

2010

Tecnología Thin Client

Sistemas Informáticos

Luis Miguel Moya Moirón
G990192
25/01/2010



Indice

Sistema Informático desarrollado.....	3
Objetivo.....	7
Introducción: ¿Qué es un Thin Client?.....	7
Visión general de la tecnología Thin Client.....	8
Arquitectura en Red.....	9
Hardware de un Thin Client.....	15
Software Thin Client.....	22
Fundamentos Thin Client.....	25
Arranque por red.....	25
Sesión Remota.....	29
Ventajas de los Thin Client.....	35
Desventajas de los Thin Client.....	39
Escenarios Thin Client.....	41
Protocolos Thin Client.....	44
Protocolos para el arranque por red.....	44
Protocolo PXE.....	44
Protocolo DHCP.....	45
Protocolo TFTP.....	47
Funcionamiento del protocolo PXE.....	49
NBP.....	51
Protocolos para sesiones de escritorio remoto.....	51
X-Window System / XDMCP.....	52
VNC.....	59
Terminal Services / RDP.....	60
Tecnología NX.....	63
Comparativa Software Escritorio Remoto.....	65
Software Libre en la Tecnología Thin Client.....	72
Conclusiones.....	76
Bibliografía.....	76

Sistema Informático desarrollado

El sistema desarrollado consiste en la **Adaptación del Sistema Lan Core para el Sistema Operativo Linux**, desarrollado inicialmente para el sistema operativo Windows.

Lan Core es proyecto software libre con licencia GPL, que implementa en un sistema completo la **Tecnología Thin Client** que se detallará más adelante en este documento. En este documento nos vamos a centrar en resumir brevemente la adaptación realizada del sistema Lan Core a Linux y en presentar la Tecnología Thin Client.

Para más información de Lan Core se puede consultar las siguientes direcciones Web:

- Web oficial de Lan Core: <http://lancore.sourceforge.net/>
- Wiki de Lan Core: http://tedeco.fi.upm.es/wiki/Lan_Core
- Source Forge de Lan Core: <http://sourceforge.net/projects/lancore>

El desarrollo se ha realizado sobre un entorno Virtual VMWARE, en donde tenemos una máquina Linux con distribución Ubuntu 9.1 que actúa como servidor y una máquina que simula el Thin Client, se trata de una máquina de 128 Mb de RAM que no tiene disco duro.

El desarrollo se ha llevado a cabo en varios pasos:

1. Configuración de los servicios en el servidor Linux para permitir al cliente el arranque por red. Esta tarea se ha realizado mediante la instalación y configuración los siguientes servicios:
 - Directorio boot de Lancore original. Este directorio mantiene todos los elementos que permiten arrancar al Thin Client: el cargador de arranque del sistema operativo PXELinux, una imagen de un sistema operativo empotrado basado en el kernel de Linux, un sistema de ficheros empotrado, drivers para los dispositivos gráficos y ficheros de configuración para el Thin Client.

- Instalación y configuración de un servidor dhcp para permitir la asignación de una dirección IP al Thin Client. Esta tarea se ha llevado mediante la instalación del paquete dhcp3-server y la configuración del fichero de configuración dhcpd.conf.
 - Instalación y configuración de un servidor TFTP con soporte para el protocolo PXE para permitir la descarga del cargador de arranque y del sistema operativo en el arranque del Thin Client. Esta tarea se ha llevado a cabo mediante la instalación del paquete atftpd y su fichero de configuración.
2. Arranque por red del Thin Client y conexión a escritorio remoto de Windows. De esta manera se prueba que el arranque por red sobre Linux funciona correctamente, para posteriormente continuar con el flujo de ejecución que tenía en la versión para Windows y conectarse remotamente al escritorio Windows por RDP.
3. Configuración de un cliente Linux acceda remotamente a un escritorio Linux. Para realizar esta tarea se han realizado los siguientes pasos:
- Instalación de una nueva máquina virtual que arranca una imagen Ubuntu 9.1 para simular un cliente Linux.
 - Instalación del paquete Xnest.
 - Configuración y habilitación en la máquina servidora de los servicios necesarios para permitir la conexión al escritorio remoto por XDMCP.
 - Prueba de conexión en local desde la máquina servidora a sí misma.
 - Comprobación del acceso en red desde la máquina cliente al servidor.

- Prueba de conexión a escritorio remoto desde el cliente Linux al servidor por Xnest y login en la máquina servidora.
4. Generación del sistema operativo empotrado. En este paso lo que se ha realizado es la generación del sistema empotrado para el Thin Client, actualizando la versión del kernel a una actual.
- Descarga de una versión del kernel Linux 2.4.6.
 - Configuración de los parámetros del kernel para generar una imagen genérica para i386 sobre vmware eliminando las funcionalidades innecesarias.
 - Compilación de una nueva imagen del kernel Generic-i386VMWare e instalación sobre nuestro directorio boot en el servidor, cambiando los parámetros de configuración para que el Thin Client arranque con la nueva imagen.
 - Prueba de arranque del Thin Client con la nueva versión del Kernel.
5. Generación del sistema de ficheros empotrado con soporte XDMCP. En este paso se ha generado un sistema de ficheros empotrado para el Thin Client con soporte para XDMCP, incluyendo Xnest como cliente para conexión a escritorio remoto, además se ha actualizado el sistema a las librerías y herramientas actuales.
- Actualización de la versión busybox en el sistema de ficheros.
 - Actualización de las librerías utilizadas en nuestro sistema de ficheros empotrado.
 - Actualización de drivers, binarios y ficheros de configuración relacionados con el sistema X-Window.
 - Inclusión de versión actual, acorde a las librerías actualizadas, del binario Xnest.

- Generación del fichero configuración del sistema X-Window para soporte genérico de dispositivos de E/S y optimizado para fbdev.
 - Generación del fichero de arranque de X para que llame al nuevo cliente Xnest.
 - Construcción de los nuevos módulos y drivers X-Window correspondientes a nuestro nuevo sistema actualizado en un fichero Generic-i386VMWare.sys de dispositivos.
 - Construcción del nuevo protocolo XDMCP.bin
 - Actualización del directorio boot con el nuevo protocolo, fichero de dispositivos e imagen del kernel.
 - Configuración de los parámetros para que nuestro Thin Client arranque con el nuevo kernel y sistema de ficheros empotrado.
6. Arranque Thin Client por red sobre Linux y conexión remota a Linux mediante el protocolo XDMCP.

En este punto final se han probado los dos pasos anteriores y se ha comprobado el correcto funcionamiento del sistema. El Thin Client arranca por red gracias a nuestro servidor Linux, descarga el sistema actualizado y se conecta remotamente al propio servidor Linux mediante el nuevo protocolo XDMCP.

Todos estos pasos se detallarán y explicarán en un Trabajo de Fin de Carrera en el cuál se entrará en profundidad en cada uno de los detalles del ThinOS de Lan Core y en cómo se realiza la actualización y generación de un nuevo sistema operativo empotrado, sistema de ficheros empotrado y protocolo de comunicaciones para conexión a escritorio remoto.

A continuación continuamos con un trabajo de documentación en el que se presenta la tecnología Thin Client.

Objetivo

En este documento vamos a presentar y dar a conocer la tecnología Thin Client (Clientes Ligeros) dando una visión global. Los aspectos que se pretenden mostrar son los siguientes:

- Visión general de la tecnología.
- Como se organiza su arquitectura.
- Hardware que entra en juego en la tecnología.
- Software que soporta esta tecnología.
- Ventajas y beneficios que aporta e inconvenientes.
- Escenarios de ejemplo donde se pueden implantar.
- Funcionamiento y bases de los sistemas Thin Client.
- Protocolos involucrados.
- Proyectos software libre actuales relacionados con esta tecnología.

Introducción: ¿Qué es un Thin Client?

Thin Client (Cliente ligero) es una computadora cliente en una arquitectura de red cliente-servidor que depende primariamente del servidor central para las tareas de procesamiento, y principalmente se enfoca en transportar la entrada y la salida entre el usuario y el servidor remoto, aunque normalmente el término Thin Client no se usa únicamente para denominar al cliente ligero, sino que se usa para nombrar a toda la red cliente-servidor como un sistema informático.

A lo largo del presente documento usaremos el término sistema Thin Client para denominar a toda la red como un sistema informático o como una tecnología y cliente ligero para referirnos a una computadora cliente específica del sistema Thin Client.

Visión general de la tecnología Thin Client

Al diseñar un sistema informático o una red informática, hay decisiones que tomar acerca del procesamiento, el almacenamiento, los recursos hardware, el software, la comunicación en red, el sistema operativo, la interfaz de usuario, etc.

Podemos tener diversas configuraciones o diseños de una red informática para un negocio, una biblioteca, una escuela, un cyber, una sala de juegos, un laboratorio de una facultad, etc. El diseño de este sistema informático puede venir motivado por diversos motivos, como pueden ser los requisitos funcionales, las necesidades y preferencias, las formas de uso del sistema, el presupuesto y los recursos disponibles, etc.

En un sistema Thin Client tenemos una arquitectura de red cliente-servidor donde los clientes son ligeros, el único software que es instalado en el cliente ligero es la interface de usuario, algunas aplicaciones frecuentemente usadas y un sistema operativo de red. Este software puede ser cargado de una unidad de disco local, de una unidad externa, del servidor en tiempo de arranque, o según lo que se necesite.

En el sistema Thin Client el servidor, o un cluster de servidores tienen el peso total de todas las aplicaciones, servicios, y datos. Al mantener algunos servidores ocupados y muchos clientes ligeros ligeramente cargados, se tiene una administración de sistemas y mantenimiento centralizada, más fácil y de costes más bajos, así como todas las ventajas de la computación en red: almacenamiento y respaldo centralizados y una seguridad más fácil. Debido a que los clientes ligeros son numerosos pero relativamente pasivos y de bajo mantenimiento, el sistema Thin Client entero es más simple y más fácil de instalar y operar.

Un único PC servidor puede manejar cinco o más clientes ligeros. Un PC servidor más potente puede llegar a soportar hasta cien clientes ligeros a la vez. Y un servidor de alto rendimiento puede llegar a manejar 700 clientes ligeros.

