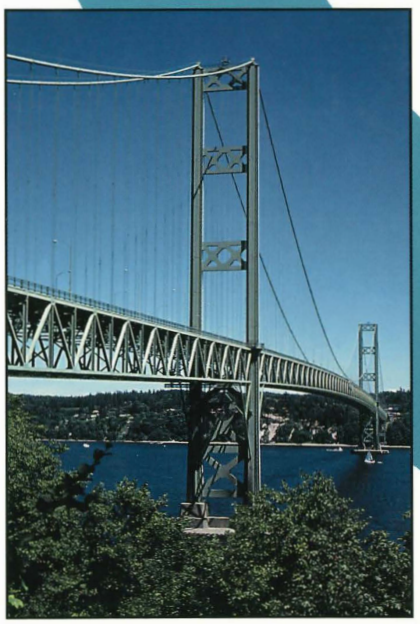
Interconnections
Second Edition

Bridges, Routers, Switches, and Internetworking Protocols

Radia Perlman



ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

Interconnections, Second Edition

Bridges, Routers, Switches, and Internetworking Protocols

Radia Perlman

ADDISON-WESLEY
An imprint of Addison Wesley Longman, Inc.
Reading, Massachusetts Harlow, England Menlo Park, California
Berkeley, California Don Mills, Ontario Sydney
Bonn Amsterdam Tokyo Mexico City

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book and Addison-Wesley was aware of a trademark claim, the designations have been printed in initial caps or all caps.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers discounts on this book when ordered in quantity for special sales. For more information, please contact:

Corporate, Government, and Special Sales Group
Addison Wesley Longman, Inc.
One Jacob Way
Reading, Massachusetts 01867

Visit AW on the Web: www.awl.com/cseng/

Library of Congress Cataloging-in-Publication Data

Perlman, Radia.
Interconnections : bridges, routers, switches and internetworking protocols / Radia Perlman. — 2nd ed.
p. cm.
Includes index.
ISBN 0-201-63448-1
1. Routers (Computer networks) 2. Computer network protocols. 3. Local area networks (Computer networks) I. Title.
TK5105.543.P47 1999 99-37086
004.6'2—dc21 CIP

Copyright © 2000 by Addison Wesley Longman, Inc.

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, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher. Printed in the United States of America. Published simultaneously in Canada.

Acquisitions Editor: Karen Gettman
Production Coordinator: Jacquelyn Doucette
Compositor: Octal Publishing, Inc.
Cover Designer: Simone R. Payment

Text printed on recycled and acid-free paper.

ISBN 0201634481

2 3 4 5 6 7 CR 03 02 01 00

2nd Printing January 2000

Chapter 1
Essential Networking Concepts

This chapter introduces concepts that are essential to understanding the specific subfield of computer networking that includes bridges and routers. It covers the OSI reference model, including layering and service models, because this model is a useful basis for some vocabulary. It also discusses various dimensions along which network designs can differ, such as scope, scalability, robustness, and autoconfigurability. Chapter 1 also describes the typical techniques involved in providing reliable two-party communication because some of the techniques used by routers can interact with techniques used by other layers.

1.1 Layers

Understanding, designing, and building a computer network would be too difficult a task unless the problem were partitioned into smaller subtasks, traditionally by dividing the problem into layers. The idea behind layering is that each layer is responsible for providing a service to the layer above by using the services of the layer below.

Each layer communicates with its *peer* layer in another node through the use of a *protocol*. This communication is accomplished through direct communication with the layer below. The communication between layer n and layer n–1 is known as an *interface*.

The OSI (Open Systems Interconnection) Reference Model defines seven layers, as shown in Figure 1.1. There is nothing magic about the number seven or the functionality in the layers. The reference model was designed before the protocols themselves, and then committees were set up to design each of the layers. Many of the layers were subsequently subdivided into further layers. The distinction between the layers is not always clear. Bridges and routers are a good example of a case in which people should rightfully be confused about which layers are which. But semantic arguments about layers are not

Ex.1014 / Page 4 of 10
Cisco Systems, Inc.

very productive. Instead, the layering should be viewed as a useful framework for discussion and not as a bible.

