INTERNET PROTOCOLS

FROM FIRST PRINCIPLES

HOW COMPUTERS COMMUNICATE ACROSS THE WORLD

BITS & SIGNALS

FROM PHYSICS TO DATA

PACKETS & FRAMES

ENCODING, FRAMING, ERROR CHECKS

ADDRESSING & ROUTING

IP, SUBNETS, ROUTING PROTOCOLS

RELIABILITY & CONGESTION

TCP, FLOW CONTROL, CONGESTION AVOIDANCE

ENCRYPTION & SECURITY

TLS, KEYS, CERTIFICATES, PRIVACY

NAMING & RESOLUTION

DNS

DOMAINS, DNS, LOOKUPS

LEONID ZAITSEV

Internet Protocols from First Principles

How Computers Communicate Across the World

Steve Publications

This book is available at
https://leanpub.com/internetprotocolsfromfirstprinciples

This version was published on 2026-08-18

Contents

How Computers Communicate Across the World

This book takes you from zero networking knowledge to expert-level understanding of how the Internet actually works. Beginning with the fundamental problem of communication between computers, we derive every concept logically: signals and bits, encoding and packets, frames and links, addressing and routing, reliability and congestion, encryption and naming. Each protocol is explained in depth, why it exists, what problem it solves, how it works internally and what trade-offs its designers made. You will learn to read packet captures, trace connections end to end through autonomous systems, implement networking components from scratch and debug real network problems. No prior networking knowledge is assumed. Only curiosity and a willingness to understand the machinery beneath the surface.

Introduction

Imagine two computers sitting in different rooms, each holding data the other needs. They cannot see each other. They share no memory. Between them lies only wire or air. How do they exchange information reliably?

This simple question is the origin of everything you will learn in this book.

The Internet protocol stack, Ethernet, IP, TCP, DNS, TLS, HTTP, is not an arbitrary collection of standards invented by committees. It is a logical response to fundamental engineering problems in distributed communication. Each layer exists because lower layers cannot solve certain problems and each protocol emerged from specific constraints: limited bandwidth, unreliable physical media, the need to connect many networks, the demand for reliability, the necessity of security.

Understanding these protocols requires more than memorizing header fields or state machines. You must understand the problems they were designed to solve, the trade-offs their creators faced and why alternative designs succeeded or failed. When you know why TCP uses a three-way handshake instead of two, or why IP deliberately omits reliability, or why BGP is simultaneously powerful and fragile, you gain something durable: the ability to reason about any networking problem, even ones involving protocols that do not yet exist.

This book is organized as a progressive derivation from first principles. We begin with physical signals and derive digital communication. From there we build links, then networks of networks, then reliable transport, then naming systems, then encrypted channels, then application protocols. Each new abstraction is motivated by a problem the previous layer cannot address. Along the way we examine packet formats field by field, trace real communications end to end through switches and routers and autonomous systems, implement components from scratch in code and learn to diagnose problems using professional tools.

By the end you will possess a unified mental model of how information moves across the Internet: what changes at each hop, what remains unchanged, where failures occur, how protocols recover, why certain designs

scale while others collapse under load. You will understand not just what the Internet does but why it works the way it does.

The journey begins with a question that seems almost too simple to merit an entire book: how do we get bits from one machine to another?

Chapter 1: The Problem of Communication

Why Computers Need to Talk

Computers were originally built as isolated calculation engines. A mainframe in the 1950s processed data loaded on punch cards, printed results and waited for the next batch. There was no concept of "networking" because there was only one machine that mattered.

But isolation is impractical. In reality we need computers to share resources: a printer in one room should serve users in another; a database in New York should be queryable from London; a web server should respond to millions of clients simultaneously. Beyond resource sharing, distributed systems offer advantages no single machine can match: redundancy so that failure of one component does not halt the whole system, geographic distribution so that computation happens closer to where data originates or where it is consumed and specialization so that different machines handle different tasks efficiently.

The fundamental problem this book addresses is therefore straightforward: how do we enable separate computers, potentially far apart, with no shared memory and no guaranteed coordination, to exchange information reliably and efficiently?

This deceptively simple question decomposes into many subproblems:

- **Physical transmission**: How do we actually move signals between machines over wires or through air?
- **Encoding**: How do we represent abstract information as physical signals that can be transmitted and reconstructed?
- **Reliability**: What happens when signals are corrupted, lost, or arrive out of order?
- **Addressing**: How does a message find its way to the correct destination among billions of computers?

- **Naming**: How do humans refer to machines without memorizing numerical addresses?
- **Multiplexing**: How do multiple applications on the same machine share network resources?
- **Security**: How do we prevent eavesdropping, tampering and impersonation?

Each of these problems is solved by protocols at different layers of the networking stack. Before we can understand those solutions, we must understand the physical reality they operate within.

From Electricity to Information

At the lowest level, communication between computers is simply the controlled movement of energy. In wired networks this energy takes the form of electrical voltage differences along copper conductors or light pulses through fiber optic cables. In wireless networks it is electromagnetic radiation, radio waves propagating through air or vacuum.

Consider a copper wire connecting two machines. To send information, one machine varies the voltage on that wire over time. The other machine measures these voltage variations and interprets them as data. This is all there is at the most fundamental level: voltage changes at one end, voltage measurements at the other.

But this simplicity masks enormous complexity. Voltage on a wire is not a clean, precise signal. It is affected by resistance in the conductor (which causes voltage drop over distance), capacitance between adjacent wires (which causes signals to "bleed" into neighboring channels, a phenomenon called crosstalk), electromagnetic interference from nearby power lines or radio transmitters and thermal noise, random electron motion that creates small unpredictable voltage fluctuations even when no signal is being transmitted.

Every communication system must contend with this noisy physical reality. The goal is not perfect transmission, that is physically impossible, but reliable transmission: the receiver reconstructs the sender's original information correctly despite imperfections in the channel.

Signals, Noise and Reliability

To understand how we achieve reliability over unreliable media, we need to distinguish between analog and digital signals.

