

This page intentionally left blank

To Julie and our three precious
ones—Chris, Charlie, and Nina
JFK

A big THANKS to my professors, colleagues,
and students all over the world.
KWR

This page intentionally left blank

Preface

Welcome to the sixth edition of *Computer Networking: A Top-Down Approach*. Since the publication of the first edition 12 years ago, our book has been adopted for use at many hundreds of colleges and universities, translated into 14 languages, and used by over one hundred thousand students and practitioners worldwide. We've heard from many of these readers and have been overwhelmed by the positive response.

What's New in the Sixth Edition?

We think one important reason for this success has been that our book continues to offer a fresh and timely approach to computer networking instruction. We've made changes in this sixth edition, but we've also kept unchanged what we believe (and the instructors and students who have used our book have confirmed) to be the most important aspects of this book: its top-down approach, its focus on the Internet and a modern treatment of computer networking, its attention to both principles and practice, and its accessible style and approach toward learning about computer networking. Nevertheless, the sixth edition has been revised and updated substantially:

- The Companion Web site has been significantly expanded and enriched to include VideoNotes and interactive exercises, as discussed later in this Preface.
- In Chapter 1, the treatment of access networks has been modernized, and the description of the Internet ISP ecosystem has been substantially revised, accounting for the recent emergence of content provider networks, such as Google's. The presentation of packet switching and circuit switching has also been reorganized, providing a more topical rather than historical orientation.
- In Chapter 2, Python has replaced Java for the presentation of socket programming. While still explicitly exposing the key ideas behind the socket API, Python code is easier to understand for the novice programmer. Moreover, unlike Java, Python provides access to raw sockets, enabling students to build a larger variety of network applications. Java-based socket programming labs have been replaced with corresponding Python labs, and a new Python-based ICMP Ping lab has been added. As always, when material is retired from the book, such as Java-based socket programming material, it remains available on the book's Companion Web site (see following text).
- In Chapter 3, the presentation of one of the reliable data transfer protocols has been simplified and a new sidebar on TCP splitting, commonly used to optimize the performance of cloud services, has been added.
- In Chapter 4, the section on router architectures has been significantly updated, reflecting recent developments and practices in the field. Several new integrative sidebars involving DNS, BGP, and OSPF are included.

- Chapter 5 has been reorganized and streamlined, accounting for the ubiquity of switched Ethernet in local area networks and the consequent increased use of Ethernet in point-to-point scenarios. Also, a new section on data center networking has been added.
- Chapter 6 has been updated to reflect recent advances in wireless networks, particularly cellular data networks and 4G services and architecture.
- Chapter 7, which focuses on multimedia networking, has gone through a major revision. The chapter now includes an in-depth discussion of streaming video, including adaptive streaming, and an entirely new and modernized discussion of CDNs. A newly added section describes the Netflix, YouTube, and Kankan video streaming systems. The material that has been removed to make way for these new topics is still available on the Companion Web site.
- Chapter 8 now contains an expanded discussion on endpoint authentication.
- Significant new material involving end-of-chapter problems has been added. As with all previous editions, homework problems have been revised, added, and removed.

Audience

This textbook is for a first course on computer networking. It can be used in both computer science and electrical engineering departments. In terms of programming languages, the book assumes only that the student has experience with C, C++, Java, or Python (and even then only in a few places). Although this book is more precise and analytical than many other introductory computer networking texts, it rarely uses any mathematical concepts that are not taught in high school. We have made a deliberate effort to avoid using any advanced calculus, probability, or stochastic process concepts (although we've included some homework problems for students with this advanced background). The book is therefore appropriate for undergraduate courses and for first-year graduate courses. It should also be useful to practitioners in the telecommunications industry.

What Is Unique about This Textbook?

The subject of computer networking is enormously complex, involving many concepts, protocols, and technologies that are woven together in an intricate manner. To cope with this scope and complexity, many computer networking texts are often organized around the “layers” of a network architecture. With a layered organization, students can see through the complexity of computer networking—they learn about the distinct concepts and protocols in one part of the architecture while seeing the big picture of how all parts fit together. From a pedagogical perspective, our personal experience has been that such a layered approach

indeed works well. Nevertheless, we have found that the traditional approach of teaching—bottom up; that is, from the physical layer towards the application layer—is not the best approach for a modern course on computer networking.

A Top-Down Approach

Our book broke new ground 12 years ago by treating networking in a top-down manner—that is, by beginning at the application layer and working its way down toward the physical layer. The feedback we received from teachers and students alike have confirmed that this top-down approach has many advantages and does indeed work well pedagogically. First, it places emphasis on the application layer (a “high growth area” in networking). Indeed, many of the recent revolutions in computer networking—including the Web, peer-to-peer file sharing, and media streaming—have taken place at the application layer. An early emphasis on application-layer issues differs from the approaches taken in most other texts, which have only a small amount of material on network applications, their requirements, application-layer paradigms (e.g., client-server and peer-to-peer), and application programming interfaces. Second, our experience as instructors (and that of many instructors who have used this text) has been that teaching networking applications near the beginning of the course is a powerful motivational tool. Students are thrilled to learn about how networking applications work—applications such as e-mail and the Web, which most students use on a daily basis. Once a student understands the applications, the student can then understand the network services needed to support these applications. The student can then, in turn, examine the various ways in which such services might be provided and implemented in the lower layers. Covering applications early thus provides motivation for the remainder of the text.

Third, a top-down approach enables instructors to introduce network application development at an early stage. Students not only see how popular applications and protocols work, but also learn how easy it is to create their own network applications and application-level protocols. With the top-down approach, students get early exposure to the notions of socket programming, service models, and protocols—important concepts that resurface in all subsequent layers. By providing socket programming examples in Python, we highlight the central ideas without confusing students with complex code. Undergraduates in electrical engineering and computer science should not have difficulty following the Python code.

An Internet Focus

Although we dropped the phrase “Featuring the Internet” from the title of this book with the fourth edition, this doesn’t mean that we dropped our focus on the Internet! Indeed, nothing could be further from the case! Instead, since the Internet has become so pervasive, we felt that any networking textbook must have a significant

focus on the Internet, and thus this phrase was somewhat unnecessary. We continue to use the Internet’s architecture and protocols as primary vehicles for studying fundamental computer networking concepts. Of course, we also include concepts and protocols from other network architectures. But the spotlight is clearly on the Internet, a fact reflected in our organizing the book around the Internet’s five-layer architecture: the application, transport, network, link, and physical layers.

Another benefit of spotlighting the Internet is that most computer science and electrical engineering students are eager to learn about the Internet and its protocols. They know that the Internet has been a revolutionary and disruptive technology and can see that it is profoundly changing our world. Given the enormous relevance of the Internet, students are naturally curious about what is “under the hood.” Thus, it is easy for an instructor to get students excited about basic principles when using the Internet as the guiding focus.

Teaching Networking Principles

Two of the unique features of the book—its top-down approach and its focus on the Internet—have appeared in the titles of our book. If we could have squeezed a *third* phrase into the subtitle, it would have contained the word *principles*. The field of networking is now mature enough that a number of fundamentally important issues can be identified. For example, in the transport layer, the fundamental issues include reliable communication over an unreliable network layer, connection establishment/teardown and handshaking, congestion and flow control, and multiplexing. Two fundamentally important network-layer issues are determining “good” paths between two routers and interconnecting a large number of heterogeneous networks. In the link layer, a fundamental problem is sharing a multiple access channel. In network security, techniques for providing confidentiality, authentication, and message integrity are all based on cryptographic fundamentals. This text identifies fundamental networking issues and studies approaches towards addressing these issues. The student learning these principles will gain knowledge with a long “shelf life”—long after today’s network standards and protocols have become obsolete, the principles they embody will remain important and relevant. We believe that the combination of using the Internet to get the student’s foot in the door and then emphasizing fundamental issues and solution approaches will allow the student to quickly understand just about any networking technology.

The Web Site

Each new copy of this textbook includes six months of access to a Companion Web site for all book readers at <http://www.pearsonhighered.com/kurose-ross>, which includes:

- *Interactive learning material.* An important new component of the sixth edition is the significantly expanded online and interactive learning material. The book’s Companion Web site now contains VideoNotes—video presentations of

important topics throughout the book done by the authors, as well as walk-throughs of solutions to problems similar to those at the end of the chapter. We've also added Interactive Exercises that can create (and present solutions for) problems similar to selected end-of-chapter problems. Since students can generate (and view solutions for) an unlimited number of similar problem instances, they can work until the material is truly mastered. We've seeded the Web site with VideoNotes and online problems for chapters 1 through 5 and will continue to actively add and update this material over time. As in earlier editions, the Web site contains the interactive Java applets that animate many key networking concepts. The site also has interactive quizzes that permit students to check their basic understanding of the subject matter. Professors can integrate these interactive features into their lectures or use them as mini labs.