Por otro lado, desde la perspectiva del usuario, la interacción con el monitor, el teclado, y el ratón cambia poco que cuando se está usando un cliente pesado, o PC como estación de trabajo.

Los clientes ligeros son una gran inversión para las escuelas y los negocios que quieren maximizar el número de estaciones de trabajo que pueden comprar con un presupuesto, reduciendo los costes en los recursos informáticos y facilitando la administración, mantenimiento y seguridad de los mismos.

Al simplificar la carga en el cliente ligero, éste puede ser un dispositivo muy pequeño y de baja energía, con un coste de compra y de operación más bajos en cada puesto.

Una unidad simple de 60 € puede reemplazar un PC o Workstation en una escuela o un negocio. También ahorraría mucha energía a largo plazo, debido a la consumición de baja energía.

Arquitectura en Red

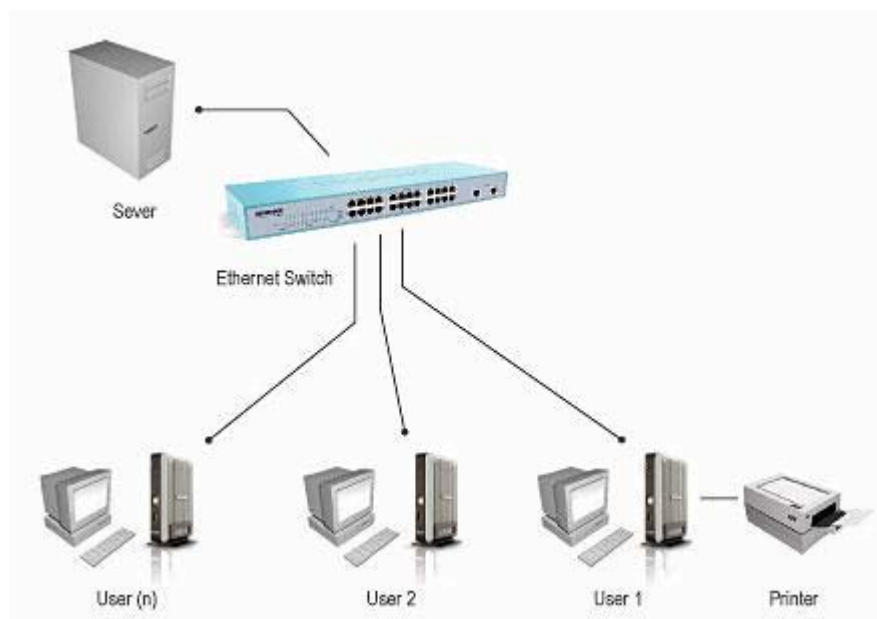
La arquitectura Thin Client como se ha comentado consta de una red típicamente LAN en la que por un lado están los clientes ligeros y por otro los servidores. Los clientes ligeros se conectan nada más arrancar al servidor y descargan su sistema operativo, el cuál les proporciona una sesión sobre el servidor.

La zona de servidor o servidores puede estar organizada de diversas formas, podemos tener un único servidor el cuál puede ser un PC o un servidor mucho más potente como un mainframe dedicado, pero también podemos organizar los servidores en cluster con balance de carga y tolerancia a caídas, teniendo una red mucho más rápida y eficiente entre los servidores, también podemos organizar los servidores y tener ciertos servidores para procesamiento, otros

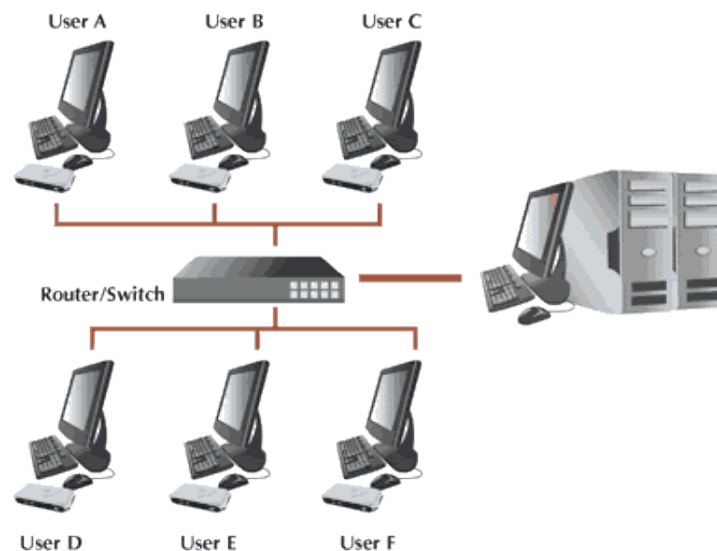
para gestionar los ficheros o el almacenamiento, otros para bases de datos, otros para servicios de correo, etc. También podemos tener los servidores en una sala dedicada para dar una mayor seguridad en la administración y mantenimiento del sistema. Es decir, la estructuración y organización de los servidores es otro punto importante a la hora de diseñar un sistema Thin Client, pero sí es necesario tener al menos un servidor que proporcione un servicio para el arranque de los clientes ligeros (si partimos de que los clientes no pueden arrancar por sí solos, ya que no disponen de unidad de almacenamiento ya sea interna o externa) y un servicio para obtener una sesión en algún servidor, que podría ser el mismo proporcionando ambos servicios.

Como vemos existen diversas configuraciones del sistema Thin Client y vamos a ver algunos ejemplos:

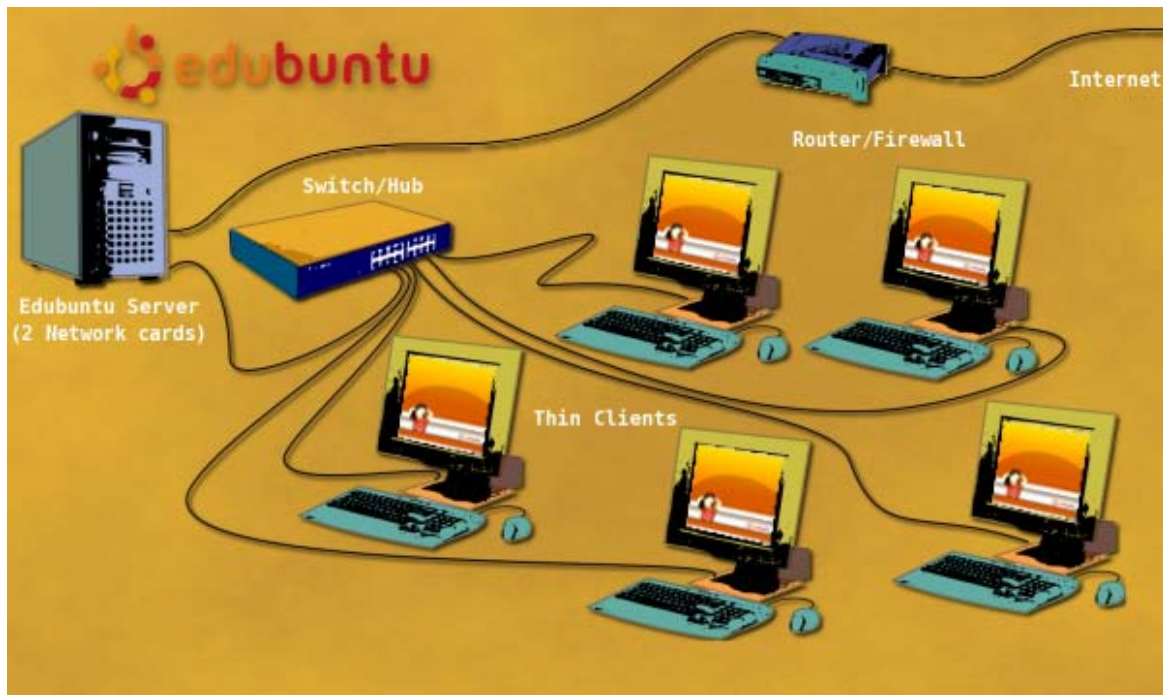
La siguiente imagen muestra una configuración típica y básica de Thin Client, en la que tenemos 3 clientes ligeros y un servidor, uno de los clientes ligeros tiene además conectada una impresora físicamente (no en red) y los clientes ligeros se conectan con el servidor en una LAN mediante un Switch Ethernet.



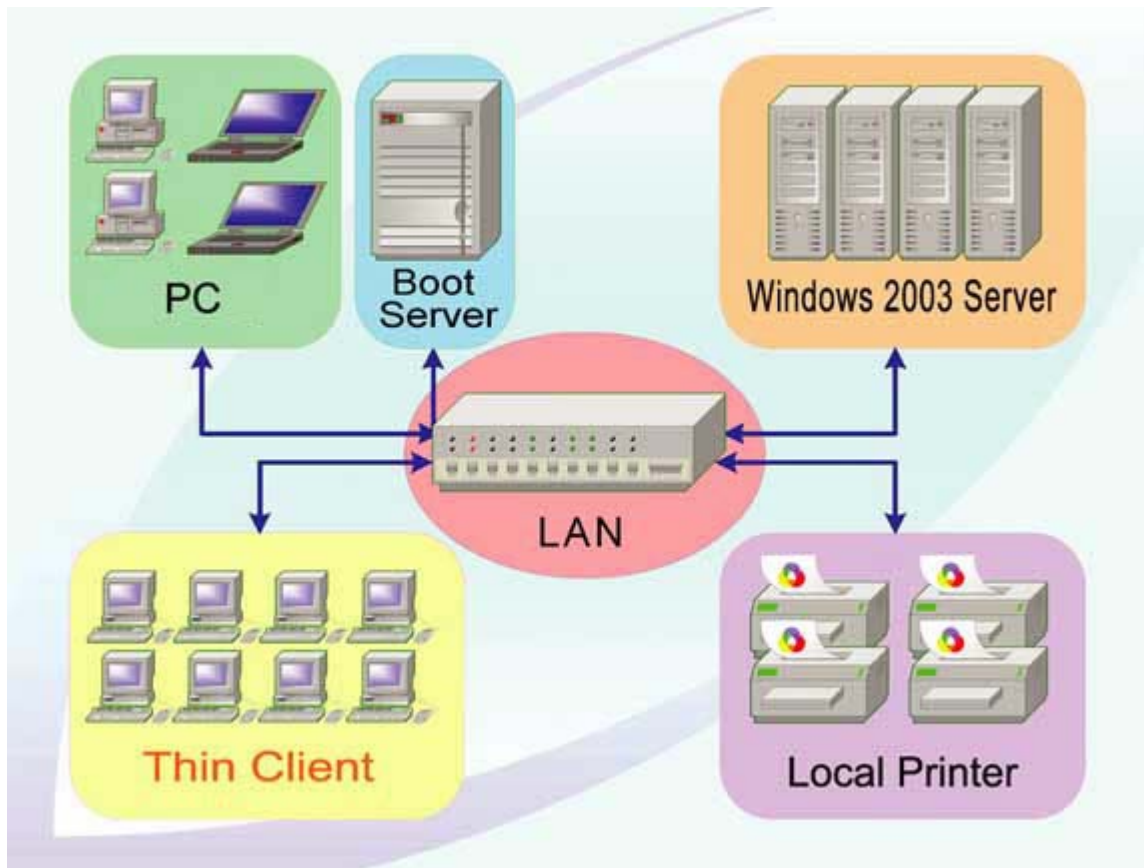
En la siguiente imagen tenemos otro ejemplo típico de Sistema Thin Client con 6 clientes ligeros conectados a un Router/Switch de una LAN que realiza el encaminamiento de paquetes de la red interna de los clientes ligeros al exterior, en este caso a dos servidores.