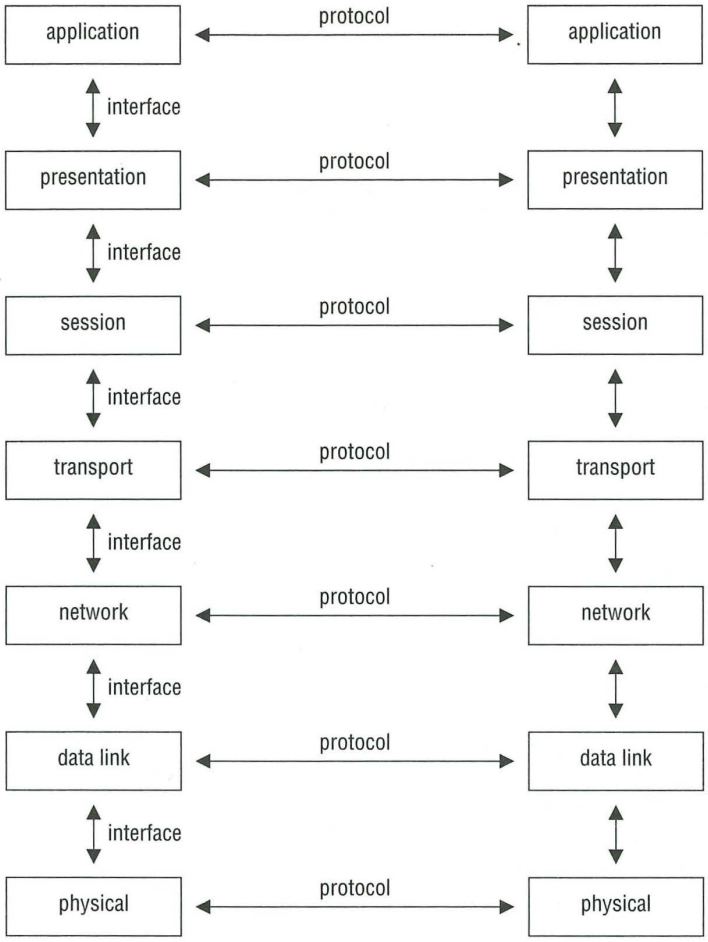


Figure 1.1 The OSI Reference Model

Layers defined by ISO:

1. *Physical layer:* The physical layer transmits bits of information across a link. It deals with such problems as size and shape of connectors, assignment of functions to pins, conversion of bits to electrical signals, and bit-level synchronization. It is usual for several different types of physical layers to exist within a network and even for multiple different types of physical layers to exist within a node, because each technology requires its own physical layer.

2. *Data link layer:* The data link layer (sometimes called the link layer) transmits chunks of information across a link. It deals with such problems as checksumming to detect data corruption; coordinating the use of shared media, as in a LAN (local area network); and addressing (when multiple systems are reachable, as in a LAN). Again, it is common for different links to implement different data link layers and for a node to support several data link layer protocols, one for each of the types of links to which the node is attached.

3. *Network layer:* The network layer enables any pair of systems in the network to communicate with each other. A *fully connected* network is one in which every pair of nodes has a direct link between its nodes, but this kind of topology does not scale beyond a few nodes. In the more typical case, the network layer must find a path through a series of connected nodes, and nodes along the path must forward packets in the appropriate direction. The network layer deals with such problems as route calculation, packet fragmentation and reassembly (when different links in the network have different maximum packet sizes), and congestion control.

4. *Transport layer:* The transport layer establishes a reliable communication stream between a pair of systems. It deals with errors that can be introduced by the network layer, such as lost packets, duplicated packets, packet reordering, and fragmentation and reassembly (so that the user of the transport layer can deal with larger-size messages and so that less-efficient network layer fragmentation and reassembly might be avoided). It is also nice if the transport layer reacts to congestion in the network by sending data more slowly in response.

5. *Session layer:* ISO had something in mind for this layer that doesn't seem useful to the Internet community. ISO's session layer offers services beyond the simple full-duplex reliable communication stream provided by transport, such as dialogue control (enforcing a particular pattern of communication between systems) and chaining (combining groups of packets so that either all or none of the packets in the group gets delivered). Whatever this layer is, it's irrelevant for bridges and routers.

6. *Presentation layer:* The goal of this layer is to agree on representations for data so that people defining structures don't have to worry about bit/byte order or what a floating point number looks like. ISO standardized on ASN.1 (Abstract Syntax Notation 1). Although I have yet to meet anyone who actually *likes* ASN.1—because it is complex and inefficient in both space and processing—a lot of the IETF (Internet Engineering Task Force) standards use it.

7. *Application layer:* As fascinating as bridging and routing is, it's actually because of applications that people want any of this stuff. Applications include file transfer, virtual terminal, Web browsing, and so on. It is common for multiple applications to be running concurrently in a node.

In this book, the data link layer is relevant because bridges operate within it and because the service provided by it is relevant to routers, which operate at the network layer, thereby making the network layer also relevant. The transport layer is somewhat relevant because it is a user of the network layer and because certain decisions that the network layer might make (such as whether to allow traffic to be split among several equivalent paths) affect the transport layer. The layers above transport are largely irrelevant to the study of bridges and routers.

