Chapter 18

Contax T / Kodak Tri-X

Gerogetown Walkabout

Network Programming Fundamentals

Learning Objectives

- *Demonstrate your understanding of basic networking concepts*
- *State the definition of the term "server hardware"*
- *State the definition of the term "server application"*
- *Describe the difference between server hardware and a server application*
- *State the definition of the term "client application"*
- *List and describe the different ways server and client applications can be physically and logically distributed*
- *Describe the properties of a multitiered application*
- *State the definition of the terms "protocol", "port", "packet", "datagram", "TCP/IP" and "UDP"*
- *List the requirements for testing both client and server applications on one computer*

Introduction

Network applications pervade today's modern computing environment. If you use email, a web browser, or a chat program like Windows Live Messenger, you're using software applications powered by network technology.

This chapter serves two primary purposes. First, it gives you a broad understanding of key networking concepts and terminology. Here you will learn the difference between server software and server hardware, the meanings of the terms *network*, *packet*, *datagram*, *TCP*/*IP* and *UDP*, and how applications can be physically and logically distributed in a networked environment.

The second purpose of this chapter is to introduce you to the concepts of *multitiered*, *distributed* applications. Modern network applications are often *logically tiered*, with one or more of their logical tiers physically deployed on different computers. It will be important for you to understand the terminology associated with these concepts as you learn to write network-enabled applications.

Upon completion of this chapter, you will have a solid foundation upon which to successfully approach *Chapter 19 - Networked Client-Server Applications*, and *Chapter 20 - Database Access & Multitiered Applications*. This chapter is not, however, a compendium on the topic of network programming or distributed applications. The subject is much too rich to adequately cover in one chapter and is quite beyond the scope of this book. If you are interested in pursuing something you learn here in more detail then I recommend you select one of the excellent sources listed in the references section and follow your interests. An excellent place to learn more about Internet programming is the Internet FAQ Archives: [http://www.faqs.org/faqs/]

What Is A Computer Network?

A *computer network* is an interconnected collection of computing devices. A computing device, for the purposes of this rather broad definition, can be any piece of equipment that exists to participate in or support a network in some fashion. Examples of computing devices include general purpose computers, special purpose computers, routers, switches, hubs, printers, etc.

Purpose Of A Network

Computer networks are built with a specific purpose in mind. The primary purpose of a computer network is *resource sharing*. A resource can be physical (*i.e.,* a printer or a computer) or metaphysical (*i.e.,* knowledge or data). Figure 18-1 shows a diagram for a simple computer network. This type of simple network is referred to as a *local area network* (LAN).

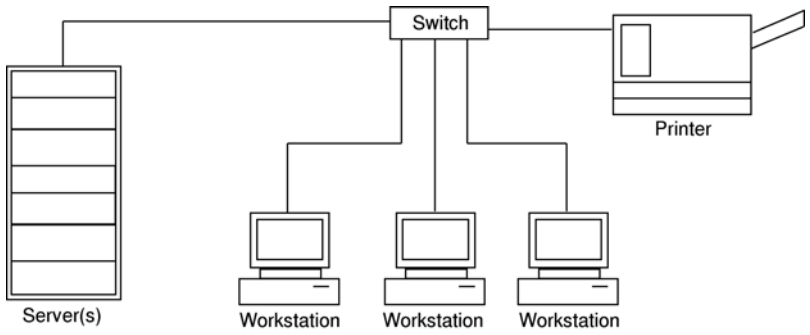


Figure 18-1: A Simple Computer Network

Referring to Figure 18-1 — the computing devices participating in this simple network include the workstations, the servers, the printer, and the switch. The switch facilitates network interconnection. In this configuration the work-

stations and servers can share the computational resources offered by each computer on the network as well as the printing services offered by the printer. Data can also be offered up for sharing on each computer as well.

The Role Of Network Protocols

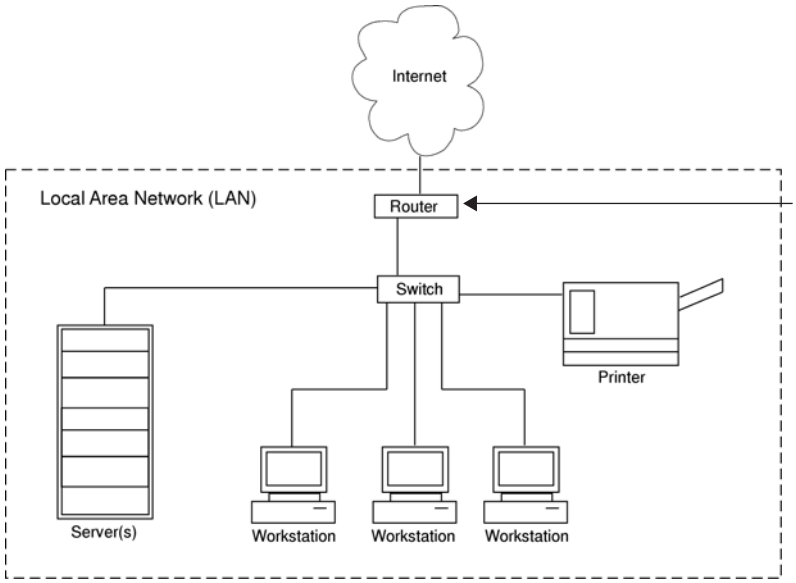
A *protocol* is a specification of rules that govern the conduct of a particular activity. Entities that implement or adhere to the protocol(s) for a given activity can participate in that activity. For example, Robert's Rules of Order specify a set of protocols for efficiently and effectively conducting meetings. A similar analogy applies to computer networking.