- *Additional technical material.* As we have added new material in each edition of our book, we've had to remove coverage of some existing topics to keep the book at manageable length. For example, to make room for the new material in this edition, we've removed material on ATM networks and the RTSP protocol for multimedia. Material that appeared in earlier editions of the text is still of interest, and can be found on the book's Web site.
- *Programming assignments.* The Web site also provides a number of detailed programming assignments, which include building a multithreaded Web server, building an e-mail client with a GUI interface, programming the sender and receiver sides of a reliable data transport protocol, programming a distributed routing algorithm, and more.
- *Wireshark labs.* One's understanding of network protocols can be greatly deepened by seeing them in action. The Web site provides numerous Wireshark assignments that enable students to actually observe the sequence of messages exchanged between two protocol entities. The Web site includes separate Wireshark labs on HTTP, DNS, TCP, UDP, IP, ICMP, Ethernet, ARP, WiFi, SSL, and on tracing all protocols involved in satisfying a request to fetch a web page. We'll continue to add new labs over time.

Pedagogical Features

We have each been teaching computer networking for more than 20 years. Together, we bring more than 50 years of teaching experience to this text, during which time we have taught many thousands of students. We have also been active researchers in computer networking during this time. (In fact, Jim and Keith first met each other as master's students in a computer networking course taught by Mischa Schwartz in 1979 at Columbia University.) We think all this gives us a good perspective on where networking has been and where it is likely to go in the future. Nevertheless, we have resisted temptations to bias the material in this book

towards our own pet research projects. We figure you can visit our personal Web sites if you are interested in our research. Thus, this book is about modern computer networking—it is about contemporary protocols and technologies as well as the underlying principles behind these protocols and technologies. We also believe that learning (and teaching!) about networking can be fun. A sense of humor, use of analogies, and real-world examples in this book will hopefully make this material more fun.

Supplements for Instructors

We provide a complete supplements package to aid instructors in teaching this course. This material can be accessed from Pearson's Instructor Resource Center (<http://www.pearsonhighered.com/irc>). Visit the Instructor Resource Center or send e-mail to computing@aw.com for information about accessing these instructor's supplements.

- *PowerPoint® slides.* We provide PowerPoint slides for all nine chapters. The slides have been completely updated with this sixth edition. The slides cover each chapter in detail. They use graphics and animations (rather than relying only on monotonous text bullets) to make the slides interesting and visually appealing. We provide the original PowerPoint slides so you can customize them to best suit your own teaching needs. Some of these slides have been contributed by other instructors who have taught from our book.
- *Homework solutions.* We provide a solutions manual for the homework problems in the text, programming assignments, and Wireshark labs. As noted earlier, we've introduced many new homework problems in the first five chapters of the book.

Chapter Dependencies

The first chapter of this text presents a self-contained overview of computer networking. Introducing many key concepts and terminology, this chapter sets the stage for the rest of the book. All of the other chapters directly depend on this first chapter. After completing Chapter 1, we recommend instructors cover Chapters 2 through 5 in sequence, following our top-down philosophy. Each of these five chapters leverages material from the preceding chapters. After completing the first five chapters, the instructor has quite a bit of flexibility. There are no interdependencies among the last four chapters, so they can be taught in any order. However, each of the last four chapters depends on the material in the first five chapters. Many instructors first teach the first five chapters and then teach one of the last four chapters for “dessert.”

One Final Note: We'd Love to Hear from You

We encourage students and instructors to e-mail us with any comments they might have about our book. It's been wonderful for us to hear from so many instructors and students from around the world about our first four editions. We've incorporated many of these suggestions into later editions of the book. We also encourage instructors to send us new homework problems (and solutions) that would complement the current homework problems. We'll post these on the instructor-only portion of the Web site. We also encourage instructors and students to create new Java applets that illustrate the concepts and protocols in this book. If you have an applet that you think would be appropriate for this text, please submit it to us. If the applet (including notation and terminology) is appropriate, we'll be happy to include it on the text's Web site, with an appropriate reference to the applet's authors.

So, as the saying goes, "Keep those cards and letters coming!" Seriously, please *do* continue to send us interesting URLs, point out typos, disagree with any of our claims, and tell us what works and what doesn't work. Tell us what you think should or shouldn't be included in the next edition. Send your e-mail to kurose@cs.umass.edu and ross@poly.edu.

Acknowledgments

Since we began writing this book in 1996, many people have given us invaluable help and have been influential in shaping our thoughts on how to best organize and teach a networking course. We want to say A BIG THANKS to everyone who has helped us from the earliest first drafts of this book, up to this fifth edition. We are also *very* thankful to the many hundreds of readers from around the world—students, faculty, practitioners—who have sent us thoughts and comments on earlier editions of the book and suggestions for future editions of the book. Special thanks go out to:

Al Aho (Columbia University)
 Hisham Al-Mubaid (University of Houston-Clear Lake)
 Pratima Akkunoor (Arizona State University)
 Paul Amer (University of Delaware)
 Shamiul Azom (Arizona State University)
 Lichun Bao (University of California at Irvine)
 Paul Barford (University of Wisconsin)
 Bobby Bhattacharjee (University of Maryland)
 Steven Bellovin (Columbia University)
 Pravin Bhagwat (Wibhu)
 Supratik Bhattacharyya (previously at Sprint)
 Ernst Biersack (Eurécom Institute)

Shahid Bokhari (University of Engineering & Technology, Lahore)
Jean Bolot (Technicolor Research)
Daniel Brushteyn (former University of Pennsylvania student)
Ken Calvert (University of Kentucky)
Evandro Cantu (Federal University of Santa Catarina)
Jeff Case (SNMP Research International)
Jeff Chaltas (Sprint)
Vinton Cerf (Google)
Byung Kyu Choi (Michigan Technological University)
Bram Cohen (BitTorrent, Inc.)
Constantine Coutras (Pace University)
John Daigle (University of Mississippi)
Edmundo A. de Souza e Silva (Federal University of Rio de Janeiro)
Philippe Decuetos (Eurécom Institute)
Christophe Diot (Technicolor Research)
Prithula Dhunghel (Akamai)
Deborah Estrin (University of California, Los Angeles)
Michalis Faloutsos (University of California at Riverside)
Wu-chi Feng (Oregon Graduate Institute)
Sally Floyd (ICIR, University of California at Berkeley)
Paul Francis (Max Planck Institute)
Lixin Gao (University of Massachusetts)
JJ Garcia-Luna-Aceves (University of California at Santa Cruz)
Mario Gerla (University of California at Los Angeles)
David Goodman (NYU-Poly)
Yang Guo (Alcatel/Lucent Bell Labs)
Tim Griffin (Cambridge University)
Max Hailperin (Gustavus Adolphus College)
Bruce Harvey (Florida A&M University, Florida State University)
Carl Hauser (Washington State University)
Rachelle Heller (George Washington University)
Phillipp Hoschka (INRIA/W3C)
Wen Hsin (Park University)
Albert Huang (former University of Pennsylvania student)
Cheng Huang (Microsoft Research)
Esther A. Hughes (Virginia Commonwealth University)
Van Jacobson (Xerox PARC)
Pinak Jain (former NYU-Poly student)
Jobin James (University of California at Riverside)
Sugih Jamin (University of Michigan)
Shivkumar Kalyanaraman (IBM Research, India)
Jussi Kangasharju (University of Helsinki)
Sneha Kasera (University of Utah)
Parviz Kermani (formerly of IBM Research)

Hyojin Kim (former University of Pennsylvania student)
 Leonard Kleinrock (University of California at Los Angeles)
 David Kotz (Dartmouth College)
 Beshan Kulapala (Arizona State University)
 Rakesh Kumar (Bloomberg)
 Miguel A. Labrador (University of South Florida)
 Simon Lam (University of Texas)
 Steve Lai (Ohio State University)
 Tom LaPorta (Penn State University)
 Tim-Berners Lee (World Wide Web Consortium)
 Arnaud Legout (INRIA)
 Lee Leitner (Drexel University)
 Brian Levine (University of Massachusetts)
 Chunchun Li (former NYU-Poly student)
 Yong Liu (NYU-Poly)
 William Liang (former University of Pennsylvania student)
 Willis Marti (Texas A&M University)
 Nick McKeown (Stanford University)
 Josh McKinzie (Park University)
 Deep Medhi (University of Missouri, Kansas City)
 Bob Metcalfe (International Data Group)
 Sue Moon (KAIST)
 Jenni Moyer (Comcast)
 Erich Nahum (IBM Research)
 Christos Papadopoulos (Colorado State University)
 Craig Partridge (BBN Technologies)
 Radia Perlman (Intel)
 Jitendra Padhye (Microsoft Research)
 Vern Paxson (University of California at Berkeley)
 Kevin Phillips (Sprint)
 George Polyzos (Athens University of Economics and Business)
 Sriram Rajagopalan (Arizona State University)
 Ramachandran Ramjee (Microsoft Research)
 Ken Reek (Rochester Institute of Technology)
 Martin Reisslein (Arizona State University)
 Jennifer Rexford (Princeton University)
 Leon Reznik (Rochester Institute of Technology)
 Pablo Rodriguez (Telefonica)
 Sumit Roy (University of Washington)
 Avi Rubin (Johns Hopkins University)
 Dan Rubenstein (Columbia University)
 Douglas Salane (John Jay College)
 Despina Saporita (Cisco Systems)
 John Schanz (Comcast)

