Chapter 1

# Thin Clients and Digital Independence

**Solutions in this chapter:**

- The Mainframe Model Meets Distributed Computing
- The Difference Between Remote Node and Remote Control
- The Evolution of the Thin Client
- Gaining Digital Independence with Thin Clients
- Achieving Bottom-Line Value with Thin Clients

☑ Summary

☑ Solutions Fast Track

☑ Frequently Asked Questions

# Introduction

Since computers were introduced, businesses have used them to simplify tasks so that their employees could be more productive and the businesses could realize cost reductions. Through acquisitions, mergers, and distributed IT administration, and sometimes merely because a mission-critical application is available on only a certain platform, businesses have accumulated different types of computer hardware, operating systems (OSs), and applications. In any business, it is not unusual to find users with multiple computing devices, whether a shared computer used for a specific application or a personal digital assistant (PDA) in every user's pocket.

The major challenge of a network with different types of computers is being able to deliver a mission-critical application to all users. This challenge is further exacerbated by the Internet because a business cannot predict the types of OS Internet users have or whether the business's application will work for them. In addition, Internet connections are notoriously inconsistent in regard to bandwidth availability. Applications delivered across the Internet must consume a fraction of the bandwidth that is available in corporate networks. In order to meet these needs, an application must be digitally independent from the OS and the infrastructure.

The thin client, such as that provided by Citrix MetaFrame XP, is the answer to these challenges. Since MetaFrame XP can deliver applications interactively via an Internet browser, it is virtually independent of a user's OS and hardware. Citrix MetaFrame's thin clients are termed "thin" because of the low bandwidth they consume. This, too, meets the need to provide applications independent of the digital infrastructure.

# The Mainframe Model Meets Distributed Computing

It might seem a bit odd to begin a book about Citrix MetaFrame XP, a breakthrough product that can distribute an application to computers that don't natively support it, with a discussion about the history of the mainframe. You might wonder exactly how the mainframe and distributed computing models have anything to do with Citrix in the first place. The fact is that understanding both models is essential to being able to manage and troubleshoot Citrix MetaFrame XP in the long run. That's entirely due to the fact that Citrix MetaFrame XP and its previous versions are built on a model that mixes the mainframe model into the distributed computing model in order to gain the benefits of both.

Some of the advantages that you will see arising from this mixed model are:

- Running applications with transparent high performance over slow, unreliable connections

- Delivering applications to Internet users from within a browser
- Providing users with an application that is incompatible with their OSs without requiring them to change to a different machine or unfamiliar OS

# A History of the Mainframe

A person who accessed a 100-megabyte (MB) hard drive weighing two and a quarter tons and costing more than $130,000 in the 1960s experiences a keen sense of wonder when tucking one of today's mobile multigigabyte (GB) drives, purchased for less than $1000 and weighing less than a pound, into a pocket. Computing during the 1960s and earlier was highlighted by exceptionally sizeable components. In fact, early computing was completely centralized, with dumb terminals surrounding an all-powerful mainframe. The earliest of these mainframes were large buildings filled with vacuum-tubed mammoths. Today's mainframes are contained in small boxes, some diminutive enough to fit under a desk. Regardless of the size or shape that these computers take, they serve an identical function: they act as a centralized brain used to compute multiple applications used by multiple, simultaneous end users through dumb terminal interfaces.

The original computer was the Harvard Mark I. Completed in 1943 with the collaboration of IBM engineers, the Harvard Mark I was so named because it occupied an entire Harvard University building. The Mark I was basically a huge calculator that had the ability to handle arithmetic functions and subroutines that could perform logarithms and trigonometry. Originally, IBM Selectric typewriters were used for Mark I input and output, but these were later replaced by IBM's punch-card technology. The Mark I was not the only computer at work in the early 1940s. In England, the Colossus Mark I was constructed and pressed into service during World War II to crack the German Enigma code.

On the heels of the Mark computers, the United States Army constructed the Electronic Numeral Integrator and Calculator (ENIAC) in 1946. Whereas the Harvard Mark I was a monstrously large calculator, ENIAC was an even larger computer capable of performing 100,000 calculations per second.

From the Harvard Mark I until nearly 1960, all computers used hundreds and thousands of vacuum tubes to execute calculations. It wasn't until 1958 that the first fully transistorized mainframe, the IBM 7000, was created by IBM. Considered the start of computing's Second Generation, this computer was first installed by the U.S. Air Force to provide an air traffic control system. In 1960, IBM introduced the first Basic Input/Output System (BIOS) in its 700 series of mainframes. This innovation enabled multiple processes to occur simultaneously. Another development was the ability to communicate with the mainframe over a telephone line.

Microchips, generally considered to constitute the Third Generation of computing, were the next innovation for the mainframe, arising out of the need for smaller computers. Until the arrival of the microchip, only governments and the largest businesses could afford computers due to their size and associated cost. The microchip replaced transistors, even being called a "flat transistor." Using microchips, new, smaller computers, called *minicomputers*, were created. Minicomputers, though smaller than mainframes, provided much the same features and functions as mainframes, and just as important, they used the same computing model as the mainframe. Today, minicomputers such as the IBM AS-400 are popular among small and midsize companies.

Computing's Fourth Generation was introduced in 1971 with the Intel 4004, the world's first microprocessor. Combining processing capability on a single chip with random access memory (RAM), the personal computer (PC) was born. Today's PCs, while far more powerful, with stunning graphic capabilities and vastly different applications from the original PCs, are still considered the fourth generation of computing.

Mainframes today offer a number of benefits. With the ability to run symmetric multiprocessing on more than a dozen reduced instruction set computer (RISC) processors, mainframes offer robust computing capabilities. Hardware advances have produced a highly fault-tolerant system for mainframes. Clusters, or groups of mainframes that act as a single environment, can keep mission-critical mainframe applications available even in the event of one mainframe's failure. Given the fact that a mainframe is a single point of failure for everyone using it, fault tolerance is of critical importance.

## The Mainframe Model

Today's mainframes can support multiple simultaneous programs for multiple simultaneous users. The mainframe itself uses peripheral channels for input/output (I/O) processes in order to free up processing power. Because so many concurrent processes take place on a mainframe, it usually has an extremely fast bus speed and error-correcting hardware. The mainframe is considerably more powerful than minicomputers, even though the same computing model is used, and mainframes can support more simultaneous programs and users.

The mainframe model is centralized computing. The only processing power needed is that of the mainframe itself. The devices people use to access the programs on the mainframe are traditionally dumb terminals. Terminals are ASCII character-based devices. Figure 1.1 displays a typical mainframe model.

Also referred to as *green screens* due to their green text on a black background, these terminals are called "dumb" because they need no processing power whatsoever. Today newer "smart" terminals provide a built-in screen display instruction set. However, this does not change the location of processing within the model. Nowadays, most people use terminal emulation programs running on their networked PCs to access a mainframe.

This does not change the model in which all processing takes place on the mainframe. In this scenario, the client adds no processing power nor intelligence to the mainframe's applications. Figure 1.2 illustrates the use of clients with a mainframe and terminal server.

**Figure 1.1** A Typical Mainframe Model

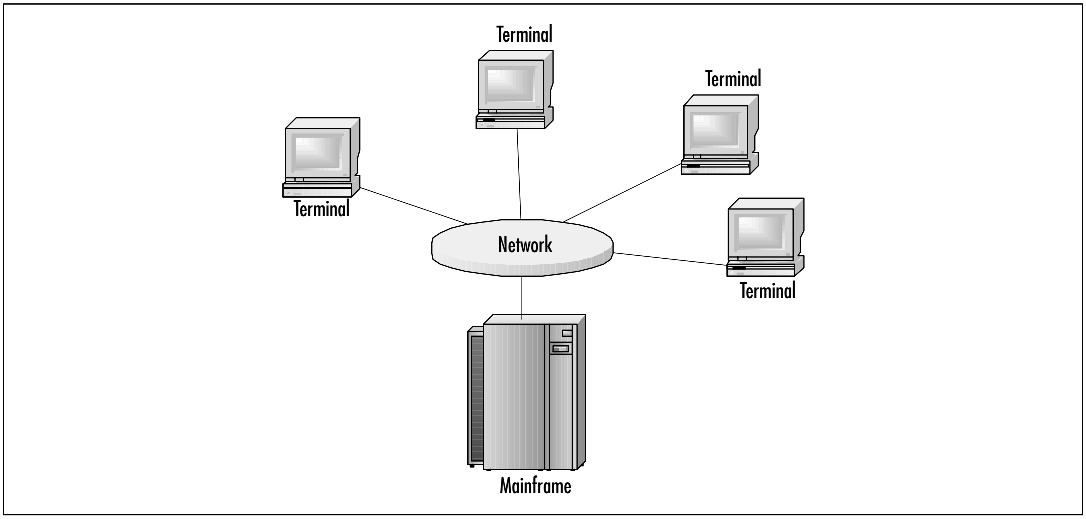
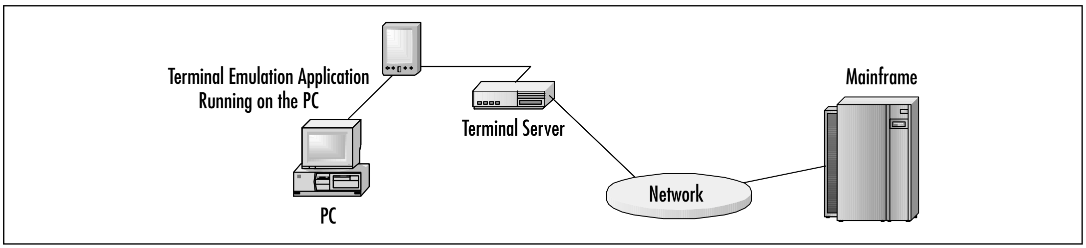



**Figure 1.2** Terminal Emulation on PCs




Centralized computing is organized into a single tier of processing. The application logic, located on the mainframe, is processed solely on that same machine. As a monolithic architecture for both applications and administration, the mainframe model provides many benefits and is used by organizations worldwide.

# Advantages of the Mainframe Model

The mainframe model essentially uses the mainframe as a central repository for information as well as processing for every application. Terminals enable input/output into the applications. Nearly all administration takes place on the mainframe itself. This scheme provides multiple benefits:

- Hardware maintenance cost reductions
- Single point of administration
- One type of administrative skill set
- Simple architecture and low bandwidth requirements

Terminals and printers are the two most common components of the mainframe model. Terminals are relatively cheap. Since you can easily add or replace terminals without causing a major impact to the budget, and because they have minimal components, terminals carry very little maintenance or overhead administrative costs related to end-user hardware. Even though the mainframe itself has a high price tag, the ongoing maintenance costs are very reasonable.

A single point of administration is greatly valued in IT administration. Although it is true that a person could administer the entire mainframe from any terminal connected to the network, the single point of administration means that there is only one thing to administer—that is, the mainframe. The mainframe applications and data are all located in a single place. This makes it far easier to troubleshoot, change passwords, add users, change access rights, and manage applications than is the case in a distributed computing environment.

When a network is based solely on a mainframe and terminals, only a single administrative skill set is required. The only thing any administrator needs to know is how to run the mainframe. Programmers find that programming skills for mainframe programming languages are all they require. By contrast, a distributed computing model necessitates many different types of skill sets. The distributed computing model demands administrative knowledge of the network topology, client OSs, network OSs, protocols, and applications that are processed partially on the client and partially on the server. The complexity of distributed computing can drive up administrative costs merely in finding people with all the various skills or training them.

Designing a mainframe architecture is fairly straightforward. Because of the centralized nature of the model, the optimal placement of the mainframe is in a central location. Whereas originally terminals were hardwired to asynchronous communication ports on the mainframe itself, nowadays terminals are connected to a terminal server and can access multiple network resources. Due to the processing taking place only on the mainframe, the only information that travels between the mainframe and the terminals is keystrokes and display. This results in very low bandwidth demand on the network. Saturating a network with text and display traffic is difficult in today's high-capacity networks, which releases the network designer from a great deal of pressure.

## Disadvantages of Centralized Computing

The mainframe itself has a major drawback. Mainframes are very expensive. That purchase price tag, which can run from several hundred thousand to millions of dollars, is followed by maintenance contract costs for both the mainframe hardware and mainframe applications. These costs are tolerable for companies that balance them against the benefits of office automation. The minicomputer, with a much lower price—in the tens of thousands to low hundreds of thousands of dollars—offers similar benefits and is often the choice for a small and midsize businesses.

A centralized computing model includes its own unwanted baggage, regardless of whether the central computer is a mainframe or minicomputer:

- **Single point of failure** The largest challenge with centralized computing is overcoming the fact that a mainframe is a single point of failure. Since most mainframes house a business's most critically important applications and are used by nearly every person in the company, mainframe failure can have a devastating impact on corporate productivity. Even though clustering can reduce the impact somewhat, a failure in connectivity between the mainframe cluster and terminals will render the same impact.
- **Character-based applications** Users who use dumb terminals often bemoan the fact that the mainframe applications are textual. Graphics applications on a mainframe are rare at best. Character-based applications are usually the only types available for mainframe users.
- **Potential bottlenecks due to time-sharing systems** Because all users access a mainframe simultaneously, there is a potential for bottlenecks. In times of heavy usage, the mainframe might not have the processing capabilities to respond to every user's session in a timely manner.