En la siguiente imagen tenemos una arquitectura Thin Client con acceso a internet mediante la distribución de GNU/Linux Ubuntu. Como vemos tenemos un único servidor con dos interfaces de red, por un lado está conectado a un Router que nos da acceso a internet proporcionándonos seguridad mediante un Firewall y por otro lado proporciona sesiones a los 5 Thin Clients conectados en una LAN mediante un Switch/Hub.

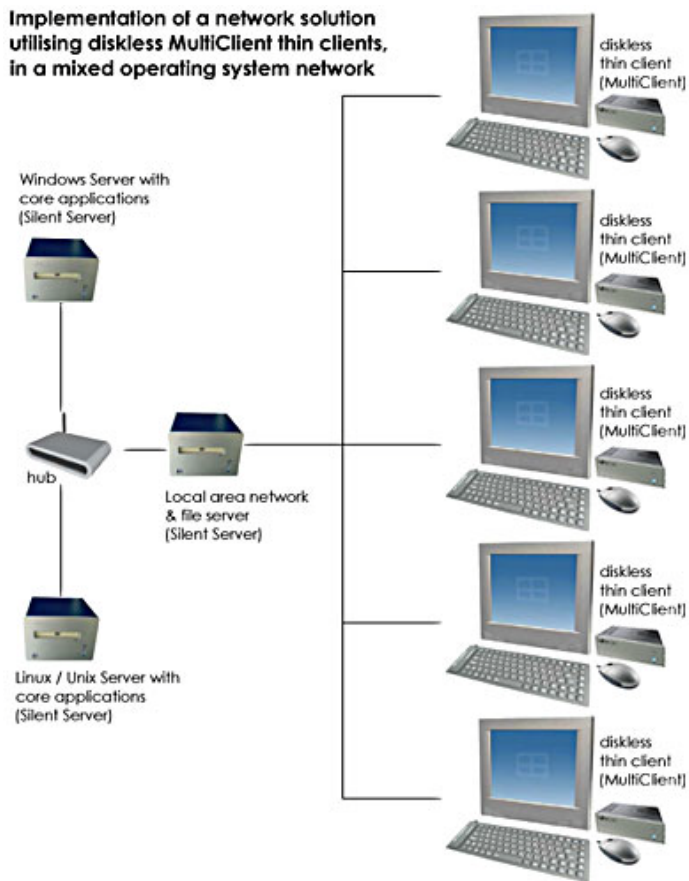


En esta otra imagen tenemos otro ejemplo de arquitectura en donde además de tener clientes ligeros, tenemos clientes pesados, los clientes pesados podrían usarse como clientes ligeros y arrancar desde el Boot Server, pero podrían también ser independientes y ser PCs o Workstation con su propio sistema operativo y aplicaciones, lo cual sería una sistema híbrido o mixto entre un sistema Thin Client y una red de área local típica. Además del Boot Server, que es el servidor que nos permite arrancar los clientes ligeros, tenemos los servidores con Windows 2003 Server que nos dan los servicios que tengamos a disposición de los usuarios, siendo uno de ellos la impresión mediante las impresoras en red, como vemos el espacio para los servidores está separado por la LAN, que por una lado tiene una interfaz o interfaces para los clientes y por otro tiene varias interfaces para los servicios en red.



En esta nueva imagen tenemos un sistema Thin Client con varios sistemas operativos. Como vemos los clientes ligeros no tienen disco y arrancan su sistema por red, tenemos un Hub que nos interconecta todos los clientes ligeros de la red y tres servidores: un servidor Windows con sus aplicaciones, un servidor Linux/Unix con sus aplicaciones y un Servidor de ficheros para el almacenamiento de datos. En este caso los clientes ligeros pueden arrancar su sesión sobre Linux o sobre Windows, dependiendo del servidor al que se conecten en el arranque.

**Implementation of a network solution
utilising diskless MultiClient thin clients,
in a mixed operating system network**



Estos son algunos ejemplos de configuración de los Thin Client en red, como vemos podemos jugar con muchos aspectos a la hora de diseñar el sistema: con los sistemas operativos, los servicios que se proporcionan, la propia configuración de la red mediante la que están conectados, la distribución lógica y física de la red, el sistema de almacenamiento de datos, la seguridad de la red, el acceso a internet, el número de clientes ligeros, la combinación con clientes pesados, los dispositivos adicionales de los clientes ligeros o hasta qué punto son más o menos ligeros.

Con esto es suficiente y no pretendemos entrar más en detalle, ya que sería otro tema a abordar si se entra en profundidad, pero se quiere reflejar la diversidad que existe a la hora de dar una implementación o solución de sistema en red usando la tecnología Thin Client.

Hardware de un Thin Client

Como hemos visto podemos usar como Hardware para un cliente ligero cualquier PC o Workstation, incluso aquellos viejos PCs que no tienen una suficiente potencia de procesamiento para ejecutar aplicaciones actuales, un portátil, un notebook, un netbook e incluso podemos usar PDAs o dispositivos móviles. Pero además de poder reutilizar dispositivos y aparatos existe hardware orientado exclusivamente a sistemas Thin Client.

De forma general, el Hardware de un cliente ligero es básico, ya que no necesita disco duro para almacenamiento, el procesador es de una potencia mucho menor que el de un PC ya que las operaciones que realiza el cliente ligero son de Entrada y Salida, es decir, recoger la entrada del teclado y del ratón y mandárselas al servidor por red y recoger por red los datos de entrada del servidor y mostrarlos en la pantalla.

Lo único que necesita un cliente ligero es una CPU, una memoria RAM, una memoria FLASH para software interno de configuración del cliente ligero, un chipset gráfico con su memoria que nos permita realizar las operaciones gráficas y mostrarlas en pantallas con diferentes resoluciones, una tarjeta de red Ethernet para conectar el dispositivo a la LAN, una fuente de alimentación mínima y las conexiones de entrada y salida necesarias que podrían ser: dos puertos PS/2 uno para teclado y otro para ratón, un conector RJ45 sobre la Ethernet para conectar nuestro cable de red, conectando el dispositivo a la LAN, una salida de video VGA de la tarjeta gráfica para conectar nuestro cliente a un monitor y la conexión de la fuente para suministrar la energía de la red eléctrica. Con este hardware tenemos un cliente ligero, opcionalmente puede tener indicadores LEDs de Power y del adaptador de red.

En la imagen vemos un cliente ligero con estas características



Como vemos en la siguiente figura el cliente ligero es muy pequeño comparado con un PC.



No necesariamente tenemos que disponer únicamente de estas funcionalidades Hardware, el cliente ligero puede llevar por ejemplo conexiones USB para llevarnos datos con los que hemos trabajado o traer datos a nuestra cuenta en el servidor, o una tarjeta de sonido integrada como el que vemos a continuación en la siguiente imagen con entradas y salidas de audio, tanto para reproducción como para grabación, puede llevar un adaptador de red WIFI para conectarlo en una red wireless, puertos paralelo para conectar impresoras antiguas, etc.

Aunque realmente con la entrada del estándar usb y la gran variedad de dispositivos existentes en el mercado que soportan esta conexión, los puertos serie y paralelo van desapareciendo.

En la siguiente imagen vemos un cliente ligero con algunas características más que el anterior como son puertos usb y tarjeta de sonido con entrada y salida de audio, comparado con una típica torre de PC donde se ve claramente la gran diferencia de tamaño.



Aquí tenemos un ejemplo de cliente ligero con las interfaces de Entrada y Salida conectadas en un puesto de trabajo de una biblioteca, vemos que el cliente ligero podría ocupar tanto como un pequeño router/switch de uso doméstico.