Henning Schulzrinne (Columbia University)
 Mischa Schwartz (Columbia University)
 Ardash Sethi (University of Delaware)
 Harish Sethu (Drexel University)
 K. Sam Shanmugan (University of Kansas)
 Prashant Shenoy (University of Massachusetts)
 Clay Shields (Georgetown University)
 Subin Shrestha (University of Pennsylvania)
 Bojie Shu (former NYU-Poly student)
 Mihail L. Sichitiu (NC State University)
 Peter Steenkiste (Carnegie Mellon University)
 Tatsuya Suda (University of California at Irvine)
 Kin Sun Tam (State University of New York at Albany)
 Don Towsley (University of Massachusetts)
 David Turner (California State University, San Bernardino)
 Nitin Vaidya (University of Illinois)
 Michele Weigle (Clemson University)
 David Wetherall (University of Washington)
 Ira Winston (University of Pennsylvania)
 Di Wu (Sun Yat-sen University)
 Shirley Wynn (NYU-Poly)
 Raj Yavatkar (Intel)
 Yechiam Yemini (Columbia University)
 Ming Yu (State University of New York at Binghamton)
 Ellen Zegura (Georgia Institute of Technology)
 Honggang Zhang (Suffolk University)
 Hui Zhang (Carnegie Mellon University)
 Lixia Zhang (University of California at Los Angeles)
 Meng Zhang (former NYU-Poly student)
 Shuchun Zhang (former University of Pennsylvania student)
 Xiaodong Zhang (Ohio State University)
 ZhiLi Zhang (University of Minnesota)
 Phil Zimmermann (independent consultant)
 Cliff C. Zou (University of Central Florida)

We also want to thank the entire Addison-Wesley team—in particular, Michael Hirsch, Marilyn Lloyd, and Emma Snider—who have done an absolutely outstanding job on this sixth edition (and who have put up with two very finicky authors who seem congenitally unable to meet deadlines!). Thanks also to our artists, Janet Theurer and Patrice Rossi Calkin, for their work on the beautiful figures in this book, and to Andrea Stefanowicz and her team at PreMediaGlobal for their wonderful production work on this edition. Finally, a most special thanks go to Michael Hirsch, our editor at Addison-Wesley, and Susan Hartman, our former editor at Addison-Wesley. This book would not be what it is (and may well not have been at all) without their graceful management, constant encouragement, nearly infinite patience, good humor, and perseverance.

Table of Contents

Chapter 1	Computer Networks and the Internet	1
1.1	What Is the Internet?	2
1.1.1	A Nuts-and-Bolts Description	2
1.1.2	A Services Description	5
1.1.3	What Is a Protocol?	7
1.2	The Network Edge	9
1.2.1	Access Networks	12
1.2.2	Physical Media	18
1.3	The Network Core	22
1.3.1	Packet Switching	22
1.3.2	Circuit Switching	27
1.3.3	A Network of Networks	32
1.4	Delay, Loss, and Throughput in Packet-Switched Networks	35
1.4.1	Overview of Delay in Packet-Switched Networks	35
1.4.2	Queuing Delay and Packet Loss	39
1.4.3	End-to-End Delay	42
1.4.4	Throughput in Computer Networks	44
1.5	Protocol Layers and Their Service Models	47
1.5.1	Layered Architecture	47
1.5.2	Encapsulation	53
1.6	Networks Under Attack	55
1.7	History of Computer Networking and the Internet	60
1.7.1	The Development of Packet Switching: 1961–1972	60
1.7.2	Proprietary Networks and Internetworking: 1972–1980	62
1.7.3	A Proliferation of Networks: 1980–1990	63
1.7.4	The Internet Explosion: The 1990s	64
1.7.5	The New Millennium	65
1.8	Summary	66
	Homework Problems and Questions	68
	Wireshark Lab	78
	Interview: Leonard Kleinrock	80

indeed works well. Nevertheless, we have found that the traditional approach of teaching—bottom up; that is, from the physical layer towards the application layer—is not the best approach for a modern course on computer networking.

A Top-Down Approach

Our book broke new ground 12 years ago by treating networking in a top-down manner—that is, by beginning at the application layer and working its way down toward the physical layer. The feedback we received from teachers and students alike have confirmed that this top-down approach has many advantages and does indeed work well pedagogically. First, it places emphasis on the application layer (a “high growth area” in networking). Indeed, many of the recent revolutions in computer networking—including the Web, peer-to-peer file sharing, and media streaming—have taken place at the application layer. An early emphasis on application-layer issues differs from the approaches taken in most other texts, which have only a small amount of material on network applications, their requirements, application-layer paradigms (e.g., client-server and peer-to-peer), and application programming interfaces. Second, our experience as instructors (and that of many instructors who have used this text) has been that teaching networking applications near the beginning of the course is a powerful motivational tool. Students are thrilled to learn about how networking applications work—applications such as e-mail and the Web, which most students use on a daily basis. Once a student understands the applications, the student can then understand the network services needed to support these applications. The student can then, in turn, examine the various ways in which such services might be provided and implemented in the lower layers. Covering applications early thus provides motivation for the remainder of the text.

Third, a top-down approach enables instructors to introduce network application development at an early stage. Students not only see how popular applications and protocols work, but also learn how easy it is to create their own network applications and application-level protocols. With the top-down approach, students get early exposure to the notions of socket programming, service models, and protocols—important concepts that resurface in all subsequent layers. By providing socket programming examples in Python, we highlight the central ideas without confusing students with complex code. Undergraduates in electrical engineering and computer science should not have difficulty following the Python code.

An Internet Focus

Although we dropped the phrase “Featuring the Internet” from the title of this book with the fourth edition, this doesn’t mean that we dropped our focus on the Internet! Indeed, nothing could be further from the case! Instead, since the Internet has become so pervasive, we felt that any networking textbook must have a significant

focus on the Internet, and thus this phrase was somewhat unnecessary. We continue to use the Internet's architecture and protocols as primary vehicles for studying fundamental computer networking concepts. Of course, we also include concepts and protocols from other network architectures. But the spotlight is clearly on the Internet, a fact reflected in our organizing the book around the Internet's five-layer architecture: the application, transport, network, link, and physical layers.

Another benefit of spotlighting the Internet is that most computer science and electrical engineering students are eager to learn about the Internet and its protocols. They know that the Internet has been a revolutionary and disruptive technology and can see that it is profoundly changing our world. Given the enormous relevance of the Internet, students are naturally curious about what is “under the hood.” Thus, it is easy for an instructor to get students excited about basic principles when using the Internet as the guiding focus.

Teaching Networking Principles

Two of the unique features of the book—its top-down approach and its focus on the Internet—have appeared in the titles of our book. If we could have squeezed a *third* phrase into the subtitle, it would have contained the word *principles*. The field of networking is now mature enough that a number of fundamentally important issues can be identified. For example, in the transport layer, the fundamental issues include reliable communication over an unreliable network layer, connection establishment/teardown and handshaking, congestion and flow control, and multiplexing. Two fundamentally important network-layer issues are determining “good” paths between two routers and interconnecting a large number of heterogeneous networks. In the link layer, a fundamental problem is sharing a multiple access channel. In network security, techniques for providing confidentiality, authentication, and message integrity are all based on cryptographic fundamentals. This text identifies fundamental networking issues and studies approaches towards addressing these issues. The student learning these principles will gain knowledge with a long “shelf life”—long after today's network standards and protocols have become obsolete, the principles they embody will remain important and relevant. We believe that the combination of using the Internet to get the student's foot in the door and then emphasizing fundamental issues and solution approaches will allow the student to quickly understand just about any networking technology.

The Web Site

Each new copy of this textbook includes six months of access to a Companion Web site for all book readers at <http://www.pearsonhighered.com/kurose-ross>, which includes:

- *Interactive learning material.* An important new component of the sixth edition is the significantly expanded online and interactive learning material. The book's Companion Web site now contains VideoNotes—video presentations of

important topics throughout the book done by the authors, as well as walk-throughs of solutions to problems similar to those at the end of the chapter. We've also added Interactive Exercises that can create (and present solutions for) problems similar to selected end-of-chapter problems. Since students can generate (and view solutions for) an unlimited number of similar problem instances, they can work until the material is truly mastered. We've seeded the Web site with VideoNotes and online problems for chapters 1 through 5 and will continue to actively add and update this material over time. As in earlier editions, the Web site contains the interactive Java applets that animate many key networking concepts. The site also has interactive quizzes that permit students to check their basic understanding of the subject matter. Professors can integrate these interactive features into their lectures or use them as mini labs.