Homogeneous Vs. Heterogeneous Networks

Computers participating in a computer network communicate with each other via a set of networking protocols. There are generally two types of network environments: 1) *homogeneous* - where all the computers are built by the same company and can talk to each other via that company's proprietary networking protocol, or 2) *heterogeneous* - where the computers are built by different companies, have different operating systems, and therefore different proprietary networking protocols. An example of a homogenous network would be one comprised entirely of Apple Macintosh computers and Apple peripherals. The Macintosh computers could communicate perfectly fine with each other via AppleTalk which is an Apple networking protocol. In a perfect world, we would all use Apple Macintosh computers but the world is, alas, imperfect, and almost every network in existence is heterogeneous in nature. Apple Macs running OS X must communicate with computers running Sun Solaris, Microsoft Windows, Linux, and a host of other hardware and operating system combinations.

The Unifying Network Protocols: TCP/IP

In today's heterogeneous computer network environment, the protocols that power the Internet — Transmission Control Protocol (TCP) and Internet Protocol (IP) — collectively referred to as TCP/IP, have emerged as the standard network protocols through which different types of computers can talk to each other. Figure 18-2 shows the local area network connected to the Internet via a router. So long as the computers on the LAN utilize an operating system that implements TCP/IP, then they can access the computational and data resources made available both internally and via the Internet. If the LAN does not utilize TCP/IP, then a *bridge* or *gateway* device would be required to perform the necessary internetwork protocol translation.



The terms *router* and *gateway* are synonymous. A router sits at the edge of a local area network (LAN) and routes IP datagrams toward their destination on the Internet wide area network (WAN).

Figure 18-2: Local Area Network Connected to the Internet

What's So Special About The Internet?

What makes the Internet so special? The answer is — TCP/IP. The Internet is a vast network of computer networks. All of the networks on the Internet communicate with each other via TCP/IP. The TCP/IP protocols were developed with Department of Defense (DoD) funding. What the DoD wanted was a computer and communications network that was resilient to attack. If a piece of the Internet was destroyed by a nuclear blast, then data would be automatically routed through the surviving network connections. When one computer communicates with another computer via the Internet, the data it sends is separated into packets and transmitted one packet at a time to the designated computer. TCP/IP provides *packet routing* and *guaranteed packet delivery.* Because of the functionality provided by the TCP/IP protocols, the Internet is considered to be a robust and reliable way to transmit and receive data. Figure 18-3 shows how the simple network of Agency A can share resources with other agencies via the Internet.

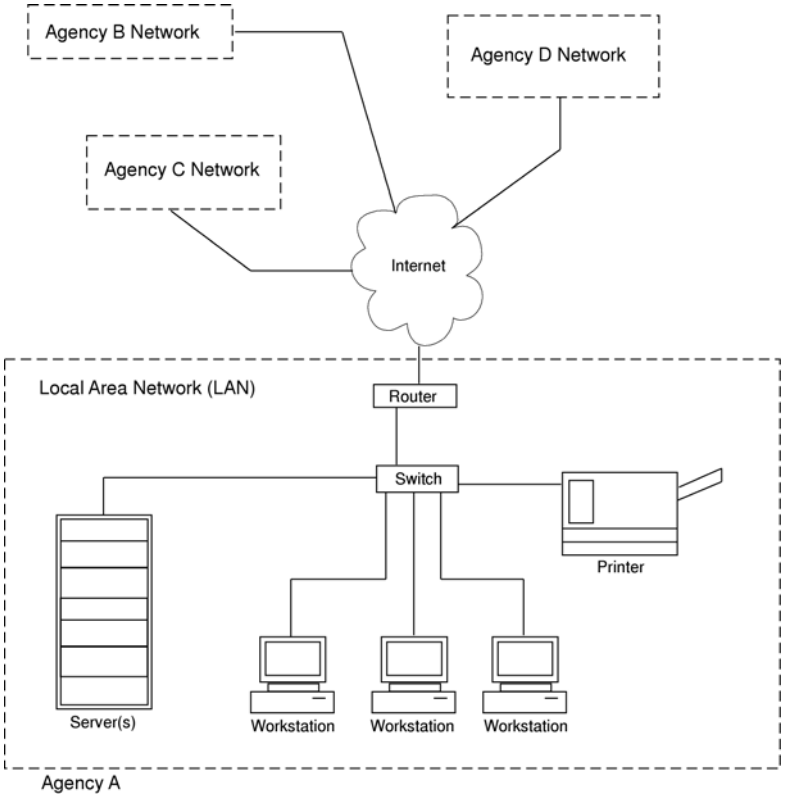


Figure 18-3: The Internet — A Network of Networks Communicating via Internet Protocols

Quick Review

A *computer network* is an interconnected collection of computing devices. Examples of computing devices include general purpose computers, special purpose computers, routers, switches, hubs, printers, etc. The primary purpose of a computer network is resource sharing. A resource can be physical (*i.e.*, a printer) or metaphysical (*i.e.*, data).