# A History of Distributed Computing

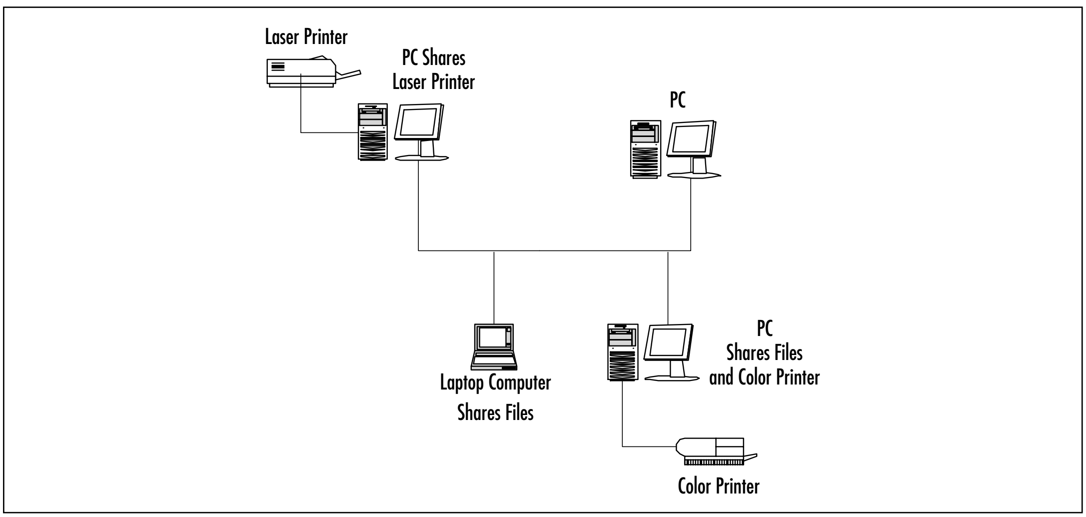
The development of the microchip and smaller computers, along with the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol stack, soon brought about networks with distributed computing capabilities. The keys to an effective network were interoperability and interconnectivity. The innovation of networking achieved communication between computers using dissimilar OSs. Networking also gave rise to client/server computing.

At first, connecting multiple computers to share files was beneficial. Then, with hard drives costing hundreds of dollars per megabyte, it became very desirable to share hard drive space. Organizations were soon able to save additional money by sharing printers through networking. It didn't take long for people to look for ways to harness and share the server processing power. This motivation developed into a *client/server architecture* model.

Xerox Corporation's Palo Alto Research Center (PARC) was the first to develop an Ethernet network coupled with Xerox Distributed File Server (XDFS) in order to duplicate the capabilities of a mainframe using smart workstations. The result was a tremendous decrease in cost to automate an organization. Apple Corporation borrowed these technologies and built them into its computer line. Graphical user interfaces (GUIs) added the ability for color pictures to be displayed. The mouse brought the now-familiar point-and-click methodology so that users can more easily manipulate graphics-based data. UNIX workstations, too, grew to incorporate graphical capabilities. Since then, Microsoft has come to dominate the personal computer market with its popular Windows OS.

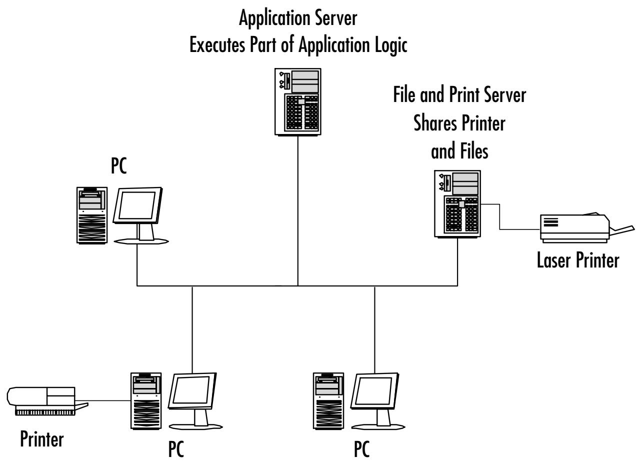
Some centralization of duties became increasingly necessary. UNIX offered native server computing capabilities on every workstation. That is, any workstation could share resources, hold its own lists of users and passwords, and at the same time connect to other UNIX workstations and access those shared resources. Managing users on multiple machines added a tremendous amount of administrative overhead. This type of *peer-to-peer* networking, in which each PC acts as either a server or a client, is displayed in Figure 1.3.

**Figure 1.3** The Peer-to-Peer Computing Model




Novell's NetWare server OS moved in a more centralized direction. With a centralized server OS, it became much easier to administer users, passwords, printers, files, and other resources. This provided organizations with powerful applications, centralized management, and the low cost of PC computing. This client/server computing model is shown in Figure 1.4.

**Figure 1.4** The Client/Server Computing Model

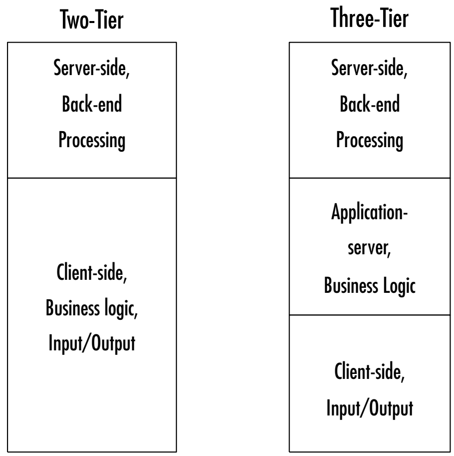



Even though the server offers a more centralized administrative model, it does not match the mainframe computing model. Applications do not run solely on a server and get delivered to a dumb terminal. Instead, depending on the application's individual architecture, application logic can be processed on the server, the client, or both. Because every computer in a PC network is capable of performing its own processing, applications vary based on the programmer's design. Whereas some applications use the server as merely the location from which files are shared, others divide their processing across both clients and servers.

When the application logic is processed on both the client and the server, it is considered a *two-tiered* architecture. In two-tiered architecture, the server runs a software application and processes information that does not need to be interactive. The client also runs a software application with logic that can access the server, perform edits, and do some business logic processing. This is a typical scenario for database computing, where the database is housed on a server and the server performs record searches and record generation, and where the application on the client accesses the database and performs edits.

Some database applications stretch this type of computing to a *third tier*. In this scenario, a database server houses the database, an application server performs business logic, and a client application performs input/output. Both of these models are shown in Figure 1.5.

**Figure 1.5** Client/Server Architecture in Two- and Three-Tiered Structures




## Advantages of Distributed Computing

As many businesses have discovered, the distributed computing model carries several advantages:

- Reduced hardware costs
- No single point of failure
- Flexibility
- Scalable architecture

Mainframe hardware is extremely pricey. Servers, by comparison, are far less expensive. Even though individual PCs are much more expensive than terminals, the overall initial cost of implementing a distributed computing network is relatively small.

A truly distributed computing network has many servers running in multiple locations. If a single server fails, people are able to continue working using the resources of other servers. Even when connectivity to another location fails, users are able to work with local resources shared by local servers. When information is replicated across several servers, a server going offline does not prevent information access. This ensures that people can continue working. Furthermore, fault tolerance of redundant links further reduces the possibility of catastrophic failures.

In a distributed computing environment, a business is not dedicated to a single OS. Servers can run network OSs that include Microsoft Windows 2000, various flavors of UNIX, Novell NetWare, and others. At the same time, the business can select any number of OSs to run on the desktop. This provides for a very flexible environment.

With a distributed computing environment, a person can run an application remotely over phone lines, even those available on airplanes. Leveraging this flexibility, a company can truly maximize employee efficiency.

The very nature of distributed computing is to be scalable. Whenever growth of the business demands increased computing power, new servers can be added to the network. Server hardware manufacturers offer clustering, symmetric multiprocessing, and fault-tolerant components to scale up an individual server as well.

## Disadvantages of Distributed Computing

The distributed computing model bears some undesirable aspects for organizations, which include:

- Heightened administrative costs
- Increased security risks
- Lack of centralized backup

The distributed computing model is rife with administrative costs. Because every server and client on the network is intelligent and has its own OS, each one requires administrative support for its hardware and OS. Though standardization is a best practice, it is not unusual to find multiple types of desktop and server hardware, multiple operating systems, even multiple versions of the same operating systems on a network. Applications add further complexity to the network, also requiring administration, upgrades, changes, and configuration. Even though you are avoiding the price of a mainframe, you must include the costs of the desktop devices to the distributed computing model's bottom line; they are at least five times more expensive than dumb terminals. When it comes down to brass tacks, if you use a distributed computing network as opposed to using a mainframe with dumb terminals, it simply costs more at the desktop, costs less at the server, and takes more people to manage the same number of end users.

Security risks increase with every server on the network. Although it isn't written in any manual, a network OS in its out-of-the-box configuration is not secure. The installer must configure security to match the policies of the business and seal off any server vulnerabilities. To top it off, the server administrator must apply any patches for newly discovered vulnerabilities of the server or any of its applications. Unfortunately, this process takes considerable vigilance, and even so, security holes do open up. Saboteurs have been known to exploit the weakness of one server in order to gain access to other, mission-critical servers. The physical security of a distributed computing network is also more susceptible to breaches because of the lack of centralized location for servers. Not every facility provides a locked, climate-controlled room in which to store its servers. Quite often, smaller offices place a server beneath a desk or in a closet.

A clear-cut set of security policies and procedures is required in order to conquer this shortcoming.

Backing up the data on a server is a crucial activity toward maintaining business continuity. In the event of a mainframe failure, the latest backup can be retrieved and restored, and business can continue. However, in the distributed computing model, many of the servers are located anywhere across the globe. Connectivity between those locations can be extremely slow and unreliable. It simply is not feasible to conduct a backup of data over such a link. An administrator must rely on people in each location to execute and manage backup procedures.

## The MetaFrame XP Thin Client Model

It's now time to explain why we needed to review the mainframe and distributed computing models. You see, Citrix MetaFrame XP is built on a combination of these two models in such a way that it can maximize the benefits and minimize the disadvantages of both. Sound interesting? Let's look at the *thin client model*.

The thin client model uses some of the same concepts as the mainframe model. Rather than requiring the access devices to provide any intelligence, the thin client model treats them as terminals. However, these are not green screen terminals or terminal emulation programs used by the mainframe, since they must support graphical applications. Instead, these are Independent Computing Architecture (ICA) terminals and ICA client applications that enable emulation of a session running on a Citrix MetaFrame XP server.

**NOTE**

ICA terminals are also called *thin client terminals* and *Windows terminals*. If you are running Windows 2000 Server with Terminal Services but not using Citrix MetaFrame XP or previous versions, you will be running Remote Desktop Protocol (RDP) and will require RDP-compatible terminals and RDP client applications.

In the mainframe model, organizations are 100 percent dependent on a centralized computing power yet can easily administer it from a single seat. Because all the applications run on the mainframe, it is fairly easy to administer them with one place to go for installations, configuration changes, and rights access. The thin client model offers the same capabilities inasmuch as you are considering only the Citrix MetaFrame XP servers. It is likely that most organizations will combine a distributed computing model with the thin client model in order to leverage the benefits of both. It is also likely that some of those distributed computing servers will participate in the same domains as

Citrix MetaFrame XP servers and share in the user rights and privileges administration. It is also highly probable that end users will access the network from fully functional PCs with ICA client applications rather than using ICA terminals. So, when we say that the thin client model offers a centralized administration approach to applications, we refer to a network that implements *only* Citrix MetaFrame XP: When a network consists solely of a Citrix MetaFrame XP thin client model, the applications will be installed, configured, and administered only on the Citrix servers. No support is required at the deskside, since all sessions can be shadowed. (We discuss this hip little feature later in the book.)

So far in our discussion, the thin client model sounds a lot like a mainframe model. However, it shares some of the characteristics of distributed computing as well. The first and most obvious characteristic is that there is no mainframe in the thin client model. It uses servers—exactly the same types of servers as the ones being used for file and print services, application services, databases, and Web services.

Originally created to support remote users needing access to colorful Windows applications, the thin client model delivers a fully graphical experience to the user's desktop. Rather than inundating the user with fuzzy green characters on a black screen, an ICA client might appear no different from running Windows on the desktop.

Another important trait shared with distributed computing is the three-tiered architecture of application logic. The difference here is that most applications that Citrix MetaFrame XP can deliver in three tiers were not originally developed to be used that way. The thin client model actually extends two-tiered applications into previously unexplored territory. For administrators, this is an adventure just begun. Table 1.1 lists several of the characteristics of the various computing models.

**Table 1.1** Computing Model Characteristics

| Characteristic | Mainframe | Distributed Computing | Thin Client |
|---|---|---|---|
| Text-based mainframe or minicomputer | Yes | No | No |
| Terminals | Yes | No | Yes |
| Centralized administration of applications | Yes | No | Yes |
| Centralized installation of applications | Yes | No | Yes |
| Reduced costs for administrative overhead | Yes | No | Yes |
| Low bandwidth utilization | Yes | No | Yes |

Continued

**Table 1.1** Continued

| Characteristic | Mainframe | Distributed Computing | Thin Client |
|---|---|---|---|
| Low-cost servers (relative to mainframe cost) | No | Yes | Yes |
| Graphical capabilities | No | Yes | Yes |
| Three-tiered architecture | No | Yes | Yes |
| 32-bit applications delivered to any type of operating system or ICA terminal | No | No | Yes |
| Offline computing capabilities | No | Yes | No |

# Combined Advantages of the Two Models

As mentioned previously, the thin client model can maximize the advantages of both the mainframe model and the distributed computing model while minimizing both models' disadvantages. Looking at this statement in a little more depth, consider the advantages of the mainframe model:

- Hardware maintenance cost reductions
- Single point of administration
- One type of administrative skill set
- Simple architecture and low bandwidth requirements

The thin client model has many of the same advantages. Hardware maintenance cost reductions are easily realized in the thin client model. Unlike the mainframe model, the thin client model offers two ways in which it can reduce hardware maintenance costs. First, if a business decides to implement the thin client model, it can do so with ICA terminals that cost about a quarter of the price of a standard desktop PC. From that point forward, that business will not have much in the way of hardware maintenance costs due to the low maintenance requirements of terminal hardware.

