0.254/32. Your default gateway on your network would be the .254, but the other IPs would 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. Some other considerations 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. Just because the IP is still reachable does not mean that it will just continue to work.

Also, keep in mind that if you have a router that you wish to serve as a backup router, 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 need to put some thought into what happens if one interface goes down on your primary router and not the entire router as well!

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. These tunnels you will route through, thus reducing network overhead and broadcast domains. Also, these provide routing abilities, so you can really control traffic on each segment, provide queuing, and traffic shaping as well as 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 though, 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 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 to 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.

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 vs. 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 as 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 bridging it.

	Name	Type
R	 eoip-tunnel1	EoIP Tunnel

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 shows that the interface is down. It never goes down, and hence, anything that is based on the interface never changes or failover due to this fact. Also, unless you pass data to the other side you will not know if the link is 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 not 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 through. 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

IPIP is IP inside IP. It simply encapsulates IP packets inside other IP packets. IPIP, unlike EoIP, is a standardized tunnel type and 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 ping or ARP to verify that this tunnel is running.



The image shows a 'New Interface' configuration window with two tabs: 'General' and 'Traffic'. The 'Traffic' tab is selected. The configuration fields are as follows:

Field	Value
Name:	ipip1
Type:	IP Tunnel
MTU:	1480
Local Address:	local IP on interface
Remote Address:	remote IP

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. 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. The PPP system is a package that is installed by default; this package supports PPP, PPTP, L2TP, PPPoE and even ISDN PPP. 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 will share these. 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 Secret <user1>

Name:

Password:

Service: ▼

Caller ID: ▼

Profile: ▼

Local Address: ▼

Remote Address: ▼

Routes: ▼

Limit Bytes In: ▼

Limit Bytes Out: ▼

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 that you can setup what username/password they have, what service they can 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 use 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.

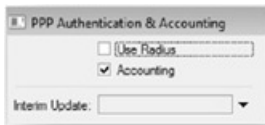
ppp

Interface PPPoE Servers Secrets Profiles Active Connections

PPP Authentication & Accounting

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 will 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 though, you will need to setup a radius server with the PPP service as well for this to work.



[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 also wish to require them to encrypt your data via MPPE128.

The image shows a screenshot of the 'New PPP Profile' dialog box, specifically the 'Limits' tab. The dialog has two tabs: 'General' and 'Limits'. The 'Limits' tab is active. It contains several fields and options:

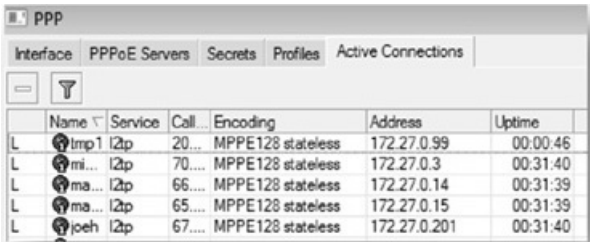
- Name:** A text field containing 'profile1'.
- Local Address:** A text field containing '192.168.11.1' with a dropdown arrow on the right.
- Remote Address:** A text field containing 'hs-pool-4' with a dropdown arrow on the right.
- Bridge:** A text field with a dropdown arrow.
- Incoming Filter:** A text field with a dropdown arrow.
- Outgoing Filter:** A text field with a dropdown arrow.
- Address List:** A text field with a dropdown arrow.
- DNS Server:** A text field with a dropdown arrow.
- WINS Server:** A text field with a dropdown arrow.
- Use Compression:** A section with a radio button selected for 'default', and radio buttons for 'no' and 'yes'.
- Use VJ Compression:** A section with a radio button selected for 'default', and radio buttons for 'no' and 'yes'.
- Use Encryption:** A section with a radio button selected for 'default', and radio buttons for 'no', 'yes', and 'required'.
- Change TCP MSS:** A section with a radio button selected for 'default', and radio buttons for 'no' and 'yes'.

To configure your PPP Profiles, you will click on PPP --> then in the PPP windows select 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.

Another option here is the ability to change your TCP MSS or Maximum Segment Size. 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 would include 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.



The screenshot shows a window titled "PPP" with several tabs: "Interface", "PPPoE Servers", "Secrets", "Profiles", and "Active Connections". The "Active Connections" tab is selected. Below the tabs are two icons: a minus sign and a funnel. Below these is a table with the following columns: "Name", "Service", "Call...", "Encoding", "Address", and "Uptime". The table contains five rows of data, each with a small icon to the left of the "Name" column.

	Name	Service	Call...	Encoding	Address	Uptime
L	lmp1	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, as well you can specify if you are going to use a null modem cable or not. Typically for a modem you would not.

New Interface

General | Dial in | Status | Traffic

Name:

Type:

Max MTU:

Max MRU:

MRRU:

Port:

Modem Init:

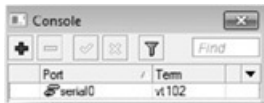
☐ Null Modem

One of the hang up points on 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.

Port List

	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.



Now you will see that the port list does not have a used by 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, however, 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.

Under the dial-in tab, you can specify what profile you wish to use. Think of this, not as a serial console, but a method to get IP connection via a modem. The profile will specify the IP information as well 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 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

The PPP Client is used to dial some connection. 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.



The 'New Interface' dialog box is shown with the 'General' tab selected. It contains the following fields and options:

- Name: ppp-out1
- Type: PPP Client
- Max MTU: 1500
- Max MRU: 1500
- MRRU: (empty field with a dropdown arrow)
- Port: serial0 (dropdown menu)
- Modem Init: (empty field with a dropdown arrow)
- ☐ Null Modem

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 issue it here as well.



The 'New Interface' dialog box is shown with the 'PPP' tab selected. It contains the following fields and options:

- Phone: (empty field with a dropdown arrow)
- Dial Command: ATDT
- User: (empty field)
- Password: (empty field)
- Profile: default (dropdown menu)
- ☐ Dial On Demand
- ☒ Add Default Route
- ☒ Use Peer DNS
- Allow -
- ☒ pap
- ☒ mschap1
- ☒ chap
- ☒ 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 what 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 big 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. 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, however, since we are using the USB port, we will need to ensure that RouterOS knows how to handle this card, i.e. are the drivers included in RouterOS for your card. Refer to MikroTik's website and 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 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 where 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 luck with L2TP connections on lossy or other high latency applications. If I had to make a recommendation, I would use PPTP.



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 you wish to allow and what the default profile to be. By enabling this, you effectively turn this on all IPs on the router.

As you can see the configuration for the L2TP Server is very close to the PPTP Server configuration. Again remember by enabling this server, you turn it on basically on every interface and every IP that comes into the router.

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 use 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.



The image shows a configuration window titled "L2TP Server". It contains several settings:

- An "Enabled" checkbox, which is checked.
- "Max MTU:" set to 1460.
- "Max MRU:" set to 1460.
- "MRRU:" with a dropdown arrow.
- "Default Profile:" set to "default-encryption" with a dropdown arrow.
- A section titled "Authentication" with a horizontal line below it, containing four checked checkboxes:
 - pap
 - chap
 - mschap1
 - mschap2

Windows PPTP VPN Users

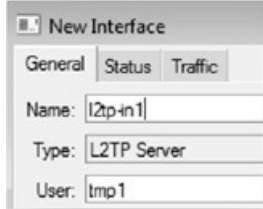
Since Windows 98, we have had a built in PPTP VPN client. This client uses by

default PPTP, but on newer versions 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 interface 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.



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.

Interface <pptp-2K>

General Dial Out Status Traffic

Connect To: 151.58181

User: officenet

Password: *****

Profile: default-encryption

☐ Dial On Demand

☐ Add Default Route

- Allow -

☒ pap ☒ chap

☒ mschap1 ☒ mschap2

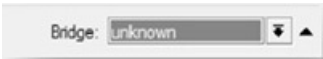
[L2TP/PPTP Client](#)

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 what profile that you want to use on the client. You also can set the Authentication method you wish to send. As well as putting in a default route.

Note that you also have a profile here, most of the time this profile that you specify is very basic, usually the built in profile as most of the information inside the profiles are for the server to hand out to the client. However, when using compression and encryption, both the client/server settings will need to match. Most of the time I use the built in profiles to either run with or without encryption, everything else that is needed is handed out by the server profile.

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 Layer2 Traffic across your encrypted tunnel. This only works in PPTP not L2TP, note that. 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.



At the time of writing this, there are a few bugs in this application, specifically the need to define the bridge group in the profile outside of WinBox. Using Telnet, SSH or the terminal window is fine. To do this, you will use: `/ppp profile set profilenumber bridge=bridgegroup`. Even though we have done it in WinBox, it seems to not take effect until you do it in the new terminal. Of course I have reported it! So this may be fixed in the version that you have!

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+ Meg 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:

Interface:

Max MTU:

Max MRU:

MRRU:

Keepalive Timeout:

Default Profile:

☐ 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 need. You can select what authentication methods to allow as well as what default profile you wish to use. Just like in the PPTP 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.

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. So 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 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!

PPPoE Client

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 you normally will need to access the Internet or network. It will receive your IP address, subnet information, 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.

New Interface

General Dial Out Status Traffic

Service:

AC Name:

User:

Password:

Profile: default

☐ Dial On Demand

☒ Add Default Route

☐ Use Peer DNS

- Allow -

☒ pap ☒ chap

☒ mschap1 ☒ mschap2

Even if you are using this on a wireless or Ethernet interface, remember the PPPoE client is an interface. So if you are doing masquerading, many people forget to change the masquerade rule to have an out interface of the PPPoE Client vs. 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 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. That is usually the goal, 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 vs. having another parameter to configure in the client or server.

Multi-Link or MLPPoE

RouterOS also offers Multi-Link PPPoE. What 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. By enabling this, it will automatically attempt 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, as there typically are some additional overhead, but in all, a decent speed gain. Also this method is a true bonding so if you have 2 x 2meg/6meg Internet connections, then you will actually get on a single TCP Connection around 4meg/12meg



[OpenVPN](#)

OpenVPN is an open source 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 PPPTP or L2TP tunnel vs. 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.

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

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. So if you wish 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.

[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. 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 have all of the 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.

New Interface

General Dial Out Status Traffic

Connect To: 0.0.0.0

Port: 1194

Mode: p

User:

Password:

Profile: default

Certificate: none

Auth.: md5

Cipher: aes 256

☐ Add Default Route

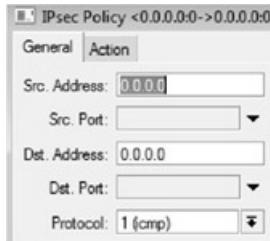
I really like using OpenVPN, the reason is that it gives me the security of IPSEC, and when dealing with financial or private information, this high security is a must. Moreover though, is that it creates an interface. This interface is “SIMPLE” in comparison to route, firewall and do common IP tasks 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, but you will also have to change the mode on both ends to Ethernet, or TAP.

IPSec

Internet Protocol Suite, 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 to IPSec.



The image shows a screenshot of the "IPsec Policy" configuration window in RouterOS. The window title is "IPsec Policy <0.0.0.0->0.0.0.0". It has two tabs: "General" and "Action". The "General" tab is selected. The configuration fields are as follows:

Field	Value
Src. Address:	0.0.0.0
Src. Port:	[Empty] ▼
Dest. Address:	0.0.0.0
Dest. Port:	[Empty] ▼
Protocol:	1 (icmp) ▼



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), but 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 is created under IP --> IPSEC--> Policies Tab. These policies tell your Router what to do with data, 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, you must match it and then perform 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.

All of this data matching does not do you any good unless you have some security; this is where the SA or Security Association comes into play. Each rule will have been associated with SAs that say what and how the packets get encrypted. On top

of all of this security you can even have multiple rules, use their own SAs, or use a common SA. The level field controls this. If you specify the use level, it will send the packet unencrypted, but if you say require, that means you must have a SA for that data to go through, and this will ask the IKE domain (something we will talk about in a few lines), to go ahead and create a SA. The last one 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

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 request from a remote connection. When it does this it has two phases of operations.

New IPsec Peer

Address: 0000

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

Lifeytes:

DPD Interval: 0 (disable DPD) s

DPD Maximum Failures: 1

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 of when they will become inactive. That can be based off a lifetime value, a timed SA, or lifeytes value; it 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 though have two actual values, a soft and hard. Once the soft value has been reached the IKE domain is contacted again, and an attempt to create a new SA before the first one reaches its second timer. The second is the hard value, once it reaches this value, the SA is invalid and hopefully there is a new SA already in place, else, data will have to wait for that, or be dropped.

If you wish to have even more security, you can regenerate that keying material every time the phase 2 operation starts. So even though the SA has been created and the lifetime or lifebytes value is getting ready to expire, now we will create an entire new key, to generate new SAs from that is totally different than the original 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 vs. a pre-shared key; however the key is the most common.

In this section you will also set your exchange mode. I use main 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 work 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/lifebyte check. Basically saying

how it should act if these values are different on one router vs. another. I would suggest ensuring that they are identical on both peers to ensure operations. These are also set here in the peer options.

The rest of the options will allow you to set what kind of encryption and proposals you wish to have. These will need to be identical on both peers of course. You can also set to generate policies. What this does, is 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, but not having to create many complex IPSec policies.

Proposals



The proposal is basically the start of the conversation. This starts a secure channel between the two IPSec peers and allows them to communicate securely even during the start of the conversation. When configuring this, you will need to have the same information on both ends. The Authentication algorithm is the two sides authenticate against each other. The Encryption Algorithm is the method of encryption.

Since we are talking about encryption, now is a good time to discuss the different types of encryption and the performance out of each. Most people will know Triple DES, or 3DES. This is a very common high-security encryption method that is wildly supported. However this Algorithm is fairly slow in most cases. Performance and encryption using this method will take 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 DOD (Department of Defense) standard. This 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 go AES.

The Lifetime and PFS (Perfect Forward Secrecy) Groups are specified here as well, these will need to match the other end of your IPSEC tunnel.

Choosing a Tunnel Type

Choosing your tunnel type can be confusing. Between all of the acronyms and security options, you have a daunting task. 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. The end result is that the type of data that you are going to send over your tunnel is really what matters!

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 or 3	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 things that you want to remember is that IPSEC and OpenVPN will require quite a bit of CPU power. OpenVPN is not difficult to setup, but its more time consuming than PPTP tunnels. If you are in the need to ensure that you have private data, things like complete customer financial data, credit card numbers that are not already encrypted, as well as bank information, encrypted with something higher than the MPPE. However, if you are not transporting that information, use PPTP or L2TP, as these are much simpler to setup, and troubleshoot!

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 has full support for a number of radio cards other than MikroTik brand ones as well. There are many different modes of operations, radio frequencies and other abilities inside RouterOS as well as their own High-Performance protocols. Most of the radio cards that you will find with RouterOS 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 gigahertz or 2.4 gigahertz as some radio cards are designed for only specific frequencies. There are also down conversion radios available. These cards use the 802.11a standard and then down convert the frequencies from the 5 gigahertz band, to other bands. The XR9 and XR7 are cards that do this from Ubiquity Networks. There are other cards 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, and 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 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 it 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 on. 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>

General Wireless WDS Nstreme Status Traffic

Mode: ap bridge

Band: 2.4GHz-B

Frequency: 2442 MHz

SSID: access point

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 will say that if the MAC is not in the access list, by default it will let the MAC address connect. If it is in the access list, it will perform the action listed in that rule. The same goes for the default forwarding 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. If you open your laptop and look for wireless networks, the listing there is generated by the beacon frames that were transmitted. 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 frequencies 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.



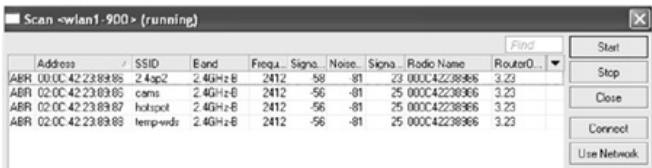
Wireless Tools

RouterOS being a very powerful router also 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 disconnect your wireless interface. If 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 within range of the wireless interface card. You will need to setup your band prior to scanning, and if you are using n-streme, you will need to enable that as well before you will see n-streme enabled SSIDs.



