

N9LYA TECHNICAL GUIDES

MikroTik RB5009

One Router, Covered Completely

Jerome M. Kutche

N9LYA

Second Edition · September 2026

About This Book

The RB5009 is the best router most people will ever own, and the majority of them are running it at a fraction of what it can do.

That is not a criticism of the owners. It is a consequence of how the device works. RouterOS gives you complete control over the forwarding path, which means it also gives you complete freedom to build a configuration that quietly routes every packet through the CPU when the switch chip could have handled it in silicon. The router will work. It will look fine. And it will be doing ten times the work it needs to.

This book is about closing that gap.

What This Book Is

A complete treatment of one specific device: what it is made of, how packets actually move through it, how to configure it correctly from an empty state, and how to tune it until the hardware is the limit rather than your configuration.

It is opinionated. Where there are three ways to do something and one of them is right for this hardware, I say so and explain why. Where MikroTik's defaults are wrong for a competent operator, I say that too.

It is also specific. Every command in this book is a command you can paste. Every claim about performance is tied to something you can measure on your own device with tools RouterOS already includes.

What This Book Is Not

It is not a RouterOS reference manual. MikroTik's documentation is comprehensive and free and I am not going to reproduce it. What the documentation does not do well is explain why — why hardware offload silently disengages, why your firewall rule order costs you throughput, why the 10 Gbps SFP+ port does not give you 10 Gbps of routed traffic. That is the gap this book fills.

It is not a CCNA course. I assume you know what an IP address is, what a VLAN does conceptually, and roughly what a firewall is for. If you are missing those, get them first; this book will make far more sense afterward.

Who This Is For

The homelab operator who bought an RB5009 because the forums said to and now has a working configuration they do not entirely understand.

The small network administrator running one or several of these at sites and wanting a defensible, repeatable, documented baseline.

The engineer coming from Cisco, Juniper, or pfSense who needs to map existing knowledge onto RouterOS's genuinely different model, and who is tired of tutorials that show the clicks without the concepts.

The person who already runs one well and wants the last twenty percent: the offload verification, the queue tuning, the profiling method, the parts nobody writes about.

How to Use This Book

Part I explains the machine. Do not skip it. Nearly every optimization decision later in the book traces back to the architecture described in Chapter 2, and if you understand that chapter properly, half the rest becomes obvious.

Parts II through V build a configuration from nothing, in order. If you follow them sequentially you will end with a documented, segmented, hardened, offloaded, monitored router and a clear understanding of every line in it.

Appendix A is the complete resulting configuration as a paste-ready script. It is there for reference and for rebuilds, not as a substitute for understanding what it does.

Each chapter ends with a **Bench Check** — verification commands and short exercises. These are not filler. Several of them will reveal that something you assumed was working is not, which is the entire point.

Conventions

Commands are shown in RouterOS CLI form, because the CLI is unambiguous, scriptable, and portable across every interface. Everything here can be done in WinBox or WebFig; the menu paths mirror the command paths directly.

Where a command produces output worth examining, I show it. Where a value is site-specific I use an obvious placeholder.

Configuration examples use this address plan throughout:

Segment	VLAN	Subnet
Management	10	10.10.10.0/24
Servers	20	10.10.20.0/24
Trusted	30	10.10.30.0/24
IoT	40	10.10.40.0/24
Guest	50	10.10.50.0/24

Substitute your own. The structure is what matters.

A Word on Version

RouterOS moves. Stable, long-term, testing, and development branch versions change over time, and the best choice depends on the features and risk profile of the actual installation. Chapter 23 covers branch selection properly. Check MikroTik's current download page and release notes immediately before an upgrade rather than relying on a version number printed here.

Nearly everything in this book has been stable across the 7.x line and will remain so. Where a feature is version-sensitive I flag it. Where MikroTik has changed behavior between versions — and they do — I note the change. Verify against the current documentation before you deploy anything to a network people depend on. That is good practice regardless of what any book tells you.