The second way that a business can realize reduced hardware maintenance costs is unique to the thin client model: An organization can take an existing distributed computing network and migrate it to thin clients. To the end user, the thin client experience is no different than running a 32-bit Windows operating system on the desktop. But the hardware required to support an ICA client does not require Pentium processors, a large amount of RAM, or gigabytes of storage. Instead, an ancient 386 PC with minimal RAM (about 4MB) and storage (megabytes rather than gigabytes) can run an ICA client and appear to be a much more powerful machine running Windows 2000 locally. Not only does this extend the life of the machine, but because there is very little processing required of it, the machine is not overtaxed. Users can log on to any machine in the entire network and receive their own individual desktops, without worrying about leaving data on a local hard drive. This fact removes the need to move PCs around the network or reconfigure them individually. Therefore, a company can use thin clients to extend the life of their hardware, greatly decreasing their maintenance costs.

The thin client model provides a centralized approach to administration. Like a mainframe, all applications are located in a single place—on the thin client server. Applications need only be installed, configured, or deleted from a single location. Most, if not all of the rights, privileges, and administrative changes can take place in one server or through the domain management tools. The reason that this model does not necessarily use a 100-percent centralized administration is entirely related to the design and complexity of the environment. If it is a centralized design, more centralized benefits will be achieved.

It's a stretch to state that the thin client model requires only one type of administrative skill set. In a pure thin client model, using only Citrix MetaFrame XP servers and ICA terminal hardware, this statement is true. However, in most businesses that implement Citrix MetaFrame XP, the thin client model is mixed with the distributed computing model and can consist of multiple types of network and desktop operating systems and access devices, resulting in the need for multiple types of administrative skill sets.

Simple architecture, arguably, is not a characteristic of the thin client model. In fact, it can become quite complex to integrate a thin client model into a distributed computing model. The benefit in which the thin client model does succeed is low bandwidth requirements. An ICA client requires a minimum of a 14.4K modem link in order to function reasonably. Since most remote users connect via 28K or 56K modems, a thin client model performs extremely well.

The distributed computing model has other benefits that the thin client model can maximize:

- Reduced hardware costs
- No single point of failure
- Flexibility
- Scalable architecture

Earlier in the chapter, we mentioned that the distributed computing model avoided the huge price tag of hardware implementation that is suffered by the mainframe model. In the thin client model, this is also true. The thin client model uses network servers and does not require excessive investments in back-end hardware.

The mainframe model's biggest disadvantage is that it constructs a single point of failure for mission-critical applications. This is not true for distributed computing or the thin client model, which shares that ability to distribute thin client servers across a network and avoid a single point of failure. The benefits of distributed thin client servers are best realized using Citrix MetaFrame XP, due to its ability to distribute the load across a number of similar servers, which is discussed in Chapter 10.

Flexibility in computing enables companies to maximize employee effectiveness. The distributed computing model enjoys flexibility to a degree. The thin client model can take that flexibility to the next level. For example, one of the main benefits for businesses with a growing virtual workforce is to offer remote computing. Sometimes computing must be executed offline. At other times, it must be carefully controlled and run only online. A thin client model blended with a distributed computing model offers the optimal environment for these businesses. Online computing, of course, can be carefully controlled and centrally managed on a thin client server, whereas offline computing can take place using the device's operating system and local applications. It is up to the network administrator to decide which applications must be managed and which can be freely distributed. Furthermore, wireless public networks are adding new methods for remote computing. Because wireless networks have, up to this point, fairly low bandwidth capabilities, using a "thick" client application across one is next to impossible—that is, it's impossible without using a thin client.

An architecture that is scalable is yet another advantage shared by the thin client and distributed computing models. With the hardware available for clustering, storage area networks (SANs), symmetric multiprocessing (SMP), and redundant components, each server can scale up in size. The entire network can be scaled out through the addition of new servers. Furthermore, with Citrix MetaFrame XP, the servers can be load balanced so that sessions are spread evenly throughout the Citrix MetaFrame XP server farm.

**NOTE**

When you begin designing your thin client servers, you can throw out the rules that you follow for file and print servers. Because thin client servers have a much more intense usage of processors and memory, their design requires more of them. In addition, you have the options of scaling horizontally, by creating an application server farm, or scaling vertically, by beefing up a single server, to accommodate more simultaneous users. Make certain that your choices will meet your needs for user experience, power, performance, and availability of sessions.

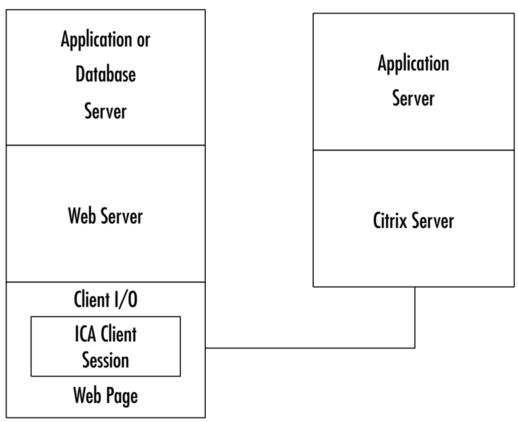
# New Thin Client Model Advantages

The popularity of today's personal computers has driven use of the distributed computing model. Even in businesses in which the mainframe model was fully implemented, distributed computing was adopted and meshed into the fabric of the network so that benefits of both could be realized. Doing so, however, also maximized the disadvantages of both models, since effectively, both models were still in use simultaneously. It is only with the thin client model that businesses can minimize the disadvantages.

One issue will continue to plague the thin client model: It relies heavily on administrative knowledge! The network administrator who walks in from a distributed computing or mainframe model background can easily become perplexed by the variety of options for designing, implementing, and managing a thin client network.

One of the benefits of the thin client model can become somewhat baffling to figure out. This is the ability to leverage the Internet to deploy applications. What you can do with Citrix MetaFrame XP is launch the ICA client from within a Web page and run an application. Doing so logically combines two different three-tiered architectures into the system, as shown in Figure 1.6.

**Figure 1.6** Logical Processes in Launching an ICA Client from a Web Browser




# Overcoming the Disadvantages of Previous Versions of MetaFrame

Citrix MetaFrame XP is the latest version of the thin client solution offered by Citrix Systems, Inc. Each newer version introduced to market has surmounted many of the challenges found in earlier versions, a trend evidenced by the many improvements incorporated into Citrix MetaFrame XP.

Designing & Planning…

## How the Internet Figures into Client/Server Computing

The Internet is the world's largest network. Originally developed by the U.S. Department of Defense's (DoD) Advanced Research Projects Agency (ARPA) for the purpose of sharing radar data, the Internet, then called ARPANET, was turned on for the first time in 1971, with 19 hosts. After growing as a government and educational internetwork for over 20 years, the Internet was finally opened up as a global internetwork for commerce in the early 1990s.

Anyone with device and communication line access can access the Internet, surf Web pages, send e-mail, participate in chats and instant messages—even host their own Web sites for others to access. Businesses have begun to reevaluate and reprogram many back-end applications to fit the Internet, leveraging its global capabilities in order to reduce costs and execute business-to-business processes.

The challenge faced by many organizations is in overcoming low bandwidth. Even though cable modems, Integrated Services Digital Network (ISDN), and Digital Subscriber Line (DSL) lines can reach megabit-bandwidth rates, most people rely on telephone lines for Internet access. With typically 28K to 56K of bandwidth, these methods of downloading graphics and applications result in long waits that are increasingly intolerable to end users.

Citrix MetaFrame XP offers the ability to use thin client computing across the Internet in a variety of ways. By reorganizing applications to execute through a Citrix server, the barrier of low bandwidth is removed.

One of the more interesting changes is Citrix's recognition that people used Citrix MetaFrame's previous versions in different environments for different purposes. The product was flexible and had many features, but some were not necessary for certain types of implementations. As a result, Citrix MetaFrame XP has three different flavors: XPs, XPa, and XPe, scaling up from the small business implementation to a server farm management system. These versions are described in more detail later in this chapter.

Although prior versions could be installed on either a Windows NT or Windows 2000 server, a common theme for Citrix MetaFrame implementation was to deploy applications either over a virtual private network (VPN) or via a Web page. The latest Citrix MetaFrame XP is optimized to run on Windows 2000 and to deploy applications across the Internet. This is an improvement in performance for Windows 2000 servers as well as for businesses extending applications across the Web.

Taking the server farm concept to the next level, scalability of Citrix MetaFrame XP is unparalleled. Previous versions, although beginning to adopt the server farm concept, do not have the same manageability. In fact, Citrix MetaFrame XP removes the boundaries of geography from the server farm. An administrator can manage a Citrix MetaFrame XP server farm from anywhere around the globe.

One of the missions of the Citrix MetaFrame product series has been to deliver "any application to any device over any connection." Citrix MetaFrame XP provides the back-end delivery of these applications, while ICA clients work on PCs, Apple Macintoshes, UNIX, Linux, ICA terminals, and the latest in wireless devices. The infrastructure can vary from an Internetwork Packet Exchange/Sequenced Packet Exchange (IPX/SPX) network with native Novell NetWare servers to public wireless networks connected through the Internet to VPN solutions installed on corporate networks.

# The Difference Between Remote Node and Remote Control

One of the very first reasons Citrix created its first WinView product was to help companies reduce the cost of providing remote control services to end users. WinView was a predecessor to Citrix MetaFrame XP based on OS/2. It allowed a user to dial in to the WinView server and use a DOS or Windows 3.1 version session running applications through remote control. By implementing WinView, a company could replace 15 dedicated remote control PCs with a single one. How times change!

Remote computing has grown up from two distinctly different methods:

- **Remote node** This is a form of computing that connects a PC or even a server to the network across a modem line. Often called *remote access*, remote nodes have grown to include VPN so that PCs can connect to a network across the Internet.
- **Remote control** This is the form of computing in which a remote PC takes control of the desktop of a PC that exists on the network. Originally, this type of computing was able to share a single session on a single network PC, and users had to dial up directly to that PC.

The unique thing about Citrix MetaFrame XP is that these two types of remote computing are combined and extended. Not only does it merge them, but a Citrix MetaFrame XP server can be implemented in conjunction with other vendors' solutions for remote node computing. In this section, we'll look at both types of remote computing and how they apply to Citrix MetaFrame XP.

# Remote Node Computing

Remote node computing traditionally is handled via dialup modems. When the computer dials the network, the user is prompted to logon to the network, and from that point on the computer acts exactly the same as if it were locally connected to the network, barring any security limitations imposed by the remote access system. The user can usually send e-mail, transfer files, run network applications and communicate through instant messaging. The only difference is that, due to the low bandwidth capabilities usually demonstrated by dialup modems, network performance is typically very slow. For example, on the local area network (LAN), downloading a 3MB file can appear nearly instantaneous, whereas across a remote node connection through a 56Kbps modem, it can take more than 10 minutes to pull down the file.

Remote access today can be established through any of the following types of connectivity:

- Plain old telephone service (POTS)
- ISDN
- X.25
- VPN through the Internet using modems, wireless networks, cable, or DSL

The original method of remote access was performed using a dialup modem to connect to POTS, also known as the *public switched telephone network (PSTN),* which then connected to a remote access server's node. In doing so, the remote access client became transparently connected to the network. The remote-client-to-remote-server connection is referred to as *a point-to-point connection*, since only two nodes, or *points,* share a single connection. Figure 1.7 shows remote access via POTS.

ISDN offers a faster connectivity method compared to POTS. Developed as an upgrade to POTS, ISDN offers digital connectivity, typically at speeds of 128Kbps. The increased speed makes it very attractive as a remote node method. The point-to-point connection used is shown in Figure 1.7, but the modems are replaced by ISDN modems.

Remote access connectivity through X.25 is rarely used anymore. Even though it is an international standard for data transmission, X.25 is a legacy packet-based network that was used for wide area networks (WANs) and remote access in the early days of networking. Today, few X.25 networks are left; they have been replaced by networks with much faster connectivity capabilities.

VPNs are connected through the Internet using modems, wireless networks, cable, or DSL. A VPN enables the remote client to use the Internet to create a virtual point-to-point connection to a remote access server. Because the Internet is a public network, the virtual point-to-point link must be protected. This is done by *tunneling* the data within IP packets so that it cannot be decrypted by eavesdroppers. A typical VPN connection is shown in Figure 1.8.

