

## **Networking Fundamentals**

In order to get the most of The TCP/IP Guide, a certain level of knowledge regarding the basics of networking is very helpful. Unlike many other resources, however, I did not want to start with the assumption that my reader knew what networking was all about. After all, that's why you are reading this Guide!

So, I decided to include this section, which provides an overview of some of the basic issues related to networking. This area is where I place discussions of some of the most fundamental networking concepts and ideas. It serves not only to provide you, the reader, with useful background material, but also as a repository for “general” information so that I don't need to repeat it in many different places elsewhere in the Guide. This in turn allows those who already know about these basics to avoid having to skip over them in many other locations.

In this section I have provided several subsections that introduce networking and talk about some of its most important fundamental concepts. First, I introduce networking in very broad terms, discussing what it is and why it is important. Then, I describe several fundamental characteristics of networks that you will need to understand the various networking technologies. I discuss the different general types and sizes of networks and how they are differentiated. I then talk about many different matters related to network performance. I explain the importance of networking standards and standards organizations. Finally, I provide a background section that describes the fundamentals of how data is stored and manipulated in computers; if you are new to computing you may find that information useful when reading some parts of this Guide.

## **Introduction to Networking**

In this day and age, networks are everywhere. The Internet has also revolutionized not only the computer world, but the lives of millions in a variety of ways even in the “real world”. We tend to take for granted that computers should be connected together. In fact, these days, whenever I have two computers in the same room, I have a difficult time **not** connecting them together!

Given the ubiquitousness of networking, it's hard to believe that the field is still a relatively young one, especially when it comes to hooking up small computers like PCs. In approaching any discussion of networking, it is very useful to take a step back and look at networking from a high level. What is it, exactly, and why is it now considered so important that it is assumed that most PCs and other devices should be networked?

In this section, I provide a quick introduction to networking, discussing what it is all about in general terms. I begin by defining networking in the most general terms. I then place networking in an overall context by describing some of its advantages and benefits, as well as some of its disadvantages and costs.

## **What Is Networking?**

For such an extensive and involved subject, which includes so many different technologies, hardware devices and protocols, the definition of networking is actually quite simple. A network is simply a collection of computers or other hardware devices that are connected together, either physically or logically, using special hardware and software, to allow them to exchange information and cooperate. Networking is the term that describes the processes involved in designing, implementing, upgrading, managing and otherwise working with networks and network technologies.



**Key Concept:** A network is a set of hardware devices connected together, either physically or logically to allow them to exchange information.

Networks are used for an incredible array of different purposes. In fact, the definitions above are so simple for the specific reason that networks can be used so broadly, and can allow such a wide variety of tasks to be accomplished. While most people learning about networking focus on the interconnection of PCs and other “true” computers, you use various types of networks every day. Each time you pick up a phone, use a credit card at a store, get cash from an ATM machine, or even plug in an electrical appliance, you are using some type of network.

In fact, the definition can even be expanded beyond the world of technology altogether: I’m sure you’ve heard the term “networking” used to describe the process of finding an employer or employee by talking to friends and associates. In this case too, the idea is that independent units are connected together to share information and cooperate.

The widespread networking of personal computers is a relatively new phenomenon. For the first decade or so of their existence, PCs were very much “islands unto themselves”, and were rarely connected together. In the early 1990s, PC networking began to grow in popularity as businesses realized the advantages that networking could provide. By the late 1990s, networking in homes with two or more PCs started to really take off as well.

This interconnection of small devices represents, in a way, a return to the “good old days” of mainframe computers. Before computers were small and personal, they were large and centralized machines that were shared by many users operating remote terminals. While having all of the computer power in one place had many disadvantages, one benefit was that all users were connected because they shared the central computer.

Individualized PCs took away that advantage, in favor of the benefits of independence. Networking attempts to move computing into the middle ground, providing PC users with the best of both worlds: the independence and flexibility of personal computers, and the connectivity and resource sharing of mainframes. In fact, networking is today considered so vital that it's hard to conceive of an organization with two or more computers that would not want to connect them together!

## **The Advantages (Benefits) of Networking**

You have undoubtedly heard the “the whole is greater than the sum of its parts”. This phrase describes networking very well, and explains why it has become so popular. A network isn't just a bunch of computers with wires running between them. Properly implemented, a network is a system that provides its users with unique capabilities, above and beyond what the individual machines and their software applications can provide.

