Open Source Networking Textbooks: A Clean Slate

Larry Peterson
Systems Approach
llp@cs.princeton.edu

Bruce Davie
Systems Approach
bsd@systemsapproach.org

ABSTRACT

After three decades of producing networking textbooks, we observe that the Internet has changed enough to warrant a fresh approach to educating the next generation of students and practitioners. We propose a clean-slate refactoring of our networking textbook that aims to give students the necessary perspective to form a big-picture view of networking and the skills to deal with the complexity of modern networks. We treat networking as a set of system design problems and draw on existing artifacts such as Ethernet, HTTP and IP to illustrate how these challenges have been solved in the past, while aiming to equip readers with the perspective and systems thinking skills to shape the networks of the future.

KEYWORDS

Networking Education, System Architecture

1 INTRODUCTION

At last year's SIGCOMM keynote, Alex Snoeren posed a question about the future of networking textbooks. That question was: After 30 years, what would you do differently? The question was premised on the observation that the Internet has become dramatically more complex in the 30 years since *Computer Networks: A Systems Approach* was published in 1996—so much so that it is hard to know the right way to teach a networking course. Bruce's response was consistent with the book's theme through its six editions—that it's important to focus on the timeless principles, while also teaching tools to manage complexity. This exchange set in motion a discussion about the magnitude of changes we've seen over the decades, and led to our undertaking a major refactoring of our textbook for its next edition.

We've decided that it is time for a generational change with our book. Eight months have elapsed since SIGCOMM 2025 and we now have a nearly complete draft of what will be the seventh edition. This paper describes the thinking that went into that clean-slate redesign.

While our initial motivation for a significant refactoring of our book was the need to deal with the greater complexity of the Internet today compared to the one we were describing in 1996, there are other factors that come into play. Notably, there has actually been something of a simplification of the lower layers of the network stack, at least in terms of how many technologies are in widespread use. Ethernet has "won" the competition among many different link layers for wired networks, while there remain a handful of wireless options (flavors of 802.11, 5G/6G, satellite links, etc.). Similarly, the dominant upper layer protocol is HTTP, in contrast to the broader ecosystem of application layer protocols in 1996. We see this consolidation as something that can be leveraged to help readers manage the complexity of modern networks.

The other significant factor casting a shadow over any textbook or educational endeavor is the rise of large language models (LLMs).[1] When large numbers of students turn to LLMs for the answer to any question about networking, it raises a challenge for the textbook author to provide something of value (beyond fodder for the training of the next round of LLMs). Since day one, our books have aimed to provide a sense of perspective and to explain architectural principles. We aim to teach students about system design, not to list networking technologies like an encyclopedia. We use existing technologies as artifacts to illustrate design principles. We believe this approach will continue to provide value in the new LLM-enabled environment.

As with the previous edition of our book, we are developing this book as an open source project (described in [1]). The source files are hosted on GitHub [2], and we solicit input from the community. The book is licensed under a creative commons license (CC BY 4.0 [3]). The draft of the seventh edition is now online [4] and we are soliciting input from the community.

2 FITTING IT ALL IN YOUR HEAD

A challenge posed by the increased complexity of modern networks is the difficulty of holding a mental model of the entire networking stack in one's head. We are reminded of a quote from Jon Postel:

> *[There was] a space that we were exploring and, in the early days, we figured out this consistent path through the space: IP, TCP, and so on. What's been happening over the last few years is that the IETF is filling the rest of the space with every alternative approach, not necessarily any better.*

Certainly we can't hope to teach the next generation of students about the details of every RFC published in the

[1] We prefer to avoid the term AI as being too broad and ill-defined to be useful; LLMs are the technology that has changed the way students consume information, for better or worse.

last 30 years. It would be unhelpful to try. This is what we mean by giving a sense of perspective. Within that massive space of design possibilities and protocols, there are certain topics that stand out because they have out-sized importance in how networks operate, or that teach a particular design principle.