A *protocol* is a specification of rules that govern the conduct of a particular activity. Entities that implement or adhere to the protocol(s) for a given activity can participate in that activity. Computers participating in a computer network communicate with each other via a set of networking protocols. There are generally two types of network environments: 1) *homogeneous* - where all the computers are built by the same company and can talk to each other via that company's proprietary networking protocol, or 2) *heterogeneous* - where the computers are built by different companies, have different operating systems, and therefore different proprietary networking protocols. In today's heterogeneous computer network environment, the protocols that power the Internet — Transmission Control Protocol (TCP) and Internet Protocol (IP) — collectively referred to as TCP/IP, have emerged as the standard network protocols through which different types of computers can talk to each other.

What makes the Internet so special? The answer — TCP/IP. When one computer communicates with another computer via the Internet, the data it sends is separated into packets and transmitted a packet at a time to the designated computer. TCP/IP provides for packet routing and guaranteed packet delivery. Because of the functionality provided by the TCP/IP protocols, the Internet is considered to be a robust and reliable way to transmit and receive data.

Servers & Clients

The terms *server* and *client* each have both a hardware and software connotation. This section briefly discusses these terms in both aspects in greater detail to provide you with a foundation for the material presented in the next section.

Server Hardware And Software

The term *server* is often used to refer both to a piece of computing hardware on which a *server application* runs and to the server application itself. I will use the term server to refer to hardware. I will use the term *server application* to refer to a software component whose job is to provide some level of service to another entity.

As Figure 18-4 illustrates, it is the job of a server to host server applications. However, as desktop computing power increases, the lines between client and server hardware become increasingly blurry. A good definition for a server then is any computer used to host one or more server applications as its primary job. A server is usually (should be) treated as a critical piece of capital equipment within an organization. Server operating requirements are used to specify air conditioning, electrical, and flooring requirements for data centers. Servers are supported by data backup and recovery procedures and, if they are truly agency critical, will have some form of fault tolerance and redundancy designed in as well.

A server running a server application is also referred to as a *host*. The term host extends to any computer running any application.

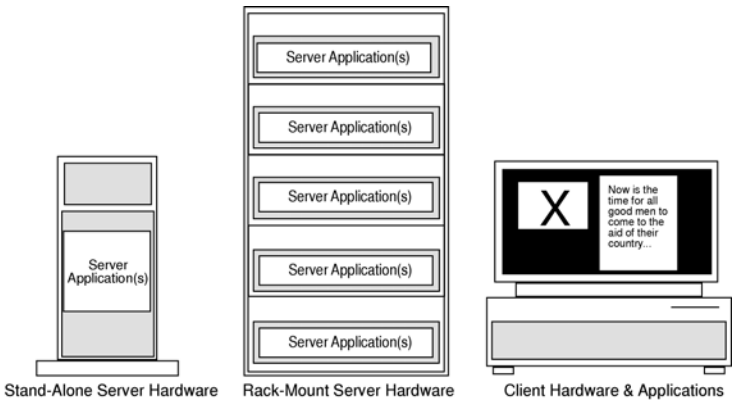


Figure 18-4: Client and Server Hardware and Applications

Client Hardware And Software

The term *client* is also used to describe both hardware and software. Client hardware is any computing device that hosts an application that requires or uses the services of a server application. Client software is any application that requires or uses the services provided by a server application. For example, when you run Microsoft Internet Explorer on your home computer, you are running a client application. You use Internet Explorer to access web sites via the Internet. These web sites are served up by a web server *(i.e.*, an HTTP server), which is a server application hosted on a server somewhere out there in Internet land.

Quick Review

The terms *server* and *client* each have both a hardware and software aspect. The term server is often used to refer both to a piece of computing hardware on which a server application runs and to the server application itself. A good definition for a server then is any computer used to run one or more server applications as its primary job. A server running a server application is also referred to as a *host*. The term host extends to any computer running any application. The term client is also used to describe both hardware and software. Client hardware is any computing device that hosts an application that requires or uses the services of a server application. Client software is any application that requires or uses the services provided by a server application.

Application Distribution

The term *application distribution* refers to where (*i.e.,* on what physical computer) one or more pieces of a network application reside. This section discusses the concepts of physically distributing client and server applications. Server applications themselves can be further divided into multiple application layers with each distinct application layer being physically deployed to one or more computers. The concepts associated with multilayered applications are presented and discussed in the next section.

Physical Distribution On One Computer

Client and server applications can both be deployed on the same computer. This is most often done for the purposes of testing during development. When you write client-server applications in Chapter 19, you will test them on your development machine. If you are fortunate enough to have a home network that includes multiple computers, you can test your client-server applications in a more real world setting. Figure 18-5 illustrates the concept of running client and server applications on the same physical hardware.