Most of the benefits of networking can be divided into two generic categories: *connectivity* and *sharing*. Networks allow computers, and hence their users, to be connected together. They also allow for the easy sharing of information and resources, and cooperation between the devices in other ways. Since modern business depends so much on the intelligent flow and management of information, this tells you a lot about why networking is so valuable.

Here, in no particular order, are some of the specific advantages generally associated with networking:

- **Connectivity and Communication:** Networks connect computers and the users of those computers. Individuals within a building or work group can be connected into *local area networks (LANs)*; LANs in distant locations can be interconnected into larger *wide area networks (WANs)*. Once connected, it is possible for network users to communicate with each other using technologies such as electronic mail. This makes the transmission of business (or non-business) information easier, more efficient and less expensive than it would be without the network.

- **Data Sharing:** One of the most important uses of networking is to allow the sharing of data. Before networking was common, an accounting employee who wanted to prepare a report for her manager would have to produce it on his PC, put it on a floppy disk, and then walk it over to the manager, who would transfer the data to her PC's hard disk. (This sort of "shoe-based network" was sometimes sarcastically called a "sneakernet".)

True networking allows thousands of employees to share data much more easily and quickly than this. More so, it makes possible applications that rely on the ability of many people to access and share the same data, such as databases, group software development, and much more. [Intranets and extranets](#) can be used to distribute corporate information between sites and to business partners.

- **Hardware Sharing:** Networks facilitate the sharing of hardware devices. For example, instead of giving each of 10 employees in a department an expensive color printer (or resorting to the "sneakernet" again), one printer can be placed on the network for everyone to share.
- **Internet Access:** The Internet is itself an enormous network, so whenever you access the Internet, you are using a network. The significance of the Internet on modern society is hard to exaggerate, especially for those of us in technical fields.
- **Internet Access Sharing:** Small computer networks allow multiple users to share a single Internet connection. Special hardware devices allow the bandwidth of the connection to be easily allocated to various individuals as they need it, and permit an organization to purchase one high-speed connection instead of many slower ones.
- **Data Security and Management:** In a business environment, a network allows the administrators to much better manage the company's critical data. Instead of having this data spread over dozens or even hundreds of small computers in a haphazard fashion as their users create it, data can be centralized on shared servers. This makes it easy for everyone to find the data, makes it possible for the administrators to ensure that the data is regularly backed up, and also allows for the implementation of security measures to control who can read or change various pieces of critical information.
- **Performance Enhancement and Balancing:** Under some circumstances, a network can be used to enhance the overall performance of some applications by distributing the computation tasks to various computers on the network.

- **Entertainment:** Networks facilitate many types of games and entertainment. The Internet itself offers many sources of entertainment, of course. In addition, many multi-player games exist that operate over a local area network. Many home networks are set up for this reason, and gaming across wide area networks (including the Internet) has also become quite popular. Of course, if you are running a business and have easily-amused employees, you might insist that this is really a **disadvantage** of networking and not an advantage!



**Key Concept:** At a high level, networks are advantageous because they allow computers and people to be connected together, so they can share resources. Some of the specific benefits of networking include communication, data sharing, Internet access, data security and management, application performance enhancement, and entertainment.

Well, if that list isn't enough to convince you that networking is worthwhile, then... I have **no** idea what it is you do with your computers! ☺ At any rate, it's quite possible that only some of the above items will match your particular circumstances, but at least one will definitely apply to almost every situation, assuming you own or manage more than one computer.

## The Disadvantages (Costs) of Networking

Now that I have portrayed the [great value and many useful benefits of networking](#), I must bring you crashing back to earth with that old nemesis of the realistic: TANSTAAFL. For those who are not Heinlein fans, this acronym stands for “There Ain’t No Such Thing As A Free Lunch”. Even though networking really does represent a “whole that is greater than the sum of its parts”, it does have some real and significant costs and drawbacks associated with it.

Here are a few of the items that balance against the advantages of networking.

- **Network Hardware, Software and Setup Costs:** Computers don't just magically network themselves, of course. Setting up a network requires an investment in hardware and software, as well as funds for planning, designing and implementing the network. For a home with a small network of two or three PCs, this is relatively inexpensive, possibly amounting to less than a hundred dollars with today's low prices for network hardware, and operating systems already designed for networks. For a large company, cost can easily run into tens of thousands of dollars—or more.

