

Data Communication and Networks

Bhushan Trivedi

*Director, GLS Institute of Computer Technology
&
Dean, School of Computer Technology, GLS University
Ahmedabad*

OXFORD
UNIVERSITY PRESS

OXFORD
UNIVERSITY PRESS

Oxford University Press is a department of the University of Oxford.
It furthers the University's objective of excellence in research, scholarship,
and education by publishing worldwide. Oxford is a registered trade mark of
Oxford University Press in the UK and in certain other countries.

Published in India by
Oxford University Press
YMCA Library Building, 1 Jai Singh Road, New Delhi 110001, India

© Oxford University Press 2016

The moral rights of the author/s have been asserted.

First published in 2016

All rights reserved. No part of this publication may be reproduced, stored in
a retrieval system, or transmitted, in any form or by any means, without the
prior permission in writing of Oxford University Press, or as expressly permitted
by law, by licence, or under terms agreed with the appropriate reprographics
rights organization. Enquiries concerning reproduction outside the scope of the
above should be sent to the Rights Department, Oxford University Press, at the
address above.

You must not circulate this work in any other form
and you must impose this same condition on any acquirer.

ISBN-13: 978-0-19-945599-7
ISBN-10: 0-19-945599-6

Typeset in Times New Roman
by Cameo Corporate Services Limited, Chennai
Printed in India by Magic International (P) Ltd, Greater Noida

Third-party website addresses mentioned in this book are provided
by Oxford University Press in good faith and for information only.
Oxford University Press disclaims any responsibility for the material contained therein.

Dedicated to
All my teachers
whose unselfish love and dedication towards their work
have taught me more than their words

Features of

3.2 SIGNALLING AND TRANSMISSION

When we store a file using any file storage mechanism on our website, it stores them in the form of zeros and ones.

15.1 IPSEC—SECURITY AT NETWORK LAYER

The IPsec is run at the network layer and paves way for the data to travel through. The significance of using IPsec is that it runs below the applications. When

Comprehensive Coverage

The book provides a comprehensive coverage of topics ranging from basic to advanced concepts.

Table 11.1 Difference between connection-oriented and connectionless service

Criteria	Connectionless	Connection oriented
Establishment and termination	No	Yes
Routing	With every packet	Only with the first packet
Speed of operation	Takes more time	Takes less time
Avoidance of congestion	Not easy	Admission control
Recovery from congestion	Possible	Not possible
Addressing	Requires complete address	Smaller IDs are used
Robustness	Can sustain line or node failure	Brittle
Quality of service	Difficult	Easy

Figures and Tables

A plethora of clearly illustrated figures and tables, interspersed at suitable positions in the text, enable easy understanding of concepts.

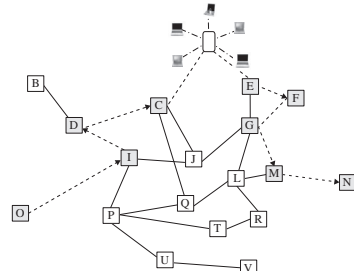


Fig. 12.18 Longer path taken when there is congestion. All packets sent at this point travel the path O-I-...-E-F-M-N

Example 10.7 Suppose a user wants a network address for his/her 20-node network. Now, the ISP has addresses starting from 20.0.0.0 and has not yet allocated any addresses from that lot. How are these addresses provided? Find out the range of IP addresses that the user will get.

Solution The user needs 20 addresses and the nearest multiple of 2 is 32. Thus, he/she needs to have a 5-bit host ID. Thus, from 20.0.0.0 to 20.0.0.31, all addresses

are available. If you have a 5-bit host ID, the remaining 3 bits are for the network ID.

Example 10.8 Suppose a user wants a network address for his/her 20-node network. Now, the ISP has addresses starting from 20.0.0.0 and has not yet allocated any addresses from that lot. How are these addresses provided? Find out the range of IP addresses that the user will get.

Solution For 20 addresses, the nearest multiple of 2 is 32. Thus, he/she needs to have a 5-bit host ID. Thus, from 20.0.0.0 to 20.0.0.31, all addresses

Examples

Each chapter has numerous solved illustrations along with the solution.

Sidebars

The text is interspersed with sidebars that highlight important concepts.

Every SONET frame is sent in 125 μ s to make 8000 frames a per second.

Layering in networks help conquer a complex problem by dividing it into independent simpler modules.

Digital data is converted into another digital form that is better for communication.

the Book

Points to Remember

A list of points at the end of each chapter highlights the key concepts discussed in the chapter to enable quick revision.

POINTS TO REMEMBER

- The communication link between the sender and the receiver can be either wired or wireless. Cable lines and conventional landline systems use wired physical connections. Direct to home and mobile communications are examples where wireless physical connections are used.
- xDSL technology is quite promising as it provides a large bandwidth using the same technology that is used in telephones.
- SONET allows a frame to be in any position.
- Wireless LANs use 802.11 known as Wi-Fi.
- Wireless MAN uses 802.16 protocol.
- Access point is a device that enables communication from multiple devices to a central point.
- Wireless networks are also used in places where access points are not available.

KEYWORDS

802.11 It is a standard to implement wireless LAN based on ethernet.

802.16 It is a standard to implement wireless MAN.

Access point It is a device that controls transmission from multiple nodes belonging to a single network segment when the wireless network works in an infrastructure or a PCF mode.

Ad hoc mode The wireless communication between devices that occur without using access points is known as ad hoc mode transmission. It is also known as distributed mode.

Discrete multitone A type of modulation where each sub-carrier is dynamically adjusted to the channel conditions.

Dispersion The spreading of a signal due to different propagation speeds of its components.

DSL It is a type of digital subscriber line technology that is used for high-speed data transmission over copper telephone lines.

Keywords

All chapters provide a list of key terms along with their definitions for quick recapitulation of important concepts.

Objective Questions

Multiple-choice questions, along with their answers, are provided at the end of each chapter to facilitate revision.

MULTIPLE-CHOICE QUESTIONS

1. A network contains
(a) only computers
(b) computers, printers, scanners, and auxiliary devices
(c) computers, internetworking devices, and auxiliary devices
(d) any device that can connect
2. Current state-of-the-art LANs use _____ topology.
(a) star
(b) bus
(c) ring
(d) mesh
3. For wireless connections, larger the _____ band, larger the amount of traffic that can be transmitted.
(a) of the exact type
(b) of the exact speed
(c) a generic type of network
(d) multiple networks
4. The connectionless connection does not require
(a) a connection establishment process
(b) a connection close process
(c) a both establishment and close process
(d) a physical connection
5. The application layer can also provide
(a) connection
(b) API
(c) data
(d) service
6. Home networks face the problem of
(a) security
(b) bandwidth
(c) both security and bandwidth
(d) none of these

REVIEW QUESTIONS

1. List the duties of the data link layer.
2. What is the need for a data link layer to multiplex multiple network layer connections?
3. Explain the difference between a circuit-switched network and a packet-switched network.
4. Explain the difference between a circuit-switched network and a packet-switched network.
5. Explain the difference between a circuit-switched network and a packet-switched network.
6. Explain the difference between a circuit-switched network and a packet-switched network.
7. Explain the difference between a circuit-switched network and a packet-switched network.
8. Explain the difference between a circuit-switched network and a packet-switched network.
9. How does Manchester coding work?
10. Explain the difference between a circuit-switched network and a packet-switched network.

PRACTICAL PROBLEMS

1. If the frequency of a wave is 10 Hz, what is the period of that wave?
(Ans: 0.1 oscillations/second)
2. If a shooter shoots 1000 shots in 30 s, what is the frequency of the shots and what is the average period?
(Ans: 33.33 shots/s, 0.03 s/shot)
3. If the velocity of a wave is 2×10^8 m/s in a specific medium and the wavelength of that wave in vacuum is 1000 nm, what is the wavelength of that wave in the given medium?
4. Suppose there is a channel with a bandwidth of 10 MHz. If we use 8 signal levels, how many bits can be transmitted per second?
5. Suppose there is a channel with a bandwidth of 10 MHz. If we use 8 signal levels, how many bits can be transmitted per second?
6. Suppose there is a channel with a bandwidth of 10 MHz. If we use 8 signal levels, how many bits can be transmitted per second?
7. Suppose there is a channel with a bandwidth of 10 MHz. If we use 8 signal levels, how many bits can be transmitted per second?
8. Suppose there is a channel with a bandwidth of 10 MHz. If we use 8 signal levels, how many bits can be transmitted per second?
9. Suppose there is a channel with a bandwidth of 10 MHz. If we use 8 signal levels, how many bits can be transmitted per second?
10. Suppose there is a channel with a bandwidth of 10 MHz. If we use 8 signal levels, how many bits can be transmitted per second?

Review Questions and Practical Problems

Numerous review questions are provided at the end of every chapter and practical problems in relevant chapters to test the readers' understanding of the concepts learnt.

Companion Online Resources



Visit india.oup.com/orcs/9780199455997 to access both teaching and learning solutions online.

The following resources are available to support the faculty and students using this book:

For Faculty

- Solutions Manual for Select Chapter-end Exercises
- Chapter-wise PowerPoint Slides
- Question Bank

For Students

- Test Generator

Steps to register and access Online Resources

Resources for instructors and students are developed to complement each text book and varies from book to book.

Step 1: Getting Started

- Go to india.oup.com

Step 2: Browse quickly by

- BASIC SEARCH
 - AUTHOR
 - TITLE
 - ISBN
- ADVANCED SEARCH
 - KEYWORDS
 - AUTHOR
 - TITLE
 - SUBTITLE
 - PUBLICATION DATE

Step 3: Select title

- Select Product
- Select Online Resources

Step 4: View Resources

- Click on "View all resources"

[View all resources](#)

Step 5: Sign in with your Oxford ID

Sign in with your Oxford ID

I am a returning user

Phone no.

Email Address

Password

(Register your password?)

[Sign in](#)

Step 6: If you do not have an Oxford ID, register with us

Personal Details

Name

Email Address

Register for an Oxford ID

User Name

Password
Must be at least 8 characters and should include at least one capital letter, lower-case letter and number.

Confirm Password

☐ Do you accept the terms and conditions?

[Continue](#)

Step 7: Fill in your details

- Fill the detailed registration form with correct particulars.
- Fields marked with '*' in the form, are mandatory.
- Update

[Update](#)

Step 8: Validation

- We shall revert to you within 48 hours after verifying the details provided by you. Once validated, please login using your username and the password and access the resources.

Step 9: Confirmation

- You will receive a confirmation on your email ID.

Step 10: Visit us again

- Go to india.oup.com
- Sign in with Oxford ID

Step 11: Visit your licensed products

- Go to "Resources" section

Resources

You currently have access to:

Product	Status	Licence

Step 12: Download Resources

- Click on the title
- View online resources
- Select resource type
- Download the resource you require.

 For any further queries, please write to us at HEMarketing.in@oup.com with your mobile number.

Preface

Computer networks have slowly and surely become a part of our day-to-day lives. It is rightly said that when a technology becomes a part of our life, so much so that we do not even notice its existence, it has really matured and has turned out to be successful. When we seamlessly connect to the rest of the world, either by means of the Internet or our corporate or institute network, we may not even realize the amount of effort that goes into making it possible. This book, *Data Communication and Networks*, is designed to showcase how exactly this has been made possible.

Today, it is hard to imagine a computer working in isolation from any network. A computer network, often simply referred to as a network, is a group of computers and devices interconnected by communication channels that facilitate communications among users and allow users to share resources. If we say the network is designed to cater to the needs of only computers, we are wrong. The so-called computer network is extended into a network of heterogeneous devices communicating with each other using different types of protocols and numerous user application forms.

One way of looking at a network is by understanding the set of commands one would like to type to get the desired job done (such as typing FTP to download or upload a file, or Telnet to connect to a remote machine). Another way of analysing networks is to find out how machines are connected to the network physically (adding network cards, attaching IO cables to connect switches with computers or laptops, fixing access points for wireless access, etc). Neither approach can remove all doubts from a student's mind. A student would like to know what goes on inside the network when a specific command is typed, what happens 'behind the scenes' when an email is sent and received, and seek answers to other similar questions. Data is communicated using wired and wireless devices, smartphones, and other smart devices. The conventional TCP/IP model is evolving to adapt to the changes around us.

Data Communication and Networks is thus offered as an important course across fields such as computer science, information technology, and electronics and communication, with the aim of throwing some light on all these issues.

ABOUT THE BOOK

Data Communication and Networks is designed as a textbook for students of engineering (BE/BTech), IT, and MCA. It presents the theoretical aspects of data communication and computer networks in a comprehensive manner to make the concepts clear to the users. The book presumes readers have basic knowledge of C programming and data structures.

There are two ways of understanding a subject—one is to enumerate the features one after another and allow the reader to gradually build the relationship between the different components of the subject himself/herself, whereas the other is to provide conceptual understanding, where the onus is on the reader to determine the real-life situations that fit these concepts. An attempt has been made in this book to balance both the approaches, that is, by providing conceptual understanding, which is obviously very important for a student, augmented with real-world examples such as ethernet and the Internet. In fact, the book starts with an example demonstrating the role of each layer of the computer network before plunging into the details of layering and the technologies related to those layers.

The other important part of the design of this book is that it is based on real-world needs and solutions. Almost all the topics discussed in the book contain a part explaining where, in the real world, the topic under consideration is applied and used.

Conventionally, networks are taught to students as a set of layers; the industry also publishes material citing different layers (a layer 4 switch is one example). In this book, the same approach has been followed.

In addition, the book includes several analogies interspersed across chapters to enable easy understanding of concepts. There are numerous exercises, including multiple-choice questions, review questions, and numerical problems, which are given at the end of each chapter, for practice.

KEY FEATURES

- Incorporates the layered approach emphasizing TCP/IP model with real-world applications such as the Internet and ethernet technologies
- Provides detailed discussions on topics such as electromagnetic spectrum, switches, coding, wired LAN, Wi-fi, WiMax, and VLAN
- Covers the fundamentals of the data link layer and its related protocols, medium access sublayer, and classful and classless addressing
- Explains how WiMax or 802.16 provides quality of service, use of scalable OFDM in 802.16, and why congestion control is used at the transport layer rather than the network layer on the Internet
- Provides in-depth coverage of application layer concepts such as DNS, WWW, HTTP, email system, FTP, bluetooth, multimedia, and DHCP
- Includes numerous illustrations, sidebars, tables, key terms, and points to remember to enable quick recapitulation

CONTENT AND STRUCTURE

The book has 15 chapters and a brief description of each chapter is given here.

Chapter 1 deals with the layering concept and describes how data communication is carried out using multiple layers. It describes each of the five layers of the TCP/IP stack and the seven layers of the OSI stack. The functionality of each layer is discussed next. The chapter also describes the scenarios or situations when the layering solution does not work. It throws light on some other important issues like client-server computing and distributed systems. It ends with a discussion on standardization bodies.

Chapter 2 describes the different categories under which a network can be classified. It also discusses the different network components such as the repeater, hub, bridge, switch, and router. It explains the difference between physical and logical connections, the relation between users and services, and the different applications of computer networks. The chapter ends with a note on social engineering and the security issues related to networks.

Chapter 3 discusses the data communication fundamentals. First, the difference between bandwidth and data rate is established. Then other issues, such as analog and digital signalling and transmission, are discussed. After these, the different modulation schemes for analog signalling, errors, multiplexing, etc., are explained. The MDR for a channel is expounded as well.

Chapter 4 throws light on the fundamentals of digital communication which begins with digital-to-digital and analog-to-digital conversion. Multiplexing, FDM, TDM, and STDM are introduced here. The chapter also discusses the different types of coding schemes and the factors used in evaluating these schemes.

Chapter 5 lays emphasis on the functioning of the physical layer. It describes how bits are transferred from one end to the other and the problems faced in doing so. The chapter explains the need to have techniques other than FDM and TDM for data transmission in bursty traffic. The chapter explains spread spectrum methods, circuit switching, and packet switching, in addition to the

electromagnetic spectrum and its different bands. Moreover, the chapter explains the suitability of each particular band for data communication.

Chapter 6 extends the discussion on physical layer by discussing wired and wireless physical layers, where fibre-optic and UTP cables are mentioned. The chapter also explores telephone lines as a medium for data communication, SONET, wireless physical layer and the specific problems to be addressed in this layer, the different components of wireless networks, 802.11, 802.16, and satellite wireless physical networks.

Chapter 7 deals with the data link layer. Why the idea of no monopoly lead to packet switching is described in the beginning and why and how framing is done is described next. The different types of errors, and the difference between error detection and error correction is also described. The chapter also touches upon flow control.

Chapter 8 elucidates the communication protocols. The chapter demonstrates the methods and issues in data communication at the data link layer by proposing different communication models in increasingly complex cases. It discusses two protocols used in the real world, HDLC and PPP, in depth.

Chapter 9 deals with medium access sublayer, a special part of the data link layer used in local area networks such as ethernet and Wi-fi (802.11). The chapter begins with channel access methods and the issues with shared communication channels. The management of arbitration issues is described next. This is followed by a study on cellular telephony, wired MAC layer including classic ethernet and later versions of ethernet, wireless network layer including 802.11 and 802.16, and extensions of 802.11. The problems connecting heterogeneous networks and their solutions are examined next. The chapter ends with a description of virtual LAN and ethernet virtual LAN standard (802.1Q).

Chapter 10 is dedicated to the understanding of the Internet addressing mechanism. The three types of addressing, old classful addressing, and CIDR-based methods based on IPv4 with introduction to NAT are covered in this chapter. The real solution to the addressing problem, the addressing based on IPv6, is discussed in the end. The chapter ends with a discussion on dynamic addressing used on the Internet, based on DHCP.

Chapter 11 deals with the network layer. It illustrates the difference between routing and forwarding and then differentiates between connection-oriented forwarding and connectionless forwarding. The routing algorithms for external and internal routing including DV, LS, and BGP are discussed. Hierarchical, broadcast, and multicast routing are explained next. The chapter discusses congestion and the different methods of combating congestion, network layer switching, and internetworking issues. IGMP, ICMPv6, IPv4, and IPv6 are also covered. The chapter ends with IPsec as a solution to network layer secure solution.

Chapter 12 on the transport layer lays emphasis on the transport layer as an end-to-end solution provider and how it manages problems in an end-to-end solution. Connection management, congestion control, and related issues are discussed from the point of view of the communication between a sender and a receiver using TCP as a case study. A discussion on client-server communication is followed by its problems and solutions, and a sample client-server program. Standard and non-standard performance issues are then dealt with. SCTP protocol is briefed upon in the end.

Chapter 13 gives an outline of concepts such as the functioning of the application layer, domain name system, the registration process, how names are resolved using resource records, the latest techniques such as DDNS and DNSSec, the HTTP protocol as a vehicle for the World Wide Web, the Internet email system based on SMTP protocol, web and ISP-based mailing, and file transfer protocol, the next protocol at the application layer. Recent advancements such as secure FTP and SSL-FTP, bluetooth, RTP, and SIP are also presented.

Chapter 14 introduces one of the most discussed topics of today, which is cryptography. The chapter begins with the description of a conventional security model and the different components,

followed by block ciphers, their different modes, and quantum cryptography. The chapter explains cryptanalysis and digital signatures using different methods. This is followed by public key management and authentication protocols including Kerberos. The chapter ends with other security issues including information security.

Chapter 15 delves into security, including a detailed description on network layer security using IPsec and application layer security using PGP and S/MIME. The chapter ends with a discussion on intrusion detection and prevention.

ONLINE RESOURCES

To aid the faculty and students using this book, additional resources are available @ india.oup.com/orcs/9780199455997

For Faculty

- Solutions Manual for Select Chapter-end Exercises
- Chapter-wise PowerPoint Slides
- Question Bank

For Students

- Test Generator

ACKNOWLEDGEMENTS

After the success of my first book, *Programming with ANSI C++*, and second book, *Computer Networks*, I was motivated to write another one on data communication and networks in view of the academic requirements. The team at OUP had a host of ideas for the book. I have thoroughly enjoyed working with OUP, yet again. The project took longer in the making than expected due to the dynamic nature of the field and the academic scenario in India. The OUP team is extremely dedicated to publishing high-quality texts—errors are caught and corrected at various stages of revision. I acknowledge the help received from the OUP team from the bottom of my heart.

I express my gratitude towards my colleagues at GLS, especially our Vice-president, Shri Sudir Nanavati, who congratulated me on both my books and insisted that I continue this effort, encouraging me to write more texts.

I also acknowledge the support of my wife Arpita, son Jay who is my first critic, daughter Sonu, and my late parents Kantaben and Harshadrai in this effort. Writing a book isolates one for a significant period and family members need to be patient and support the ‘author’ by lending a helping hand. I am thankful to them for the same.

The students at GLS are always ready to test my knowledge and ask thought-provoking questions. Some of my research students, who work on network-related topics, are also not far behind. Their questions, and sometimes their answers in response to my questions, taught me things I would have otherwise overlooked. Invigorating discussions with them gave me new perspectives to well-established concepts. All this helped me gain better knowledge and lead to this book. I would like to acknowledge their help here.

Suggestions to improve the content of the book would be welcome. You may send your feedback at bhtrivedi@gmail.com.