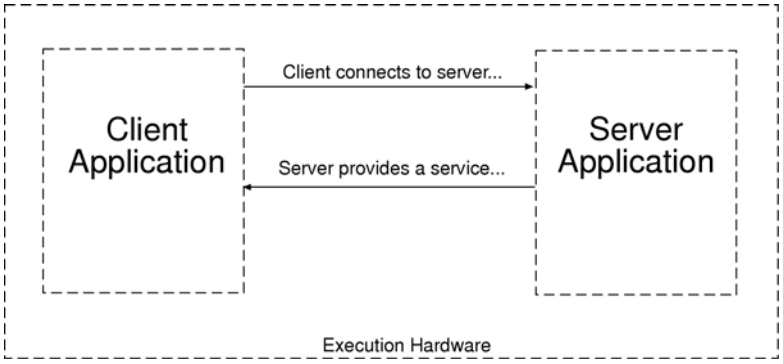


Figure 18-5: Client and Server Applications Physically Deployed to the Same Computer

Running Multiple Clients On The Same Computer

You can run multiple client applications on the same computer. To do this, your server application must be capable of handling multiple concurrent client requests for service. A server application with this capability is generally referred to as being *multithreaded*. Each incoming client connection is passed off to a unique thread for processing. The execution of multiple client applications, in addition to the server application, on the same hardware, is common practice during a software project's development and testing phases. Figure 18-6 illustrates the concept of running multiple client applications and the server application on the same hardware.

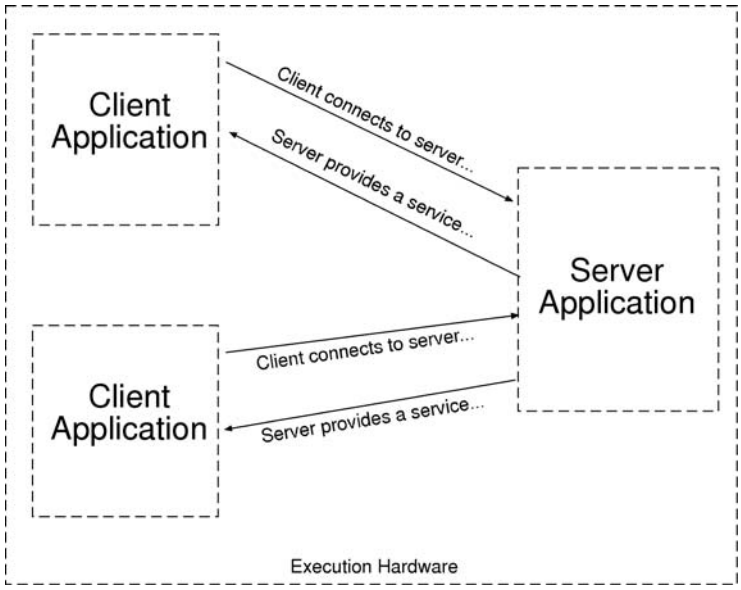


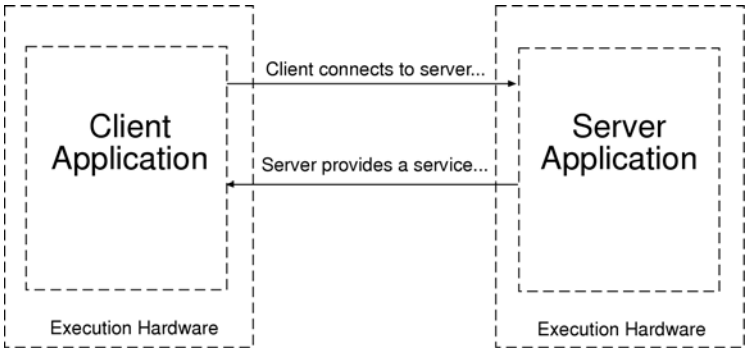
Figure 18-6: Running Multiple Clients on Same Hardware

Addressing The Local Machine

When testing client-server applications on your local machine, you can use the localhost IP address of 127.0.0.1 as the server application's host address.

Physical Distribution Across Multiple Computers

Although client and server applications can be co-located on the same hardware, it is more often the case that they are physically deployed on different machines geographically separated by great distance. Figure 18-7 illustrates this concept.



These computers may be in the same room on the same local area network (LAN) or each may be located half-way around the planet from the other connected via the Internet.

Figure 18-7: Client and Server Applications Deployed on Different Computers

Quick Review

The term *application distribution* refers to where (*i.e.*, on what physical computer) one or more pieces of a network application reside. Client and server applications can be deployed to the same physical computer or to different physical computers. These computers may be in the same room or located a great distance from each other.

Multitiered Applications

Up until now I have referred to client and server applications as if they were monolithic components. In reality, modern client-server applications are logically segmented into functional layers. These layers are also referred to as *application tiers*. An application composed of more than one tier is referred to as a *multitiered* application. This section discusses the concepts related to multitiered applications in greater detail.