Scan <wlan1-900> (running)

Address	SSID	Band	Frequ.	Signal	Noise	Signal	Radio Name	RouterOS
ABR 00:0C:42:23:89:85	2.4sp2	2.4GHz-B	2412	-58	-81	23	000C42238986	3.23
ABR 02:0C:42:23:89:85	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:42:23:89:89	temp-wds	2.4GHz-B	2412	-56	-81	25	000C42238986	3.23

Find

Start

Stop

Close

Connect

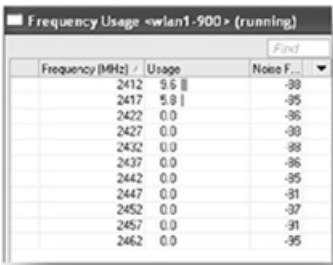
Use Network

This will also give you the MAC addresses of your access points, and if there is MikroTik proprietary extensions transmitted, such as radio names and RouterOS information. You will also see the signal strengths, SNR and noise floor information.

By clicking on one of these, you can then select connect to automatically change your wireless interface to station mode, as well as set the proper SSID and frequency.

Frequency Usage

The frequency usage tool will take all packets and data received by your wireless interface give you a noise floor based on the channels and show you the % 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 usage. Sometimes that's not an option, but maybe.



Frequency (MHz)	Usage	Noise F...
2412	9.6	-39
2417	5.9	-35
2422	0.0	-36
2427	0.0	-39
2432	0.0	-38
2437	0.0	-36
2442	0.0	-35
2447	0.0	-31
2452	0.0	-37
2457	0.0	-31
2462	0.0	-35

Sniffing

The Sniffer is another wireless tool. This is basically the same as the packet sniffer tool, but instead of having to be connected to a wireless interface, this pulls packets out of the air. You can use multiple channels, as it does not look at SSIDs. You can also use this in conjunction with a streaming server; this is covered in the packet sniffing section under RouterOS tools more.

Snooping

Using the snoop tool you will be able to see all of the wireless stations, access points and statistical 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 and channel they are using and even the actual bandwidth being used by each device!

Snooper <wlan1-900> (running)

Networks Stations

Find

	Frequenc...	Band	Address	SSID	Of Freq (%)	Of Total (%)	Bandwidth	Net...	Stat...	
00	2412	2.4GHz...	00:0C:42:23:E9:86	2.4ap2	4.3	24.5	36.0 kbps	4	4	
00	2412	2.4GHz...	02:0C:42:23:E9:86	came	1.0	24.6	6.9 kbps		1	
00	2412	2.4GHz...	02:0C:42:23:E9:07	hotspot	1.1	25.1	5.0 kbps		1	
00	2412	2.4GHz...	02:0C:42:23:E9:88	temp-wids	1.1	25.3	5.1 kbps		1	
00	2417	2.4GHz...			2.9		24.0 kbps	0	0	
00	2422	2.4GHz...			0.0		0 bps	0	0	
00	2427	2.4GHz...			0.0		0 bps	0	0	
00	2432	2.4GHz...			0.0		0 bps	0	0	
00	2437	2.4GHz...			0.0		0 bps	0	0	
00	2442	2.4GHz...			0.0		0 bps	0	0	
00	2447	2.4GHz...			0.0		0 bps	0	0	
00	2452	2.4GHz...			0.0		0 bps	0	0	

[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 is an access point, or what the max speed of a wireless point-to-point link is. Then when I tell them, they will say “But I am connected at 54meg”. So let’s clarify this information!

We will start with 802.11b. The max air-rate that you can get is 11meg. But the actual data transfer rate is right at 6 to 7 Meg depending on the type of traffic. UDP traffic will be on the higher side. However, that assumes one wireless client! As you add more and more clients, you will have to keep lowering total possible data rates.

What that means is, with an 802.11b access point, the absolute most bandwidth you can get for all clients connected to that access point would be around 6 megabits. Then you add more clients, each client uses up a bit more access point time, so that actual throughput drops a bit more with each connection you add.

Now let's talk about 802.11a/g. If you have a data connection at 54megAir-Rate, the max data rate will be around 30-40 Meg in one direction. As you drop your frequency mode, from standard 20 MHz channels down to 10 or 5 MHz channels you also will get a cut in throughput by $\frac{1}{2}$ as well.

Access Point Time

Another thing I get is questions about access point time. Questions mostly about how many people can you put on a single access point etc. This is really a measure of modulation and connect rates. The example is that if you have a client connected at 11meg connect rate, they have to use 11 times more of the access point time, to transfer 500k of data vs. a client connected at 1meg

So recommendations, normal usage, 30-40 clients on a B/G access point is high, but if most are pushing 11meg connections on most of your clients, then you can get upwards of 50-60! On 802.11a, I would say this is a bit more, assuming again, good data rates, upwards of 60-70 clients. This also depends on how much bandwidth you are giving each connection as well. If you are a Wireless ISP, then selling 4 Meg down Internet access on a B access point you can only sell a few connections on one access point!

Bands

There are a number of bands that RouterOS will 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 air rates. You can also run G-

Turbo mode as well, this uses a 40 MHz channel size vs. 10 MHz. Doing this in 2.4 GHz typically will reduce the number of non-overlapping channels to about two, but sometimes will give you higher than expected data rates. In a wireless ISP scenario, I would stick to the smaller channel sizes.



You also have options for 2GHz 10 and 5 MHz channel sizes. The reason for these options is to have more channels available, and to reduce interference as well. For every 1/2 cut in channel size, you will receive around 3dbi more SNR, just due to the fact that you are now only listening in 1/2 of the frequency range. This does not show up in signal strength gain, just in SNR.

In 5 GHz with 802.11a, you have the same options, both A-Turbo using 40Mhz channel sizes as well as smaller 10MHz 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 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 N-Streme Dual Slave mode in the N-Streme Dual section vs. in the operation modes section.

[AP-Bridge \(P2MP Access Point\) Mode](#)

You may be familiar with two of the most common radio modes; one is the AP-Bridge mode in RouterOS. This is your 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 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 mode 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 (Client Premise Equipment) that WISPs would install would 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 requiring bridging the proper way, per 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 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 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 standard 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 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.

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 not limited to the size 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 to and from buildings. Chances are 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 on them. What if you were a mile behind one of the antennas, well you would get at least the transmission from the far end of the link (depending on power levels). So now you are a mile away and receiving 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. Connections are one thing, this prevents them from being 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. 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 xxx MAC address to connect to the access point. Keep in mind that this is not encryption, so data is in the air that is 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 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 to static keys. If you make it optional, which means that they don't have to have WEP to connect, but if 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.





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.

With that said, my recommendation is to NOT use WEP. WEP is outdated, originally created in 1997. With any Linux based laptop for the most part it takes about 20 seconds to break. It's considered very easy to break and should not be used if you are wishing to have a quality 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.

New Security Profile

General **RADIUS** EAP Static Keys

Name:

Mode: ▼

– 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:

Typically you will deploy WPA the same as you would as WEP, this time though go to your security profiles and when creating a new security profile, select the mode as dynamic keys. You can then select 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 do relate. What 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.

The screenshot shows the 'New AP Access Rule' configuration window. It contains the following fields and options:

- MAC Address:** 00:00:00:00:00:00
- Interface:** all
- Signal Strength Range:** -70..120
- AP Tx Limit:** (empty)
- Client Tx Limit:** (empty)
- ☒ Authentication
- ☒ Forwarding
- Private Key:** none
- Private Pre Shared Key:** (empty)
- 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, allowing you to allow any MAC to connect during

lunch hour, but otherwise, only allow a few MACs to connect. Remember that this is not MAC authentication only. If you have WPA2 running, then you will still need that WPA2 shared key. But you can also set 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 is typically what is needed to have an 11megAir Rate Connection. With a rule like the one above, every MAC must have between a -70 and a 120db signal to 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 customers 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...	Signe
 000C423A...	00:0C:42:3A:D2:AC	wlan1	00:56:55	no	no	0.500	

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



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

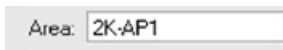
Connection lists are the exact opposite of the access lists. Access lists, if you read back a few pages, are for controlling access to an access point. Connection lists are for telling your station what and how to connect to access points! To use them, remember that they are an ordered rule list, just 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.

An example is that you can say connect to SSID 'Tower1' only if the signal strength is above -70. If it drops below that signal, 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 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 etc.

Area / Area Prefixes

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. It of course will have to match the rest of the values in your connect rule as well.



Area: 2K-AP1

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 etc, all while running a single radio card.



The screenshot shows the 'New Interface' configuration window in RouterOS. The 'Wireless' tab is selected, and the following settings are visible:

- SSID:** A text input field with a dropdown arrow.
- Master Interface:** A text input field containing 'wlan1' with a dropdown arrow.
- Security Profile:** A text input field containing 'default' with a dropdown arrow.
- Default AP Tx Rate:** A text input field with a dropdown arrow and 'bps' unit.
- Default Client Tx Rate:** A text input field with a dropdown arrow and 'bps' unit.
- Checkboxes:**
 - ☒ Default Authenticate
 - ☒ Default Forward
 - ☐ Hide SSID

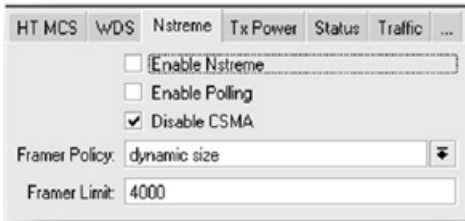
To create a virtual-AP, simply click on Interfaces → Add → Virtual AP. The main three settings you will need are the new SSID, what actual radio card you will be transmitting this new virtual SSID from, and the security profile. All of the other settings are the same as any other wireless interface. 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, remember, even though you have two separate SSIDs, they are still on the same channel as the master wireless interface. They share the bandwidth of the frequency they you have them connected on.

N-Streme

N-Streme is a proprietary extension of the 802.11x design that MikroTik created to overcome some of the limitations and increase 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. N-streme does a few things, including compression, polling, and no limits on distance. It will also combine frames similar to the way M3P does as well.



The image shows a configuration window for the N-Streme feature. At the top, there is a tabbed interface with the following tabs: HT MCS, WDS, Nstreme (selected), Tx Power, Status, Traffic, and an ellipsis (...). Below the tabs, the configuration options are as follows:

- ☐ Enable Nstreme
- ☐ Enable Polling
- ☒ Disable CSMA
- Framer Policy: [down arrow]
- Framer Limit:

To enable N-Streme on your access point, you will simply need to check the Enable N-Streme 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 N-streme enabled, you are looking only for n-streme enabled access points, not standard a/b/g/n access points.

In typical usages, nstreme mode will provide higher data throughput, however, typically increase latency 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 vs. running standard 802.11x. We have seen 52 Meg connections using 5gig-Turbo modes and N-Streme, however, with 802.11n; this performance is upwards of 70meg 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 may be.

N-Streme Dual

Dual N-Streme uses two wireless interface cards to create a full duplex wireless link. You will need to setup your two wireless interfaces to 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 n-streme dual interface that you will create. To create this, click on interfaces → Plus sign → Nstreme Dual Interface.

General	Nstreme Dual	Data Rates	Status	Traffic
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				
<input type="checkbox"/> 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 allowing one card to receive only and another to send. On the Nstreme Dual tab of your new interface, you will need to setup what radio card is receiving and what one is sending. These would be the TX and RX radio settings.

General	Nstreme Dual	Data Rates	Status	Traffic
Rx Signal Strength: <input type="text"/>				
Tx Signal Strength: <input type="text"/>				
Rx Rate: <input type="text"/>				
Tx Rate: <input type="text"/>				
Packets (Tx/Rx): <input type="text" value="0/0"/>				
Bytes (Tx/Rx): <input type="text" value="0/0"/>				
Frames (Tx/Rx): <input type="text" value="0/0"/>				
Frame Bytes (Tx/Rx): <input type="text" value="0/0"/>				
Hw. Frames (Tx/Rx): <input type="text" value="0/0"/>				
Hw. Frame Bytes (Tx/Rx): <input type="text" value="0/0"/>				
Tx Retries Timeout: <input type="text" value="0"/>				
Tx Retries Lost: <input type="text" value="0"/>				
Rx Bad Seqs: <input type="text" value="0"/>				
Rx Duplicates: <input type="text" value="0"/>				
<input type="checkbox"/> Connected				

One of the most common mistakes that are made in the Nstreme is that once you configure one end, you will have frequencies for TX and RX; these frequencies are flipped on your remote system. So the example is if you transmit at 5180 then you have to receive at 5180 on the remote side. This is a common mistake. The second is where to get your Remote MACs from. The remote MAC is the MAC address of the far ends 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 ends 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 db 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.

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 from each other just as if there was a wire 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 as upon the creation of your WDS interface, the system will either dynamically add the new WDS interface to the bridge group, or you will add it once the WDS interface is created.

HT MCS	WDS	Nstreme	Tx Power	Status	Traffic	...
WDS Mode:		dynamic				⌵
WDS Default Bridge:		none				⌵
WDS Default Cost:		100				
WDS Cost Range:		50-150				
<input type="checkbox"/> WDS Ignore SSID						

On the remote end, you can use station-wds, or even bridge if you wish. 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. Upon the link coming 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 doing this method is the slow performance and bridging loops. When you first bring up a new unit with dynamic WDS enabled, if it can see 4 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 first comes in with bridging loops. The interfaces are not smart enough to make a determination if WDS link 1 is better than 2, and number two 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. This method is just simply not practical in many cases.

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. If we assume that we have a chain of WDS enabled access points. Performance of a client connected to the main or first access point would be at 11 Meg assuming they have that good of a connection. The second access point though, even though it shows an 11Meg connection, has $\frac{1}{2}$ of the performance. This is due to the access point taking up $\frac{1}{2}$ of the air time to retransmit what the client transmitted back to the first access point. So now the possible performance is only 5.5 Meg given a perfect world (no interference, perfect signal qualities, etc). Now we go to the third access point, and the actual performance here is cut by $\frac{1}{2}$ again. So now the possible performance is 2.75 Meg This is air rate too, so if we are using 802.11b, the actual data rate would be under 1.5 Meg This may be enough for your solution however, that's given a perfect world with one client. As you start adding 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 interfaces registered, and then in the same bridge group you will have another radio that can be a different SSID and channel that will be for clients as well as the next WDS link. That link will use again, a radio card in station-wds connected to the access point from node 2. This will eliminate the need for the same radio card to do the retransmission to the next node allowing a dedicated radio do this. You can setup routing as well using WDS dual radio setups.

In some cases this will also eliminate the bridging issues as there are much fewer systems on the same SSID and channel for WDS to dynamically pair up with. However, it can still occur and I would recommend using Spanning Tree as well.

WDS and 802.11n

At the time of writing this book, MikroTik has released its 802.11N wireless radio cards as well as support for these cards and protocols. The reason I bring it up in the WDS section is that there is no provision for WDS in the 802.11N standard. Even though RouterOS does support bridging via WDS on these radio cards, the performance that you would otherwise get with 802.11N is negated by the WDS system. Check with MikroTik to see if this is still the case or if they have made a work around on this issue.

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 a wireless link. For clarification, these optimizations are really designed for Point-to-Point wireless links that are fixed on both ends, an example will be a tower to tower wireless link.

Keep it Simple First

First, when setting up a wireless link, start with the most basic configuration possible. Setup one end in bridge mode and the other in station mode, do not put a security profile in place, and so leave it as an open access point. Don't worry, we will secure it later. The idea here is that you will keep things as simple as possible until you think you have the best link possible.

Hardware Selection

Selection of your hardware is necessary now. It is ALWAYS better to have a larger antenna and lower power radio. Antennas amplify in both directions; they have receive and transmit gain. If you are using a 20dbi antenna, that means what signal it receives will be increased by 20dbi and the power going from the radio card at 12dbi then is increased by 20dbi as it leaves the antenna. IF you can use a 60mw radio card and a 20dbi antenna, it will be better than using a 600mw radio card and a 12dbi antenna. Again, lower power output larger antenna, of course this doesn't work out all of the time, but it will help. Last thing is that when dealing with backhaul links, don't skimp on the CPU power. Go with AH RouterBoards vs. the cheapest thing possible!

