NETWORKING WITH MICROSOFT TCP/IP

CERTIFIED ADMINISTRATOR'S RESOURCE EDITION

DREW HEYWOOD

ROB SCRIMGER

New Riders Publishing, Indianapolis, Indiana

Networking with Microsoft TCP/IP, Certified Administrator's Resource Edition

By Drew Heywood and Rob Scrimger

Published by:
New Riders Publishing
201 West 103rd Street
Indianapolis, IN 46290 USA

Printed in the United States of America 2 3 4 5 6 7 8 9 0

Library of Congress Cataloging-in-Publication Data

```
***CIP data available upon request***
```

ISBN: 1-56205-791-X

Warning and Disclaimer

This book is designed to provide information about TCP/IP. Every effort has been made to make this book as complete and as accurate as possible, but no warranty of fitness is implied.

The information is provided on an "as is" basis. The authors and New Riders Publishing shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

Associate Publisher	*David Dwyer*
Publishing Manager	*Laurie Petrycki*
Marketing Manager	*Kourtnaye Sturgeon*
Managing Editor	*Sarah Kearns*
Director of Development	*Kezia Endsley*

Product Development Specialist and Acquisitions Editor
Sean Angus

Development Editor
Ami Frank

Project Editor
Suzanne Snyder

Copy Editors
Jennifer Clark
Keith Cline, Karen Walsh

Technical Editor
Glenn Berg

Coordinator of Editorial Resources
Suzanne Snyder

Software Product Developer
Steve Flatt

Software Acquisitions and Development
Dustin Sullivan

Assistant Marketing Manager
Gretchen Schlesinger

Acquisitions Coordinator
Amy Lewis

Manufacturing Coordinator
Brook Farling

Cover Designer
Dan Armstrong

Cover Production
Nathan Clement

Book Designer
Glenn Larsen

Director of Production
Larry Klein

Production Team Supervisor
Laurie Casey

Graphics Image Specialists
Kevin Cliburn, Wil Cruz, Tammy Graham, Oliver Jackson

Production Analysts
Dan Harris
Erich J. Richter

Production Team
Lori Cliburn, Kim Cofer, Mary Hunt, Kristy Nash, Elizabeth San Miguel, Scott Tullis

Indexer
Tim Wright

The TCP/IP Model

When looking at the TCP/IP model for networking, you will notice only four layers. This is a result of the layers covering more functions. Figure 2.9 illustrates and the following list discusses the four layers of the TCP/IP model.

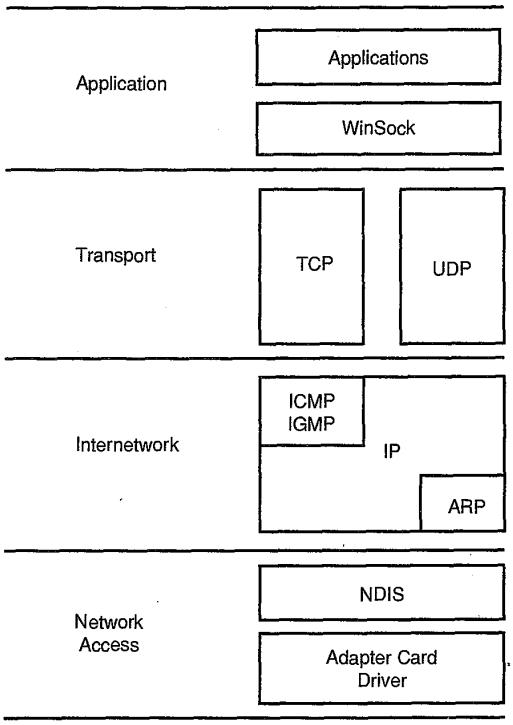

Figure 2.9
The layers in the TCP/IP protocol stack.

- **Application.** This combines the functions of both the Application and Presentation layers in the OSI model. The Application layer contains various services (protocols) such as NNTP (Network News Transfer Protocol) or SMTP (Simple Mail Transfer Protocol). The WinSock API is also in the Application layer.
- **Transport.** Just as in the OSI model, the Transport layer is the actual language of the network. All requests use one of two different transport protocols—either TCP (Transmission Control Protocol) or UDP (User Datagram Protocol).
- **Internet.** This replaces the network layer in the OSI model, and deals not only with finding other hosts (computers) on the same network, but with routing information (in the form of packets) to other networks.
- **Network Access.** Replaces the Data Link Control and Physical layers by treating them as one. This layer still handles framing the data and merging it to the wire, but the IP layer takes care of deciding which systems to send to.

send it by registered mail and have a guarantee of delivery. UDP (User Datagram Protocol) is like sending it regular mail—no guarantee. TCP (Transmission Control Protocol), on the other hand, creates a session, and can therefore guarantee delivery.

Overview of TCP

TCP is used, as stated several times earlier, to provide a connection-oriented delivery service for the higher-level protocols. To do this, TCP must first establish a session with the remote communicating host. It does this by means of a three-way handshake.

First the host initiating the communications sends a packet to the other host that contains information about itself and a SYN (or synchronize flag) telling the other host that a session is requested. The other host receives this packet and responds with information about itself—the SYN flag and an ACK (acknowledgment) of the information that it received. Finally the first host ACKs the information it received from the other, and a session now exists between the two systems.

