

N9LYA TECHNICAL GUIDES

Networking

From Beginner to Expert

Jerome M. Kutche

N9LYA

Second Edition · September 2026

About This Book

Networking is taught badly, and it's taught badly for a specific reason: most networking books are written to a certification exam.

Exams need material that can be tested unambiguously, so the books are full of things that can be tested unambiguously — the seven layers of the OSI model, the number of bits in a MAC address, the administrative distance of OSPF. All true, all testable, and almost none of it is what you need at 11 p.m. when something is down and people are waiting.

What you need then is different. You need a mental model of how packets actually move, a systematic way to bisect a problem, and enough understanding of each layer to know which one you're standing on.

This book is written for that.

The Approach

Build first, theorize second. Every concept is tied to something you can construct and verify. There is no value in knowing what a broadcast domain is until you've made two of them and watched traffic fail to cross between them.

Troubleshooting gets its own chapters, plural, because it's the actual job. Chapter 19 covers the methodology and Chapter 20 covers packet capture, which is the instrument that ends arguments.

The physical layer is treated seriously. A surprising share of “network problems” are cable problems, and the people who diagnose them fastest are the ones who understand what's happening on the wire.

Two chapters cover ground most networking books skip entirely. Chapter 22 covers industrial and operational technology networking — deterministic Ethernet, ring redundancy, PROFINET and Modbus, and why a network engineer's instincts can shut down a production line. Chapter 23 covers amateur radio networking, packet radio, and AMPRNet, which is a real IP network on real allocated address space that almost nobody outside the hobby knows exists.

I include both because I've worked in both. Thirty years in industrial controls taught me that the assumptions of enterprise networking do not survive contact with a plant floor. And administering Indiana's allocation within the amateur radio address space taught me that there's an entire parallel internet that predates most of what people think of as the internet.

Who This Is For

Beginners. Part I assumes nothing beyond having used a computer. Read it in order.

Homelabbers and self-hosters. You've got a router, some VLANs, and a growing pile of services. Parts III and IV are where you'll live, and Chapter 15 will probably change how you've laid things out.

IT generalists. You administer systems and inherit the network. Start at Part II, read Part V carefully.

Industrial and controls people. Chapter 22 is written for you specifically, and Parts I and II will fill in the IT vocabulary that plant networks increasingly require.

Amateur radio operators. Chapter 23 covers packet, AMPRNet, and mesh networking, and the rest of the book covers the IP fundamentals that modern digital modes assume you have.

Conventions

Commands appear in the form you'd type them. Where a concept applies across platforms, I show Linux first because it's universal, and note the equivalents.

Each chapter ends with a **Wire Check** — verification tasks on a network you own or control. Do them. Networking rewards hands-on verification more than almost any technical subject, because the failure modes are rarely what you predicted.

Address examples use documentation ranges where possible, and 10.x and 192.168.x for lab examples, so nothing here will collide with a real network.

About the Author

Jerome M. Kutche holds the amateur radio callsign N9LYA, first licensed in 1991. He serves as Indiana's 44Net/AMPRNet coordinator, administering the state's allocation within the amateur radio IPv4 address space, and is trustee for two amateur radio clubs. He operates packet radio nodes carrying store-and-forward traffic and has spent a considerable number of hours diagnosing why a link that worked yesterday doesn't work today.

He spent over three decades in industrial electrical and controls work across manufacturing and heavy industry, including Allen-Bradley and Siemens PLC systems, plant electrical maintenance, and supervision, holding NFPA 70E and thirty-nine other professional certifications. Before that, from 1990 to 1996, he built RF test and measurement equipment for the telecommunications and cable television industries.

His homelab runs on segmented VLANs behind a MikroTik router, with reverse proxying, filtering DNS, site-to-site VPN, and rather more monitoring than is strictly necessary. Most of the practical advice in this book comes from having broken it first.

Part I — Foundations

Chapter 1: What a Network Actually Is

Strip away the vocabulary and a network does one thing: it moves a chunk of data from one device to another. Everything else is detail about how that gets arranged.

The Core Problem

Two computers need to exchange information. Solving that requires answering several questions, and each answer became a layer of the technology:

How do we physically move bits? Voltage on copper, light in fiber, modulated radio. That's the physical layer.

How do multiple devices share one medium without talking over each other, and how does a device know a message is for it? Addressing and access control on a local segment. That's Ethernet.