Bhushan Trivedi

Brief Contents

Features of the Book iv

Companion Online Resources vi

Preface vii

Detailed Contents xii

1. Introduction to Computer Networks and Data Communication	1
2. Network Fundamentals	44
3. Data Communication Fundamentals	76
4. Digital Communication	119
5. The Physical Layer	174
6. Wired and Wireless Physical Layers	209
7. The Data Link Layer	245
8. Network Protocols	285
9. The Medium Access Sub-layer	320
10. Addressing in Internet	392
11. The Network Layer	436
12. The Transport Layer	545
13. The Application Layer	624
14. Cryptography and Security	711
15. IP Security and Other Security Solutions	784

Index 852

About the Author 861

Detailed Contents

Features of the Book iv
Companion Online Resources vi
Preface vii
Brief Contents xi

1. Introduction to Computer Networks and Data Communication

Introduction	1
1.1 Data Communication	1
1.1.1 Delivery Problems	2
1.1.2 Communication Components	3
1.1.3 Data Representation	4
1.2 Data Transmission Modes	4
1.2.1 Simplex Mode	5
1.2.2 Half-duplex Mode	5
1.2.3 Full-duplex Mode	6
1.3 Concept of Layering	6
1.3.1 Layering Example	6
1.3.2 Advantages of Layering Mechanism	9
1.3.3 Disadvantages of Layering	10
1.4 TCP/IP and OSI Layering Models	11
1.4.1 OSI Model	12
1.4.2 TCP/IP Model	13
1.4.3 Connection-oriented vs Connectionless Transfer	13
1.4.4 Differences between TCP/IP and OSI Models	16
1.4.5 Top-down and Bottom-up Approaches to Study Layers	17
1.4.6 Functions of Each Layer	18
1.5 Communication Process	25
1.6 Distributed Systems and Networks	30
1.7 Peer-to-peer and Client–Server Networks	30
1.8 Connection-oriented Networks—X.25 and Frame Relay	32
1.9 Network and Communication Standardization	34
1.9.1 Standardization Bodies	35
1.10 The Internet	38
1.10.1 History of the Internet	38
1.10.2 Internet Architecture	39

2. Network Fundamentals

Introduction	44
2.1 Definition and Prerequisites of Networks	44

2.1.1 Definition—Network	45
2.2 Network Categories	46
2.2.1 Personal Area Network	46
2.2.2 Local Area Network	46
2.2.3 Metropolitan Area Network	47
2.2.4 Wide Area Network	48
2.3 Network Components	50
2.3.1 Network Interface Card	50
2.3.2 Cable for Wired Connection	51
2.3.3 Frequency Band for Wireless Transmission	51
2.3.4 Servers and Nodes	52
2.3.5 Interconnecting Devices	52
2.4 Connection	57
2.4.1 Physical Connection	57
2.4.2 Logical Connection	59
2.5 Layers and Services	63
2.5.1 Design of Layers as Service Providers and Users	63
2.5.2 Standardization of Services	64
2.5.3 Quality of Service Issues	64
2.6 Users and Services	65
2.6.1 Desired Characteristics of Application Layer	66
2.7 Example of Internet Protocol	66
2.7.1 Connections and Protocols	67
2.7.2 Layers and Protocols	67
2.8 Applications of Computer Networks	69
2.8.1 Home Networking Applications	69
2.8.2 Mobile Networking Applications	69
2.8.3 Wireless Networking Applications	70
2.9 Security Issues	70
2.9.1 Hidden Dangers	70
2.9.2 Policy Issues	71
2.9.3 User Convenience vs Secure Network	71
2.9.4 Social Engineering	71

3. Data Communication Fundamentals

Introduction	76
3.1 Bandwidth and Data Rate	76
3.1.1 Frequency and Band	78
3.2 Signalling and Transmission	78

76

3.2.1 Analog and Digital Data and Signalling	78	4.5.2 Wavelength Division Multiplexing	155
3.2.2 Periodic and Aperiodic Waves	79	4.5.3 Time Division Multiplexing	156
3.2.3 Analog Signals—Composite Signals	84	4.5.4 Statistical Time Division Multiplexing	164
3.2.4 Digital Signal and Fourier Components	87	4.6 Multiplexing at All Layers	166
3.2.5 Filters	89	4.7 Switching and Routing	167
3.2.6 Implementing Analog Signalling	90		
3.2.7 Implementing Digital Signalling	90	5. The Physical Layer	174
3.2.8 Difference between Analog and Digital Signalling	94	Introduction	174
3.2.9 Digital Signalling and Error	94	5.1 Duties of Physical Layer	175
3.2.10 Analog and Digital Transmission	96	5.1.1 Converting Bits into Signals	175
3.3 Types of Communication Channels	97	5.1.2 Delivery and Synchronization between Sender and Receiver	175
3.3.1 Low-pass or Baseband Channels	97	5.1.3 Multiplexing and Demultiplexing Data	177
3.3.2 Band-pass or Broadband Channels	98	5.2 Inappropriateness of Frequency Division Multiplexing and Time Division Multiplexing for Bursty Data	178
3.3.3 Wide and Narrow Bandwidth Channels	98	5.3 Spread Spectrum	179
3.4 Maximum Data Rate of Channel	99	5.3.1 Spread Spectrum Generation Process	179
3.4.1 Properties of Channels	99	5.3.2 Frequency Hopping Spread Spectrum	180
3.4.2 MDR based on Nyquist Theorem	100	5.3.3 Multiple Users and Multiple Frequencies	181
3.4.3 Signal-to-Noise Ratio	101	5.3.4 Direct Sequence Spread Spectrum	183
3.4.4 MDR based on Shannon Theorem	101	5.4 Switching	185
3.5 Transmission and Errors	103	5.4.1 Packet Switching	186
3.5.1 Effect of Media on Bandwidth of Signal	103	5.4.2 Circuit Switching	188
3.5.2 Attenuation	103	5.4.3 Space Division Switches	190
3.5.3 Noise	106	5.4.4 Time Division Switches	195
3.6 Communication Line Performance	106	5.4.5 Virtual Circuit Switching	196
3.6.1 Bandwidth	107	5.4.6 Message Switching	197
3.6.2 Throughput	107	5.5 Electromagnetic Spectrum	197
3.6.3 Delay	108	5.5.1 Radio Waves	198
3.7 Modulation	109	5.5.2 Microwaves	199
3.7.1 Amplitude Modulation	109	5.5.3 Infrared and Millimetre Waves	201
3.7.2 Frequency Modulation	110	5.5.4 Industrial, Scientific, and Medical Bands	202
3.7.3 Phase Modulation	110	5.5.5 Optical Light and Free Space Optics	203
3.7.4 Modulation in Practice	111	5.5.6 X-rays and Gamma Rays	203
		5.6 Future Trends	203
4. Digital Communication	119		
Introduction	119	6. Wired and Wireless Physical Layers	209
4.1 Need for Digital to Digital Conversion	119	Introduction	209
4.2 Digital Encoding (Digital Data—Digital Signals)	119	6.1 Wired Physical Layer	209
4.2.1 Synchronizing Sender and Receiver	120	6.1.1 Unshielded Twisted Pair Cable	210
4.2.2 Automatic Synchronization by Signalling	122	6.1.2 Principle of Total Internal Reflection	211
4.2.3 Line Coding Schemes	127	6.2 Fibre-optic Cables	211
4.2.4 Block Encoding	135	6.2.1 Design of Fibre Cables	213
4.2.5 Scrambling	138	6.2.2 Sending and Receiving Devices	214
4.3 Analog to Digital Conversion	140	6.2.3 Comparison between Unshielded Twisted Pair and Fibre Optics	215
4.3.1 Pulse Code Modulation	140	6.2.4 Other Cables—Coaxial and Shielded Twisted Pair	216
4.3.2 Delta Modulation	146		
4.4 Multiplexing and Demultiplexing	148		
4.5 Types of Multiplexing	149		
4.5.1 Frequency Division Multiplexing	149		

6.3 Physical Layer Based on Telephone Line	216
6.3.1 Telephone and Modems	216
6.3.2 xDSL	217
6.3.3 Discrete Multitone	219
6.3.4 Cable Internet	220
6.4 Synchronous Optical Network	221
6.4.1 Synchronous Optical Network Architecture and Components	223
6.4.2 Synchronous Optical Network Devices	223
6.4.3 Synchronous Optical Network Layers	224
6.4.4 Synchronous Optical Network Frames	224
6.4.5 Synchronous Transport Signal Multiplexing	225
6.4.6 Synchronous Optical Networks	227
6.4.7 Fault Tolerance in Synchronous Optical Network	228
6.5 Wireless Physical Layer	228
6.5.1 Two Special Cases—Hidden and Exposed Stations	230
6.5.2 Solution to Hidden and Exposed Station Problem	231
6.5.3 Components of Wireless System	231
6.6 Wireless Lan	232
6.6.1 802.11 Standard	232
6.6.2 802.11 Physical Layer	233
6.6.3 802.16 Physical Layer—WiMax	234
6.7 Wireless Communication Using Satellites	237
6.7.1 Geosynchronous Orbit	237
6.7.2 Other Orbits	237
6.8 Whitespaces	238
6.8.1 IEEE 802.22	239

7. The Data Link Layer

245

Introduction	245
7.1 Duties of Data Link Layer	246
7.1.1 Other Service Needs	246
7.1.2 Nodes and Connections	248
7.1.3 Media Access Control Layer	249
7.1.4 Wired and Wireless Data Link Layers	249
7.1.5 Link Layer Addressing	249
7.1.6 Unicast, Multicast, and Broadcast Addresses	249
7.1.7 Example of Communication between Network and Data Link Layer	250
7.1.8 Need for Address Resolution Protocol-like Solutions	252
7.2 Framing	254
7.2.1 Framing Techniques	255
7.3 Error Control	259

7.3.1 Types of Errors	259
7.3.2 Error Handling	260
7.4 Coding	261
7.4.1 Hamming Distance	262
7.4.2 Convolution Coding	263
7.4.3 Reed–Solomon Code	265
7.4.4 Low-density Parity Check and Turbo Codes	266
7.5 Error Detection	268
7.5.1 Checksum	268
7.5.2 Cyclic Redundancy Check	269
7.5.3 Analysing Cyclic Redundancy Check	272
7.5.4 Implementation in Hardware	274
7.5.5 Forward Error Correction	275
7.5.6 Correction vs Detection	280
7.6 Flow Control	280
7.6.1 Interfacing with Network and Physical Layers	281

8. Network Protocols

285

Introduction	285
8.1 Sender and Receiver Concept	285
8.1.1 Acknowledgement	286
8.1.2 Timers and Timeout Events	288
8.1.3 Sending and Receiving Windows	288
8.2 Retransmission	292
8.2.1 Duplicate Frames	293
8.2.2 Go-Back-N	294
8.2.3 Selective Repeat	297
8.3 Coding for Protocols	299
8.3.1 Prerequisites for Coding Protocols	300
8.3.2 Protocol 1	300
8.3.3 Protocol 2	301
8.3.4 Protocol 3	302
8.3.5 Protocol 4	305
8.3.6 Protocol 5—Go-Back-N	308
8.3.7 Protocol 6—Selective Repeat	308
8.3.8 Issues not Addressed	308
8.4 Real-world Protocols	309
8.4.1 High-level Data Link Control	309
8.4.2 Point-to-point Protocol	312

9. The Medium Access Sub-layer

320

Introduction	320
9.1 Shared Channel	322
9.1.1 Collision Detection and Avoidance	322
9.1.2 Channel-acquisition Issues	323
9.2 Channel Access Methods for Channel Sharing	323

9.2.1	Frequency Division Multiple Access	324
9.2.2	Time Division Multiple Access	326
9.2.3	Code Division Multiple Access	329
9.2.4	Space Division Multiple Access	334
9.3	Cellular Telephony	335
9.3.1	Generations of Cell Phones—0G to 4G	335
9.4	Wired Medium Access Control Layer	340
9.4.1	Prerequisites to Ethernet, ALOHA, and Slotted ALOHA	340
9.5	Ethernet	343
9.5.1	Classic Ethernet	344
9.5.2	Fast Ethernet	350
9.5.3	Gigabit Ethernet	351
9.5.4	10-Gb Ethernet	354
9.6	Wireless Medium Access Control Layer	356
9.6.1	Wireless LAN Protocol (802.11)	356
9.6.2	Distributed-coordinated Functioning Mode	357
9.6.3	Point-coordinated Functioning Mode	359
9.6.4	Managing Point-coordinated Functioning and Distributed-coordinated Functioning Modes Together	361
9.6.5	802.11 Transmission	362
9.6.6	Extensions of 802.11	366
9.7	Wireless Broadband	369
9.7.1	Wireless Broadband Protocol (802.16) Sub-layers	370
9.7.2	Medium Access Control Sub-layer—802.16	371
9.7.3	Service Classes	372
9.7.4	Generic Frame Structure	373
9.8	Connecting Devices at Data Link Layer	374
9.8.1	Bridges and Switches	376
9.8.2	Connecting Heterogeneous Networks	380
9.9	Virtual LAN	381
9.9.1	IEEE 802.1Q Standard	383
9.9.2	VLAN Operation for 802.1Q	383

10. Addressing in Internet

392

Introduction		392
10.1	Addresses	392
10.1.1	Types of Internet Addressing	393
10.2	Classful Addressing Scheme	393
10.2.1	Self-identification	395
10.2.2	Dotted Decimal Notation	397
10.2.3	Special Addresses	397
10.2.4	Subnets	399
10.3	Classless Inter-domain Routing	401
10.3.1	Slash Notation	403
10.3.2	Identifying Network ID in Classless Inter-domain Routing	403

10.3.3	Longest Match Paradigm	407
10.3.4	Unique Address Requirement	410
10.3.5	Network Address Translation	411
10.4	IPv6 Address	413
10.4.1	Managing Addresses in IPv6	413
10.4.2	Types of IPv6 Addresses	414
10.4.3	Address Assignment in IPv6	415
10.4.4	Administrative Improvizations in IPv6 Addressing	418
10.5	Automatically Allocating Addresses—Dynamic Host Configuration Protocol	421
10.5.1	Need for Dynamic Allocation of Addresses	421
10.5.2	Characteristics of Dynamic Host Configuration Protocol	422
10.5.3	Dynamic Host Configuration Protocol Data Format	422
10.5.4	Leasing Process and Management	425
10.6	DHCPv6	427
10.6.1	Managed and Unmanaged Address Assignment in IPv6	428
10.6.2	IPv6 Neighbour Discovery Protocol	429
10.6.3	Internet Control Message Protocol Messages Used by NDP	429

11. The Network Layer

436

Introduction		436
11.1	Duties of Network Layer	436
11.1.1	Unicast Routing	437
11.1.2	User Accounting Functions	439
11.1.3	Addressing Functions	439
11.1.4	Multiplexing and Demultiplexing Multiple Transport Layer Connections	442
11.2	Types of Forwarding	443
11.2.1	Connection-oriented Forwarding Using Virtual Circuits	444
11.2.2	Connectionless Forwarding Using Datagrams	446
11.2.3	Connection-oriented Forwarding Example	447
11.2.4	Comparison between Connection-oriented and Connectionless Forwarding	448
11.3	Routing Algorithms	450
11.3.1	Requirements of Good Routing Algorithms	452
11.3.2	Functions of Routers	454
11.3.3	Internal and External Routing Algorithms	457
11.3.4	Distance Vector Routing	458
11.3.5	Link State Routing	466
11.4	Routing Initiation Protocol	472

11.4.1	RIP Operations	472
11.4.2	RIP Messages	474
11.5	Open Shortest Path First	475
11.5.1	OSPFv2 Message Formats for IPv4	476
11.5.2	Hello Packet Format	477
11.5.3	Database Description Message Format	477
11.5.4	Types of Router Links in Open Shortest Path First	478
11.5.5	Link Status Request and Update Messages	480
11.5.6	OSPFv3 for IPv6	481
11.6	Routing in MANet	482
11.6.1	Ad hoc On-demand Distance Vector	483
11.7	Border Gateway Protocol	486
11.7.1	Border Gateway Protocol Operation	488
11.7.2	BGPv4	491
11.7.3	Path Vector Routing	491
11.8	Hierarchical Routing	493
11.9	Broadcast Routing	494
11.9.1	Individual Delivery	494
11.9.2	Flooding	495
11.9.3	Reverse Path Forwarding	495
11.9.4	Spanning Tree	496
11.10	Multicast Routing	496
11.10.1	Intradomain Multicast Protocols	497
11.10.2	Interdomain Multicast Protocols	501
11.11	Anycast Routing	501
11.11.1	ARP	503
11.11.2	RARP and BootP	504
11.12	Internet Group Management Protocol	505
11.12.1	IGMP Messages	505
11.13	IPv4 Packet Format	506
11.14	IPv6 Protocol	508
11.14.1	IPv6 Packet Format	509
11.14.2	IPv6 Extension Headers	510
11.15	ICMPv6	511
11.15.1	Error Reporting Messages	511
11.15.2	Informational Messages	512
11.16	Transition from IPv4 to IPv6	513
11.16.1	Tunnelling	513
11.16.2	Dual Stack	513
11.17	Congestion	514
11.17.1	Congestion Control	515
11.17.2	Congestion Control Algorithms	516
11.18	Network Layer Switching	520
11.18.1	Multiprotocol Label Switching	521
11.19	Internetworking Issues	526
11.19.1	Heterogeneity in Networks	526
11.19.2	Fragmentation	528
11.19.3	Tunnelling	530

11.19.4	Security Issues at Network Layer and IPSec	535
11.19.5	Overview of IPSec	535

12. The Transport Layer

545

Introduction	545
12.1	Duties of Transport Layer 546
12.1.1	Transport Layer Services 547
12.1.2	Services Not Provided by TCP 554
12.2	Retransmission, Round-trip Time, and Timeout Calculations 555
12.2.1	Retransmission 558
12.2.2	Ambiguous Ack 558
12.2.3	Other Timers 559
12.3	Connection Management at Transport Layer 563
12.3.1	Delayed Duplicates 563
12.3.2	Connection Establishment 570
12.3.3	Three-way Handshake 572
12.3.4	SYN and Ack bits in TCP 574
12.3.5	Connection Release 575
12.4	TCP Segment Format 578
12.4.1	Source and Destination Ports 580
12.4.2	Sequence Number 580
12.4.3	Acknowledgement Number 580
12.4.4	Data Offset 580
12.4.5	Flags 580
12.4.6	Window 581
12.4.7	Checksum 581
12.4.8	Urgent pointer 581
12.4.9	Options 581
12.4.10	Padding 581
12.5	Congestion Control 581
12.5.1	Detecting Congestions 582
12.5.2	Reacting to Congestion Using Random Early 582
12.5.3	Fast Recovery and Additive Increase Multiplicative Decrease 584
12.6	Comparison with Data Link Layer 589
12.7	Client–Server Communication 590
12.7.1	Problems and Solutions 591
12.7.2	Sockets and Client–Server Communication 591
12.7.3	State Transition Diagram 594
12.7.4	Sample Client–Server Program 595
12.8	Efficient Management of Dynamic Connections 602
12.8.1	Buffer Management 602
12.8.2	Crash Recovery 605
12.8.3	Performance Measures 606

12.8.4 Non-standard Performance Improvement Measures	610	13.4.4 Accessing File Transmission Protocol Using Menu-driven Programs and Browsers	685
12.9 SCTP	612	13.4.5 Anonymous File Transmission Protocol	685
12.9.1 SCTP Association	613	13.4.6 Secure File Transmission Protocol, Secure Socket Layer–File Transmission Protocol	686
12.9.2 SCTP Packet Format	614	13.5 Bluetooth	686
12.9.3 SCTP Operations	615	13.5.1 Bluetooth Architecture	687
13. The Application Layer	624	13.5.2 Bluetooth Applications	688
Introduction	624	13.5.3 Bluetooth Profiles	688
13.1 Domain Name System	625	13.5.4 Bluetooth Protocol Stack	691
13.1.1 Domain Namespace	626	13.5.5 Bluetooth Frame Structure	692
13.1.2 Registration Process	635	13.5.6 Pairing	693
13.1.3 Name Servers	636	13.6 Multimedia	693
13.1.4 Resource Records	637	13.6.1 Multimedia Protocols	695
13.1.5 Types of Resource Records	639	13.6.2 Real-time Protocol	695
13.1.6 Dynamic DNS	643	13.6.3 Session Initiation Protocol	698
13.2 World Wide Web and HTTP	646	13.6.4 H.323	701
13.2.1 HTTP Queries and Responses	648	13.7 Network Management and Simple Network Management Protocol (SNMP)	702
13.2.2 Structure of Queries and Responses	650	13.7.1 Simple Network Management Protocol Architecture	703
13.2.3 Other Methods to Query Server	650	13.7.2 Simple Network Management Protocol Versions	704
13.2.4 Persistent Connection with HTTP 1.1	652	14. Cryptography and Security	711
13.2.5 Cookies	654	Introduction	711
13.2.6 Session Variables	656	14.1 Cryptography	712
13.2.7 Conditional Download	657	14.1.1 Conventional Security Model	712
13.2.8 Proxies as Intermediaries	657	14.1.2 Substitution and Transposition	713
13.2.9 Dynamic Web	660	14.1.3 Symmetric Key Algorithms	715
13.3 Email System	661	14.1.4 Block Ciphers	718
13.3.1 Simple Mail Transfer Protocol and Components of Email System	663	14.1.5 One-time Pads	726
13.3.2 Mailboxes, Mail Aliases, and Alias Expansion	665	14.1.6 Quantum Cryptography	727
13.3.3 User Agent	666	14.1.7 Avoiding Random and Replay Attacks	731
13.3.4 Internet Mail Standard for Mail Content—RFC 2822	667	14.1.8 Cipher Modes	732
13.3.5 Message Transfer Agent and Simple Mail Transfer Protocol—Message Transfer Standard of Internet	668	14.2 Cryptanalysis	740
13.3.6 Internet Media Types	671	14.2.1 Attacking Block Ciphers	741
13.3.7 Comparison of SMTP and HTTP	672	14.3 Public Key Algorithms	742
13.3.8 Base-64 or Quoted Printable Encoding	673	14.4 Digital Signatures	749
13.3.9 Intermediaries Used in Mailing	675	14.5 Public Key Management	755
13.3.10 Post Office Protocol 3 and Internet Message Access Protocol	676	14.6 Authentication Protocols	758
13.3.11 Webmail	678	14.6.1 Authentication Based on Shared Secret Key	759
13.3.12 Filters and Spam	678	14.6.2 Shared Secret Key Using Hashed Message Authentication Code	763
13.4 File Transfer Protocol	679	14.6.3 Diffie–Hellman Key Exchange	764
13.4.1 Control and Data Connections	680	14.6.4 Authentication Using Key Distribution Centre	766
13.4.2 FTP Port Numbers	681	14.6.5 Authentication Using Public Key Cryptography	772
13.4.3 Seven-bit Network Virtual Terminal American Standard Code for Information Interchange—Format for Data Transfer	684		