- **Hardware and Software Management and Administration Costs:** In all but the smallest of implementations, ongoing maintenance and management of the network requires the care and attention of an IT professional. In a smaller organization that already has a system administrator, a network may fall within this person's job responsibilities, but it will take time away from other tasks. In more substantial organizations, a network administrator may need to be hired, and in large companies an entire department may be necessary.
- **Undesirable Sharing:** With the good comes the bad; while networking allows the easy sharing of useful information, it also allows the sharing of undesirable data. One significant "sharing problem" in this regard has to do with viruses, which are easily spread over networks and the Internet. Mitigating these effects costs more time, money and administrative effort.
- **Illegal or Undesirable Behavior:** Similar to the point above, networking facilitates useful connectivity and communication, but also brings difficulties with it. Typical problems include abuse of company resources, distractions that reduce productivity, downloading of illegal or illicit materials, and even software piracy. In larger organizations, these issues must be managed through explicit policies and monitoring, which again, further increases management costs.
- **Data Security Concerns:** If a network is implemented properly, it is possible to greatly improve the security of important data. In contrast, a poorly-secured network puts critical data at risk, exposing it to the potential problems associated with hackers, unauthorized access and even sabotage.

Most of these costs and potential problems can be managed; that's a big part of the job of those who set up and run networks. In the end, as with any other decision, whether to network or not is a matter of weighing the advantages against the disadvantages. Of course today, nearly everyone decides that networking *is* worthwhile.



**Key Concept:** Networking has a few drawbacks that balance against its many positive aspects. Setting up a network has costs in hardware, software, maintenance and administration. It is also necessary to manage a network to keep it running smoothly, and to address possible misuse or abuse. Data security also becomes a much bigger concern when computers are connected together.



## **Segments, Networks, Subnetworks and Internetworks**

One of the reasons that networks are so powerful is that they can not only be used to connect computers together, but to connect groups of computers together. Thus, network connections can exist at multiple levels; one network can be attached to another network, and that entire whole can be attached to another set of networks, and so on. The ultimate example of this is, of course, the Internet, a huge collection of networks that have been interconnected into... dare I say a "Web"? ☺

This means a larger network can be described as consisting of several smaller networks or even parts of networks that are linked together. Conversely, we can talk about taking individual networks or network portions and assembling them into larger structures. The reason why this concept is important is that certain technologies are best explained when looking at an entire large network at a high level, while others really require that we drill down to the detailed level of how constituent network pieces work.

### ***Common Terms Describing the Size of Networks***

Over time, a collection of terms has evolved in the networking world to describe the relative sizes of larger and smaller networks. Understanding these different terms is important not only for helping you comprehend what you read about networks, but also because they are important concepts in network design. This is particularly true for local area networking, where decisions regarding how to set up segments and how to connect them to each other have an important impact on the overall performance and usability of the network. Here are some of the most common ones.

#### **Network**

This is the least specific of the terms mentioned here. Basically, a network can be of pretty much any size, from two devices to thousands. When networks get very large, however, and are clearly comprised of smaller networks connected together, they are often no longer called networks but internetworks, as we will see momentarily. Despite this, it is fairly common to hear someone refer to something like "Microsoft's corporate network", which obviously contains thousands or even tens of thousands of machines.

#### **Subnetwork (Subnet)**

A subnetwork is a portion of a network, or a network that is part of a larger internetwork. This term is also a rather subjective one; subnetworks can in fact be rather large when they are part of a network that is very large.

The abbreviated term “subnet” can refer generically to a subnetwork, but also has a [specific meaning in the context of TCP/IP addressing](#).

### Segment (Network Segment)

A segment is a small section of a network. In some contexts, a segment is the same as a subnetwork and the terms are used interchangeably. More often, however, the term “segment” implies something smaller than a subnetwork. Networks are often designed so that, for the sake of efficiency, with computers that are related to each other or that are used by the same groups of people put on the same network segment.

This term is notably problematic because it is routinely used in two different ways, especially in discussions related to Ethernet. The earliest forms of Ethernet used coaxial cables, and the coax cable itself was called a “segment”. The segment was shared by all devices connected to it, and became the *collision domain* for the network (a phrase referring generally to a collection of hardware devices where only one can transmit at a time.)

Each Ethernet physical layer had specific rules about how many devices could be on a segment, how many segments could be connected together, and so on, depending on what sort of network interconnection devices were being used. Devices such as hubs and repeaters were used to extend collision domains by connecting together these segments of cable into wider networks. Over time, the terms “collision domain” and “segment” started to be used interchangeably. Thus today a “segment” can refer either to a specific piece of cable, or to a collection of cables connected electrically that represent a single collision domain.