About the Author

Jerome M. Kutche holds the amateur radio callsign N9LYA, first licensed in 1991. He spent more than three decades in industrial electrical and controls engineering — Allen-Bradley and Siemens PLC systems, motor control, plant electrical supervision — across manufacturing and heavy industry, holding NFPA 70E, lockout/tagout, and thirty-nine other professional certifications.

He serves as Indiana's 44Net/AMPRNet coordinator, administering the state's allocation of amateur radio internet address space, and is trustee for two amateur radio clubs. He operates a packet radio node, a monitored fifty-panel solar array, and a homelab whose edge is an RB5009 that has been configured, misconfigured, benchmarked, broken, and rebuilt more times than was strictly necessary.

The habits in this book come from thirty years of industrial work, where a change you cannot explain is a change you do not make, and where the person who understands the system is the person who documented it.

Part I — Understanding the Machine

Chapter 1: The RB5009 in Detail

Before configuring anything, know what you have. The RB5009’s specifications explain most of its behavior, and several of them are non-obvious in ways that matter enormously later.

The Variants

Two models share the RB5009 designation and the same core hardware.

RB5009UG+S+IN — the standard model. Seven gigabit Ethernet ports, one 2.5 gigabit Ethernet port, one 10G SFP+ cage, one USB 3.0 type A port. Powered by DC jack, PoE-in on ether1, or a 2-pin terminal block on the side.

RB5009UPr+S+IN — the PoE model. Same ports, same CPU, same memory, plus **PoE-out on all eight Ethernet ports**, 802.3af/at compliant, with a total output budget of 130 W and a 48 V 96 W supply included. It also carries PoE-in on all eight ports, giving it an unusually large number of powering options.

If you are running access points, cameras, or phones off the router directly, the UPr is worth the difference. If you are not, the UG is the same router for less money.

Be careful with third-party comparison articles on this device. There is a meaningful amount of inaccurate material online claiming, among other things, that the UG lacks 2.5 GbE or that the UPr lacks USB. Both claims are wrong. MikroTik’s own product pages are authoritative; use them.

The Core Specification

Component	Specification
CPU	Marvell Armada, quad-core ARMv8 (64-bit), 1.4 GHz
Switch chip	Marvell Amethyst family
CPU ↔ switch link	10 Gbps full duplex
RAM	1 GB DDR4
Storage	1 GB NAND
Ethernet	7 × 1 GbE, 1 × 2.5 GbE (ether1)
Fiber	1 × SFP+ (10 Gbps)
USB	1 × USB 3.0 type A
License	RouterOS level 5
Cooling	Passive — the case is the heatsink

Four numbers on that list determine most of what follows.

Quad-core, 1.4 GHz, ARM64. Substantial for a router this size, and the reason the RB5009 outperforms similarly priced devices under CPU-heavy workloads like IPsec, deep queueing, and complex firewalling. But it is still four modest ARM cores, and RouterOS does not parallelize every workload across them evenly. A single-flow workload frequently lands on one core.

1 GB of RAM. Generous for routing, finite for everything else. Connection tracking, large routing tables, address lists with hundreds of thousands of entries, and packet capture all consume it. This is not a machine on which to be careless with a million-entry blocklist.

1 GB of NAND. Enough for RouterOS plus packages plus configuration backups. Not enough to treat as storage. Logging to disk aggressively will fill it, and a full NAND on a router is an unpleasant place to be. Log to a remote syslog server.

10 Gbps full duplex between switch chip and CPU. This is the single most consequential number in the entire device, and Chapter 2 is devoted to it.

The Port Layout

All nine ports — the seven gigabit copper ports, the 2.5 gigabit copper port, and the SFP+ cage — connect to the Marvell Amethyst switch chip. The switch chip connects to the CPU over a single 10 Gbps full-duplex line.