Logical Application Tiers

Figure 18-8 illustrates the concept of a multitiered application.

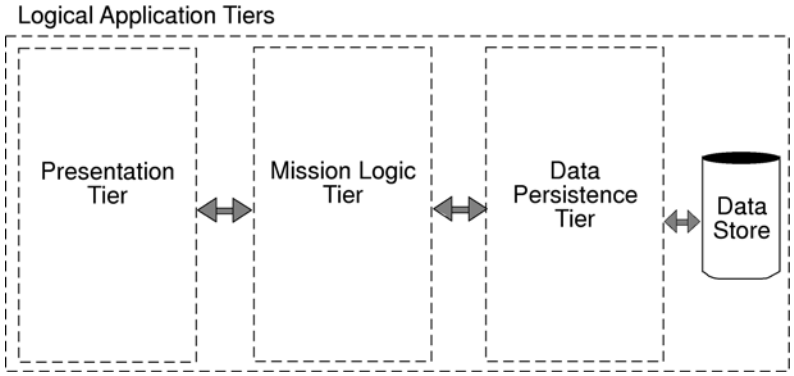


Figure 18-8: A Multitiered Application

Referring to Figure 18-8 — in this example the application comprises three functional tiers: 1) *presentation tier*, 2) *mission logic tier*, and 3) *data persistence tier*. As their names suggest, each tier has a distinct responsibility for delivering specific application functionality. The presentation tier is concerned with rendering the user interface. The mission logic tier (*a.k.a.* business logic tier) contains the code that implements the application's services. (*i.e.,* data processing algorithms, mission-oriented processes, etc.) (I use the term mission logic tier interchangeably with the term business logic tier when referring to multitiered applications written for Department of Defense clients.) The data persistence tier is responsible for servicing the data needs (*i.e.,* data storage and retrieval) of the mission logic layer as quickly and reliably as possible.

Another way to think about each tier's responsibilities is as a separation of concerns:

- the presentation tier is concerned with how a user interacts with an application
- the mission logic tier is concerned with implementing mission support processes
- the data persistence tier is concerned with reliable data storage and retrieval in support of mission processes

Physical Tier Distribution

The logical application tiers may be physically deployed on the same computer, as is illustrated in Figure 18-9.

It is more likely the case, however, that logical application tiers are physically deployed to separate and distinct computing nodes located some distance apart. Figure 18-10 illustrates this concept by showing each logical tier deployed to a different computer. In between this extreme lies any combination of logical tier deployments as best supports an agency's mission requirements

Quick Review

Client and server applications can be logically separated into distinct functional areas called *tiers*. Applications logically segmented in this fashion are referred to as *multitiered applications*.

Three possible logical application tiers include: 1) *the presentation tier*, which is concerned with rendering the application's user interface, 2) the *mission logic tier*, which is concerned with implementing mission process logic,

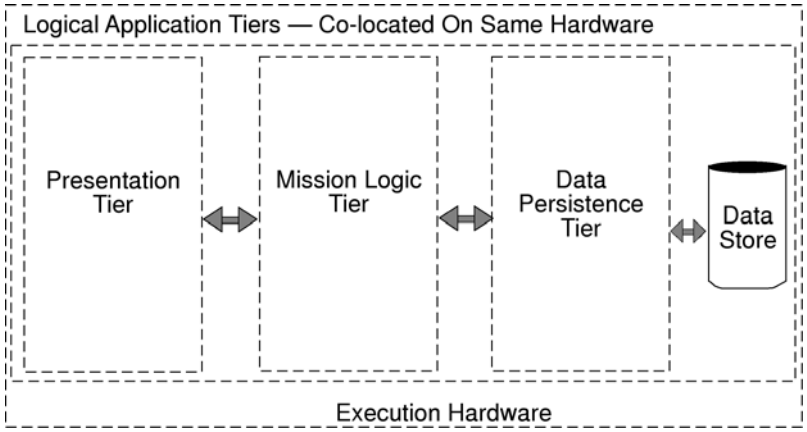


Figure 18-9: Physically Deploying Logical Application Tiers on Same Computer

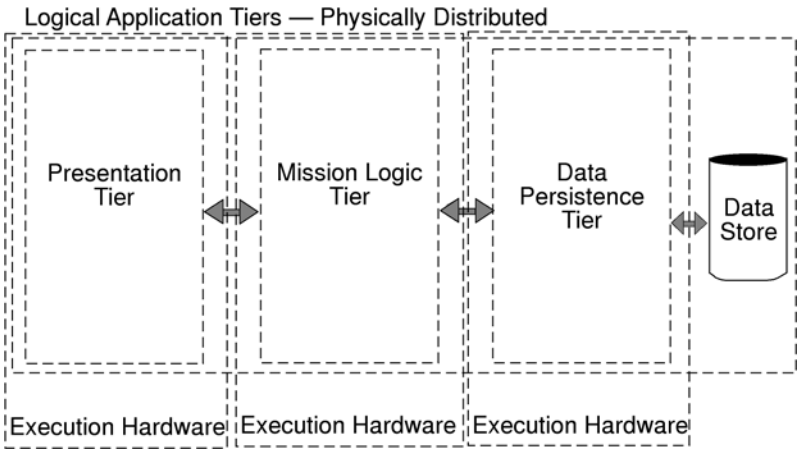


Figure 18-10: Logical Application Tiers Physically Deployed to Different Computers

and 3) the *data persistence tier*, which is concerned with the quick and reliable delivery of data to the mission logic tier. A multitiered application can be physically deployed on one computer or across several computers geographically separated by great distances.