Antenna coax and selection

On your radio to antenna connection, you want the least amount of loss possible. I recommend using integrated radios when possible. This cuts your loss down to the least amount possible. You need to run outdoor, UV rated cat5 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 setup, due to distance or wind loading on your tower, then you will need to either use the shortest amount of coax as possible, or high quality coax. An example is that I typically would not go more than 20-30 feet with LMR400 cable before I started looking at putting in 5/8 inch Heliac coax. We are dealing with milliwatts of power, not watts, so even a few dbi of loss is substantial.

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 about what dbi you should receive on each end based on this planning. So if you have a 10 mile link with 19dbi antennas and 320mw radios, you should have close to a -70dbi signal on both ends. When you align the links first start with your horizontal or side to side azimuth, and then once you get the signal as much as possible that way, drop in from the highest point where you lose signal and go till you get the best possible signal vertically. The idea is that you want your antenna with as much up tilt as your signal will allow. This will prevent other ground based interference.

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 5gig range (if that's what you are using). You should record this for as long as possible, but I would think the minimum time would be 30-45 minutes. After this recording is done, look at it and find the cleanest channels you can use. Hopefully you will have a number of channels to choose from. In the US, if you want to run in the 5.4GHz band and have the proper license from the FCC, you will need to add super channel license to your RouterOS system. Now that you have the channels, select a channel that is the cleanest. 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 +/- 2dbi of the planned link quality? If it is not, then there may be something wrong. Is it 20dbi off? If it is, then you may have an antenna in the wrong polarization! Typically the difference between vertical and horizontal polarization is around 20dbi, and if you are using a single antenna and you are 20dbi off from what you calculated, then guess what, turn one of your antennas 90 degrees! What if your signal is fluctuating wildly, more than 5dbi +/-? This may be that you have Fresnel zone issues, check your link again and your path and make sure there are no obstructions. Moving one end up or down 5-10 foot may provide you with relief from this issue.

Secure your Link and Testing

Now 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. The reason for this is that moving data across the wireless link takes CPU power, you don't want to add more CPU time by generating data to do the bandwidth test with, and you need to have separate board

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 54meg to 48meg to 36 Meg and so on all of the time while you are doing your transfer, you may have rate flapping issue. What you want to do is lock in your data rate to the highest rate that you stay at constantly. If you move data and it stays 90% of the time at 48meg, then unselect the 54meg air rate. Changing from 48 to 54 Meg 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 Meg data rate in there, but remove the lower data rates. This will prevent you from constantly changing data rates 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.

Using Nstreme

Now that you don't have a bunch of jitter in your link, now try enabling the default nstreme settings. Remember to do it on your far end first, and then the end you are at, so that you don't lose connectivity. Upon the connection being 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 speed, is to try using turbo channel sizes, and if you can't use Nstreme, try turning on M3P as well on both interfaces, this could save you quite a bit of bandwidth as well. However, there is already some compression done with Nstreme

and I have found M3P doesn't really help too much with Nstreme turned on.

Troubleshooting Wireless Links

Low Signal

When you create your wireless link the first time and based on your path analysis, then you should have link budget created. 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 db! IF your signal is not within a few db, in excess of 4 db off, I would recommend looking at your link further. The number one reason is antenna alignment. Follow the new link section on aligning your antenna. The second reason is Fresnel zone encroachment. Again, your path analysis should show if you have something in the Fresnel zone that can affect your signal. Finally on a new link, simply a bad radio card or antenna connector could cause you issues. I highly recommend using a pre-built RouterBoard solution that is tested with signal strengths to ensure that you don't have this issue in the field.

On an old link, if your signal has gone down then you would need to look at antenna alignment. This is usually due to lose bolts over time. If the signal is wandering 2db +/-, then look at the following section.

Wandering/Fluctuating Signal

A wandering signal or fluctuating signal would be +/- 4dbi of signal within a few minutes. So if you just installed your link and the signal is changing wildly, again, 2dbi +/-, then I would go ahead and look for issues. If this is a new link, then I would first look at Fresnel zone issues. Move one side of the link up 5-10 foot and see if that changes things. If this is an old link that just started to have this issue, I would ask 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 needs to be wrapped extremely well to prevent water from getting into them.

Bad CCQ

Bad CCQ can be a result of 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 with 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 causing many retransmits on your link. Put in a spectrum analyzer and test for this if you have done everything else. Remember you want to use the larger antenna with a tighter antenna pattern to minimize interference from other links.

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-2-peer traffic as well as by using stateful packet inspection or Layer 7 identification rules. With this you can build an extremely sophisticated queuing system that can provide quality of service, QoS, to your customer's data, based on any method you wish!

Providing QoS is difficult for most systems, and they also only allow you to identify specific types of traffic. Lots of switches look for simple ToS bit, however, that may not be the only traffic that you wish to prioritize. 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 to being able to 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. Typically these can be as simple as an IP 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 identification. For most people, simply allowing an IP address to access up to

a specific amount of bandwidth is most of the battle.

In this section, we are going to talk about how MikroTik does its queuing, the methods of queuing available, as well as how to ensure QoS with applications, controlling Peer-2-Peer traffic and help you understand how bursting works as well.

Identifying Queue Data

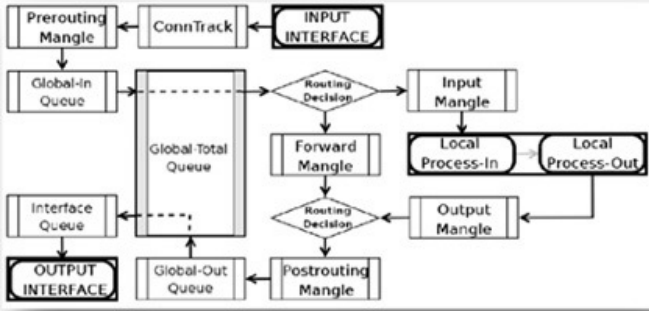
Normally we would go into a long section on how to identify data; however, we already covered this in our firewalling and mangle section. However, it's important to note that if you wish to identify data by using ports and protocols, you will need to create packet marks so that our queuing system has 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 has 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.

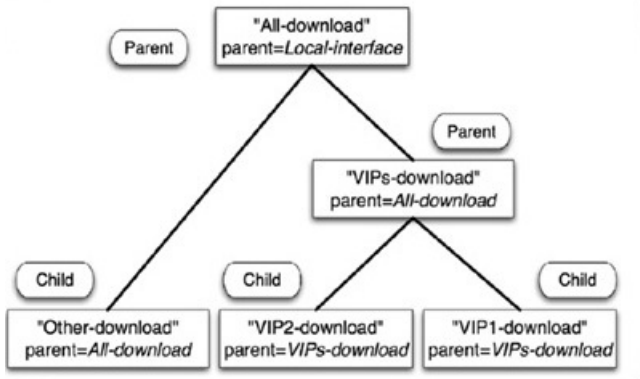
HTB Packet Flow

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 Meg for each of your 10 customers, if you only have 5 Meg of Internet connection, then you really can't guarantee that bandwidth. But if you have 20 Meg, 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, \geq , 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

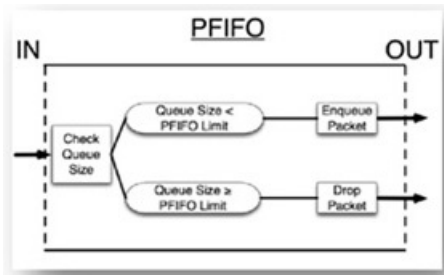
There are a number of different types of queues. RouterOS supports four different types of queues, FIFO, RED, SFQ and PCQ. To help you decide on what you will use, the chart below will assist you!

Queue Type	Reason to Use	Pros / Cons
FIFO	Use for simple bandwidth limiting and control. Simplest and fastest.	<i>Pros:</i> Very quick, low CPU overhead. <i>Cons:</i> Provides only two priorities.
RED	Have never found any.	<i>Pros:</i> Still very quick. <i>Cons:</i> Never had a Need for the randomfeature.
SFQ	Gives you up to 16 queue levels, a must if you are wishing to provide QoS.	<i>Pros:</i> Provides up to 16 priority levels, and works great for providing QoS configuration. <i>Cons:</i> Highest in CPU cost.
PCQ	Use if you wish to share bandwidth equally among many users.	<i>Pros:</i> VERY FAST, one queue can serve hundreds of clients. <i>Cons:</i> 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. As data comes in it goes out. It 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 just on different types of data, one works on entire packets vs. the other working on bytes of data.



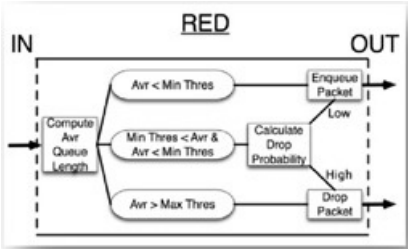
The way this works, is simple, as data comes in, it flows through a queue, think of the queue as a bucket of water with a hose going in and a valve for a drain. As data flows into this bucket, the bucket is constantly draining at the rate that you specified in the queue. So if you have a drain that can fit 1 Meg of data through then that would be its max-limit. As data comes into the bucket, it drains back out, but sometimes, data comes into the bucket faster than it can drain out. This is normal, so what happens? Well the bucket eventually becomes full. This is 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 that bucket at the max-limit rate, and then eventually that 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 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. Remember though as they drain, there is no priority between each individual queue and bucket.

RED Queues

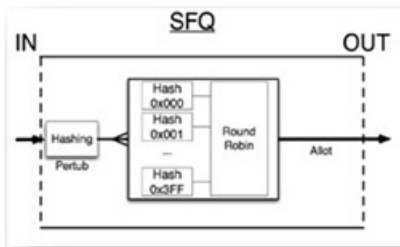
When I first started typing this section, I questioned if I should even put this queue type in the book. I have never really found a good use for RED Queues (Random Early Detection Queues). These queues function just like a FIFO queue, with only one exception. It gives an additional possibility that packets coming in may be taken out of the bucket randomly. Now the idea behind this to prevent what is called Global Sync. We won't get into that here, but basically it says that as 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 is suppose to fix this issue.



I typically do not use RED in production, there just simply does not seem to be a need for this in most cases. However, your situation may warrant such a queuing system, and it is built in with RouterOS.

SFQ Queues

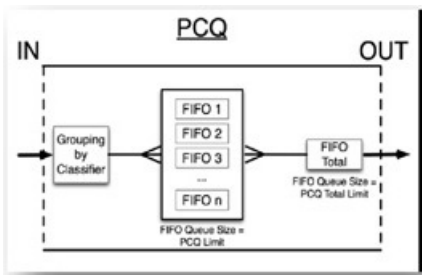
SFQ or Stochastic Fairness Queuing is the way to go if you are looking for great QoS implementation. This system will take advantage of priorities, max-limits and limit-at's in your queues. It works by using a hash value from up to 4 different classifiers, typically but not limited to using both source 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 this 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 is a MikroTik Specific queue type. This was designed to simply distribute traffic evenly across a large subnet and then provide the ability to

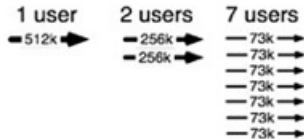
limit each sub-stream that is created while maintaining a super low CPU requirement. PCQ works by taking classifiers, and then based on those, forming sub-streams. Each of those are basically an individual FIFO queue. In most WISPs and ISP implementations, the idea is to have an entire subnet have the 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.

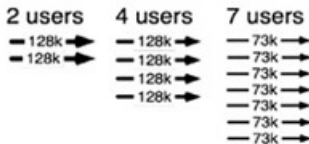
pcq-rate=0

———— max-limit=512k —————→



pcq-rate=128000

———— max-limit=512k —————→



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.

While using PCQ these queues get created, you also need to look at the limit and total-limit settings of your PCQ Queues. The reason for this is, 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 total-limit divided into the limit equals that 40 user number. You should have at least 10-20 packets available for each user, so you will need to increase this number as your user count grows. If you give a limit of 50, figure 20-25 packets per user, so if you have 300 users, you would need a total-limit of around 7500.

As this grows you will need to take into account RAM usage as well. PCQ uses about 4.2 Meg's of RAM if you have a total-limit at 2000, and around 10.5 Meg 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 Meg. Take your total-limit and divide that by around 470 or so. That will get you a good number for RAM.

Using PCQ

Now that you understand how PCQ works, I want to go through on 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.

Type Name:

Kind: 

Rate:

Limit:

Total Limit:

— Classifier —

<input checked="" type="checkbox"/> Src. Address	<input type="checkbox"/> Dst. Address
<input type="checkbox"/> Src. Port	<input type="checkbox"/> Dst. Port

First go to 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 type will use a destination address as its classifier.

Type Name:

Kind: 

Rate:

Limit:

Total Limit:

— Classifier —

<input type="checkbox"/> Src. Address	<input checked="" type="checkbox"/> Dst. Address
<input type="checkbox"/> Src. Port	<input type="checkbox"/> 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 wished 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.

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 Meg or whatever your Ethernet connection is, and not divide up the bandwidth accordingly. So you have to setup a rule that knows how much bandwidth you wish to divide evenly between all of your customers!

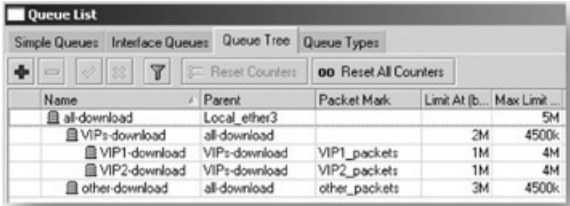
So now we create a simple queue rule, under advanced we select our upload and download queue types to our new PCQ Queue types that we created. WE specify our target address so that we know what data we are aiming for, in this case our private subnets, and then specify a Max-Limit so that the PCQ system knows where to start dividing the bandwidth up at.

The screenshot shows the Mikrotik WinBox configuration window for a Queue Rule. The 'Advanced' tab is selected. The 'Name' field contains 'q1'. The 'Target Address' field contains '10.0.0.0/8'. Under the 'Target Upload' and 'Target Download' sections, the 'Max Limit' is set to '3M' for both. The 'Limit At' is set to 'unlimited' for both. The 'Queue Type' is set to 'upstream' for upload and 'downstream' for download. The units are 'bits/s'.

General	Advanced	Statistics	Traffic	Total	Total Statistics
Name: <input type="text" value="q1"/>					
Target Address: <input type="text" value="10.0.0.0/8"/>					
<input checked="" type="checkbox"/> Target Upload <input checked="" type="checkbox"/> Target Download					
Max Limit: <input type="text" value="3M"/> <input type="button" value="v"/> <input type="text" value="3M"/> <input type="button" value="v"/> bits/s					
Target Upload Target Download					
Limit At: <input type="text" value="unlimited"/> <input type="button" value="v"/> <input type="text" value="unlimited"/> <input type="button" value="v"/> bits/s					
Queue Type: <input type="text" value="upstream"/> <input type="button" value="v"/> <input type="text" value="downstream"/> <input type="button" value="v"/>					

[Queue Trees](#)

The Queue tree is an implementation of 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 are 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, as if you mark web traffic for instance, data that is going out your WAN interface is your up traffic and data going out your LAN interface is your down traffic. You can specify speeds here as well and your priorities.



Queue List					
Simple Queues		Interface Queues		Queue Tree	
Queue Types					
Name	Parent	Packet Mark	Limit At (b...	Max Limit ...	
all-download	Local_ethernet3			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 that is 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. You can type the speed in bytes, so 512k would be 512000, or specify in k by typing 512k, and megabits with the M letter.