En la siguiente imagen vamos a mostrar otro ejemplo de cliente ligero con sus especificaciones técnica, se trata de un Linux Embedded Thin Client - DLW Term 1222 diskless



Operating Software:

- . Linux Embedded (Kernel 2.6)
- . rdesktop 1.5.0, Citrix ICA 10.6, X1(XDMCP), VNC, NX Client, Tarantella, Go-Global Client
- . Sun Java 1.4.2, Macromedia Flash Player 7.0, Acrobat PDF Reader 5.09, Real Player 10.0
- . TCP/IP with DNS, WINS, PPP, DHCP, and PPPoE, PXE, RAS, VPN, WLAN, SNMP Agent and SNMP MIB II, SNTP, SSH
- . NFS File System Support, USB External Storage Support and Sharing, LPR/LPD Print Support, Samba Print Support, Thin

	<ul style="list-style-type: none"> Print Support, COM Port Redirect . Password-Protected Setup, USB Security Manager . Compressed Read-only File System, Desktop Mode
Terminal Emulation:	<ul style="list-style-type: none"> . ASCII/ANSI Emulation . SCO Console ANSI . DEC VT420/320/220/100/52 . Wyse 325/120/60/50+ . TVI 925/910, ADDS A2 . IBM Emulation . TN5250(E), TN3270(E)
Server Operating System Support:	<ul style="list-style-type: none"> . RDP: <ul style="list-style-type: none"> . Windows 2003 Server Family + RDP service . Windows XP Pro + RDP service . Windows 2000 Server Family . Windows NT Server 4.0 + TSE . ICA: <ul style="list-style-type: none"> . Citrix MetaFrame Presentation Server . Citrix MetaFrame XP + Windows 2003 Server Family . Citrix MetaFrame XP + Windows 2000 Server Family . Citrix MetaFrame XP + Windows NT Server 4.0 + TSE . Windows NT Server 4.0 + TSE
CPU:	<ul style="list-style-type: none"> . 1 GHz Via Eden V4 Processor (Fanless, x86 compatible)
Memory:	<ul style="list-style-type: none"> . 256MB DDR2 standard . 1GB DDR2 max
Compact Flash Memory:	<ul style="list-style-type: none"> . 256MB
Chipset:	<ul style="list-style-type: none"> . CN700 . VT8237R+
Display:	<ul style="list-style-type: none"> . Integrated AGP 8x Graphics Core . Shared System memory 8MB to 64MB . Maximum Resolution: 1680 x 1050, 60 Hz refresh, 24-bit high color <ul style="list-style-type: none"> . 800 x 600 . 1024 x 768 . 1280 x 1024 . 1600 x 1200 . Widescreen Resolutions: <ul style="list-style-type: none"> . 1280 x 800 . 1440 x 900 . 1680 x 1050
Networking:	<ul style="list-style-type: none"> . 10/100 Base-T Realtek 8100C Fast Ethernet
I/O Connectors:	<ul style="list-style-type: none"> . Winbond W83697HF (LPC), (1) Line-Out, (1) Mic-In . (1) Serial port, RS-232C compatible, DB9M connector, baud up to 115.2K bps . (1) Parallel port, DB25F connector, bi-directional centronics-compatible and locally printable . Analog VGA video output, DB-15 connector . (4) USB 2.0 ports (two at front, two at rear) for floppy disk drive, mouse, CD-ROM, hard-drive, etc. . (2) PS/2 ports for mouse and keyboard . RTL 8100C+ 10/100 Mbps, one RJ-45 connector
LED Indicators:	<ul style="list-style-type: none"> . (1) Power indicator (On front bezel) . (2) Network adapter indicators (On rear panel)
Warranty:	<ul style="list-style-type: none"> . One year from date of invoice.
Power Supply:	<ul style="list-style-type: none"> . External, autosensing . Input: 100-240V @ 50~60Hz
Dimensions (L X W X H):	<ul style="list-style-type: none"> . 7.5" x 5.5" x 1.6", 190mm x 140mm x 40mm
Weight:	<ul style="list-style-type: none"> . Unit 1.5 lbs, .7 kgs

	. Shipping 4.0 lbs, 1.7 kgs
Regulatory Compliance:	. Emissions FCC Class B, TUV, UL/CUL, CE, BSMI, C-TICK, CB, VCCI
Temperature:	. Operating: 41F~104F (5C~40C) . Storage: -40F~140F (-40C~60C)
Humidity:	. 20% to 80% non-condensing . Operating altitude range from 0 to 10,000 feet (0 to 3050 meters)

Este es un cliente ligero con muchas más características y mayor calidad que los anteriores y es un ejemplo de lo que nos podemos encontrar en el mercado actualmente.

Por último otra imagen de ejemplo de cliente ligero en la que el propio aparato lo tenemos acoplado en la parte trasera de un monitor TFT Vesa para ahorrar más espacio en el escritorio, en este caso además de conexiones USB y tarjeta de sonido integrada, soporta conexión WIFI y tiene soporte antirrobo.



Como vemos en los clientes ligeros tenemos también diversas opciones en lo que se refiere a procesamiento, memoria, chipset gráfico y las entradas y salidas que soporta, el sistema operativo empotrado y el software preinstalado.

Existen diversas empresas fabricantes y distribuidores de tecnología Thin Client, a continuación presentamos una lista de algunos de ellos para tenerlos como referencia:

- ChipPC
- Devon IT
- LISCON
- AXUS Microsystems Inc.
- Cute-Box.com.
- Fujitsu Technology Solutions.
- ICOP Technology Inc.
- IEI Technology Corp.
- Termtek Computer Co.
- EPATec.
- 10ZiG Technology.
- HP (NeoWare).
- IBM.
- IGEL Technology.
- Sun Microsystems.
- Wyse Technology.

Software Thin Client

Como hemos visto el software del cliente ligero es mínimo ya que el procesamiento se ejecuta en el servidor. El software del cliente ligero tiene dos servicios, aunque sólo uno de los dos es estrictamente necesario en la tecnología Thin Client.

El primer servicio es el arranque por red y el segundo es la conexión con un escritorio remoto. El primer servicio es opcional ya que el cliente ligero puede incorporar en su hardware un disco duro, o una unidad de arranque externa como puede ser una unidad de disquete, de cd/dvd o usb, aunque normalmente no es así. Normalmente el cliente ligero es diskless, no tiene disco duro local y no tiene unidades de almacenamiento externas, en este caso el cliente ligero necesita arrancar por red, ya que no dispone de software.

Tanto en un caso como en otro el software que necesita el cliente ligero para arrancar es un cargador de arranque del sistema operativo, una imagen de un sistema operativo empujado que manejará el sistema y los protocolos de comunicación en red del cliente ligero y un sistema de ficheros con el que trabajar donde tendrá los servicios de configuración y arranque del cliente ligero, los clientes de acceso a escritorio remoto y opcionalmente alguna aplicación.

En el caso de que el cliente ligero pueda arrancar por una unidad de almacenamiento, todo este software se encuentra en esa unidad, por ejemplo si el cliente ligero dispone de disco duro local, todo este software estará almacenado en el disco, pero en el caso de que no disponga de estas unidades que es lo habitual en un hardware de cliente ligero, el cliente ligero necesita obtener este software por red desde un servidor y esto lo consigue arrancando por red.

Nótese que el software del cliente ligero puede variar y ser mucho más extenso y estar almacenado en un disco duro local, pero esta idea se acerca más a la de cliente pesado que a la tecnología Thin Client. En la tecnología Thin Client contamos con que los clientes ligeros pueden no tener unidad de disco local y que el sistema operativo y el sistema de ficheros será cargado en la memoria del propio cliente ligero, es por ello que tanto el sistema operativo y el sistema de ficheros deben ocupar un espacio mínimo por esto utilizamos el término sistema operativo empotrado. Otra posibilidad en cuanto al sistema de ficheros es montarlo por red, estando el sistema de ficheros del cliente ligero en un espacio del servidor.

Vamos a citar las necesidades software del cliente ligero con una breve descripción:

- Software BIOS: Este software está en una memoria ROM propia del cliente ligero y nos permite configurar algunos parámetros de los dispositivos hardware del cliente así como la unidad de arranque local o el arranque por red.
- Software EPROM: Este software está incluido en una memoria en el adaptador de red del cliente ligero y nos permite arrancar por red. Como hemos visto no es estrictamente necesario, si el cliente ligero dispone de unidades internas o externas que se puedan conectar físicamente, donde esté almacenado el software que describimos a continuación.
- Cargador de arranque: Es el cargador de arranque del sistema operativo del cliente ligero en el caso de que el cliente ligero arranque por red, el cargador deberá proporcionar también la carga del sistema operativo por red.

- Sistema Operativo Empotrado: Es el sistema operativo del cliente ligero que se cargará en su memoria RAM al arrancar. Nos permite inicializar el sistema, proporciona protocolos de comunicaciones del cliente ligero en red, controladores para los dispositivos del cliente ligero (chipset gráfico, adaptador de red, manejo de puertos de entrada y salida como PS2, USB, puerto serie, etc.), servicios para la ejecución de programas, gestión de procesos, manejo del sistema de ficheros, gestión de la memoria del cliente, etc. Es decir, proporciona los servicios de un sistema operativo, pero se trata de un sistema operativo empotrado, creado específicamente para el propio cliente ligero sobre el que se ejecuta para ocupar un tamaño mínimo.
- Sistema de ficheros empotrado: Es el sistema de ficheros del cliente ligero, podrá cargarse en la RAM del cliente o acceder a él como un sistema de ficheros remoto que está en el servidor en el caso de que el cliente, como es habitual, no tenga unidades de almacenamiento. Contiene scripts de inicialización y configuración del sistema y opcionalmente programas y aplicaciones básicas. Estas aplicaciones pueden ser de cualquier tipo, desde emulación de terminales hasta navegadores web. Todo este conjunto debe ser mínimo ya que estamos en un cliente ligero, si tuviésemos un sistema de ficheros lleno de aplicaciones estaríamos más cerca de lo que es un cliente pesado.
- Clientes de acceso a escritorio remoto: Son los programas clientes que nos permiten realizar una conexión de escritorio remoto con el servidor. El cliente ligero puede disponer de distintos clientes de acceso a escritorio remoto con el fin de posibilitar la conexión con distintos sistemas operativos servidor, ya que algunas conexiones de escritorio remoto son dependientes del sistema operativo, por ejemplo Linux no dispone del protocolo RDP de Windows, por lo tanto, si el cliente ligero sólo dispone de este cliente de conexión, no va a poder acceder a servidores Linux.

Fundamentos Thin Client

Proceso de arranque

En el proceso de arranque el cargador va a cargarnos el sistema operativo en memoria y una vez haya realizado esta operación cederá el control al sistema operativo detectará los dispositivos, cargará los módulos necesarios para el cliente ligero, montará un sistema de ficheros e inicializará el sistema, finalmente creará un proceso de inicio que comenzará a ejecutar la configuración del sistema y posteriormente se mostrará el cliente de acceso a escritorio remoto, una vez llegados a este punto el cliente ligero ha terminado su proceso de arranque.