Internet Networking Protocols: Nuts & Bolts

This section discusses the concepts associated with the Internet protocols and related terminology in greater detail. You'll find this background information helpful when navigating your way through the System.Net namespace looking for a solution to your network programming problem.

The Internet Protocols: TCP, UDP, And IP

The Internet protocols facilitate the transmission and reception of data between participating client and server applications in a *packet-switched network* environment. The term packet-switched network means that data traveling along network pathways is divided into small, routable packages referred to as *packets*. If a communication link between two points on a network goes down, the packets are routed through remaining network connections to their intended destination.

The Internet protocols work together as a layered *protocol stack* as is shown in Figure 18-11.

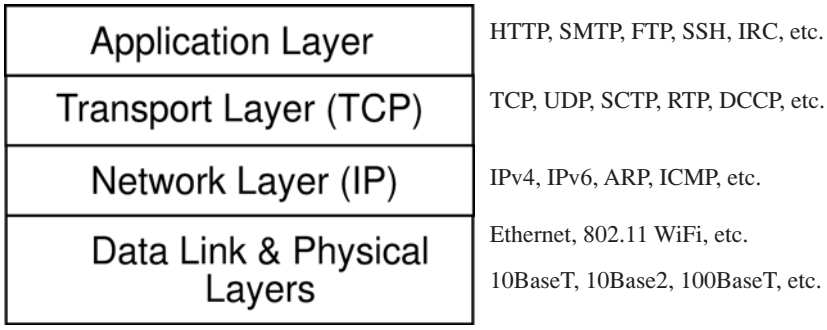


Figure 18-11: TCP/IP Protocol Stack

The layered protocol stack consists of the *application layer*, the *transport layer*, the *network layer*, the *data link layer*, and the *physical layer*. Each protocol stack layer provides a set of services to the layer above it. Several examples of protocols that may be employed at each level in the application stack are also shown in Figure 18-11. For more information on protocols not discussed in this chapter, please consult the sources listed in the references section.

The Application Layer

The *application layer* represents any internet enabled application that requires the services of the transport layer. Typical applications you may be familiar with include File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), TELNET, or a custom internet application such as one you might write. The application layer relies on services provided by the transport layer.

Transport Layer

The purpose of the *transport layer* is to provide host-to-host, connection-oriented, data transmission service to the application layer. Two internet protocols that function at the transport layer include the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). The .NET Framework supports both of these protocols directly in that normal network communication takes place using TCP, but UDP can be utilized if required.

Transmission Control Protocol (TCP)

The purpose of TCP is to provide highly reliable, host-to-host communication. To achieve this, TCP manages several important issues including basic data transfer, reliability, flow control, multiplexing, connections, and precedence and security.

The sending and receiving TCP modules work together to achieve the level of service mandated by the TCP protocol. The sending TCP module packages *octets* of data into *segments*, which it forwards to the network layer and the Internet Protocol (IP) for further transmission. TCP tags each octet with a sequence number. The receiving TCP module signals an acknowledgement when it receives each segment and orders the octets according to sequence number, eliminating duplicates and properly handling those that may have been received out of order.

In short — TCP guarantees data delivery and saves you the worry.

User Datagram Protocol (UDP)

UDP is used to send and receive data as quickly as possible without the overhead incurred when using TCP. UDP is an extremely lightweight protocol when compared with TCP. It provides direct access to the IP datagram level. However, the quick data transmission provided by UDP comes at a price. Data is *not* guaranteed to arrive at its intended destination when sent via UDP.

Now, you might ask yourself, “Self, what’s UDP good for?” Generally speaking, any application that needs to send data quickly and doesn’t particularly care about lost datagrams might stand to benefit from using UDP. Examples include data streams where previously sent data is of little or no use because of its age. (*i.e.,* stock market quote streams, voice transmissions, etc.)

In short — UDP is faster than TCP but unreliable.

Network Layer

The *network layer* is responsible for the routing of data traffic between internet hosts. These hosts may be located on a local area network or on another network somewhere on the Internet. The Internet Protocol (IP) resides at this layer and provides data routing services to the transport layer protocols TCP or UDP.

Internet Protocol (IP)

The Internet Protocol (IP) is a *connectionless* service that permits the exchange of data between hosts without a prior call setup. (Hence the term connectionless.) It packages data submitted by TCP or UDP into blocks called *datagrams*. IP uses *IP addresses* and *routing tables* to properly route datagrams to their intended destination networks.

Data Link And Physical Layers

The *data link* and *physical* layers are the lowest layers of the networking protocol stack. It is here that data is placed "on the wire" for transmission across the LAN or across the world.