14.7 Information Security	773
14.7.1 Vulnerabilities	774
14.7.2 Common Attacks	775
15. IP Security and Other Security Solutions	784
Introduction	784
15.1 IPSEC—Security at Network Layer	784
15.1.1 Working of IPsec	786
15.1.2 IPsec Design	789
15.1.3 Need for Internet Key Exchange Protocol	790
15.1.4 IP Security Policy—Security Association Database and Security Policy Database	794
15.1.5 Traffic Processing	801
15.1.6 Processing Outgoing Packets	803
15.1.7 Difference between SAD and SPD	804
15.1.8 Encapsulating Security Payload	805
15.2 Internet Key Exchange	809
15.2.1 Methods to Construct Cookies	811
15.2.2 Groups	811
15.2.3 Modified Diffie-Hellman Algorithm	811
15.2.4 Authentication Types	812
15.2.5 IKEv2 Protocol	812
15.2.6 CREATE_CHILD_SA Exchange	815
15.3 Introduction to Firewalls	816
15.3.1 Need for Firewall	818
15.3.2 Characteristics of Firewalls	819
15.3.3 Types of Firewalls	820
15.4 Introduction to Application Layer Security	830
15.4.1 Email	831
15.5 S/MIME	837
15.5.1 Cryptographic Message Syntax	838
15.5.2 S/MIME Message	838
15.5.3 Enveloped Data	840
15.5.4 Signing Mail	841
15.6 Intrusion Detection and Prevention	844
15.6.1 Complementing Firewalls Vital	844
15.6.2 Software Not Designed for Security	845
15.6.3 Classification of IDS	847
Index	852
About the Author	861

Introduction to Computer Networks and Data Communication

Learning Objectives

After studying this chapter, the reader will be able to understand the following:

- Fundamentals of data communication, data representation and transmission
- Layers and the layering mechanism
- The OSI and TCP/IP models, and the top-down and bottom-up approaches to study layers
- Functions of each layer and their implementation using HTTP and SMTP
- Distributed systems and how they differ from networks
- Peer-to-peer (P2P), and client-server networks and connection-oriented networks
- Communication standardization bodies
- Internet architecture and administration

INTRODUCTION

Why do we need networks? A quick response to that would be ‘resource sharing’. It is necessary to generate and disseminate information for completing many of our tasks. In group function, information should also be accessible to all the users in the group. The groups may be spread across a single organization or multiple organizations and at times across the globe. The following are a few examples:

1. Music files that are downloaded and uploaded onto the Internet or from a LAN
2. Emails that are sent and received around the world
3. Phone calls that are made or received using the Internet

All these services are bounded by a thread known as ‘network’, which is either a local network or the Internet. In all the aforementioned cases, resources are shared, and the network acts as a medium for sharing. Therefore, the prerequisite for all such services is the networking infrastructure. The following questions arise when all these services are used:

1. How can a file stored in a faraway network be downloaded to our machine?
2. How do our emails reach their intended recipients?
3. How is it that some of us have wired connections to the network, whereas some others have a wireless version of connections but they still work in the same way?
4. How is something that works without our knowledge, for example, the receipt of new data (like a new antivirus update), handled and by whom?
5. How does a social networking website update the status of our friends?
6. How does a cloud store and retrieve information?
7. How do peer-to-peer (P2P) networking sites enable users to communicate with peers?

We will find the answers to these questions, in addition to others, through this chapter. Before that, let us discuss another important component of communication—data.

1.1 DATA COMMUNICATION

Data with meaning is information¹. For example, 206 is data, but this value becomes information, when 206 is provided with additional details, for example, the number of runs scored by a batsman.

¹ Data when interpreted to convey a certain meaning is known as information. Data becomes information based on how one perceives it.
© Oxford University Press. All rights reserved.

Since information is shared, it should be ensured that everyone is able to communicate information to others. For example, if a teacher prepares a timetable that is to be shared by all other teachers, he/she must communicate it to the others. Social networking sites provide a strong and easy way to communicate such information. Thus, the purpose of communication is sharing of information; hence, the term data communication is used instead of information sharing. If we look at the classic definition, there is only a minute difference between the two terms.



Note: One may ask why is it that communication systems not transfer information but only data. Data is a simplified expression. The communicating system must generate an integer representation of 206 and transmit the same without any specific details, that is, whether 206 is a cricket score or roll number of a student. Such simplification helps in reducing the complexity of the overall operation and also reduces the burden on the system to retain the meaning of the data being sent and received. Thus, when 206 is sent, it is sent as 206 without the information of it being a roll number, the score of a batsman, or an apartment number. Hence, it is called data and not information.



The carrier is not aware of the meaning of the bits being transmitted across; therefore, it is called data communication and not information communication.

1.1.1 Delivery Problems

The process of data communication from a sender to a receiver encounters four problems which are listed in Fig. 1.1. First, the receiver may not receive the exact data sent due to delivery problems. Second, data may not be received by the intended recipient. Third, data may not be delivered in time but much later or earlier. Fourth, the rate at which data is received may not be constant, that is, it changes over a period of time. Data can be delivered at either a faster or a slower rate. The solutions to these four problems are discussed in detail in this chapter.

Incorrect data receipt This problem can occur due to the following two reasons.

1. Owing to technical reasons, the communication line and other intermediary components introduce errors.
2. It can be done intentionally by malicious human beings (i.e., hackers).

The problem of incorrect data receipt should be solved by considering both reasons. Solutions to both the problems are discussed in Chapters 7, 14, and 15 respectively.

Delivery to incorrect destination If data is not received by the intended recipient, the process of data communication is not complete. Usually, this problem can be solved by sending an acknowledgement from the receiver. An acknowledgement from receiver (R) about data (D) clearly informs the sender that receiver R has received the exact data D. Here, D does not represent the entire data, but a value is generated from it. The value is generated in such a way that if the receiver does not receive the same data sent by the sender, the value will change and the sender

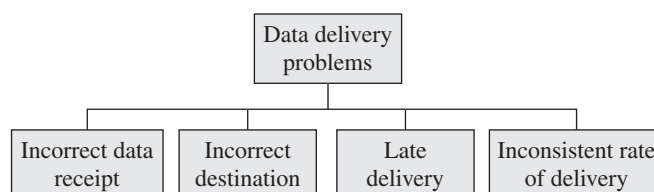


Fig. 1.1 Data delivery problems

will come to know about the same. To solve this as well as the previous problem, a method known as CRC is used, which is discussed in Chapter 7.

Late delivery The data must be delivered on time. If not, it might make no sense to the recipient. The following are some examples of such instances:

1. If the stock market price change is not reflected on the user's computer every few minutes, the user might incur huge losses.
2. If a video being sent to a user is unavailable for some time, the user might get frustrated and close the connection.

Thus, delivery on time is very important. There is an important difference between audio/video and mail being sent over a communication channel. When audio or video is being sent, the receiver's tolerance is much less compared to when a mail is being sent. Audio and video are examples of real-time transmission, whereas mail is an example of non-real-time transmission. It was clearly shown that real-time transmission is more prone to problems when data is not delivered in time.²

Inconsistent rate of delivery When data (which travels in the form of packets) arrives with different speeds at the receiver, it creates problems for real-time data. For example, a user is watching a cricket match when the video frames do not arrive consistently; the user will not be happy when the video is not smooth. It is important to understand the difference between late delivery and inconsistent delivery. For example, if every packet is delayed by 5 s, while this delay is a problem, the data is consistent. The user will be able to view the video late by 5 s, and once he/she starts watching the video, it will go on smoothly without any disruption. Data arrival is said to be inconsistent, when the first data packet reaches the receiver in time, but the second data packet reaches after a 5 s delay and similarly, the third data packet reaches in time, whereas the fourth data packet is received 2 s late.

1.1.2 Communication Components

In order to provide data communication, at least one sender, one receiver, and a communication medium are required. Figure 1.2 shows the components of communication besides the data. These components are discussed later in this chapter, but brief descriptions are given here:

1. Every communication has at least one sender. It is assumed that the sender will be in a position to generate information that interests the receiver and will be able to gather where exactly the receiver is placed and send it to the receiver.
2. Every communication has at least one receiver who is interested in the information that the sender is sending. The receiver is capable of receiving and consuming the information being sent.³
3. There is a communication channel that delivers the data being sent by the sender and received by the receiver. The communication channel may or may not be physical. All wireless channels belong to the non-physical type.
4. There are some rules that govern how the sender is going to send the information and how the receiver would process the same. This is because if both of them do not interpret the data being communicated in a similar fashion, there will be miscommunication between these two parties

² There is an interesting point to note here. Retransmission does not help real-time traffic. We will learn why when we describe connectionless and connection-oriented data transfer.

³ Here, the word information is used by the sender and receiver, who definitely know what they are sending and receiving. They can clearly distinguish that the first value of 206 is the score of a batsman, whereas the second one is the apartment number.

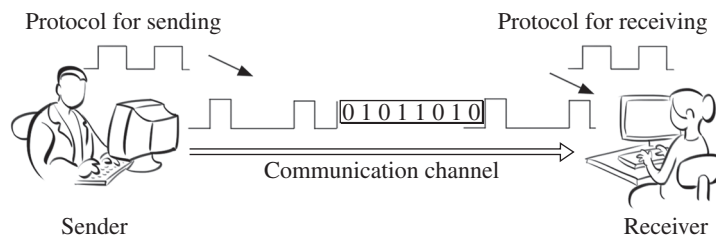


Fig. 1.2 Communication components

rather than communication, which defeats the purpose of this exercise. This set of rules is called *protocol*.

1.1.3 Data Representation

This section discusses the many ways in which data can be represented. For example, arrays can be used to represent homogeneous information (e.g., storing information about multiple students belonging to the same class) or a linked list to represent data that is dynamic in nature (e.g., processes that are kept in ready state). Similarly, the data that is being communicated requires standard representation so that the sender and receiver have no ambiguity in understanding what is being sent. For example, data like 'A024798BEF...0234591D' can be an image file stored in hexadecimal format using the jpeg format. Unless the receiver learns that the hexadecimal value it receives is a jpeg file, it will not be able to produce it in the same format to the user. Therefore, data representation should be standardized and understood by all those who are participating in the communication.

Data communication can be optimized with suitable data representation. Standardization of data representation also helps the sender and receiver to unambiguously interpret the data.

Some of the data representations are explained as follows:

Text Text is the most common form of data transmission. Computers are not capable of storing data as text. A computer system can only store information in the form of bit patterns and thus requires some form of conversion from text into a bit pattern.

Numeric values An integer number is usually represented by its binary form, whereas a floating point number is represented by storing both the mantissa and exponent separately. When a number is represented this way, for example, representing an integer as a binary value, it helps mathematical operations to be easily carried out as no data conversion is involved.

Images An image can be represented in many ways, including popular jpeg, bmp, and gif formats. An image is conventionally a collection of pixels. Each pixel represents a point in a graphic image. The same image can be represented by a different number of pixels. The number of pixels used to represent an image is known as resolution of that image.

Audio and video Technically, audio and video represent analog content. They are not similar to other types of content. For example, some representation of text might have a value 0 or 1 but not 0.5 or 0.7. However, an analog signal can have any value in between. Both analog and digital transmission and signalling are discussed in Chapter 3.

1.2 DATA TRANSMISSION MODES

Data transmission can occur in three different ways (Fig. 1.3). They are called simplex, half-duplex, and full-duplex modes, which are discussed here.

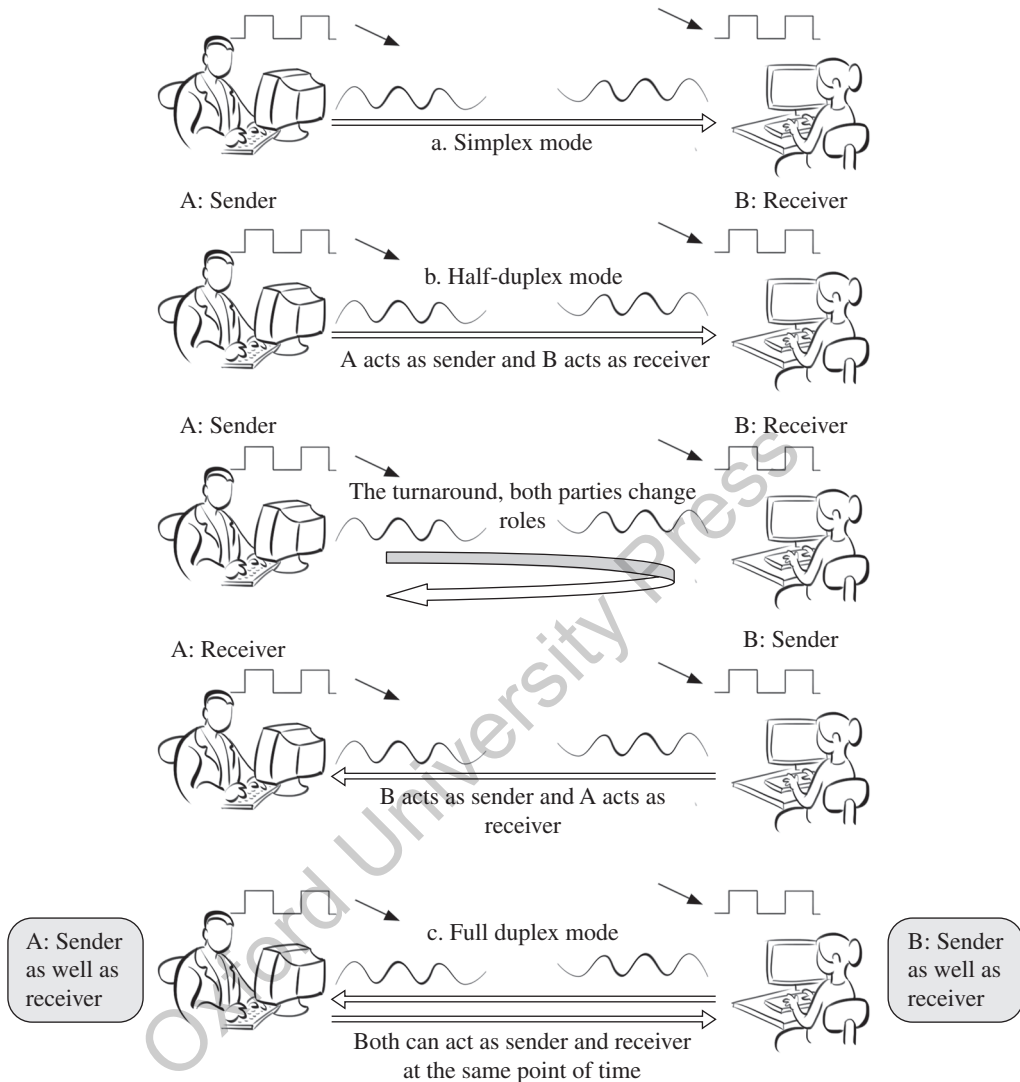


Fig. 1.3 Communication modes

1.2.1 Simplex Mode

This is the simplest form of transmission. Only one party sends, whereas the rest listen. The mouse connected to a computer is an example. The mouse can only send the signals and the computer connected at the other end can only receive. A keyboard is another example of a simplex sender. On the contrary, the printer is an example of a simplex receiver⁴.

1.2.2 Half-duplex Mode

If you have watched old spy movies, you probably have an idea of what a half-duplex mode is. A spy with a walkie-talkie speaks over his/her instrument and conveys 'over' to indicate that he/she will

⁴ Technically, printers do send back some information to computers they are connected to for control purposes but do not send any data back.

switch over to listen mode now. The other end will start speaking now and then take turns to listen by indicating 'over'.

A half-duplex mode is a mode where both parties can send information, but not at the same time; they take turns. When one party finishes, it sends an indication to the other end, so the party at the other end can start sending. The half-duplex communication has three different phases.

1. Sending (one of the parties sending data to the other end)
2. Turnaround (i.e., changes in the direction of transmission)
3. Another party sending data

The earlier versions of GSM phones and ethernet cards are a few other examples.

1.2.3 Full-duplex Mode

Full-duplex mode is the most common type of transmission. At a time, both parties send and receive. Conventional telephone communication⁵, client-server communication in networks, and many other forms of communication that we see are all of the same type, the full duplex⁶. Many times, a full-duplex channel is inherently a combination of two simplex channels in different directions. For example, a usual fibre-optic full-duplex communication has two fibre-optic cables designed to carry simplex traffic in different directions.

1.3 CONCEPT OF LAYERING


Divide and conquer is a well-known mechanism to solve complex problems. The entire networking problem is solved in pieces, each of which is independent to a large extent. The fact that the evolution of one piece does not hinder the evolution of other pieces is the advantage of this division. All such solution pieces are called *layers* in networking parlance. All these layers are arranged from top to bottom for the purpose of pedagogy. In this arrangement, each layer provides a service to the layer above it. Here, each layer represents a specific networking service, which is more or less independent of other services. A simple analogy to understand the concept of layering is discussed in the following section.

1.3.1 Layering Example

Assume a company ABC, which sells computers, has a manager, a secretary, a route operator, a warehouse keeper-cum-security officer, and a transporter with a few trucks located at a place called A. Suppose the manager receives an order for 5000 units from a company XYZ located at a place B.

He/She conveys the message to the secretary in Hindi as both are comfortable with Hindi.

Now, the secretary writes a letter in English that has to be carried with the consignment, addressed to the secretary of XYZ. The letter indicates that the consignment of 5000 computers is sent. It also requests the recipient to send an acknowledgement after receiving the consignment and letter. This letter will only be opened when it reaches the desk of the secretary of XYZ.

 Layering in networks helps conquer a complex problem by dividing it into independent, simpler modules.

⁵ A mobile phone communication seems like a full-duplex communication, but technically it is half duplex on different frequency bands for GSM. A mobile device that is sending voice is not receiving at that point of time and while receiving, it stores and refrains from sending what the speaker is speaking.

⁶ One wonders why people ever use half duplex when both parties are interested in communicating. The major reason is that half duplex uses the same frequency band for sending and receiving. As it is not sending while receiving and vice versa, it can afford to use the same band for both and thus saves on frequency. When the frequency is rare, as in the case of old walkie-talkies, half-duplex mode is more effective. In addition, building radios with half duplex used to cost less once upon a time.

Then, the secretary instructs the route operator to deliver the consignment to XYZ. The secretary and the route manager may discuss everything in Punjabi.

Now, the route operator's job starts. Let us assume he/she has the map given in Fig. 1.4, and he/she tries to derive a route for the consignment looking at the map.

From the map shown in Fig. 1.4, ABC and XYZ are located far away from each other. They are denoted A and B, respectively, in the map. To go from A to B, there are three junctions to cross. The three junctions are R1, R2, and R3. The paths from A to R1 and R3 to B are roads where trucks carry the consignment. From R1 to R2, there is a railway track. Between R2 and R3, there is a river flowing in between. A boat is used to go from R2 to R3. There are four different hops (junctions) through which the consignment should travel.

Assume ABC has its transport offices at R1, R2, and R3. The route operator is ideally aware of the path and map. So he/she can decide to send the consignment to R1 whenever the destination is B. The map contains only one path, but in a real case, there may be more than one path between any two places in the map. If the terrain suffers from heavy traffic jams, he/she would need the latest map that displays only those paths that are available and not others where there may be traffic jams. For the time being, ignore all such complexities and assume that the entire path is open for communication. Thus, the route operator decides to follow the path shown in Fig. 1.4.

When the route operator thinks that the path outlined in the map is the route to be chosen, he/she must manage to send the batch of computers to R1. R1, being an intermediate destination, must be able to forward the consignments sent to it.

Now, the route operator, after deciding to send the consignment to R1, tells the warehouse keeper-cum-security officer to collect 5000 computers from various warehouses and send them to R1. He/She also asks to paste a label on each component of the consignment as 'From: ABC' and 'To: XYZ'.

Now, the warehouse keeper-cum-security officer decides to send these 5000 computers from the warehouse and decides that one truck can carry 500 such computers together. He/She prepares 10 big boxes that can carry 500 computers each. Now, each truck can carry one such box. Each box is labelled as 'From: ABC' and 'To: R1'. It is interesting to see that the computers themselves are also labelled but as 'From: ABC' and 'To: XYZ'.