- *Additional technical material.* As we have added new material in each edition of our book, we've had to remove coverage of some existing topics to keep the book at manageable length. For example, to make room for the new material in this edition, we've removed material on ATM networks and the RTSP protocol for multimedia. Material that appeared in earlier editions of the text is still of interest, and can be found on the book's Web site.
- *Programming assignments.* The Web site also provides a number of detailed programming assignments, which include building a multithreaded Web server, building an e-mail client with a GUI interface, programming the sender and receiver sides of a reliable data transport protocol, programming a distributed routing algorithm, and more.
- *Wireshark labs.* One's understanding of network protocols can be greatly deepened by seeing them in action. The Web site provides numerous Wireshark assignments that enable students to actually observe the sequence of messages exchanged between two protocol entities. The Web site includes separate Wireshark labs on HTTP, DNS, TCP, UDP, IP, ICMP, Ethernet, ARP, WiFi, SSL, and on tracing all protocols involved in satisfying a request to fetch a web page. We'll continue to add new labs over time.

Pedagogical Features

We have each been teaching computer networking for more than 20 years. Together, we bring more than 50 years of teaching experience to this text, during which time we have taught many thousands of students. We have also been active researchers in computer networking during this time. (In fact, Jim and Keith first met each other as master's students in a computer networking course taught by Mischa Schwartz in 1979 at Columbia University.) We think all this gives us a good perspective on where networking has been and where it is likely to go in the future. Nevertheless, we have resisted temptations to bias the material in this book

towards our own pet research projects. We figure you can visit our personal Web sites if you are interested in our research. Thus, this book is about modern computer networking—it is about contemporary protocols and technologies as well as the underlying principles behind these protocols and technologies. We also believe that learning (and teaching!) about networking can be fun. A sense of humor, use of analogies, and real-world examples in this book will hopefully make this material more fun.

Supplements for Instructors

We provide a complete supplements package to aid instructors in teaching this course. This material can be accessed from Pearson's Instructor Resource Center (<http://www.pearsonhighered.com/irc>). Visit the Instructor Resource Center or send e-mail to computing@aw.com for information about accessing these instructor's supplements.

- *PowerPoint® slides.* We provide PowerPoint slides for all nine chapters. The slides have been completely updated with this sixth edition. The slides cover each chapter in detail. They use graphics and animations (rather than relying only on monotonous text bullets) to make the slides interesting and visually appealing. We provide the original PowerPoint slides so you can customize them to best suit your own teaching needs. Some of these slides have been contributed by other instructors who have taught from our book.
- *Homework solutions.* We provide a solutions manual for the homework problems in the text, programming assignments, and Wireshark labs. As noted earlier, we've introduced many new homework problems in the first five chapters of the book.

Chapter Dependencies

The first chapter of this text presents a self-contained overview of computer networking. Introducing many key concepts and terminology, this chapter sets the stage for the rest of the book. All of the other chapters directly depend on this first chapter. After completing Chapter 1, we recommend instructors cover Chapters 2 through 5 in sequence, following our top-down philosophy. Each of these five chapters leverages material from the preceding chapters. After completing the first five chapters, the instructor has quite a bit of flexibility. There are no interdependencies among the last four chapters, so they can be taught in any order. However, each of the last four chapters depends on the material in the first five chapters. Many instructors first teach the first five chapters and then teach one of the last four chapters for “dessert.”

One Final Note: We'd Love to Hear from You

We encourage students and instructors to e-mail us with any comments they might have about our book. It's been wonderful for us to hear from so many instructors and students from around the world about our first four editions. We've incorporated many of these suggestions into later editions of the book. We also encourage instructors to send us new homework problems (and solutions) that would complement the current homework problems. We'll post these on the instructor-only portion of the Web site. We also encourage instructors and students to create new Java applets that illustrate the concepts and protocols in this book. If you have an applet that you think would be appropriate for this text, please submit it to us. If the applet (including notation and terminology) is appropriate, we'll be happy to include it on the text's Web site, with an appropriate reference to the applet's authors.

So, as the saying goes, "Keep those cards and letters coming!" Seriously, please *do* continue to send us interesting URLs, point out typos, disagree with any of our claims, and tell us what works and what doesn't work. Tell us what you think should or shouldn't be included in the next edition. Send your e-mail to kurose@cs.umass.edu and ross@poly.edu.

Acknowledgments

Since we began writing this book in 1996, many people have given us invaluable help and have been influential in shaping our thoughts on how to best organize and teach a networking course. We want to say A BIG THANKS to everyone who has helped us from the earliest first drafts of this book, up to this fifth edition. We are also *very* thankful to the many hundreds of readers from around the world—students, faculty, practitioners—who have sent us thoughts and comments on earlier editions of the book and suggestions for future editions of the book. Special thanks go out to:

Al Aho (Columbia University)
 Hisham Al-Mubaid (University of Houston-Clear Lake)
 Pratima Akkunoor (Arizona State University)
 Paul Amer (University of Delaware)
 Shamiul Azom (Arizona State University)
 Lichun Bao (University of California at Irvine)
 Paul Barford (University of Wisconsin)
 Bobby Bhattacharjee (University of Maryland)
 Steven Bellovin (Columbia University)
 Pravin Bhagwat (Wibhu)
 Supratik Bhattacharyya (previously at Sprint)
 Ernst Biersack (Eurécom Institute)

Shahid Bokhari (University of Engineering & Technology, Lahore)
Jean Bolot (Technicolor Research)
Daniel Brushteyn (former University of Pennsylvania student)
Ken Calvert (University of Kentucky)
Evandro Cantu (Federal University of Santa Catarina)
Jeff Case (SNMP Research International)
Jeff Chaltas (Sprint)
Vinton Cerf (Google)
Byung Kyu Choi (Michigan Technological University)
Bram Cohen (BitTorrent, Inc.)
Constantine Coutras (Pace University)
John Daigle (University of Mississippi)
Edmundo A. de Souza e Silva (Federal University of Rio de Janeiro)
Philippe Decuetos (Eurécom Institute)
Christophe Diot (Technicolor Research)
Prithula Dhunghel (Akamai)
Deborah Estrin (University of California, Los Angeles)
Michalis Faloutsos (University of California at Riverside)
Wu-chi Feng (Oregon Graduate Institute)
Sally Floyd (ICIR, University of California at Berkeley)
Paul Francis (Max Planck Institute)
Lixin Gao (University of Massachusetts)
JJ Garcia-Luna-Aceves (University of California at Santa Cruz)
Mario Gerla (University of California at Los Angeles)
David Goodman (NYU-Poly)
Yang Guo (Alcatel/Lucent Bell Labs)
Tim Griffin (Cambridge University)
Max Hailperin (Gustavus Adolphus College)
Bruce Harvey (Florida A&M University, Florida State University)
Carl Hauser (Washington State University)
Rachelle Heller (George Washington University)
Phillipp Hoschka (INRIA/W3C)
Wen Hsin (Park University)
Albert Huang (former University of Pennsylvania student)
Cheng Huang (Microsoft Research)
Esther A. Hughes (Virginia Commonwealth University)
Van Jacobson (Xerox PARC)
Pinak Jain (former NYU-Poly student)
Jobin James (University of California at Riverside)
Sugih Jamin (University of Michigan)
Shivkumar Kalyanaraman (IBM Research, India)
Jussi Kangasharju (University of Helsinki)
Sneha Kasera (University of Utah)
Parviz Kermani (formerly of IBM Research)

Hyojin Kim (former University of Pennsylvania student)
 Leonard Kleinrock (University of California at Los Angeles)
 David Kotz (Dartmouth College)
 Beshan Kulapala (Arizona State University)
 Rakesh Kumar (Bloomberg)
 Miguel A. Labrador (University of South Florida)
 Simon Lam (University of Texas)
 Steve Lai (Ohio State University)
 Tom LaPorta (Penn State University)
 Tim-Berners Lee (World Wide Web Consortium)
 Arnaud Legout (INRIA)
 Lee Leitner (Drexel University)
 Brian Levine (University of Massachusetts)
 Chunchun Li (former NYU-Poly student)
 Yong Liu (NYU-Poly)
 William Liang (former University of Pennsylvania student)
 Willis Marti (Texas A&M University)
 Nick McKeown (Stanford University)
 Josh McKinzie (Park University)
 Deep Medhi (University of Missouri, Kansas City)
 Bob Metcalfe (International Data Group)
 Sue Moon (KAIST)
 Jenni Moyer (Comcast)
 Erich Nahum (IBM Research)
 Christos Papadopoulos (Colorado State University)
 Craig Partridge (BBN Technologies)
 Radia Perlman (Intel)
 Jitendra Padhye (Microsoft Research)
 Vern Paxson (University of California at Berkeley)
 Kevin Phillips (Sprint)
 George Polyzos (Athens University of Economics and Business)
 Sriram Rajagopalan (Arizona State University)
 Ramachandran Ramjee (Microsoft Research)
 Ken Reek (Rochester Institute of Technology)
 Martin Reisslein (Arizona State University)
 Jennifer Rexford (Princeton University)
 Leon Reznik (Rochester Institute of Technology)
 Pablo Rodriguez (Telefonica)
 Sumit Roy (University of Washington)
 Avi Rubin (Johns Hopkins University)
 Dan Rubenstein (Columbia University)
 Douglas Salane (John Jay College)
 Despina Saporita (Cisco Systems)
 John Schanz (Comcast)