An **analog signal** varies continuously. The voltage on a wire can take any value within a range: 2.37 volts, 2.371 volts, 2.3705 volts and so on without limit. Analog communication encodes information in the precise amplitude, frequency, or phase of the signal. Traditional telephony used analog signals to transmit human voice as electrical waveforms that mirrored sound pressure variations.

Analog signals have a critical weakness: noise is indistinguishable from signal. If thermal noise adds 0.01 volts to your signal, the receiver cannot tell whether that extra voltage was intentional or accidental. Every time an analog signal is amplified or copied, noise accumulates. After enough copies, the original information becomes unrecoverable, this is why old cassette tapes sound hissy and why copying a VHS tape degrades quality.

Digital signals solve this problem through discretization. Instead of using infinitely many voltage levels, we use only two: one level represents “0” and another represents “1”. For example, voltages below 0.8 volts might mean “0” while voltages above 2.0 volts mean “1”, with a gap in between where no valid signal should appear.

This binary approach transforms the problem. Noise can still corrupt the signal, a voltage meant to be 3.3 volts (representing “1”) might arrive as 2.5 volts due to resistance and interference, but as long as it stays above the threshold of 2.0 volts, the receiver correctly interprets it as “1”. The gap between valid levels provides a noise margin: immunity to small errors.

Digital signals can also be regenerated rather than merely amplified. A repeater receives a noisy signal, decides whether it was meant to be a 0 or a 1 based on threshold comparison and then transmits a fresh, clean signal at the full original amplitude. This regeneration prevents noise accumulation over long distances, each repeater resets the noise clock to zero.

The trade-off is bandwidth. A digital signal that represents only two values carries less information per unit of time than an analog signal using many levels. But this limitation is manageable: we compensate for lower per-symbol information by transmitting symbols faster and more importantly,

digital signals enable powerful error detection and correction techniques that analog systems cannot match.

Digital Representation: Bits and Bytes

The fundamental unit of digital information is the **bit**, short for binary digit, which can take one of two values: 0 or 1. A single bit carries very little information: it can distinguish between only two possibilities. But we combine bits into larger units to represent more complex data.

Eight bits grouped together form a **byte**, the standard unit of digital storage and transmission. A byte can represent 2^8 = 256 different values, enough to encode all uppercase letters, lowercase letters, digits, punctuation marks and control characters used in text. The ASCII encoding maps each of the first 128 byte values to a specific character: for example, the byte value 0x41 (binary 01000001) represents the letter "A", while 0x61 (binary 01100001) represents "a".

Modern systems use UTF-8 encoding, which extends ASCII backward-compatibly to represent all characters in virtually every writing system on Earth. In UTF-8, common English characters still use single bytes with the same values as ASCII, while less common characters use two, three, or four bytes.

But computers do not just store text. Numbers, images, audio, video and executable programs are all represented as sequences of bits. An image might be stored as a grid of pixels where each pixel is described by three bytes representing red, green and blue intensity values. A program is a sequence of bytes that the CPU interprets as instructions: "load this value into register X", "add these two numbers", "jump to instruction at address Y".

When we say a computer "sends data" over a network, we mean it converts some internal representation, whether text, numbers, or program state, into a sequence of bits and transmits those bits one by one across the physical medium. The receiver reconstructs the bit sequence and interprets it according to agreed-upon rules about what those bits mean.

This interpretation layer is called a **protocol**. A protocol is simply a shared understanding between sender and receiver about how to encode, transmit and interpret information. Human language is a protocol: we agree that certain sound patterns mean certain things. Internet protocols are the same idea applied to machine communication.

Synchronization and Clocks

Transmitting bits over a wire requires not just encoding each bit as a voltage level but also agreeing on timing. The receiver must know when to sample the signal to determine whether each bit is a 0 or a 1. If the sender transmits one bit every microsecond, the receiver must also read one value every microsecond and it must stay synchronized with the sender's timing.

This synchronization problem has two aspects: **clock rate** agreement (both sides agree on how fast bits are transmitted) and **bit alignment** (the receiver knows where each bit boundary falls).

In early serial communication systems, these problems were solved using **start-stop framing**. Before transmitting each byte, the sender places the line in a "idle" state (typically high voltage), then drops to low voltage for one bit period as a **start bit** to signal that data is coming. The receiver detects this transition from high to low and knows that the next eight bits are data. After the data bits come one or more **stop bits** (high voltage) to return the line to idle and give the receiver time to prepare for the next byte.

Start-stop framing has a crucial advantage: it does not require sender and receiver to share a precise clock. Each byte is self-synchronizing because the start bit marks where that byte begins. The receiver only needs its internal clock to be approximately correct, within maybe 10 percent of the true rate, because it samples each bit in the middle of its expected time slot, where timing errors have had the least accumulation.

Modern high-speed networks use more sophisticated techniques. Instead of start-stop bits for every byte, they use continuous streams with embedded clock information through line coding schemes like Manchester encoding (which guarantees a voltage transition in the middle of every bit period, allowing the receiver to extract timing from the signal itself) or 8b/10b encoding (which maps 8 data bits to 10 transmitted bits chosen to ensure enough transitions for clock recovery while limiting runs of identical values).

The key insight is that synchronization is not optional: without agreement on when bits start and end, voltage levels are meaningless. Every physical layer protocol must solve this problem before higher-level protocols can function.

The First Networks

With these fundamentals in place, digital signals, encoding, timing, we can understand how the first computer networks emerged.

In the early 1960s, researchers at universities and government labs had expensive mainframe computers that sat idle much of the time. It seemed wasteful: why not let users at one institution access computers at another? But this raised technical problems never before considered. How do you connect two different computer models with incompatible instruction sets and memory layouts? How do you share a single communication channel among many users?

The initial approach was simple point-to-point connections using serial lines (the same technology used for teleprinters and modems). A terminal at one location connected directly to a mainframe at another via telephone lines. This worked but scaled poorly: connecting N computers required up to N(N-1)/2 separate links if every pair needed direct communication.