**Note:** As if that potential ambiguity in the use of the word “segment” isn’t bad enough, it also has another, totally unrelated meaning: it is the name of [the messages sent in the Transmission Control Protocol](#)!

### Internetwork (or Internet)

Most often, this refers to a larger networking structure that is formed by connecting together smaller ones. Again, the term can have either a generic or a specific meaning, depending on context. In some technologies, an internetwork is just a very large network that has networks as components. In others, a network



is differentiated from an internetwork based on how the devices are connected together.

An important example of the latter definition is TCP/IP, where a network usually refers to a collection of machines that are linked at [layer two of the OSI Reference Model](#), using technologies like Ethernet or Token Ring and interconnection devices such as hubs and switches. An internetwork is formed when these networks are linked together at [layer three](#), using routers that pass Internet Protocol datagrams between networks. Naturally, this is highly simplified, but in studying TCP/IP you should keep this in mind when you encounter the terms “network” and “internetwork”.



**Note:** The shorter form of the word internetwork (“internet”) is often avoided by people who wish to avoid confusion with the proper noun form (“The Internet”). The latter of course refers only to the [well-known global internetwork of computers](#) and all the services it provides. I personally try to use the word “internetwork” most of the time in this Guide instead of “internet”, for this very reason.



**Key Concept:** Several terms are often used to describe the relative sizes of networks and parts of networks. The most basic term is *network* itself, which can refer to most anything, but often means a set of devices connected using an OSI layer two technology. A *subnetwork* is a part of a network (or internetwork), as is a *segment*, though the latter often has a more specific meaning in certain technologies. An *internetwork* refers either generically to a very large network, or specifically to a set of layer-two networks connected using routers at layer three.

## Performance Measurements: Speed, Bandwidth, Throughput and Latency

There are a number of terms that are commonly used to refer to various aspects of network performance. Some of them are quite similar to each other, and you will often see them used—and in many cases, misused or even **abused**. ☺ It's a good idea for us to take a look at each of them, therefore, discuss how they are commonly used and what they really mean.

More than just the issue of different terms related to performance, however, is the more important reality that there are multiple **facets** to performance. Depending on the application, the manner in which data is sent across the

network may be more important than the raw speed at which it is transported. In particular, many multimedia applications require real-time performance; they need data sent in such a manner that it will be delivered steadily. For these purposes, raw speed isn't as important as **consistent** speed, and this is an issue that is often not properly recognized.

### ***Performance Measurement Terms***

Let's take a look at the most common performance measurement terms and see what they are all about.

#### **Speed**

This is the most generic performance term used in networking. As such, it can mean just about **anything**. Most commonly, however, it refers to the *rated* or *nominal* speed of a particular networking technology. For example, Fast Ethernet has a nominal speed of 100 megabits per second; it is for that reason often called 100 Mbit Ethernet, or given a designation such as "100BASE-TX".

Rated speed is the biggest "performance magic number" in networking—you see it used to label hardware devices, and many people bandy the numbers about as if they actually were the real "speed of the network". The problem with using nominal speed ratings is that they are *theoretical* only, and as such, tell an incomplete story. No networking technology can run at its full rated speed, and many run **substantially** below it, due to [real-world performance factors](#).

Speed ratings such as "100 Mbps Ethernet" are also often referred to as the "throughput" of a technology, even though the maximum theoretical speed of a technology is more analogous to bandwidth than throughput, and the two are not identical. More on this in the next two bullet points.

#### **Bandwidth**

Bandwidth is a widely-used term that usually refers to the data-carrying capacity of a network or data transmission medium. It indicates the maximum amount of data that can pass from one point to another in a unit of time. The term comes from the study of electromagnetic radiation, where it refers to the width of a band of frequencies used to carry data. It is usually given in a theoretical context, though not always.


Bandwidth is still used in these two senses: "frequency band width" and data capacity. For example, radio frequencies are used for wireless technologies, and the bandwidth of such technologies can refer to how wide the RF band is. More

commonly, though, it refers to how much data can be sent down a network, and is often used in relative terms. For example, for Internet access, a cable or xDSL is considered “high bandwidth” access; using a regular analog modem is “low bandwidth”.

## Throughput

Throughput is a measure of how much actual data can be sent per unit of time across a network, channel or interface. While throughput can be a theoretical term like bandwidth, it is more often used in a practical sense, for example, to measure the amount of data actually sent across a network in the “real world”. Throughput is limited by bandwidth, or by rated speed: if an Ethernet network is rated at 100 megabits per second, that's the absolute upper limit on throughput, even though you will normally get quite a bit less. So, you may see someone say that they are using 100 Mbps Ethernet but getting throughput of say, 71.9 Mbps on their network.

