# THIN CLIENT NETWORKS

## Overview and History of Thin Client Computing

The concept of *thin client* computing is not a new one – before PCs came along most large businesses used mainframe computers to process data. These generally consisted of powerful central computers that handled all data processing. Users of these computers had dumb terminals that could only handle simple text entry and display – no mice! These dumb terminals communicated directly to the central mainframe for almost everything they did.

One beauty of this set up is that support costs were minimal – in fact, modern-style IT helpdesks were relatively unheard of as there was no need to have engineers running around fixing problems. Either it was a programming problem or some piece of *hardware* needed replacing! Then along came the PC. Suddenly data was being held on computers running on people's desks. It seemed the age of centralised computing was over as users got used to graphical displays and being able to perform a number of tasks with just one computer rather than different dumb terminals for different applications.

In 1995 a company called Citrix released a product called Winframe – this was a play on Windows-Mainframe and was literally intended to be that. Users could have slightly-less-dumb terminals on their desks and connect to powerful central servers that ran lots of user sessions. All the data and processing stayed in the computer room away from user's desks – all the user saw was the display of a Windows computer as if it was running on their PC just like normal.

It's worth noting that Unix – another *network operating system* that Linux is based on - had been able to do this type of thing for many years already. X Windows, the *Unix* graphical user

interface, had the capability to split application processing from the graphical display built into its design. Winframe was significant though because it brought some of these capabilities to the Microsoft Windows environment which was running on the majority of desktop computers at the time.

## Key Benefits and Issues

Thin client computing offers a number of benefits including:

- Greater security as data isn't being stored out on desks where it might not be physically secure or even being backed up
- No need to constantly upgrade desktop PCs – if more processing power is needed only a small number of central computers need to be upgraded. Obvious benefits for the budget and also for the environment
- Users can connect to their desktop and applications from anywhere that is connected to the network – even from home or on the road
- Applications can be updated for 100s of users in minutes as only the central computers need to be updated
- Support costs are reduced as only simple devices are required out on users' desks
- Incremental costs of adding additional users should be lower than if they used conventional PCs through the saving from reduced hardware costs and the cost of configuring each new PC

In with all the good things are some **drawbacks**. These include:

- Heavily graphical applications require a lot of processing and bandwidth to work and are probably not best suited to this type of environment

- Installing lots of applications and hardware such as printers on central computers can cause conflicts – the design of the system must be right or there can be real problems

- If the central network or servers go down users may not be able to use their computers at all – if you are using dumb terminals they will need the central network to be able to do even the most basic things

- Connecting external devices such as PDAs can be very difficult or impossible depending on the type of the device. Quite often it's easier to include some fat client machines on the network for those people that need them

## Technologies Available

There are four main systems that this document will focus on. They are:

- Microsoft Terminal Services

- Citrix Presentation Server

- The X Window System

- The Web / Browser-Based Applications

**Microsoft Terminal Services**

Microsoft saw the potential of the new *software* from Citrix and rather than write their own software they reached a deal with Citrix (helped by the fact Microsoft had invested in Citrix early on) where Microsoft would licence some of the code from the Citrix system. This effectively gave Microsoft a cut down version of the Citrix system. Microsoft then couldn't lose as if someone wanted to implement the Citrix system, they still had to purchase Terminal Services licences. Great business!

For the more serious thin-client solution running a Windows environment, Citrix was always the choice but Terminal Services offered a relatively straight-forward and cost-effective option, especially for not for profit organisations where licensing costs are relatively small.

In recent years tools have been developed to allow Terminal Services to share some of the higher services of Citrix but at a much lower cost. These include load balancing based on server load, seamless applications (where an application appears on your normal desktop as an icon and runs as if it is installed as normal but is in fact running on a Terminal Server), secure gateways to increase access security and encryption.

Terminal Services is based on the Remote Desktop *Protocol* (RDP) and is now even included in Windows XP for remote access or support purposes allowing users the ability to let a remote technician take over their PC without installing additional software.

**Citrix**

The current version of the system from Citrix is called Presentation Server. As mentioned earlier, Citrix is generally targeted at the larger enterprise due to its greater cost (even for not for profit organisations) and greater scalability.

For the small and medium sized organisation it can be difficult to justify the additional expense unless there is an application-specific requirement for Citrix over Terminal Services.

Citrix Presentation server is based on the ICA (Independent Computing Architecture) protocol.

**X Window**

The X Window System is the underlying protocol for the dominant windowing system for Unix and Linux based systems. In combination with an X Terminal of some sort (either dedicated

hardware or as software running on a PC) applications can be run in effectively the same way as with Terminal Services and Citrix.

The X Window System originated is the 1980s through a partnership between MIT, Stanford University and IBM with the goal of creating a graphics system that was platform independent.

As X Windows is part of Unix / Linux systems it would not be suitable for running Windows-based applications on the central server. However, there are Windows and Citrix clients available for Linux which does mean that you can run Windows applications on Linux desktops.

**The Web / Browser-Based Applications**

Although you may not have considered it, many of the websites we use today are in effect thin client applications. Next time you enter your details into a web page have a think about what is happening. You type some information into the page, click a button and off goes the data to be processed by some other computer. The results are then displayed back at your screen.

New technologies such as Ajax are being developed to help make using applications in a *web browser* as interactive as normal applications installed onto your computer. Most new line of business applications either have browser support or have moved completely over to being browser-based.

It's almost as if the browser (*Internet* Explorer, Firefox etc) has become the operating system in which applications can run. Now it (almost) doesn't matter whether you are running Linux / Unix / Windows / Something Else – as long as you have a standard browser and your applications are written to those accepted standards, you should in theory be able to run whatever you need.

# Future of Thin Client Computing

Many people see the web as the future of computing in general – you can tell this by the likes of Google and Microsoft jostling for position with the web-based applications.

But don't expect everything to change overnight! In the year 2000, at the height of the dot com boom everyone thought we would be using Application Service Providers for all our computing needs by now and internal IT departments would be long gone! It might be heading in that direction but IT is a pretty conservative world and big changes often take longer than people might think.

One of the biggest drawbacks with applications being hosted by external companies is what happens if that company goes bust – if your software supplier disappears and you run everything on your internal servers no problem – you have time to find a new supplier and you can make do with what you have. But if you can't access you donor *database* because they haven't paid the electricity bill you have a major business problem. Disasters such as Enron and WorldCom have highlighted that even the biggest organisations can go bust.

So, for the near future it is likely that applications will continue to make better use of web technologies to make the desktop environment (Windows, Linux, *Mac*) less and less important. Many applications will use a browser to run in but will talk to internal servers so organisations have control over their key data. Technologies such as Terminal Services, Citrix and X Windows still have a massive role to play. Most applications still do require installing and managing in the traditional way and many applications will take a lot of work to move away from the current architecture. Some smaller vendors may take longer to move their systems over due to the expense.

Source: http://www.ictknowledgebase.org.uk/thinclientnetworks