One way we approach the magnitude of the task is to focus on the fundamental problem of building a network. The opening paragraph of our new edition is not greatly different from that of the first edition:

> Suppose you want to build a computer network, one that has the potential to grow to global scale and support applications as diverse as video conferencing, live and on-demand streaming, social media, e-commerce, automated manufacturing, and more. What available technologies would serve as the underlying building blocks, and what kind of software architecture would you design to integrate these building blocks into an effective communication service? Answering this question is the overriding goal of this book—to describe the available building materials and then to show how they can be used to construct a network from the ground up.

The focus on building leads us to teach system design and architectural principles rather than trying to be an encyclopedia of networking technologies. We are not just trying to help students understand the artifacts that exists today: we are trying to equip them to shape the networks of the future.

One important change noted above is that the set of building blocks has solidified. Ethernet should be taken as a given as the underlying network technology. The Internet Protocol (IP) is clearly central to networking to a greater extent than it was in 1996. And HTTP dominates the application protocol space. The central role of these three artifacts presents an opportunity to give the reader a reasonably complete picture of today's "default" network architecture: HTTP at the top of the stack, providing application layer support; IP at the narrow waist, serving as the demarcation point between what runs inside the network and what runs at the edges; and Ethernet anchoring the bottom of the stack, providing a ubiquitous link technology.

3 NEITHER TOP-DOWN NOR BOTTOM-UP

We have argued, since the first edition, that while layering can be a useful tool to understand networks, strict layering is too reductive to allow for a deep understanding of how a networking system operates. So while our prior editions treated link layer issues early in the book and application layer issues near the end, we did this not to provide a "bottom-up" view but to allow us to tackle problems in an orderly fashion. For example, it's easier to explain how a reliable transport protocol works when you have already understood how unreliable datagram delivery works. But issues such as security and resource allocation are not confined to a single layer, and so need their own chapters.

We have embraced this non-layerist view more strongly in the new edition. Many of the "big ideas in networking" that we believe we need to cover don't naturally align with particular layers. Many of the chapters stand on their own, but of course we need some way to organize the topics to help the reader develop their mental model of networking. We don't want readers to have to suspend disbelief if a chapter depends on material that hasn't yet been covered in the book.

One way we accomplish this is to adopt a "2-pass" strategy. In the first pass (corresponding to Part I of the book) we introduce all the foundational concepts need to understand the problem-specific examples described later in the book. These include the basics of network architectures, over-the-wire protocols, service interfaces, encapsulation, and so on. This first pass also discusses network applications (helping the student ground these concepts in something that's likely to be familiar) and network hardware (helping the student ground the concepts in something tangible). The second pass (corresponding to Parts II and III of the book) then show how these general concepts apply to specific subsystems, both *inside the network* (Part II) and *at the edge of the network* (Part III). This latter split helps to separate concerns, with IP's best-effort packet delivery service defining the boundary between the two.

We've tried to minimize the dependency between Parts II and III—they are designed to be read in either order—but it turns out there is a rich set of interdependencies within each part. This is because of our view that each part describes a multi-faceted ecosystem, with many interrelated ideas with subtle dependencies. We include generous cross-references to adjacent topics that readers can follow, representing these interweaving threads across the software stack. Following these threads is not a requirement for understanding each topic, but it does help the reader appreciate that everything is connected (just not always in a strictly top-down order).

One consequence of each chapter tackling a fundamental challenge in networking, rather than the next layer in a prescribed stack, is that we are forced to treat each challenge as an independent system design problem, one that comes with its own set of requirements, assumptions, and design choices. This means we begin each chapter with a focus on architectural questions, reinforcing our sense that the book not only covers the important concepts and practices in networking, but also shows—by example—how computer systems are

designed and built in practice. We have decided to emphasize the design process (while grounding the discussion with coverage of today's artifacts) because we believe that mastering the design process will be a requirement for anyone who wants to take advantage of AI (however it is defined) rather than be replaced by AI. We believe this will be true whether you work on networking subsystems, networked applications, or higher-level services.