The Data Link Layer

The *data link layer* sits below the network layer and is responsible for the transmission of data across a particular communications link. It provides for flow control and error correction of transmitted data. An example protocol that operates at the data link layer is Ethernet.

The Physical Layer

The *physical layer* is responsible for the actual transmission of data across the physical communication lines. Physical layer protocols concern themselves with the types of signals used to transmit data. (*i.e.,* electrical, optical, etc.) and the type of media used to convey the signals (*i.e.,* fiber optic, twisted pair, coaxial, etc.).

Putting It All Together

Computers that participate in a TCP/IP networking environment must be running an instance of the TCP/IP protocol stack as is illustrated in Figure 18-12.

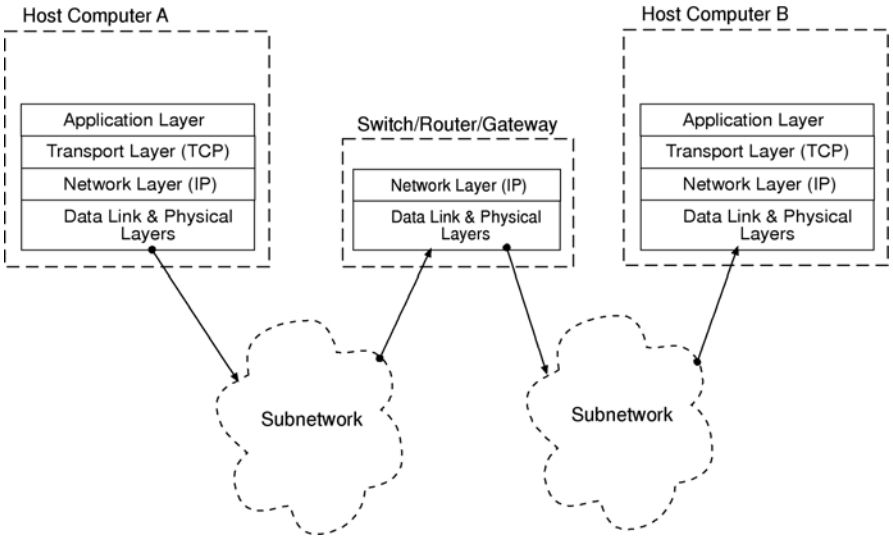


Figure 18-12: Internet Protocol Stack Operations

Referring to Figure 18-12 — when Host Computer A sends data to Host Computer B via the Internet, the data is passed from the application layer to the physical layer on Host Computer A and sent to the gateway that links the two subnetworks. At the gateway the packets are passed back up to the network layer to determine the forwarding address, then repackaged and sent to the destination computer. When the packets arrive at Host Computer B they are passed back up the protocol stack and the original data is presented to the application layer.

What You Need To Know

Now that you have some idea of what's involved with moving data between host computers on the Internet or on a local area network using the Internet protocols, you can pretty much forget about all these nasty details. The .NET Framework provides a set of classes in the System.Net namespace that makes network programming easy.

Quick Review

The *Internet protocols* facilitate the transmission and reception of data between participating client and server applications in a *packet-switched network* environment. The term packet-switched network means that data traveling along network pathways is divided into small, routable packages referred to as *packets*. If a communication link between two points on a network goes down, the packets can be routed through remaining network connections to their intended destination.

The Internet protocols work together as a *layered protocol stack*. The layered protocol stack consists of the *application layer*, the *transport layer*, the *network layer*, the *data link layer*, and the *physical layer*. Each layer in the protocol stack provides a set of services to the layer above it.

The application layer represents any Internet enabled application that requires the services of the transport layer. The purpose of the transport layer is to provide host-to-host, connection-oriented, data transmission service to the application layer. Two Internet protocols that function at the transport layer include the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). TCP guarantees data delivery and saves you the worry. UDP is faster than TCP but unreliable.

The network layer is responsible for the routing of data traffic between Internet hosts. The Internet Protocol (IP) is a *connectionless* service that permits the exchange of data between hosts without a prior call setup. (Hence the term connectionless.) It packages data submitted by TCP or UDP into blocks called *datagrams*. IP uses *IP addresses* and *routing tables* to properly route datagrams to their intended destination networks.

The data link and physical layers are the lowest layers of the networking protocol stack. The data link layer sits below the network layer and is responsible for the transmission of data across a particular communications link. The physical layer is responsible for the actual transmission of data across the physical communication lines.

Computers that participate in a TCP/IP networking environment must be running an instance of the TCP/IP protocol stack. The TCP/IP protocol stack is part of a computer's operating system.

Summary

A *computer network* is an interconnected collection of computing devices. Examples of computing devices include general purpose computers, special purpose computers, routers, switches, hubs, printers, personal digital assistants (PDAs), etc. The primary purpose of a computer network is *resource sharing*. A resource can be physical (*i.e.*, a printer) or metaphysical (*i.e.*, data).