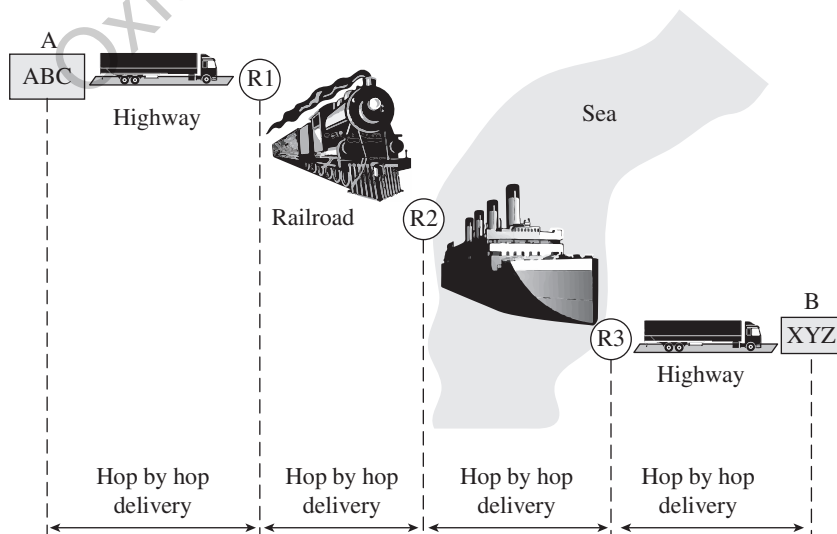


Fig. 1.4 Route of consignment from point A to point B

The label on the box also contains details about the computers as ‘Quantity (500)’, ‘Checked by’ (the name of the person who checked them), ‘Content’ (computers), etc.

Now, the warehouse keeper instructs the transporter to carry the consignment to R1. The transporter now instructs the truck drivers who carry them one by one to R1. They load the consignment at A and unload them at R1. They may use any truck that is available for the job.

The transporter at R1 receives the trucks, inspects its contents, checks if the destination is on his/her network before doing so, and verifies the sign on the container. Then, he/she opens the box and verifies the number on the container box. If everything is in order, he/she passes these computers to the warehouse keeper to store them until the scheduled train arrives⁷. The warehouse keeper at R1 contacts the route operator at R1 to decide the route. The route operator confirms that the branch office is not XYZ, so it is not the destination. Then, he/she finds out the real destination of these consignments.

The route operator is in contact with other route operators and is aware of both train schedules and other modes of transport (which we are ignoring at the moment). He/She chooses the train as the best mode to transport the consignment. It is possible that the train runs just once in a week, and the computers are needed to be stored for a few days. He/She passes this information to the warehouse keeper. The train compartments are of different sizes, so the warehouse keeper needs to pack 1000 computers in a single compartment, and gets five such compartments booked till R2. The warehouse keeper passes this information to the transporter at that junction R1. The transporter now manages the loading process at R1. The transporter at R2 should manage the unloading part.

Similar processing occurs at R2 and R3 and eventually the consignment reaches XYZ. On receipt of the consignment, the route operator at XYZ reads the receiver’s address, he/she comes to know that the address is his/her own, and also finds out from the content that a specific secretary has to receive the consignment and acknowledge the sender. The secretary at XYZ reads the letter from ABC’s secretary and writes a reply confirming the delivery of computers. The secretary at XYZ also passes this information to the manager. The acknowledgement letter will reach the secretary of ABC, and then, the manager of ABC will come to know that the order is completed.

Now, let us discuss how this simple example helps us understand the advantage of having layers in a networking environment.

The work of each employee is clearly defined. What the manager expects from a secretary is also clear. In the aforementioned case, the manager passes information pertaining to sending the computers to the secretary and expects an acknowledgement after the order reaches its destination. Similarly, the secretary makes sure that the secretary at the other end gets the delivery letter and sends an acknowledgement. Two points about the description are worth noting.

1. Every employee is either providing or taking a service from somebody else, or doing both. The manager takes the service of the secretary, the secretary takes the service of the route operator, the route operator in turn takes some service from the warehouse keeper, and the warehouse keeper from the transporter.
2. The example does not fit into real-world scenarios. We do not have route operators or warehouse keepers to handle packing or unpacking jobs. The transporter usually does the job.



Note: Though this contrived example tries to provide an analogy to the actual real-world scenario, it is not complete. No analogy can completely describe the real case.

⁷ If the transporter thinks it is not possible for the consignment to get damaged, he/she can let it go as is.

1.3.2 Advantages of Layering Mechanism

Some of the networking applications encountered in day-to-day life are as follows:

1. Telnet for remote logging⁸
2. File transfer protocol (FTP) for downloading or uploading a file⁹
3. Web browsers like Firefox or Internet Explorer to access a website or email
4. Social networking applications (e.g., Facebook) to send and receive data from other members of the network
5. Exact place to visit and the path to it using Google Maps
6. Storage and retrieval of information stored in clouds (using cloud client programs) without having to worry about where they are stored

The above applications are also made up of a few layers, each of which carries out a specific function in the overall mechanism. The advantage in the layering mechanism is that the work of each layer (employee) is clearly defined, thereby reducing the complexity of the overall mechanism.

These programs instruct the transport layer entity, a secretary (in most cases, the TCP or transmission control protocol), to do the required job. The TCP in turn passes it down to the Internet protocol (IP), a routing operator which decides how to route the data to the destination. Social networking and cloud operations are harder jobs to be carried out as they require multiple clients and multiple servers to interact with each other at the same point of time.

The data link layer and the physical layer reside on a device called network card¹⁰, which does the job of framing and is similar to putting the data in a box by the warehouse keeper with specific addresses written on the label. The physical layer is similar to the transporter that transports the data, either using wires of different types or in a wireless manner (such as trucks, railway compartments, or boats). The physical layer manages to transfer the data to an intermediate location (R1, R2, and R3 in our case).

It is easy to devise protocols for a special layer (e.g., employing somebody in the hierarchy with specific instructions to work) without bothering much about other layers. We will soon understand what protocols are. For the time being, let us take it as a mechanism to solve a problem. This mechanism is standardized in the sense that both parties involved are aware of it. The complexity of the entire system is divided into multiple modules of less complexity.

Apart from reducing the complexity, the layering mechanism offers other advantages as well. They are discussed as follows:

Division of work Every layer works on what is assigned to it. In the given example, the security officer-cum-warehouse keeper is responsible for packing and unpacking the goods. He/She also ensures whether the goods are ready to be dispatched, and verifies and signs the consignment. The route operator decides the immediate or final destination and the transporter manages the sending, loading, and unloading of goods. This is what we mean by division of work. We will study in the following chapters that the data link layer does a job similar to the warehouse keeper and the physical layer works like the transporter.

Standard interfacing between components As discussed earlier, when the secretary (read transport layer or TCP in Internet) writes a letter to another, he/she may follow the language that both of them have agreed upon, that is, English in our case. Now, this enables anybody who knows English

⁸ Telnet is a utility to log in to a remote machine, sometimes on a remote network.

⁹ FTP is a utility to download and upload files from local or remote networks.

¹⁰ Ethernet card is a popular network card that is a de facto standard for wired networking today. Wireless network cards almost do the same thing.

to interpret the message of the secretary. Similarly, a transport layer can interact with another transport layer if both of them use the same language (read protocol). It is interesting to note that the language used by the manager while interacting with the secretary or route operator is different from the language used by the secretary to communicate with another secretary.

The communication that takes place between different layers of the same communication stack is known as *interface*. For example, a transport layer of a system interacting with a network layer of the same system is known as an interface (e.g., in our analogy, the communication between a manager of ABC and a secretary of ABC is an interface), whereas when a layer interacts with a peer layer of some other system (in our example, the secretary of ABC to secretary of XYZ), the communication is described as a *protocol*. The advantage of this mechanism is that it is possible to standardize the language for communication between peer entities of different systems (e.g., if the secretary is instructed that he/she has to always communicate with other secretaries in English, any secretary who can communicate in English will do irrespective of the language used in the organization).

Easy replacement of components The IP is the route operator for Internet. Currently, IP version 4 (usually referred to as IPv4 in short) is used. This is being replaced by a new version 6 usually referred to as IPv6¹¹. Suppose we are running IPv4 on our computer and we now switch over to IPv6, do we have to change the browser? Do we need to remove our ethernet card and have one that supports IPv6? Technically, the answer is a plain ‘No’. This is a direct consequence of standardization. This makes replacement of components an easy job. Similarly, when one replaces an ethernet card with a higher-end ethernet card, for example, replacing a 100 Mb card with a 1 Gb card, one need not change or reinstall the SMTP, FTP, Telnet (application layer protocols), TCP (a transport layer protocol), or IP (a network layer protocol). The only change is the ‘driver’ of the card to be installed to make sure the operating system can work with the card in a seamless manner.

Independence in protocol design When one is designing an application layer protocol for a new application, say a customer support application where customers from all over the world can interact with the system and get the status of their orders, does one need to constrain the customers with what the specifications of their transport layer protocol, ethernet cards, or the operating system should be? The answer is ‘No!’ If the application is based on TCP, UDP, or SCTP, the customer should be told to run that protocol¹². No other instruction needs to be given. The protocol (for the time being, assume protocol to be the language and sequence of the messaging needed to communicate between peers) between the client and server of the system can be developed independently of other protocols used by other layers. The manager and secretary interacting in Hindi do not have any effect on the secretary who writes in English to the other secretary. Similarly, the secretary of ABC talking to the route manger in Punjabi does not have any impact on the secretary at XYZ talking to her own route operator in Marathi. Similarly, nothing is going to stop a transporter talking to another transporter in some other language of their choice.

We have seen some of the important advantages of the layering system; most of them are so important that it is commonly observed to have layered network architecture on the Internet. Having said that, there exist some disadvantages as well that make the layered architecture not suitable for some applications.

1.3.3 Disadvantages of Layering

Some of the disadvantages are listed here.

¹¹ We will study both of them in Chapter 8.

¹² Almost all machines, either a part of the Internet or a part of any other network, run these protocols. They are almost always installed in the customer’s machine and do not require any special installation process.

Not efficient for small problems Layering architecture is inefficient when applied to small problems. It is like dividing a small work into pieces. When a work is divided, it produces two additional overheads. The first is that somebody must divide the work, assign them to different workers, get it back when completed, and collate them. This overhead defies the advantages that we obtain from layered architecture.

Synchronization The second problem is to synchronize and regulate the function of all workers. It is more important when multiple employees are working in a system. Assume the route operator has to manage multiple consignments from various offices, and each consignment requires a different kind of treatment. Take the example of a consignment of crockery that is to be handled with special care. A consignment of medicines requires temperature regulation; a consignment of perishable items like food requires quick delivery, and so on. The route manager must synchronize all such route requests with the appropriate quality of service required for such cases. An application layer must be in a position to provide specific instructions to all other layers below to follow if such requirements are to be observed.

Minimizing inter-layer communication The advantages of independence of layers have been discussed earlier. In order to preserve this independence, this inter-layer communication should be avoided (as discussed earlier, this is called interface). Therefore, a certain amount of independence should be maintained between layers. At the same time, layers cannot remain blind to what other layers are doing. Therefore, possible independence should be provided and minimum internal communication is allowed between different layers of a single protocol stack.

Providing synchronization and independence to layers adds extra burden on the programmer and complicates the design of the overall system. This is one disadvantage of the layering mechanism.

Reduced speed and performance When the work is handled by multiple layers, its speed of execution gets reduced as the process is executed in pipeline by various layers. The speed of execution is dependent on the slowest layer in the pipe. When inter-layer communication is minimized to preserve the independence of layers, it becomes more time-consuming for a layer to judge the environment. In the aforementioned case, if the secretary has got no information about the traffic details from the route operator, he/she must assume the route is congested. He/She must employ judicious mechanisms to handle the situation. The execution of such mechanisms takes additional time and hinders the performance.



Layering is not a panacea. In some cases, layering is not recommended as it adds more overhead than the advantages it provides.

Increased memory usage In a multi-layered architecture, different stages of a process running at different layers (e.g., in the Internet, applications such as SMTP, FTP, and Telnet run at the application layer; at the same point of time, the TCP and UDP run at the transport layer and IP runs at the network layer) have to be stored and retrieved repeatedly as and when a particular process running at a specific layer starts, stops, and resumes its operation. Therefore, the system requires more storage space (memory) to execute more number of processes.

Not suitable for sensor networks The sensor network is a network of tiny computing units known as sensors. Sensors are devices with limited memory and processing power and usually run on a battery that is required to be used optimally. Such an architecture demands less layering and more compact solution. This is the reason why the sensor networks do not follow strict layering.

1.4 TCP/IP AND OSI LAYERING MODELS

There are two layering schemes available to network designers. The OSI layering mechanism was designed by the Open System Interconnection group from the International Standards Organization (ISO).

The OSI model employed a seven-layer scheme that was heavily influenced by a model known as systems network architecture (SNA) from IBM. On the contrary, the TCP/IP model is a retrofit to a practical solution provided by the Internet community. The OSI model is almost non-existent in terms of its implementation. Almost all the machines use the TCP/IP model for communication. In this book, TCP/IP model is used for discussion and whenever necessary, we may consider the OSI model as well. Let us try to understand the difference between these two models by means of an example.

Suppose we are planning to arrange a birthday party. There are two approaches to go about that. One is an ad hoc solution mechanism (read TCP/IP model). Just go ahead and organize the party with whatever resources you have at your disposal. Learn what the problems are and improve the functioning in the next party. Our job will improve every time we arrange a party. After a while, as our expertise in planning increases, arranging a birthday party would not be a hassle. We know that the cake is the most important thing to have, followed by a knife and a few candles. We can go without other decorating material and still arrange a birthday party. The emphasis here is on experimenting and learning. Though ad hoc, this mechanism is quite practical and is also a working solution. This mechanism also lays emphasis on taking advice from everyone who is involved in the process, and improving the solution by multiple trial runs.

The other mechanism (read OSI model) is to plan everything in advance taking advice from all the wise people around, almost none of them having any experience¹³ of arranging a birthday party. The decision is highly influenced by the individual having (so-called) experience. It is also possible that somebody who does not have any idea about what a birthday party is starts giving advice and we have to listen to him/her as he/she is an elder family member. People without any experience may provide serious changes to the design. It is quite possible that such a system will have the cake and other decorative elements but no knife or candles.

The first solution is not a great design, but sometimes incorrectly designed solutions are good enough. It is possible to improve them without much modification. The second solution is a great design on paper, but fails to work as no practical feedback is provided. The first solution might not have the decorative elements but will have all the essential elements required to run a birthday party. On the other hand, the second design may flop miserably even though a great entrance is prepared and a big cake is ordered. A smart person will go for the first solution rather than the second one. A smarter person would filter the good elements of both solutions and combine them. Unfortunately, we have to choose only one solution. All the network users are smart enough, as a result of which OSI is almost not in the fray.

Fortunately, for discussing networks, it is good to combine both models. That is exactly what we have tried in the following text.

1.4.1 OSI Model

The layering mechanism of OSI model contains seven layers. They are detailed in Fig. 1.5. It contains two extra layers as compared to the TCP/IP model, that is, the presentation layer and the session layer. These two layers are like having a decorated entrance to the hall where the birthday party is arranged and allocating an attendant to every guest to help them around. One may like to have them around, though having other things are much more important. If we miss on the entrance or the attendant, we do not lose much.

¹³ Experience has nothing to do with wisdom. However wise you are, you need experience to master something; for example, take the case of driving a car on crowded roads. Running a network with a variety of users and their demands is just like that. OSI is precisely the example to illustrate that point. The committee members represented well-known companies who knew a lot about computers, but they had limited knowledge about the network and the problems related to it.

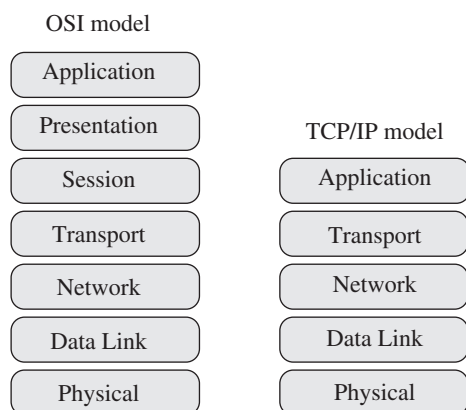


Fig. 1.5 Layers of OSI and TCP/IP models

The OSI model is very elaborate but does not work; the TCP/IP model works but is not elaborate.

The presentation layer is designed to check minor differences in the way the data is presented by the sender and the receiver. For example, the sender and the receiver may have different ways of storing integer values in the memory.

The session layer is designed to manage sessions between the sender and the receiver. Both functionalities (the presentation and session layers) are too small to deserve a layer. The TCP/IP model does not have either of them.

Additionally, the TCP/IP model does not even mention anything about the data link or the physical layer (which exists in the OSI model), but all the network cards used for networking have them; hence, they are mentioned here.

1.4.2 TCP/IP Model

As shown in Fig. 1.5, the TCP/IP model contains only five layers. The two layers at the bottom, that is, both the data link layer and the physical layer, are not actually a part of the model, but practically they are always present. This model is a retrofit to the working solution found by researchers, academicians (designed by teachers and programmed by students!), and some people from the military. Thus, the model was designed after the protocols were designed and in working condition already. Before we discuss how the TCP/IP model differs from the OSI model, let us elaborate on the difference between connectionless and connection-oriented transfer and a few other important concepts.

1.4.3 Connection-oriented vs Connectionless Transfer

When we make an international call, the telephone line, including the expensive intercontinental trunk, is occupied for the duration of the call. This is an example of connection-oriented solution. It initiates with establishing a connection, followed by the transfer of voice and finally closing the connection. Connectionless transfer, on the contrary, omits the first and the last step, initiating and closing a connection. Almost half of our verbal communication is punctuated with pauses without which our sentences do not make sense. The problem with connection-oriented transfer is that it keeps the line busy even though there is no data transfer.

The connectionless mechanism does not establish the connection beforehand, so the entire line is not occupied throughout the duration of the call. The only line that is transferring our packet (the voice travels as small digitized chunks known as packets in connectionless mechanism) is occupied. Consider the example depicted in Fig. 1.6 to understand connection-oriented communication.

The caller calls from Ahmedabad. The call is connected to an exchange at Vadodara. From Vadodara, it gets connected to Mumbai from where an international trunk operates it through Port Elizabeth. The ultimate destination, Johannesburg, is reached via an exchange at Bloemfontein. It is assumed that the traffic flows through this route, which is geographically correct; however, telephone companies might follow another route. For our purpose, the route is not going to make much of a difference, so we will not be worried about that. Now in case of telephone lines, the entire line from Ahmedabad to Johannesburg is occupied, conceptually.¹⁴ Now, let us consider the connectionless transfer. In this case, the packets being transferred work in a different way. When the caller

¹⁴ The line may not be completely occupied as it is shared between multiple callers, but some specific portion of the line is always set aside for us when our call is on. The normal standard is to use the pulse code modulation technique, which requires reserving 64 Kb for a single voice call throughout the communication. The intercontinental fibre-optic cables have the capacity to carry traffic in the range of terabytes and thus are capable of managing many calls in parallel.

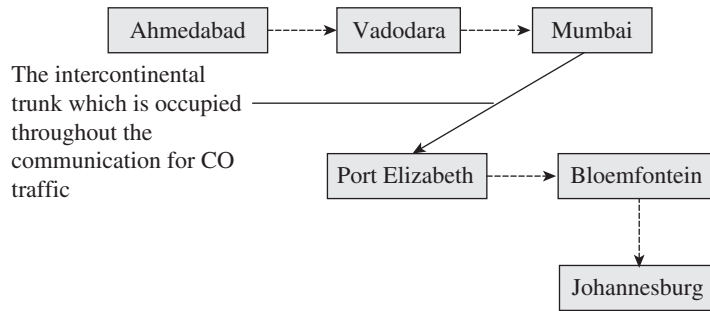


Fig. 1.6 Difference between connection-oriented and connectionless communication. A path from Ahmedabad to Johannesburg passing through various links including the expensive intercontinental trunk Mumbai to Port Elizabeth link

calls from Ahmedabad, a specific portion of the line¹⁵ from Ahmedabad to Vadodara is occupied, and when the packet reaches the destination, it is released. When the packet starts travelling from Vadodara to Mumbai, the line connecting only these two cities is used. When the packet reaches Mumbai, the line between Vadodara to Mumbai is released. Similarly, as soon as the packet reaches Port Elizabeth, the portion is released for other use. Here, only local lines are occupied.

Therefore, the advantage of connectionless scheme is that the lines can be used for other callers when we are not using that line and thus we have to pay less. There are two more advantages of this scheme. First, the packet, being in digital form, can be compressed, which can be as less as 8 Kb (Kb in networking parlance is kilobyte, i.e., 1000 bytes and not 1024 bytes) per second. Normal telephone lines use 64 Kb per connection¹⁶. So the requirement is eight times less. The other advantage is that the packet is multiplexed with other packets. Thus, the sender packs as many packets as possible on the line. If we are not speaking and not generating packets, then the line is free to send other packets at that point of time. That means we are not paying when we halt to take a breath while talking! Now, we can understand why calling through the Internet is cheaper. It uses connectionless communication unlike conventional telephone calls.

To summarize, connectionless transfer is cheaper and more robust. These are the advantages of connectionless transfer.