**Figure 1.7** A Remote Node Connected over POTS

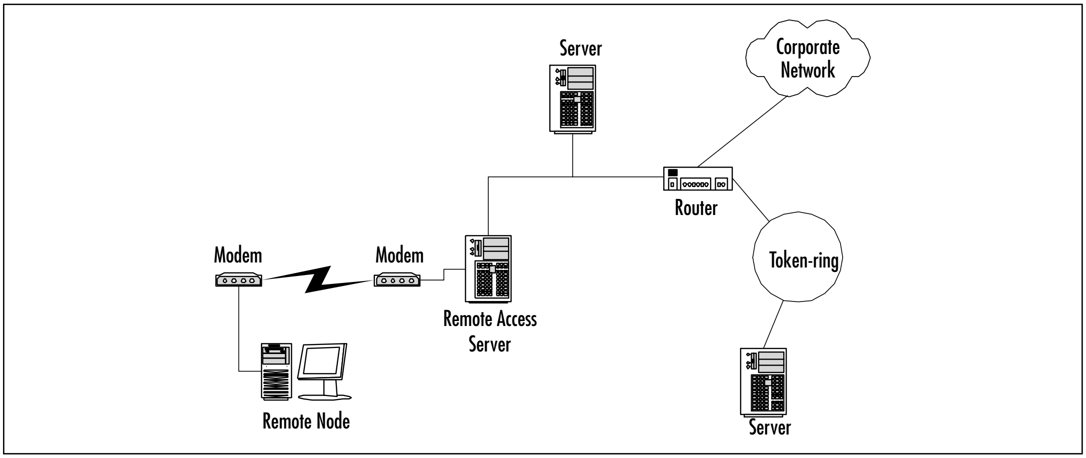
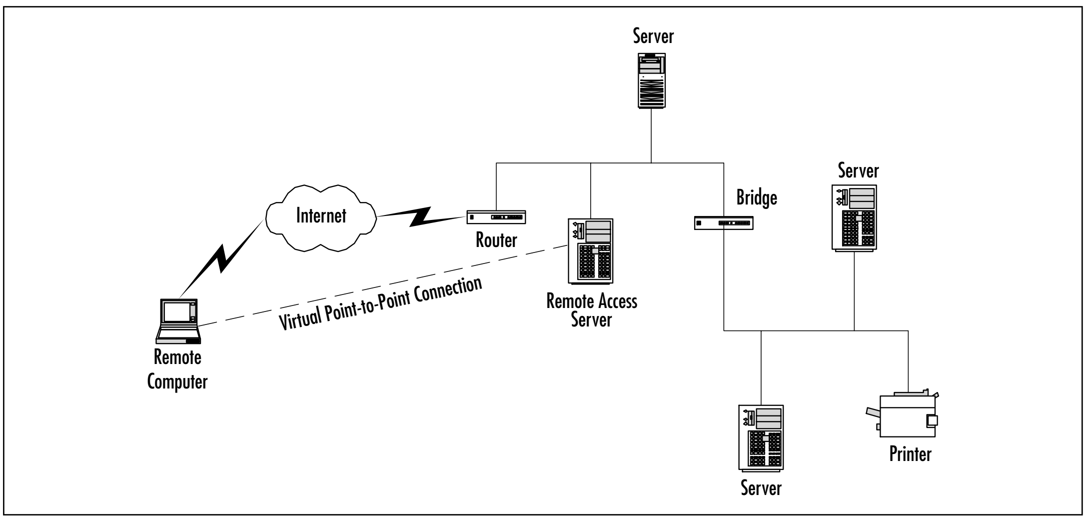



**Figure 1.8** A Virtual Private Network Connection




Remote access solutions offer many business benefits. It is rare to find a place where a phone line is not available. Being able to connect users to the local network from any location offers businesses the ability to extend their networks anywhere. Besides removing geographic limitations, telecommuting extends work hours to 24 hours a day, seven days a week. Remote node solutions have many advantages technically as well:

- Simple to configure
- Options for accessibility
- Security services to secure data
- Easy-to-manage hardware requirements

Remote node computing is fairly simple to configure. At the corporate network, the administrator installs a remote access server. The server is configured with either a set of modems or a link to the Internet, or both. The client needs remote access client software installed and configured to connect to the remote access server. Most desktop operating systems, such as Windows XP, include native remote access client software that is compatible with standard remote access servers.

A key benefit is to be able to connect over different types of media using a variety of networking protocols. As mentioned earlier, a remote node connection can be established across a phone line, an ISDN line, a legacy X.25 network, a cable network using a cable modem—in fact, across any Internet connection or dialup method. Remote node computing uses a set of standard protocols, such as Serial Line Interface Protocol (SLIP), Point to Point Protocol (PPP), and Asynchronous NetBEUI (AsyBEUI). These protocols can support multiple types of protocol stacks. For example, a PPP connection can transmit both TCP/IP and IPX/SPX traffic. For these reasons, remote access servers can be installed to work on any type of network.

Any aperture in the fabric of a network can expose sensitive information with potentially devastating consequences to a company. Since remote node services extend the network to any location, it is vitally important that the remote access servers and connections are secured from intruders and eavesdroppers. Many standard security protocols have been developed to help secure remote access. To ensure authentication, tunneling, and encryption, a network administrator can install Internet Protocol Security (IPSec), Challenge Authentication Protocol (CHAP), or Layer 2 Tunneling Protocol (L2TP), among others.

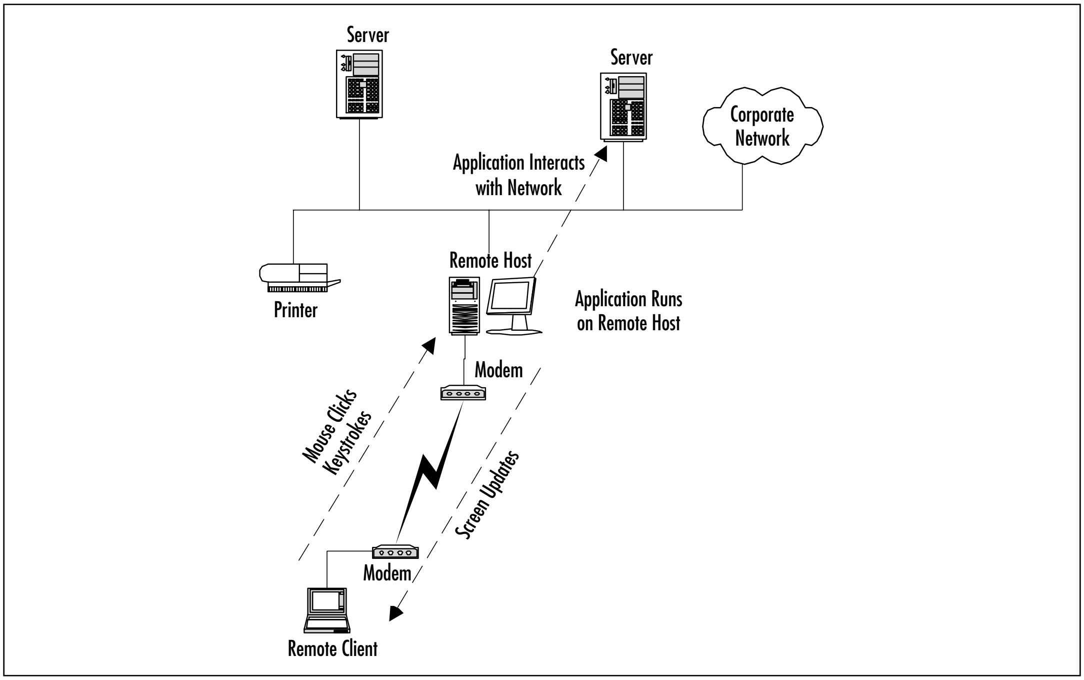
The hardware and management requirements for remote access computing are much easier to meet than those for remote control computing. To provide remote control computing, an administrator must place computers on the network to host remote control sessions in addition to providing computers as remote control clients. By contrast, remote node computing requires only the remote clients and one or a few remote access servers. Managing remote control hosts can be cumbersome compared with managing remote access servers.

## Remote Control Computing

Remote control solutions enable a user to control a session running on a remote host. The keystrokes that the user types from the remote client are transmitted across the wire to the remote host, along with any mouse clicks. The graphics on the remote host's screen are transmitted back to the remote client. The applications are actually processed on the remote host. Most types of remote control solutions enable some form of file transfer between the remote host and the remote client. Unlike the rest of the remote session, file transfers are processed locally on the remote client.

One of the surprising benefits of remote control solutions is the ability for two people to view the same screen of information, even though those people are in two different locations. Some enterprising administrators used early remote control solutions to manage mission-critical workstations and servers on the network. They found that they could administer these servers from anywhere around the network, reducing the need to go directly to the servers to execute standard administrative tasks. Network management solutions today typically incorporate a remote control application for network workstations. Administrators use remote control sessions in response to help requests while the end user asking for help is still using the remote control host. This way, the administrator can diagnose and fix problems as well as demonstrate to the user how to carry out some function. Figure 1.9 exhibits how the remote control session functions.

**Figure 1.9** A Remote Control Session




One of the main advantages to remote control software is the ability it offers support personnel to diagnose and troubleshoot remote systems without being physically at the system's location. Whether used to collect information for trending, license usage, or monitoring, this capability reduces support costs in terms of travel, time, and effort. It also improves support response times and results in higher end-user satisfaction.

Remote control software is useful for demonstrations, even as a teaching tool. A teacher can work on a remote control host while a student watches from a remote control client session. With either the teacher or the student able to control the session, the student can perform specific tasks under supervision.

For government and highly regulated industries, there are often rules regarding which PCs can store certain types of information. A remote control solution enables a company to maintain its security restrictions yet still enable key personnel to work with the data from their own workstations. Security for such a remote control solution should be configurable to prevent downloading of data from the remote control host to the remote control client.

There are some limitations to remote control. Like remote node, the bandwidth for dialup connections to a remote control host is very low. Unlike remote node, the type of data that transmits across the wire consists of graphics rather than file transfers. The higher the resolution of those graphics and the larger the number of colors used, the more bandwidth is consumed. Applications that heavily utilize graphics suffer degraded performance when executed via remote control.

A rather wasteful aspect of remote control is how many computers it uses. Since each remote control host supports a single session, it must wait, unused, until a remote client logs on. Then, during the session, both the remote control host and the remote control client are dedicated to a single session. When remote control software programs were introduced to the market, some businesses set up rooms with hundreds of PCs shelved and attached to modems simply to support telecommuting. Just one word describes this setup: *expensive*.

Both remote control and remote node have pros and cons. Citrix MetaFrame XP has the ability to combine the two types of remote computing in order to maximize the benefits and minimize the drawbacks. Table 1.2 demonstrates the characteristics of each type of remote computing.

**Table 1.2** Remote Computing Characteristics

| Characteristics | Remote Control | Remote Node | Citrix MetaFrame XP |
|---|---|---|---|
| Updates graphics on the remote client, receives keystrokes and mouse clicks | Yes | No | Yes |
| Many users can connect to a single server | No | Yes | Yes |
| Clients with older technology can run newer applications | Yes | No | Yes |

Continued

**Table 1.2** Continued

| Characteristics | Remote Control | Remote Node | Citrix MetaFrame XP |
|---|---|---|---|
| Graphics-intensive applications hamper performance | Yes | No | Yes |
| Security options for encryption and authentication | Sometimes | Yes | Yes |
| Data can be accessible yet prevented from being copied to remote client | Yes | No | Yes |
| Full desktop is presented in a window to client | Yes | No | Yes |
| Applications can be launched transparently and appear to be locally launched | No | No | Yes |
| Can be used over any dialup or Internet connection | Yes | Yes | Yes |

# The Evolution of the Thin Client

The thin client model evolved from an idea developed originally in Citrix's WinView product in which multiple remote control sessions could run on a single server. Citrix was able to see the benefits and limitations of the mainframe and distributed computing models as well as remote control and remote node computing. Since these types of computing are merged into a single thin client model, their benefits increase.

Today's thin client model uses Citrix MetaFrame XP running on Windows 2000 Terminal Services. A remote client uses the ICA client software to connect to a server over any type of dialup or Internet link. Once connected, the remote client can open a virtual desktop or launch an individual application transparently. All application processing takes place on the Citrix MetaFrame XP server, with only screen updates being displayed on the remote client. With the thin client model, Windows terminal hardware can be used to connect to the thin client server.

In reviewing the thin client model, there are two key terms to understand:

**Thin client** A thin client is the session that takes place between the server and its client. This session consists solely of the screen updates, keyboard strokes, and mouse clicks transferred between the two machines. The client is considered "thin" because it consumes very little bandwidth with its compressed traffic. Windows terminals are now available for use with thin client servers. These thin client machines are similar to mainframe terminals, except for heightened graphics capabilities, occasional support for Web browsers, and some memory to increase performance. With such capabilities, they are considered "intelligent," even though they do not have the ability to process any applications locally.

**Fat client** A fat client is the session that takes place between a server and its client that transmits actual data between the two. A fat client is considered "fat" because of the relatively large amount of data transmitted and high consumption of bandwidth. A typical database application uses a fat client when client and server are on separate machines.

## The History of Citrix MetaFrame

Citrix grew out of a concept developed by Ed Iacobucci, who headed the OS/2 program for IBM and Microsoft. Iacobucci conceived of a system whereby computers that were not designed to run OS/2 could start doing so. In 1989, Iacobucci formed Citrix Systems, Inc. (the name is derived from *citrus*, since the company is based in Florida) to take this concept forward. The name of the technology? MultiWin.

### Product Evolution: From Citrix WinView to WinFrame to MetaFrame

MultiWin enables multiple users to simultaneously run multiple user sessions on a computer. In the process, the sessions share the system resources, including network interfaces, processor, memory, input/output, and other resources. After creating WinView on the OS/2 platform, the company soon realized that OS/2's days were numbered. In 1991, Citrix licensed the source code for Windows NT from Microsoft and began developing MultiWin on the Windows NT platform. Microsoft also invested in Citrix, buying 6 percent of the company. In 1995, Citrix unveiled its WinFrame product, the first one to combine MultiWin technology, the ICA protocol, and the Windows NT operating system.

The WinFrame product was extremely successful. Microsoft enjoyed an explosion in sales of Windows NT at the same time. Because of this success, Microsoft believed that it did not need Citrix's help to sell Windows NT and in February 1997 notified Citrix that it intended to develop its own multiuser technology. Citrix stockowners will remember the date: Citrix stock plummeted more than 60 percent in a single day. The public was not sure that Citrix would survive. Citrix and Microsoft, however, negotiated over the next few months until they struck a deal.

Microsoft had one objective: to become a player in the thin client market. Citrix had its own objective: to continue to be in the thin client market. The deal provided both. Microsoft licensed the MultiWin technology from Citrix. Citrix retained the right to continue development on the WinFrame product. Citrix also retained the ICA protocol and the ability to develop an add-on product for the Microsoft Windows NT Terminal Services products. The deal specified that Microsoft could develop its own thin clients, but only for Microsoft Windows operating systems. Citrix was free to develop ICA clients for any type of operating system. Citrix's stock revived immediately based on this news.