El servicio de escritorio remoto será un cliente que nos permitirá conectarnos con el servidor remoto mediante login en el servidor, inicializando así una sesión en el servidor. Este servicio sí es obligatorio en la tecnología Thin Client independientemente de la forma en que haya arrancado, ya sea por red, desde un disco duro local o desde una unidad externa de almacenamiento, una vez obtenida la sesión en el servidor, toda la ejecución se realiza en el servidor aunque la realizamos desde el cliente ligero.

A continuación vamos a ver con más detalle, los dos servicios que son el fundamento de la tecnología Thin Client.

Arranque por red

Para que un cliente ligero pueda arrancar por red, necesita estar conectado físicamente a una red LAN y tener configurado en su setup interno (BIOS) esta modalidad de arranque por red (el adaptador de red también necesariamente debe soportar esta forma de arranque) en lugar del típico arranque mediante

disco duro, de tal manera que el cliente ligero lo primero que hace cuando lo encendemos es arrancar por red mediante el protocolo PXE que consta de dos protocolos que veremos en detalle más adelante, que son DHCP y TFTP.

El arranque por red consiste en que el cliente ligero posee una rutina de arranque en una memoria permanente que incluye en el propio hardware, que le permite contactar con el servidor y obtener así el sistema operativo empotrado con el que continuar para enlazar con un sistema de ficheros y conectar remotamente con el servidor.

Para poder arrancar por red, el cliente ligero debe obtener:

1. Una identidad
2. La imagen de un sistema operativo.
3. Un sistema de ficheros con el que trabajar.

Supongamos un cliente ligero sin disco que tiene una ROM para el arranque por red. Este cliente podría ser uno entre varios idénticos. ¿Cómo podemos distinguir este del resto? Existe una información que es única para este cliente ligero (en realidad para su adaptador de red) y se trata de su dirección física Ethernet. Cualquier adaptador de Ethernet en el mundo posee una dirección Ethernet de 48 bits exclusiva para él, porque a todo fabricante de hardware Ethernet, le han sido asignados bloques de direcciones. Por convención estos bloques de direcciones se representan en forma de dígitos hexadecimales separados por dos puntos en grupos de dos; por ejemplo 00:60:08:C7:A3:D8.

Los protocolos usados para obtener una dirección IP, dada una dirección Ethernet, son llamados Boot Protocol (BOOTP) y Dynamic Host Configuration Protocol (DHCP). DHCP es una evolución del BOOTP. En realidad BOOTP y DHCP no solo se encargan de traducir direcciones Ethernet, en previsión los

diseñadores de BOOTP y DHCP dejaron la capacidad para poder trabajar con cualquier tipo de dirección hardware, sin embargo Ethernet es la más extendida.

Un ejemplo de intercambio DHCP sería:

Cliente Ligero: Hola, mi dirección hardware es 00:60:08:C7:A3:D8, por favor, dame mi dirección IP.

Servidor DHCP: (Busca la dirección en su base de datos). Tu nombre es *pepito* tu dirección IP es 192.168.1.32, tu servidor es 192.168.1.1 y el fichero del que se supone que debes arrancar es /tftpboot/vmlinux (y alguna información adicional).

En la primera fase cabe la duda de ¿cómo encontró el cliente ligero la dirección del servidor de BOOTP o DHCP? La respuesta es que no lo hizo, ya que realizó la petición en forma de multidifusión (broadcast) dentro de la red local a la que está conectado, de forma que cualquier servidor DHCP que pudiera responder la petición, lo haría.

Después de obtener la dirección IP, es decir, su identidad, debe conseguir la imagen del sistema operativo y lanzarlo a ejecución. En esta fase, se usa otro protocolo de Internet, conocida como Trivial File Transfer Protocol (TFTP). Éste es una versión reducida del famoso FTP: no contempla autenticación, trabaja a través del User Datagram Protocol (UDP) en vez de Transmission Control Protocol (TCP). Se prefirió usar UDP a TCP por simplicidad. La implementación de UDP en el cliente ligero puede ser suficientemente reducida como para caber en una ROM. Ya que UDP es un protocolo orientado a la transmisión por bloques, en oposición a la transmisión por cadenas, la transferencia se realiza bloque a bloque de la siguiente forma:

Cliente ligero: Dame el bloque 1 de /tftpboot/vmlinux

Servidor TFTP: Aquí lo tienes.

Cliente ligero: Dame el bloque 2.

Y así en adelante, hasta que se transfiera el fichero completo. El funcionamiento consiste básicamente en el reconocimiento de cada bloque, y la pérdida de paquetes se soluciona mediante su retransmisión al cabo de un tiempo establecido. Cuando todos los bloques han sido recibidos, la ROM de arranque de la red pasa el control a la imagen del sistema operativo.

Pero en realidad lo que se descarga no es la imagen del sistema operativo, sino el cargador de arranque del sistema operativo. Es el cargador quién se descarga la imagen del sistema operativo y además el sistema de ficheros remoto, una vez obtenida la imagen y el sistema de ficheros los carga en la memoria RAM del cliente ligero y cede el control al sistema operativo.

Por otro lado, en el servidor que nos proporciona el arranque necesitamos los servicios correspondientes para dar servicio a los clientes ligeros que quieren arrancar, como hemos visto necesitamos:

1. Un servidor BOOTP o DHCP para dar identidad a los clientes en la red.
2. Un servidor TFTP en el servidor que permita al cliente la descarga de su sistema operativo.
3. Un cargador de arranque del sistema operativo del cliente.
4. Una imagen del sistema operativo empujado que se va a proporcionar a los clientes, compilado previamente específicamente para sus necesidades.
6. Un sistema de ficheros proporcionado para el cliente, este sistema de ficheros debe contener la estructura de directorios con los ficheros del cliente ligero.

Sesión remota

Existen diversas formas de establecer una conexión remota con el servidor para obtener una sesión. En función de las necesidades que queramos podemos establecer una sesión remota gráfica o una sesión remota en modo texto.

Sesión en modo texto

Podemos iniciar una sesión remota en modo texto mediante telnet, de esta manera el cliente ligero trabajaría como un terminal en modo texto, en el servidor proporcionaríamos un servidor de telnet para que esto fuera posible. De esta manera trabajaríamos en el cliente ligero desde la línea de comandos, el protocolo de comunicación a nivel de aplicación entre el cliente ligero y el servidor sería **Telnet** (TELEcommunication NETwork). Telnet sólo sirve para acceder en modo terminal, es decir, sin gráficos y además su mayor problema es la seguridad, ya que los datos no van cifrados en la red. Todos los nombres de usuario y contraseñas necesarias para entrar en la máquina servidora viajan por la red como texto plano (cadenas de texto sin cifrar). Esto facilita que cualquiera que espíe el tráfico de la red pueda obtener los nombres de usuario y contraseñas. También carece de un esquema de autenticación que permita asegurar que la comunicación esté siendo realizada entre los anfitriones deseados, y no interceptada entre ellos. De todas formas, a pesar de de sus graves problemas, lo citamos como una posibilidad, ya que se ha usado durante años, también cabe citar que el uso de telnet nos da la posibilidad de tener un cliente ligero con un chipset gráfico mínimo, ya que no se manejan operaciones gráficas complejas y las necesidades de red también son mínimas.

Una alternativa a telnet en modo texto, que no nos consume muchos más recursos y que nos da seguridad en nuestra conexión remota es **SSH** (Secure Shell), que nos provee de toda la funcionalidad presente en telnet, nos da la adición de un cifrado fuerte para evitar que los datos sensibles tales como

contraseñas sean interceptados, y la autenticación mediante llave pública, para asegurarse de que el computador remoto es realmente quién dice ser.

Realmente SSH es un protocolo además del programa que implementa el intérprete de órdenes y nos permite redirigir el tráfico de X para poder ejecutar programas gráficos si tenemos un Servidor X, nos permite copiar datos de forma segura, gestionar claves RSA para no escribir claves al conectar, etc.

La diferencia principal entre SSH y Telnet es que SSH usa técnicas de cifrado que hacen que la información que viaja por el medio de comunicación vaya de manera no legible y ninguna tercera persona pueda descubrir el usuario y contraseña de la conexión ni lo que se escribe durante toda la sesión, aunque es posible atacar este tipo de sistemas por medio de ataques de REPLAY y manipular así la información entre destinos.

Cabe destacar aquí, que una sesión remota está más referida al concepto de terminal que al de Thin Client. El concepto de terminal o una red formada por terminales son las redes que se usaban en los años 70 donde las capacidades de red eran menores que las actuales que soportan 100 Mbps o incluso 1 Gbps, con las redes actuales se pueden soportar conexiones de escritorio remoto que es el tipo de sesión que caracteriza realmente la tecnología Thin Client, y se diferencia de un terminal en que el terminal tiene un procesamiento de mucha menor carga que la que realiza un Thin Client, que realiza complejas operaciones gráficas. No obstante, hemos señalado este tipo de conexión modo texto como un paso previo que se realizó años atrás, a la nueva tecnología Thin Client.

Sesión en modo gráfico (Sesión de escritorio remoto)

La siguiente opción es que nuestro cliente ligero arranque una sesión gráfica sobre el servidor, también aquí tenemos varias opciones. A continuación vamos a citar algunas.

Mediante el protocolo XDCMP (X Display Manager Control Protocol, protocolo de Control de Administrador de Pantalla X) es un protocolo utilizado en redes para comunicar una máquina servidor que ejecuta un sistema operativo con un gestor de ventanas basado en X-Window con el resto de clientes que se conectarán a éste con propósitos interactivos. Este sistema es usado en máquinas Linux/Unix.

Mediante VNC (Virtual Network Computing, computación en red virtual), VNC es un programa de software libre basado en estructura cliente-servidor el cual nos permite tomar el control del ordenador servidor remotamente a través de un ordenador cliente. VNC permite que el sistema operativo en cada computadora sea distinto, es posible compartir la pantalla de una máquina de “cualquier” sistema operativo conectando desde cualquier otro ordenador o dispositivo que disponga de un cliente VNC portado.