How do we reach a device that isn't on our local segment? Logical addressing and routing. That's IP.

How do we ensure the data arrives complete and in order — or deliberately not bother? That's TCP and UDP.

How do we identify which application on the machine wants it? Port numbers.

How do humans refer to machines without memorizing numbers? That's DNS.

Every one of those is a separate problem with a separate solution, and understanding networking is mostly understanding which problem each piece solves.

Packets

Data doesn't travel as a stream. It's broken into **packets** — discrete chunks, each with addressing information attached.

Why packets rather than a continuous connection?

Sharing. Many conversations interleave on one link. A dedicated circuit per conversation, as the old telephone network used, wastes capacity when nobody's talking.

Resilience. Packets route independently. A failed link means subsequent packets take another path. This was the original design motivation.

Error handling. A corrupted packet is retransmitted. A corrupted stream would have to restart.

The consequence: packets can arrive out of order, arrive duplicated, or not arrive at all. Higher layers deal with that, and whether they bother is the difference between TCP and UDP.

Encapsulation

This is the single most useful concept in the whole subject, and it explains the layered model better than the layered model explains itself.

Each layer wraps the layer above it in its own header, like nesting envelopes.

Your browser produces an HTTP request. TCP wraps it with a header containing port numbers and sequence information. IP wraps that with source and destination addresses. Ethernet wraps that with MAC addresses and a checksum. The result goes on the wire.

At the far end, each layer strips its own header and passes the contents up.

[Ethernet [IP [TCP [HTTP data]]]]

Why this matters practically: it tells you where to look. If a packet arrives but the application does nothing, the Ethernet and IP layers worked and the problem is above them. If nothing arrives at all, you're looking lower. **Every troubleshooting decision is fundamentally a question about which envelope failed.**

It also explains **MTU** — the maximum transmission unit. Each layer's headers consume space. A standard Ethernet frame carries 1500 bytes of payload; IP and TCP headers eat 40 of that, leaving 1460 for data. Add a VPN tunnel and its headers eat more. Chapter 6 covers why this causes some of the most frustrating problems in networking.

Addresses, Plural

A device has several addresses simultaneously, at different layers, and confusing them is a common source of muddle.

MAC address — 48 bits, burned into the network interface, unique per interface. Used only on the local segment. A packet crossing a router gets a new MAC address at each hop.

IP address — logical, assigned, and routable. This is what identifies a device across networks. It changes when the device moves; the MAC doesn't.

Port number — identifies which application on the device.

Hostname — a human-readable name that DNS resolves to an IP address.

The useful analogy: the IP address is the street address, the MAC address is which door on the building, the port is which apartment, and the hostname is "the Johnsons' place."

Network Topologies

Physical topology is how things are actually wired. **Logical topology** is how traffic flows. They're frequently different, which is why a network diagram that shows only cabling misleads.

Star — everything connects to a central switch. Modern Ethernet is physically a star.

Bus — everything shares one medium. Historical for wired Ethernet, and still exactly how wireless works — which is why wireless behaves differently from wired in ways Chapter 13 covers.

Ring — each device connects to two neighbors. Rare in enterprise, common in industrial networks for its fast, deterministic failover.

Mesh — multiple paths between nodes. The internet's structure, and increasingly how wireless backhaul is built.

Hierarchical — the classic enterprise design: access layer where devices connect, distribution layer aggregating and applying policy, core layer moving traffic fast. Smaller networks collapse these, and the thinking still applies — you should know which role each device plays.

Bandwidth, Throughput, Latency

Three different things that get conflated constantly.

Bandwidth — the theoretical maximum capacity of a link. What you bought.

Throughput — what you actually achieve. Always less, because of protocol overhead, contention, and the limits of the devices at each end.

Latency — the delay for a packet to make the trip. Measured one-way or as round-trip time.

Jitter — the variation in latency. Voice and video care about jitter more than about raw bandwidth.

Packet loss — the proportion that don't arrive. Small amounts devastate TCP throughput, because TCP interprets loss as congestion and backs off.

The key insight that surprises people: bandwidth and latency are independent. A satellite link can have enormous bandwidth and terrible latency. Adding bandwidth does nothing for a latency problem, and a great deal of money has been spent on the wrong one.

