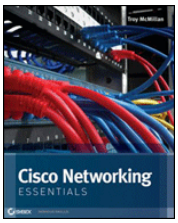

Chapter 1 - Networks
Cisco Networking Essentials
by Troy McMillan

Previous Next

Defining Network Architectures

The *architecture* (or structure) of a network can be discussed from both a physical and a logical viewpoint. For example, in the previous section you looked at how distance can be used to differentiate networks into architectures called LANs and WANs. The architecture of a network can also describe the rules and processes used on the network. The security relationships that exist among the computers on the network can define different architectures. In this section, the difference between peer-to-peer and client-server architectures is explored.

Understanding Peer-to-Peer Networks

Peer-to-peer networks were the first type of networks to appear. This type of network is often referred to as a workgroup. In a peer-to-peer network, each computer is in charge of its own security, and the computers have no security relationship with one another. This does *not* mean that the users on the computers cannot share resources; otherwise, it wouldn't be a network!

There are certain shortcomings to this paradigm. In a workgroup, a user can access resources on another computer only if that user has an account on the computer where the resource resides. Moreover, depending on how the sharing is set up, she may also have to identify herself and provide a password to access the resource.

The ramifications of this can be illustrated with an example. Suppose you have four computers in an office that are used by four different users. If your goal is to allow all users to access resources located on all four computers, you would have to create an account for each person on all four computers. That means you would be creating 16 accounts in all (4 computers × 4 people). That's a lot of work! (I guess it's a form of job security!)

Figure 1.7 illustrates this situation. Each computer is named after its user, and as you can see, all users must have an account on all computers. Also note each user can be given different levels of access. Note that the passwords that a user has been assigned on any two computers have no relationship to each other. A user can have the same password on all computers, or a different password on each computer, with no effect on functionality because they are not related to each other in any way in a peer-to-peer network.

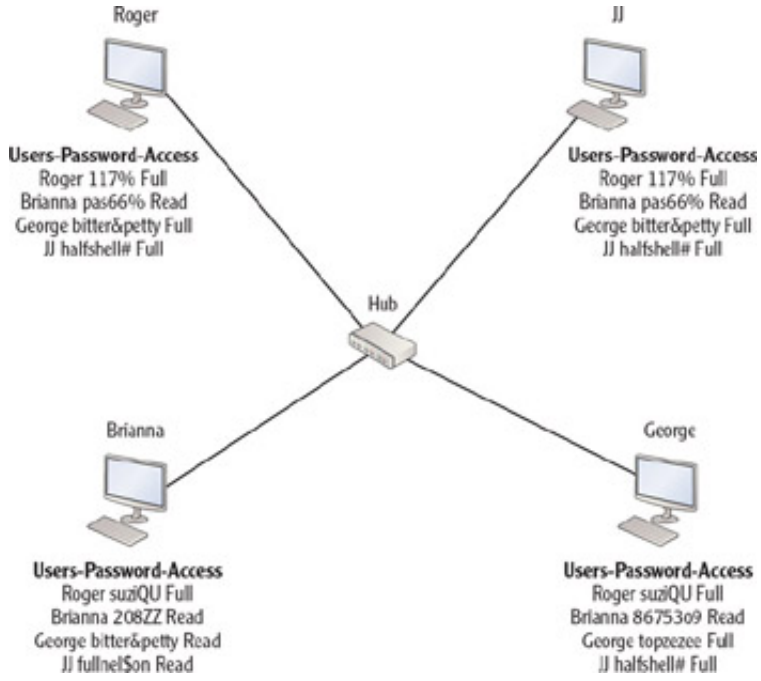

Figure 1.7: Peer-to-peer architecture

Another challenge with workgroups is that after the number of computers gets beyond 10, two problems occur. First, because of the nature of the communication process that occurs in a workgroup, traffic overwhelms the physical infrastructure, and the network gets very slow. This occurs because in order to locate each other, the computers must broadcast for one another. A broadcast is akin to a person calling out in a crowded room, "Who is Joe?" Then, when Joe answers, you send him the data. In Figure 1.8, workstation 10 is seeking to connect to a computer named Bannarama, so a broadcast is sent out to every computer. Then Bannarama answers with its IP address.

Note An *IP address* is a number in a specific format that is used to identify a computer. This topic is covered in detail in Chapter 7, "Classful IP Addressing."