Tecnología NX. NX realiza conexiones remotas X11 muy rápidas, lo que permite a los usuarios acceder a escritorios remotos de Linux o Unix incluso bajo conexiones lentas como las realizadas por módem. NX realiza una compresión directa del protocolo X11, lo que permite una mayor eficiencia que VNC, además la información se envía por SSH, por lo que toda la información que se intercambian servidor y cliente está cifrada, por lo tanto la compresión nos da una mayor velocidad y disminución de tráfico en la red y el cifrado nos aporta seguridad.

Existen más protocolos de conexión a escritorio remoto como RDP de Microsoft Windows, pero no vamos a citar más de momento, ya que más adelante entraremos en profundidad en cada uno de ellos.

A continuación se va a presentar en pasos un ejemplo de lo que podría ser el arranque de un cliente ligero mediante Linux, este ejemplo es un caso particular y no tiene porqué ser así, ya que existen varias posibilidades como hemos visto en cuanto a protocolos. En concreto, en este caso se está montando una imagen inicial en memoria mediante initramfs (sistema de ficheros sobre la memoria RAM) y se usa LTSP (Linux Terminal Server Project) un proyecto Linux para el arranque de terminales desde el servidor, se utiliza XDCMP como protocolo de comunicación para realizar la conexión al escritorio remoto.

Ejemplo del proceso de arranque de un cliente ligero sobre Linux:

1. Cargar el kernel de Linux en la memoria del cliente ligero. Esto puede hacerse de distintas maneras, entre ellas: PXE, gPXE, Etherboot.
2. En la mayoría de los casos usando una tarjeta de red con PXE para arrancar es la forma más rápida y sencilla de hacerlo.
3. Una vez que el kernel ha sido cargado en la memoria, comenzará la ejecución.
4. El núcleo inicializa todo el sistema y los periféricos que reconozca.
5. Durante el proceso de carga del núcleo, una imagen initramfs también será cargada en la memoria.
6. Normalmente, cuando el núcleo ha terminado de arrancar, se pondrá en marcha el nuevo task launcher upstart, que se encargará de la puesta en marcha de un servidor o una estación de trabajo. Pero, en este caso, hemos instruido al kernel para que cargue un pequeño shell script en su lugar. Este script se llama /init y reside en la raíz de initramfs.
7. El script /init empieza por montar /proc y /sys, arranca udev para descubrir e inicializar el hardware, especialmente la tarjeta de red, que es necesaria en todo aspecto para arrancar de aquí en más. También, crea un pequeño disco en RAM, donde se puede guardar cualquier cosa que se necesite, por ejemplo, para configurar el archivo xorg.conf.

8. Se configura la interfaz de loopback de la red. Esta es la interfaz de la red que tiene 127.0.0.1 como dirección IP.
9. Se corre un cliente DHCP llamado ipconfig para hacer otra consulta al servidor DHCP. Esta nueva consulta obtiene información tal como el nfs root server, el default gateway y otros parámetros importantes del sistema del archivo dhcp.conf
10. Cuando ipconfig obtiene una respuesta del servidor, la información que se recibe se usa para configurar la interfaz Ethernet y para determinar desde que servidor se va a montar el root.
11. Hasta este momento el sistema de directorios ha sido un disco ram. Ahora el script /init montará un nuevo sistema de directorios via NBD o NFS. En el caso que sea via NBD, la imagen que será cargada será generalmente /opt/ltsp/images/i386.img. Si la raíz es montada via NFS, entonces el directorio que será exportado desde el servidor es típicamente /opt/ltsp/i386. No es posible montar el nuevo sistema de directorios como /i. Primero debe ser montado a un directorio separado. Luego hará un run-init, que cambiará el sistema de directorios raíz actual por un nuevo sistema de directorios. Cuando esta etapa se completa, el sistema de directorios quedará montado en /. Cualquier directorio que necesite ser creado o escrito para un arranque normal, como por ejemplo /tmp, o /var, son montados en este momento.
12. Una vez que el montado del nuevo sistema de directorios fue finalizado, se ha terminado con el script /init y se necesita llamar al verdadero programa /sbin/init.
13. El programa init va a leer el directorio /etc/event.d y comenzará a configurar el ambiente del cliente ligero. A partir de aquí se ejecutará el primer comando de /etc/rcS.d.
14. El primer comando S32ltsp-client-setup configurará muchos aspectos del ambiente del cliente ligero, como por ejemplo chequear si los dispositivos locales necesitan ser arrancados, cargar algún módulo específico, etc.
15. A continuación el programa init va a comenzar a ejecutar comandos en el directorio /etc/rc2.d
16. Uno de los items en el directorio /etc/rc2.d es el comando S20ltsp-client-core que estará ejecutándose mientras el cliente ligero está arrancando.
17. El archivo lts.conf será parseado, y todos los parámetros en ese archivo que hacen referencia a ese cliente ligero serán definidos como variables de ambiente para ser usadas por el script S20ltsp-client-core.

18. Si el sonido está configurado el demonio pulseaudio se activará permitiendo conexiones remotas de audio desde el servidor para tocar en el cliente.
19. Si el cliente soporta dispositivos locales, el programa ltspfsd es ejecutado permitiendo al servidor leer de dispositivos como "memory sticks" o CDs mapeados con el cliente ligero.
20. En este momento alguna de las sesiones definidas en el archivo lts.conf será ejecutada. Las sesiones por pantalla es lo que se va a ver en todas las pantallas de las terminales. Éstas son las pantallas virtuales estándares que tienen las distribuciones de GNU/Linux, por ejemplo de Alt-F1, hasta Alt-F10. Por defecto va a correr en la pantalla 1 un "getty" estándar basado en caracteres (SCREEN_01 en el archivo lts.conf). También, si no se especifica nada en el archivo lts.conf, un script de pantalla ldm va a ser ejecutado en SCREEN_07. El LTSP Display Manager (ldm) es el administrador de login por defecto para LTSP.
21. Si SCREEN_07 está cargada con un valor ldm, o startx, entonces el sistema X Windows será ejecutado brindándole una interfaz gráfica al usuario. Por defecto el servidor Xorg va a testear la tarjeta, va a crear un archivo /etc/X11/xorg.conf con valores por defecto en el disco ram de la terminal y va a comenzar xorg con esa configuración.
22. El servidor X va a crear un túnel cifrado ssh hacia el servidor LTSP (en caso de ldm) o una consulta XDMCP (en el caso de startx). Para cualquiera de los dos casos va a aparecer una caja de login en la terminal.
23. En este momento el usuario puede loguearse. Cuando lo haga obtendrá una sesión en el servidor.

Esto puede confundir al principio ya que estamos sentados en un cliente ligero, pero se está ejecutando una sesión en el servidor. Todos los comandos que se ejecutan se ejecutan en el servidor, pero la salida se mostrará en el cliente ligero.

Como introducción al Software Thin Client vamos a parar aquí ya que la idea inicial es mostrar una breve descripción del software y del proceso de arranque del cliente ligero. Más adelante entraremos en profundidad en los protocolos que intervienen en la comunicación.

Ventajas de los Thin Client

- **Menores costes administrativos de IT.** Los clientes ligeros son controlados prácticamente en el servidor. El hardware tiene menos lugares donde puede fallar, el entorno local es altamente restringido, y el cliente es más simple y a menudo carece de almacenamiento permanente, proporcionando protección contra el malware.
- **Información centralizada.** Como la información se encuentra en un solo lugar facilita la realización de backups y evita que se guarden archivos que no sean del negocio.
- **Más fácil de asegurar.** Los clientes ligeros pueden ser diseñados de modo que ni siquiera los datos de aplicación residan en el cliente, centralizando la protección contra el malware y reduciendo los riesgos de robo de los datos físicos.
- **Seguridad de datos mejorada.** Si un dispositivo del cliente ligero sufre una seria desgracia o accidente de trabajo, no se perderá ningún dato, puesto que residen en el servidor de terminales y no en el dispositivo de punto de operación.
- **Más bajos costes de hardware.** El hardware del cliente ligero es generalmente más barato porque no contiene disco duro, memoria de aplicaciones, o un procesador poderoso. Generalmente también tienen un período más largo antes de requerir una mejora o llegar a ser obsoletos. Hay menos piezas móviles y uno actualiza o mejora el servidor y la red en lugar de los clientes, porque la limitación en desempeño es la resolución de pantalla que tiene un ciclo de vida muy largo. Muchos clientes pesados son reemplazados después de 3 años para evitar fallos del hardware en servicio y para usar el último software, mientras que los clientes ligeros pueden hacer la misma tarea de desplegar imágenes

durante muchos años. Los requisitos totales de hardware para un sistema cliente ligero (incluyendo tanto servidores como clientes) son generalmente mucho más bajos comparados a un sistema con clientes pesados. Una razón de esto es que el hardware es mejor utilizado. Una CPU en una estación de trabajo pesada está ociosa la mayor parte del tiempo. Con los clientes ligeros, los ciclos del CPU son compartidos. Si varios usuarios están ejecutando la misma aplicación, solo necesita ser cargada una sola vez en un servidor central (si la aplicación está programada para soportar esta capacidad). Con los clientes pesados, cada estación de trabajo debe tener en memoria su propia copia del programa.

- **Más bajos costes de software.** Al tener un sistema Thin Client, el software no está duplicado a diferencia de una red típica donde cada PC tiene una réplica del sistema operativo y de las aplicaciones. En un sistema Thin Client este software lo tiene el servidores o servidores, con lo que el coste de las licencias de este software es notablemente inferior, ya que gracias a esta tecnología podemos utilizar este software. La inversión de una empresa en esta tecnología reduce notablemente los costes en software.

- **Menos consumo de energía.** El hardware dedicado del cliente ligero tiene mucho más bajo consumo de energía que los típicos PC de clientes pesados, ahorran hasta un 80% de electricidad y cuidan el medio ambiente. Esto no sólo reduce los costes de energía en los sistemas de computación, en algunos casos puede significar que los sistemas de aire acondicionado no son requeridos o no necesitan ser actualizados lo que puede ser un ahorro de costes significativos y contribuir a alcanzar los objetivos en ahorro de energía. Sin embargo, se necesitan servidores y sistemas de comunicaciones más potentes.