Typically, the way layer n works is that it receives a chunk of data from layer n+1 along with additional information (such as the destination address) that might be required. Layer n must transmit the data to the layer n process in the destination node, which delivers it to the layer n+1 process in the destination node. Layer n often needs to include with the data certain information—for example, the address of the destination—that will be interpreted by other layer n entities. To get the information to the destination node, layer n hands down a buffer to layer n–1, including the data received from layer n+1 and the control information added by layer n. Additionally, layer n might pass other information in the layer n/n–1 interface along with the buffer.

Let's look at an example of how layering works. Assume that the physical layer allows a stream of bits to pass from one machine to another. The data link layer marks the bit stream so that the beginning and end of a packet can be found; it also adds a checksum to the packet so that the receiving machine can detect whether noise on the line introduced errors.

There are various interesting techniques to ensure that the marker indicating the beginning or end of the packet does not appear inside the data. In one technique, known as *bit stuffing*, the marker is a sequence of six 1's. To ensure that six consecutive 1's do not appear in the data portion of a packet, the transmitter adds an extra 0 after five consecutive 1's. The receiver knows that if the next bit after five consecutive 1's is a 0, then the 0 should be removed and ignored. If the next bit after five consecutive 1's is a 1, then it is a signal for the beginning or end of a packet. Another technique involves using different physical signals for data bits (1's and 0's) than for markers.

The network layer allows communication across multiple hops by cooperating with the network layers in all the connected machines to compute routes.

When the network layer receives a packet from the transport layer for transmission, the network layer adds an *envelope*, which consists of information glued onto the beginning (known as a *header*) and/or onto the end (known as a *trailer*). The envelope includes information such as the source and destination addresses. The network layer chooses an appropriate link on which to dispatch the packet and then hands the packet plus the network layer envelope to the data link layer process responsible for the outgoing link.

When the packet is received by an intermediate node, it is processed by the data link layer. Then the data link layer envelope is removed and the packet is passed up to the network layer, where the packet looks exactly the way it did when the previous network layer handed the packet to the data link layer—that is, it has everything transport sent down plus the network layer envelope. The network layer process at the receiving node looks at the network layer envelope, makes a decision as to the direction in which the packet should go based on that envelope, modifies the envelope as necessary (for example, incrementing a hop count to indicate how many nodes the packet has passed through), and gives the modified packet to the data link layer process responsible for the outgoing link (see Figure 1.2).

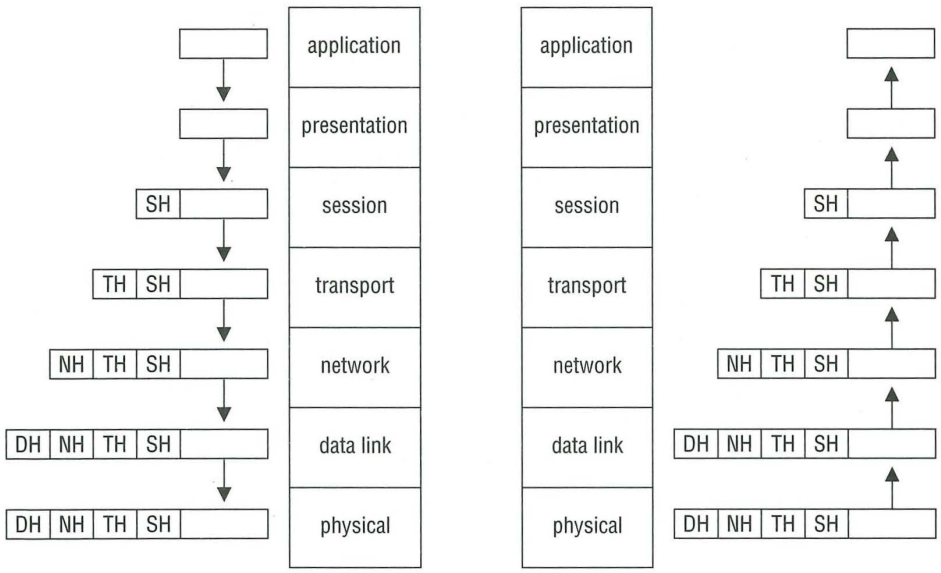


Figure 1.2 Envelopes added by lower layers

In the preceding description, words such as *packet* can be confusing. ISO has invented terminology that makes everything precise. Each layer communicates with its peer layer through a *protocol data unit*, or PDU. To make it clear which layer is being discussed, a single-letter prefix is added to PDU. The data link layer communicates with a peer data link layer by transmitting LPDUs. The network layer communicates with other

network layers through NPDUs. The transport layer communicates with other transport layers through TPDUs.