General	Advanced	Statistics	Traffic	Total	Total Statistics
Name: <input type="text" value="queue1"/>					
Target Address: <input type="text" value="Customer IP"/>					
<input checked="" type="checkbox"/> Target Upload <input checked="" type="checkbox"/> Target Download					
Max Limit: <input type="text" value="512k"/> <input type="button" value="v"/> <input type="text" value="2M"/> <input type="button" value="v"/> bits/s					
▼ Burst <input type="text"/>					
▼ Time <input type="text"/>					

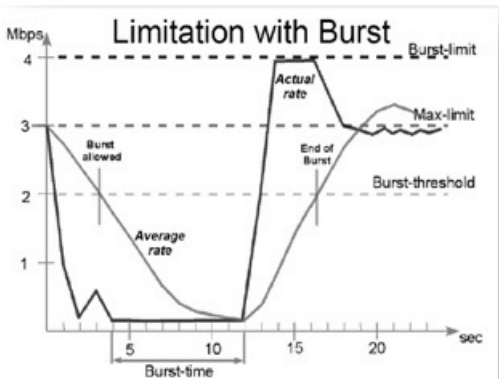
Bursting

Once you have setup your customer with their simple queue, you can also do bursting. Bursting allows you to specify several options; the goal 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 the customer basically sits there moving no data while they 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 small download that is a few hundred K, you can download that maybe at 2 Meg but once you get over a few megabytes of download, it slows down.

The screenshot shows a configuration window with tabs: General, Advanced, Statistics, Traffic, Total, and Total Statistics. The 'Advanced' tab is selected. The 'Name' field contains 'queue1'. The 'Target Address' field contains 'Customer IP'. Below these are two checked checkboxes: 'Target Upload' and 'Target Download'. For both, the 'Max Limit' is set to '1M' bits/s. A section titled 'Burst' is expanded, showing 'Burst Limit' set to '2M' bits/s and 'Burst Threshold' set to '900k' bits/s. The 'Burst Time' is set to '60' seconds. A 'Time' section is collapsed at the bottom.

Field	Value	Unit
Name	queue1	
Target Address	Customer IP	
Target Upload	1M	bits/s
Target Download	1M	bits/s
Burst Limit	2M	bits/s
Burst Threshold	900k	bits/s
Burst Time	60	s

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 Meg 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 60 seconds, 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 Meg of bandwidth. As their download progresses, the average data rate for the customer over time, goes up. Since we are dealing with 1 Meg and 2 Meg, 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 Meg.

Creating Queue Priorities with Parents

This is where some people get lost on understanding how the parents and child queues work. You create your parents 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. 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. This master queue has plenty of bandwidth, so we are not limiting bandwidth except for 100 Megabit, however, in a single clock cycle, packets that are P1 will go out before packets with a status of P8. These are arbitrary 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 it will share bandwidth with the master queue.

Parent: ▾

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?

So first to ensure bandwidth you have to be able to identify your traffic. So, 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	0	

A basic VoIP QoS system is above. This assumes 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 1/3 of the bandwidth that we have.

The number one thing everyone forgets is to setup some form of total limit. Something that says our Internet connection is 3 Megabit, etc. If you don't do this, then when your VoIP traffic goes up to 1 Meg RouterOS does not know to pull bandwidth from other queues. In our case, we have specified that VoIP has the number 1 priority, not to mention it can use all 3 Meg of bandwidth if needed. In most cases, this would not occur, but we don't limit the bandwidth to something small.

With all of this you will need to create mangle rules that apply to your network, remember that what someone else creates is not necessarily what you want! 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 wish 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 Meg Up and 1 Meg Down connection, you can ensure the VoIP and Web traffic are at a higher priority than their P2P traffic.

To do this, you will need to mark your data twice. The first mark is done in your prerouting chain. This is where you will mark data based on traffic. You would identify web traffic, p2p, mail, and VoIP here. Once you do this, you would then create a 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 though you will need to keep your mangle rules down, as having lots of mangle rules will create load on your system. This system will allow you to have not only customer queuing on an individual simple queue, but also will 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; this affects your network more if you are allowing them a substantial amount of bandwidth. If you use PCQ systems, this typically should balance out your data and customers, so one customer will not negatively affect the rest of your users. Regardless of your reasons, I find it interesting that we 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, let's identify that by a download that has went over 10 Meg of data, we then can separate that data out from that customers individual queue and group all of these large downloads together into one small pool.



To do this, we have two steps. One is identifying that large download data. This will be done 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 bytes option under the advanced tab of your mangle. In our case, 10 Meg is roughly 10,240,000 bytes. Once we get a connection that goes over that in bytes, 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 than all of our customer's individual queues. The reason we put it above our individual queues, is because we want this data to match before their queues. Once their connection goes above that 10 Megabytes 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 large connections into one queue with very limited bandwidth.

General	Advanced	Statistics	Traffic	Total	Total Statistics
Name: <input type="text" value="Large Connections"/>					
Target Address: <input type="text"/>					
<input checked="" type="checkbox"/> Target Upload <input checked="" type="checkbox"/> Target Download					
Max Limit: <input type="text" value="200k"/> <input type="button" value="v"/> <input type="text" value="200k"/> <input type="button" value="v"/> bits/s					

[Setting Multiple PCQ Rates](#)

We have covered quite a few different ways of limiting traffic and setup 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 say from 0-10240000 bytes and mark that data with a packet mark. This gives us connections under 10 Meg to allow bursting on. How do we do 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 2M.

Setting up your simple queues is complicated though in 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

What occurs is that as new connections are being created, until they are at 10 Meg of data transferred, 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 Meg transfer per connection, 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 simple as creating multiple PCQ rules. The difference is instead of identifying traffic via 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

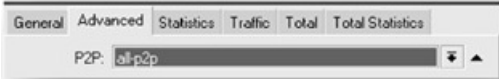
With this system, now each one of your customers get different PCQ bandwidth packages. If you wished to as well you can create a Queue that has a limit-at that guarantees bandwidth over other queues. Keep in mind that you will also need to specify the SFQ queue type in your parent. If you do this, you may have business customers that are guaranteed bandwidth vs. your other customers. 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 creates traffic when users are not at home, or at their PC. Plus, 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 usage. Second, especially for WISPs is access point time. Having many small connections going, streams a large amount of packets per second to your access point, using up access point time, 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 amount of connections P2P applications can open up. The primary way 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.



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 vs. 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 as well as the bandwidth in their 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 this though, I match P2P across all data going through the RouterOS system; this 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 if 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. This is 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, when hotspots can run on any TCP/IP medium including Ethernet, and 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 wish users to pay for hotspot services (Internet access) by allowing enough usage to get the users to process a credit card in return 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 hotspots on wireless. Setting them up is very easy as well with RouterOS. I wanted to touch though on one of the biggest common mistakes I see business and engineers doing with RouterOS 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 that 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 that, but chances are 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 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, 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 that would like to have Internet connections are a potential place for a hotspot. Hotels, and coffee shops are great places, as well as 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 would put in a free hotspot system to attract more coffee drinkers and business people. The idea is that now they can stay connected 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. Upon the hotel guest starting their web browser, they will get these ads delivered to their screen before they are allowed on the Internet. Of course, this could be a great benefit to the local pizza place!

You can do this though also with 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 an account 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 integrated security system for small hotspots as well as trial users to allow users limited access for specific amounts of time. You can use a centralized radius server to allow network access as well as 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, and 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 vs. 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 will 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; 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 shops website we talked about in the free hotspot section. Items that you list in RouterOS walled-garden users will be able to access 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, and 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

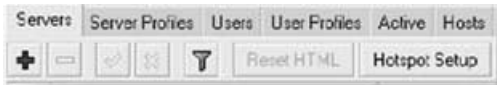
RouterOS offers an IP Binding system. This allow you to setup one-to-one NAT translation, allows you to bypass login/authentication requirements to specific hosts as well as allows you to block specific hosts and subnets from your hotspot system.

Hotspot Interface

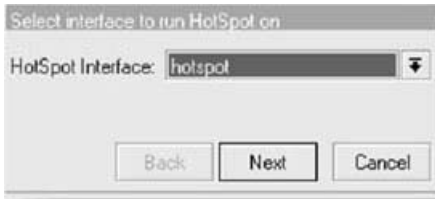
Hotspots run at layer 2 in the OSI model. Therefore they are applied to an interface. When you apply a hotspot to an interface, once the setup is complete, you will assume that all devices, MAC addresses, and IPs behind that interface must authenticate somehow. For this reason if you place a hotspot 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. So in our example we will use 10.5.5.1/24 as the IP address on our hotspot interface. So place this IP on your interface. Next we will configure the hotspot. RouterOS makes a wizard that I recommend you using. You will find this in the IP → Hotspot menu options.



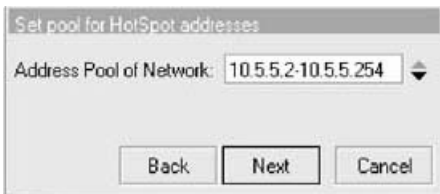
Here you will find a Hotspot setup button. This wizard does a number of things that you will need to have or your hotspot will not function.



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.

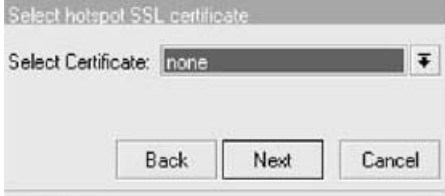


Next, you will select the local address of your hotspot network. This address is the IP address of your hotspot network, plus it asks if you wish to masquerade the hotspot network. During the

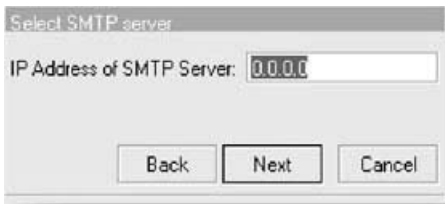


setup process, it will add the correct NAT rule if you wish it to.

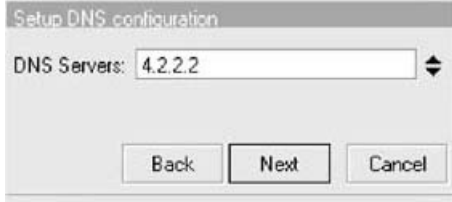
Here it creates an address pool to hand out DHCP with. Here you can select the IPs in the pool. Normally, you can accept the defaults here, however, sometimes if you have more access points out there that you would like to manage, you may reserve some IPS for these other devices on your hotspot network.



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.



The SMTP address is what 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

Back Next Cancel

These are the DNS server information to place in your hotspot system. Remember that DNS is a very important part of your hotspot system. List your DNS servers here.

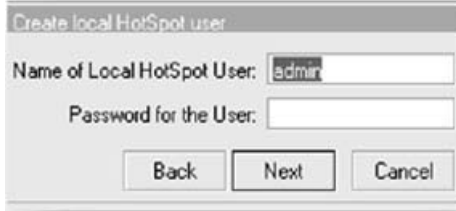


DNS name of local hotspot server

DNS Name:

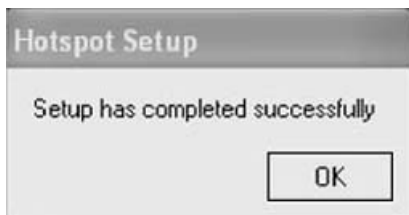
Back Next Cancel

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.



A dialog box titled "Create local HotSpot user". It contains two text input fields. The first field is labeled "Name of Local HotSpot User:" and contains the text "admin". The second field is labeled "Password for the User:". Below the fields are three buttons: "Back", "Next", and "Cancel".

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.



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 interface. 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 that IP pool that the DHCP server uses as well. It also enters static DNS information in if you plopped in the DNS name in that step; otherwise, the hotspot IP address is used.

As you can see there are lots of functions that need to occur to get the splash page from a RouterOS system. I do recommend using the wizard as well as it does all of the things you need it to in one easy to use interface.

Configuration of Servers and Server Profiles

The servers and server profiles tab gives 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 has paid Internet access normally, however that week a group of MikroTik people come in for a MUM and MikroTik paid for free Internet access. By having two different hotspot profiles, you can effect a major change. This includes the splash page, the method of authentication, etc 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 staying on-line even though they have left. The default for this option is five minutes, and I have found this to be way 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 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 Meg Internet connection and you don't want the hotspot to use more than 2 Meg 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.

General

Login

RADIUS

— Login By —

☐ MAC

☒ HTTP CHAP

☐ HTTP PAP

☒ Cookie

☐ HTTPS

☐ Trial

MAC Auth. Password:

HTTP Cookie Lifetime:

SSL Certificate: 

☐ Split User Domain

Trial Uptime Limit:

Trial Uptime Reset:

Trial User Profile: 

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 there 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 understand how this feature works 100%. So 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. We may have an option for that user to create a paid account, but the user ends up not using 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 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 though, 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 and 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 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, another way to use this feature is to tell the trial users that they have a slow Internet connection and of course, paying for Internet access would give them much faster Internet access. In some cases, we created filters for the trial users that prohibit most Internet activity with the exception of web surfing. We could

even add websites that we don't want the free Internet users (trial users) to get to.

Hotspots with Radius

RouterOS has two methods of authenticating users. The Internal method is to use the built in user database that RouterOS offers. This is the users and user profiles tabs inside 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. Inside here you will have various radius options as well as the ability to send radius accounting information as well. The MAC format is used if 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.

General

Login

RADIUS

☒ Use RADIUS

Default Domain:

Location ID:

Location Name:

MAC Format:

☒ Accounting

Interim Update:

NAS Port Type:

Internal Hotspot User Management

The image shows a screenshot of the RouterOS User Management configuration window, specifically the 'General' tab. The window has three tabs: 'General', 'Limits', and 'Statistics'. The 'General' tab is active. The configuration fields are as follows:

- Server:** A dropdown menu with 'all' selected.
- Name:** A text input field containing 'user1'.
- Password:** An empty text input field.
- Address:** An empty text input field with a dropdown arrow on the right.
- MAC Address:** An empty text input field with a dropdown arrow on the right.
- Profile:** A dropdown menu with 'default' selected.
- Routes:** An empty text input field with a dropdown arrow on the right.
- Email:** An empty text input field with a dropdown arrow on the right.

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. 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 amount of users that can share the profile, rate-limits, filters, packet marks, scripts and any advertisements.

Under the users section, you can say 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 can be automatically created.

The image shows a screenshot of the Mikrotik WinBox interface for configuring a User Profile. The window has three tabs: 'General' (selected), 'Advertise', and 'Scripts'. The 'General' tab contains the following settings:

- Name:** uprol1
- Address Pool:** none
- Session Timeout:** (empty field with a dropdown arrow)
- Idle Timeout:** none
- Keepalive Timeout:** 00:02:00
- Status Autorefresh:** 00:01:00
- Shared Users:** 1
- Rate Limit (rx/tx):** (empty field with a dropdown arrow)
- Incoming Filter:** (empty field with a dropdown arrow)
- Outgoing Filter:** (empty field with a dropdown arrow)
- Incoming Packet Mark:** (empty field with a dropdown arrow)
- Outgoing Packet Mark:** (empty field with a dropdown arrow)
- Open Status Page:** always
- ☒ **Transparent Proxy**

Under the User Profiles, you have lots of options here. This includes 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 is 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 would be a page with an 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 till 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-up from coming up, and this could lead to confusion by the end user. I recommend 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

IP Bindings is a way to setup One to One NAT translations, but also it's used to bypass hotspot clients without authentication. You can also use it to block specific hosts or subnets as well.



The image shows a configuration window for IP Bindings. It contains five fields, each with a label and a dropdown arrow on the right:

- MAC Address: [empty field]
- Address: 0.0.0.0
- To Address: [empty field]
- Server: all
- Type: regular

The most common feature I use IP bindings for is to bypass other access points and routers IPs 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 but 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, as that will mess up the works. The best way of doing this is 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 into your bindings system, this will prevent typos with the MAC and IP address.

[Creating Walled Garden Entries](#)

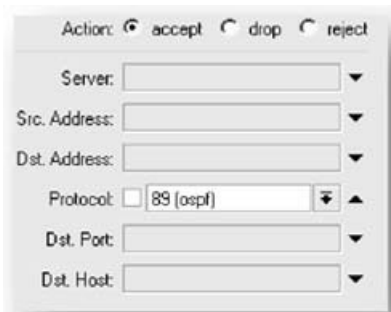


The image shows a configuration window for a Walled Garden entry in RouterOS. At the top, there is an 'Action:' label with two radio buttons: 'allow' (which is selected) and 'deny'. Below this are several input fields, each with a dropdown arrow on the right: 'Server:', 'Src. Address:', 'Dst. Address:', 'Method:', 'Dst. Host:', 'Dst. Port:', and 'Path:'. The 'Dst. Host:' field is currently filled with the text '*.linktechs.net' and has a small square checkbox to its left. The 'Dst. Host:' field also has a small upward-pointing arrow on its right side.