- **Una más fácil gerencia de fallos de hardware.** Si un cliente ligero falla, un reemplazo por otro cliente es suficiente mientras es reparado y el usuario no será incomodado porque sus datos no están en el cliente.

- **Operable en ambientes hostiles.** La mayoría de los clientes ligeros no tienen piezas móviles así que pueden ser usados en ambientes polvorientos sin la preocupación que puede haber con la obstrucción de los ventiladores de los PC que puede recalentarlos y quemarlos.

- **Vale menos en caso de robo.** El hardware del cliente ligero, ya sea si es dedicado o un simple hardware viejo que ha sido reorientado, es menos útil fuera del sistema Thin Client. Alguien que robe un cliente ligero, se lleva algo menos valioso fuera del sistema Thin Client y menos valioso que un PC workstation típico.

- **Menos ancho de banda de la red.** Puesto que los servidores de terminales típicamente residen en la misma espina dorsal de red de alta velocidad que los servidores de archivo, la mayor parte del tráfico de red está confinado al cuarto del servidor. En un ambiente de cliente pesado si se abre un documento de 10MB, estos 10MB son transferidos del servidor de archivos a su PC. Cuando usted lo guarda, hay otra transferencia de 10MB de su PC al servidor. Cuando usted lo imprime sucede lo mismo otra vez, otros 10MB son transferidos sobre la red a su servidor de impresión y entonces 10MB hacia la impresora. Esto es altamente ineficiente. En un sistema de cliente ligero solamente los movimientos del ratón, las pulsaciones del teclado y las actualizaciones de la pantalla son transmitidas desde/hasta el usuario final. Sobre protocolos eficientes esto puede consumir tan poco como un ancho de banda de 5 kbit/s.

- **Uso más eficiente de los recursos de computación.** Un típico cliente pesado será especificado para hacer frente a la carga máxima de las necesidades del usuario, lo que puede ser ineficiente en los momentos en que no es usado. En contraste, los clientes ligeros usan solamente la cantidad exacta de recursos de computación requeridos para la tarea actual. En una red grande, hay una alta probabilidad de que la carga de cada usuario fluctuará en un ciclo diferente a la de otro usuario, es decir, los picos de uno corresponderán muy probablemente a los bajos de uso de otro. Éste es un resultado natural del efecto aditivo de muchas cargas independientes al azar.

- **Simple trayectoria de actualización de hardware.** Si el pico de recursos está sobre un límite predefinido, es un proceso relativamente simple agregar otro componente a un rack de servidor (ya sea energía, procesamiento, o almacenamiento), empujando los recursos exactamente a la cantidad requerida. Las unidades existentes pueden continuar sirviendo junto a la nueva, mientras que un modelo de cliente pesado requiere que sea reemplazada una unidad de escritorio completa, resultando en tiempo muerto para el usuario, y el problema de disponer de la unidad vieja.

- **Menor ruido.** El ya mencionado retiro de ventiladores reduce el ruido producido por la unidad. Esto puede crear un ambiente de trabajo más agradable y más productivo.

- **Menos hardware desperdiciado.** El hardware contiene metales pesados y plásticos y requiere energía y recursos para ser construido. Los clientes ligeros pueden permanecer en servicio por más tiempo y producen menos hardware excedente que una equivalente instalación de cliente pesado porque pueden ser hechos sin partes móviles. Los ventiladores y unidades de disco del computador (usados para enfriar y el almacenamiento de datos en los clientes pesados) tienen un tiempo medio antes de fallos de muchas miles de horas, pero los transistores y los conductores en el cliente ligero tienen tiempos medios de fallos de millones de horas. Un cliente pesado es considerado viejo después de uno o dos ciclos de la ley de Moore para mantener el procesamiento con el aumento de software inflado (bloatware), pero un cliente ligero puede hacer el mismo trabajo simple año tras año. Por otro lado, un cliente ligero, será reemplazado solamente cuando carezca de una cierta característica considerada esencial. Con audio, video y USB, los clientes ligeros han cambiado poco en 15 años, siendo esencialmente PCs sencillos a los que se le han quitado componentes.

Desventajas de los Thin Client

- **Más requerimientos del servidor.** Un servidor de cliente pesado no requiere tan alto nivel de desempeño como un servidor de cliente ligero (puesto que los clientes pesados por sí mismos hacen mucho del procesamiento de la aplicación). Esto resulta en servidores más caros.

- **Peor desempeño multimedia.** Los clientes pesados tienen ventajas en aplicaciones ricas en multimedia que serían intensivas en ancho de banda si estuvieran completamente residentes en los servidores. Por ejemplo, los clientes pesados están bien adaptados para la edición de vídeo y el video juego, dos casos donde el ancho de banda de la red usado por clientes ligeros será mucho más grande y podrá generar demasiado tráfico en la red obstruyendo a otros usuarios.

- **Menos flexibilidad.** En algunos sistemas operativos (como Microsoft Windows) los productos de software son diseñados para los computadores personales que tienen sus propios recursos locales. Intentar ejecutar este software en un sistema Thin Client puede ser difícil o imposible.

- **Peor soporte de periféricos.** Los clientes ligeros son típicamente cajas muy pequeñas, selladas, sin la posibilidad de extensión interna, y la posibilidad limitada o no existente de extensión externa. Incluso si por ejemplo, un dispositivo de USB puede ser conectado físicamente a un cliente ligero, el software del cliente ligero puede no soportar los periféricos más allá de los dispositivos básicos de entrada y salida por ejemplo, puede no ser compatible con las tarjetas digitalizadoras, cámaras digitales o scanners.

- **Inapropiado para conexiones de red pobres.** Los clientes ligeros pueden ser inusualmente lentos, o muy frustrantes para usar, sobre una conexión de red de

alta latencia. Por otra parte, no trabajan en absoluto cuando la red está caída. Con un cliente pesado, puede ser posible trabajar fuera de línea, aunque la manera orientada a red en la que mucha gente trabaja hoy en día, significa que el uso del cliente pesado también puede ser restringido si la red está caída.

- **Más difícil de reutilizar.** Mientras que un cliente pesado puede ser usado en aplicaciones de cliente ligero, cuando el hardware se vuelve obsoleto para el uso como cliente pesado, debido a que es estándar y puede operar de una manera autónoma, el hardware de cliente ligero es más difícil de revender o reutilizar para otro propósito en caso de que sea retirado.

Escenarios Thin Client

Como vemos existen numerosas ventajas a la hora de utilizar la tecnología Thin Client que nos pueden aportar grandes beneficios. A continuación vamos a citar a modo de ejemplo varios escenarios en los que podríamos usar esta tecnología aprovechándonos de los beneficios que nos aporta.

Cybercafe: Podemos implantar una red Thin Client en un negocio de Cyber Café destinado como objetivo a proporcionar un acceso a internet. El mantenimiento del cyber puede estar centralizado en los servidores, proporcionando a los clientes únicamente lo necesario para realizar su conexión con una cuota de tiempo. El ahorro en costes de hardware y licencias nos proporcionará un mayor número de terminales.

Biblioteca: Una biblioteca donde tengamos un servicio informático para los usuarios, en donde poder consultar o gestionar información de la propia biblioteca a los administradores, dar acceso a internet y un entorno de escritorio y ofimática para las necesidades de los usuarios. Es un ambiente donde se aprovecha la ventajas que nos ofrece un cliente ligero en cuanto a menor ruido frente a un PC.

Aula formativa: Cualquier tipo de aula en un colegio, escuela, una universidad o un centro educativo en general, que precise de una red informática es otro escenario propicio para implantar un sistema Thin Client, en donde el servidor proporcione únicamente los servicios necesarios para las necesidades que se imparten en las aulas. Teniendo una zona de administración de servidores del sistema Thin Client y en las aulas únicamente los clientes ligeros que acceden a los servicios específicos de cada aula.

Laboratorios: Los laboratorios de ordenadores donde tengamos una red es otro escenario donde se puede implantar la tecnología Thin Client, pueden ser laboratorios en las escuelas o universidades, laboratorios de investigación, farmacéuticos, etc.

Centro de desarrollo: En una empresa de desarrollo de software, como pueda ser una consultora tecnológica, donde los desarrolladores trabajan con un entorno de desarrollo perfectamente definido se puede implantar uno o varios sistemas Thin Client en función de las necesidades. En este caso la labor de mantenimiento y la seguridad centralizada proporciona grandes ventajas frente a una red LAN tradicional con PCs en cada puesto con los consiguientes ahorros de presupuesto para la empresa, no solo en el hardware sino en el coste de personal de administración, en los tiempos de instalación de una incorporación nueva en el equipo de desarrollo, mejora en la eficiencia del trabajo de los empleados, así como en la integración de las herramientas de trabajo de cada desarrollador que refuerza el dominio y la formación de los empleados.

Centro de información o gestión: Cualquier centro de información o gestión que haga uso de aplicaciones corporativas donde los gestores se apoyan de medios informáticos para trabajar, como pueda ser una compañía de seguros, una agencia de viajes, el departamento comercial de una empresa, el departamento de atención al cliente, etc.

Centro ofimático. Generalizando el caso de la biblioteca en cualquier empresa donde se tengan unas necesidades ofimáticas para el trabajo, como pueda ser por ejemplo una revista o un periódico.

Red doméstica. En una red doméstica es otro ámbito aunque no lo parezca donde se puede usar Thin Client, en este caso la ventaja no se da tanto en el ahorro económico por sustituir una gran cantidad de terminales pesados por otros más ligeros sino porque que podemos aprovechar la reutilización de equipos viejos, para tener un puesto más para otros miembros de la familia,

potenciando el viejo equipo y aprovechándose de los recursos que le ofrece el equipo más nuevo.