The terms bandwidth and throughput are often used interchangeably, even though they are really not exactly the same, as I just discussed.

 **Key Concept:** The three terms used most often to refer to the overall performance of a network are *speed*, *bandwidth*, and *throughput*. These are related and often used interchangeably, but are not identical. The term *speed* is the most generic and often refers to the rated or nominal speed of a networking technology. *Bandwidth* can refer either to the of a frequency band used by a technology, or more generally to data capacity, where it is more of a theoretical measure. *Throughput* is a specific measure of how much data flows over a channel in a given period of time. It is usually a practical measurement.

## Latency

This very important, often overlooked term, refers to the *timing* of data transfers on a communications channel or network. One important aspect of latency is how long it takes from the time a request for data is made until it starts to arrive. Another aspect is how much control a device has over the timing of the data that is sent, and whether the network can be arranged to allow for the consistent delivery of data over a period of time. Low latency is considered better than high latency.

## ***Applying Performance Measurement Terms***

As with all networking terms, there are no hard and fast rules; many people are rather loose with their use of the terms above. You will even see terms such as “throughput bandwidth”, “bandwidth throughput” and other charming inventions from the department of redundancy department. ☺ More often, you will just see a lot of mish-mashed term usage, and especially, spurious conclusions being drawn about what data streams a network can handle based on its rated speed. Making matters worse is that speed ratings are usually specified in bits per second, but throughput may be given in bits or bytes per second.

In general, “speed”, bandwidth and throughput get a lot of attention, while latency gets little. Yet latency considerations are very important for many real-time applications such as streaming audio and video and interactive gaming. In fact, they are often more important than raw bandwidth.

For example, suppose you move to a rural home and your choices for Internet access are a regular 28.8 kbps modem connection or fancy satellite Internet. The companies selling satellite connectivity call it “broadband” and advertise very high rated speeds—400 kbps or more. They make a big deal about it being “over 10 times as fast as dialup” and they certainly charge a lot for this very high-tech service. This is a slam dunk, right?

Wrong. The satellite connection has high bandwidth, but very poor (high) latency due to the time it takes for the signals to travel to and from the satellite. It is definitely much better than the modem for downloading that nice little 150 MB patch from Microsoft. However, it is much **worse** than the modem for playing the latest online video game with your buddy over the Internet, because of the latency, or *lag*, in transmissions. Every move you make in your game will be delayed for over half a second as the signal bounces around between the satellite and the earth, making online gaming nearly impossible. Thus, whether satellite Internet is worth the extra money depends entirely on what you plan to use it for.



**Related Information:** An important issue closely related to latency is quality of service, a general term that refers (among other things) to the ability of networks to deliver necessary bandwidth and reliable data transfer for applications that need it. [See the topic devoted to this subject later in this section.](#)



**Key Concept:** Where bandwidth and throughput indicate how fast data moves across a network, *latency* describes the nature of how it is conveyed. It is most often used to describe the delay between the time that data is

requested and the time when it arrives. A networking technology with very high throughput and bad (high) latency can be worse for some applications than one with relatively low throughput but good (low) latency.

## **Network Structural Models and Client/Server and Peer-to-Peer Networking**

I mentioned in [my discussion of the advantages of networking](#) that networks are normally set up for two primary purposes: *connectivity* and *sharing*. If you have a network with a number of different machines on it, each computer can interact with the hardware and software of the others, to enable a variety of tasks to be performed. How precisely this is done depends to a large degree on the overall design of the network.