Bandwidth-delay product — bandwidth times round-trip time — is the amount of data “in flight” on a link at any moment. On high-bandwidth, high-latency paths this is large, and TCP needs a correspondingly large window to keep the pipe full. Under-sized windows are why a fast link across a long distance can deliver disappointing throughput.

Wire Check

1. Find your device's MAC address, IP address, and hostname. Note that they're three different things.
2. Ping something local and something distant. Compare round-trip times.
3. Run a speed test and compare the result to your subscribed bandwidth.
4. Draw your own network. Distinguish physical wiring from logical traffic flow.

Chapter 2: The Layered Model

Layering is how a problem this large stays tractable. Each layer provides a service to the one above and doesn't care how the one below does its job.

OSI and TCP/IP

Two models, and you'll encounter both.

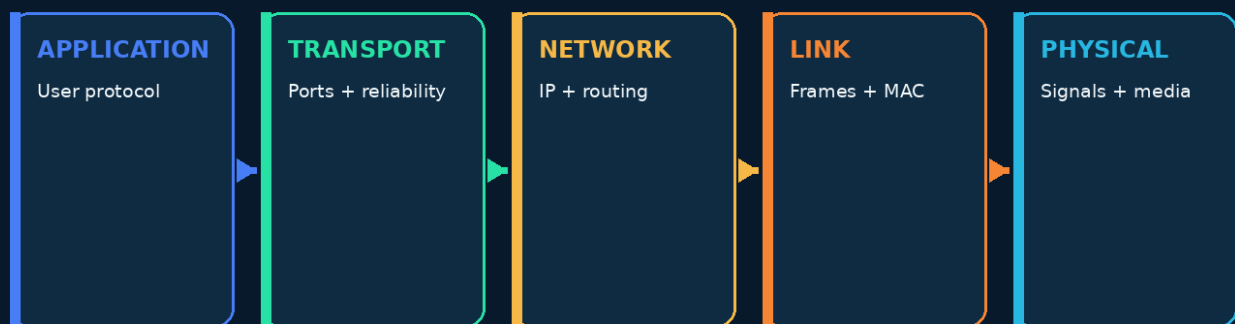
The OSI model has seven layers. It's a teaching framework and a shared vocabulary rather than a description of anything actually implemented.

Layer	Name	Function	Examples
7	Application	Interface to the user's software	HTTP, SMTP, DNS
6	Presentation	Encoding, encryption, compression	TLS, character encoding
5	Session	Establishing and managing sessions	Session management
4	Transport	End-to-end delivery, ports	TCP, UDP
3	Network	Logical addressing, routing	IP, ICMP
2	Data Link	Local addressing, framing	Ethernet, ARP, VLANs
1	Physical	Bits on the medium	Copper, fiber, radio

The TCP/IP model has four layers and describes what's actually built:

How Network Layers Work Together

Each layer provides a service to the layer above



N9LYA TECHNICAL GUIDES

Figure 1: How network layers work together

Layer	Contains OSI
Application	5, 6, 7
Transport	4
Internet	3
Link	1, 2

Which should you use? TCP/IP describes reality. OSI is the vocabulary everyone speaks — people say “layer 3 device” and “a layer 2 problem” constantly, and you need to understand them. Learn OSI’s numbering, think in TCP/IP’s structure.

Why Layer Numbers Are Useful

The numbers are shorthand for where the problem is, and that’s their real value.

“It’s a layer 1 problem” — cable, connector, transceiver, port. Check the link light.

“Layer 2” — switching, VLANs, MAC addressing, spanning tree. Devices on the same segment can’t reach each other.

“Layer 3” — routing, IP addressing, subnetting. Devices on different segments can’t reach each other.

“Layer 4” — ports and firewall rules. Ping works, the application doesn’t.

“Layer 7” — the application itself. Everything connects and the software misbehaves.

“A layer 8 problem” is a joke about the user, and it’s used more often than it should be — most apparent user errors are actually design problems.

The Bottom-Up Rule

When troubleshooting, work from the bottom up.

There is no point examining a firewall rule when the cable is unplugged. There is no point debugging an application when routing is broken.

The sequence:

1. **Physical** — link light? cable seated? correct port?
2. **Data link** — MAC address learned? correct VLAN? interface up?
3. **Network** — IP address correct? subnet mask correct? gateway reachable? routing table sane?
4. **Transport** — port open? firewall permitting? service listening?
5. **Application** — does the software work?