The connection-oriented mechanism has its own advantages, some of which are as follows:

1. When the line is reserved, we get guaranteed quality in voice without any delay and compression. The Internet, by design itself, does route each packet independently. It makes each packet reach the destination with delay differences that create distortion in voice. (If we have to sing songs, may be for telephonic Indian idol, we should choose a normal telephone line and not an Internet connection!) The quality of voice through the Internet is poor as compared to telephone lines due to the compression and delay differences between packets.
2. When we use the connection-oriented mechanism to transfer data, for example, while browsing the Net, downloading a file, or remote logging to a server using Telnet, the connection between the sender and the receiver is established before the actual communication starts. Thus, every intermediate router comes to know about that connection beforehand. So resources like buffers are reserved in advance and better service can be provided.


¹⁵ The portion of the channel occupied depends on the compression algorithm used. Some algorithms are capable of reducing the requirement to as small as 8 Kb.

¹⁶ The telephone line connected to our house is analog. The transfer of our call between different exchanges converts the analog signal into digital using a technique called pulse code modulation. Here each call traditionally uses 64 Kb.

3. One more advantage that is always associated with a connection-oriented system is the *order*. A connection-oriented system is always expected to maintain the order of packets. The packet that is sent first is received first.
4. Another advantage is *reliability*. As the entire line is at our disposal, we are not going to be bothered till the connection is over. In connectionless transfer, a single greedy user sharing a line with us can make us feel the difference. He/She can start pushing a lot of traffic into the network and slow down all other connections considerably.

The connection-oriented scheme is better and more reliable for the sender and the receiver but is more troublesome for the underlying network


It is also possible to provide dedicated (connection-oriented) service over a connectionless connection. The TCP that we will explore further in Chapter 8 is the layer that is responsible for providing a reliable service when the underlying network layer (IP) is designed to be connectionless. The other candidate for the transport layer on the Internet is the user datagram protocol (UDP), which does not provide a reliable connection-oriented service.

 The Internet prefers a connection-oriented service over a connectionless delivery mechanism for most applications such as mail, web, and file transfer.

TCP is an example of connection-oriented service over connectionless transfer provided by IP, the network layer protocol, which is connectionless. When data is given to send across the network, the TCP divides it into segments or data chunks. It instructs the IP to send those chunks across. Being connectionless, the IP sends it across without bothering to have a connection at the other end and does not care if data is lost in transit or not received in order. The TCP layers of the sender and the receiver talk to each other to figure out lost segments and their order, and ensure that the lost segments are retransmitted and given in the same order as they were sent.

Circuit and packet switching Conventional telephone lines use a method called circuit switching to connect caller to callee. How it is done? Each telephone exchange has a physical structure to connect the incoming connection line to an outgoing connection line. The capacity of an exchange is determined by the number of such lines. That means if an exchange has a capacity of 25,000 concurrent operations, 25,000 callers of that exchange can talk to the other end at the same point of time. If the 25,001st customer tries to dial in, he/she will get a busy tone.

Telephone exchanges work as follows. For an incoming call, the exchange tries to find a free outgoing line. If it finds one, it will connect the incoming line to an outgoing line (almost) physically and does not let it disconnect unless the call is over. (You might find the discussion similar to what we discussed earlier. That is correct; we are discussing the same concept.) It is called circuit switching and is closely associated with connection-oriented services. Telephone lines use circuit switching to provide connection-oriented services to their customers.

 Data communication prefers packet switching.

On the contrary, packet switching is usually used in data communication. Packet switching divides the message into short packets and sends the packets one by one. It does not occupy the entire line in advance, and only the immediate sending and receiving packets are involved. The packet-switching process is useful for data communication as it does not monopolize the channel and the throughput (utilization) of the channel is far better compared to circuit switching.


Now you can understand that the example seen earlier is not only about connection-oriented versus connectionless transfer but also involves both circuit and packet switching. Table 1.1 lists the differences between telephone lines and communication lines.

Connection and switching The connection-oriented service usually deploys the circuit-switching technique as it is easier to manage. On the contrary, the packet-switching technique is better for connectionless transfer.

Table 1.1 Telephone lines vs data communication lines

Telephone lines	Data communication lines
Telephone lines are designed to carry voice.	Data communication lines are designed to carry data (files, web pages, mails, etc.).
These follow circuit switching and thus provide excellent voice quality but with a cost per unit time.	These lines follow packet switching, and thus, cost is based on volume and not time. Voice quality is poor.
Telephone lines involve telephone exchanges to route telephonic communication.	Data communication lines involve routers for a similar purpose.
Circuit switching does not allow change of exchanges for the entire communication. Thus, all exchanges that are part of the communication route at the time of connection establishment will remain the same till the end of communication.	Routers of these lines are free to decide the next router to send, and thus, the path may vary for each packet.
Telephone lines carry data or voice using analog signalling in the local loop.	Data communication lines usually carry data or voice using digital signalling.
These are used to carry voice (e.g., VOIP).	These have been used to carry data using dial-up modems for a very long period but now DSL is being used. They also carry voice using VOIP.

The connection-oriented delivery requires all intermediaries to be informed well in advance about the connection. This process usually happens during the connection establishment process. Another important point that happens during a connection establishment process is feasibility checking. For example, if a connection is set up to render online video, it might require say 1.5 Mb of sustained bandwidth. If any one of the intermediaries is not able to provide that service, the connection cannot be established. The other important characteristic of circuit switching, that is, occupying the entire line throughout the connection helps meeting the specifications for the connection like the one mentioned earlier.

 In the TCP/IP model, IP implements connectionless delivery, whereas TCP provides connection-oriented service on top of IP. UDP is another candidate that provides connectionless service on top of IP.

Service vs delivery Service and delivery are two important terms with respect to connection. A connection-oriented delivery means that the delivery is made after establishing a connection and connection is terminated after the delivery is done.

On the other end, connectionless delivery is done without a formal connection establishment process and obviously a termination.

We will now discuss what service is all about. A connection-oriented service is provided to a sender and a receiver where the receiver gets information in the same order as sent by the sender. The connection-oriented service may or may not be implemented using a connection-oriented delivery system. TCP provides connection-oriented services to applications like SMTP running on top of it, whereas IP only provides connectionless delivery to the other end. How TCP provides the connection-oriented service is quite interesting and we will study the same in Chapter 9.

1.4.4 Differences between TCP/IP and OSI Models

Some important differences between these two models are discussed here.

Number of layers OSI has seven layers, whereas TCP/IP has only five layers.

Default delivery system The OSI layering scheme was based on the decisions made by their committee members, most of whom are telecoms. They preferred connection-oriented transfer for their

communication. It worked like telephones, where a connection was first established, after which the data transfer took place, and finally, the connection was closed. On the other hand, the TCP/IP model was based on connectionless transfer where the data is pumped into the network without establishing any connection to the recipient. This mechanism is more like sending a telegram to somebody. That is why the data chunks travelling in the network are usually referred to as *datagrams*. We just send it without really knowing if the recipient's address is correct or not or whether the recipient is ready to receive our message.

Choice to customer The TCP/IP model used connectionless mechanism for data transfer. At the same time, it provides a choice of connectionless or connection-oriented services to the customers. This is quite an intelligent move, as we have already observed that customers will have a choice in case of TCP/IP. The services that demand connection-oriented (CO) get CO service, whereas those that want connectionless (CL) get CL service.

Protocol and model The TCP/IP model actually describes an existing set of protocols (the practised learning of arranging birthday parties is put on paper). The OSI model was designed before any actual network was designed. Thus, in general, it can be used to describe any other model. The TCP/IP model is tailor-made for TCP/IP protocols. It fits the TCP/IP protocols perfectly. Fortunately, non-TCP/IP protocols are not in much use.

Interface and protocol The OSI model distinguishes between an interface and a protocol. An interface defines the communication that takes place between a lower and an upper layer (e.g., communication between the manager and the secretary or the secretary and the route operator, or the route operator and the warehouse keeper). A protocol defines the communication that takes place between two peer entities (e.g., the secretary of ABC to the secretary of XYZ and the transporter of R1 to the transporter of R2). TCP is the protocol between two transport layers, whereas IP is the protocol between two network layers and ethernet is the protocol between two data link and physical layers.

Protocol structure The OSI model clearly mentions the physical and data link layers. They are required for a complete model description. (We have to describe who will put the consignment into fitting boxes and who will carry them at the other end and how it is done.) The TCP/IP model assumes something of that sort is available and it does not describe them.

Having discussed the differences between the TCP/IP and OSI models, each layer will be studied independently. There are two ways in which one can study these layers; these are described in the next section.

1.4.5 Top-down and Bottom-up Approaches to Study Layers

The description of all the layers can begin from the lowermost layer, the physical layer (the transporter). The description should include what it does and what services it renders to the data link layer (the warehouse keeper-cum-security officer). Then, we describe the functions of data link layer and the services provided to the network layer and so on.

The other way is to start with the application layer and describes its functions. To do that, we need to discuss the reason for specific services required by the application layer. Then, we describe how these services are provided by the transport layer and so on.

One thing is common in both the approaches. In the bottom-up approach, we need to refer to upper layers to explain the functioning of a bottom layer. For example, when we are discussing physical layer, we may need to explain that an email is being sent (through the application layer) that travels down the hierarchy and is eventually given to the physical layer for transmission.

Though the network layers can be studied from top to bottom or vice versa, inherently all of them are designed to solve a single problem and work in cohesion; one must talk about them as a whole to understand either of the case.

Similarly, when we discuss the function of application layer, we have to explain how an email is transmitted down to the physical layer and then transmitted across to the destination and again up to the receiving application layer. Therefore, if we follow the top-down approach, we need to discuss the lower layers to show how a specific service is implemented, which is required at the upper layers. If we consider the bottom-up approach, we need to discuss the services that are to be provided for the functionality that we are describing.

Thus, there is not much difference between the two approaches. Here, therefore, the bottom-up approach is followed.

1.4.6 Functions of Each Layer

We have had several descriptions of layers in this chapter using analogies. What are these layers actually accountable for? The following sections provide a brief summary of the functions of the different layers. We will start with the physical layer and gradually move up to the application layer. In later chapters, we will describe the applications of each layer in detail.

Physical layer The job of the physical layer is that of a transporter: to carry bits from one end to another. It utilizes the available communication medium, that is, a wired or a wireless connection, to transfer the bits to the other end. It is interesting to note that there is more than one mechanism to transfer the bits from one end to another using the same medium (Fig. 1.7). The study of the physical layer describes the different ways of transferring bits from one end to another and their pros and cons.

What should be done when multiple senders and recipients are in the fray? In such cases, one way of sending a message is to broadcast it to the entire network. The intended receiver accepts the message and others reject it. The other way is to find out the exact recipient and send the message using a direct path. The first case does not require much trouble on the senders' part, but the second one requires some knowledge about where the receiver is located in the network. Both these approaches are discussed in Chapter 6.

Thus, the physical layer handles the transfer of bits from the sender to the receiver, converts the bits to voltages or light pulses, and at the other end converts the incoming voltage values or light pulses into bits. The physical layer checks whether the message has to be broadcasted or sent to a single node.

Data link layer The job of data link layer is to send bits using the physical layer. Additionally, it provides quality control measures by ensuring the bits sent and the bits received remain identical.

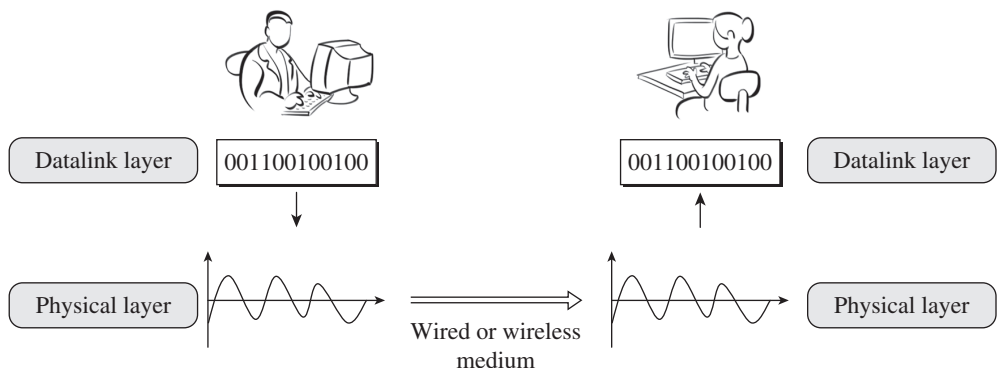





Fig. 1.7 Transfer of bits from sender to receiver through a physical medium

 The function of the physical layer is to carry bits from the sender to the receiver using wired or wireless links.

This is important as there is a possibility of the data getting corrupted during transit. The data link layer provides ways for the sender and the receiver to recognize erroneous or unintended data. To ensure error-free transmission, the data link layer adds additional bits to the data using an algorithm. These additional bits are calculated from the data itself. The calculation procedure is designed in such a way that the additional bits are different for different data. The same algorithm is also applied to the data at the other end, and the results are compared. If the results match, the data is accepted; otherwise, it is rejected. This is known as the *error-detection mechanism*. In some cases, the bits added are designed not only to detect errors but also to correct them. In this case, the algorithm is designed in a manner that the pattern of change in bits due to an error is predictable by observing the additional bits. This is known as the *error-correction process*.

 **Note:** One must understand that these procedures are designed for accidental change in data due to technical reasons and not purposeful change made by humans, usually for malicious purposes. Those measures are discussed in Chapters 14 and 15.

 The function of the data link layer is to receive data from the network layer, add a header and footer for error handling and other operations, and pass data to the physical layer for transmission.

To perform these actions, the data link layer has to encapsulate the data into the body of a unit called *frame*. For identity, it has the senders' and receivers' addresses on the frame. (It is same as the addresses written on the box.) The data link layer also needs to put some mark on the frame to detect errors, if any.

It is possible that the data link layer asks for the confirmation of delivery from the other end. This is achieved by providing acknowledgements from the receiver as soon as the data is received. When the sender receives the acknowledgement, it realizes that the frame has reached the other end.

It is also possible that the sender is sending frames at a faster rate than the receiver can handle. For solving this problem, a mechanism called *flow control* has to be employed. It usually involves a message from the receiver asking the sender to slow down the process.

Thus, the data link layer has to deal with error handling, framing, acknowledgements, and flow control (Fig. 1.8).

Network layer The job of the network layer is to look at the actual destination address and decide the intermediate router through which the packet can be delivered. Figure 1.9 shows the processes at the sender's network layer.

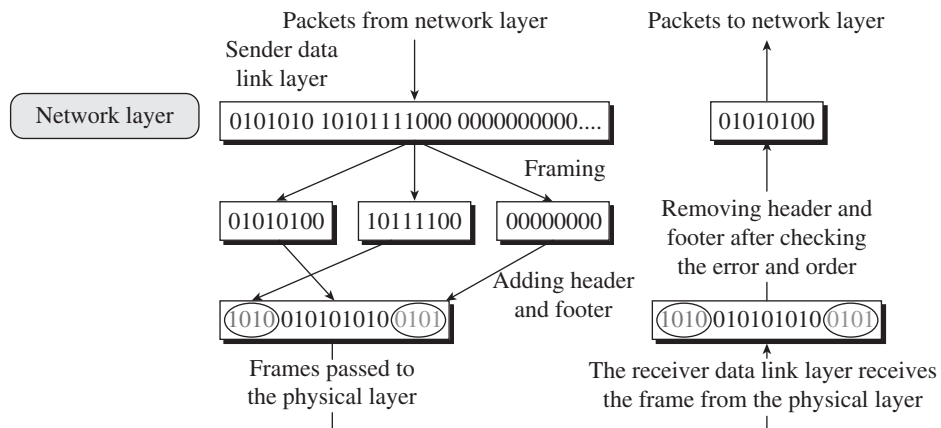


Fig. 1.8 Working of data link layer

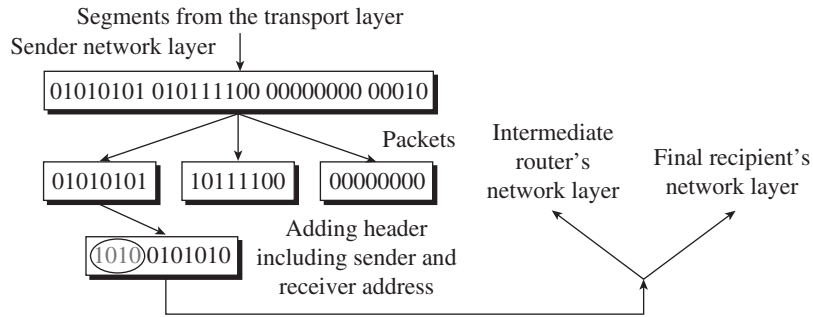


Fig. 1.9 Sender's network layer. It constructs packets from the incoming segments of the transport layer. The network layer adds a header which contains the sender's and receiver's addresses. The receiver's address describes the ultimate destination. It is used by the intermediate routers to find out where to send this packet next.

Autonomy of routers is an important idea behind the decision to keep the network layer connectionless in TCP/IP model.

Major role of the network layer is to manage routing and forwarding.

The process of finding the next immediate router for a final destination is actually two processes in one. First, the network layer must be aware of the locations of different routers and therefore, it can decide a path to the final destination. To know who is in the network, the network layer deploys multiple methods. All these methods are known as *routing algorithms*. The outcome of a routing algorithm is usually a table known as the *routing table*. Once this table is in place, for every incoming packet, the network layer can decide a route. This part is known as *forwarding*. Every network layer must perform both routing and forwarding.

The routing invariably includes a process to handle *congestion*. It is a situation where lines and routers get choked up with more packets than those they can handle. Consider a case of driving through your city. To avoid congested roads, we try to choose a path having less traffic. Similarly, it is important for the layer that decides the route to avoid congested paths that can delay the communication. Driving a car yourself gives you the advantage of autonomy to choose roads that are less congested, whereas travelling by a city bus does not allow this.

The main function of the network layer is to handle routing and forwarding. In addition, it manages the connections for connection-oriented transfer and also determines solutions for problems such as congestion and a variety of traffic issues by using tunnelling. Tunnelling will be discussed in detail in Chapter 7.

Transport layer The transport layer is the secretary of the application layer. In most cases, it provides a sense of responsibility to the actual communication by deploying various techniques. The functions of transport layer include direct communication to the transport layer at the other end. We have already seen that the secretary in our example sends a letter and receives an acknowledgement to confirm to the manager that the job is done.

Let us assume we are using an application like Telnet (technically called the Telnet client). When we press *ls* (this is a command to list all the files on the remote Telnet server running Linux or UNIX), Telnet passes on that command to the TCP process running on our machine. The TCP process running on our machine establishes a connection and sends the message *ls* to the TCP on the other end¹⁷ and receives the acknowledgement back. When the Telnet server sends a list of files to

¹⁷ When the transport layer sends a message to the other end and gets a response, it is always via all the layers below it and all the layers up to the transport layer of the receiving party.



Acknowledgements and retransmission are the key components of TCP for providing reliable communication.

us using the TCP connection, our TCP process sends the acknowledgement back and passes on the content (the list of files) to our Telnet client so that we can see the list on our screen.

For data transmission, usually, a connection-oriented service is preferred, but in case of real-time transmission, live audio or video in particular, it is essential to have a connectionless service. Here, the transport layer is not supposed to provide connection-oriented, reliable service to the application layer and the reason for this will be explained in the following paragraphs.

The transport layer ensures reliability by employing a simple technique of *timing out* and *retransmission*. Let us try to understand the same using an example.

Suppose A is sending a file to B. The file contains five paragraphs. The paragraphs are numbered from 1 to 5. Suppose A starts sending the file at time t . Thus, the TCP process running at that place receives the file at time t from our application. At the same time, the TCP process of our machine sends the first paragraph to the other end. After a time delay of t , the second paragraph is sent and so on. The TCP, after sending each paragraph, starts a timer. The value of the timer indicates the time the acknowledgement of the paragraph is expected. The timer value is decided by the TCP using a well-designed algorithm, and usually, is a good estimate of the round-trip time to the specific destination.

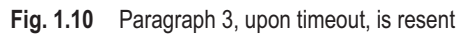
Suppose the round-trip time is calculated as x . Now, the timer value is set to (usually) $t + 2x$ for the first packet and $t + \Delta t + 2x$ for the second¹⁸ packet and so on. The timer value is more than the actual round-trip time (it is considered to be double than the expected round-trip time) to avoid unnecessary retransmission in case of negligible delays. If the acknowledgement does not reach in that time, the timer is said to have expired, and the TCP process retransmits that paragraph (without consulting the sending application, FTP in this case).

There can be various reasons for no receipt of acknowledgement in time. Some of the plausible reasons are as follows:

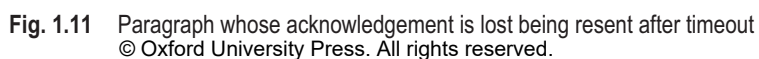
1. It is possible that the paragraph gets lost in transit and the TCP process at the other end failed to receive.
2. The process has received the paragraph and sent an acknowledgement as well, but the acknowledgement is lost in transit.
3. Neither the paragraph nor the acknowledgement of the transmission is lost. It may be the case that either the paragraph took more time to reach the receiver or the acknowledgement took more time to reach us. It can occur due to accidental congestion in the network. Our timer, without really knowing about this, has timed out, as the acknowledgement does not reach in time.