Henning Schulzrinne (Columbia University)
 Mischa Schwartz (Columbia University)
 Ardash Sethi (University of Delaware)
 Harish Sethu (Drexel University)
 K. Sam Shanmugan (University of Kansas)
 Prashant Shenoy (University of Massachusetts)
 Clay Shields (Georgetown University)
 Subin Shrestha (University of Pennsylvania)
 Bojie Shu (former NYU-Poly student)
 Mihail L. Sichitiu (NC State University)
 Peter Steenkiste (Carnegie Mellon University)
 Tatsuya Suda (University of California at Irvine)
 Kin Sun Tam (State University of New York at Albany)
 Don Towsley (University of Massachusetts)
 David Turner (California State University, San Bernardino)
 Nitin Vaidya (University of Illinois)
 Michele Weigle (Clemson University)
 David Wetherall (University of Washington)
 Ira Winston (University of Pennsylvania)
 Di Wu (Sun Yat-sen University)
 Shirley Wynn (NYU-Poly)
 Raj Yavatkar (Intel)
 Yechiam Yemini (Columbia University)
 Ming Yu (State University of New York at Binghamton)
 Ellen Zegura (Georgia Institute of Technology)
 Honggang Zhang (Suffolk University)
 Hui Zhang (Carnegie Mellon University)
 Lixia Zhang (University of California at Los Angeles)
 Meng Zhang (former NYU-Poly student)
 Shuchun Zhang (former University of Pennsylvania student)
 Xiaodong Zhang (Ohio State University)
 ZhiLi Zhang (University of Minnesota)
 Phil Zimmermann (independent consultant)
 Cliff C. Zou (University of Central Florida)

We also want to thank the entire Addison-Wesley team—in particular, Michael Hirsch, Marilyn Lloyd, and Emma Snider—who have done an absolutely outstanding job on this sixth edition (and who have put up with two very finicky authors who seem congenitally unable to meet deadlines!). Thanks also to our artists, Janet Theurer and Patrice Rossi Calkin, for their work on the beautiful figures in this book, and to Andrea Stefanowicz and her team at PreMediaGlobal for their wonderful production work on this edition. Finally, a most special thanks go to Michael Hirsch, our editor at Addison-Wesley, and Susan Hartman, our former editor at Addison-Wesley. This book would not be what it is (and may well not have been at all) without their graceful management, constant encouragement, nearly infinite patience, good humor, and perseverance.

Table of Contents

Chapter 1	Computer Networks and the Internet	1
1.1	What Is the Internet?	2
1.1.1	A Nuts-and-Bolts Description	2
1.1.2	A Services Description	5
1.1.3	What Is a Protocol?	7
1.2	The Network Edge	9
1.2.1	Access Networks	12
1.2.2	Physical Media	18
1.3	The Network Core	22
1.3.1	Packet Switching	22
1.3.2	Circuit Switching	27
1.3.3	A Network of Networks	32
1.4	Delay, Loss, and Throughput in Packet-Switched Networks	35
1.4.1	Overview of Delay in Packet-Switched Networks	35
1.4.2	Queuing Delay and Packet Loss	39
1.4.3	End-to-End Delay	42
1.4.4	Throughput in Computer Networks	44
1.5	Protocol Layers and Their Service Models	47
1.5.1	Layered Architecture	47
1.5.2	Encapsulation	53
1.6	Networks Under Attack	55
1.7	History of Computer Networking and the Internet	60
1.7.1	The Development of Packet Switching: 1961–1972	60
1.7.2	Proprietary Networks and Internetworking: 1972–1980	62
1.7.3	A Proliferation of Networks: 1980–1990	63
1.7.4	The Internet Explosion: The 1990s	64
1.7.5	The New Millennium	65
1.8	Summary	66
	Homework Problems and Questions	68
	Wireshark Lab	78
	Interview: Leonard Kleinrock	80

Chapter 2	Application Layer	83
2.1	Principles of Network Applications	84
2.1.1	Network Application Architectures	86
2.1.2	Processes Communicating	88
2.1.3	Transport Services Available to Applications	91
2.1.4	Transport Services Provided by the Internet	93
2.1.5	Application-Layer Protocols	96
2.1.6	Network Applications Covered in This Book	97
2.2	The Web and HTTP	98
2.2.1	Overview of HTTP	98
2.2.2	Non-Persistent and Persistent Connections	100
2.2.3	HTTP Message Format	103
2.2.4	User-Server Interaction: Cookies	108
2.2.5	Web Caching	110
2.2.6	The Conditional GET	114
2.3	File Transfer: FTP	116
2.3.1	FTP Commands and Replies	118
2.4	Electronic Mail in the Internet	118
2.4.1	SMTP	121
2.4.2	Comparison with HTTP	124
2.4.3	Mail Message Format	125
2.4.4	Mail Access Protocols	125
2.5	DNS—The Internet’s Directory Service	130
2.5.1	Services Provided by DNS	131
2.5.2	Overview of How DNS Works	133
2.5.3	DNS Records and Messages	139
2.6	Peer-to-Peer Applications	144
2.6.1	P2P File Distribution	145
2.6.2	Distributed Hash Tables (DHTs)	151
2.7	Socket Programming: Creating Network Applications	156
2.7.1	Socket Programming with UDP	157
2.7.2	Socket Programming with TCP	163
2.8	Summary	168
	Homework Problems and Questions	169
	Socket Programming Assignments	179
	Wireshark Labs: HTTP, DNS	181
	Interview: Marc Andreessen	182

Chapter 3 Transport Layer 185

3.1	Introduction and Transport-Layer Services	186
3.1.1	Relationship Between Transport and Network Layers	186
3.1.2	Overview of the Transport Layer in the Internet	189
3.2	Multiplexing and Demultiplexing	191
3.3	Connectionless Transport: UDP	198
3.3.1	UDP Segment Structure	202
3.3.2	UDP Checksum	202
3.4	Principles of Reliable Data Transfer	204
3.4.1	Building a Reliable Data Transfer Protocol	206
3.4.2	Pipelined Reliable Data Transfer Protocols	215
3.4.3	Go-Back-N (GBN)	218
3.4.4	Selective Repeat (SR)	223
3.5	Connection-Oriented Transport: TCP	230
3.5.1	The TCP Connection	231
3.5.2	TCP Segment Structure	233
3.5.3	Round-Trip Time Estimation and Timeout	238
3.5.4	Reliable Data Transfer	242
3.5.5	Flow Control	250
3.5.6	TCP Connection Management	252
3.6	Principles of Congestion Control	259
3.6.1	The Causes and the Costs of Congestion	259
3.6.2	Approaches to Congestion Control	265
3.6.3	Network-Assisted Congestion-Control Example: ATM ABR Congestion Control	266
3.7	TCP Congestion Control	269
3.7.1	Fairness	279
3.8	Summary	283
	Homework Problems and Questions	285
	Programming Assignments	300
	Wireshark Labs: TCP, UDP	301
	Interview: Van Jacobson	302

Chapter 4 The Network Layer 305

4.1	Introduction	306
4.1.1	Forwarding and Routing	308
4.1.2	Network Service Models	310
4.2	Virtual Circuit and Datagram Networks	313
4.2.1	Virtual-Circuit Networks	314
4.2.2	Datagram Networks	317
4.2.3	Origins of VC and Datagram Networks	319

4.3	What's Inside a Router?	320
4.3.1	Input Processing	322
4.3.2	Switching	324
4.3.3	Output Processing	326
4.3.4	Where Does Queuing Occur?	327
4.3.5	The Routing Control Plane	331
4.4	The Internet Protocol (IP): Forwarding and Addressing in the Internet	331
4.4.1	Datagram Format	332
4.4.2	IPv4 Addressing	338
4.4.3	Internet Control Message Protocol (ICMP)	353
4.4.4	IPv6	356
4.4.5	A Brief Foray into IP Security	362
4.5	Routing Algorithms	363
4.5.1	The Link-State (LS) Routing Algorithm	366
4.5.2	The Distance-Vector (DV) Routing Algorithm	371
4.5.3	Hierarchical Routing	379
4.6	Routing in the Internet	383
4.6.1	Intra-AS Routing in the Internet: RIP	384
4.6.2	Intra-AS Routing in the Internet: OSPF	388
4.6.3	Inter-AS Routing: BGP	390
4.7	Broadcast and Multicast Routing	399
4.7.1	Broadcast Routing Algorithms	400
4.7.2	Multicast	405
4.8	Summary	412
	Homework Problems and Questions	413
	Programming Assignments	429
	Wireshark Labs: IP, ICMP	430
	Interview: Vinton G. Cerf	431