There is no port on this device that bypasses the switch chip and attaches directly to the CPU. Every packet that enters the RB5009 arrives at the switch chip first.

ether1 is the 2.5 gigabit port and is also the PoE-in port on the UG model. MikroTik's default configuration treats it as the WAN port. You are not obliged to follow that convention, but there is a reason to: if your internet connection exceeds a gigabit, ether1 is the only copper port that can carry it.

ether2 through ether8 are standard gigabit ports.

sfp-sfpplus1 is the fiber cage and it runs at 10 Gbps. It does not negotiate down to 1G or 2.5G in the way people frequently expect — plan on a 10G module or a 10G DAC. If you need a gigabit fiber uplink, verify module compatibility before ordering, because this trips up a great many people.

Thermal Behavior

The CPU and switch chip are mounted on the underside of the board, and the aluminum case is the heatsink. There is no fan. This is an excellent design — silent, no moving parts to fail — and it has one consequence you must respect:

The case gets hot, and that is correct. A warm RB5009 is a working RB5009. What matters is not the case temperature you feel but the die temperature the device reports.

`/system health print`

On this platform you get CPU temperature, board temperature, and on most units voltage and current draw as well. Check it under load. Typical operating temperatures sit comfortably in the 40–55 °C range depending on ambient and load.

Give it air. Do not stack anything on top of it. Do not put it in a sealed enclosure. Do not mount it directly against another heat-producing device. In a rack it is common to mount four of these in a single 1U space using MikroTik's mounting kit — which works, but leave a blank U above if you can.

Because voltage and current are monitored, you can graph power draw over time, which is a genuinely useful diagnostic. A device drawing more than usual is doing more work than usual.

Power

Three input methods on the UG: DC jack, PoE-in on ether1, and a 2-pin terminal block. The UPr adds PoE-in on any of the eight Ethernet ports.

Typical draw for a UG under normal routing load is modest — in the range of ten to fifteen watts, well under the maximum rating. This is one of the RB5009's underappreciated strengths: it delivers genuine multi-gigabit routing at a power budget that costs perhaps twenty dollars a year to run continuously.

The multiple input options make redundant power straightforward. A DC supply on the jack and a PoE injector on ether1 gives you two independent feeds, and the device will ride through the loss of either. For anything you care about, do this.

The License

The RB5009 ships with a **RouterOS level 5** license, which is generous. Level 5 removes essentially every limit a homelab or small business will encounter: unlimited tunnels, unlimited hotspot users beyond any realistic need, and full access to the routing protocol suite.

You do not need to buy anything. The license is tied to the hardware and survives upgrades and configuration resets.

What the RB5009 Is Good At

Routing at multi-gigabit speeds with real firewall rules. This is the core competency and it is excellent at it.

Complex configurations. VLANs, policy routing, multiple WANs, VPN concentration, BGP. The CPU has enough headroom that you can build something genuinely sophisticated without hitting a wall.

Encryption, within reason. ARMv8 with crypto extensions handles IPsec and WireGuard meaningfully faster than the older MIPS-based MikroTik devices. Expect useful throughput, not line rate.

Silent, low-power, always-on operation. Passive cooling and a small power budget make it appropriate for living spaces.

What It Is Not

It is not a 10 Gbps switch. It can switch at high rates in hardware, but the port layout — one SFP+ and eight copper — means it is not the right device for aggregating multiple 10G links. Buy a CRS for that.

It is not a deep packet inspection appliance. RouterOS has no application-layer inspection engine in the sense that a commercial NGFW does. It filters on addresses, ports, protocols, connection state, and some layer-7 regex matching that is expensive and limited.

It is not appliance-simple. RouterOS gives you enough rope to build something excellent or something broken. That is the trade, and it is why this book exists.

Bench Check

1. Identify your exact model from the label and from `/system routerboard print`.
2. Run `/system resource print` and note the CPU, memory, and free storage figures.
3. Run `/system health print` under load and record the temperatures.
4. Confirm your RouterOS version and which release branch it came from.
5. Verify your power arrangement. If the device matters, add a second independent feed.

Chapter 2: The Architecture That Determines Everything

This is the most important chapter in the book. Every performance decision later traces back to what is described here.