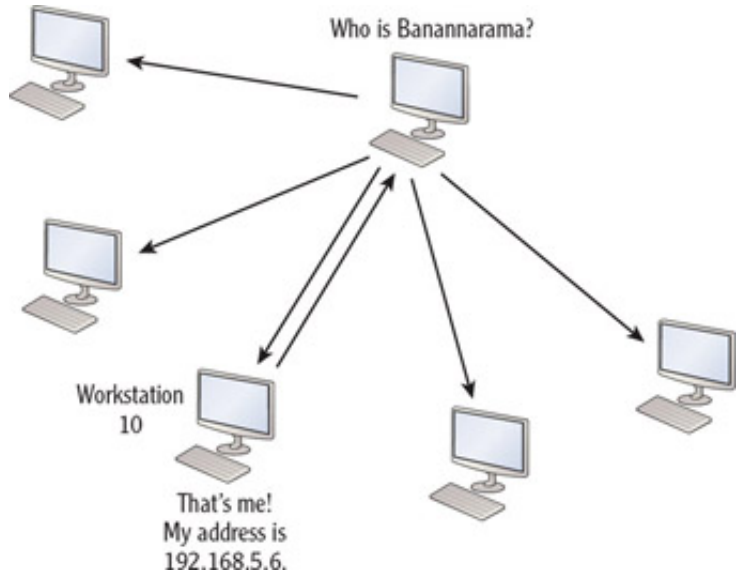

Figure 1.8: Broadcasting

Moreover, unlike humans, the computers can remember who is who for only a minute or so, and then they must broadcast again.

The second problem that occurs when more than 10 computers are present in a peer-to-peer network has to do with the design of client operating systems. Most client operating systems (meaning any operating system that is not a server operating system) can host only 10 concurrent connections from other computers at a time. So if a popular file is located on a computer in a workgroup, and 10 computers are already connected, the 11th computer won't be able to access the resource until a computer disconnects!

Workgroups still have their place and their advantages. One is low cost when compared with a client-server network. Obviously, no servers (which cost more than client computers) need to be purchased. Workgroups are also quite simple to set up when compared with client-server networks. Home networks are usually peer-to-peer, and many small office and home office (SOHO) networks function well as workgroups.

However, in medium to large networks, the management of security becomes an administrative nightmare. As discussed earlier, each user must have an account on every computer that he will use or access over the network. Also, peer-to-peer networks are not scalable. When a network can be grown (with respect to the number of computers) without causing additional network traffic or additional administrative effort, it is said to be scalable.

In summary, the advantages of a peer-to-peer network are as follows:

- Low cost
- Easy to set up
- No server required

The disadvantages of a peer-to-peer network are as follows:

- No centralized control of security
- Administrative burden of maintaining accounts on all computers
- Not scalable

Understanding Client-Server Networks

The most obvious difference between a client-server network and a peer-to-peer network is the presence of at least one server. This brings up an issue that needs to be addressed before you encounter it. There are two explanations of a *client-server network* that are commonly used. Both are applicable, so let's cover both.

First, a client-server network can be explained in terms of resource access. When viewed from this perspective, it means that the shared data is centralized on a device called a file server.

What's the Difference between a Client and a Server, Anyway?

Which computer is the client and which is the server is simply a matter of perspective. If the computer is seeking to access a resource on another computer, it is acting as a *client*. If it possesses a resource that another computer accesses, it is acting as a *server*. Consequently, computers in a peer-to-peer network will be acting as either at various times, depending on whether they are accessing a resource or allowing access to a resource.

A *file server* is a computer that contains resources (files) that users in the network need. A server's operating system is designed differently than one that will be used on client computers. It is not bound by a limit to the number of connections. Hundreds of computers can connect. The advantage is that the security surrounding the resources can be centralized on that server.

Using our example from Figure 1.7, if there was a file server in that network, we would not have to create an account for every user on all computers. We would have to do that only one time, on the server where the resources are located.

The other explanation of a client-server network takes this a step further. These networks are sometimes called *domain-based networks*. In this case, the server is a special type of server called a directory server or domain controller.

Note A directory server or domain controller maintains the location of all resources in the network (including the computers themselves) and the locations of each. The computers in the network use this server to find things. Instead of broadcasting to find resources, the computers check with the directory server, which results in a great reduction of traffic!

The domain controller creates a group security association between the computers that are members of what is commonly called a *domain* (or a *realm* in Unix). After a user is made a member of the domain, the user will have

two types of user accounts: a local account on her computer, as she had in the peer-to-peer network, and a domain account. The domain account will be created on the domain controller where it will be stored.