Under the new agreement, Microsoft quickly adapted MultiWin in Windows NT 4.0, Terminal Server Edition, with its own client using Remote Desktop Protocol (RDP). Although the surface of the Terminal Server Edition looked the same as the standard version, the inner workings were much different. The two required separate service packs, hot fixes, and special drivers. Microsoft asked vendors to certify its printer drivers to work with Terminal Server Edition.

Citrix quickly released the MetaFrame 1.0 for Windows NT 4.0 Terminal Server Edition. Later, it improved the product and released MetaFrame 1.8. These products offered several advantages over the "plain vanilla" Windows NT 4.0 with Terminal Services:

- Ability to deliver 32-bit applications to platforms other than Windows
- Low bandwidth consumption using ICA protocol
- Multiprotocol capabilities (RDP runs only over TCP/IP)
- Ability to launch the ICA client and applications from within a browser
- Ability to launch higher-resolution graphics yet retain low bandwidth consumption

Rather than release a separate edition of Windows 2000 for customers who wanted to run thin clients, Microsoft delivered Windows 2000 Server, Windows 2000 Advanced Server, and Windows 2000 DataCenter Server with the native ability to run Terminal Services. Microsoft had received a great deal of feedback about the ease of managing a server using the remote control features. Therefore, the company included a native license for two sessions to be running on each Windows 2000 server. When Terminal Services are activated, they can be activated for the two sessions in remote administration mode, or Terminal Services can be activated for multiple clients, each requiring a thin client license.

Citrix released MetaFrame 1.8 for Windows 2000 servers. Today, Citrix offers MetaFrame XP for Windows 2000 servers. Citrix continues to develop ICA clients that work with any version of MetaFrame. These clients can attach anything from a Linux box to an Epoch mobile device to a Citrix MetaFrame server and then launch a 32-bit Windows application.

## MetaFrame XPs, XPa, and XPe

Citrix MetaFrame XP comes in three versions. Like Windows 2000, which scales from Windows 2000 Server to Windows 2000 Advanced Server to Windows 2000 DataCenter Server, the three versions of Citrix MetaFrame XP scale up in capabilities as well. The three versions are:

- Citrix MetaFrame XPs
- Citrix MetaFrame XPa
- Citrix MetaFrame XPe

Citrix MetaFrame XPs is the base version of the three. Created for a small deployment of thin client, it provides the application services across LANs, WANs, the Internet, intranets, and extranets. MetaFrame XPs offers many features. It uses the Citrix Management Console and Program Neighborhood. It has centralized printer management and license management. Not only does it support both Active Directory and Novell Directory Services, it can encrypt session data using the Secure Socket Layer (SSL). It also includes the NFuse version 1.6, described in the next section of this chapter.

Citrix MetaFrame XPa was designed with features that meet the needs of small and medium-size businesses. It contains all the features of XPs, but XPa also includes Advanced Load Management. Moving toward the server farm paradigm, Advanced Load Management enables the XPa version to dynamically manage the number of sessions assigned to any one of a group of XPa servers.

Citrix MetaFrame XPe builds on the features of XPa and adds management capabilities that are well suited for large organizations. In deploying a server farm, the XPe version should be selected due to its features. On top of all the other features, XPe includes the ability to package and deliver applications as well as system monitoring and analysis.

## NFuse

NFuse 1.6 is best described as a portal product for publishing applications to the Web. The solution consists of three components:

- Citrix MetaFrame server
- Web server
- ICA client with a Web browser

Using NFuse, a Citrix MetaFrame XP administrator can publish any interactive 32-bit Windows applications or UNIX applications (when using Citrix MetaFrame 1.1 for UNIX) to the Internet so that they can be launched from within a standard browser.

With a little effort and organization, a portal with personalized information can be created for users so that they have Web-based access to all their tools and applications, without needing to install them locally. There is no need to reengineer a business application to fit the Web. Using NFuse, the Web can be leveraged with existing applications, reducing the time, effort, and costs involved in such a project.

### Citrix MetaFrame 1.1 for UNIX

UNIX applications are used throughout businesses today. Since these applications can only be used on UNIX machines, in the past many organizations provided multiple machines per user so that the user could access both UNIX and Windows applications. With Citrix MetaFrame XP, a person who uses UNIX daily can also receive a Windows session on the same machine. However, with the prevalence of Windows on the desktop and the higher likelihood of finding users who are familiar with Windows, there is a need to deliver UNIX applications to people with non-UNIX platforms. This is the goal of Citrix MetaFrame 1.1 for UNIX. Besides offering remote access to applications, as do many other terminal emulation applications, Citrix MetaFrame for UNIX offers a high-performance graphical session using the same ICA client that can access Citrix MetaFrame XP servers.

This product can provide UNIX and Java applications to any ICA client, whether Linux, UNIX, Epoch, Windows 32-bit, or DOS, over the same types of connections that Citrix MetaFrame XP can use. Improving the performance of graphical UNIX applications delivered across low-bandwidth connections and able to implement higher security, Citrix MetaFrame 1.1 for UNIX works with Sun Solaris, AIX, and HP-UX versions.

## Windows 2000 Terminal Services

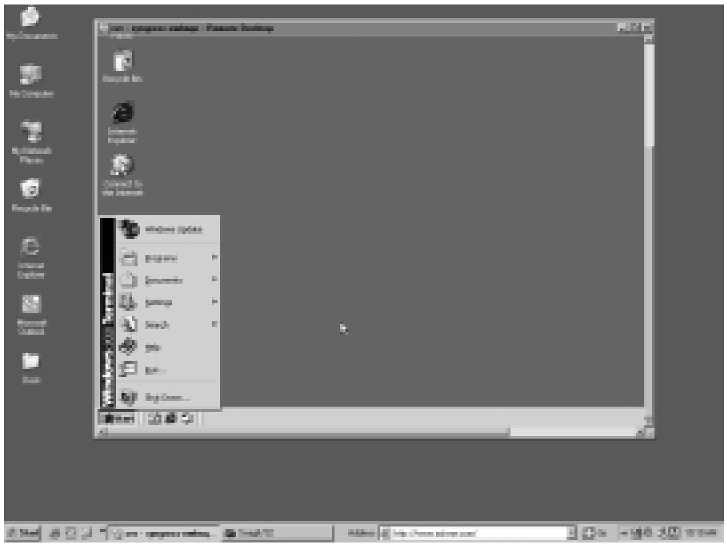
Windows 2000 with Terminal Services is the Microsoft thin client server that uses the MultiWin technology combined with Microsoft's RDP thin client. Windows 2000 Terminal Services uses RDP version 5.0. Previously, Windows NT 4.0, Terminal Server Edition, used RDP version 4.0.

With Windows 2000 Terminal Services, a business has a complete multiuser thin client system. Users who connect to the server receive a traditional Windows 2000 desktop view. The client is presented in a window on the desktop. Figure 1.10 displays the standard Terminal Services desktop.

### Special Mode for Installing Applications

Even though Windows 2000 Server has the Terminal Services capability built right into the code, you don't have to install it. You can choose to have a standard Windows 2000 file and print server. You can also choose to install Terminal Services in remote administration mode or in applications services mode. As stated earlier, remote administration mode enables two sessions for RDP clients to connect to the server. When Terminal Services is installed in applications services mode, many more clients can run sessions connected simultaneously.

**Figure 1.10** The Terminal Services Client Desktop



When Terminal Services have been installed, multiple interactive sessions can run on the server at the same time. Win32API calls have been reprogrammed so that they allow traditional Windows applications to use configuration files and registry information in a multiuser model. Some applications, such as Microsoft's Office suite, must be installed with a different template in order to function in a multiuser system. Other applications must be installed using *install mode*. There are three ways to put the server in install mode:

- You can use the Add/Remove icon in the Control Panel to invoke install mode.
- If an application's installation starts and you select the option to install the program for all users rather than just yourself, the server is automatically placed in install mode.
- You can also open a command prompt window and type **Change User /Install** to bring up install mode.

The normal mode for Terminal Services is *execute mode*. Whenever you are completing an installation, you should open a command prompt window and type **Change User /Execute**. More about installing applications appears in Chapter 9.

## Overview of Terminal Services Licensing

Starting a new Windows 2000 server with Terminal Services, thin client services are immediately available. However, you must install them in order to use them. Furthermore, if you install the application services mode (as opposed to remote administration mode), you have 90 days to activate a license server to manage Terminal Services client licenses.

Terminal Services client licenses are different from standard client access licenses (CALs) for Windows 2000 Server. A standard CAL is required to access a file and print server. If you set up a Windows 2000 server with Terminal Services and never share files or printers from it, none of the users needs a standard CAL. They will need Terminal Services CALs (TS CALs), which are available as follows:

- You do not need to purchase a TS CAL for the most recent Microsoft Windows operating systems. Each Windows 2000 Professional, Windows NT 4.0 Workstation, or Windows XP machine has a built-in TS CAL.
- You may purchase from Microsoft a TS CAL for any other operating system installed on a network computer.
- You may purchase a Work-at-Home TS CAL from Microsoft. This is a discounted CAL that combines the standard CAL and the Terminal Services CAL and is intended for telecommuters.
- You may purchase an Internet Connection License. This is a special type of license for allowing up to 200 anonymous users to connect concurrently to Terminal Services over the Internet. This license is intended for users who are not employees of the company.

The Terminal Services licensing server is an option that can be installed on any Windows 2000 domain controller in Active Directory or on a Windows 2000 server in a Windows NT 4.0 domain. The licensing server tracks the available licenses and distributes them to client devices. When the licensing server is installed, it needs to receive a digital certificate from the Microsoft Clearinghouse in order to be activated. After activation, you can install licenses. You do not need a Terminal Services licensing server in order to run servers in remote administration mode. You can find more about Terminal Services licensing in Chapter 6.

## Overview of Terminal Services Consoles

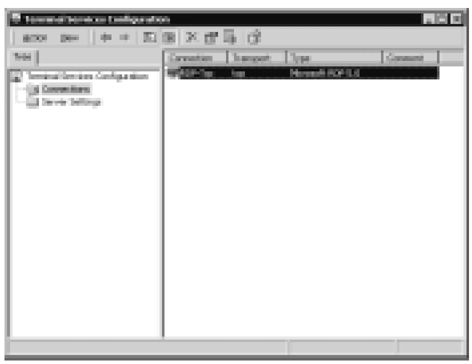
The native Windows 2000 management consoles will become a Terminal Services administrator's home turf for managing thin client connectivity. Other tool sets are available to help manage a Terminal Server from the Windows 2000 Resource Kit.

More in-depth discussions about these consoles and how to use them can be found throughout this book. All in all, you should become familiar with several different consoles, each handling a unique administrative task:

- Terminal Services Connection Configuration
- Active Directory Users and Computers
- Terminal Services Manager
- Terminal Services Licensing Manager
- Terminal Services Client Creator
- Terminal Services User Manager for Domains
- Application Security Registration

The Terminal Services Connection Configuration console, shown in Figure 1.11, is used for creating sessions and configuring the variables within them. You will use this console to establish the protocols that should be used by a session as well as the security for each session, including access and encryption for each individual connection.

**Figure 1.11** The Terminal Services Connection Configuration Console



Each connection has multiple options for configuration. By double-clicking a connection name, the administrator can edit security options, remote control settings, the network adapter the connection is available to, and more. The properties for an RDP connection are shown in Figure 1.12.

The Active Directory Users and Computers Console is the primary method of managing users. In this console, an administrator can create new users, change user information, and delete users from the Active Directory domain. This is primarily a network administration tool for user information, it does include Terminal Services configuration options for users, as shown in Figure 1.12. The new Terminal Services information for users is distributed among four tabs:

**Figure 1.12** Properties for a Connection

- **Terminal Services Profile** This tab allows the administrator to create a specific profile for thin client sessions. This profile can differ from a file-and-print profile. It is useful for administrators to be able to separate the thin client profile from standard profiles, especially when an administrator does not want to overwrite a telecommuter's or anonymous Internet user's home profile.
- **Sessions** This tab offers specific configuration information for a user's session. This information can customize how a session acts for a particular user, such as increasing or decreasing the disconnect time for a session rather than inheriting the standard session configuration.
- **Environment** This tab configures the way that a client device is handled when it is initially connected to the server.
- **Remote Control** This tab sets the security options for remote control sessions.

The Terminal Services Manager is the most interactive and useful of the administrative consoles. Using the Terminal Services Manager, you can view processes that are currently active on all Windows 2000 Terminal Services servers within the domain. The Terminal Services Manager is the optimal tool for monitoring and troubleshooting the sessions run by thin client users. Shown in Figure 1.13, this is the console that the administrator uses to view the processes users execute in their sessions.

A Terminal Services administrator is given only 90 days to activate a licensing server and add licenses for the sessions used on the server. During the 90-day period, the licensing server provides temporary TS CALs to machines that connect. These licenses are tracked. After the 90 days have completed for each temporary TS CAL, that machine can no longer access the server. However, once the licensing server has been activated and TS CALs have been installed, the machines connect to the server, and the temporary TS CAL is automatically upgraded to a standard TS CAL. To add licenses, you use the Terminal Services Licensing Manager. Figure 1.14 displays this console.