4 PUTTING IT ALL TOGETHER

To recap, our approach focuses on design principles rather than encyclopedic coverage of specific technologies; we anchor the discussion around three central artifacts (Ethernet, IP, and HTTP); and we tackle topics that are central to the problem of building a network in a non-layerist manner. This has led to the following organization (see Appendix for full table of contents):

- **Part I: Foundation**
  Provides a big picture view of networking, explaining the key challenges such as scale and performance. Introduces fundamental concepts of networking such as protocols and encapsulation. Examines some representative applications, with a focus on HTTP as the protocol underpinning many modern apps. Explains the basics of packet switching and statistical multiplexing with a focus on Ethernet as the canonical example of modern packet-switched networks.
- **Part II: Inside The Network**
  Each chapter tackles one of the problems that needs to be solved to build a global network of packet switches, starting with how to calculate and distribute routes. Other chapters include: mediating shared access (wireless networks); federating networks (internetworking and IP); allocating resources; virtual networks; and network operations.
- **Part III: The Edge of the Network**
  We move our focus out to the systems, applications and protocols running at the edges of the network. We start with a discussion of naming, including both DNS and application level names. We cover reliable delivery (TCP and QUIC); remote procedure call; congestion control; secure channels (TLS); overlays and CDNs; and real-time applications.

There are some recurring themes that show up repeatedly throughout the book rather than forming their own chapter. Security is a notable example. While the chapter on secure channels covers quite a few security topics (encryption, authentication, public keys, certificate hierarchies, etc.), it does so in the service of solving a problem: how to secure the traffic between two endpoints (including a discussion of what it means to "secure" a channel). Security is an aspect of networking that should not be treated as something added at the end, and so it shows up throughout the book, e.g., in wireless networks, in the security of routing, in discussion of VPNs, etc. Similarly, many topics have a role to play in datacenter networking (SDN, congestion control, RPC, virtualization, etc.) and so we tackle them as they arise rather than calling out datacenters in their own chapter.

Another common theme is the symbiotic relationship between the Internet and the cloud. It's straightforward to explain how the cloud is built on top of an Internet-based network substrate, but it's equally important to take into account—and help students appreciate—how much the Internet has changed in response to the existence of the cloud. This includes both the feature velocity that cloud software enables (which impacts interoperability and operations) and the global availability of computing resources (which impacts function placement). It seems important that we shift the networking mindset away from "fixed protocol stacks" and closer to "agile software ecosystems".

This relates to the earlier point about teaching students how to shape the networks of the future, not just how they work today. The ready availability of cloud-based resources and their central role in networking means it is much more feasible for students to experiment with new networking concepts than it was in an era where most networking functionality lived in proprietary routers operated by ISPs.

Given our focus on systems design principles, which don't necessarily stand out at the level of a section heading, we have decided to add a new design element to the book. These highlighted paragraphs will serve to draw attention to important principles of system design that generalize beyond the specific artifact under discussion. For example, the separation of policy and mechanism is a well-known systems principle that happens to show up when we discuss queuing algorithms. We seize the opportunity to talk more broadly about this principle at that point in the book and use a visual design element to draw attention to it.

5 CONCLUSIONS

Textbooks are just one part of a networking course, but they can provide a coherent perspective on the subject, helping readers to connect the dots. A big picture view helps the student make sense of all the details and see how various technologies fit to from the whole. It enables them to know what questions to ask, not just telling them how some bit of technology works today, but setting them up to shape the networks of the future.

It is important that students don't perceive networking as a set of fixed technologies that they need to learn, but see it as an evolving field. Building the networks for data

centers to train LLMs, for example, might be where much of the excitement is today, but the center of attention is sure to move again in the future just as it has done in the past. Our job is to give them the understanding of system design principles and the intellectual framework to manage complexity that will set them up for the future of networking, whatever it may be.

ACKNOWLEDGMENTS

Thanks to all those who have contributed to our books over the decades, including all those who submitted contributions since we started using an open source approach. We also thank the reviewers of this paper for their constructive comments.