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 wished to allow customers 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 etc, 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 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 Rate	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	02:39:11	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
D	00:0F:66...	10.59.0.30	10.59.0.30	hotspot1	01:30:44	0 bps	0 bps

On the left we have letters that identify what the host is currently doing. A would be 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 work. H is plan hosts that typically have received a DHCP address.







General
Statistics
Traffic

MAC Address: 00:0F:66:C1:00:D5
Address: 10.59.0.69
To Address: 10.59.0.69
Server: hotspot1
Bridge Port: unknown

OK
Remove
Make Binding

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. This is 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 authenticated users via either radius or the internal

database. This list will show you their uptime, 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	gio...		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 22:31:39	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.

One 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, 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, we find an issue that if the other routers are not bypassed, or authenticated in some way, the dynamic routing protocol packets are not accepted and dropped. This is the correct thing that the hotspot should do on the hotspot interface. Routers behind your hotspot interface should not have this issue; it is simply the interface that the hotspot is actually running on that does. 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 300ms should be fine. Keep in mind 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 1000ms.

General

Status

Service

☐ ppp

☐ login

☐ hotspot

☐ wireless

☐ dhcp

Called ID: Domain: Address: Secret: Authentication Port: Accounting Port: Timeout: ms☐ Accounting BackupRealm: Src. Address:

The Service 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.

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 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 it to move through the list.

Troubleshooting Radius Client Issues

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. That would show that everything is working correctly. If the radius server is getting timeouts, we 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.

General	Status
Pending:	0
Requests:	4909
Accepts:	4844
Rejects:	63
Resends:	7
Timeouts:	2
Bad Replies:	1
Last Request RTT:	80

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

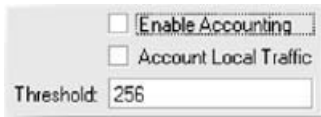
from your radius server. In the image above it is taking 80ms. IF your timeout value is under 80, then you will get more timeouts, however, if you see higher times, you may need to simply bump your timeout value accordingly to give your server time to respond.

Nuts and Bolts

RouterOS has many features, and quite a few of them do not require a huge section, however, there is some great information in this section and I recommend that you read up on all of the nifty features. Some of these have also been covered in part somewhere else in the book.

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

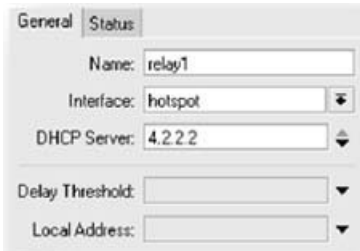
Threshold: 256

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 with a max of 8192. Once you have 8192 IP pairs, anything left over or unmatched will go into an uncounted counter. You can take a snapshot; this does two things, displays the IP Pairs along with their counters, but also clears out the table.

Most people will use this with some form of data collection application. You would normally enable the web-access system, 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.

DHCP Relaying

We covered DHCP server and client in other chapters, but we did not cover much about DHCP relaying! 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.



The image shows a configuration window for DHCP Relay. It has two tabs: 'General' and 'Status'. The 'General' tab is active. The fields are as follows:

Field	Value
Name:	relay1
Interface:	hotspot
DHCP Server:	4.2.2.2
Delay Threshold:	[Empty]
Local Address:	[Empty]

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.

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 does use UDP Protocol 5678 and broadcasts every 60 seconds. It only discards routers that have been removed after 180 seconds.

Neighbors

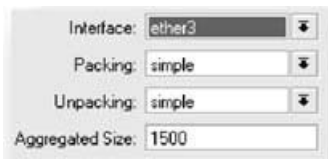
Discovery Interfaces



	Interface	IP Address	MAC Address	Iden...	Platform	Version	Board Name	Age (s)
	 ether1	172.25.200.2	00:0C:42:30:2A:D1	jmh...	MikroTik	4.0beta3		32
	 ether1	172.25.0.124	00:0C:42:0F:01:DA	jm...	MikroTik	3.19		30

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 this. Something like VoIP packets that are very small packets, around 100 bytes, could be combined into one larger packet and transmitted faster and quicker. In some cases this could 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.

A screenshot of a MikroTik configuration window for M3P. It contains four fields: 'Interface' with a dropdown menu showing 'ether3', 'Packing' with a dropdown menu showing 'simple', 'Unpacking' with a dropdown menu showing 'simple', and 'Aggregated Size' with a text input field containing '1500'. Each dropdown menu has a small arrow icon to its right.

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 the simple packing types and see what your CPU does before doing compression etc.

Pools

IP → Pools is your IP pools for both your DHCP and other systems like your universal client in your hotspot. Most of the time your IP pools is setup by other processes like when you setup your DHCP servers etc. The pool is exactly what it sounds like. It 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.



A screenshot of the RouterOS IP Pool configuration window. The window has a light gray background and a thin border. It contains three main fields: 'Name' with the value 'hs-pool-2', 'Addresses' with the value '10.5.5.2-10.5.5.25' and a small diamond-shaped icon to its right, and 'Next Pool' with the value 'none' and two small square buttons with up and down arrows to its right.

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 have one. To configure this, click IP → Socks. In the settings you will enable this as well as set the port, timeout and the max connections. Under the access tab, you will create access rules, to allow access to your socks server.

<input type="checkbox"/>	Enabled	
Port:	1080	
Connection Idle Timeout:	00:02:00	
Max Connections:	200	
Src. Address:	<input type="text"/>	▼
Src Port:	<input type="text"/>	▼
Dst. Address:	<input type="text"/>	▼
Dst. Port:	<input type="text"/>	▼
Action:	accept	⬇

[Clock](#)

The clock is the time and date system for your RouterOS system. RouterBoard systems do NOT remember the date/time upon 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.

The image shows a screenshot of the 'Time' configuration window in RouterOS, specifically the 'Manual Time Zone' tab. The window has a light gray background and a standard Windows-style border. At the top, the 'Time' tab is selected, and 'Manual Time Zone' is the active sub-tab. Below this, there are four input fields: 'Date' with the value 'Aug/05/2009', 'Time' with the value '13:46:54', 'Time Zone Name' with a dropdown menu showing 'manual' and a downward arrow, and 'GMT Offset' with the value '+00:00'. At the bottom, there is a checkbox labeled 'DST Active' which is currently unchecked.

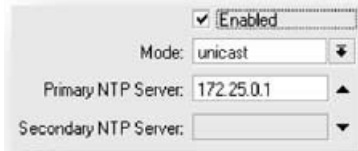
Field	Value
Date	Aug/05/2009
Time	13:46:54
Time Zone Name	manual
GMT Offset	+00:00
DST Active	<input type="checkbox"/>

NTP

Since RouterBoard do not have a clock, RouterOS does support NTP or the Network Time Protocol. 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.

Client

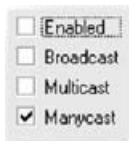
To configure the NTP Client, all you have to do is enable it, specify what mode to operate in and the IPs of the NTP servers if applicable. One Example could be us.pool.ntp.org A domain will resolve to an IP address if the DNS is working correctly on the RouterOS.



A screenshot of the RouterOS NTP server configuration window. It features a checked checkbox labeled 'Enabled'. Below it, the 'Mode' is set to 'unicast' in a dropdown menu. The 'Primary NTP Server' field contains the IP address '172.25.0.1'. The 'Secondary NTP Server' field is currently empty.

Server

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 request should it respond to. It will reply with the date and time from its clock.



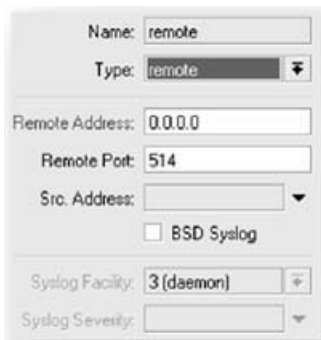
A screenshot of the RouterOS NTP server request type selection window. It contains four checkboxes: 'Enabled' (unchecked), 'Broadcast' (unchecked), 'Multicast' (unchecked), and 'Manycast' (checked).

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. This does show up on WinBox Discovery as well as the IP Neighbors Discovery systems.

Logging

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, etc; you will need to specify the remote address as well as the remote port for this to work.



The screenshot shows the configuration window for a logging action in RouterOS. The fields are as follows:

Name:	remote
Type:	remote
Remote Address:	0.0.0.0
Remote Port:	514
Src. Address:	
<input type="checkbox"/> BSD Syslog	
Syslog Facility:	3 (daemon)
Syslog Severity:	

Once you have setup your logging actions, you will then need to setup logging rules. The rules say 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.

Rules		Actions	
<div><div>+</div><div>=</div><div>✓</div><div>✗</div><div>Y</div></div>		<div>Find</div>	
Topics	Prefix	Action	
critical		echo	
error		memory	
info		memory	
warning		memory	

Reset Configuration

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 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.

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.

The image shows two overlapping configuration windows from the RouterOS WinBox interface. The top window is for creating a new script, and the bottom window is for scheduling a task.

Script Configuration Window:

- Name:
- Owner:
- Policy:
 - ☒ reboot
 - ☒ write
 - ☒ test
 - ☒ sniff
 - ☒ read
 - ☒ policy
 - ☒ password
- Last Time Started:
- Run Count:
- Source:

Schedule Configuration Window:

- Name:
- Start Date:
- Start Time:
- Interval:
- Delay:

On Event:

Owner:

Policy

<input type="checkbox"/> reboot	<input type="checkbox"/> read
<input type="checkbox"/> write	<input type="checkbox"/> policy
<input type="checkbox"/> test	<input type="checkbox"/> password
<input type="checkbox"/> sniff	

Run Count:

Next Run:

Scheduler

The schedule 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 wish scheduler to run at the correct time.

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: [REDACTED]

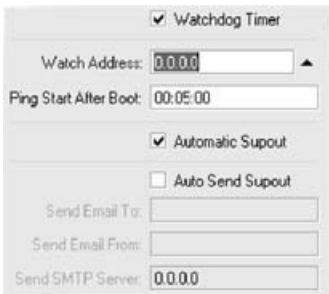
Once you have defined your package sources under your package source tab, you can then click on your available package list, and select refresh. This will force the system to go out to the FTP servers and download these lists. Remember that you must have the correct username/password in the package source 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.

Available Packages		Upgrade Package Sources			
	Refresh	Download	Download All...		
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

The watchdog system runs in two different modes, both a hardware and software mode. RouterBoards have a hardware based watchdog that this feature uses, but if you are on an x86 system, the watchdog runs as a software based service. By enabling the watchdog timer, RouterOS pings the watched address. Once it 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 time has passed it will attempt to ping again, this is to prevent the watching from constantly rebooting and gives us time to login to the radio and disable the watchdog if necessary.



The screenshot shows the Watchdog configuration window. It has a title bar with a checked checkbox for "Watchdog Timer". Below the title bar, there are several fields and checkboxes. The "Watch Address" field contains "0.0.0.0". The "Ping Start After Boot" field contains "00:05:00". There is a checked checkbox for "Automatic Supout" and an unchecked checkbox for "Auto Send Supout". Below these are three empty text fields for "Send Email To:", "Send Email From:", and "Send SMTP Server:". The "Send SMTP Server:" field contains "0.0.0.0".

<input checked="" type="checkbox"/> Watchdog Timer	
Watch Address:	0.0.0.0
Ping Start After Boot:	00:05:00
<input checked="" type="checkbox"/> Automatic Supout	
<input type="checkbox"/> Auto Send Supout	
Send Email To:	
Send Email From:	
Send SMTP Server:	0.0.0.0

Bandwidth Test Server

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 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. Inside here we have an option for our BTest Server Settings. Here we can enable or disable the server, specify the max number of bandwidth test sessions as well as 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.



<input checked="" type="checkbox"/> Enabled
<input checked="" type="checkbox"/> Authenticate
Allocate UDP Ports From: 2000
Max Sessions: 100

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 can not generate enough traffic to test even a 100meg Ethernet connection, where the 600 or 1000 RouterBoard would be able to. 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 would 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 in 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 vs. just a single one, therefore giving you the ability to test over 600 Meg connections.

Test To:

Protocol: ☒ udp ☐ tcp

Local UDP Tx Size:

Remote UDP Tx Size:

Direction: ▼

TCP Connection Count:

Local Tx Speed: ▼ bps

Remote Tx Speed: ▼ bps

User: ▼

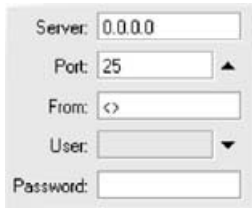
Password: ▼

Tx/Rx 10s Average:

Tx/Rx Average:

E-Mail System

MikroTik put in a function 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.



Server: 0.0.0.0

Port: 25 ▲

From: <>

User: ▼

Password:

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 correct with your outbound SMTP Server, the email should go out without issues. These commands inside scripting will allow you to send e-mails upon someone logging 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 from FTP or HTTP, just simply enter the address, if there is a username/password as 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.



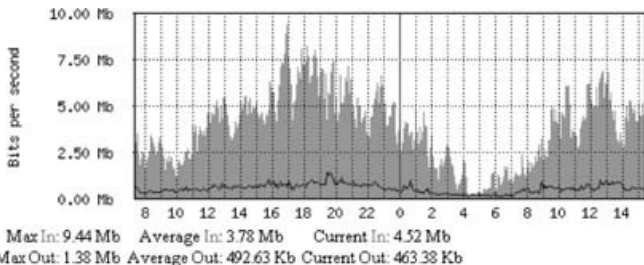
Inside your rules, you will have the option to turn on graphing for each of the different types of rules. You also can turn on the allowed addresses to view this graph. With x86 RouterOS system, you can store data to a disk and they will be on-

line even after a reboot. RouterBoard products do not store this information regardless if you have the store on disk selected. You will get 4 different graphs, 1 hour, 1 week, 1 month and 1 year graphing

Allow Address:

☒ Store on Disk

"Daily" Graph (5 Minute Average)



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.

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 more data than if you had the entire packet. You can also save that data into a data file if you wish by specifying a file name and limit.



The screenshot shows the 'General' tab of a packet sniffer configuration window. It contains the following fields and controls:

- Interface:** A dropdown menu currently set to 'all'.
- Memory Limit:** A text input field containing '10' followed by a 'kb' unit.
- Only Headers:** An unchecked checkbox.
- File Name:** A text input field with a dropdown arrow on the right.
- File Limit:** A text input field containing '10' followed by a 'kb' unit.

Once you have selected your general information, you may wish to filter that data so that you only look at frames, or only IP information via 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 it later.

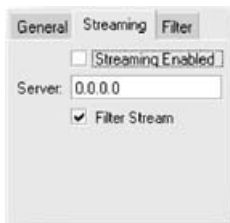


The screenshot shows the 'Filter' tab of the same packet sniffer configuration window. It contains the following fields and controls:

- Protocol:** A dropdown menu currently set to 'ip only'.
- Address 1:** A text input field containing '0.0.0.0/0'.
- Port 1:** A dropdown menu.
- Address 2:** A text input field containing '0.0.0.0/0'.
- Port 2:** A dropdown menu.

[Streaming Packet Sniffer Data](#)

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 Ethereal or Wireshark. Others could be used though as well.



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 are being generated by itself to sent to you. I always use filter stream.

[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 and add what IPs are allowed to read from the server, and set what file names you wish to have. It also has options if the file can be written to or not.



IP Addresses: 

Req. Filename:

Real Filename:

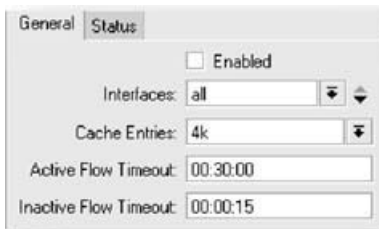
☒ Allow

☒ Read Only

Hits:



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. These applications are outside the context 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.