A better approach was the **packet-switched network**. Instead of dedicating an entire communication channel to a single conversation (as in traditional telephony), packet switching breaks data into small chunks called packets and sends each packet independently across shared links. Multiple conversations share the same physical infrastructure because their packets are interleaved, distinguished by addressing information in each packet's header.

This idea was developed independently at several places in the late 1960s: Paul Baran at RAND Corporation described "distributed adaptive message blocking" for military communications; Donald Davies at the National Physical Laboratory in Britain coined the term "packet switching"; and Leonard Kleinrock at MIT provided the mathematical foundation through queueing theory.

The ARPANET, funded by DARPA and launched in 1969, was the first operational packet-switched network. It connected four nodes: UCLA, Stanford Research Institute, University of Utah and University of California Santa Barbara. Each node used an Interface Message Processor (IMP), essentially a specialized minicomputer, to handle packet switching while the attached host computers focused on applications.

The ARPANET had to solve problems that remain central to networking today: routing packets across multiple hops when direct paths are unavailable,

handling congested links when too many packets compete for bandwidth and allowing different host computer systems to communicate despite incompatible architectures. The solutions developed for ARPANET evolved into the Internet protocol suite we use today.

But before ARPANET could route packets between sites, it needed a way to transmit those packets across physical links connecting adjacent IMPs. That problem, communication over a single connection, is what we call the **link layer** and it is where our next chapter begins.

Chapter 2: Encoding Data for Transmission

Line Coding: Voltage Levels and Bit Patterns

We now face a concrete engineering question: given a sequence of bits to transmit, how do we map those abstract 0s and 1s onto physical voltage levels on a wire? This mapping is called **line coding** and the choice of encoding scheme has profound consequences for reliability, bandwidth efficiency and implementation complexity.

The simplest possible encoding is **Non-Return-to-Zero Level** (NRZ-L): use one voltage level for 0 and another for 1. For example, -5 volts might represent 0 while +5 volts represents 1. The voltage stays at that level for the entire duration of each bit period and changes only when the next bit has a different value.

NRZ-L has an obvious problem: if you transmit many consecutive bits with the same value, say, eight zeros in a row, the voltage stays constant for the entire duration. A receiver relying on transitions to detect timing information loses synchronization because there is nothing changing to measure. The receiver's clock might drift slightly faster or slower than the sender's and without periodic transitions to correct against, the accumulated error eventually causes the receiver to sample at the wrong time within a bit period, reading incorrect values.

Non-Return-to-Zero Inverted (NRZI) addresses this partially by encoding bits as transitions rather than levels: a transition at the start of a bit period means "1", no transition means "0". This guarantees some transitions when data contains many 1s but still fails with long runs of 0s.

A more robust solution is **Manchester encoding**, used in classic Ethernet (10BASE-T). In Manchester encoding, every bit period has exactly one transition in the middle: a high-to-low transition represents "1", while a low-to-high transition represents "0". This guarantees a transition every bit period

regardless of the data pattern, providing continuous timing information for clock recovery.

The trade-off is bandwidth: Manchester encoding requires twice as many transitions as NRZ-L for the same data rate because every bit generates at least one transition. For a 10 Mbps Ethernet link using Manchester encoding, the signal changes state at up to 20 MHz, the signaling rate (baud rate) is double the data rate.

Modern high-speed Ethernet uses more sophisticated encodings. **4B/5B encoding** (used in Fast Ethernet) maps every 4 data bits to a 5-bit code chosen from a set that guarantees enough transitions for clock recovery while keeping the overhead at only 25 percent rather than Manchester's 100 percent. **8b/10b encoding** (used in Gigabit Ethernet, USB, SATA) similarly maps 8 data bits to 10 transmitted bits with controlled transition properties.

These encodings also solve another subtle problem: **DC balance**. If an encoding produces significantly more 1s than 0s over time, the average voltage on the wire shifts away from zero. This DC bias can cause problems in AC-coupled circuits (which use capacitors to block DC voltage) and increases power consumption. Good line codes maintain roughly equal numbers of 1s and 0s over time, keeping the signal centered around zero volts.

Serial vs Parallel Transmission

Before transmitting bits across a link, we must decide whether to send them one at a time over a single wire (serial transmission) or multiple bits simultaneously over multiple wires (parallel transmission).

Parallel transmission seems obviously faster: eight wires can transmit one byte per clock cycle while a single serial wire transmits only one bit per cycle. Early computer buses used parallel transmission extensively, the original IBM PC's ISA bus transferred data in 16-bit words and later PCI buses used 32 or 64 bits in parallel.

But parallel transmission has a fatal flaw at high speeds: **skew**. Different wires have slightly different lengths and electrical properties, so signals travel at slightly different speeds. At low speeds this difference is negligible, but at high speeds the bits of a single byte arrive at measurably different times. The receiver must wait for all bits to arrive before it can interpret the byte correctly, limiting effective speed. Worse, adjacent wires in a parallel bus

interfere with each other through crosstalk: when one wire switches voltage, electromagnetic coupling induces unwanted voltage changes on neighboring wires.

Serial transmission avoids both problems because there is only one wire (or pair). Modern serial links achieve astonishing speeds, USB 3.2 reaches 20 Gbps, PCIe 5.0 reaches 32 Gbps per lane, by using sophisticated signaling techniques: equalization to compensate for frequency-dependent attenuation, forward error correction to recover from bit errors without retransmission and advanced line codes that pack more information into each symbol.

The lesson is counterintuitive but important: simplicity at the physical layer (one wire instead of many) enables complexity at higher layers. The Internet runs on serial links because serial transmission scales to high speeds in a way parallel transmission cannot.

Start Bits, Stop Bits and Framing Signals

We have solved how to encode individual bits as voltage levels. Now we must solve how to group those bits into meaningful units that the receiver can identify and process. This is the **framing** problem: how does the receiver know where one byte ends and the next begins, or where one message ends and another begins?