**Figure 1.13** Terminal Services Manager

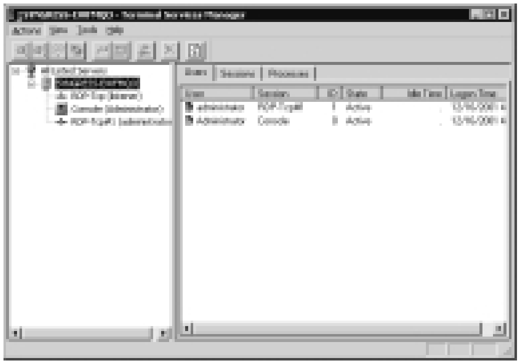
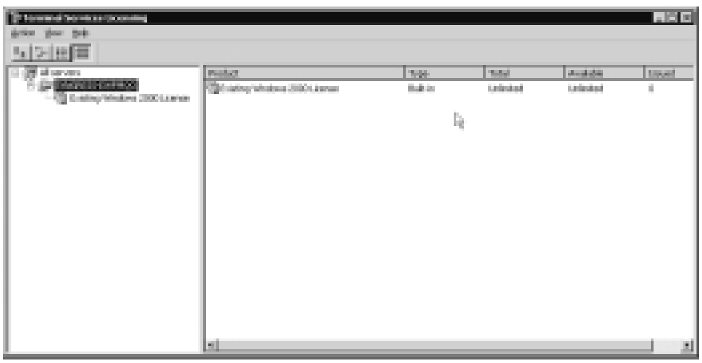


**Figure 1.14** Terminal Services Licensing Manager



The client software for the RDP client can be generated from Windows 2000 Server with Terminal Services. This is performed through the Terminal Services Client Creator. Pictured in Figure 1.15, this console offers the ability to generate installation disks for either 16-bit Windows or 32-bit Windows RDP clients.

**Figure 1.15** Terminal Services Client Creator

Although it is not installed as a default tool, you could find the Terminal Services User Manager for Domains an essential tool if you are using a Windows 2000 Terminal Services server within a Windows NT 4.0 domain. The standard User Manager for Domains in a Windows NT domain does not include the configuration options for users that you need for specific configurations. You should be careful not to use the Terminal Services User Manager for regular domain administration, since it will increase the record size for users due to the additional user parameter fields.

The Windows 2000 Server Resource Kit contains the Application Security Registration utility console. This is a tool that offers administrators tighter control over applications and who can use them. When this console is in use, a user cannot execute a listed application unless that user has Administrator privileges.

## Remote Desktop Protocol

Remote Desktop Protocol (RDP) is to Windows 2000 Terminal Services as ICA is to Citrix MetaFrame XP. It is the protocol that carries the thin client session information, including transmission of graphical data and mouse/keyboard clicks, between the server and the access device.

The original version of RDP, version 4.0, was incorporated into the Windows NT 4.0 Terminal Services Edition. Windows 2000 Terminal Services and thin clients use RDP version 5.0. Based on the T.120 standards, RDP handles graphical transmission through an RDP display driver, which is run separately for each session. Each thin client session also runs its own Win32 kernel, with the display drivers contained in a session address of virtual memory. Receiving commands from the graphical display interface (GDI), the RDP display driver then passes them on to the Terminal Services device driver, which then encodes them in RDP format. The information is then sent to the Transport layer and through the protocol stack onto the wire to the client access device.

The new version of RDP, version 5.0, is packed with new features:

- **Bitmap caching** Storing graphics locally can enhance a client's performance. This is a new capability within RDP 5.0. During the session, graphics are stored locally and the client sends a list of keys corresponding to the bitmaps in cache. These keys are used to either call up locally stored graphics or to request new graphics from the server. Preset to 10MB, the bitmap cache storage size cannot be altered.
- **Client printer mapping** Without going into the difficulties of managing printers with RDP 4.0, the client printer-mapping capability adds a great benefit for administrators under RDP 5.0. When a client connects to a Windows 2000 Terminal Services session, the local printer (or printers) will be mapped to appear as network printer connections in the session. The client can print locally either from within the session or from applications running locally on

the desktop. The prerequisite for this functionality is that the drivers for the printer exist on the server.

- **Local/remote clipboard copying** When launching a remote desktop, the clipboard in the remote desktop is separate from that of the local desktop. With RDP 5.0, information that is copied to the clipboard in the remote session can be pasted to the local clipboard.
- **Shadowing** One of the values that help desks and administrators have found in thin client computing is the ability to shadow a user's session and troubleshoot user problems. This capability is available with RDP 5.0. The administrator can connect to a user's session and "shadow" it, viewing the session on the administrator's desktop—even taking control of the session. Security settings must be established to enable shadowing, and rights must be granted to let a shadowed session take place. Prerequisites for shadowing are that video resolutions must be the same or better on the administrator's machine, and both the administrator and the shadowed user must be logged in to the same server.

The RDP client, which runs only on TCP/IP networks, is available for 16-bit Windows (Windows for Workgroups 3.11) or 32-bit Windows (Windows 95, Windows 98, Windows Me, Windows NT 3.51, Windows NT 4.0, and Windows 2000). Windows XP contains a Remote Desktop client in its native code. For any machine that will be connecting to a Windows 2000 Terminal Server, the RDP client must be installed. For any machine connecting to a Windows 2000 Terminal Server with Citrix MetaFrame XP, an ICA client must be installed, or applications can be launched from a compatible browser. You can read more about Terminal Services RDP clients and Citrix MetaFrame XP ICA clients in Chapter 6.

## Citrix MetaFrame XP

Windows 2000 Terminal Services is the base operating system for Citrix MetaFrame XP. Extended by Citrix MetaFrame, the thin client server can support the ICA protocol clients in addition to RDP clients. The ICA protocol and Citrix MetaFrame's base features offer many advantages over standard Windows 2000 Terminal Services, one of the greatest advantages is the ability to treat large implementations of thin client servers as a single server farm.

### Independent Computing Architecture

Independent Computing Architecture (ICA) clients are unique to Citrix MetaFrame XP. If you do not have Citrix MetaFrame XP installed, you cannot use the ICA protocol clients, even though the ICA client software is freely available for download from Citrix's Web site. The ICA client offers some advantages over the RDP client:

- Available for all Windows operating systems, DOS, UNIX, Linux, Macintosh, and OS/2
- Available for EPOCH and Pocket PC mobile devices
- Windows Based Terminals
- Full client device mapping, including audio, COM port, and printer mapping
- Local drive remapping
- SpeedScreen2 latency reduction
- Multiprotocol usage
- Seamless windows

Citrix offers software for ICA clients that operate on DOS and 16-bit Windows 3.1 as well as Windows for Workgroups 3.11, Windows 95, Windows 98, Windows Me, Windows NT 3.51, Windows NT 4.0, Windows 2000, and Windows XP. In addition to these, Citrix offers clients for a variety of UNIX operating systems, Linux, Macintosh, and OS/2.

Wireless and mobile devices offer a new opportunity for thin clients. Because these devices are so small, they cannot support extensive applications. They can, however, support thin clients and run full-featured applications in remote-controlled sessions. Being able to perform such computing over public wireless networks or using wireless modems, businesses can leverage wireless devices and thin client computing to increase productivity to new levels. Citrix offers ICA clients for EPOCH and Pocket PC mobile devices and will likely continue to release new clients for other mobile devices as this market develops.

ICA thin client terminal hardware is referred to as a *Windows Based Terminal (WBT).* Each WBT runs an embedded ICA client that is immediately available to connect to the network. A WBT cannot run without a Citrix MetaFrame server on the network. Often used as a point-of-sale (POS) terminal or in kiosks, thin client machines do not contain complex hardware components. They mainly consist of a monitor, keyboard, and mouse along with the embedded ICA client.

An ICA client is able to run applications with sound by mapping remote audio to local audio. An ICA client can also map COM ports and LPT ports so that modems and other serial devices as well as all printers can be accessed. Keep in mind, however, that when audio mapping is used, it increases the bandwidth consumption of the session transmission.

Within a thin client session, the user sees the drives that are local to the server. This is not helpful if the user needs to save a file to the local computer. ICA clients can map a local drive to the remote session so that it appears as a drive letter within the remote session. In doing so, the user can save files and transfer files using standard drag-and-drop methods in the session's Explorer or simply save to the mapped drive letter through the application itself. There can be confusion, however, for users who assume that the C: drive in the remote session is the same drive as their local hard drive. Citrix has a solution to this problem. When you install Citrix MetaFrame XP, you are prompted to remap the local drives to different letters. The default is to remap the C: drive to M:. When you choose this option, a client connects to a session and sees the local C: drive as the session's C: drive, and the server's drives begin mapping from M: onward in the session. If you do *not* select the remapping function, when a client connects to a session, it sees the server's C: drive as the session's C: drive and the client's local C: drive as the session's V: drive (and any other local drives map backward alphabetically so that the D: drive is U: and so on.). Citrix MetaFrame XP allows the administrator to turn off client drive mapping altogether so that data cannot be saved to local drives at all. This option is helpful when you use thin client sessions in highly secure situations in which data should not be saved to other machines.

SpeedScreen2 is a feature that increases the performance of ICA. By cutting the average packet size 25 to 30 percent, the transmission rate is improved up to 60 percent. In remote locations, especially those that depend on unreliable satellite connections, SpeedScreen2 offers a tremendous advantage.

ICA is known for its high performance, even when transmitted to mobile and wireless devices. The reason it performs so well is that ICA consumes no more than 20Kbps. Because of its small size, a standard modem connection speed of 28.8Kbps appears to be running locally.

When you consider the Open Systems Interconnection (OSI) protocol reference model, which is shown in Figure 1.16, the ICA protocol typically sits at the Presentation and Session layers, just as RDP does. However, ICA was built to flexibly integrate with multiple protocols. It requires a Transport layer. If ICA is transmitted across a protocol that does not include a Transport layer, such as Internetwork Packet Exchange (IPX) or Internet Protocol (IP), ICA incorporates the services that are provided at the Transport layer. Figure 1.17 depicts how ICA works in conjunction with IPX running on an Ethernet segment.

In addition to all these functions, ICA clients have the same features as RDP clients, including:

- Bitmap caching
- Local/remote clipboard copying
- Shadowing

**Figure 1.16** The OSI Protocol Model and RDP and ICA Protocols

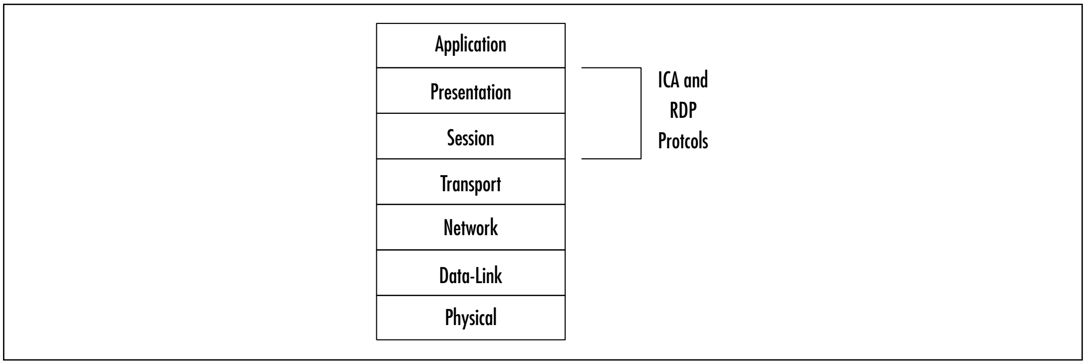
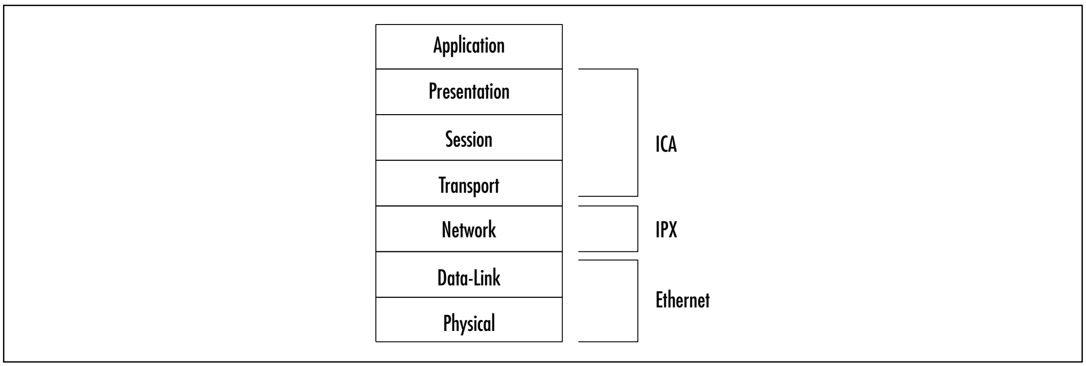



**Figure 1.17** ICA with IPX over Ethernet




Many highly regulated industries, the government, and businesses that intend to use thin clients for confidential information require encryption to secure the transmission of data between the thin client and the Citrix MetaFrame XP server. Citrix offers SecureICA, which is available with 40-bit, 56-bit, and 128-bit capabilities, to encrypt session transmissions. This encryption uses the RC5 standard created by RSA.