General Status

☐ Enabled

Interfaces:  

Cache Entries: 

Active Flow Timeout:

Inactive Flow Timeout:

A screenshot of a web-based configuration interface for RouterOS. It features five input fields: 'Address' with the value '0.0.0.0', 'Port' with the value '1234', 'Version' with a dropdown menu showing '5', 'v9 Template Refresh' with the value '20', and 'v9 Template Timeout' with the value '1800'.

Address:	0.0.0.0
Port:	1234
Version:	5
v9 Template Refresh:	20
v9 Template Timeout:	1800

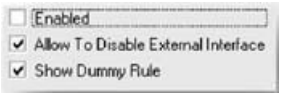
The first thing to do is to enable and configure the basic information about your traffic flow system. Under the traffic flow settings button, you can enable and disable the system, as well as specify what interface, how much data to cache, and specify timeout values. Next, you will need to create a 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.

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.

[UPnP](#)

Universal Plug and Play applications communicate with your RouterOS system to open and forward ports through NAT that is necessary for the application or UPnP system to function correctly. For the ISP or WISP, I would not configure UPnP on any core devices, only customer CPEs would be normal for us to configure 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 a 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 is

used. It is used to simply make holes in your NAT system easy for software and devices.



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 are no more configurations needed with UPnP. If the device is working correctly you should see Dynamic NAT rules created for specific ports to forward to internal addressing

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 San. Then select what interface you wish to run the scan on, and the address range you wish to scan. When you run it, it 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	Start			
Address Range:	172.25.0.0/24	Stop			
		Close			
		Find			
Address	MAC Address	Time (ms)	DNS	SNMP	Netbios
172.25.0.1		0	core.linktechs...		

Web Proxy

The Web proxy system provides 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 has a hit rate over 45.9%. That means for every Gig of data passed through the web proxy system, 1.459 Gig of data was passed through to the customer. So you saved almost $\frac{1}{2}$ of a Gig of data that would have otherwise been moved through your Internet connection.

General	Status	Lookups	Inserts	Refreshes
<input checked="" type="checkbox"/> Enabled				
Src. Address:		<input type="text"/>		
Port:		<input type="text" value="8989"/>		
Parent Proxy:		<input type="text"/>		
Parent Proxy Port:		<input type="text"/>		
Cache Administrator:		<input type="text" value="webmaster"/>		
Max. Cache Size:		<input type="text" value="65325000"/> KB		
<input checked="" type="checkbox"/> Cache On Disk				
Max. Client Connections:		<input type="text" value="2000"/>		
Max. Server Connections:		<input type="text" value="2000"/>		
Max Fresh Time:		<input type="text" value="190d 00:00:00"/>		
<input type="checkbox"/> Serialize Connections				
<input type="checkbox"/> Always From Cache				
Cache Hit DSCP (TOS):		<input type="text" value="10"/>		
Cache Drive:		<input type="text" value="primary-slave"/>		

The web caching system is configured in RouterOS by going to IP → Web Proxy. Under the access tab 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 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 caching to disk, 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 connections are important if you have a large system with many connections. This simply 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, it 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 page contains this data, it will be used, however, many pages do not, and the max fresh time is the default timeout value for them.

The cache hit DSCP a TOS bit 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. You can do a number of things, but the best one I like to do is specify an unlimited queue to deliver data from the caching system as fast as possible outside of the customer's normal queues.

General	Status	Lookups	Inserts	Refreshes
Uptime: 6d 04:01:17				
Requests: 218951				
Hits: 47288				
Cache Used: 53 588 839 KiB				
Total RAM Used: 3 526 KiB				
Received From Servers: 16 055 713 KiB				
Sent To Clients: 16 695 499 KiB				
Hits Sent To Clients: 596 756 KiB				

The status tab of your web proxy settings will show you all of the stats that you will need to evaluate how your web proxy system is working. The most important numbers I use are how much the hits sent to clients will go into the sent to clients number. In this case, 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 to the left, 53 Gig of data is stored

[Web Proxy Access List](#)

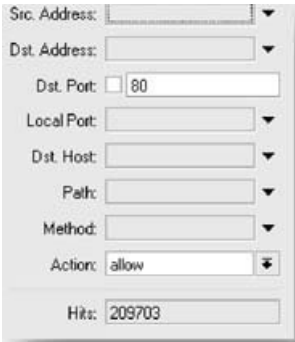
The access list is a list of IPs, and or ports and protocols that can use the web caching system. This is used to secure your web proxy system. I recommend a setup like the following image:

#	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.



A screenshot of a web proxy configuration window. The window contains several input fields and dropdown menus for configuring rules. The fields are labeled as follows:

- Src. Address: [Empty text box]
- Dst. Address: [Empty text box]
- Dst. Port: ☐ 80
- Local Port: [Empty text box]
- Dst. Host: [Empty text box]
- Path: [Empty text box]
- Method: [Empty text box]
- 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.

General Advanced Extra Action Statistics

Chain: dstnat

Src. Address: 172.25.0.0/24

Dst. Address:

Protocol: 6 (tcp)

Src. Port:

Dst. Port: 80

General Advanced Extra Action Statistics

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 what their settings are in their browser.

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 to the production system. Most small flash drives though do not take long. The hard disks upwards of 60+ Gigabytes of storage take a considerable amount of time.



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 to remain on your primary system disk, and the web proxy data 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

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, but not actually have 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.



A screenshot of a configuration window for a MetaRouter. It contains four fields: 'Name' with the value 'mr1', 'Memory Size' with the value '16' and unit 'MiB', 'Disk Size' with an empty field and a dropdown arrow pointing to 'KiB', and 'Used Disk' with the value '0 KiB'.

Notice that inside the MetaRouter options, you can also reboot, shutdown and start these, as well gain console access to the MetaRouter.



A vertical stack of four buttons: 'Console', 'Start', 'Shut down', and 'Reboot'.

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, what occurs is an interface is created on the MetaRouter and that upon starting attaches to a bridge group on the physical router.



I have used MetaRouters 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. So make sure you do not need to move lots of data through these types of systems.

Dynamic Routing

When you are using RouterBoards with at least a level 4 license, you will have the ability to run all of the Dynamic Routing protocols offered. In an x86 version of RouterOS, you will have to have a Level 5 or better to run BGP. A level 4 license with x86 systems does allow you to use RIP and OSPF.

There are many books on OSPF, BGP and RIP as well, I will run through the basic setup, however we will not discuss routing techniques, and most troubleshooting abilities 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.

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

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

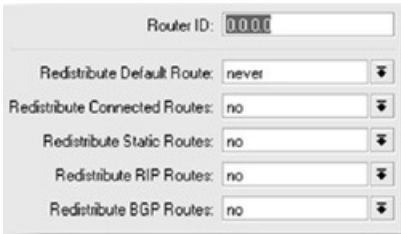
Bad Routes: 0

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 if you wish.

Address: 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 state changes. So if you unplug an Ethernet cable, OSPF sends an update. When you plug it back in, OSPF sends another update, therefore can affect routing changes very quickly. OSPF uses Protocol 86 to communicate between routers. To configure OSPF, click on Routing→OSPF in your WinBox Application.



The image shows a configuration window for OSPF. At the top, there is a field for 'Router ID' with the value '0.0.0.0'. Below this, there are five rows of configuration options, each with a label, a text field, and a dropdown arrow:

Option	Value
Redistribute Default Router:	never
Redistribute Connected Routes:	no
Redistribute Static Routes:	no
Redistribute RIP Routes:	no
Redistribute BGP Routes:	no

Inside your interfaces tab you can also click the OSPF Settings button. Here, just like RIP you will have the options on what routes to distribute, as well as change default metrics.

If you do not wish to secure your OSPF communication, I don't recommend this, 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.

Changing Path Costs

OSPFs 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.

General

Status

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 however perform only in 1/2 duplex, such as wireless links, you can set these links up to create a 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 for 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.

BGP

We do have full support for BGP within RouterOS. BGP or Border Gateway Protocol is the key protocol on the Internet. It supplies interdomain 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 are 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 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 thing to start. 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.



The image shows a configuration window for a BGP instance in RouterOS. The window has a light gray background and contains several fields and checkboxes. At the top, there is a 'Name' field with the value 'default'. Below it is an 'AS' field with the value '65530'. The 'Router ID' field is empty and has a dropdown arrow. Below these fields is a section with five checkboxes, all of which are unchecked: 'Redistribute Connected', 'Redistribute Static', 'Redistribute RIP', 'Redistribute OSPF', and 'Redistribute Other BGP'. Below the checkboxes is an 'Out Filter' field with a dropdown arrow. Below that is a 'Confederation' field with a dropdown arrow. Below that is a 'Confederation Peers' field with a dropdown arrow. Below that is a 'Cluster ID' field with a dropdown arrow. At the bottom, there are two checkboxes: 'Client To Client Reflection' which is checked, and 'Ignore AS Path Length' which is unchecked.

Name:	default
AS:	65530
Router ID:	
<input type="checkbox"/> Redistribute Connected	
<input type="checkbox"/> Redistribute Static	
<input type="checkbox"/> Redistribute RIP	
<input type="checkbox"/> Redistribute OSPF	
<input type="checkbox"/> Redistribute Other BGP	
Out Filter:	
Confederation:	
Confederation Peers:	
Cluster ID:	
<input checked="" type="checkbox"/> Client To Client Reflection	
<input type="checkbox"/> Ignore AS Path Length	

You will also have the options to redistribute 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 would also setup an out filter here as well. You would not wish to distribute IPs that are not yours nor IPs that are not valid, such as private addresses.

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. However, you can use BGP multi-hop as well to provide a BGP peer. We will cover that a bit further in the chapter.

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. You will specify the instance that you will be using, the remote IP for your peer, as well as port, 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.

General Advanced Status

Name:

Instance:

Remote Address:

Remote Port:

Remote AS:

TCP MD5 Key:

Nexthop Choice:

☐ Multihop

☐ Route Reflect

Hold Time:

TTL:

Max Prefix Limit:

Max Prefix Restart Time:

In Filter:

Out Filter:

☐ Default Originate

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 as well, 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

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 is in the IGP routing table. For instance, if you put in the above network, 187.1.1.0/24, and check the Sync 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.

A small dialog box with a light gray background. It has a label "Network:" followed by a text input field containing "187.1.1.0/24". Below this is a checkbox labeled "Synchronize" which is currently unchecked.

Network: 187.1.1.0/24

☐ Synchronize

Aggregates

BGP Aggregates are meant to summarize or only send specific prefixes vs. the entire routing table. If you have an entire /24 subnet dedicated to /30 subnets, you don't wish to advertise 128 /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.

A configuration dialog box for BGP aggregates. It has a "Prefix:" label followed by a text input field containing "0.0.0.0/0". Below this are two checked checkboxes: "Summary Only" and "Inherit Attributes". There are three filter sections, each with a label and a text input field followed by a dropdown arrow: "Attribute Filter:", "Suppress Filter:", and "Advertise Filter:". At the bottom is a "Routes Used Count:" label followed by a text input field.

Prefix: 0.0.0.0/0

☒ Summary Only

☒ Inherit Attributes

Attribute Filter: ▾

Suppress Filter: ▾

Advertise Filter: ▾

Routes Used Count:

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 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.



The image shows a configuration window for a Routing Filter. It contains several input fields, each with a dropdown arrow on the right side. The fields are: Chain, Prefix, Prefix Length, Match Chain, Distance, Scope, Target Scope, Pref. Source, Routing Mark, Route Comment, and Tag.

Field	Value
Chain	
Prefix	
Prefix Length	
Match Chain	
Distance	
Scope	
Target Scope	
Pref. Source	
Routing Mark	
Route Comment	
Tag	

There are plenty of options here to match data with, that's the goal, just like a filter or mangle rule, but now you are matching routes! 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, as you want the data to run through your router, unless you wish to drop or discard routes. There are many options including BGP and BGP community options here as well as even RIP options. Most of these options though will be through BGP sessions. The ones that I commonly use are your BGP Prepends and local pref as well!

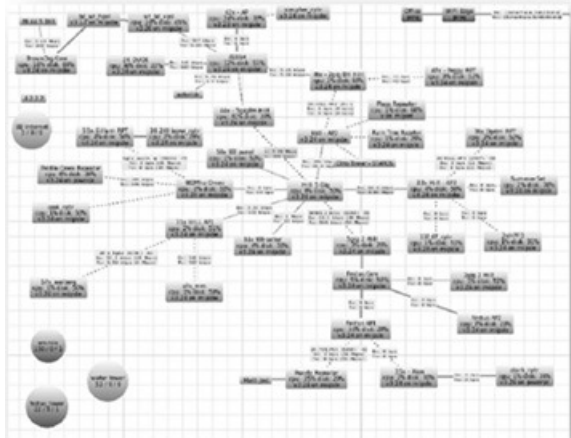
▲ BGP
 BGP AS Path: ▼
 BGP AS Path Length: ▼
 BGP Weight: ▼
 BGP Local Pref: ▼
 BGP MED: ▼
 BGP Atomic Aggregate: ▼
 BGP Origin: ▼
 ▲ BGP Communities
 BGP Communities: ▼
☐ Invert BGP Communities
☐ Invert Match

Action: ▼
 Jump Target: ▼
 Set Distance: ▼
 Set Scope: ▼
 Set Target Scope: ▼
 Set Pref. Source: ▼
 Set In Nexthop: ▼
 Set In Nexthop Direct: ▼
 Set Out Nexthop: ▼
 Set Routing Mark: ▼
 Set Route Comment: ▼
 Set Check Gateway: ▼
 Set Disabled: ▼
 Set Type: ▼
 ▲ BGP
 Set BGP Weight: ▼
 Set BGP Local Pref: ▼
 Set BGP Prepend: ▼
 Set BGP MED: ▼
 ▲ Set BGP Communities
 BGP Communities: ▼
 ▲ Append BGP Communities
 BGP Communities: ▼

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 perform upgrades quickly with just a few mouse clicks, but it is also a very powerful NMS or network monitoring system. You can monitor up and down status of virtually any kind of network device. You can setup SNMP probes 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 Dude to cost considerably, however, 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.

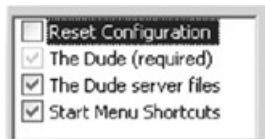


Installation

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 that you wish to use it on via Mikrotik's website at <http://www.MikroTik.com/thedude.php>. Here you can download the Dude for Windows, 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 application 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 has really 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, 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.



After the selection of the components to install, you will then select what folder you wish to install under. Now, 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. So 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 using. 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 does use RAM and after it monitors and collects data for a while, I have seen Dude installations grow considerably. I have used RouterBoard 433Ahs with a one or two gig Micro-SD card for storage of Dude Data. I have seen Dude data exceed 800 megabytes 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, when you make a backup of the Dude application, it creates a XML file, which it has to, yes you guessed it, store 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 instance, if you have multiple hotspot networks behind different types of broadband connections,

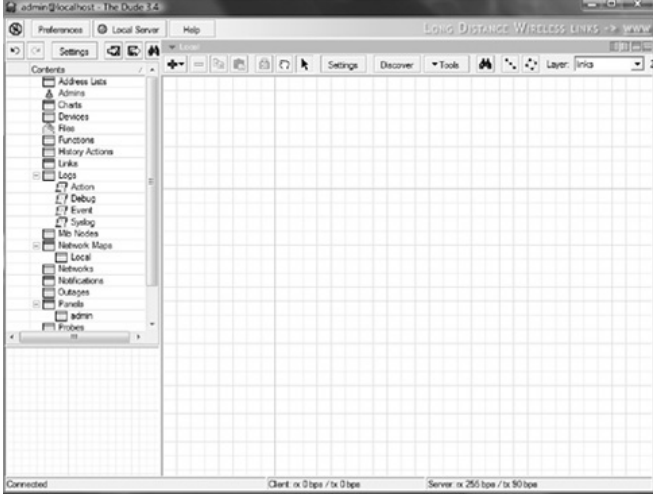
and these hotspots share the same common IP structure. In this case, if you had a single Dude server, you would only be able to ping and monitor devices with public IPs for the most part. However, with a Dude agent, your primary Dude server can request the Dude Agent that has both a public IP and a private 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 shows that you have a local server running by the green indicator light. If you click this, this will give you your options for the Dude server. If you uncheck the “Enable On Localhost” 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.