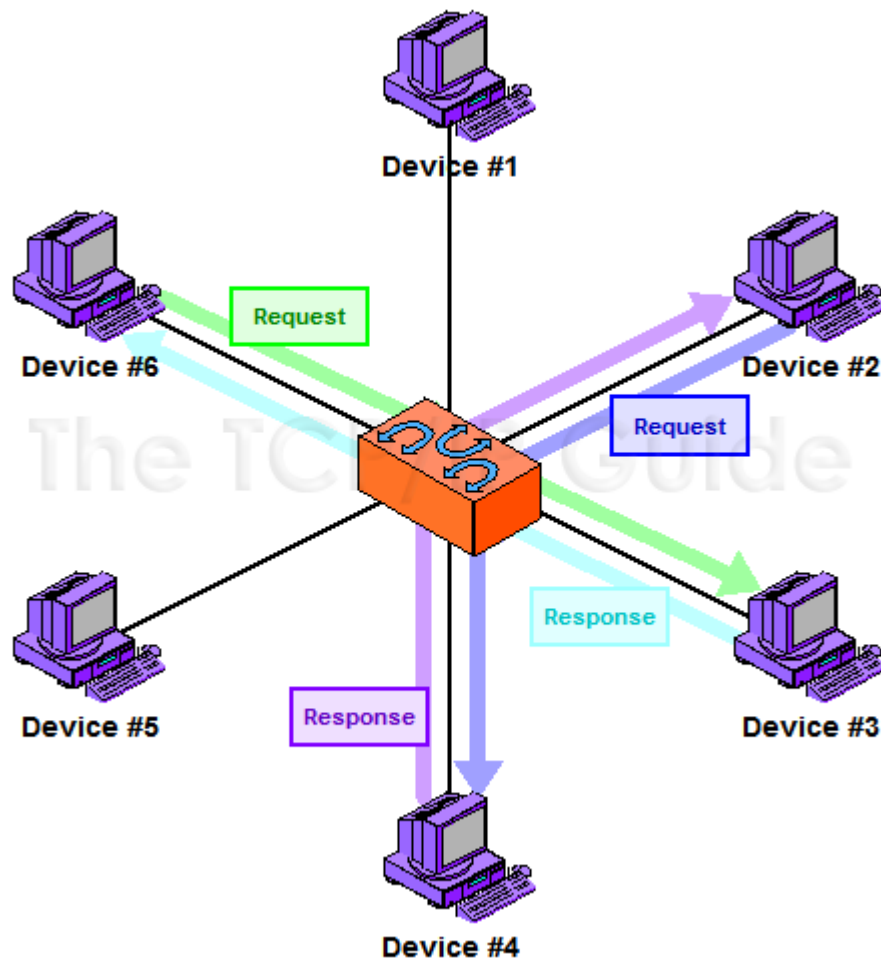
### ***Resource Sharing Roles and Structural Models***

One very important issue in network design is how to configure the network for the sharing of resources. Specifically, the network designer must decide whether or not to dedicate resource management functions to the devices that constitute it. In some networks, all devices are treated equal in this regard, while in others, each computer is responsible for a particular job in the overall function of providing services. In this latter arrangement, the devices are sometimes said to have *roles*, somewhat like actors in a play.

Two common terms are used to describe these different approaches to setting up a network, sometimes called choosing a *structural model*.

### **Peer-to-Peer Networking**

In a strict peer-to-peer networking setup, every computer is an equal, a *peer* in the network. Each machine can have resources that are shared with any other machine. There is no assigned role for any particular device, and each of the devices usually runs similar software. Any device can and will send requests to any other, as illustrated in Figure 5.



**Figure 5: Peer-to-Peer Networking**

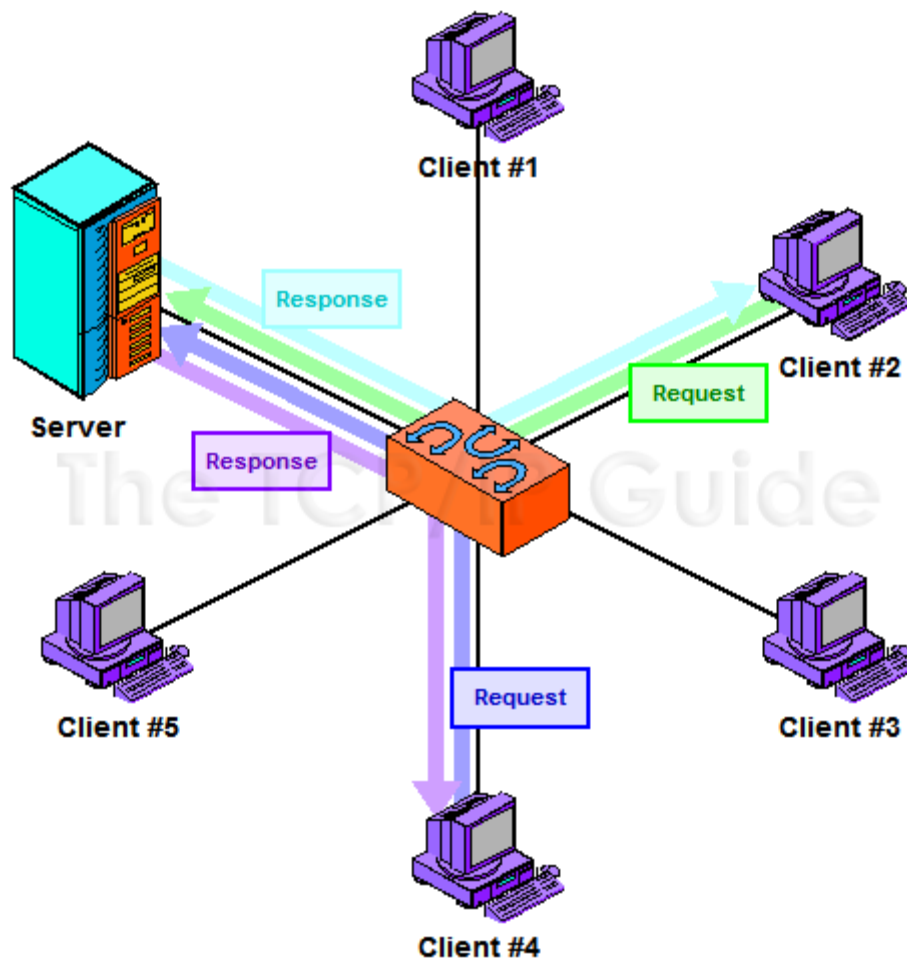
In this model, each device on the network is treated as a peer, or equal. Each device can send requests and responses, and none are specifically designated as performing a particular role. This model is more often used in very small networks. Contrast to Figure 6.