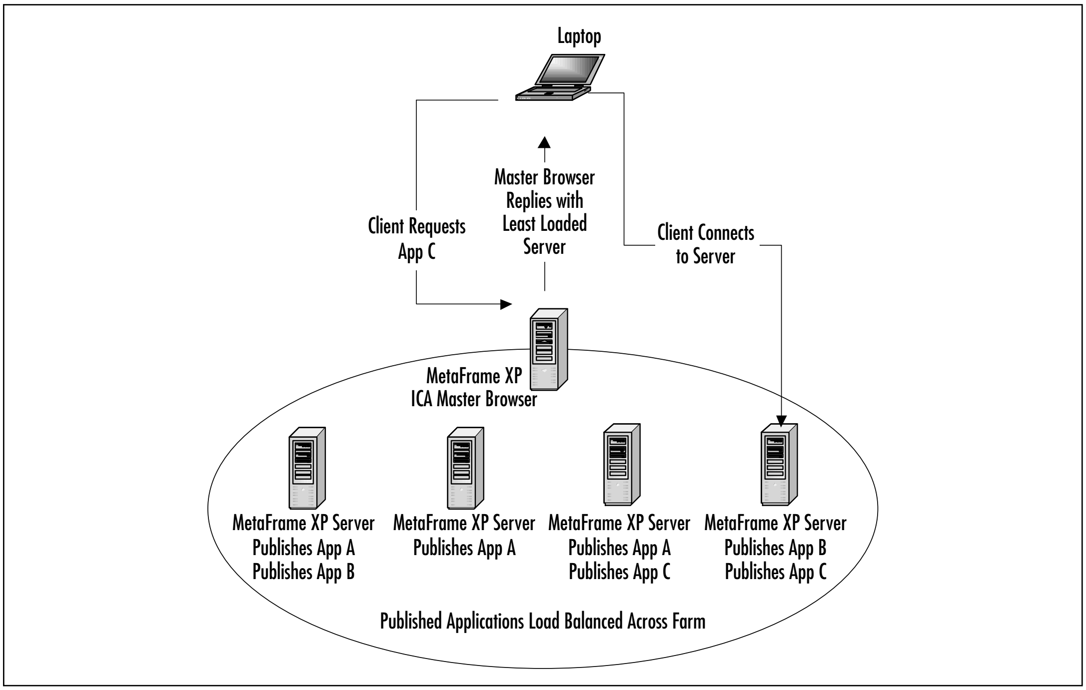
ICA clients support seamless windows. An administrator can publish an application so that it launches from its own window without the remote desktop appearing behind it. This capability makes it appear as though the application has launched locally. Available only with 32-bit Windows ICA clients, the remote application appears in the taskbar and responds to **Alt + Tab** to switch between applications. Users will not know whether the application is remote or local. When using seamless windows, you should consider using license pooling, in which a client can open multiple remote applications and consume only a single ICA license. However, if you choose not to use license pooling, a client will consume an ICA license for each remote application executed.

# Moving to the Server Farm

The key concept of the *server farm* has little to do with the ICA client. Server farms are centered around managing multiple servers on the network and preferably being able to treat them as a single resource for user access. Citrix MetaFrame XP offers enterprise-level scalability in its server management tools.

One advantage to Citrix MetaFrame XP is that it supports load balancing of published applications. Through application publishing, users connect to the name of an application rather than the name of a server. The back-end processes include load calculations to determine which server has the most available resources to support a new session. These back-end processes are shown in Figure 1.18.

**Figure 1.18** Load Balancing of Published Applications




Load balancing for published applications is the first step. In a Citrix MetaFrame XP server farm, a group of servers that typically publish the same applications can be logically grouped together to facilitate application management. When using a server farm, users are not prompted for multiple logons. Instead, their credentials are passed to any server when a session request is made.

Citrix MetaFrame XP introduces a new concept called *Independent Management Architecture (IMA)*. IMA is a service that installs on each server. IMA services communicate with each other, supply data to a database, and update a local host cache in order for the server farm to be managed as a single unit. Citrix MetaFrame XP includes a number of management utilities:

- **Citrix Management Console** This is an IMA-based tool for managing the server farm. The administrator need only log on to a single Citrix Management Console to manage every Citrix MetaFrame XP server in the network. The administrator will be able to view and update all the servers within the enterprise, regardless of the farms to which they belong.
- **Citrix Web Console** This Web-based utility for managing the server farm is intended for installation on a Windows 2000 server with Internet Information Services (IIS). From the client perspective, you must also use Internet Explorer 4.0 or later. The Citrix Web Console does not support Netscape.

Because Citrix MetaFrame XPe is built for enterprise deployments, it includes additional management features that are not included in other versions, namely Installation Manager. Installation Manager helps an administrator deploy applications to predefined groups of servers. By creating groups and establishing this method of deployment, an administrator can save a great deal of time and effort in application deployment. Resource Manager is another feature included only in Citrix MetaFrame XPe. Resource Manager is a centralized location for settings, thresholds, and metrics and helps monitor the performance of servers in the farm. Network Manager is a third feature found only within Citrix MetaFrame XPe and helps integrate server farm management into enterprise network management.

# Gaining Digital Independence with Thin Clients

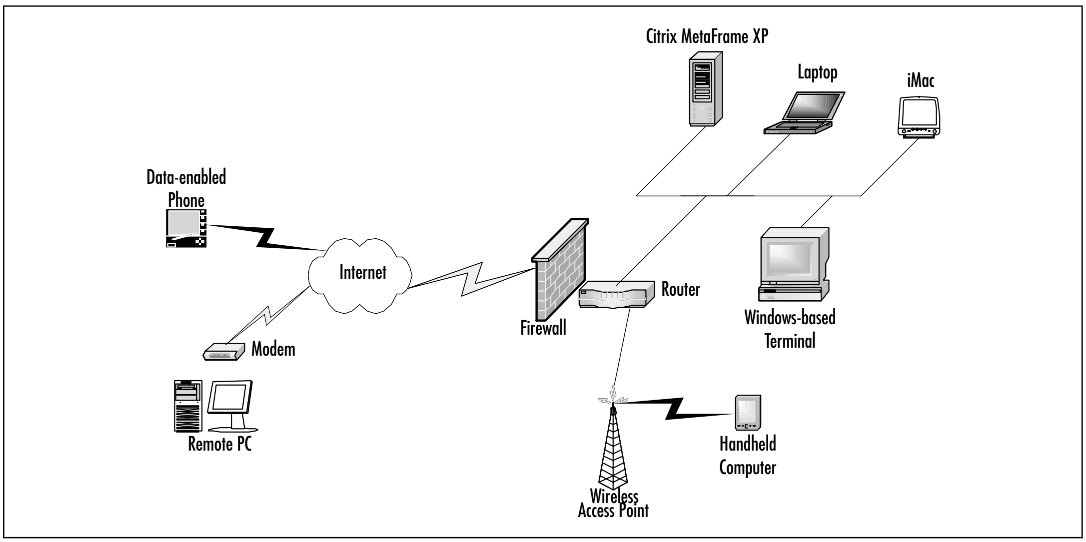
The Internet brought about a certain sense of independence in computing. Using a common global internetwork, people could publish formatted text documents with graphics and reasonably expect that any operating system could access the information. Prior to the Internet and Hypertext Markup Language (HTML), the interoperability of various operating systems was limited because of the proprietary nature of document formats. Since then, browser capabilities have increased to include scripts, applications, and services. Digital independence, in the form of a person delivering an application to any device in any location over any connection, seemed to be the next step.

Mobile computing has brought a new element to the Internet. Mobile devices have limited capabilities because, although they are becoming smaller and more intelligent, they still do not have the screen resolution and storage capacity available in a desktop PC. This means that rendering Internet data to a data-enabled cell phone or Pocket PC must be adjusted for the smaller screen and storage of the mobile device. The transmission of the data through wireless networking is constrained due to the low-bandwidth capabilities of public wireless networks. Digital independence is not as easy to achieve as it might seem.

Citrix MetaFrame XP can be used to help break the bonds of proprietary networking. Using WBTs, a network administrator does not need to manage PCs in order to deliver 32-bit Windows desktops to the network.

For administrators with networks that include many different types of operating systems, whether UNIX, Windows, or others, being able to run the same application on any operating system can remove many headaches as well as reduce costs. In a business that has many different operating systems, when a mission-critical application is rolled out to all end users, some of the operating systems might be incompatible. An administrator is left with options that include developing special client applications for the incompatible operating systems, using a Web-based application that might not be fully featured, or not rolling out the application at all. This is a situation that calls for Citrix MetaFrame XP. Offering full digital independence, a 32-bit Windows application can be delivered to any ICA client, regardless of the operating system of the access device, even if it is a mobile one connected to a wireless network. In fact, an application can be delivered from within a browser on the Internet to anonymous users, if need be. Figure 1.19 displays how Citrix MetaFrame XP can be used to achieve digital independence.

**Figure 1.19** Digital Independence with Citrix MetaFrame XP



# Achieving Bottom-Line Value with Thin Clients

When faced with a large deployment project for Citrix MetaFrame XP, most network administrators must justify the costs of the project. They need proof of corporate benefits in order to obtain budget dollars. A cost/benefit analysis might be necessary for making a decision between rolling out Citrix MetaFrame XP or selecting a different technology. Regardless of the impetus, looking at the financial impact of such a project is usually part of the decision-making process.

Usually two types of costs and benefits are reviewed: *hard dollars* and *soft dollars*. Hard-dollar costs are costs of items that are paid for with cash. Hard-dollar benefits involve items that can be accounted for as increased revenues. Hard-dollar benefits can also be costs that are removed, thereby increasing profits. For example, if you currently pay $100 per month for each PC's warranty, and by implementing Citrix MetaFrame XP with WBTs, you no longer pay $100 per month for any PCs, you will see a great improvement in profits.

Soft-dollar costs are items that cost time and effort but are not accounted for in a budget. For example, if 30 Sun Solaris users must walk 20 minutes to a different department daily to use a Windows 2000 PC in order to run a specific application, and their average hourly wage is $10, you are spending 600 minutes' worth of soft dollars a day, which equates to 10 hours of time, or $100, per day. Soft-dollar benefits are identical in nature. When you can save time or effort or increase productivity, you receive a soft-dollar benefit. The exact dollar value of any soft-dollar cost or benefit is not easy to figure out. Take our example of the $100-per-day soft-dollar cost. If you removed such a soft-dollar cost, you would receive a soft-dollar benefit. Given that an average person in that department had a productivity level of $100 per hour, you might think that the soft-dollar benefit would be an increase of 10 hours, or $1000, per day of productivity increase. But people are not that predictable. The likelihood that each person in that department would put all of the extra 20 minutes toward productive work is very low. In reality, you might expect a 50-percent productivity increase for those hours.

## Calculating Costs

This section covers hard costs and then discusses how to calculate soft costs. When you calculate the hard costs of a Citrix MetaFrame XP project, you must list all the hardware and software to be purchased as well as the cost of any labor used in the execution of the project. When rolling out Citrix MetaFrame XP, you should consider the following costs:

- **Server hardware**
  1. Number of servers
  2. Type of servers
  3. Number of symmetric multiprocessors (SMP)
  4. Storage or Redundant Array of Inexpensive Disks (RAID)
  5. Amount of RAM
  6. Single or redundant network interface cards (NICs)
  7. Single or redundant power supplies
- **Peripheral equipment**
  1. Uninterruptible power supply (UPS)
  2. Number of keyboards
  3. Number of mice
  4. Number of monitors
  5. Number of keyboard/monitor/mouse switches
  6. Racks for storing server equipment
  7. Cables
- **Network connectivity** In most cases, you will likely leverage existing connections. However, if you will be installing new network or Internet connectivity, these costs may be incurred:
  1. Installation of new connections
  2. Monthly lease or periodic charges for connectivity
  3. Number of routers
  4. Number of hubs
  5. Number of switches
  6. Installation of cabling or wireless access points
  7. Number of wireless or standard NICs to replace existing
- **End-user equipment** Depending on your deployment, these costs might be incurred, regardless of whether or not you roll out Citrix MetaFrame XP. If so, you should not consider them in a cost comparison. If you are comparing a desktop rollout to a Citrix MetaFrame XP rollout, you will likely

incur fewer costs in this area and therefore *must* include these costs for a true cost comparison:

1. Number of desktops
2. Number of WBTs
3. Number of mobile/handheld devices
4. Number of monitors
5. Number of keyboards
6. Number of mice
7. Number of NICs
8. Cables

- **Software**

1. Windows 2000 Server licenses
2. Windows 2000 Terminal Services clients
3. Citrix MetaFrame XP licenses
4. ICA client licenses
5. Number of application licenses
6. Number of new operating system licenses (only when rolling out new desktops with upgrades)
7. Utilities selected for use in deployment

- **Labor**

1. Design services
2. Assessment services
3. Configuration services
4. Installation services
5. Administrative and operational services
6. End-user training services

You will likely go back to your hard costs list and add or subtract items many times over. In the area of end-user equipment, you could find that you can use some old PCs currently stored in a closet or avoid rolling out PCs to a group of users because their equipment can be reused.

Soft costs are not as easy to quantify. You need to use a little imagination in order to determine the soft costs that will be incurred. You should consider how the project will be executed, delays that will be caused for end users, and how those delays will affect productivity. You should estimate any increase in needs for troubleshooting and help desk support, which generally increase immediately following a new technology deployment. The following list might help you:

- **Support costs**
  1. Increased help desk costs
  2. After hours support needs
  3. 24 x 7 management and monitoring costs
  4. Warranty support
- **Software costs**
  1. Software maintenance contracts
  2. Hardware maintenance costs
  3. Upgrade costs and frequency

## Calculating Benefits

Businesses with thin client implementations derive benefits that affect the bottom line. Not only can a business implement a thin client solution to take advantage of new opportunities that can generate new revenue, but the business can reduce costs and avoid future costs. The situation depends on the project's business goals and how the technology is deployed to meet them. Some of the most common business objectives for thin client projects are:

- Reduce costs by extending the life of older equipment
- Reduce costs by avoiding implementation of higher-bandwidth links
- Reduce costs through use of remote control support methods

### Extending the Life of Older Equipment

Avoiding the need to upgrade operating systems and desktop hardware can save millions of dollars in an enterprise network. The ability to deploy a thin client solution to every user in an enterprise is unlikely. Some users, through their heavy use of graphics-intensive applications or their need to work offline, cannot use a thin client 100 percent of the time. On the other hand, a business can deploy a thin client solution to users who never are expected to need offline computing. Reduced costs in this scenario consist of:

- Avoided costs of new equipment purchases
- Avoided costs of operating systems licenses
- Reduced administrative costs and increased productivity, since users can switch machines without loss of data
- Increased viability of data because data is stored to centralized servers
- Avoided costs of training for new equipment and operating systems

## Reducing Bandwidth Consumption

In a telecommuting environment, the ability to execute mission-critical applications is increased. Running that mission-critical application across the wire can become frustrating due to performance across a low-bandwidth connection. Even if an individual connection could perform well, the corporate conduit to the Internet might not support multiple users running the application. If a client application is "fat," a modem connection might not be usable to support it. A company could be faced with installing cable modem connections, DSL, or ISDN links in order to support telecommuters. In the example shown in Figure 1.20, 10 users are running an application through the Internet. If the bandwidth consumption of these fat clients adds up to more than 1.544Mbps, the corporate Internet connection becomes a bottleneck for the users. Performance is affected, not only for the users on the Internet, but also for any users who are attempting to access the Internet from within the corporate network. The options are to either upgrade the Internet connection or to implement a thin client.