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”, will do exactly what it says! When you start the Dude client application, it will run, and when you close it, 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.





Menus and Options










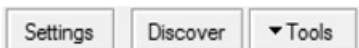
The left context menu has the management features of your Dude System. The first  is an undo command for The Dude. If you delete something you can undo this. There is also an undo contents list that shows the commands that you can undo. The  is a redo command option in case you wish to redo the undo!

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 really quick.

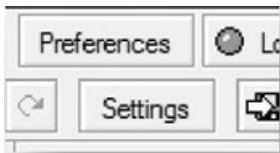
The import button, the one on the left, does the exact opposite of the export; it takes an import XML file and imports it into the system. Once done, the Dude server will restart and apply that configuration. This sometimes can take a few minutes on RouterBoards. You will see their CPU jump to 100% for 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,  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 etc. 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 that 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 help automatically layout devices.

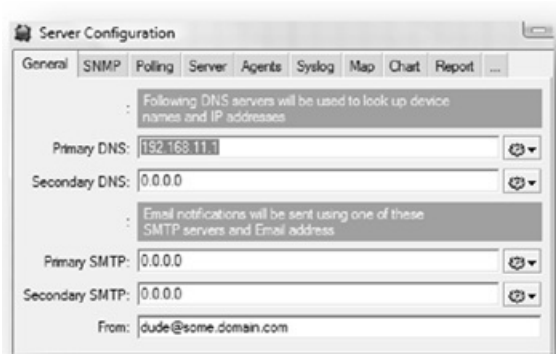
[Server Configuration](#)



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. So 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 what 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 unauthenticated 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 address regardless of authentication. I would opt for the IP address for security

reasons.



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.

General SNMP Polling Server Agents Syslog Map Chart Report ...

Default options for Simple Network Management Protocol (SNMP)

Default: v1 public

Name	Vers ...	Community	Port	Notes
v1-public	1	public	161	
v2-public	2c	public	161	
no-snmp	none			

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, as well as 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.

SNMP Polling Server Agents Syslog Map Chart Report Discover ...

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

☐

☒

Name
beep
flash
log to events
log to syslog
popup

Notifications:

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 etc 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 vs. 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 are Dude servers that function as an agent of a primary Dude server. These Dude servers allow you to relay your Dude probes through them. One of the usages 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 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, is it on-line or not, and 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 through 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 only are for data coming in on your Syslog server.

☒ Enable

Port: 514

#	Source Add...	Regexp	Action	Notification
1			accept	log to syslog

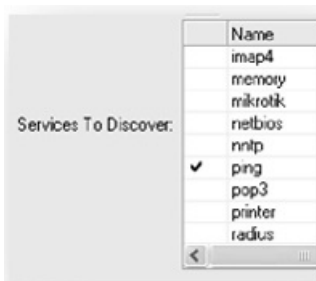
Dude Discovery Services

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 the 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 plenty, the discovery process can lead to a messy map, even though there are layout tools. I prefer to build them 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 specific types of probes I wish it to discover. Specifically pings are what I am mostly interested in, however sometimes CPU comes into play as well.

Note in the screenshot above, I have turned off most of the discovery services, and do not let the service run to a big network ether. The reason for doing something like this is to prevent the discovery service from starting to run, during its process, it sends out hundreds of connections and probes looking for devices etc. On a small network, this is ok, but on larger networks this can cause huge outages and have messy results. In the services to discover, I will leave on ping 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 opinions.



[Admins](#)

The admins section of the Dude is the user management system. This allows users to login to the Dude application, view network status as well as login abilities to the web access if applicable. Just like RouterOS' user management system, you have groups that you can specify rights and policies. The ability to login remotely, via web, locally, 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.

Admins	Groups	Active
		
		
		
Name	Group	
kalob	full	
admin	full	
rodneyb	read	
readonly	read	

Name:

— 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 oids 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.



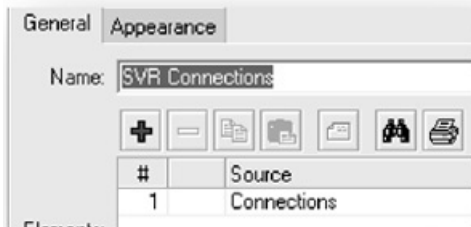
A screenshot of a configuration window for a data source. It contains several labeled input fields and dropdown menus. The fields are: Name (text box with 'Connections'), Type (dropdown menu with 'snmp oid'), Data (dropdown menu with 'gauge (absolute value)'), Scale Mode (dropdown menu with 'multiply'), Scale (text box with '1.000'), Unit (text box with 'Count'), and Rate (dropdown menu with 'none').

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 to 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 will be 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.

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. Now, 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 per the interval you put in the data source.



Devices

The devices pane gives you lots 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, 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

List	Tree	RouterOS	Types	Mac Mappings		
+	-					
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:50:00:12	Some Device	Master View	
▶	w_schum...	10.0.11.13	UbiquitiNe:64:35:FB		festus_tower	
▶	w_argana	10.0.11.91	UbiquitiNe:63:3D:CD		festus_tower	
▶	n_i_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 lots of information as well. You can add notes to each device; this is extremely useful when you are swapping radios etc. As well as 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.

RouterOS									
List Tree RouterOS Types Mac Mappings									
Device Group Wireless Registration Simple Queue									
Upgrade									
		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	2kn/es_2ksfes	3.24	mipsle	RB532		routeros-mipsle	
+	✓	ok	2kn/es	3.24	mipsle	RB532		routeros-mipsle	
+	✓	ok	31x.HILL API	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 to them.

Device Group Wireless Registration Simple Queue									
	Device	Radio Na...	MAC	AP	WDS	Tx/Rx Rate	Tx/Rx Sig...	Comment	Last IP
	bd_vpol	000C422...	Routerboa:28:52...	yes	no	48Mbps	-73		10.100.0.10
	bd_wl_h...	000C422...	Routerboa:28:53...	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...	Routerboa: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. Any 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 devices that you have listed! Along with their signals, IPs, and any comments you wish to place 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 Group Wireless Registration Simple Queue												
Device	Group	Wireless Registration	Simple Queue									
+	✓	✗	🔍									
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	2w_s			10.0.12.100/32			24231 kB	283 MB	20850	27995	1.6 kbps	2.01 kbps
WiFi Edge	2w_s			10.0.33.101/32	128 kbps	256 kbps	36.7 MB	35.2 MB	38370	117803	23.4 kbps	8.73 kbps
WiFi Edge	2w_s			10.13.91.0/24	256 kbps	1 Mbps	65.3 MB	748.6 MB	437917	854408	7.44 kbps	1.65 kbps
WiFi Edge	2w_s			10.13.92.254/32	128 kbps	512 kbps	24.1 MB	121.1 MB	232400	249950		
WiFi Edge	2w_s			10.0.96.251/32	256 kbps	1 Mbps	12.8 MB	507.4 MB	204086	380344		

This image shows the Simple queue tab under RouterOS. Again here, we can see queues, data rates, and limits setup from 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:

Probe Timeout:

Probe Down Count:

☐ Use Notifications

Name
E-Mail
beep
email-LTI
flash
log to events
log to syslog
popup
speak

Notifications:

Under the Polling tab, you have options on 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, yellow instead of red, showing that the device is reachable just some services are not responding. If all of the services are down, then the device will show as red, indicating all services have failed.

The outages tab is a wonderful history of 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 in the image we have the time/date and the duration of the outages just for this device.

Remove Resolved

	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

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, etc. This data is nice to be able to see inside the Dude, however, is mostly for show in the Device. The RouterOS tab is the exact same thing as the SNMP tab as well, 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.

Interface

Ip

Route

Arp

Bridge Fdb

Storage

Cpu

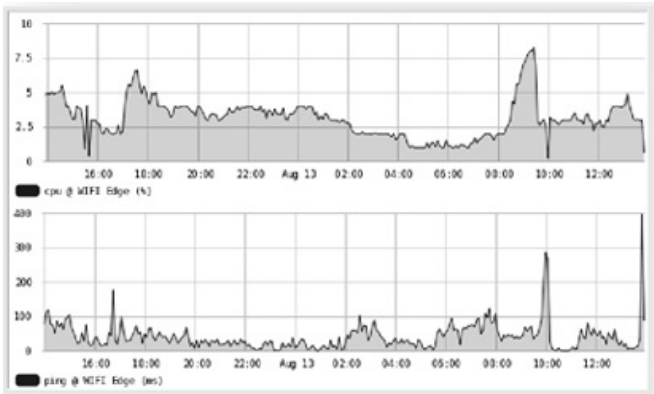
Wireless Station

Regis

	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 be a number of different types of graphs. For instance, 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 on 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

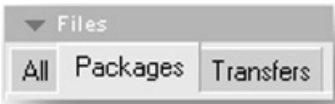
points and we included what channel and polarity

The screenshot shows the 'General' tab of a configuration window. The 'Type' is set to 'item' and the 'Item Type' is 'device'. A grey box contains the text: 'Map specific values of following settings are used for this item if not specified here'. Below this are two buttons: 'Insert Variable' and 'Insert Old'. The 'Label' field contains the text: '[Device Name] [device_performance([Device.ServicesDown],[Device.RosVersion]) on [Device.RosArchitecture]]'. The 'Label Refresh Interval' is set to 'Default'. Below this are several fields with dropdown arrows: 'Unknown:', 'Up:', 'Down Partial:', 'Down Complete:', 'Acked:', 'Shape:', and 'Font:'.

Files

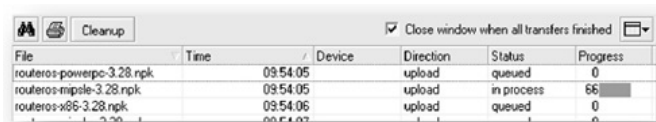
The Dude system contains a file system with two different areas. The all section is for files such as images, graphics etc for you to use with devices, background maps, etc. The second section, packages, is much more important. What these are for is to be able to upload packages, RouterOS packages to 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](#)

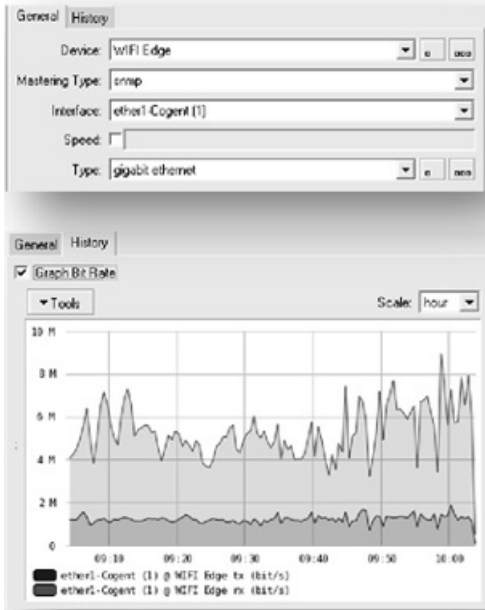
Uploading or downloading files to and from the Dude is a simple drag and drop action. Simply highlight the files you wish to move and drag and drop them into the packages window. At that point the system will upload or download the file as necessary. You will also get a transfer window, pictured below showing files that need to be transferred and are in process, as well as what was completed.



[Links](#)

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. As well as see history of the link. Making changes to the link can be done here also.

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, but 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.



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.

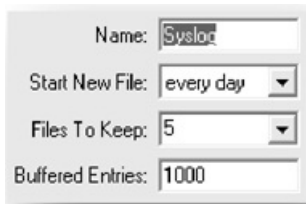
Name:	<input type="text" value="point to point"/>
Style:	<input type="text" value="dotted"/>
Thickness:	<input type="text" value="4"/>
Snmp Type:	<input type="text" value="ppp"/>
Snmp Speed:	<input type="text" value="20000000"/>

[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 use this feature with backup links. Specifically, I will set the link speed to something very small, say a few hundred k. Something that I know if the link actually starts getting used to another link failing, that link will immediately become red and be something that sticks out to be able to see that the link is being used. Normally there is an outage too, a radio down etc, however, not all of the time.

Logs

The logging system of the Dude contains three to 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.



A screenshot of a settings window for a logging system. It contains four rows of controls:

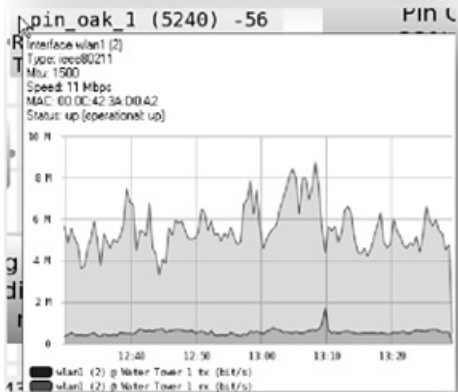
- Name:** A text input field containing the text "Syslog".
- Start New File:** A dropdown menu with "every day" selected.
- Files To Keep:** A dropdown menu with "5" selected.
- Buffered Entries:** A text input field containing the number "1000".

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 jewel 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 glance status indicators, green being good links and devices vs. red being down devices and overloaded links!

pin_oak_1 (5240) -56
Rx: 5.26 Mbps (122 Mbps)
Tx: 558 kbps (150 Mbps)



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, sudden increase or decrease in bandwidth usage, on a link.

Map Settings

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 to consider the device down. You also have the options here to setup what notification settings you want all of your devices to default too.

General

Polling

Appearance

Image

Export

☒ Enabled

Probe Interval: 00:00:30

Probe Timeout: default

Probe Down Count: 2

☒ Use Notifications

...

	Name
<input checked="" type="checkbox"/>	E-Mail
	beep
	email-LTI
<input checked="" type="checkbox"/>	flash
<input checked="" type="checkbox"/>	log to events
	log to syslog
	popup
	speak

General

Polling

Appearance

Image

Export

Label Refresh Interval: 00:00:30

The appearance tab allows you to change the color, looks of the map, including the default colors, 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 as you may have 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 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; here 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 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 have only a few services that you wish to monitor. You can though click on the plus sign and add individual services that you may wish to use.

Add services you want to monitor on this host

Discover

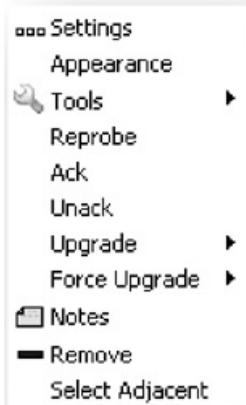
Type	Problem	Notes

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, 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 your 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

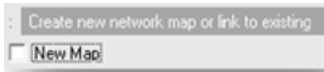
Links as described in the link section can be used to show bandwidth usage and stats on a link between two devices. To create a link, right-click on your network maps, and then select Add Link. Next, click and HOLD on one of the two devices you wish to create a link from and to, and drag your mouse from 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.

The image shows a screenshot of the 'Add Link' dialog box in the Dude network management software. The dialog has a light gray background and contains the following fields and controls:

- Device:** A dropdown menu with '1.1.1.1' selected.
- Mastering Type:** A dropdown menu with 'simple' selected.
- Speed:** A checkbox labeled 'Speed:' followed by an empty text input field.
- Type:** A dropdown menu with 'unknown' selected, followed by two small buttons: a square button and a button with three circles.

Creating and Linking to Submaps



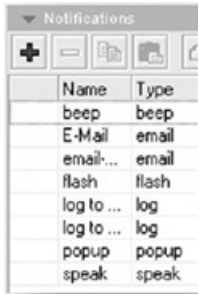
Remember that all of your maps are submaps, however when you have a single map, you will need to create a second map to link to. This is done very simply by starting to link to a submap, as there is a simple option to create the submap. To start, just like when you created devices and links, we will need to right click on the background of the current map. This will then give us the option to add a submap. The very first option is asking 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.



Once you click next, if you are creating a new Map, it will ask you for the name of the map, but 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



The screenshot shows a window titled "Notifications" with a dropdown arrow on the left. Below the title bar are five icons: a plus sign, a minus sign, a tower icon, a document icon, and a speech bubble icon. The main area of the window is a table with two columns: "Name" and "Type".