This ordering is the most valuable procedural habit in networking, and Chapter 19 develops it fully. People skip to layer 7 because that’s where the symptom appears, and lose hours.

Protocol Data Units

Each layer names its chunk differently. The terminology appears constantly in documentation and error messages:

Layer	Unit
Application	Data / message
Transport	Segment (TCP) / Datagram (UDP)
Network	Packet
Data Link	Frame
Physical	Bits

People use “packet” loosely for all of them. Precision helps when you’re reading a capture, where the distinction is visible.

Devices by Layer

Device	Operates At	Function
Cable, transceiver	1	Carries bits
Hub	1	Repeats to all ports. Obsolete.
Switch	2	Forwards frames by MAC address
Layer 3 switch	2 and 3	Switches and routes
Router	3	Forwards packets between networks
Firewall	3, 4, and often 7	Filters by policy
Load balancer	4 or 7	Distributes connections
Proxy	7	Intermediates application traffic

Modern devices blur these boundaries. A router with a built-in switch, firewall, wireless access point, and VPN server is one physical box performing five roles at four layers. **When troubleshooting, think about which role is involved rather than which box.**

What Each Layer Actually Buys You

Layer 1 — the ability to move bits between two points.

Layer 2 — the ability for many devices to share a medium, addressed individually. Its limit is that it doesn’t scale; a Layer 2 network floods broadcasts to every device, so a large flat network drowns in them.

Layer 3 — the ability to build a network of networks. Routers separate broadcast domains and make hierarchy possible, which is what makes the internet possible.

Layer 4 — the ability to run many conversations to one host, and to choose reliability or speed.

Layer 7 — everything people actually use.

Wire Check

1. For each layer, name one thing you could check on your own network.
2. Identify which layers each device in your network operates at.
3. Take a recent network problem you had. Which layer was it at? Did you check the layers below it first?
4. Watch a link go down and back up. Which layers were affected, and in what order did they recover?

Chapter 3: Ethernet and Layer 2

Ethernet won. It’s in every office, every home, every data center, and increasingly on the plant floor. Understanding how it works locally is the foundation for everything above it.

Frames

An Ethernet frame carries:

- **Preamble** — a synchronization pattern
- **Destination MAC** — 6 bytes
- **Source MAC** — 6 bytes
- **EtherType or length** — what’s inside (0x0800 for IPv4, 0x86DD for IPv6, 0x0806 for ARP)
- **Payload** — 46 to 1500 bytes normally

Encapsulation and Delivery

Headers are added on send and removed on receipt



N9IYA TECHNICAL GUIDES

Figure 2: Network encapsulation and delivery

- **Frame check sequence** — a CRC for error detection

A frame with a bad CRC is discarded silently. Not retransmitted, not reported to the application — just dropped, with a counter incrementing on the interface. **Those error counters are the first thing to check on a link that's "slow"**, because a cable problem causing a few percent packet loss will devastate TCP throughput while the link light stays cheerfully green.

MAC Addresses

48 bits, written as six hex bytes: 00:1A:2B:3C:4D:5E.

The first three bytes are the **OUI** — the organizationally unique identifier, assigned to the manufacturer. Look one up and you learn what kind of device you're dealing with, which is genuinely useful when you find an unknown MAC on your network.

Two bits in the first byte have meaning: one marks a multicast address, and one marks a locally administered address — which is what randomized MAC addresses on phones and laptops use.

MAC addresses are local. They're used to deliver a frame across one segment. Cross a router and the frame gets rebuilt with new source and destination MACs. **The IP addresses stay the same end to end; the MAC addresses change at every hop.** That single distinction clears up an enormous amount of confusion.

Broadcast is FF:FF:FF:FF:FF:FF — every device on the segment receives it.

Switching

A switch learns and forwards:

1. A frame arrives on a port.
2. The switch records the **source** MAC and the port it came from, in its MAC address table.
3. It looks up the **destination** MAC.
4. If known, it forwards out that port only.
5. If unknown, it floods out all ports except the one it arrived on.
6. Broadcasts and unknown multicast are always flooded.

Entries age out, typically after five minutes of silence, which is why a device that's been quiet gets flooded to once before the switch relearns it.

```
show mac address-table      # most switch CLIs
bridge fdb show             # Linux
```

Practical uses of the MAC table: find which port a device is on, detect a device that's moved, and spot a MAC appearing on two ports alternately — which indicates a loop or a duplicated address.