For simple serial communication between two devices (like a computer connected to a serial terminal), the solution used for decades is **start-stop framing**, also called asynchronous serial communication. The protocol works as follows:

1. When idle, the transmitting device holds the line at a "mark" state (typically high voltage, representing logical 1).
2. To begin transmitting a byte, it drops the line to "space" (low voltage) for one bit period. This is the **start bit**.
3. It then transmits the eight data bits of the byte, least significant first.
4. Finally it transmits one or more **stop bits** at the mark state to return the line to idle.

The receiver continuously monitors the line. When it detects a transition from mark to space (the start bit), it knows a byte is coming. It then samples

the line at predetermined intervals, typically in the middle of each expected bit period, to read the eight data bits and verify the stop bit(s).

Start-stop framing has several important properties:

- **Self-synchronization**: Each byte carries its own timing marker (the start bit), so sender and receiver do not need to share a precise clock. They only need clocks that are approximately matched within maybe 10 percent accuracy.
- **Variable gaps**: Bytes can be separated by arbitrary amounts of idle time. The sender can transmit bytes at irregular intervals and the receiver handles this naturally because it simply waits for the next start bit.
- **Simple error detection**: If the stop bit is not at the expected mark state, the receiver knows something went wrong (the line was still in space when it should have returned to mark). This indicates a framing error: bits may have been lost or added and the byte should be discarded.

Start-stop framing is inefficient for high-speed transmission because each byte requires overhead bits (at minimum 1 start bit plus 1 stop bit = 2 extra bits per 8 data bits, or 20 percent overhead). It also cannot handle very high speeds because clock drift accumulates over the duration of each byte, limiting reliable operation to roughly 115 kbps for typical hardware.

Modern high-speed protocols use different framing approaches. Ethernet frames begin with a distinctive bit pattern called the **preamble** (7 bytes of alternating 1s and 0s: 10101010…) followed by a **start frame delimiter** (SFD: 10101011). The preamble allows the receiver to synchronize its clock through the regular transitions and the SFD's distinctive pattern (two consecutive 1s where only alternating bits were expected) signals that actual frame data follows.

Clock Recovery and Timing

At higher speeds, start-stop framing becomes impractical because each byte's self-synchronization overhead is too expensive and clock drift over even a single byte's duration causes errors. Instead we need **clock recovery**: extracting timing information directly from the transmitted signal so the receiver can stay synchronized without per-byte markers.

Clock recovery works by detecting transitions in the signal and using those transitions to adjust the receiver's internal clock phase. A common implementation uses a **Phase-Locked Loop** (PLL), an electronic circuit that continuously compares the phase of its internal oscillator against the phase of incoming signal transitions and adjusts frequency to minimize the difference.

For clock recovery to work, the signal must contain enough transitions. This is why line codes like Manchester encoding guarantee transitions in every bit period: they make clock recovery reliable regardless of data content. Without guaranteed transitions, long runs of identical bits (all 0s or all 1s) cause the PLL to lose lock because there are no edges to track.

The requirement for sufficient transitions constrains both line coding choices and data patterns. Some protocols use **bit stuffing**: after detecting a certain number of consecutive identical bits in the data stream (say, five 1s), the transmitter automatically inserts an extra bit of the opposite value (a 0) to break the run. The receiver removes these stuffed bits when it sees the same pattern. This ensures transitions exist even if the original data contains long runs.

Clock recovery is not perfect. Even with a PLL, there is always some residual timing jitter, small variations in when edges actually occur versus when they ideally should. Protocol designers must account for this jitter by ensuring that signals are sampled well within the middle of each bit period, far enough from transitions that jitter does not cause incorrect sampling.

Error Detection with Parity and CRC

No physical transmission is perfect. Despite all our encoding and timing techniques, bits still get corrupted: a voltage meant to be high arrives low due to interference, or electromagnetic noise induces a spurious transition. We need mechanisms to detect when errors have occurred so we can request retransmission or correct the error.

The simplest error detection mechanism is **parity**. A parity bit is appended to each byte such that the total number of 1s in the byte plus parity bit is either always even (even parity) or always odd (odd parity). For example, with even parity, the byte 0x5A (binary 01011010, which has four 1s) would get a parity bit of 0 (total remains even), while byte 0xFF (eight 1s) also gets parity bit 0, but byte 0xFE (seven 1s) gets parity bit 1 to make the total eight.

Parity can detect any single-bit error in the protected data: if one bit flips during transmission, the parity of received bits no longer matches the expected even or odd count. But parity cannot detect errors that flip an even number of bits (two bits flipping preserves parity) and it provides no information about which bit is wrong so correction is impossible.

A much more powerful technique is the **Cyclic Redundancy Check** (CRC). A CRC treats the data as a polynomial over GF(2), essentially a sequence of bits where addition is XOR and multiplication is shift-and-XOR and divides it by a fixed generator polynomial. The remainder of this division becomes the CRC value, which is appended to the data.

For example, Ethernet uses a 32-bit CRC (CRC-32) computed with the polynomial $x^{32+x}26 + x^{23+x}22 + x^{16+x}12 + x^{11+x}10 + x^{8+x}7 + x^{5+x}4 + x\text{^}2 + x + 1$ (represented as the hex value 0x04C11DB7). The sender computes this CRC over all frame bytes and appends it as the Frame Check Sequence (FCS). The receiver performs the same computation on received data including the FCS; if the result is zero, the frame passed the check.

CRC-32 can detect:

- All single-bit errors
- All double-bit errors
- All errors affecting an odd number of bits
- All burst errors up to 32 bits long
- Longer burst errors with probability very close to 1 (specifically, 1 - 2^(-32) for random errors longer than 32 bits)

The mathematical properties that make CRCs effective stem from polynomial algebra over GF(2). The generator polynomial is chosen so that common error patterns (single bit flips, burst errors from interference) correspond to polynomials that are not divisible by the generator, guaranteeing detection.

CRC computation can be implemented efficiently in hardware using a shift register with feedback taps corresponding to the polynomial's terms, or in software using table lookup methods that process multiple bits at once. Modern network interface cards compute and verify CRCs in hardware at line speed without involving the CPU.