The Two Forwarding Paths

A packet arriving at the RB5009 can take one of two fundamentally different journeys.

Path one: the switch chip. The packet arrives at a port, the Marvell Amethyst switch chip examines its destination MAC address, finds the egress port in its hardware forwarding table, and sends it out. The CPU is never involved. This happens in silicon at wire speed across all ports simultaneously, and it costs essentially nothing.

Path two: the CPU. The packet arrives at a port and the switch chip determines it must go to the CPU — because it needs routing, because a firewall rule must evaluate it, because it is destined for the router itself, or because the configuration has disabled hardware forwarding for that traffic. The packet crosses the 10 Gbps link to the CPU, RouterOS processes it through the packet flow described in Chapter 10, and it crosses the link back to the switch chip for transmission.

Path one is free. Path two is where all your capacity goes.

The entire discipline of optimizing an RB5009 is deciding, deliberately, which traffic takes which path.

RB5009 Architecture

Performance depends on which traffic reaches the CPU

PORTS

Copper / SFP+
Physical ingress

SWITCH CHIP

Layer 2 forwarding
Hardware offload

CPU

Routing / firewall
Services

STORAGE / USB

Packages / logs
External devices

N9LYA TECHNICAL GUIDES

Figure 1: RB5009 traffic architecture

The 10 Gbps Link

The switch chip and the CPU are joined by a single 10 Gbps full-duplex connection. This is the aggregate capacity for everything the CPU handles.

Full duplex means 10 Gbps in each direction simultaneously, so a routed packet consuming the switch-to-CPU direction on ingress and the CPU-to-switch direction on egress does not double-count against a shared budget. The practical ceiling on CPU-forwarded traffic is therefore in the neighborhood of 10 Gbps — assuming the CPU itself can keep up, which is a separate and usually tighter constraint.

Two consequences worth internalizing:

Your SFP+ port cannot deliver 10 Gbps of routed traffic through the CPU while anything else is happening. The link is shared by every packet the CPU touches, from every port.

Switched traffic between ports on the same bridge, with hardware offload active, does not consume this link at all. Two servers on ether3 and ether4 exchanging data at full gigabit speed impose zero load on the CPU and zero load on the CPU link. You can verify this: the CPU graph will not move.

Hardware Offload

Hardware offload — often shortened to “hw-offload” — means the switch chip is doing the forwarding for a given bridge port instead of the CPU.

This is not a global setting you turn on. It is a state that each bridge port either achieves or does not, determined by whether your configuration uses features the switch chip can implement. Add one incompatible feature and RouterOS silently moves that traffic to the CPU. The router keeps working. Your throughput quietly drops and your CPU usage quietly rises.

Verifying offload is therefore a routine operational task, not a one-time check.

```
/interface bridge port print
```

Look for the **H** flag on each port. H means hardware offload is active on that port. No H means that port’s traffic is being forwarded by the CPU.

Flags: I - INACTIVE; H - HW-OFFLOAD

#	INTERFACE	BRIDGE	HW	PVID
0	H ether2	bridge	yes	30
1	H ether3	bridge	yes	20
2	H ether4	bridge	yes	20
3	ether5	bridge	yes	40

In that output, ether5 has lost offload and is being handled in software. That is a fault to investigate, not a cosmetic detail.

Note the distinction between the HW column and the H flag. The column shows what you requested. The flag shows what you got. They disagree more often than people expect, and only the flag matters.

What Breaks Offload

The list changes between RouterOS versions, and the authoritative answer is always the H flag on your device running your version. But the recurring causes:

use-ip-firewall=yes on bridge settings. This forces bridged traffic through the IP firewall, which requires the CPU. It is occasionally necessary and it is expensive. Leave it off unless you have a specific requirement you can articulate.

Certain STP modes and configurations, depending on version and hardware.

Port mirroring and sniffing. Capturing traffic requires the CPU to see it.

Bridge features the chip does not implement, which varies by chip family and by RouterOS release.