Chapter 5 The Link Layer: Links, Access Networks, and LANs 433

5.1	Introduction to the Link Layer	434
5.1.1	The Services Provided by the Link Layer	436
5.1.2	Where Is the Link Layer Implemented?	437
5.2	Error-Detection and -Correction Techniques	438
5.2.1	Parity Checks	440
5.2.2	Checksumming Methods	442
5.2.3	Cyclic Redundancy Check (CRC)	443
5.3	Multiple Access Links and Protocols	445
5.3.1	Channel Partitioning Protocols	448
5.3.2	Random Access Protocols	449
5.3.3	Taking-Turns Protocols	459
5.3.4	DOCSIS: The Link-Layer Protocol for Cable Internet Access	460

5.4	Switched Local Area Networks	461
5.4.1	Link-Layer Addressing and ARP	462
5.4.2	Ethernet	469
5.4.3	Link-Layer Switches	476
5.4.4	Virtual Local Area Networks (VLANs)	482
5.5	Link Virtualization: A Network as a Link Layer	486
5.5.1	Multiprotocol Label Switching (MPLS)	487
5.6	Data Center Networking	490
5.7	Retrospective: A Day in the Life of a Web Page Request	495
5.7.1	Getting Started: DHCP, UDP, IP, and Ethernet	495
5.7.2	Still Getting Started: DNS and ARP	497
5.7.3	Still Getting Started: Intra-Domain Routing to the DNS Server	498
5.7.4	Web Client-Server Interaction: TCP and HTTP	499
5.8	Summary	500
	Homework Problems and Questions	502
	Wireshark Labs: Ethernet and ARP, DHCP	510
	Interview: Simon S. Lam	511

Chapter 6 Wireless and Mobile Networks 513

6.1	Introduction	514
6.2	Wireless Links and Network Characteristics	519
6.2.1	CDMA	522
6.3	WiFi: 802.11 Wireless LANs	526
6.3.1	The 802.11 Architecture	527
6.3.2	The 802.11 MAC Protocol	531
6.3.3	The IEEE 802.11 Frame	537
6.3.4	Mobility in the Same IP Subnet	541
6.3.5	Advanced Features in 802.11	542
6.3.6	Personal Area Networks: Bluetooth and Zigbee	544
6.4	Cellular Internet Access	546
6.4.1	An Overview of Cellular Network Architecture	547
6.4.2	3G Cellular Data Networks: Extending the Internet to Cellular Subscribers	550
6.4.3	On to 4G: LTE	553
6.5	Mobility Management: Principles	555
6.5.1	Addressing	557
6.5.2	Routing to a Mobile Node	559
6.6	Mobile IP	564
6.7	Managing Mobility in Cellular Networks	570
6.7.1	Routing Calls to a Mobile User	571
6.7.2	Handoffs in GSM	572

6.8	Wireless and Mobility: Impact on Higher-Layer Protocols	575
6.9	Summary	578
	Homework Problems and Questions	578
	Wireshark Lab: IEEE 802.11 (WiFi)	583
	Interview: Deborah Estrin	584

Chapter 7 Multimedia Networking 587

7.1	Multimedia Networking Applications	588
7.1.1	Properties of Video	588
7.1.2	Properties of Audio	590
7.1.3	Types of Multimedia Network Applications	591
7.2	Streaming Stored Video	593
7.2.1	UDP Streaming	595
7.2.2	HTTP Streaming	596
7.2.3	Adaptive Streaming and DASH	600
7.2.4	Content Distribution Networks	602
7.2.5	Case Studies: Netflix, YouTube, and Kankan	608
7.3	Voice-over-IP	612
7.3.1	Limitations of the Best-Effort IP Service	612
7.3.2	Removing Jitter at the Receiver for Audio	614
7.3.3	Recovering from Packet Loss	617
7.3.4	Case Study: VoIP with Skype	620
7.4	Protocols for Real-Time Conversational Applications	623
7.4.1	RTP	624
7.4.2	SIP	627
7.5	Network Support for Multimedia	632
7.5.1	Dimensioning Best-Effort Networks	634
7.5.2	Providing Multiple Classes of Service	636
7.5.3	Diffserv	648
7.5.4	Per-Connection Quality-of-Service (QoS) Guarantees: Resource Reservation and Call Admission	652
7.6	Summary	655
	Homework Problems and Questions	656
	Programming Assignment	666
	Interview: Henning Schulzrinne	668

Chapter 8 Security in Computer Networks 671

8.1	What Is Network Security?	672
8.2	Principles of Cryptography	675
8.2.1	Symmetric Key Cryptography	676
8.2.2	Public Key Encryption	683

8.3	Message Integrity and Digital Signatures	688
8.3.1	Cryptographic Hash Functions	689
8.3.2	Message Authentication Code	691
8.3.3	Digital Signatures	693
8.4	End-Point Authentication	700
8.4.1	Authentication Protocol <i>ap1.0</i>	700
8.4.2	Authentication Protocol <i>ap2.0</i>	701
8.4.3	Authentication Protocol <i>ap3.0</i>	702
8.4.4	Authentication Protocol <i>ap3.1</i>	703
8.4.5	Authentication Protocol <i>ap4.0</i>	703
8.5	Securing E-Mail	705
8.5.1	Secure E-Mail	706
8.5.2	PGP	710
8.6	Securing TCP Connections: SSL	711
8.6.1	The Big Picture	713
8.6.2	A More Complete Picture	716
8.7	Network-Layer Security: IPsec and Virtual Private Networks	718
8.7.1	IPsec and Virtual Private Networks (VPNs)	718
8.7.2	The AH and ESP Protocols	720
8.7.3	Security Associations	720
8.7.4	The IPsec Datagram	721
8.7.5	IKE: Key Management in IPsec	725
8.8	Securing Wireless LANs	726
8.8.1	Wired Equivalent Privacy (WEP)	726
8.8.2	IEEE 802.11i	728
8.9	Operational Security: Firewalls and Intrusion Detection Systems	731
8.9.1	Firewalls	731
8.9.2	Intrusion Detection Systems	739
8.10	Summary	742
	Homework Problems and Questions	744
	Wireshark Lab: SSL	752
	IPsec Lab	752
	Interview: Steven M. Bellovin	753

Chapter 9 Network Management 755

9.1	What Is Network Management?	756
9.2	The Infrastructure for Network Management	760
9.3	The Internet-Standard Management Framework	764
9.3.1	Structure of Management Information: SMI	766
9.3.2	Management Information Base: MIB	770

xxiv Table of Contents

9.3.3	SNMP Protocol Operations and Transport Mappings	772
9.3.4	Security and Administration	775
9.4	ASN.1	778
9.5	Conclusion	783
	Homework Problems and Questions	783
	Interview: Jennifer Rexford	786
	References	789
	Index	823

COMPUTER

SIXTH EDITION

NETWORKING

A Top-Down Approach



This page intentionally left blank



CHAPTER

1

Computer Networks and the Internet

Today's Internet is arguably the largest engineered system ever created by mankind, with hundreds of millions of connected computers, communication links, and switches; with billions of users who connect via laptops, tablets, and smartphones; and with an array of new Internet-connected devices such as sensors, Web cams, game consoles, picture frames, and even washing machines. Given that the Internet is so large and has so many diverse components and uses, is there any hope of understanding how it works? Are there guiding principles and structure that can provide a foundation for understanding such an amazingly large and complex system? And if so, is it possible that it actually could be both interesting *and* fun to learn about computer networks? Fortunately, the answers to all of these questions is a resounding YES! Indeed, it's our aim in this book to provide you with a modern introduction to the dynamic field of computer networking, giving you the principles and practical insights you'll need to understand not only today's networks, but tomorrow's as well.

This first chapter presents a broad overview of computer networking and the Internet. Our goal here is to paint a broad picture and set the context for the rest of this book, to see the forest through the trees. We'll cover a lot of ground in this introductory chapter and discuss a lot of the pieces of a computer network, without losing sight of the big picture.

We'll structure our overview of computer networks in this chapter as follows. After introducing some basic terminology and concepts, we'll first examine the basic hardware and software components that make up a network. We'll begin at the network's edge and look at the end systems and network applications running in the network. We'll then explore the core of a computer network, examining the links and the switches that transport data, as well as the access networks and physical media that connect end systems to the network core. We'll learn that the Internet is a network of networks, and we'll learn how these networks connect with each other.