Note You can view the sessions that your system currently has by using the NETSTAT command for strictly TCP/IP communications and NBTSTAT for NetBIOS sessions.

At the end of the communication session, a similar three-way handshake is used to drop the session with the remote host. This ensures that both of the hosts are through transmitting. It closes the session cleanly.

Overview of UDP

Compared to TCP, UDP is simple: The data from the upper-layer protocol is encapsulated and sent. UDP is used to send and receive simple messages; no session is required. The UDP protocol is used, for example, to send and receive broadcast messages.

The Internet Layer

The Internet layer has four main protocols. These protocols work together to provide a best-effort delivery service (guarantees are the responsibility of TCP or higher-level applications). IP (Internet Protocol) needs only to know which IP address to send the data to and the protocol on the other system (TCP or UDP) that should receive it.

The Internet Protocol (IP Layer)

Objective B.5

All devices that use TCP/IP have an Internet layer that includes the routers that provide the backbone for communications across the network. The IP is responsible for taking the packet and determining whether the packet is for the local network. If not, the IP must find a route for the packet to the destination network and eventually

When you are considering a network that spans the globe, you have to expect that problems connecting with specific hosts will sometimes arise. A few protocols now in place help to prevent this. Dynamic Routing is one that provides alternative routes if a link goes down. Another is the time out value that is given to each packet on the Internet. The time out represents (in theory) the maximum number of hops that a packet can make. By default in Windows NT, the time out or Time To Live (TTL) is 32 seconds. Each router is supposed to decrement the TTL by one for every second that the packet is in the router.

Today on the Internet, you will find that many routers decrement your TTL by far more than one. If the TTL expires or there is no route to the network you are trying to reach, you receive an ICMP message (request timed out or destination host unreachable).

ICMP also works to manage the flow of data on the Internet by directing traffic. If your router becomes overloaded, for example, and is unable to keep up, it might send a source quench message to your system. This tells your system to stop sending for a while. Routers also send an ICMP message if they detect that a better route to your destination is available. This would be an ICMP redirect message, telling your system to use another router.

Internet Group Management Protocol

This is the last of the protocols that reside in the lower layers of the TCP/IP stack. IGMP handles sending and receiving when groups of computers are involved. Sending to groups of computers is used to provide the systems that receive the information with a live feed. (Several radio stations do this on the Internet.) This is multicasting, which was mentioned earlier. In multicasting, you send the information from your system to a special IP address (a Class D address). You should remember that there were Class A, B, and C address. Class D, however, is only mentioned here; it is not valid as a host IP address.

When a system multicasts, it chooses an IP address (this has to be unique on the network) and sends all the information to that address. If you want to receive the information, you must tell your system to listen for that address. The problem is that your router does not know that it should listen for that address, and the packets don't get into your network. IGMP tells your router that you wish to listen to that address, enabling you to receive multicasts.

Network Access Layer

Just as in the OSI model, the Network Access layer is responsible for framing the packets of information for the underlying topology and merging the data on the wire. The Network Access layer also grabs the frames off the network. If they are for that MAC address or for broadcast/multicast, the Network Access layer passes them up to the appropriate protocol.

Headers

The data from Winsock becomes the data segment (normally UDP does not have a data segment) and then both a header and pseudo header are created. A *header* is a special data structure added to the data Winsock sent down. This header provides information about the destination and host computers, the protocols being used. The header also contains information that enables TCP to guarantee delivery. The purpose of the header is very different depending on whether it is TCP or UDP. It is, however, information for the corresponding protocol on the receiving hosts. The *pseudo header* is used to tell IP where the information is to be sent.

TCP Headers

TCP is a connection-oriented protocol, with a mechanism that guarantees delivery of information from one place to another. The header, therefore, must contain things such as the segment ID (so that you know whether you received them all), a CRC (cyclic redundancy check—a form of checksum verifies the information is intact). The TCP header looks like the image shown in figure 4.1.

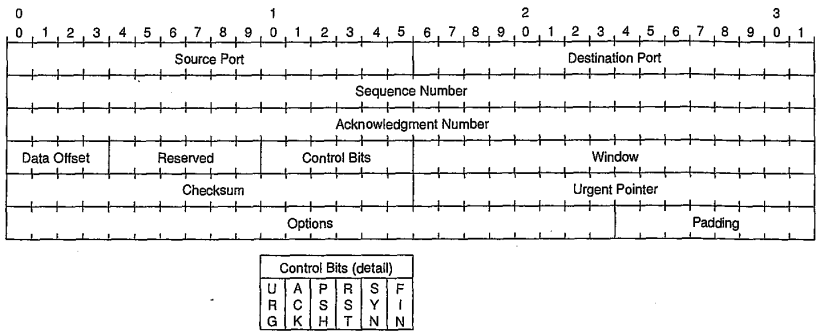

Figure 4.1

The format of a TCP header.

The information in the header is broken down into fields. The fields contain information required to enable the hosts to communicate. Included in the key information are the ports that each host is using, Sequence and Acknowledgment numbers, and the control bits. The following list identifies the entire contents of the header:

- **Source Port (16 bits).** Specifies the WinSock port sending the information.
- **Destination Port (16 bits).** Specifies the WinSock port to use on the receiving host.
- **Sequence Number (32 bits).** Specifies the sequence position of the first data byte in the segment. This enables the hosts to guarantee delivery by providing unique numbers for each segment that can be acknowledged by that number.