### **Client/Server Networking**

In this design, a small number of computers are designated as centralized *servers* and given the task of providing services to a larger number of user machines called *clients*. The servers are usually powerful computers with a lot of memory and storage space, and fast network connections. The clients are typically smaller, “regular” computers like PCs, optimized for human use.



The term “client/server” also frequently refers to protocols and software, which are designed with matching, complementary components. Usually, server software runs on server hardware, and client software is used on client computers that connect to those servers. Most of the interaction on the network is between client and server, and not between clients, as shown in Figure 6. Server software is designed to efficiently respond to requests, while client software provides the interface to the human users of the network.



**Figure 6: Client/Server Networking**

In the client/server model, a small number of devices are designated as servers and equipped with special hardware and software that allows them to more efficiently interact simultaneously with multiple client machines. While the clients can still interact with each other, most of the time they send requests of various sorts to the server, and the server sends back responses to them. Contrast this

to the peer-to-peer networking example in Figure 5.



**Key Concept:** Networks are usually configured to share resources using one of two basic *structural models*. In a *peer-to-peer network*, each device is an equal and none are assigned particular jobs. In a *client/server network*, however, devices are assigned particular roles—a small number of powerful computers are set up as *servers* and respond to requests from the other devices, which are *clients*. Client/server computing also refers to the interaction between complementary protocol elements and software programs, and is rising in popularity due to its prevalence in TCP/IP and Internet applications.

### ***Comparing Client/Server and Peer-to-Peer Networking***

The choice of client/server or peer-to-peer is another where there is no “right answer” in this regard. Which should be used depends entirely on the needs of the particular network.

Peer-to-peer networking has primary advantages of simplicity and low cost, which means it has traditionally been used on small networks. Client/server networking provides advantages in the areas of performance, scalability, security and reliability, but is more complicated and expensive to set it up. This makes it better-suited to larger networks. Over time, however, there has been a steady evolution towards client/server networking, even on smaller networks. Many years ago it was common to see even networks with 20 to 50 machines using the peer-to-peer model; today, even networks with only a half-dozen machines sometimes are set up in a client/server mode because of the advantages of centralized resource serving.

The rise in popularity of client/server networking is ironic because in some ways, it is actually a throwback to the days of large mainframes decades ago. A mainframe with attached terminals can be thought of as a client/server network with the mainframe itself being the server and the terminals being clients. This analogy is not perfect, of course, because modern client computers do a lot more work than dumb terminals do on mainframes.

One of the reasons why the client/server structural model is becoming dominant is that it is the primary model used by the world’s largest network: the Internet. Client/server architecture is the basis for most TCP/IP protocols and services. For example, the term “Web browser” is really another name for a “Web client”, and a “Web site” is really a “Web server”.



**Related Information:** For more information on client/server computing, I recommend you read [the topic devoted to TCP/IP client/server operation](#). That topic also contains a very relevant exposition on the different meanings of the terms “client” and “server” in hardware, software and transactional contexts.