This domain account will allow the user to log into the domain from any computer that is a member of the domain. This simplifies the account creation process in the same way illustrated in the explanation of using a file server. The accounts are created one time on the domain controller, and then the account will work on any computer in the domain.

The domain controller, rather than the individual computers, is responsible for validating the credentials of users. Whenever a user logs into the domain from a member computer, the login request is sent to the domain controller, which verifies the name and password and then sends the user an access token. An *access token* is a file that lists the resources that the user is allowed to access in the network, regardless of where the resource is located.

The benefit of this security paradigm is a feature called *single sign-on*. After logging into the domain, a user will not be prompted for a password again, even when accessing resources. It doesn't even matter which computer the resource is on!

On other hand, there are disadvantages to implementing a client-server network. The hardware and software required to deploy servers is significantly more expensive than client software found in a peer-to-peer network. Configuring and maintaining these servers also requires a much higher degree of skill.

Moreover, when a single domain controller is in use, a single point of failure has been introduced to the operation of the network. If something happens to the domain controller, such as a hardware failure, all access to resources can be interrupted. For these reasons, most networks deploy multiple domain controllers to eliminate this single point of failure, further adding to the cost of deploying a client-server network.

In summary, these are the advantages of a client-server network:

- Centralized administration
- Single sign-on
- Reduced broadcast traffic
- Scalability

  Note *Scalability* means that the network can grow without the congestion problems that arise when a peer-to-peer network grows larger.

Disadvantages of a client-server network are as follows:

- Higher cost for server software and hardware
- More challenging technically to implement
- Single point of failure with a single domain controller or single file server

Figure 1.9 compares the peer-to-peer and client-server networks.

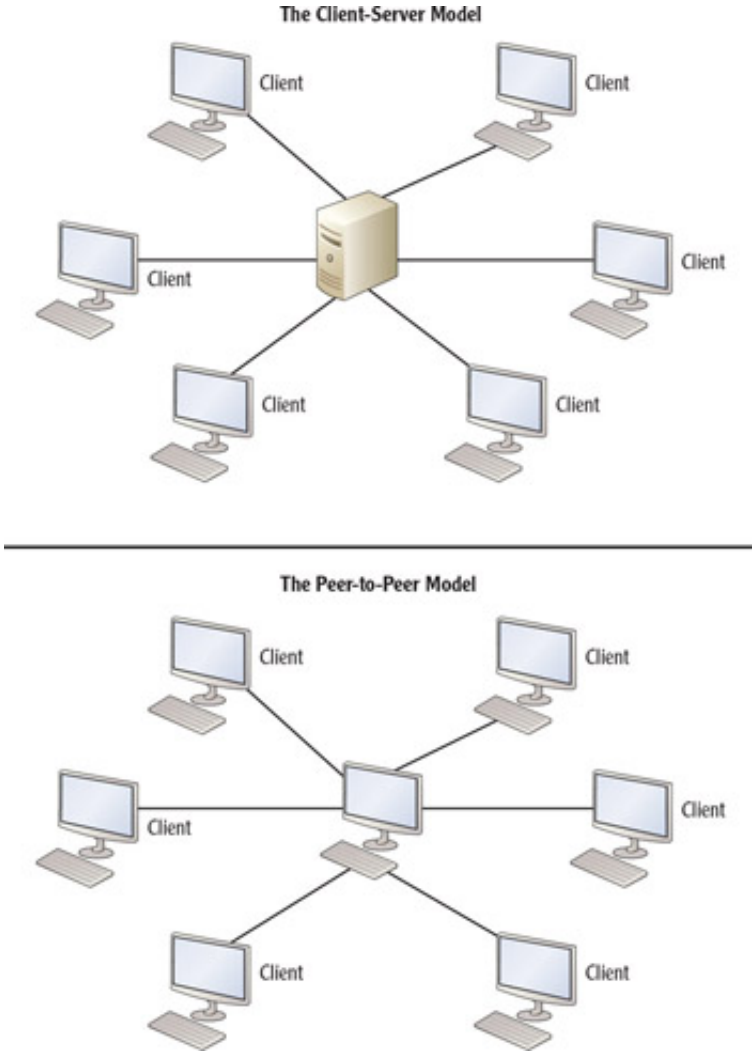

Figure 1.9: The client-server model (top) and the peer-to-peer model (bottom)

Previous Next