When a CRC check fails, the frame is silently discarded. Higher-layer protocols (like TCP) notice that expected data never arrived and request retransmission. The link layer does not attempt retransmission itself, that would

require acknowledgment messages and state management that complicate the design. Instead it provides a simple service: deliver frames that pass CRC checking and discard those that do not.

Modulation for Wireless Media

So far we have discussed wired transmission where bits are encoded as voltage levels on a conductor. Wireless communication faces different physical constraints: there is no wire to guide the signal, so information must be carried by electromagnetic waves propagating through free space.

Wireless modulation maps digital data onto properties of a radio carrier wave, typically its amplitude, frequency, or phase. The three basic modulation types are:

Amplitude Shift Keying (ASK) varies the carrier's amplitude: high amplitude for 1, low amplitude for 0. Simple but vulnerable to noise because amplitude is easily affected by interference and distance-related attenuation.

Frequency Shift Keying (FSK) uses different frequencies for different symbols: one frequency for 0, another for 1. More robust than ASK because frequency changes are less affected by amplitude noise. Used in early wireless modems and some IoT protocols.

Phase Shift Keying (PSK) varies the phase of the carrier wave. In Binary PSK (BPSK), a 0 might be represented by phase 0 degrees while a 1 uses phase 180 degrees. More sophisticated versions like Quadrature PSK (QPSK) use four phases (0, 90, 180, 270 degrees) to encode two bits per symbol. Higher-order QAM (Quadrature Amplitude Modulation) combines amplitude and phase variations to encode even more bits per symbol: 16-QAM encodes 4 bits, 64-QAM encodes 6 bits, 256-QAM encodes 8 bits.

Modern Wi-Fi (IEEE 802.11ac/ax) uses OFDM (Orthogonal Frequency Division Multiplexing), which splits the available spectrum into many closely spaced subcarriers, each modulated independently using QAM. This approach is robust against frequency-selective fading (where some frequencies are attenuated more than others due to multipath reflections) because data spread across many subcarriers can be recovered even if some subcarriers experience deep fades.

Wireless channels introduce additional challenges beyond wired media:

- **Path loss**: signal strength decreases with distance following an inverse-square law (or worse in urban environments)
- **Multipath fading**: signals reflect off buildings and terrain, arriving at the receiver via multiple paths with different delays; these copies can interfere constructively or destructively
- **Doppler shift**: relative motion between transmitter and receiver shifts frequencies
- **Interference**: many devices share the same spectrum

All of these challenges must be handled by physical layer techniques before higher-layer protocols can function. The good news is that once bits are reliably transmitted, whether over copper, fiber, or air, the rest of the networking stack operates independently of the physical medium. This abstraction is one of the Internet's greatest strengths.

Chapter 3: Links, Frames and the First Layer

What Is a Link

We now have the ability to transmit bits reliably between two devices over a physical medium. A **link** is exactly that: a communication path connecting two endpoints at the physical layer. The link could be a copper cable between adjacent computers, a fiber optic strand crossing an ocean, or a wireless channel between a phone and an access point.

From the perspective of higher layers, all links share certain properties:

- They transmit data as sequences of bits
- They have finite bandwidth (maximum bits per second they can carry)
- They introduce latency (time for a bit to travel from sender to receiver)
- They may experience errors (bits corrupted during transmission)
- They connect specific endpoints

The physical layer hides the details of how bits are actually transmitted, voltage levels, modulation schemes, error correction codes and presents a simpler abstraction: a pipe that moves bits from one end to the other. But this abstraction is too coarse for practical communication. We need structure imposed on the bit stream so that receivers can identify message boundaries, determine who sent each message, detect errors and distinguish between different types of data traveling over the same link.

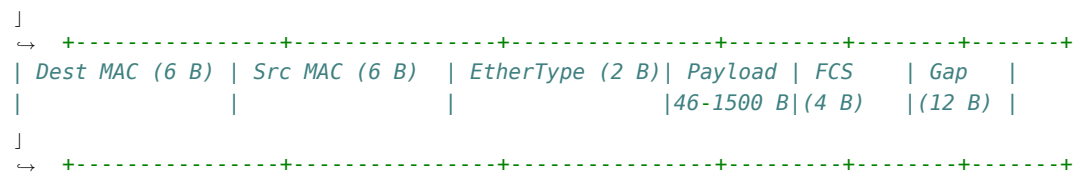
This structure is provided by the **link layer**, also called the data link layer. The link layer groups bits into **frames**, self-contained units with headers containing control information and a payload carrying data from higher layers. Frames are the fundamental unit of communication at this level, just as packets are at the network layer and segments at the transport layer.

The Frame Structure

A frame is not simply raw data wrapped in a header. Its structure solves specific problems that arise when multiple devices share a link or when the physical medium introduces errors:

1. **Addressing**: Frames include destination and source addresses so receivers can determine whether a frame is intended for them
2. **Type identification**: A type field tells the receiver how to interpret the payload, is it an IP packet, an ARP request, something else?
3. **Error detection**: A checksum or CRC allows the receiver to detect corrupted frames and discard them
4. **Boundary marking**: The frame format defines where one frame ends and another begins

Consider the Ethernet II frame format, the most widely used link layer protocol:



Let us examine each field:

Destination MAC address (6 bytes): Identifies the intended recipient of this frame on the local link. A special value with all bits set to 1 (FF:FF:FF:FF:FF:FF) means "broadcast", every device on the link should process this frame.

Source MAC address (6 bytes): Identifies the sender, allowing recipients to reply directly without needing a separate addressing mechanism.

EtherType (2 bytes): Indicates what protocol the payload contains. Common values include 0x0800 for IPv4 packets, 0x86DD for IPv6 and 0x0806 for ARP. This field enables multiple network layer protocols to coexist on the same physical link, a concept called **multiplexing**.

Payload (46-1500 bytes): Contains data from higher layers, most commonly an IP packet. The minimum of 46 bytes exists because classic Ethernet

required frames to be at least 64 bytes total (including headers and FCS) for collision detection to work properly; short payloads are padded with zeros. The maximum of 1500 bytes is called the Maximum Transmission Unit (MTU) and limits how much data fits in a single frame.