Now assume that the third paragraph is lost. It is retransmitted after the fifth paragraph as the timer expires just when the fifth paragraph is transmitted. Now, we are sending the retransmitted third paragraph and it is received by the receiver. If the receiver presents the paragraphs in the order in which they arrived, then the user will get the fourth paragraph after the second one, and similarly, the third paragraph will be presented after the fifth one. This might create chaos. To avoid this situation, the receiver should collect paragraphs and give it to the user only if it is in order; otherwise, the receiver should wait till the right paragraph comes. The out-of-order paragraph can wait in the storage till its turn comes. The receiver has some storage capacity especially reserved for the same. This is a usual mechanism for file transfer (FTP), remote login (Telnet), or even Internet browsing (HTTP) to provide reliability in data transfer. Data transmission using TCP automatically ensures retransmission of data that has not been delivered, and the situation is depicted in Fig. 1.10.

¹⁸ The constant value 2 is used as a multiplication factor to the round-trip time in earlier versions of TCP. Current routers use a value based on variance calculation, which is explained in Chapter 8.



The mechanism of resending the missing data does not work well in case of real-time audio or video. We have to display each video frame as soon as it reaches, to make the movie look continuous. In a video display, if one or two frames are missed, then the best option is to skip those frames. Similarly, in an audio transfer, if a word or two is lost in transit, just keep listening to the words after that! A human viewer or listener has great ability to pick up what is missing in the presentation. If in a live video, a frame or a sequence of frames is missed for a short period, then the user can always guess the missing content from the context. Therefore, it is ok if the lost data is kept lost. If we provide retransmission and display a frame later, it will create more confusion.




 Retransmission is not good for real-time data.

Let us consider an example to reinforce this issue. Suppose in an animation movie, Mickey Mouse is shown moving from left to right. There is a creature sitting in the middle of the screen. Frame by frame, Mickey is coming closer to that creature. In one of the frames, Mickey kicks that creature, the next frame shows the creature in the sky, and the next sequence shows the creature falling on top of Mickey. Now, assume the frames are sent one by one and the kicking frame is lost. Then, the viewers will automatically assume that the creature must have been kicked. It is absolutely fine till now. If the kicking frame is retransmitted, then we have a sequencing problem. If Mickey is shown kicking that creature after the creature falls down, then it will create confusion in the minds of the viewers as to how Mickey came out from beneath the crashed creature and kicked it. So it is better not to transmit the lost frame and keep it lost. Now a question arises—if we do not need retransmission of lost data, then what is the necessity to have timers and count the number of transmitted frames? If nothing is required, TCP becomes an overhead.

For all such cases, user datagram protocol (UDP), an alternative to TCP as a transport layer protocol on the Internet, is preferred. UDP is a protocol that skips a few checks that TCP normally does and is comparatively lightweight as it does not check for lost segments; it also does not need to retransmit them and so does not require elaborate mechanisms for doing so. Almost all real-time transmissions use UDP as their transport layer protocol.

There are three protocols (Fig. 1.12) at the transport layer (discussed in Chapter 11). One provides a connection-oriented service over a connectionless network layer (TCP over IP) and another provides a connectionless service over a connectionless network layer (UDP over IP). Both protocols are representatives of the two possible transport layer services¹⁹. The application layer is responsible for modelling applications written by the user. So when two different transport layers are provided, an application writer (a programmer) can have two choices for the application. Though not very popular as yet, there is a third protocol called stream control transmission protocol (SCTP) which can also be used in the TCP/IP model. SCTP is a better design for multimedia traffic but due to legacy of systems designed over TCP and UDP, it is really hard for designers to move those applications running on the new protocol.

 TCP and UDP are conventionally used for the TCP/IP transport layer communication, SCTP is also another not-so-preferred but potentially better candidate for multimedia traffic.

In the OSI model, the network layer is provided. It is an interesting case. In the OSI protocol stack, a network layer can be either connection-oriented or connectionless. Thus, the data transfer can be performed either way. It gives a choice to the service provider (ISP). He/She can provide either a connectionless data transfer (like IP) or a connection-oriented transfer (like Wimax or 802.16). Unfortunately, in this case, the choice is not directly given to an application. The transport layer has a choice of selecting a network layer, but an application layer does not have a choice.

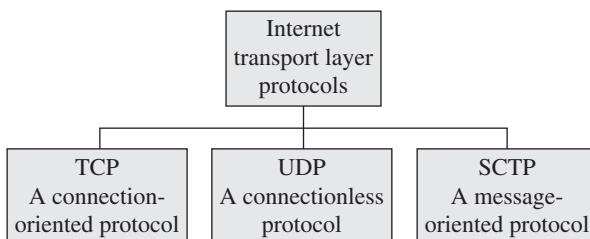


Fig. 1.12 Transport layer protocols

The transport layer performs one more important function. There can be multiple applications running on our machine, all of them taking the services of the same transport layer. In such cases, it is important to track the requirements of each application and provide requisite services accordingly. This process is known as *multiplexing*. It is important to note that this issue arises in other layers as well, but the amount, complexity, and dynamism in numbers is much more in the transport layer as compared to others.

¹⁹ Now there is one more player in the fray, namely, SCTP, which amalgamates the features of TCP and UDP. We will study SCTP later.

The function of the transport layer is to communicate the application layer message to the other end, reliably if the application wishes so, otherwise using connectionless methods.

The application layer helps the user to communicate with the application and pass his/her commands to the other end in a transparent way.

Thus, the transport layer manages the connection with other transport layers, handles all the issues that are in the data link layer, and also takes care of multiplexing.

Application layer The application layer is the topmost layer. In this layer, the user directly interacts with the software application. When we connect the Telnet with a remote server or an FTP to upload or download a file, or with HTTP to browse the Internet, the FTP client or the Telnet client starts interacting with the application layer that is embedded in the application that we run (i.e., the FTP client program). It is also said that the client runs at the application server. The FTP program that we run is actually a client to a larger program known as FTP server. The server also runs at the application layer. Our FTP client communicates with the FTP server at the other end to download and upload files. The interaction proceeds by utilizing the entire protocol stack involving the application layer to the physical layer that we discussed earlier. The FTP client sends the command (i.e., *ls*, *cd*, *put*, *get*) to the application layer. The application layer decides the server that should reply to this request. Then,

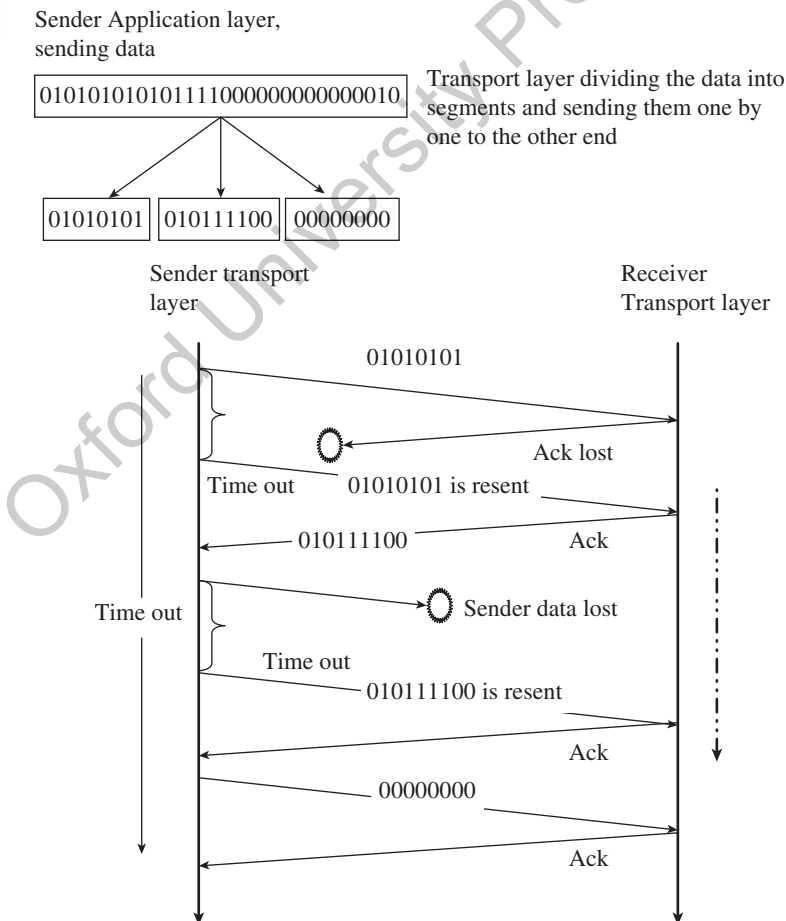


Fig. 1.13 Transport layer sending three different segments over the connection one after another and ensuring retransmission in case a segment or its acknowledgement is lost

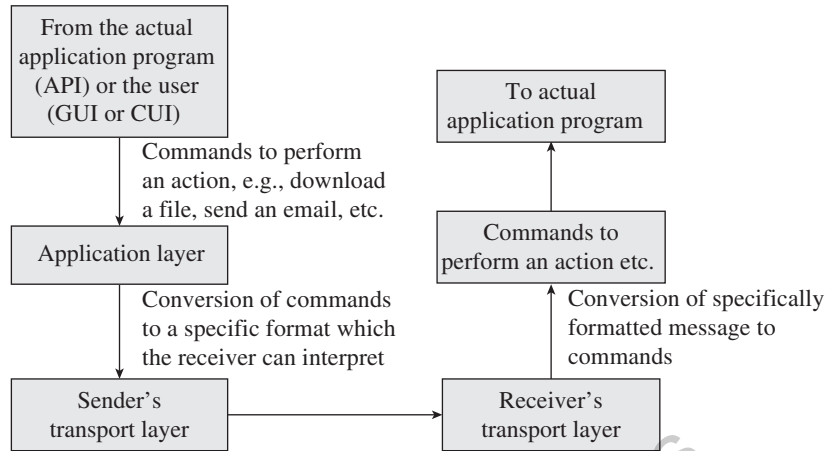


Fig. 1.14 Application layer obeys the command from the application, generates a specific message understandable to the receiver, and passes it on to the transport layer. Transport layer transmits the message to the other end using all the layers below it

the application layer asks either the TCP or the UDP (or some other transport protocol for that matter) to carry the request to that server. In case of FTP clients, we may be looking at an FTP server on a machine using TCP. If our FTP client decides to have a connection with an FTP server, then our TCP process establishes a connection with the TCP process running on that server. After that, the TCP process, on the receiving machine, connects to its FTP server. Thus, the application layer connects an application on one machine with another application on a different machine.

The application layer is important for the following reasons:

1. In this layer, the user directly interacts with the software application. It must be equipped with a good user interface. When we run Telnet, FTP, or HTTP (the browser), we usually have such an interface. Other layers can concentrate more on functioning rather than the interface.
2. This provides services to the user. Users may require services in various formats. Application programming interface (API) is a popular mechanism. It provides the user interface to interact with the application. API is very useful for programmers. If you would like to incorporate FTP or Telnet in your own application or in case you want to have a routine for sending and receiving mails in your own program, then API is the best option. Another popular interface is the graphical user interface (GUI). Nowadays, almost all applications provide this interface. There is another interface called character user interface (CUI). It comes handy in some cases, for example, normal mobile phones with menus. Mobile UI is a complex problem to solve because it deals with relatively small screens.
3. The application layer must provide open solutions to make sure that all these interfaces can be laid on top of it. Such versatility is not required in interfaces for other layers.

The application layer must work differently with different applications. In fact, we need many more application layers than any other layer because we deal with many applications using the same transport, network, and other layer (Fig. 1.14).

1.5 COMMUNICATION PROCESS

Now, let us look at the entire communication process considering all the layers collectively. Assume that we are sending an email to the address `bhushan@glisict.org`. Here, 'bhushan' is a mailbox on the mail server of 'glisict.org'. It is like having a few mailboxes at the entry of an apartment

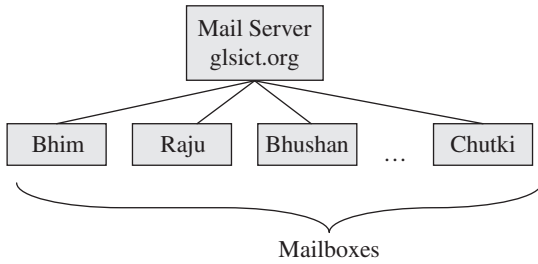


Fig. 1.15 Mail server and mail boxes

for each of its residents. The mailboxes are shown in Fig. 1.15. They are all part of a mail server located at glsict.org. Here, the address glsict.org is similar to an apartment address, which is common for all mailbox holders at the same place. All of us who reside at 'glsict' are given a separate mailbox to store received mails.

Whenever a postman has to deliver a mail to a mailbox *X* belonging to apartment *Y*, first, he finds out the apartment *Y* and then drops the letter in the mailbox *X*. Therefore, the first job is to find out the address of *Y*. Once it is found, a mailbox *X* may be found. The glsict.org, which is part of the email ID bhushan@glsict.org, is not the actual address.

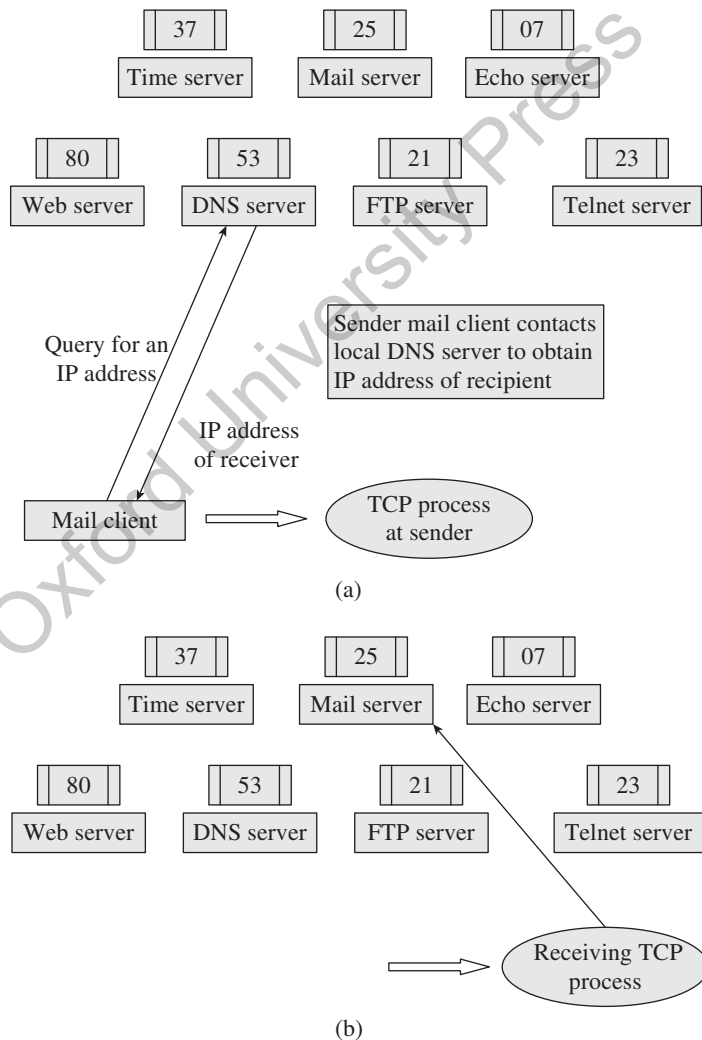


Fig. 1.16 (a) Mail client connects to DNS server to find out address associated with glsict.org (b) Receiving TCP process passes the mail to mail server at port number 25

It gets converted to a 32-bit integer known as the Internet protocol (IP) address which is the actual address. A process known as domain name system (DNS) is invited to find the IP address for the server `glsict.org`. Once that address is obtained, we can proceed further to get mailbox `bhushan` to deliver the mail.

We will send this mail to a mail server on the machine represented by `glsict.org`. It is noted that there may be more than one process running on the server. The mail server is just one of them. When we send something, how would the recipient know that the message is for the mail server and not for anybody else? The problem is solved by providing a separate number for each service. This integer, which indicates a service, is known as the *port number*. The mail server's port number is usually fixed and known globally. All machines should run their mail servers at port no. 25 (Fig. 1.16). When the sender sends a mail, the DNS helps in getting an IP address from the recipient server from `glsict.org`. That address is used by TCP to connect to the other end. When the mail reaches the other end, it connects to the receiver's mail server by specifying the port number 25.

After getting the IP address from the DNS, our mail client requests the TCP to establish a connection with the IP address and the port number specified. Upon receiving this request, TCP generates a connection request segment and passes it to the IP layer below. Before passing it to the IP layer, TCP adds a header indicating a few important parameters, including the sender's and receiver's port number.

The IP layer looks at the destination IP address and decides where to send that packet next. It is possible that our recipient is 10 networks away on a specific path. The IP layer decides the router on the next network. Then, it prepares the header, including the sender's and receiver's IP addresses. Thereafter, the TCP segment is embedded inside this packet. It then passes the packet to the network interface card (NIC) (e.g., the ethernet card or the Centrino card).

This NIC contains the data link layer that generates a frame. The IP address of the next recipient is also converted to the recipient's physical address by means of a process known as *address resolution*. Now, the frame is constructed. The sender's physical address is taken from the card itself (as it is the card's own address), and the recipient's address is taken from the address resolution process.

Now, the frame is sent to the next immediate router (Fig. 1.17). The router's NIC (ethernet card) receives that frame. When the physical layer receives the bits, they are passed up. The data link layer now understands the bit streams received as a frame. It checks the destination address, which is the

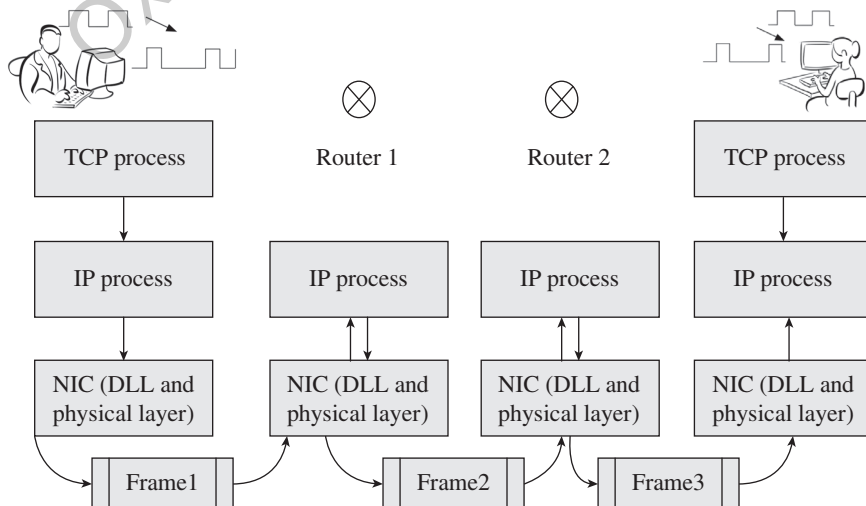


Fig. 1.17 Mail being communicated to other end

Destination	IP address	Subnet Mask	Interface
Changa	1.2.3.4	255.255.0.0	1
Nadiad	2.3.4.5	255.255.0.0	2
....		

Fig. 1.18 Routing table which IP process refers to

card's own address²⁰. If it is correct, then the content of the frame (i.e., the network layer data) is passed to the network layer. Now, the network layer checks the network address of the destination. If that address is not its own, it has to forward that packet. In case of intermediate routers, the destination network's address is not its own address, so they have to route that packet to some other router. Hence, they refer to the table known as the *routing table* (Fig. 1.18). This table suggests the route to reach a given destination. Once the router learns from the table where to send the packet, the router constructs a new frame. The new frame will contain the sender's address as the router's own address and the receiver's address as the next destination address. Then, it is passed to the physical layer to transmit it to the physical layer of the next router. Remember that the next router is decided at the network layer looking at the routing table.

It is important to observe that the router processes this message till the network layer and does not let it go up as the message is not for the router and the transport layer data is of no use for the router. This is similar to a consignment-related message sent by the secretary, which is of no use to the intermediate offices (read routers). The only information needed is the receiver's address, looking at which they can find out where to send the packet next. The router will be able to route successfully with the data available at the network layer itself. That is why when the message reaches the network layer of an intermediate router, it just takes a return journey to the physical layer and does not go up.

Then, the packet is received by the next router. The next router performs a similar processing to find out its next router, constructs the frame, and sends it across. This process will continue till the next destination is the final one.

When the final destination network layer receives the packet, it concludes that the address is its own. Thus, instead of forwarding the packet, it passes the content of the packet up to the transport layer. The transport layer, upon receiving the segment, sends an acknowledgement on the reverse channel in a similar fashion. This is analogous to the document being signed by the recipient secretary in our example. The transport layer passes the data up to the application layer, and the application acts upon the command sent by the sender. If the sender has sent the command *ls* to the receiver, which is a Telnet server, the receiver interprets that the sender wants to have a directory listing. It acts upon that command and prepares the directory listing. It then sends the listing back on the reverse channel. In this case it sends back the mailbox content.

²⁰ It is possible that the card receives a packet not destined for it. There are a few reasons for this. It may be done when there is an error in the communication system. The second possibility is that the received packet is a broadcast or multicast packet. Broadcast means relaying a message to every machine in the network, whereas multicast means sending to a specific group of machines. If our machine is a part of a multicast group, then we may receive that packet. If we receive a packet with the destination address of a group we belong to, then the card will have to accept that packet. Similarly, in case of a broadcast address, the card must accept that packet.