Configurations that RouterOS simply does not offload yet. MikroTik expands hardware support over time; a feature that broke offload two years ago may work today.

The diagnostic method is simple and reliable: get offload working, then add one feature at a time, checking the H flag after each. When it disappears, you have found the cause and you can decide whether the feature is worth the cost.

Where Routing Fits

Hardware offload applies to **bridged** traffic — layer 2 forwarding within a bridge. **Routed** traffic — moving between subnets — inherently requires the CPU, because routing is a layer 3 decision the switch chip does not make in this configuration.

This produces the central design tension of the RB5009:

Traffic within a VLAN is free. Traffic between VLANs costs CPU.

This is not a reason to avoid segmentation. Segmentation is the most valuable security work available and Chapter 7 argues for it strongly. It is a reason to design your segmentation so that high-volume flows stay within a segment where you can arrange it, and to understand that inter-VLAN traffic is CPU traffic.

It is also the reason FastTrack exists.

FastTrack

FastTrack is a RouterOS mechanism that short-circuits the CPU forwarding path. When a connection is fast-tracked, subsequent packets in that connection skip most of the packet flow — most firewall processing, most queueing, most mangle — and take a dramatically shorter path through the CPU.

The throughput difference is large. On this hardware it is frequently the difference between saturating a gigabit WAN with CPU headroom to spare and running the CPU hard to achieve the same result.

```
/ip firewall filter
add chain=forward action=fasttrack-connection hw-offload=yes \
    connection-state=established,related \
    comment="FastTrack established and related"
```

The `hw-offload=yes` parameter engages hardware assistance where the platform supports it.

What FastTrack costs you. Fast-tracked packets bypass most of what comes after. That means:

- Simple queues and queue trees generally do not see fast-tracked traffic
- Mangle rules after the FastTrack rule do not see it
- Traffic accounting and per-connection visibility are reduced
- Some firewall rules will not evaluate against it

This is the fundamental trade in RouterOS performance tuning: **FastTrack and queueing are largely mutually exclusive.** You can have maximum throughput or you can have fine-grained traffic control. Chapters 15 and 16 cover how to get most of both by fast-tracking selectively rather than universally.

The important thing to understand now is that this trade exists and that it is a decision, not a default.

The Three Tiers of Traffic

Putting it together, traffic on an RB5009 falls into three performance tiers:

Tier 1 — Hardware switched. Same bridge, same VLAN, offload active. Wire speed, zero CPU, does not touch the 10 Gbps link. This is free.

Tier 2 — FastTracked. Routed, but on an established connection that bypasses most processing. Very fast, modest CPU, consumes CPU link bandwidth.

Tier 3 — Fully processed. Routed through the complete packet flow: connection tracking, all firewall chains, mangle, queueing, NAT. This is where your CPU budget goes.

An optimized RB5009 configuration is one where each class of traffic is in the tier it belongs in, deliberately, and where you know why.

A Mental Model

When you look at a configuration change and wonder whether it will cost you performance, ask three questions in order:

1. **Does this move traffic from the switch chip to the CPU?** If yes, that is the largest possible cost.
2. **Does this prevent traffic from being fast-tracked?** If yes, that is the second largest.
3. **Does this add per-packet work on the CPU path?** If yes, that is a real but usually smaller cost.

Most configuration decisions that hurt performance do so through question one or two. Optimizing the third while ignoring the first two is a common and unproductive way to spend an evening.

Bench Check

1. Run `/interface bridge port print` and confirm the H flag on every bridge port. Investigate any that lack it.
2. Copy a large file between two devices on the same VLAN while watching `/system resource cpu print`. Note that CPU barely moves.
3. Repeat the transfer between devices on different VLANs. Note the difference.
4. Check whether `use-ip-firewall` is enabled: `/interface bridge settings print`.
5. Classify your three highest-volume traffic flows into the three tiers above.

Chapter 3: First Contact — Bootstrap and Baseline

A new or reset RB5009 needs a deliberate first configuration. The defaults MikroTik ships are reasonable for getting online quickly and inadequate for anything you intend to keep.