Frame Check Sequence (FCS, 4 bytes): A CRC-32 computed over all preceding fields. The receiver recalculates this value and discards the frame if it does not match.

Interframe Gap (12 bytes worth of idle time): After each frame, the sender must wait before transmitting the next frame, giving receivers time to process what they received and prepare for more data.

The total minimum Ethernet frame size is 64 bytes (preamble excluded) and the maximum standard size is 1518 bytes. Frames shorter than 64 bytes are called "runt frames" and are discarded, they usually indicate a collision or error.

Physical Addresses (MAC)

Every network interface on an Ethernet link needs a unique identifier so frames can be addressed to specific devices. This identifier is the **Media Access Control address** (MAC address), a 48-bit value traditionally represented as six hexadecimal byte pairs separated by colons: for example, 00:1A:2B:3C:4D:5E.

Historically, MAC addresses were assigned by manufacturers and burned into hardware during production. The first 24 bits (three bytes) identify the manufacturer, registered with the IEEE, while the remaining 24 bits are a unique serial number assigned by that manufacturer. This structure meant that seeing a MAC address told you who made the network card.

Modern systems often use **locally administered addresses** instead of factory-assigned ones for privacy reasons. Operating systems can generate random MAC addresses when connecting to Wi-Fi networks, preventing tracking across different locations based on a fixed hardware identifier. The second least significant bit of the first byte indicates whether an address is globally unique (manufacturer-assigned) or locally administered: if that bit is 1, the address was generated by software rather than assigned by IEEE registration.

MAC addresses operate at link scope only: they are meaningful only on a single physical link. When a frame crosses from one network to another

through a router, the MAC addresses change, the router strips the incoming frame's Ethernet header, makes a routing decision based on the IP address inside and creates a new Ethernet frame with fresh MAC addresses appropriate for the outgoing link. This separation between link-layer addressing (MAC) and network-layer addressing (IP) is fundamental to how the Internet scales.

Collision Detection and Avoidance

Early Ethernet used a shared medium: all devices on a segment connected to the same coaxial cable and any transmission was received by every device. This created a problem: what happens if two devices transmit simultaneously? Their signals collide on the wire, producing garbled data that no one can decode.

Classic Ethernet solved this with **Carrier Sense Multiple Access with Collision Detection** (CSMA/CD). The protocol works as follows:

1. Before transmitting, a device listens to the cable (**carrier sense**). If it detects another transmission in progress, it waits until the channel is idle.
2. If the channel is idle, the device begins transmitting.
3. While transmitting, it continues listening (**collision detection**). If it detects that the signal on the wire does not match what it is sending, indicating another device is also transmitting, it knows a collision has occurred.
4. When a collision is detected, both devices stop transmitting immediately and send a "jam" signal to ensure all devices notice the collision.
5. Each colliding device then waits for a random backoff period before retrying. The backoff time is chosen from an increasing range: after the first collision, wait 0 or 1 slot times; after the second, 0-3 slots; after the third, 0-7 slots; and so on up to a maximum of 1023 slots. This **exponential backoff** reduces the probability that the same two devices collide repeatedly.

CSMA/CD worked well for early Ethernet at 10 Mbps over coaxial cable segments up to 500 meters. But it has fundamental limitations:

- Collisions waste bandwidth because collided frames must be retransmitted

- Performance degrades as more devices share the medium (more potential collisions)
- The collision detection mechanism requires that a device can detect a collision before finishing transmission, which constrains maximum cable length relative to data rate

Modern Ethernet largely eliminates collisions through **switching**. Instead of all devices sharing one cable, each device connects to a dedicated switch port with its own point-to-point link. Since no two devices share the same wire segment, collisions cannot occur (except in rare cases with half-duplex operation). Switched Ethernet operates in full-duplex mode: each device can transmit and receive simultaneously on separate wire pairs, doubling effective bandwidth.

CSMA/CD remains in the Ethernet specification for backward compatibility but is effectively obsolete in modern networks. Wireless networks still face collision problems (a device cannot transmit and listen simultaneously on the same frequency), so Wi-Fi uses **Collision Avoidance** instead of detection: devices reserve the channel before transmitting using Request-to-Send/Clear-to-Send (RTS/CTS) handshakes or by observing when the medium is idle for a random backoff period.

Switching vs Broadcasting

When a device sends an Ethernet frame, how does that frame reach its destination? There are two fundamental modes:

Broadcast: The frame's destination MAC address is set to FF:FF:FF:FF:FF:FF, the broadcast address. Every device on the link receives and processes this frame. Broadcast is essential for certain protocols: when a new device joins a network and needs to discover its configuration (ARP requests, DHCP discovery), it has no way to address specific known devices, so it broadcasts and waits for appropriate responses.

Unicast: The frame's destination MAC address identifies exactly one device. Only that device accepts the frame; all others discard it after checking the destination address. This is the normal mode for most communication.

In early hub-based Ethernet, every transmission was effectively broadcast: a hub is a dumb repeater that copies incoming signals to all ports. Every device

saw every frame and had to filter based on MAC address. This worked but wasted bandwidth and processing capacity.

Modern **switches** are intelligent devices that learn which MAC addresses are reachable through which ports. When a switch receives a frame, it examines the source MAC address and records: "device with this address is reachable via this port." It builds a forwarding table mapping MAC addresses to ports over time.

When forwarding a unicast frame, the switch looks up the destination MAC in its table and sends the frame only out the appropriate port, not all ports. This drastically reduces unnecessary traffic because frames travel only where needed. If the destination MAC is not in the table (perhaps it is a new device), the switch floods the frame to all ports except the one it arrived on, ensuring delivery while learning the correct path for future frames.

Switches also isolate collisions: each port is its own collision domain (or more accurately in modern full-duplex operation, there are no collisions at all). This means multiple pairs of devices can communicate simultaneously through the same switch without interfering with each other.