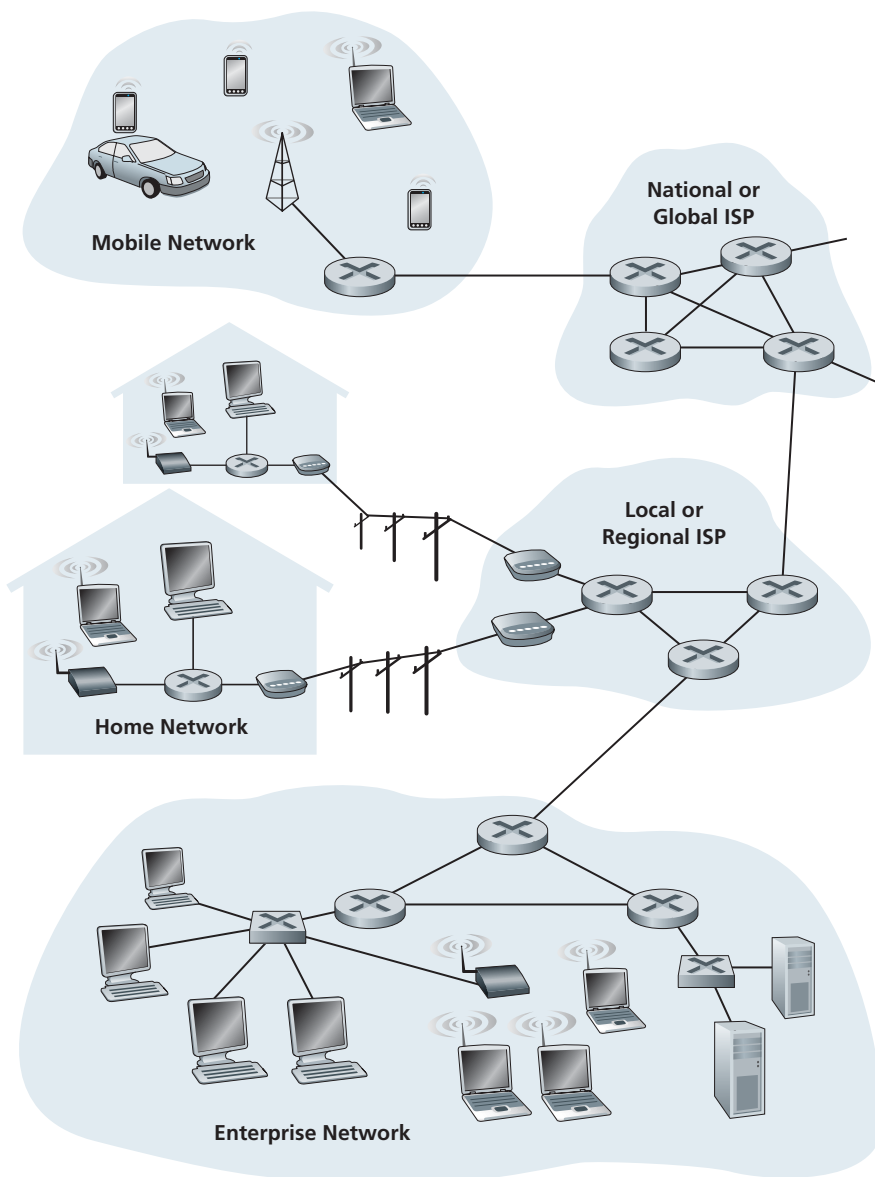
After having completed this overview of the edge and core of a computer network, we'll take the broader and more abstract view in the second half of this chapter. We'll examine delay, loss, and throughput of data in a computer network and provide simple quantitative models for end-to-end throughput and delay: models that take into account transmission, propagation, and queuing delays. We'll then introduce some of the key architectural principles in computer networking, namely, protocol layering and service models. We'll also learn that computer networks are vulnerable to many different types of attacks; we'll survey some of these attacks and consider how computer networks can be made more secure. Finally, we'll close this chapter with a brief history of computer networking.

1.1 What Is the Internet?

In this book, we'll use the public Internet, a specific computer network, as our principal vehicle for discussing computer networks and their protocols. But what *is* the Internet? There are a couple of ways to answer this question. First, we can describe the nuts and bolts of the Internet, that is, the basic hardware and software components that make up the Internet. Second, we can describe the Internet in terms of a networking infrastructure that provides services to distributed applications. Let's begin with the nuts-and-bolts description, using Figure 1.1 to illustrate our discussion.

1.1.1 A Nuts-and-Bolts Description

The Internet is a computer network that interconnects hundreds of millions of computing devices throughout the world. Not too long ago, these computing devices were primarily traditional desktop PCs, Linux workstations, and so-called servers that store and transmit information such as Web pages and e-mail messages. Increasingly, however, nontraditional Internet end systems such as laptops, smartphones, tablets, TVs, gaming consoles, Web cams, automobiles, environmental sensing devices, picture frames, and home electrical and security systems are being connected to the Internet. Indeed, the term *computer network* is beginning to sound a bit dated, given the many nontraditional devices that are being hooked up to the Internet. In Internet jargon, all of these devices are called **hosts** or **end systems**. As of July 2011, there were



Key:

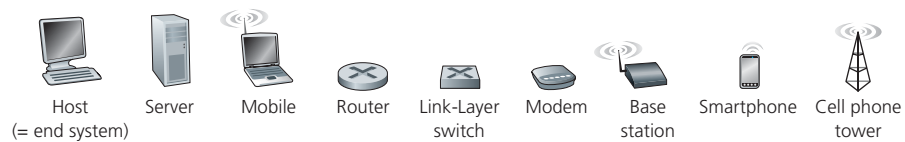


Figure 1.1 ♦ Some pieces of the Internet

nearly 850 million end systems attached to the Internet [ISC 2012], not counting smartphones, laptops, and other devices that are only intermittently connected to the Internet. Overall, there are an estimated 2 billion Internet users [ITU 2011].

End systems are connected together by a network of **communication links** and **packet switches**. We'll see in Section 1.2 that there are many types of communication links, which are made up of different types of physical media, including coaxial cable, copper wire, optical fiber, and radio spectrum. Different links can transmit data at different rates, with the **transmission rate** of a link measured in bits/second. When one end system has data to send to another end system, the sending end system segments the data and adds header bytes to each segment. The resulting packages of information, known as **packets** in the jargon of computer networks, are then sent through the network to the destination end system, where they are reassembled into the original data.

A packet switch takes a packet arriving on one of its incoming communication links and forwards that packet on one of its outgoing communication links. Packet switches come in many shapes and flavors, but the two most prominent types in today's Internet are **routers** and **link-layer switches**. Both types of switches forward packets toward their ultimate destinations. Link-layer switches are typically used in access networks, while routers are typically used in the network core. The sequence of communication links and packet switches traversed by a packet from the sending end system to the receiving end system is known as a **route** or **path** through the network. The exact amount of traffic being carried in the Internet is difficult to estimate but Cisco [Cisco VNI 2011] estimates global Internet traffic will be nearly 40 exabytes per month in 2012.

Packet-switched networks (which transport packets) are in many ways similar to transportation networks of highways, roads, and intersections (which transport vehicles). Consider, for example, a factory that needs to move a large amount of cargo to some destination warehouse located thousands of kilometers away. At the factory, the cargo is segmented and loaded into a fleet of trucks. Each of the trucks then independently travels through the network of highways, roads, and intersections to the destination warehouse. At the destination warehouse, the cargo is unloaded and grouped with the rest of the cargo arriving from the same shipment. Thus, in many ways, packets are analogous to trucks, communication links are analogous to highways and roads, packet switches are analogous to intersections, and end systems are analogous to buildings. Just as a truck takes a path through the transportation network, a packet takes a path through a computer network.

End systems access the Internet through **Internet Service Providers (ISPs)**, including residential ISPs such as local cable or telephone companies; corporate ISPs; university ISPs; and ISPs that provide WiFi access in airports, hotels, coffee shops, and other public places. Each ISP is in itself a network of packet switches and communication links. ISPs provide a variety of types of network access to the end systems, including residential broadband access such as cable modem or DSL,

high-speed local area network access, wireless access, and 56 kbps dial-up modem access. ISPs also provide Internet access to content providers, connecting Web sites directly to the Internet. The Internet is all about connecting end systems to each other, so the ISPs that provide access to end systems must also be interconnected. These lower-tier ISPs are interconnected through national and international upper-tier ISPs such as Level 3 Communications, AT&T, Sprint, and NTT. An upper-tier ISP consists of high-speed routers interconnected with high-speed fiber-optic links. Each ISP network, whether upper-tier or lower-tier, is managed independently, runs the IP protocol (see below), and conforms to certain naming and address conventions. We'll examine ISPs and their interconnection more closely in Section 1.3.

End systems, packet switches, and other pieces of the Internet run **protocols** that control the sending and receiving of information within the Internet. The **Transmission Control Protocol (TCP)** and the **Internet Protocol (IP)** are two of the most important protocols in the Internet. The IP protocol specifies the format of the packets that are sent and received among routers and end systems. The Internet's principal protocols are collectively known as **TCP/IP**. We'll begin looking into protocols in this introductory chapter. But that's just a start—much of this book is concerned with computer network protocols!