	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 that you configure notifications. What I mean, is where you can set what notifications go with what device or devices. Starting out, we have 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 only 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.

However, you can also create groups of notifications, so you can place items with names of your techs, say, Bob and Jim. And 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.



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 play. Now 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!

Reset

	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: 5

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

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 maybe 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 also is a valuable tool. These will allow you to resend the notification, if the device is still down, every so often, 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 may be reminded that the outage is still there.

Outages

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 when they see a list of outages that need to be addressed. You can also add notes to each of these as you wish as well. 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.

Outages

Remove Resolved

Status	Time	Duration	Device	Service
active	Aug/13 04:12:34	1d 04:50...	w_rodger...	ping
active	Aug/12 21:24:25	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...	HFFD	ping
active	Aug/12 10:42:08	1d 22:21...	2dHPFD	ping
active	Aug/10 18:38:00	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_l_howell	ping
active	Jul/10 18:25:06	34d 16:3...	cooring	ping
active	Jun/16 04:23:08	59d 04:4...	w_leick_l...	ping
active	Jun/10 15:20:05	64d 16:4...	pinoak	ping
active	Jun/10 15:19:58	64d 16:4...	bd_hpoi	ping
active	May/29 09:59:19	77d 00:0...	w_growel...	ping
active	May/21 05:13:13	85d 02:5...	squally?	ping
active	May/04 15:56:45	101d 16...	w_golf_li...	ping
active	Apr/11 14:21:02	124d 18...	n_curl_b...	ping
active	Apr/10 03:51:16	126d 05...	n_lndkie	ping
resolved	09:53:51	00:00:27	Ran Tre...	ping
resolved	09:20:49	00:00:27	Ran Tre...	ping
resolved	09:18:49	00:00:27	Ran Tre...	ping

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 probe that I will cover here is the TCP/UDP and SNMP probe. The other types, such as functions etc., are really outside the scope of this book. As these require programming logic, functions, and if-then-equals that simply are more complicated than what most people wish to accomplish.

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 connection-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 receive 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.

The image shows a configuration window for a network probe. At the top, there are three fields: 'Name' with the value 'Probe', 'Type' with a dropdown menu showing 'TCP', and 'Agent' with a dropdown menu showing 'default'. Below these is a description box that says 'General TCP probe, that can be used for various TCP protocol checking'. Underneath the description is a 'Port' field with the value '0'. There are two checkboxes: 'Connect Only' and 'First Receive, Then Send', both of which are currently unchecked. Below the checkboxes are six input fields arranged in three pairs, each labeled 'Send' and 'Receive' respectively. The 'Send' and 'Receive' labels are positioned to the left of their respective input fields.

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.

The screenshot shows a configuration window for a probe. The fields are as follows:

- Name: Probe
- Type: SNMP
- Agent: default
- Snmp Profile: default
- ☐ Treat service as available only if up
- OID: (empty)
- OID Type: integer
- Compare Method: == (equal)
- Integer Value: 0

A text box in the center of the window contains the following text: "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".

Tools

The tools pane allows you to add and control tools that you can access by right clicking the device. There is a number of built in tools, including WinBox, telnet, snmpwalk, etc, 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*. So 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
	<input type="button" value="▼ Insert Variable"/>
Command:	<code>mstsc /v:[Device.FirstAddress]</code>
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.

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, 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 use their username/password they created to login. Typically this is used in a hotspot environment allowing users to pay for internet access time and 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

The 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, one RAM, you need at least 32 Meg of RAM for User Manager and the second 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 Gig 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 Meg 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.

At the time of the book, user manager is still in v3 for production, many things have already changed to increase your options and customizability in v4beta. Since it is not production, I have not included its settings in this book.

Installation of User Manager

Installing User Manager 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.



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 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.



The screenshot displays the MikroTik RouterOS User Manager web interface. On the left is a sidebar menu with the following items: Status, Routers, Credits, Users, Sessions, Customers, Reports, Logs, and Logout. The main content area is divided into three sections. The top-left section, titled 'Search users', contains a text input field and a 'Search' button. Below this, it shows 'Active users: 0' with a 'Show' button. The bottom-left section shows 'Active sessions: 0' with a 'Show' button. The right section, titled 'Add users', contains several form fields: 'Number of users' (set to 1), 'Rate limits' (with a plus icon), 'Uptime limit' (set to 0s), 'Group', 'Download limit', 'Upload limit', and 'Transfer limit'. Below these is a 'Prepaid' section with a dropdown menu set to 'no credits available', and two checked checkboxes: 'Generate CSV file' and 'Generate vouchers'. At the bottom of this section is a 'Users per page' dropdown set to 1 and an 'Add' button.

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 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 explaining some of this, you will 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:

View report

Save

Customers are people who are selling services. Myself, 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 customers 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.

A screenshot of a web-based configuration form for a customer. The form contains the following fields and options:

- Login:** Text input field containing "admin".
- Password:** Text input field containing "*****".
- Parent:** Dropdown menu showing "admin".
- Permissions:** Dropdown menu showing "Owner".
- Public ID:** Text input field containing "wifi".
- Public Host:** Text input field containing "billing.wifimw.com".
- Private Information:** Section header with a minus icon.
- User Prefix:** Text input field.
- Signup Options:** Section header with a minus icon.
- Authorize.Net:** Section header with a minus icon.
- PayPal:** Section header with a minus icon.
- Date Format:** Dropdown menu showing "%b/%c/%Y". Below it, text reads "(%Y - year, %b - month, %d - day)".
- Currency:** Text input field containing "USD".
- Time Zone:** Dropdown menu showing "-5:00".
- Voucher Template:** Section header with a minus icon.
- Save:** A button at the bottom right of the form.

Subscribers are customers. The difference is they are the “top” customer. They have their own authorize.net, account, their own routers, 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 is not the same, there is no m constant for month in User Manager. The image to the right shows the constants that User Manger uses.

Name: 1 Day

Time: 1d

Full Price: ☒ available
9.99

Extended Price: ☒ available
7.99

Save

■ w - week (equals 7 days)

■ d - day (equals 24 hours)

■ h - hour (equals 60 minutes)

■ m - minute (equals 60 seconds)

■ s - second

Name: 123

IP Address: 172.25.0.1

Shared Secret:

Log events: ☐ Authorisation ok
☐ Authorisation failed
☐ Accounting ok
☐ Accounting failed

Save

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.

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 at all of the settings with User Manager, 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 Authroize.net, a SSL installed.

That's it, there are not many requirements, but they are requirements to make the system work. First off, could we not run with 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 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 cert 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 our payment gateway information, sign-up information as well as our 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 customers installed. 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 wifi 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 of a valid URL on the internet. The User Prefix option is if you wished to have multiple subscribers on the system,

each user would get a prefix to identify what subscriber they belong too.

Next we will fill out the Private Information section. These fields are not required, but you can fill them out if you wish. The sign-up Options is next.

Signup Options: ☰

Signup Allowed: ☒

Signup Email Subject:

Signup Email Body:

The Sign-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 Sign-Up e-mail as necessary.

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 otherwise you cannot expand the authorize.net section. If you are, 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.



Authorize.Net: ☹

Allow Payments: ☒

Change Login ID:

Change Transaction Key:

Change MD5 Value:

Title:

Return URL:

Use Test Gateway: ☐

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, what is the PayPal payment address, or e-mail. 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):

Secure response: ☒

Accept pending: ☒

Return URL:

In regards to what system do I prefer, I think the aauthorize.net system is more business oriented, and 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 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☒

Add

Configuration of Credits

Name:	<input type="text"/>
Time:	<input type="text" value="0s"/>
Full Price:	<input type="checkbox"/> available
Extended Price:	<input type="checkbox"/> available
<input type="button" value="Add"/>	

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. So if you wish you could give existing users a discount.

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.



MikroTik
RouterOS User Manager

email

login admin

password

confirm password

Prepaid time

time choose one ▼

pay with ☒ Credit Card

Please remember this data as it will be required to log in later:

login: **idd**

password: *********

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. In this process it is not sending data to authorize.net and delivering 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.

User Sign-In Page

The users also have a page that they can sign in and update their account, and add more time. This page is <http://ipaddress/user>.


RouterOS User Manager

Status

Payments

Settings

Logout

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

<input type="checkbox"/>	ID	From Time	Till Time	Uptime	Download	Upload
<input type="checkbox"/>	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. Upon them logging out the radius system should receive accounting information updates, showing how much time they used, as well as data transfer information

<input type="checkbox"/>	Username	Prepaid	Uptime	Time left	Price (USD)	Download	Upload
<input type="checkbox"/>	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.

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.

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)

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 ▾

Add

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 they 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 user manager system, and you can print these out and give these username/passwords out as you wish.

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, 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, the 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=dstnat 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=dstnat 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.

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 out from the failover system.
- Link one, the one we wish to prefer, will be plugged into ether 1 on both RouterOS units on each end
- Link two, will be plugged into ether2 on both ends.
- The cable going to the rest of the network will be on ether3 on each RouterOS.
- Setup Bridges on both ends, with STP or RSTP.
- Setup Ether1, 2 and 3 as bridge ports
- Increase the Priority for Ether2, on both sides to 90+, or another higher number than the default that is created on the other ports.
- Once this is setup, your ether1 should be your designated port, and ether2 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 prerouting 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 o 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 srcnat 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.

¹ RouterOS contains many features, which make it have almost endless configurations.

² DOM or Disk on Module is a Flash disk that plugs into either a SATA or IDE port.

³ Extended Frequency licenses override the country frequencies that are listed in the basic RouterOS configuration. This allows you to operate in a band or on a wireless frequency that is typically not allowed. Be sure to check local laws for regulations in your area.

Appendix

Features Only Available via Command Line Interface

- Export Command – Used to create text configuration export. Processing an under a section of RouterOS will process only that section and sections under that section. 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.

[Index](#)

Access List, 57
Access Lists, 227
Accounting 294
Active Users, 73
Address Lists, 129
Address Resolution Protocol, 109
API, 76
Area Prefixes, 231
ARP, 109
ARP List, 110
Auto Upgrades, 302
Backup / Restore, 79
Bandwidth Test Client, 304
Bandwidth Test Server, 303
BGP, 322
Bindings, 275
Bonding, 177
Bridge, 170
Bridge Ports, 171
Bridged Access Point Configuration, 104
Bridged Client, 105
Brute Force Attacks, 145
Bursting 260
Chains, 125

- Change MSS, 153
- Change TOS Bit, 153
- Checking Gateways, 114
- Choosing a Tunnel Type, 212
- Clock, 297
- Command Line Interface, 377, 386
- Connection Bytes, 132
- Connection Limiting, 135
- Connection Lists, 230
- Connection Marks, 152
- Connection States, 122
- Controlling P2P, 269
- CPE – Client Premise Equipment Configuration, 105
- Default Routes, 86
- Default User and Password, 44
- DHCP Relaying, 295
- DHCP Server, 58
- DHCP-Client, 88
- DHCP-Server, 90
- DNS Caching, 87
- Double Queuing, 264
- DST Limit, 138
- Dude, 328
- Dude Agents, 330
- Dude Devices, 340
- Dynamic Routing, 318
- ECMP, 115
- E-Mail System, 305
- Ensuring Bandwidth Allocations, 262

- EoIP, 185
- EoIP Tunnel, 187
- Ethernet, 164
- Extended Frequency, 29
- Fetch, 305
- FIFO Queues, 252
- Firewall Actions, 139
- Forward Chain, 125
- Free Hotspots, 273
- FTP, 76
- Graphing, 306
- Hard Disk Installation, 40
- Hierarchical Token Bucket, 248
- Hotspot Interface, 275
- Hotspot Login Methods, 279
- Hotspots, 272
- Hotspots with Radius, 282
- Learn RouterOS by Dennis Burgess
- HTB, 248
- Importing Scripts, 81
- Inbound NAT, 157
- Ingress Priority, 137
- Input Chain, 125
- IP Bindings, 284
- IP Pools, 96
- IP Scan, 310
- IPIP, 188
- IPsec, 208
- Jumping to Chains, 126

- Large Transfer Queues, 264
- Layer 7
- Filters, 134
- Licensing, 27
- Limit, 138
- Links, 355
- Logging, 82
- Logging Rules, 83
- Lose your RouterOS License, 41
- M3P, 296
- Mangle, 150
- Masquerading, 98, 156
- MESH, 180
- MetaRouters, 316
- methods of accessing a RouterOS System?, 44
- MikroTik Packet Packing Protocol, 296
- Multi-Link, 204
- Multiple Radius Servers, 291
- NAT, 98
- Neighborhood Viewer, 44
- Neighbors, 295
- Net booting, 34
- NetInstall, 40
- Network Address Translation, 155
- Network Maps, 350
- Non-Paying Customer, 382
- Notifications, 356
- N-Stream, 233
- N-Stream Dual, 234

- Nth, 138
- NTP, 298
- One-to-One NAT, 159
- OpenVPN, 205
 - OpenVPN Client, 206
 - OpenVPN Server, 205
- OSPF, 320
- Other Chains, 126
- Outages, 359
- Outbound NAT, 158
- Output Chain, 125
- Packet Flow, 124
- Packet Marks, 151
- Packet Sniffer, 307
- Paid Hotspots, 273
- Parents, 261
- Peer to Peer Filtering, 133
- Per Connection Load Balancing, 383
- POD Attacks, 146
- Policy Based Routing, 116
- Pools, 296
- Port Scan Detection, 136
- PPP, 189
- PPPoE, 201
 - PPPoE Client, 57
- Private VPN, 384
- Probes, 359
- Protecting Networks, 143
- Protecting Your Router, 142

- Pseudobridge Mode, 106
- Queue Trees, 258
- Queue Types, 251
- Quick Reference Guide, 380
- Radius Client, 290
- Radius RouterOS Users, 74
- Random, 137
- RED Queues, 253
- Redirect, 162, 382
- Registration Table, 53, 229
- Reset Configuration, 300
- Returning from Chains, 128
- RIP, 318
- Routed / NAT CPE, 106
- RouterBoard Devices, 20
- RouterOS User Groups, 72
- Routing 54
- Routing Filters, 325
- Routing Marks, 151
- Routing Policies, 117
- Scripting 301
- Secure Shell Access, 47
- Security Profiles, 224
- Selective Port Forwarding 161
- Server Profiles, 278
- Setup Transparent Web Proxy System, 382
- SFQ Queues, 254
- Simple Queues, 56, 259
- Socks, 297

- Solar Power, 23
- SPAM Prevention, 144
- Splash Page, 274
- SSH, 47, 76
- SSH Keys, 73
- Static Routing 111
- Store, 315
- Switch Controls, 165
- System Identity, 299
- System Options, 54
- Tarpit, 141
- Telnet, 46
- TFTP Server, 308
- The Dude, 328
- The PowerRouter 732, 24
- Time, 139
- Tools, 361
- Traffic Control, 246
- Traffic Identification, 121
- Traffic-Flow, 308
- Transparent Web Caching, 313
- Transparent Web Proxy, 382
- Troubleshooting Wireless Links, 243
- Tunnels, 185
- UPnP, 309
- User Defaults, 70
- User Management, 70
- User Manager, 364
- User Sign-In Page, 374

- User Sign-Ups, 373
- Using Distances, 115
- Using Marks, 151
- Using NetInstall, 31
- Using Trail Users, 280
- Using WinBox, 60
- Virtual Access Points, 232
- Virtual Ethernet, 168
- Virtual Ethernet Interfaces, 168
- Virtual LAN, 175
- VLANs, 175
- Vouchers, 374
- VRRP, 182
- Walled Garden, 284
- Walled-Garden, 275
- Watchdog, 303
- WDS, 235
- Web Proxy, 311
- WebBox, 49
- WIC – Wireless Interface Cards, 214
- WinBox Menus, 64
- Wireless Distribution System, 235
- Wireless Interfaces, 52
- Wireless Link Optimization, 239
- Wireless Operational Modes, 222
- Wireless Tools, 218
- WWW, 76
- X86 Based RouterOS Systems, 24
- x86 Hardware, 26