Why We Need Layers

We have now seen how bits become signals, signals become frames and frames travel across links between devices connected to the same physical medium. But this solves only a narrow problem: communication between directly connected devices on a single link.

Real networks are far more complex. Devices may be separated by many intermediate links, each using different technologies (Ethernet in an office building, fiber optics across a city, undersea cables between continents, Wi-Fi connecting a phone). A frame formatted for one type of link cannot necessarily traverse another type. And devices need to communicate even when they are not on the same physical link at all.

The solution is **layering**: dividing the communication problem into distinct subproblems, each solved by a separate layer that uses services from the layer below and provides services to the layer above. Each layer adds its own header (or encapsulation) around data from the layer above, creating a nested structure:

```
Application data
  + Transport header → segment
    + Network header → packet
      + Link header → frame
        + Physical encoding → bits on wire
```

This layered architecture has several critical benefits:

1. **Separation of concerns**: Each layer solves one class of problems. The link layer handles reliable bit transmission over a single medium; the network layer handles routing across many links; the transport layer handles end-to-end reliability and flow control. Changes in one layer do not necessarily require changes in others.
2. **Technology independence**: Higher layers do not care what physical technology is used below. TCP works identically whether the underlying link is Ethernet, Wi-Fi, cellular, or satellite because each link type provides the same abstraction: delivery of frames between directly connected endpoints.
3. **Incremental development**: New protocols can be added at one layer without redesigning the entire stack. IPv6 was developed to replace IPv4 while leaving transport and application layers largely unchanged.
4. **Reuse**: A single layer's services can be used by many different protocols above it. IP routing serves HTTP, email, file transfer, video streaming and countless other applications without modification.

The Internet protocol suite, sometimes called TCP/IP after its two most famous protocols, uses four conceptual layers:

- **Link layer** (Layer 2): Ethernet, Wi-Fi, PPP; delivers frames between directly connected devices
- **Network/Internet layer** (Layer 3): IP, ICMP; routes packets across many links from source to destination
- **Transport layer** (Layer 4): TCP, UDP; provides end-to-end communication between processes on different hosts
- **Application layer**: HTTP, DNS, TLS, SMTP; implements specific application protocols

Each layer depends only on the interface provided by the layer below, not on how that layer achieves its goals. This abstraction discipline is what makes the Internet both flexible and scalable and it is the organizing principle of everything we will study in this book.

Chapter 4: Ethernet in Detail

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Ethernet History and Design Philosophy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Frame Format Field by Field

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

MAC Address Structure and Assignment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

CSMA/CD: How Classic Ethernet Worked

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Switched Ethernet and Learning Bridges

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

VLANs and Tagged Frames

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Gigabit Ethernet and Beyond

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 5: Layered Architectures

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Abstraction Problem in Large Systems

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Encapsulation: Wrapping Data in Headers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The OSI Model: Seven Layers Explained

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Internet Protocol Stack: Four Layers That Work

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Interface Contracts Between Layers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Why Layering Matters for the Internet

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 6: IPv4, Design and Structure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

From Local Networks to a Network of Networks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The End-to-End Design Philosophy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

IPv4 Address Format and Classes (Historical)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Packet Header Field by Field

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Time to Live and Loop Prevention

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Fragmentation and Reassembly

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

IP as an Unreliable Best-Effort Protocol

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 7: Subnetting, CIDR and Address Allocation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Classful System and Its Failure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Subnet Masks and Network Prefixes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

CIDR Notation and Aggregation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Variable-Length Subnet Masking (VLSM)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Worked Subnetting Examples

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Global Address Allocation Hierarchy

Chapter 8: IPv6, Modern Internet Protocol

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Running Out of Addresses

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

IPv6 Design Goals

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Simplified Header Format

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Address Types and Formats

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Stateless Address Autoconfiguration (SLAAC)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Neighbor Discovery Protocol

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Transition Mechanisms: Dual Stack and Tunneling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 9: ARP and Link-Layer Address Resolution

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Address Mapping Problem

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

ARP Request and Reply Format

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

ARP Cache Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Gratuitous ARP

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

ARP Spoofing Attacks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

IPv6 Neighbor Discovery

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 10: Routing Fundamentals

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

What a Router Does

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Forwarding Tables and Longest Prefix Match

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Default Routes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Routing vs Forwarding: Control Plane and Data Plane

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Static Routing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Dynamic Routing Motivation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 11: Interior Gateway Protocols, RIP, OSPF, IS-IS

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Autonomous Systems Explained

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Distance-Vector Routing: RIP

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Count-to-Infinity Problem

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Link-State Routing: OSPF

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Dijkstra's Algorithm in Practice

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

IS-IS: The Telecom Alternative

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Route Summarization and Hierarchy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 12: Border Gateway Protocol (BGP)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Why We Need an Exterior Gateway Protocol

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Path-Vector Routing Concept

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

BGP Peering and Sessions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

BGP Attributes: AS_PATH, NEXT_HOP, LOCAL_PREF

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Route Selection Algorithm

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Route Hijacking and Security

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

BGP at Internet Scale

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 13: ICMP, Diagnostics and Error Reporting

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Why IP Needs a Signaling Protocol

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

ICMP Message Types and Codes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Echo Request and Reply: How Ping Works

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Time Exceeded: How Traceroute Works

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Destination Unreachable Messages

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

ICMP Rate Limiting and Security

Chapter 14: DHCP, Dynamic Address Assignment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Configuration Problem

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The DORA Process: Discover, Offer, Request, Acknowledge

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

DHCP Message Format and Options

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Lease Renewal and Release

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

DHCP Relay Agents

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

DHCPv6 and SLAAC Interaction

Chapter 15: DNS, The Internet's Naming System

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Why We Need Names Instead of Numbers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Hierarchical Namespace

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Domain Name Structure and Labels

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Record Types: A, AAAA, CNAME, MX, TXT, SRV

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Resolution Process Step by Step

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Recursive Resolvers and Root Servers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

DNS Caching and TTL

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

DNSSEC: Authenticating Responses

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 16: UDP, Simple Transport

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Why We Need a Transport Layer