Given the importance of protocols to the Internet, it's important that everyone agree on what each and every protocol does, so that people can create systems and products that interoperate. This is where standards come into play. **Internet standards** are developed by the Internet Engineering Task Force (IETF)[IETF 2012]. The IETF standards documents are called **requests for comments (RFCs)**. RFCs started out as general requests for comments (hence the name) to resolve network and protocol design problems that faced the precursor to the Internet [Allman 2011]. RFCs tend to be quite technical and detailed. They define protocols such as TCP, IP, HTTP (for the Web), and SMTP (for e-mail). There are currently more than 6,000 RFCs. Other bodies also specify standards for network components, most notably for network links. The IEEE 802 LAN/MAN Standards Committee [IEEE 802 2012], for example, specifies the Ethernet and wireless WiFi standards.

1.1.2 A Services Description

Our discussion above has identified many of the pieces that make up the Internet. But we can also describe the Internet from an entirely different angle—namely, as *an infrastructure that provides services to applications*. These applications include electronic mail, Web surfing, social networks, instant messaging, Voice-over-IP (VoIP), video streaming, distributed games, peer-to-peer (P2P) file sharing, television over the Internet, remote login, and much, much more. The applications are said to be **distributed applications**, since they involve multiple end systems that exchange data with each other. Importantly, Internet applications

run on end systems—they do not run in the packet switches in the network core. Although packet switches facilitate the exchange of data among end systems, they are not concerned with the application that is the source or sink of data.

Let's explore a little more what we mean by an infrastructure that provides services to applications. To this end, suppose you have an exciting new idea for a distributed Internet application, one that may greatly benefit humanity or one that may simply make you rich and famous. How might you go about transforming this idea into an actual Internet application? Because applications run on end systems, you are going to need to write programs that run on the end systems. You might, for example, write your programs in Java, C, or Python. Now, because you are developing a distributed Internet application, the programs running on the different end systems will need to send data to each other. And here we get to a central issue—one that leads to the alternative way of describing the Internet as a platform for applications. How does one program running on one end system instruct the Internet to deliver data to another program running on another end system?

End systems attached to the Internet provide an **Application Programming Interface (API)** that specifies how a program running on one end system asks the Internet infrastructure to deliver data to a specific destination program running on another end system. This Internet API is a set of rules that the sending program must follow so that the Internet can deliver the data to the destination program. We'll discuss the Internet API in detail in Chapter 2. For now, let's draw upon a simple analogy, one that we will frequently use in this book. Suppose Alice wants to send a letter to Bob using the postal service. Alice, of course, can't just write the letter (the data) and drop the letter out her window. Instead, the postal service requires that Alice put the letter in an envelope; write Bob's full name, address, and zip code in the center of the envelope; seal the envelope; put a stamp in the upper-right-hand corner of the envelope; and finally, drop the envelope into an official postal service mailbox. Thus, the postal service has its own "postal service API," or set of rules, that Alice must follow to have the postal service deliver her letter to Bob. In a similar manner, the Internet has an API that the program sending data must follow to have the Internet deliver the data to the program that will receive the data.

The postal service, of course, provides more than one service to its customers. It provides express delivery, reception confirmation, ordinary use, and many more services. In a similar manner, the Internet provides multiple services to its applications. When you develop an Internet application, you too must choose one of the Internet's services for your application. We'll describe the Internet's services in Chapter 2.

We have just given two descriptions of the Internet; one in terms of its hardware and software components, the other in terms of an infrastructure for providing services to distributed applications. But perhaps you are still confused as to what the

Internet is. What are packet switching and TCP/IP? What are routers? What kinds of communication links are present in the Internet? What is a distributed application? How can a toaster or a weather sensor be attached to the Internet? If you feel a bit overwhelmed by all of this now, don't worry—the purpose of this book is to introduce you to both the nuts and bolts of the Internet and the principles that govern how and why it works. We'll explain these important terms and questions in the following sections and chapters.

1.1.3 What Is a Protocol?

Now that we've got a bit of a feel for what the Internet is, let's consider another important buzzword in computer networking: *protocol*. What is a protocol? What does a protocol do?

A Human Analogy

It is probably easiest to understand the notion of a computer network protocol by first considering some human analogies, since we humans execute protocols all of the time. Consider what you do when you want to ask someone for the time of day. A typical exchange is shown in Figure 1.2. Human protocol (or good manners, at least) dictates that one first offer a greeting (the first “Hi” in Figure 1.2) to initiate communication with someone else. The typical response to a “Hi” is a returned “Hi” message. Implicitly, one then takes a cordial “Hi” response as an indication that one can proceed and ask for the time of day. A different response to the initial “Hi” (such as “Don't bother me!” or “I don't speak English,” or some unprintable reply) might indicate an unwillingness or inability to communicate. In this case, the human protocol would be not to ask for the time of day. Sometimes one gets no response at all to a question, in which case one typically gives up asking that person for the time. Note that in our human protocol, *there are specific messages we send, and specific actions we take in response to the received reply messages or other events* (such as no reply within some given amount of time). Clearly, transmitted and received messages, and actions taken when these messages are sent or received or other events occur, play a central role in a human protocol. If people run different protocols (for example, if one person has manners but the other does not, or if one understands the concept of time and the other does not) the protocols do not interoperate and no useful work can be accomplished. The same is true in networking—it takes two (or more) communicating entities running the same protocol in order to accomplish a task.

Let's consider a second human analogy. Suppose you're in a college class (a computer networking class, for example!). The teacher is droning on about protocols and you're confused. The teacher stops to ask, “Are there any questions?” (a

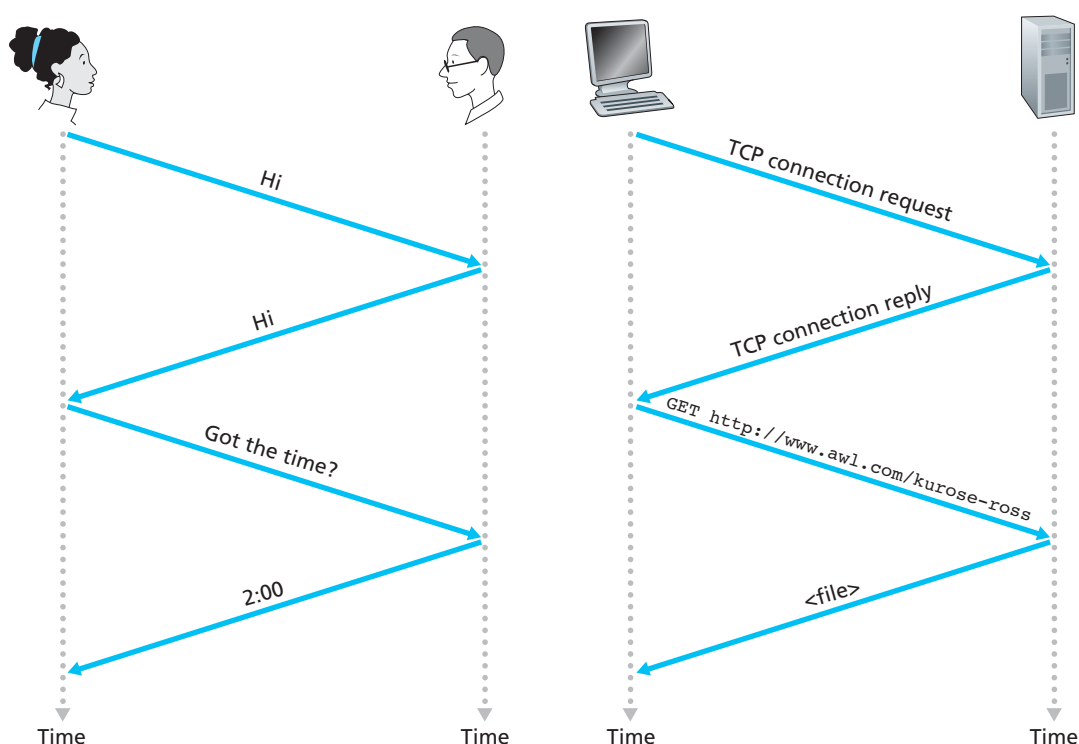


Figure 1.2 ♦ A human protocol and a computer network protocol

message that is transmitted to, and received by, all students who are not sleeping). You raise your hand (transmitting an implicit message to the teacher). Your teacher acknowledges you with a smile, saying “Yes . . .” (a transmitted message encouraging you to ask your question—teachers *love* to be asked questions), and you then ask your question (that is, transmit your message to your teacher). Your teacher hears your question (receives your question message) and answers (transmits a reply to you). Once again, we see that the transmission and receipt of messages, and a set of conventional actions taken when these messages are sent and received, are at the heart of this question-and-answer protocol.

Network Protocols

A network protocol is similar to a human protocol, except that the entities exchanging messages and taking actions are hardware or software components of some device (for example, computer, smartphone, tablet, router, or other network-capable