A *protocol* is a specification of rules that govern the conduct of a particular activity. Entities that implement the protocol(s) for a given activity can participate in that activity. Computers participating in a computer network communicate with each other via a set of networking protocols. There are generally two types of network environments: 1) *homogeneous* - where all the computers are built by the same company and can talk to each other via that company's proprietary networking protocol, or 2) *heterogeneous* - where the computers are built by different companies, have different operating systems, and therefore different proprietary networking protocols. In today's heterogeneous computer network environment the protocols that power the Internet — Transmission Control Protocol (TCP) and

Internet Protocol (IP) — collectively referred to as TCP/IP, have emerged as the standard network protocols through which different types of computers can talk to each other.

What makes the Internet so special? The answer — TCP/IP. When one computer communicates with another computer via the Internet the data it sends is separated into *packets* and transmitted a packet at a time to the designated computer. TCP/IP provides for packet routing and guaranteed packet delivery. Because of the functionality provided by the TCP/IP protocols the Internet is considered to be a robust and reliable way to transmit and receive data.

The terms *server* and *client* each have both a hardware and software aspect. The term server is often used to refer both to a piece of computing hardware on which a server application runs and to the server application itself. A good definition for a server then is any computer used to host one or more server applications as its primary job. A server running a server application is also referred to as a *host*. The term host extends to any computer running any application. The term client is also used to describe both hardware and software. Client hardware is any computing device that hosts an application that requires or uses the services of a server application. Client software is any application that requires or uses the services provided by a server application.

The term *application distribution* refers to where (*i.e.*, on what physical computer) one or more pieces of a network application reside. Client and server applications can be deployed to the same physical computer, but most likely they are deployed to different machines.

Client and server applications can be logically separated into distinct functional areas called *tiers*. Applications logically segmented in this fashion are referred to as *multitiered applications*. Three possible logical application tiers include: 1) the *presentation tier*, which is concerned with rendering the application's user interface, 2) the *mission logic tier*, which is concerned with implementing mission process logic, and 3) the *data persistence tier*, which is concerned with the quick and reliable delivery of data to the mission logic tier. A multitiered application can be physically deployed on one computer or across several computers geographically separated by great distances.

The *Internet protocols* facilitate the transmission and reception of data between participating client and server applications in a *packet-switched* network environment. The term packet-switched network means that data traveling along network pathways is divided into small, routable packages referred to as *packets*. If a communication link between two points on a network goes down the packets can be routed through remaining network connections to their intended destination.

The Internet protocols work together as a *layered protocol stack*. The layered protocol stack consists of the *application layer*, the *transport layer*, the *network layer*, the *data link layer*, and the *physical layer*. Each layer in the protocol stack provides a set of services to the layer above it.

The application layer represents any internet enabled application that requires the services of the transport layer. The purpose of the transport layer is to provide host-to-host, connection-oriented, data transmission service to the application layer. Two internet protocols that function at the transport layer include the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). TCP guarantees data delivery and saves you the worry. UDP is faster than TCP but unreliable.

The network layer is responsible for routing data traffic between internet hosts. The Internet Protocol (IP) is a connectionless service that permits the exchange of data between hosts without a prior call setup. It packages data submitted by TCP or UDP into blocks called *datagrams*. IP uses *IP addresses* and *routing tables* to properly route datagrams to their intended destination networks.

The data link and physical layers are the lowest layers of the networking protocol stack. The data link layer sits below the network layer and is responsible for the transmission of data across a particular communications link. The physical layer is responsible for the actual transmission of data across the physical communication lines.

Computers that participate in a TCP/IP networking environment must be running an instance of the TCP/IP protocol stack. The TCP/IP protocol stack is provided by a computer's operating system.

Skill-Building Exercises

1. **Web Research:** Expand your understanding of the TCP/IP protocols. Search the web for the Internet RFCs used as references for this chapter.

2. **Web Research:** Expand your understanding of network applications. Search the web for material related to packet-

switched networks, distributed applications, and multitiered applications.

Suggested Projects

1. None

Self-Test Questions

1. What is a computer network? What is the primary purpose of a computer network?

2. Describe the two types of computer networking environments.

3. Describe the purpose of the TCP/IP Internet networking protocols.

4. What's the difference between the terms *server* and *server application*? Client and client application?

5. Describe the relationship between a server application and client application.

6. List and describe at least two ways network applications can be distributed.

7. What term is used to describe a server application that can handle multiple simultaneous client connections?

8. What term is used to describe a network application logically divided into more than one functional layer? List and describe the purpose of three possible functional layers.

9. List and describe the purpose of the layers of the Internet protocol stack. Describe how data is transmitted from one computer to another via the Internet protocols.

10. What's the difference between TCP and UDP?

11. What services does IP provide?

References

RFC 791 - Internet Protocol

RFC 2396 - Uniform Resource Identifiers (URI): General Syntax

RFC 793 - Transmission Control Protocol

RFC 768 - User Datagram Protocol

Uyless Black. *Advanced Internet Technologies*. Prentice Hall Series In Advanced Communications Technologies. Prentice Hall PTR, Upper Saddle River, NJ. ISBN: 0-13-759515-8

Notes