This chapter takes a device from factory state to a secure, documented baseline.

Before You Connect Anything

Do not connect the WAN before the router is configured. The default configuration includes a firewall, but a router still using its factory credentials is a risk you do not need to take. Configure first, connect after.

Have a recovery path. For an RB5009 that means knowing where Netinstall is and having the ability to reach ether1 with a direct cable. Chapter 23 covers Netinstall in full. Know it exists before you need it.

Reaching the Device

A factory RB5009 normally has a default configuration with a bridge across the LAN ports, a DHCP server, and 192.168.88.1 as the router address. The username is commonly admin. Depending on production date and provisioning, the password may be blank or may be a unique factory password printed on the device, packaging, or quick guide. Check the device label and current MikroTik instructions instead of assuming a blank password.

Connect a laptop to ether2 through ether8, accept a DHCP lease, and reach the router.

WinBox is MikroTik's management application and can connect by MAC address, which means it works even when IP configuration is broken or absent. This makes it the recovery tool of choice. Download it from MikroTik directly and nowhere else.

Safe Router Bootstrap

Establish management and recovery before adding complexity



N9IYA TECHNICAL GUIDES

Figure 2: Safe router bootstrap

SSH and the terminal are what this book uses. Everything shown here works identically in WinBox's terminal window.

WebFig is the browser interface. Functional, and I would not leave HTTP management enabled.

Deciding: Default Configuration or Clean Slate

Two philosophies.

Keep the default and modify it. Faster to a working state. The default configuration is reasonable and includes a serviceable firewall. The disadvantage is that you inherit choices you did not make and may not notice.

Start from nothing and build deliberately. Slower, and you end up understanding every line.

For a device you intend to run seriously, start clean. It takes an extra hour once and pays back permanently.

```
/system reset-configuration no-defaults=yes skip-backup=yes
```

The router reboots with no configuration at all — no addresses, no bridge, no DHCP, no firewall. Reach it with WinBox over MAC, or connect a cable and set an address manually.

Understand what you have just done. A configuration-free RouterOS device has no firewall whatsoever. Do not connect it to the internet in this state.

Safe Mode

Before you touch anything, learn this. It will save you a drive to the site.

In a terminal, press **Ctrl+X**. The prompt changes to show [Safe Mode]. In WinBox there is a Safe Mode button.

While in safe mode, every configuration change is provisional. If your session disconnects — because you just broke your own management path — the router automatically reverts every change made during that session.

Press Ctrl+X again to commit and exit safe mode.

Use safe mode for every remote change to firewall rules, interface configuration, addressing, or bridge settings. These are the changes that lock you out, and safe mode is a complete answer to that risk. It costs one keystroke.

The Bootstrap Sequence

Working through in order. Each step is explained in more depth in a later chapter; this is the minimum to reach a safe, reachable baseline.

Identity. Name the device. When you have several, unnamed routers become genuinely confusing.

```
/system identity set name=rtr-edge-01
```

Users. Create a real account, verify it works, then disable the default.

```
/user add name=jkutche group=full password="<strong-password>" \  
comment="primary admin"
```

Log out. Log back in as the new account. Confirm full access. Only then:

```
/user set admin disabled=yes
```

Disabling rather than deleting keeps a recovery option if something goes wrong with the new account.

Services. RouterOS enables several management services by default. Disable everything you will not use.

End of the free sample

MikroTik RB5009 — Second Edition · by Jerome M. Kutche

That was the first 12 of 83 pages. The full book continues with the practical detail, worked examples and recovery steps that don't fit in a sample.

Get the full book (\$7.99, PDF + EPUB):

<https://booksrus.biz/product/mikrotik-rb5009/>

Also on Gumroad:

<https://n9lya4.gumroad.com/l/gnefat>



All 17 Second Edition guides in one bundle for \$89.99:

booksrus.biz/product/n9lya-technical-guides-complete-bundle

This sample may be shared freely in unmodified form. © Jerome M. Kutche. Second Edition.