REFERENCES

[1] Peterson, L. and Davie, B. *Open Source Networking Education: A Systems Approach.* In SIGCOMM Education Workshop. https://systemsapproach.org/2021/01/13/open-source-network-education-a-systems-approach August 2020

[2] *Computer Networks: A Systems Approach* code repository. https://github.com/SystemsApproach/7E

[3] Creative Commons. CC BY 4.0 License. https://creativecommons.org/licenses/by/4.0/

[4] *Computer Networks: A Systems Approach* 7th edition draft. https://7E.systemsapproach.org

APPENDIX: DRAFT TOC

Part I: Foundation

Chapter 1: Introduction
- 1.1 Requirements
- 1.2 Network Architecture
- 1.3 Network Protocols
- 1.4 Resource Sharing
- 1.5 Performance

Chapter 2: Network Applications
- 2.1 Anatomy of an Application
- 2.2 World Wide Web
- 2.3 Electronic Mail
- 2.4 Adaptive Streaming

Chapter 3: Network Technology
- 3.1 Communication Links
- 3.2 Packet Switches
- 3.3 Circuit Switching

Part II: Inside the Network

Chapter 4: Routing
- 4.1 Design Issues
- 4.2 Spanning Tree Protocol
- 4.3 Link-State Routing
- 4.4 Distance Vector Routing
- 4.5 Routing in Datacenters

Chapter 5: Mediating Shared Access
- 5.1 Design Issues
- 5.2 Managing Spectrum
- 5.3 Wi-Fi
- 5.4 Mobile Cellular Network
- 5.4 Passive Optical Network

Chapter 6: Federating Networks
- 6.1 Design Issues
- 6.2 Heterogeneity
- 6.3 Scale

Chapter 7: Interdomain Routing
- 7.1 Design Issues
- 7.2 Sharing Routes and Routing Policy
- 7.3 Securing Route Information

Chapter 8: Allocating Capacity
- 8.1 Design Issues
- 8.2 Packet Schedulers
- 8.3 Active Queue Management
- 8.4 Datacenter Networks
- 8.5 Traffic Engineering

Chapter 9: Virtual Networks
- 9.1 Design Issues
- 9.2 Virtual LANs (VLANs)
- 9.3 Virtual Private Networks (VPNs)
- 9.4 Network Virtualization for Datacenters

Chapter 10: Operating Networks
- 10.1 Design Issues
- 10.2 Configuration Management
- 10.3 Network Telemetry
- 10.4 Feature Velocity

Part III: Edge of the Network

Chapter 11: Naming Systems
- 11.1 Design Issues
- 11.2 Domain Naming System
- 11.3 Application Naming Schemes

Chapter 12: Reliable Byte Stream
- 12.1 Design Issues
- 12.2 Segments
- 12.3 Connection Establishment
- 12.4 Sliding Window
- 12.5 Triggering Transmission
- 12.6 Adaptive Retransmission
- 12.7 Record Boundaries
- 12.8 TCP Extensions

Chapter 13: Congestion Control
- 13.1 Design Issues
- 13.2 Loss-Based Algorithms
- 13.3 Delay-Based Algorithms
- 13.4 Domain-Specific Variants

Chapter 14: Secure Channels
- 14.1 Design Issues
- 14.2 Cryptographic Building Blocks
- 14.3 Key Distribution
- 14.4 Session Establishment
- 14.5 Record Protocol
- 14.6 Performance Issues

Chapter 15: Message Transactions
- 15.1 Design Issues
- 15.2 Remote Procedure Call
- 15.3 Optimizing RPC
- 15.4 Remote DMA
- 15.5 Optimizing Ethernet for RDMA
- 15.6 Argument Marshalling

Chapter 16: Overlay Networks
- 16.1 Design Issues
- 16.2 Content Distribution Networks
- 16.3 Teleconferencing
- 16.4 Peer-to-Peer Networks

Chapter 17: Real-time Streaming
- 17.1 Design Issues
- 17.2 Coding and Compression
- 17.3 Transport Issues
- 17.4 Session Issues
- 17.5 Scaling Real-time Applications