When layer n+1 gives information to layer n for transmission, the information is known as an SDU, or *service data unit*. As with PDUs, a single-letter prefix is added to eliminate ambiguity. When the transport layer wishes to transmit a TPDU to another transport layer, it must do so by giving the network layer an NSDU. The network layer takes the NSDU, adds an envelope, and transmits it (through the data link layer) as an NPDU (see Figure 1.3).

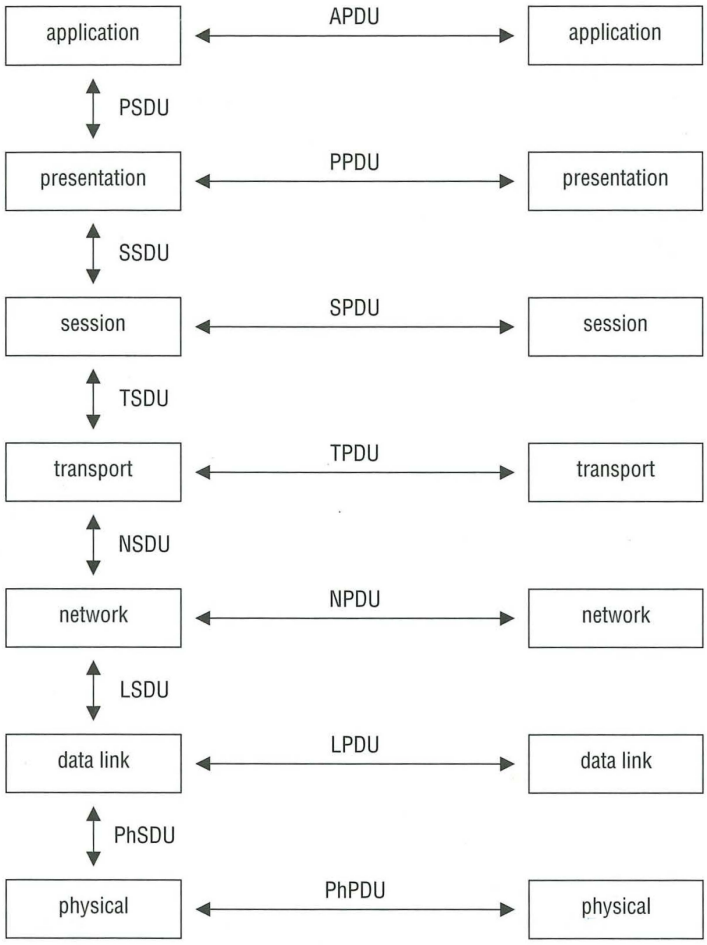


Figure 1.3 PDUs and SDUs

As a rule, the ISO terminology is not used in this book because it is wordy and hard to translate "in real time" until one has attended at least three standards meetings. However, the ISO terminology is used occasionally when it is necessary to be very precise.

1.2 Service Models

In general, the service provided by layer n–1 for layer n consists of transmitting data. Layer n provides layer n–1 with data (an SDU) plus extra information, such as the address of the destination. Layer n must also be able to receive data from a peer layer n, which it does by having layer n–1 inform it that data is available.

Layer n–1 can provide either connectionless or connection-oriented service. A *connectionless* service offers layer n two functions: it accepts packets for transmission from layer n, or it delivers packets to layer n.

With a *connection-oriented* service, a connection must be established before data can be transferred. Communication consists of three phases:

1. Connection setup
2. Data transfer (transmission or receipt of data)
3. Connection release

Associated with each of these phases are two functions: one in which layer n initiates the function and another in which layer n–1 informs layer n that its peer has initiated the function.

For connection setup, either layer n requests that a connection be set up to some destination address, or layer n–1 informs layer n that some layer n process in some other node is requesting a connection.

For connection release, either layer n requests that a connection be released, or layer n–1 informs layer n that the layer n process in the other node (or some other condition) requires release of the connection.

Various interfaces might provide other functions, but the preceding are the basic ones.

Services can also vary in their degree of reliability. A service that is purely *datagram* (also known as *best-effort*) accepts data but makes no guarantees as to delivery. Data may be lost, duplicated, delivered out of order, or mangled. A *reliable* service guarantees (or claims to guarantee) that data will be delivered in the order transmitted, without corruption (mangling of the bits), duplication, or loss. It is possible to build a connection-oriented network that is reliable (for example, X.25) or datagram (for example, ATM, Asynchronous Transfer Mode). A connectionless network layer only makes sense offering a datagram service because by definition nothing is keeping track of what gets delivered. IP, IPX, DECnet, CLNP, and AppleTalk are examples of connectionless network layers. Figure 1.4 shows examples of types of network layers.