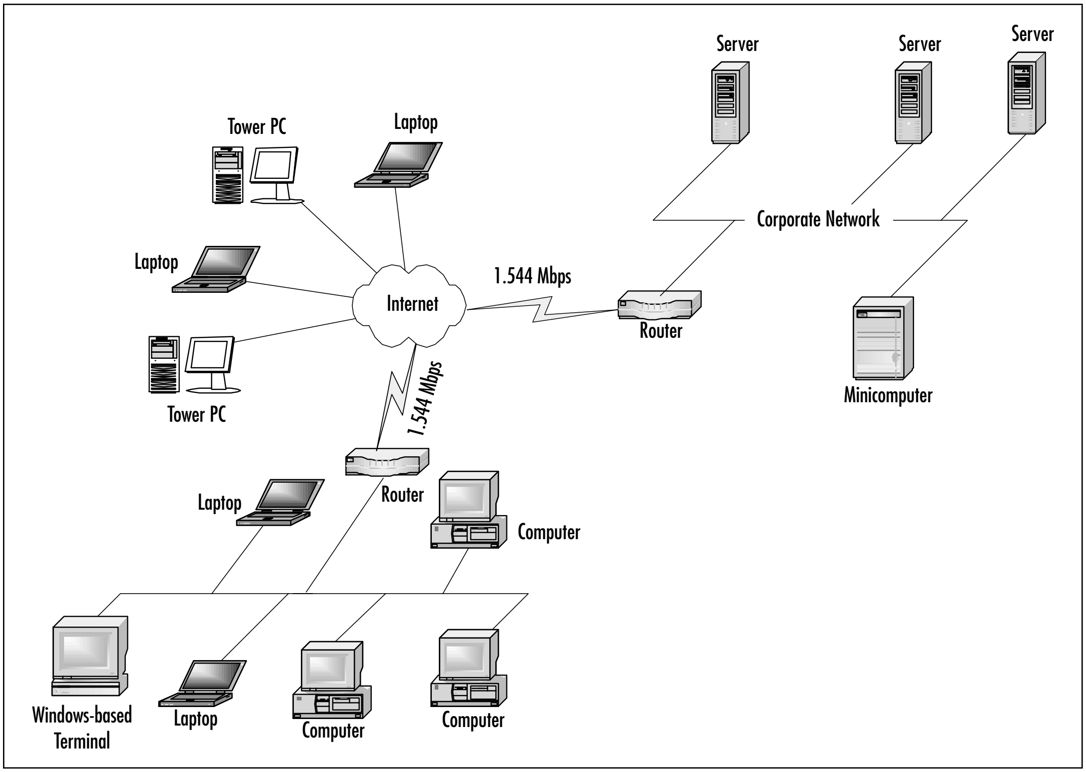
The benefits of implementing a thin client solution are:

- Avoiding the need for faster Internet links for telecommuters
- Avoiding the upgrade costs for Internet connectivity for the corporate connection
- Centralized management of the client application

## Allowing the Help Desk to Shadow Users

The process of supporting end users can be fraught with visits to the deskside. If there is a problem with an application and the help desk staff cannot walk the user through the issue on the phone, a technician is sent to the desk to actually see the problem and troubleshoot from there. The ability to see firsthand what the user is doing or what errors are taking place is usually all it takes to fix the problem.

**Figure 1.20** Running Applications Across the Internet




Shadowing a user's session can become a benefit by itself. Instead of a user calling the help desk and a technician being dispatched to the user's desk, sometimes many hours later, the help desk staff can shadow the user's session and view the problems directly, resulting in faster resolution. Since a deskside support visit costs as much as 10 times the cost of a help desk support call, shadowing can work out to be a tremendous benefit. The benefits for this type of situation consist of:

- Reduced numbers of deskside support calls
- Faster resolution for support calls
- Increased productivity for end users with support calls

# Meeting Business Needs

Business needs fall into two categories. Some business needs are simply *musts*. Driven by regulatory compliance, relief from nuisance systems, or simply a requirement handed down from upper management, these types of projects are nearly always given top priority. The other category of business need is *growth*. Using technology as a means to grow the business makes for an exciting project.

## Security

Technology security is a must. In regulated industries, data might not be allowed to be distributed across different machines, and users might not be allowed access to certain applications. Using Citrix MetaFrame XP, an administrator can easily meet these types of security needs. By installing an application on a Citrix MetaFrame XP server, the administrator can centrally manage it. The administrator can easily grant privileges or limit access to the application. Restricting the ability to print, map local drives, and copy to the local clipboard, the administrator can ensure that the data is retained solely on the Citrix MetaFrame XP server. The administrator can further encrypt the data transmissions and require authentication to either Active Directory or Novell Directory Services.

## Internet Deployment to Unknown Clients

Using the Internet to deploy applications can be a cumbersome process. Many businesses have looked to application developers to reprogram their software to be Web-compatible and then have dealt with the process of securing, encrypting, and managing those applications and their data from an entirely new back-end infrastructure. This situation is complicated by the need to ensure compatibility with different browsers and desktop operating systems.

An administrator can easily install Citrix MetaFrame XP as the means to deploy applications across the Internet. This method includes a host of benefits:

- Avoiding the cost of application programming
- Avoiding the complications of integration
- Speeding up the deployment time
- Leveraging the existing investments in technology

# Summary

Understanding the history of the mainframe and distributed computing models is the first step toward realizing the benefits that thin client computing can bring to an enterprise. The mainframe, built on a centralized system with heavy investment on the mainframe hardware and little investment in end-user devices, is easy to manage and monitor. Mainframe computing consumes very little bandwidth, which is beneficial to telecommuting. On the other hand, the mainframe is an inflexible model that does not allow users to have access to offline computing, and it does not have graphical capabilities.

Distributed computing, by contrast, is difficult to monitor and manage. It has a low investment in a central computing environment in comparison to mainframe costs, and it requires a much higher investment in end-user access devices. Although it is a flexible model, enables offline computing, and supports graphic-intensive applications, a distributed computing client can consume a great deal of bandwidth in telecommuting.

Thin client computing maximizes the benefits of both the mainframe model and the distributed computing model. By combining low bandwidth consumption, graphical applications, terminals, and centralized management, thin client computing offers the best of both models.

The history of Citrix MetaFrame XP is intertwined with that of Microsoft's Windows 2000 Terminal Services. Originally, Citrix developed a solution for OS/2 called WinView, which enabled multiple window sessions to run simultaneously to remote users running remote control solutions. This technology, named MultiWin, was ported to Microsoft's Windows NT Server, along with the Independent Computing Architecture (ICA) protocol. Microsoft realized the value of thin client computing and licensed the MultiWin technology from Citrix for their Windows NT 4.0 Server product as a Terminal Server Edition, then incorporated it into all Windows 2000 Server versions with Terminal Services. Microsoft did not license the ICA protocol and so developed the Remote Desktop Protocol (RDP) client for Windows desktops. Citrix retained ICA and incorporated it as well as the Independent Management Architecture (IMA) for server farms in the Citrix MetaFrame XP product, which installs as an add-on to Windows 2000 Terminal Services.

Employing thin client computing, users can work on sessions originating from a centralized server, similar to the mainframe model. The client requirements are very low, so businesses can extend the life of existing hardware or begin to incorporate mobile devices into their network infrastructure while still being able to deliver the latest applications. The bandwidth consumption of a thin client is much lower than a standard client for an application, making it an ideal solution for telecommuting.

Before implementing thin client computing, administrators need to consider the costs and benefits of the implementation. Using a cost-comparison method, it is easy to show how money can be saved by implementing Citrix MetaFrame XP rather than some other technology, such as a desktop upgrade project. When analyzing costs and benefits, both hard-dollar and soft-dollar costs and benefits must be taken into account. Hard-dollar costs are anything that must be paid for out of a budget. Hard-dollar benefits are found in avoided costs or increased revenues. Soft-dollar costs and benefits are not easy to determine, because they do not necessarily fall into a budget. These items can include increased productivity, reduced time consumption, and reduced effort.

This chapter has merely skimmed the surface of the concepts and technology within Windows 2000 Terminal Services and Citrix MetaFrame XP. As this book progresses, you will learn about starting a project for implementation of Citrix MetaFrame XP, the design process, installation, and ongoing administration. Solution-oriented chapters provide a holistic look at implementing Citrix MetaFrame XP with a single business goal in mind. Your adventure into thin client computing has just begun.

# Solutions Fast Track

## The Mainframe Model Meets Distributed Computing

- ☑ The mainframe model grew out of the original computing systems, dating back to the 1940s.
- ☑ Distributed computing gained in popularity when personal computers were put on the market.
- ☑ Mainframe computing has the advantages of reduced desktop hardware costs, single point of administration, single type of administrative skill set, and a simple architecture with low bandwidth consumption.
- ☑ Mainframe computing drawbacks include single point of failure, character-based-only applications, and potential bottlenecks due to time-sharing systems.
- ☑ Distributed computing has the advantages of reduced costs for centralized hardware (no mainframe), no single point of failure, flexibility, and a scalable architecture.
- ☑ The disadvantages of distributed computing are greater administrative costs, security risks, and lack of a centralized backup.

## The Difference Between Remote Node and Remote Control

- ☑ Remote node is the ability to run a computer over a remote link and access network resources in the same way as though the devices were locally connected.
- ☑ Remote control is the ability to connect to a remote control host and view its screen locally on the desktop, taking control of that remote host in order to function on the network.
- ☑ Windows 2000 Terminal Services with Citrix MetaFrame XP offers the ability to connect as a remote node as well as run remote control sessions.

## The Evolution of the Thin Client

- ☑ Citrix developed a technology called MultiWin that enabled multiple sessions to run on a single server for remote control computing.
- ☑ Citrix ported the MultiWin technology along with the Independent Computing Architecture (ICA) protocol to Windows NT in the Citrix WinFrame product.
- ☑ Microsoft licensed MultiWin in Windows NT 4.0, Terminal Services Edition, and developed a Remote Desktop Protocol (RDP) client for it.
- ☑ Today's Citrix MetaFrame XP runs as an add-on component to a Windows 2000 Terminal Services server, bringing the ICA protocol and Independent Management Architecture (IMA).

## Gaining Digital Independence with Thin Clients

- ☑ Digital independence is the ability to run an application on any device over any connection.
- ☑ Mobile computing and wireless networks, being too small to store and run full-featured applications, are an ideal platform for thin client computing.
- ☑ Using thin client computing, an administrator can deliver a 32-bit Windows application to computers that run incompatible operating systems, to Windows Based Terminals (WBTs), or to handheld computers over networks running different protocols.

## Achieving Bottom-Line Value with Thin Clients

- ☑ When weighing a thin client project, a cost/benefit analysis can help justify the project budget to a corporate sponsor.
- ☑ There are two types of costs and benefits to consider: hard-dollar costs and benefits that affect a budget directly, and soft-dollar costs and benefits that are more difficult to quantify.
- ☑ A business can implement a project with business goals of reducing costs, increasing revenues, and complying with regulatory requirements.

# Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

**Q:** How does Windows 2000 Terminal Services with Citrix MetaFrame XP supply both remote node and remote control computing?

**A:** Windows 2000 Server includes Remote Access Services (RAS). This is the server portion of a remote node computing solution. Terminal Services and Citrix MetaFrame XP add the ability to run many remote control sessions simultaneously from the server. A network administrator could install RAS and configure it to support both dialup and VPN clients. The network administrator could install Terminal Services and Citrix MetaFrame XP on the same server. A client could dial up or connect to the corporate network through the Internet and then execute a remote control session using a single server on the back end.

**Q:** How does thin client computing speed up performance?

**A:** A thin client uses very little bandwidth because it compresses the graphic instructions that are sent from the server to the client as well as the keyboard and mouse clicks that are sent from the client to the server. No actual data needs to be transferred, except when saving data to the local computer or when printing. The result is that an entire desktop appears to be running locally with excellent performance, even when the remote client is connected over a 28.8Kbps dialup link.

**Q:** We have users with Macintosh and UNIX workstations. Can we use Windows 2000 Terminal Services? Or must we install Citrix MetaFrame XP?

**A:** Windows 2000 Terminal Services supplies RDP clients for only Windows for Workgroups 3.11 and 32-bit Windows operating systems. You can look for a third-party vendor that develops RDP clients for Macintosh and UNIX workstations. Or you can deploy Citrix MetaFrame XP and use the ICA clients for them.

**Q:** We are concerned about security for our financial application. We don't want data copied to people's home computers, and we don't want people to shadow sessions. Our accounting department is proposing that they begin telecommuting. Can we use Terminal Services with Citrix MetaFrame XP to deploy the application and still maintain tight security?

**A:** Yes. You can restrict the security settings on the Citrix MetaFrame XP server to prevent all shadowing, or you can prevent shadowing on a user-by-user basis. You can encrypt the session and use directory services authentication. You can also prevent local drives from being mapped so that data can't be copied or saved to home computers; in addition, you can prevent printers from being mapped so that data can't be printed and used later on.

**Q:** We want to deploy an application over the Web without a lot of development effort. Should we use Citrix MetaFrame XP or Windows 2000 Terminal Services?

**A:** Windows 2000 Terminal Services, even though allowing a Web client, cannot publish an application individually through a Web browser, whereas Citrix MetaFrame XP can.