Let us consider one more example from the Internet domain to reinforce our understanding of the communication process and the duties of each layer. This example also illustrates some issues that we have not observed in the previous example.

If we type `http://www.glsict.org/regular.php` in our browser's address bar, you will notice that the uniform resource locator (URL) contains two parts. The `http://www.glsict.org` is the first part and `'regular.php'` is the second part. The first part indicates the protocol (`http`) and the name of the server (`www.glsict.org`). The second part (`regular.php`) indicates the name of the page that is requested from that URL. The browser requests DNS to extract the IP address that the URL pertains to. Then, the browser passes it on to the application layer. The application layer is identified by the HTTP client in this case. The HTTP client constructs a message similar to the following:

```
GET regular.php HTTP / 1.1
host : www.glsict.org
```

The aforementioned message indicates that the page `regular.php` is requested. (We will not be looking at the exact meaning of each of the components of this command.) This message is given to the transport layer (TCP) requesting it to establish a connection with the IP address obtained from the DNS (i.e., the IP address of `www.glsict.org`) with the port number 80 (because a web server typically runs at port number 80. As discussed earlier, every application has a unique port number. If we do not specify a port number, it is assumed to be 80). Assume the IP address obtained is 20.30.40.50.

Now, to establish a connection with that IP address, the TCP must send a connection request to the host. Therefore, it generates a connection request with the receiver's port number as 80 and supplies it to the IP layer.

The IP layer decides the next immediate destination for that IP address 20.30.40.50 (assuming that 20.30.40.50 is not directly accessible). The next immediate router's address is found to be 30.40.50.60. This information is gathered from the routing table as discussed earlier. Now, the IP constructs a packet indicating the final destination (20.30.40.50) and passes it to the data link layer, which in turn sends the packet to 30.40.50.60 (the next router in the chain).

Upon receiving the packet, the data link layer finds out the physical address of 30.40.50.60 (using the address resolution process) and constructs a frame indicating the physical address as a recipient. Then, it passes the frame to the physical layer.

The physical layer, upon receiving the frame, sends it bit by bit to the other end. The physical layer at the other end (of an intermediate router) passes the frame to the data link layer after the entire frame is received.

The intermediate data link layer extracts the IP layer packet and passes it up to the IP layer of the intermediate router. The IP layer, looking at the final recipient's address, decides to pass it on to 40.50.60.70 and repeats the process to pass it to the IP layer of 40.50.60.70 via the data link and physical layer.

Ultimately, the recipient IP layer receives the IP packet, and instead of passing it back to the data link layer, it passes the packet up to the TCP layer. If the TCP layer accepts the connection request, it sends back an acknowledgement using a route similar to the TCP layer. Upon receiving the acknowledgement of the request, the sender again sends the acknowledgement back. Now, the sender sends the HTTP message that we have described earlier to the receiver. It will follow the path to the recipient and utilize the services of all layers. Upon receiving the message, the TCP layer at the other end passes it on to its application layer (the web server or technically an HTTP server). The web server understands that the page containing regular faculties is requested and sends it back using the same set of layers as in the earlier case.

We have discussed two different examples to describe the communication process. However, we have left a few questions unanswered—for example, how does the DNS convert a name to an IP address? We will discuss these concepts in due course.

1.6 DISTRIBUTED SYSTEMS AND NETWORKS

The most important facility provided by a network to its users is the ability to communicate with each other. The members of a network are popularly known as *nodes*. Some of the nodes are special; they are designed to provide service to other nodes. We get information regarding the stock market, cricket scores, or what is happening in our office using the network itself. The nodes gather all necessary information and provide it to users on request. For example, when we request for stock market updates, the required data is collected from a server and presented to our system. It is important to note that the network is a facilitator and a required component for all these services. Thus, the normal nodes and the server nodes are connected by a network that acts as the medium to provide services.

When we are part of a network, we may be connected either to a Windows server or to a UNIX-based server. We move from one server to another by giving a specific command. If we use Telnet to remote login to a machine or FTP to download a file from some other network, then we must know the name or the IP address of the network. We must have a valid username and password on that network server, and we must specify the exact sequence of commands to perform the tasks. Here, the change of server or service is not transparent; the user explicitly provides commands for every such change.

All of us are familiar with the Internet (and addicted to it!); it also contains quite a large number of servers. Most of them provide a service known as *web service*. That is why they are all known as *web servers*. The service is popularly known as world wide web (WWW). When we move from one website to another by clicking a link, it is possible and not unusual to move from one server in a country to another server in another country. When we click a link, we do not realize the exchange of servers. The WWW service, which is deployed by the HTTP protocol in our browser (be it Internet Explorer, Netscape, Mozilla Firefox, or something else), enables us to do so without really knowing the details of such movements. This service, which hides the underlying complexity from the user, is known as distributed system. The WWW is one such service. The distributed service is transparent to the user where he/she does not have to give specific commands to change the server. The user just clicks a link and the distributed system takes care of changing the server for the specified link.

One must clearly understand the difference between distributed systems and networks. A network is a physical infrastructure on top of which the distributed systems are built. Network users are aware of its existence and use that knowledge to work. For example, we must know the name of the server and must have a valid username and password to work with it. In contrast, when we use the WWW, we do not need to have a username or password while switching from server to server. As in the case of other services, the WWW must have a networking infrastructure in place to provide this service.



Distributed system is a transparent access to network-based services like WWW.

1.7 PEER-TO-PEER AND CLIENT-SERVER NETWORKS

The browser is an example of a client. When we access a website using a browser, the browser communicates with the required server using a protocol called hypertext transfer protocol (HTTP). The browser usually requests the content of the URL that we have specified in the address bar of our browser. When we type www.oup.com, we are requesting the browser to send the query to a server running at www.oup.com and ask for a default page from that server. Here, both communicating parties have a specific role to

play. One is requesting a service (the client, browser in our case), and the other party is responding accordingly (the web server of Oxford University Press in our case). The server is always ready to accept (or reject at times!) requests. Whenever a client tries to connect, the server is usually ready to listen to that request. The server itself never connects to any client on its own. It is always the client who initiates the communication. When both parties are bound to such roles, the communication is known as client–server communication. It is to be noted that both the client and the server are nodes; they may belong to the same network or to different networks connected by a medium (Fig. 1.19).

When such roles are not defined, all the parties involved can initiate a connection and then send and receive whatever they choose to. In this case, communication has no basic structure, and anybody can initiate a connection with anybody else (Fig. 1.20). Thus, there is no clear-cut role of client and server—a node can act as a client at one point and a server at another, or both at the same point of time. At the same point of time, it initiates a connection to one party while receiving a connection request from another party. This kind of communication between two equal parties is known as *peer-to-peer (P2P) communication*. It is possible that the same network infrastructure may provide both types of communication. The Internet provides a client–server communication when we use either a browser to access information from the web server or an FTP to download a file from a remote location. The same Internet provides P2P communication when we use either BitTorrent or eDonkey. In both the cases, anybody can send and anybody can receive information; it thus acts as either the sever or the client. In fact, it is also possible for a machine to both download a file from one machine and upload files to some other machine at the same point of time.

P2P networks came into existence from the need for a normal user to share his/her contents, for example, videos, photos, and so on. In fact, P2P systems have many legal and useful applications. Users find it easy to store and distribute contents to their friends using P2P networks so much so that P2P traffic has surpassed web traffic. BitTorrent (Torrent) is the largest P2P network today. It has the largest share of all traffic over the Internet.



Note: From Version 6.0, the software no longer has an open source and is rebranded as μ Torrent.

The basic principle of P2P networks is that peers (the users who use such network) share their contents over the Internet and also use content provided by other such users. The desktop machines (which are ordinary PCs) form a huge content distribution network that can surpass the biggest client–server network’s traffic. The most important part is that there is no central point of control in the entire system.

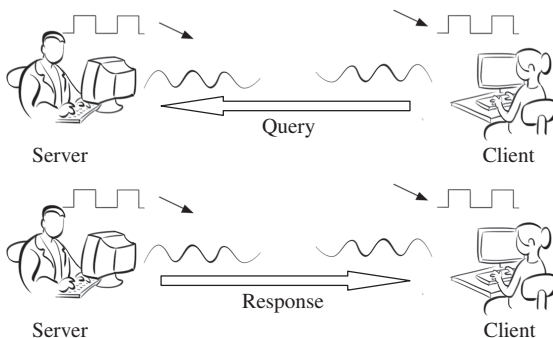


Fig. 1.19 Client–Server communication

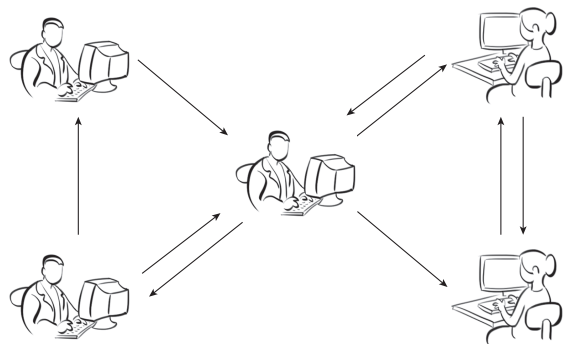


Fig. 1.20 No structure in peer-to-peer networks

Peer-to-peer communication is between two entities which change their role as client and server over a period of time.

These characteristics of P2P networks make them very good at scaling. It means that increasing the number of users does not increase the demands for bandwidth or processing from servers. It is not easy for the system to have optimum performance when users have different upload and download speeds. A serious problem with P2P network is to have intermittent connections from all users. Unlike dedicated web servers, which remain online all the time and are also ready to have many concurrent connections, normal users may join and may leave the network any time and cannot handle more than a few connections.

In P2P networks, the information share has more user control, and thus, user privacy issues are better handled. Many incidents have proven that users, who are oblivious of simple privacy-related disciplinary measures, can compromise any system and are vulnerable to attacks.

1.8 CONNECTION-ORIENTED NETWORKS—X.25 AND FRAME RELAY

As discussed earlier, the OSI designers request connection-oriented networks, whereas TCP/IP designers request connectionless networks. Although there is no true OSI network in the market, there are two protocols that are used to run at the network layer (and thus provided by ISP as a service) and used as a connection-oriented approach. In fact, it is discussed only from a historical perspective as neither of these protocols enjoy more success nor are they in serious usage. Hence, a summary of both these protocols is discussed in the following section.

X.25 (Fig. 1.21) was the first kind where telephone companies were usually government-owned in some countries and are in dominance in other countries (e.g., AT&T was considered to be the telephone company in United States). They wanted a solution that would reinforce their dominance in data communication as well. The ultimate goal was to come out with a connection-oriented model that worked well with the telephone industry. (It means that the user would be paying for the time and not the volume of data.) You can compare that with the ARPANET design, which influenced the Internet design. ARPANET, being largely funded by Department of Defence (DoD) had the notion of having the network on despite failures of nodes (as the government wanted to build the network that could continue running despite some of the nodes being blown away by the enemy). Connectionless design, which helped autonomy of routers, was well suited for that goal. Even when a router or two is blown away, the message would reach the recipient, which was not

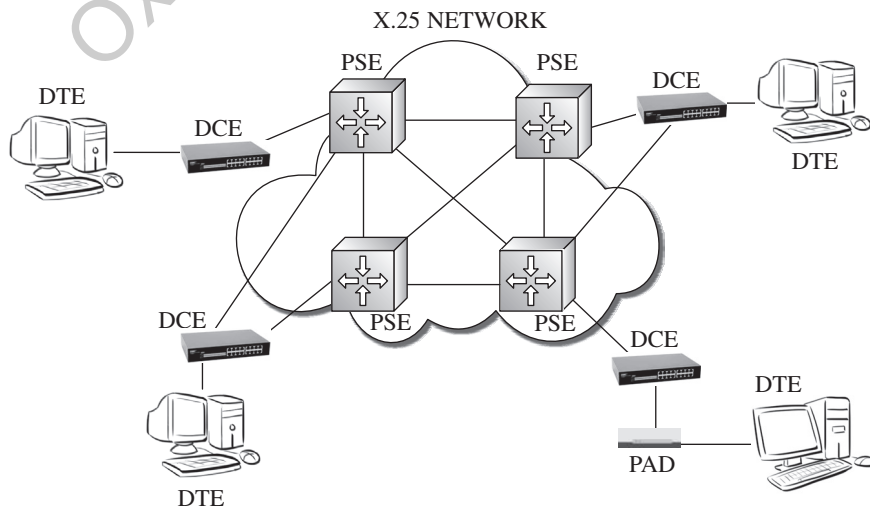


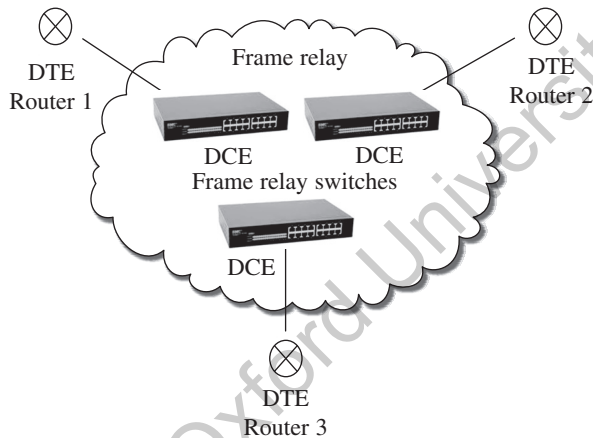
Fig. 1.21 X.25 Network

the case with connection-oriented networks. Obviously, ARPANET designers had no idea about how the mechanism would help billing a customer. In fact, the connectionless mechanism is the worst for those telephone companies that want to bill their customers. They had the model that worked for them. They wanted to charge customers from the connection establishment to termination, counting minutes. They had an idea about how a customer can be charged without a connection.

There are a few advantages of a connection-oriented approach as well. The most important advantage is the ability to provide quality of service. The quality of landline telephone call is far better than the Internet-based call, albeit more expensive. It is because a complete path from sender to receiver is reserved during the complete call, and thus there is no chance of disturbance. There is no likelihood of a new entrant inflicting bandwidth and quality of transmission problems on users who are already connected. For that reason, some of the air traffic control systems used X.25 for quite some time.

X.25 was first introduced in the 1970s by ITU-T (before OSI model was proposed), whereas Frame Relay was introduced in the 1980s. The working of X.25 was similar to a telephone. The sender established a connection to a remote computer by dialling a number associated with that computer. Each connection was given a unique number so that a sender could have multiple connections. The X.25 was designed for a public network (which means it could be established and provided to users as a service like the Internet of today).

Similar to TCP/IP, X.25 is not a single protocol but a family of protocols for the lowermost three layers. In X.25, an end-user's device is known as data-terminating equipment (DTE), whereas data control equipment (DCE) is the entry point to the network itself. Packet switching exchanges (PSEs) act like routers of the Internet, relaying traffic to the intended recipient (Fig.1.22). Sometimes a device called packet assembler dissembler (PAD) was used if the user's equipment was not smart enough to talk to the network itself. X.25 enjoyed a moderate success from the 1970s to the 1990s.



Frame relay was a connection-oriented network with no error and flow control. It was designed to provide better speeds than X.25. Some of the critics compared that to removing the breaks from a car to improve speed. Being a connection-oriented network, it could ensure the order of delivery but not the correctness as there was no error control. It may even swamp the receiver if it is not capable of managing the data flow as there is no flow control either. It is still being used where DSL and cable network cannot reach the subscriber. It contains a slightly simpler architecture than X.25 as it only contains the network and not the end nodes. It was a wide area network (WAN) service. The frame relay switches are devices like Internet routers. Interestingly, frame relay devices used DTE as routers where other networks can connect. Unlike other network layer solutions, frame relay could provide a permanent virtual circuit (where a customer does not need to establish a connection, the switches have permanent entries for the connection) and also a notion of different types of quality of service. Similar to X.25, frame relay also enjoyed fair success for a decade and is still in use at some places.

1.9 NETWORK AND COMMUNICATION STANDARDIZATION

It is important to have standardization for the following reasons:

1. There are multiple vendors of any networking device and all of them have an idea about how things have to be carried out. If all of them prepare devices using their own idea, it will result in chaos.
2. When a manufacturer produces a device, for example, a wireless card, he/she must be sure about a machine, which installs this card, to talk to an access point manufactured by another company. This is observed in all other cases, for example, an electric socket is designed so that a plug manufactured by any other company can fit into it. Not only mechanical standards (so that they perfectly fit) but also electrical standards are met (so that current passing from the socket is properly received by the plug) and so are the other standards (e.g., one of the wires in a household plug will be a neutral wire, one of them is a ground wire, and the last one carries the current). Similarly, a wireless card must fit exactly into the slot provided on the laptop. It should also have the number and sequence of pins expected by the laptop. Thus, people who manufacture devices can freely design their devices without worrying about interoperability issues.
3. The other reason is to have a proper evolution of the technology. For example, IPv4 is an Internet standard which every computer that is connected to LAN or the Internet follows. IPv6 is the next version of the same standard. The standardization process adopted by Internet Engineering Task Force (IETF) helps IP evolve by providing this standard. This IPv4 process can be seamlessly replaced by the IPv6 process.
4. The third reason is to have others who design products to have clear guidelines of what to produce. For example, if we are about to write to a mail client, reading requests for comments (RFC) about SMTP will clearly tell us what to do to write a client that can communicate with any mail server belonging to the Internet.
5. Good standards not only allow people to build products according to standard but also increase the market for products that adhere to those standards. Such a tendency increases demand for such products, which results in mass production, economy of scale in manufacturing presses, better implementations and so on, which in turn results in a reduction of cost and more demand for such devices, and thus, standardization improves the market as well.

A standard is a statement that helps us learn what we need to interoperate. Thus, once a standard like 802.11 (a standard for wireless LANs) is adopted, all companies that manufacture wireless cards and access points start competing with each other to build products according to the 802.11 standard. Unfortunately, the standard provides a few choices for the manufacturer and does not describe when to use which choice and so on. Thus, vendors of wireless products based on 802.11 came out with their own alliance to build standards for interoperability of 802.11 devices, which is popularly known as Wi-fi alliance. The Wi-fi standard is within the IEEE 802.11 standard but is provided to help manufacturers build a product that is interoperable with other manufacturers.



Standardization helps interoperability between devices and vendors.

The standard does not define a solution for everything. For example, a TCP standard that defines a transport layer protocol (TCP) for Internet does explain how the protocol should work. (TCP will be studied in depth in Chapter 9). It does not, though, discuss how the protocol should be implemented in a given machine. Such flexibility helps manufacturers to compete using better coding and implementation methods than others.

Standards in general fall into two different categories, namely *de facto* (informal) and *de jure* (formal). The *de facto* standard happens without a standardization process. All machines that

 De jure standards are designed by standardization bodies; de facto standards are not designed by standardization bodies but are in much use and thus automatically become a standard.

use the Internet invariably use HTTP, but it was not a formal standard for most of its life. One could have chosen some other way to access web servers in earlier days but almost everybody followed Tim Bernes Lee's design. Such standards are known as the de facto standard. Bluetooth, however, now a formal standard, remained the de facto standard for connecting wireless devices for a long period. It is quite common to find de facto standards turn into de jure standards after a while.

On the contrary, the de jure standards are designed carefully by standardization-bodies that are designated for it.



Note: There are many standardization bodies that work in different areas and sometimes overlap with each other (so there is no standardization in the process of standardization!). For example, IEEE 802.11, which is basically a wireless LAN standard, and Bluetooth, which is a wireless standard for low-range devices, use frequencies that overlap and can create a serious problem if both of them are running in the vicinity.

1.9.1 Standardization Bodies

The standardization bodies pertaining to networking and data communication are divided into three types—international standardization bodies, which help the world standardize in a way that the products work across the globe; Internet standardization bodies, that design standards for Internet such as HTTP and SMTP; and telecommunication standardization bodies that standardize the telephone and telegraph.

International Standards

International Standards Organization (ISO) is responsible for developing and publishing international standards for a wide range of items, from clothes and surgical instruments to networking devices.

ISO is quite large, hierarchical, and bureaucratic in nature. The real work is done by working groups that are built when a need for some standardization process arises. The members of ISO are companies that want their products to be standardized by government officials who want to get things done in a neutral way, academicians who want things to be done in an ideal way, and a few other interested parties.

The process of standardization in ISO happens in three steps, which are listed here:

Committee draft This is the first document produced by the working group in the process of standardizing a product. All members get a copy of the same and have six months time to look at the contents and criticize the same.

Draft international standard A majority of the members have to approve the committee draft for it to progress ahead. If the draft international standard is not approved within six months, it becomes invalid. It may be recirculated after six months to all members.

International standard If the draft standard is approved by a majority of members, it escalates to the International standard after all refinements, which are suggested in the earlier phase, are incorporated.

The National Institute of Standards and Technology (NIST) is part of the Department of Commerce in the United States. All US government purchases are made according to NIST standards. Chapter 11 discusses some standards defined by the NIST. NIST is a small but important player in the field of standardization.

Institute of Electrical and Electronics Engineers It is the largest professional organization of the world. Apart from producing many journals and organizing conferences worldwide, Institute of Electrical and Electronics Engineers (IEEE) is also involved in the standardization of electrical engineering and related fields in computer engineering. IEEE 802 committee has standardized many LANs, including the omnipresent ethernet and different wireless 802.11 types.

Internet Standards

The Internet is evolving very differently compared to other systems. When millions of computers are connected to the Internet and many more are joining every day, it is imperative to have a body that governs the same.