## Connection-Oriented and Connectionless Protocols

In [the previous topic](#) I described and contrasted networking technologies based on whether or not they use a dedicated path, or *circuit*, over which to send data. Another way in which technologies and protocols are differentiated has to do with whether or not they use *connections* between devices. This issue is closely related to the matter of packet versus circuit switching.

### ***Division of Protocols into Connection-Related Categories***

Protocols are divided into two categories based on their use of connections:

- **Connection-Oriented Protocols:** These protocols require that a logical connection be established between two devices before transferring data. This is generally accomplished by following a specific set of rules that specify how a connection should be initiated, negotiated, managed and eventually terminated. Usually one device begins by sending a request to open a connection, and the other responds. They pass control information to determine if and how the connection should be set up. If this is successful, data is sent between the devices. When they are finished, the connection is broken.
- **Connectionless Protocols:** These protocols do not establish a connection between devices. As soon as a device has data to send to another, it just sends it.



**Key Concept:** A *connection-oriented* protocol is one where a logical connection is first established between devices prior to data being sent. In a *connectionless* protocol, data is just sent without a connection being created.

### ***The Relationship Between Connection Orientation and Circuits***

You can probably immediately see the relationship between the concepts of circuits and connections. Obviously, in order to establish a circuit between two devices, they must also be connected. For this reason, circuit-switched networks

are inherently based on connections. This has led to the terms “circuit-switched” and “connection-oriented” being used interchangeably.

However, this is an oversimplification that results due to a common logical fallacy—people make the mistake of thinking that if A implies B, then B implies A, which is like saying that since all apples are fruit, then all fruit are apples. A connection is needed for a circuit, but a circuit is **not** a prerequisite for a connection. There are, therefore, protocols that are connection-oriented, while not being predicated on the use of circuit-based networks at all.

These connection-oriented protocols are important because they enable the implementation of applications that require connections, over packet-switched networks that have no inherent sense of a connection. For example, to use the TCP/IP [File Transfer Protocol](#), you want to be able to connect to a server, enter a login and password, and then execute commands to change directories, send or retrieve files, and so on. This requires the establishment of a connection over which commands, replies and data can be passed. Similarly, the [Telnet Protocol](#) obviously involves establishing a connection—it lets you remotely use another machine. Yet, both of these work (indirectly) over the IP protocol, which is based on the use of packets, through the principle of [layering](#).

To comprehend the way this works, one must have a basic understanding of the layered nature of modern networking architecture (as I discuss in some detail in [the chapter that talks about the OSI Reference Model](#)). Even though packets may be used at lower layers for the mechanics of sending data, a higher-layer protocol can create logical connections through the use of messages sent in those packets.



**Key Concept:** Circuit-switched networking technologies are inherently connection-oriented, but not all connection-oriented technologies use circuit switching. Logical connection-oriented protocols can in fact be implemented on top of packet switching networks to provide higher-layer services to applications that require connections.

### ***Connection-Oriented and Connectionless Protocols in TCP/IP***

Looking again at TCP/IP, it has two main protocols that operate at the [transport layer of the OSI Reference Model](#). One is the [Transmission Control Protocol \(TCP\)](#), which is connection-oriented; the other, the [User Datagram Protocol \(UDP\)](#), is connectionless. TCP is used for applications that require the establishment of connections (as well as TCP’s other service features), such as FTP; it works using a set of rules, as described earlier, by which a logical

connection is negotiated prior to sending data. UDP is used by other applications that don't need connections or other features, but do need the faster performance that UDP can offer by not needing to make such connections before sending data.

Some people consider this to be like a “simulation” of circuit-switching at higher network layers; this is perhaps a bit of a dubious analogy. Even though a TCP connection can be used to send data back and forth between devices, all that data is indeed still being sent as packets; there is no real circuit between the devices. This means that TCP must deal with all the potential pitfalls of packet-switched communication, such as the potential for data loss or receipt of data pieces in the incorrect order. Certainly, the existence of connection-oriented protocols like TCP doesn't obviate the need for circuit switching technologies, though you will get some arguments about that one too. ☺

The principle of layering also means that there are other ways that connection-oriented and connectionless protocols can be combined at different levels of an internetwork. Just as a connection-oriented protocol can be implemented over an inherently connectionless protocol, the reverse is also true: a connectionless protocol can be implemented over a connection-oriented protocol at a lower level. In a preceding example, I talked about Telnet (which requires a connection) running over IP (which is connectionless). In turn, IP can run over a connection-oriented protocol like ATM.