Collision and Broadcast Domains

A collision domain is a set of devices that can interfere with each other's transmissions. On modern switched, full-duplex Ethernet, **each port is its own collision domain** and collisions essentially don't occur. If you see collision counters incrementing on a modern switch port, you have a duplex mismatch.

A broadcast domain is the set of devices that receive each other's broadcasts. A switch forwards broadcasts everywhere; **a router does not**. One VLAN is one broadcast domain.

This is why segmentation matters for performance as well as security. Broadcast traffic scales with the number of devices, and every device must process every broadcast. A flat network of several hundred devices spends measurable CPU on broadcasts alone. Split it into VLANs and each device only processes its own segment's noise.

ARP

IP knows the destination IP address. Ethernet needs a MAC address. **ARP bridges the gap.**

1. Host needs to reach 192.168.1.50 on its own subnet.
2. It broadcasts: “who has 192.168.1.50?”
3. That host replies with its MAC.
4. The requester caches it and sends the frame.

```
ip neigh show          # Linux
arp -a                 # Windows, older Unix
```

If the destination is on another subnet, the host ARPs for the gateway’s MAC instead, and sends the frame there with the destination IP intact. This is the mechanism by which routing works at the frame level, and it’s worth being clear about: the packet is addressed to the far host, the frame is addressed to the next router.

ARP has no authentication. A device that claims an address is believed. That’s the basis of ARP spoofing, and it’s why switch features like dynamic ARP inspection exist.

Gratuitous ARP — an unsolicited announcement — is how devices update everyone’s cache after a failover, and how duplicate address detection works.

Duplicate IP addresses produce a distinctive symptom: intermittent connectivity that changes depending on which device answered ARP most recently. `ip neigh` showing an address flipping between two MACs is the diagnosis.

Duplex and Speed

Full duplex transmits and receives simultaneously. **Half duplex** does one at a time and requires collision detection.

Auto-negotiation normally works. When it doesn’t, you get a **duplex mismatch** — one end full, one end half — and the symptom is characteristic: the link works, and throughput is terrible, with late collisions on the half-duplex side and CRC errors on the full-duplex side.

The classic cause is one end hardcoded and the other on auto. A hardcoded end doesn’t negotiate; the auto end falls back to half duplex. **Either hardcode both ends or auto-negotiate both ends.** Never mix.

Modern gigabit and above requires auto-negotiation by standard, which has made this problem much rarer — and it still appears on older equipment, which industrial environments have plenty of.

Spanning Tree

A loop in a Layer 2 network is catastrophic. Broadcast frames circulate forever, multiplying at each switch, and the network saturates within seconds. There is no TTL in an Ethernet frame to stop it.

Spanning Tree Protocol prevents this by detecting loops and blocking redundant paths, activating them when the primary fails.

Variants: the original STP converges slowly (30–50 seconds). **RSTP** converges in seconds and is what you should be running. **MSTP** maps multiple VLANs to spanning tree instances. Vendors have proprietary variants too.

Practical guidance:

Set the root bridge deliberately. By default the switch with the lowest bridge ID wins, which is frequently the oldest and worst-placed device in the network. Configure your core switch with a low priority so it’s root.

Enable portfast (edge port) on access ports so end devices don’t wait through the listening and learning states. A workstation that takes 30 seconds to get a DHCP address is usually this.

Enable BPDU guard on those ports. If a switch appears where a workstation should be, the port shuts down. This prevents someone plugging a cheap switch into a wall jack and becoming your root bridge, which happens more often than you'd think.

Never disable spanning tree to "improve performance." The performance improvement is imaginary and the consequence of a loop is a total outage.

Link Aggregation

Combining multiple links into one logical link, for bandwidth and redundancy.

LACP (802.3ad) negotiates dynamically. **Static** aggregation doesn't and is easier to misconfigure.

The limitation that disappoints everyone: a single conversation uses a single link. Two bonded gigabit links give you two gigabits across many flows, not two gigabits for one transfer. The hash policy — by MAC, by IP, by IP and port — determines how flows distribute, and choosing a policy that includes ports gives better balance for hosts with many connections.

Wire Check

1. Look up the OUI of three MAC addresses on your network. Do they match what you expect?
2. Check interface error counters on every switch port. Any CRC errors?
3. Find which switch port a specific device is on, using the MAC address table.
4. Identify your spanning tree root bridge. Is it the device you'd choose?
5. Verify portfast and BPDU guard are configured on access ports.