The ARPANET, governed by a body deputed by Department of Defence in the United States, was the predecessor to the Internet. In the early 1980s, the body was given a new name—the Internet Activities Board (IAB), which later changed to Internet Architecture Board. The IAB was to look after the research process. There are many stakeholders in the process that requires monitoring. Though nobody owns it, the fund comes from National Science Foundation (NSF) and Department of Defence (DoD), and thus they do have some control over the process. IAB listens to their advice and plans accordingly. In the late 1980s, it was realized that IAB cannot continue working in the informal manner it used to. It was because many other stakeholders demanded that standardization processes be done in a more standardized manner. For example, if a few researchers find a better routing algorithm that demands a new type of router, router manufacturers like CISCO would probably dislike it if new standards demand them to throw away all existing routers and have a new breed of routers following that new standard. They required a ‘vendor friendly’ and not ‘research friendly’ approach to the evolution of the Internet.

So IAB had two new subsidiaries called Internet Research Task Force (IRTF) and IETF. Researchers were no longer the only members of the IAB. A broader Internet Society (ISOC) was created for people interested in the Internet. The Internet Society was made responsible for appointing members of the IAB.



Note: The Internet Society home page says ‘ISOC is a non-profit organization founded in 1992 to provide leadership in Internet-related standards, education, and policy. It is dedicated to ensuring the open development, evolution, and use of the Internet for the benefit of people throughout the world’. Refer to www.internetsociety.org



For the Internet, IETF solves short-term issues, whereas IRTF solves long-term issues.

Both IRTF and IETF have clear-cut separation of work. IRTF was to solve long-term issues (e.g., the deployment of IPv6), whereas IETF was to solve immediate problems to keep the Internet running (coming out with standards for upcoming technologies like Bluetooth). Working groups are basic building blocks of the IETF. There are around 100 working groups (<http://www.ietf.org/wg/> for the list of workgroups), which are now grouped into areas (<http://datatracker.ietf.org/wg/list> the areas). In total, eight areas are listed. They are listed in Table 1.2.

Telecommunication Standards

Data communication, surprisingly, is carried over telephonic lines more often than the voice itself, which it was designed to carry. Most of us use our mobile phones to access websites, especially Facebook, Twitter, and mailing sites. Landline phones, which used to only have voice carried over, are now having asymmetric digital subscriber line (ADSL) and other facilities that help people use Internet with landline connection. The telecommunication standards thus have a telling impression on data communication. Let us now discuss this.

Table 1.2 IETF areas for research

Area name	Work
Application	A new application management, for example, application layer traffic optimization
General	User information
Internet	Internet related, for example, IPv6 over WPAN (wireless personal area network)
Operations and management	Related to operating the Internet and managing it, for example, DNS operations (operations example) and energy management (management example)
Real-time applications and infrastructure	Real-time (audio and video usually) application standards, for example, Call Control UII Service for SIP
Routing	Routing related, for example, open shortest path first interior gateway protocol (IGP)
Security	Related to security, for example, DNS-based authentication of named entities
Transport	Traffic related (e.g., datagram congestion control protocol)

Telephones are managed by the government in most European countries and India. Even when decentralization is taking place and many other companies are invited to participate in the process, there is a lot of government control (e.g., Telephone Regulatory Authority of India, TRAI).

However, let us move on to learn about the different standardization efforts in this direction. A major player in the field is International Telecommunication Union (ITU, which was formerly known as CCITT, a French acronym).

ITU members (obviously) include almost all government bodies that are part of the UN including India. Apart from government players, some private companies are also members of ITU, for example, Bharti Telecom (the company that runs Airtel brand of mobile phone service) from India. They are known as sector or associate members depending on their interest.



Note: All Indian members mentioned earlier are sector members. There are two associate members from India as well. Both of them are private companies from Bangalore and New Delhi.



The international standardization body for telecommunication standard is ITU-T

There are around 200 government and 700 other members of the ITU, which makes it a mammoth organization. It is divided into three different units, ITU-T, ITU-D, and ITU-R. ITU-T is related to telecommunication standardization. ITU-D is related to development sector. It tries to run awareness programmes about information and communication technology (ICT) and promotes this. ITU-R is related to radio frequency allocation and usage.

Let us have some briefing about ITU-T. Being an international standardizing body, its recommendations are to be adopted by governments worldwide, but sometimes they are not mainly due to political and not technical reasons (that is why they are called recommendations and hence there is no compulsion). Mostly, one would not like to do so as using a standard other than the one the world is using effectively would mean cutting their telephone and telegraph users from the rest of the world.

ITU-T is divided into 10 different study groups, numbered 2 to 17 with some numbers missing from the list. The study groups are a really large entity and are further divided into other groups at multiple levels, the lowest being ad hoc groups.

ITU-T has produced many standards that are in use today, for example, the ITU-T X.805, which will give telecom network operators and enterprises the ability to provide an end-to-end architecture description from a security perspective. The popular X.509 standard for certificate is also an example.

1.10 THE INTERNET

The Internet has no owner; it is basically a network of networks, that is, a collection of many networks worldwide. The importance of the Internet is to bring all networks to all users and provide services like WWW, which enables a user to seamlessly access information from any part of the Internet.

1.10.1 History of the Internet

It all began in the late 1950s when Russia (then USSR) was a big threat to US military installations. The military installations were connected like the telephone exchanges of today. Multiple installations connect to a single level-1, and multiple level-1 switches connect to a single level-2 switch, and so on. Thus, all telephone networks used to be circuit switched in those days. Such a design is vulnerable to disintegration into small units not capable of communicating to each other if some level-2 or level-1 switches were attacked. DoD invited many parties to find solutions to the problem.

In the 1960s, Paul Baran, from RAND corporation, proposed a revolutionary idea. The idea was to have a packet-switched network instead of circuit switching and provide multiple connections between switches. This architecture was quite robust in the sense that even when few switches were blown away, there were ample alternate paths that could let the other switches communicate.

Wesley Clark, a networking expert, thought of having packet-switched connections and a concept of the router. Fortunately, he learnt about the National Physical Laboratory (NPL), England, an initiative lead by Donald Devis, which describes a similar system with full-fledged implementation for the same. The NPL system demonstrated that Baran's idea can actually work! The routers were initially named interface message processor (IMP). Their job was to connect to each other using a 56 Kb per sec line and exchange messages in the form of packets of around 1000 bits. The IMPs use store and forward technique, which means the packet must be received completely before being forwarded to the next IMP or a host. Subnets were added to the networks and application software was also introduced.



Note: Applications designed in the 1970s are quite strange if you consider current technology. For example, SMTP uses 7-bit ASCII, which demands methods to convert the data into 7-bit ASCII form before transmitting and vice versa at the other end.

In the 1980s, an important application software was introduced—domain name system (DNS). DNS was helpful in resolving host names to their addresses and simplifying Internet access. By the time DNS was introduced, National Science Foundation (NSF) also started playing its role in the development of the Internet. NSF did two important things. It funded the Computer Science Network, and connected it to ARPANET. NSF also built a backbone where other university networks could join and connect to ARPANET. This backbone was the first TCP/IP speaking network. The entire set-up (the backbone, university, and other networks connected to it) was called NSFNET.

Immediately after launching NSFNET, it started growing and soon an important change came about. Private companies were allowed to join the network. The Advanced Network and Services (ANS) was formed, which was a non-profit organization from some industry representatives including IBM and a few others. NSFNET was upgraded from 1.5 Mb lines to 45 Mb lines and it became

ANSNET. Now, the entire design started moving towards more and more private participation. NSF introduced Network Access Points (NAPs), which help any network to connect to ANSNET. In the 1990s, many other National Research Centres were introduced in other parts of the world. Then came the innovation that changed the face of the Internet, the WWW, a hyperlinked structure containing millions of documents for all users like us to see and work with. Later, the introduction of P2P networks and social networks like Facebook has transformed the experience from only reading to reading and expressing oneself.

1.10.2 Internet Architecture

There are two different parts to Internet architecture—one that lets end users like us to be connected to the Internet and another that creates the Net that is built by the Internet Service Providers (ISPs) who serve to help us access the Internet and the services based on it.

The four popular ways of connecting to the Internet are as follows:

Dial-up connection In this form, the user's computer is connected to the Internet using a device called modem (modulator demodulator). Dial-up connections are of two types, namely wired and wireless. Wired dial-up connections are on the decline today with better DSL-based connections in vogue.

Wireless connections Wireless dial-up connections are possible in two ways. First, a mobile phone is used (the mobile phone acts as a modem here), which connects to the mobile ISP. In the other method, a dongle is used, which is basically a phone-based modem in a miniature form. After 3G, this has become a preferred mode for many of us.

DSL-based connection Digital subscriber line (DSL) is preferred by most landline users. In India, BSNL and MTNL are government players in the field. There are other private service providers as well who provide Internet services. The most popular form of DSL is usually ADSL or asymmetric digital subscriber line. BSNL provides ADSL connection to its subscribers. ADSL requires a device that is popularly known as ADSL modem, but technically, is quite different from a modem. ADSL will be discussed later in Chapter 5.

Cable-based connection Cable networks are almost omnipresent. Cable operators are also replacing conventional coaxial cables with fibre-optic cables, and additional bandwidth provided by the fibre-optic cable is utilized for carrying Internet traffic. Cable operators tie up with ISPs to provide Internet connection.

A cable model connects the user's computer and the cable company at the other end. The cable company makes sure that the connection is forwarded to an ISP.

The latest development in the field is to have fibre sockets in televisions and fibre from the cable operator running directly in the TV set. That will provide an enormous amount of bandwidth to the end-user. This is popularly known as fibre to the home (FTTH).

The other part of Internet architecture is how ISPs are connected to each other and how they make sure their clients are able to access the entire Internet. Once the clients' packets hit the ISP network, ISP manages to make the packets reach their exact destination; when someone sends packets to clients of ISP, ISP manages to make the packets reach the client when the packets reach the ISP network. Therefore, the issue

remains as to how ISPs connect to each other.



The Internet architecture is roughly divided into two parts, namely inter-ISP connections and customer to ISP connections.



Note: Interestingly, a technology called IPTV provides TV channels on IP networks. BSNL provides IPTV service, which you can use to watch television channels on the same connection.

The traffic flowing through the connection is known as transit traffic or transit for short.

The traffic flowing from one ISP to another depends on how they peer. For example, if they are directly peering each other, that is, they are directly connected to each other without any intermediary, a packet might travel directly from one peer to another. Otherwise, the packet travels across the Internet depending on business relations between ISPs and customers' policy settings (more information about the same is provided in the discussion on BGP in Chapter 7). This path is not required to be the shortest and is usually not.

The ISPs are connected in loose hierarchical form. At the top, large ISPs that have thousands of routers connected to all other top-level and few lower-level ISPs with connection to Internet backbones are present. Each one who is connected to the Internet ultimately has a connection to these big ISPs as otherwise it would not be possible to connect to everybody else. They do not pay for transit as they provide services to other ISPs. They are called tier-1 ISPs. Figure 1.23 shows how ISPs are organized.

Tier-2 ISPs connect to tier-1 ISPs and have a direct connection to each other at times. They are also connected to tier-3 ISPs that are directly connected to customers. The customers are also sometimes connected directly to tier-2 ISPs. The customer to ISP is sometimes called point of presence. Some of the large Internet servers are located directly within the ISP network to ensure faster and easier access, for example, companies such as Google and Microsoft. The IXP in the figure is Internet eXchange Point, that is, a place that contains multiple routers to help exchange traffic in the best possible way.

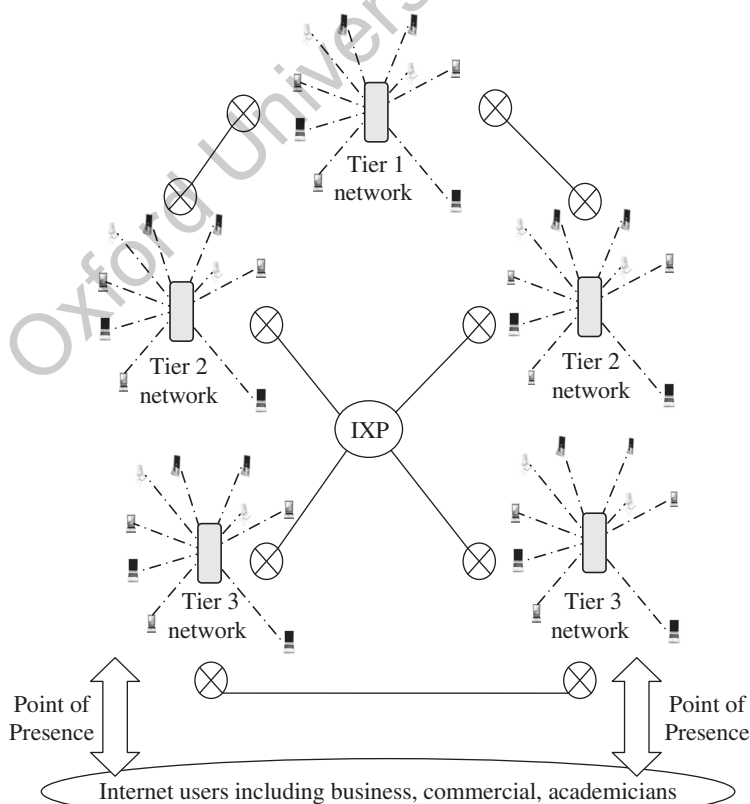


Fig. 1.23 ISP network with different tiers

POINTS TO REMEMBER

- Data communication helps data to traverse from sender to receiver.
- Data communication can occur in simplex, half-duplex, or full-duplex mode.
- A layer represents a service module of a networking application. One layer provides service to another.
- Layers provide the following advantages: (a) division of work, (b) standard interface and easy replacement for a component, and (c) independent design of protocols for each layer. On the flip side, it reduces the speed and performance of the system. It increases the memory usage as well.
- The function of physical layer is to pass on bits from the sender to the receiver.
- The function of data link layer is to send data using the physical layer such that the data is accurately received at the other end.
- The network layer is to decide where to send the packet in case there are multiple outgoing options.
- The transport layer ensures the application layer messages reach the other end and is received by the receiver's transport layer.
- The application layer helps the user take the services of the network. It provides the user with an interface to facilitate interaction with the underlying application.
- Distributed systems differ from ordinary networks in the sense that they use networks and are transparent to the user. It acts as a layer on top of the network to hide the complexity of the underlying network from the user. WWW is an example that runs on top of the Internet.
- In a client-server system, the system that seeks services is known as the client and the system that provides those services is known as the server.
- In P2P communication, the roles of client and server are not defined. BitTorrent and eDonkey are examples.
- There are three different types of standards, namely international standards, Internet standards, and telecommunication standards useful for data communication.
- ISPs are organized in hierarchy. Users are usually connected to tier-3 ISPs. Tier-1 ISPs are the topmost level ISPs.

KEYWORDS

ATM It stands for asynchronous transfer mode. It is a networking model that supports connection-oriented transfer. It is primarily designed to provide different qualities of service.

Client-Server system It is a system where the client and the server are two nodes of the same network or two different networks and are connected to each other by some means. Here, client is the machine that asks for a service and server is the machine that provides that service. The connection is always initiated by the client and never by the server.

Distributed system It is a system that uses the networking infrastructure in place to provide transparent services to users. An example of distributed system is WWW.

Full-duplex mode It is a communication mode where both parties can act as sender and receiver together at the same point of time.

Half-duplex mode It is a communication mode where a sender can send, turn around, and become a receiver.

HTTP It is an application layer protocol used to transfer web pages using a web client (the browser) and a web server.

Interface It is the process of communication between the layers that are next to each other in the same communication stack.

IP It is the network layer protocol of the Internet, that is, it handles the communication between a network layer of one machine and a network layer of another machine on the Internet.

ITU It is an international standardization body.

IXP It is a centre for different ISPs to connect.

Layer A layer is a specific module that is designed to provide specific networking services. It is designed in a way to remain independent of other layers to a large extent.

OSI It is a standard for networking developed by the International Standards Organization (ISO).

Peer-to-peer system Here, two connected nodes exchange information without having specific roles like client or server. Anybody who is in need of some information can initiate a connection to another system that has some particular information in the network.

Protocol It is the process of communication between the peer layers of different communication stacks.

Resolution The number of pixels used to represent an image is called resolution of the image. It is usually represented by matrix values, that is, rows \times columns.

Retransmission It is the capability of TCP to keep track of all the messages sent and retransmit those for which no acknowledgement was received in time. Using retransmission policy, TCP ensures reliable service.

Simplex mode It is a communication mode where sender only sends and receiver only receives.

SMTP It stands for simple mail transfer protocol. It is a protocol used to transfer mails between an SMTP client and an SMTP server.

TCP/IP model It is the networking standard that evolved from the development of a suite of protocols for the Internet.

TCP It is the transport layer protocol of the Internet, that is, it is responsible for the communication between a transport layer of one machine and a transport layer of another machine on the Internet.

Tunnelling It is the process of embedding one type of data into another. For example, ATM cells can be embedded into IP packets.

MULTIPLE-CHOICE QUESTIONS

1. Incorrect data receipt occurs when
 - (a) the data is sent incorrectly
 - (b) the data is received incorrectly
 - (c) the data is corrupted or compromised in transit
 - (d) all of these
2. Image representation size can be decided based on
 - (a) resolution
 - (b) number of bits for one pixel
 - (c) whether the pixel is black and white or colour
 - (d) all of these
3. Layered design
 - (a) is better for small problems
 - (b) helps faster communication
 - (c) requires synchronization
 - (d) all of these
4. Connectionless transfer
 - (a) requires a logical connection
 - (b) requires a physical connection
 - (c) transfers data without any connection
 - (d) is same as connection-oriented transfer
5. The physical layer
 - (a) decides how bits will be represented
 - (b) decides how bits will be transferred
 - (c) uses voltages or light pulses to transfer
 - (d) all of these
6. The data link layer
 - (a) transfers data across
 - (b) links data with each other
 - (c) detects and corrects erroneous bits
 - (d) collects data from all links
7. Network layer
 - (a) forwards packets
 - (b) manages networks
 - (c) networks computers with each other
 - (d) lists out all networks
8. Transport layer
 - (a) connects computers next to each other
 - (b) connects the sender to the receiver
 - (c) transports data across
 - (d) routes data
9. Application layer
 - (a) is controlled by applications
 - (b) should be there for all applications
 - (c) requires a user-friendly interface
 - (d) all of these
10. Peer-to-peer networks connect
 - (a) any two computers
 - (b) a client with a server
 - (c) a node with a server
 - (d) any computer with a network
11. RFID is
 - (a) remotely operated
 - (b) a type of network
 - (c) able to help any item to be connected to the network
 - (d) an ID given to each node of the network
12. LAN standards are provided by
 - (a) IETF
 - (c) W3C
 - (b) ITU-T
 - (d) IEEE
13. BSNL is
 - (a) an example of national ISP
 - (b) an example of International ISP
 - (c) the name of a router
 - (d) all of these

REVIEW QUESTIONS

1. What is the need to share resources? Explain with a few examples.
2. The layering concept is central to networking solutions. Explain in brief.

3. We used the sending of goods from one place to another as an example to show the use of layering. Furnish an example from your own domain where multiple layers are used and one layer provides services to another. The answer should not necessarily model the network or the Internet, but layers must be organized in a strict order such that the service provider is placed below the service user. (Hint: one such example is courier service. We pass our parcel to the courier service, which is delivered at the destination. You can add other layers logically.)
4. For your own example of the network, show how the layers simplify the job. Compare the advantages provided in the book with your own.
5. Take the example of browsing the Internet. How does the division of work help the user here?
6. It is mentioned in the chapter that the UDP, TCP, and SCTP have standard interfaces to IP. Do some extra reading to find out what this standard interface is.
7. Protocol independence is an important issue. Why? Find out where this important issue is ignored in the Internet.
8. Layering is not strictly observed in embedded systems and sensor networks. Why?
9. The TCP/IP model is more robust than the OSI model. Is this a correct statement? Justify your answer.
10. There are no session and presentation layers in the TCP/IP model. Why?
11. Draw the seven layers of OSI model and explain the function of each layer in brief.
12. Group the following functions according to the layer where they are performed. Note that multiple layers can share one function.
 - (a) Routing
 - (b) Communicating to the immediate next neighbour
 - (c) Sending and receiving bits
 - (d) Providing end-to-end connectivity
 - (e) Framing bits
 - (f) Providing communication solution across networks to applications
 - (g) Flow control
 - (h) Error handling
 - (i) Connectionless data transfer
 - (j) Connection-oriented data transfer
 (Hint: The last four options relate to multiple layers)
13. Differentiate between OSI and TCP/IP model on different grounds.
14. What is the difference between bottom-up and top-down approaches? Does it make any difference if we choose one approach over another? Substantiate your answer.
15. Write down the name of each layer used in the OSI and TCP/IP models and write their usage.
16. The chapter includes a mail process description. Try to generate similar descriptions for the following:
 - (a) File transfer
 - (b) Remote login
 - (c) Browsing
17. Differentiate distributed systems from computer networks.
18. Peer-to-peer communication is radically different from client-server communication. Why?

Answers to Multiple-choice Questions

- | | | | | | | | | | |
|---------|---------|---------|--------|--------|--------|--------|--------|--------|---------|
| 1. (c) | 2. (d) | 3. (c) | 4. (b) | 5. (d) | 6. (c) | 7. (a) | 8. (b) | 9. (c) | 10. (a) |
| 11. (c) | 12. (d) | 13. (a) | | | | | | | |