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Ports as Process Identifiers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The UDP Header Format

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Checksums in UDP

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

When to Use UDP

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

DNS over UDP Walkthrough

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 17: TCP, Reliable Stream Transport (Part 1)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Reliability Problem

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Byte Streams vs Datagrams

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Sequence Numbers and Acknowledgments

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Three-Way Handshake

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Connection Teardown: Four-Way Wave

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The TCP State Machine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Retransmission Timeouts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 18: TCP, Reliable Stream Transport (Part 2)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Flow Control with Sliding Windows

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Advertised Window and Zero Window

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Congestion: Why Networks Slow Down

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Congestion Avoidance Algorithms

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

TCP Reno and AIMD

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Modern Algorithms: CUBIC and BBR

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Nagle's Algorithm and Delayed ACKs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 19: QUIC, Transport for the Modern Web

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Problems with TCP + TLS + HTTP/2

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

QUIC Design Goals

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Running on UDP: Why It Works

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The QUIC Connection Establishment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Multiplexed Streams Without Blocking

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Connection Migration and Resumption

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Security Properties of QUIC

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 20: TLS, Encrypted Communication

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Eavesdropping and Spoofing Problem

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Symmetric vs Asymmetric Cryptography (Briefly)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The TLS Handshake Protocol

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Certificate Chains and Trust Anchors

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Perfect Forward Secrecy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

TLS 1.3: Faster and Safer

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Common TLS Attacks and Mitigations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 21: HTTP Family, From 1.1 to HTTP/3

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Request-Response Model

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

HTTP Methods and Status Codes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Headers, Bodies and Content Negotiation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

HTTP/1.1: Persistent Connections and Limitations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

HTTP/2: Binary Framing and Multiplexing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

HTTP/3: HTTP over QUIC

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

End-to-End HTTP Transaction Walkthrough

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 22: WebSockets and Persistent Connections

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Limitation of Request-Response

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

WebSocket Handshake Protocol

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Frame Format and Control Frames

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

When to Use WebSockets

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Server-Sent Events as an Alternative

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Brief Look at WebRTC Signaling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 23: NAT, Network Address Translation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Address Exhaustion Crisis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Private Address Spaces (RFC 1918)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

How NAT Works: Translation Tables

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Types of NAT: Full Cone, Restricted, Symmetric

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Port Forwarding and DMZ

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Why NAT Breaks End-to-End

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

NAT Traversal Techniques

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 24: Firewalls, Middleboxes and Traffic Control

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

What Is a Firewall

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Packet Filtering Rules

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Stateful Inspection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Deep Packet Inspection (DPI)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Quality of Service Marking

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Rate Limiting and Traffic Shaping

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Middlebox Problem

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 25: Content Delivery Networks and Load Balancers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Problem of Scale

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Load Balancing Fundamentals

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Layer 4 Load Balancing with NAT

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Layer 7 Load Balancing and HTTP

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Content Delivery Networks (CDNs)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Anycast Routing for CDNs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Reverse Proxies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Edge Computing and Serverless at the CDN

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 26: Multicast and Anycast Delivery Models

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Unicast, Broadcast, Multicast, Anycast

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

IP Multicast Fundamentals

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Multicast Routing Protocols

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Multicast in Practice

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Anycast in Depth

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Comparing Delivery Models

Chapter 27: VPNs, Tunneling and Overlay Networks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Need for Private Networks Over Public Infrastructure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Tunneling Fundamentals

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

IPsec, The Internet Protocol Security Suite

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

WireGuard, A Modern VPN Protocol

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Site-to-Site VPNs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Overlay Networks in Data Centers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Comparing VPN Technologies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 28: Operating System Networking Internals

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Kernel as Network Participant

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Socket API: The User-Kernel Interface

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Linux Networking Stack: sk_buff and Netfilter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Interrupts, DMA and NIC Processing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Connection Tracking and State Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Routing Tables and Policy Routing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Memory Pressure and Packet Drops

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 29: Practical Network Investigation Tools

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Philosophy of Network Debugging

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

ping, ICMP Echo and Reachability Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

traceroute, Path Discovery

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

ip and ss, Interface and Socket Inspection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

dig, DNS Investigation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

curl, HTTP Client and Protocol Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

tcpdump and Wireshark, Packet Capture and Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Systematic Debugging Workflow

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 30: Building a User-Space Networking Stack (Educational Implementation)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Why Implement Protocols from Scratch

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Packet Parsing and Serialization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Implementing ARP

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

A Minimal UDP Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Simplified TCP State Machine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Putting It Together: A Minimal Stack Architecture

Chapter 31: Data Center and Cloud Networking

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Scale Challenges in Modern Data Centers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Spine-Leaf Architecture

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Overlay Networking in Data Centers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Kubernetes Networking Model

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Cloud Provider Networking

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

High-Performance Networking in Data Centers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 32: Advanced Topics, SDN, P4, eBPF and Modern Architecture

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Software-Defined Networking (SDN)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

P4, Programming Protocol-Independent Packet Processors

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

eBPF, Programmable Kernel Networking

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

MPLS, Multi-Protocol Label Switching

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Zero Trust Networking

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Observability in Modern Networks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

The Evolving Internet Architecture

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 33: End-to-End Case Studies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Case Study 1: Opening a Website from a Fresh Machine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Case Study 2: Traversing Autonomous Systems via BGP

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Case Study 3: Reaching Cloud Infrastructure Through Load Balancers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Case Study 4: Failure Modes and Recovery

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Chapter 34: Protocol Design Principles and a Unified Mental Model

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

What Makes a Good Protocol

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Trade-offs Everywhere

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

A Unified Mental Model

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Engineering at Internet Scale

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Continuing Evolution

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Conclusion

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

References

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Core IP and Routing RFCs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Link Layer and Local Network RFCs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Transport and Security RFCs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Application Layer RFCs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

IPsec, VPNs and Tunneling RFCs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Multicast and Anycast RFCs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

ICMP and Diagnostics RFCs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.

Additional Authoritative Sources

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/internetprotocolsfromfirstprinciples.