Chapter 4: IP Addressing and Subnetting

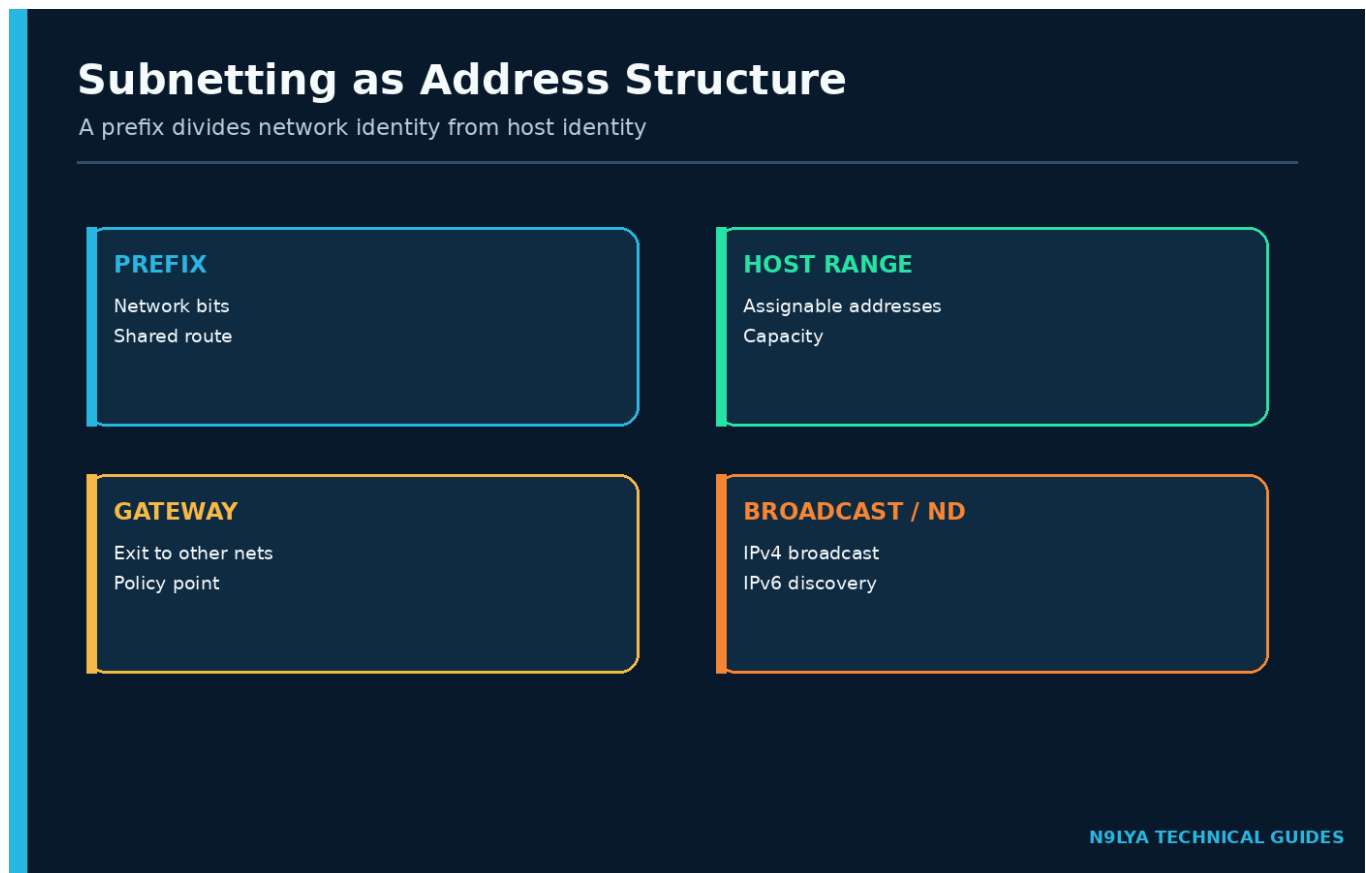


Figure 3: Subnetting as address structure

End of the free sample

Networking — Second Edition · by Jerome M. Kutche

That was the first 12 of 76 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 (\$9.99, PDF + EPUB):

<https://booksrus.biz/product/networking/>

Also on Gumroad:

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



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.