Los escenarios citados son algunos ejemplos, pero en general podríamos decir que se puede implantar un sistema Thin Client en cualquier ámbito donde sea necesario una red de ordenadores, en las que cada PC sustituido por un cliente ligero no necesite un gran procesamiento multimedia, las redes soporten una conexión que no sea pobre, no existan altas necesidades de dispositivos de entrada/salida en cada cliente y no sea necesario procesamiento en tiempo real.

Escenarios de ejemplo negativos para usar esta tecnología pueden ser los siguientes:

Sala de juegos. Un cyber café destinado al uso de juegos en red es un ámbito inapropiado, ya que los juegos requieren grandes recursos de audio y procesamiento gráfico en 3D. La actualización de estas imágenes en red puede colapsar la red frecuentemente con los consiguientes retardos.

Centro de Audio, Video o 3D. Un aula formativa para un curso de audio o Video o animaciones en 3D o en un equipo de desarrollo de estas características, donde los alumnos o trabajadores tienen sus propios ordenadores para aprender o trabajar como pueda ser para el caso de audio: grabación, mezcla, masterización, producción musical, etc., para el caso de video: montaje, producción, etc. o animaciones en 3D: creación, edición, etc. Es inapropiado, ya que los clientes trabajarían con software multimedia que trabaja con ficheros de audio, imagen o video que deben ser procesados en tiempo real y proporcionaría una gran cantidad de tráfico en la red que colapsaría y retardaría las respuestas del sistema.

Protocolos Thin Client

A continuación vamos a entrar en detalle para ver los protocolos más importantes que intervienen en la tecnología Thin Client en los dos servicios que dan lugar a esta tecnología que son el arranque por red y la sesión de escritorio remoto.

Protocolos para el Arranque por Red

Para el arranque por red de un cliente ligero existen varios protocolos, pero nos vamos a centrar en el protocolo PXE, ya que es el más extendido y el más usado.

Protocolo PXE

Hace referencia al entorno de ejecución de prearranque (Preboot eXecution Environment). Es un entorno para arrancar e instalar el sistema operativo en ordenadores a través de una red, de manera independiente de los dispositivos de almacenamiento de datos disponibles (como discos duros) o de los sistemas operativos instalados. PXE utiliza varios protocolos de red como IP, UDP, DHCP y TFTP, y conceptos como Globally Unique Identifier (GUID), Universally Unique Identifier (UUID) y Universal Network Device Interface (UNDI).

El término cliente PXE sólo se refiere al papel que la máquina juega en el proceso de arranque mediante PXE. Un cliente PXE puede ser un servidor, un ordenador de mesa, portátil o cualquier otra máquina que esté equipada con código de arranque PXE, en nuestro caso un cliente ligero.

El protocolo PXE consiste en una combinación de los protocolos DHCP y TFTP con pequeñas modificaciones en ambos. DHCP es utilizado para localizar el

servidor de arranque apropiado, con TFTP se descarga el programa inicial de bootstrap y archivos adicionales. Vamos a ver los protocolos DHCP y TFTP para posteriormente ver como PXE utiliza ambos para conseguir el arranque.

Protocolo DHCP

DHCP (Dynamic Host Configuration Protocol – Protocolo de Configuración Dinámica de Servidor) es un protocolo de red que permite a los nodos de una red IP obtener sus parámetros de configuración automáticamente. Se trata de un protocolo de tipo cliente-servidor en el que generalmente un servidor posee una lista de direcciones IP dinámicas y las va asignando a los clientes conforme éstas van estando libres, sabiendo en todo momento quién ha estado en posesión de esa IP, cuánto tiempo la ha tenido y a quién se la ha asignado después.

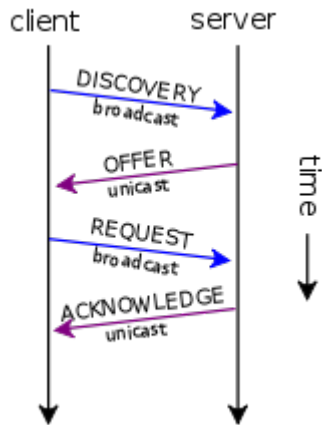
Sin DHCP cada dirección IP debe configurarse manualmente en cada computadora y si la computadora se mueve a otra subred, se debe configurar otra dirección IP diferente. DHCP permite distribuir de forma centralizada las direcciones IP necesarias, y automáticamente, asignar y enviar una nueva IP si fuera el caso en la computadora es conectada en un lugar diferente de la red.

El protocolo DHCP incluye tres métodos de asignación de direcciones IP:

- Asignación manual o estática
- Asignación automática
- Asignación dinámica.

DHCP es una alternativa a otros protocolos de gestión de direcciones IP, como el BOOTP. DHCP es más avanzado, pero ambos son los usados normalmente.

Anatomía del protocolo. Trabaja sobre los puertos 67/UDP para servidor y 68/UDP en cliente.



El cliente envía un paquete DHCPDISCOVER. Las direcciones IP origen y destino de dicho paquete serán 0.0.0.0 y 255.255.255.255 (broadcast) respectivamente. El servidor almacena los campos del paquete CHADDR (dirección Ethernet origen, MAC) y el de identificación del cliente.

El servidor envía un paquete DHCP OFFER, determina la configuración basándose en la dirección del soporte físico de la computadora cliente especificada en el registro CHADDR. El servidor especifica la dirección IP en el registro YIADDR. Como la cual se ha dado en los demás parámetros.

El cliente selecciona la configuración de los paquetes recibidos de DHCP Offer. Una vez más, el cliente solicita una dirección IP específica que indicó el servidor mediante un paquete DHCPREQUEST.

Cuando el servidor recibe el mensaje DHCPREQUEST del cliente, se inicia la fase final del proceso de configuración. Esta fase implica el reconocimiento DHCPACK el envío de un paquete al cliente. Este paquete incluye la duración y cualquier otra información de configuración que el cliente pueda tener solicitada. En este punto, la configuración TCP / IP se ha completado. El servidor reconoce la solicitud y le envía acuse de recibo al cliente. El servidor DHCP responde a la DHCPREQUEST con un DHCPACK, completando así el ciclo de iniciación. La dirección origen es la dirección IP del servidor de DHCP y la dirección destino es todavía 255.255.255.255. El campo YIADDR contiene la dirección del cliente, y los campos CHADDR son la dirección física de la tarjeta de red en el cliente. La sección de opciones del DHCP identifica el paquete como un ACK.

Existen otros dos paquetes que son:

DHCPRELEASE: Petición de un cliente al servidor DHCP para liberar su dirección IP.

DHCPINFORM: Petición de un cliente al servidor DHCP para más información que la que recibió en el DHCPACK original.

Protocolo TFTP

TFTP (Trivial file transfer Protocol – Protocolo de transferencia de archivos trivial). Es un protocolo de transferencia muy simple semejante a una versión básica de FTP. TFTP a menudo se utiliza para transferir pequeños archivos entre máquina en una red, como cuando un cliente ligero arranca desde un servidor de red. Algunas características de TFTP son:

- Utiliza UDP (puerto 69) como protocolo de transporte a diferencia de FTP que utiliza el 21 TCP.
- No puede listar el contenido de los directorios.
- No existen mecanismos de autenticación o cifrado.
- Se utiliza para leer o escribir archivos de un servidor remoto.
- Soporta tres modos diferentes de transferencia, “netascii”, “octet” y “mail” de los que los dos primeros corresponden a los modos “ascii” e “imagen” (binario) del protocolo FTP.

Ya que TFTP utiliza UDP, no hay una definición formal de sesión, cliente y servidor, aunque se considera servidor a aquel que abre el puerto 69 en modo UDP, y cliente a quien se conecta. Sin embargo, cada archivo transferido vía

TFTP constituye un intercambio independiente de paquetes, y existe una relación cliente-servidor informal entre la máquina que inicia la comunicación y la que responde.

- La máquina A, que inicia la comunicación, envía un paquete RPQ (read request/petición de lectura) o WRQ (write request/petición de escritura) a la máquina B, conteniendo el nombre del archivo y el modo de transferencia.
- La máquina B responde con un paquete ACK de confirmación, que también sirve para informa a A del puerto de la máquina B al que tendrá que enviar los paquetes restantes.
- La origen envía paquetes de datos numerados a la máquina destino, todos excepto el último conteniendo 512 bytes de datos. La máquina destino responde con paquetes ACK numerados para todos los paquetes de datos.
- El paquete de datos final debe contener menos de 512 bytes de datos para indicar que es el último. Si el tamaño del archivo transferido es un múltiplo exacto de 512 bytes, el origen envía un paquete final que contiene 0 bytes de datos.

En el protocolo PXE existen ligeras modificaciones en el protocolo para configurar el tamaño de bloque de los paquetes transferidos, con el fin de optimizar la descarga del NBP para que el arranque de por red sea más eficiente.

Una vez que hemos entrado un poco más en qué son y el funcionamiento de los protocolos DHCP y TFTP vamos a ver cómo los usa PXE para llevar a cabo su labor.

Funcionamiento del protocolo PXE

El firmware del cliente trata de encontrar un servicio de redirección PXE en la red para recabar información sobre los servidores de arranque PXE disponibles. Tras analizar la respuesta, el firmware solicitará al servidor de arranque apropiado el file path de un network bootstrap program (NBP) programa cargador de arranque por red, lo descargará en la memoria RAM del ordenador mediante TFTP, probablemente lo verificará, y finalmente lo ejecutará. Si se utiliza un único NBP para todos los clientes PXE se puede especificar mediante BOOTP sin necesidad de DHCP, pero aún será necesario un servidor TFTP.

Para iniciar una sesión de arranque con PXE el firmware envía un paquete de tipo DHCPDISCOVER extendido con algunas opciones específicas de PXE al puerto 67/UDP (puerto estándar del servicio DHCP). Estas opciones indican que el firmware es capaz de manejar PXE, que serían ignoradas por los servidores DHCP estándar.

Si un servicio de redirección PXE (Proxy DHCP) recibe un paquete DHCPDISCOVER extendido, responde con un paquete de difusión DHCPOFFER extendido con opciones PXE al puerto 68/UDP. Este paquete se difundirá hasta que la mayoría de los clientes PXE se autoconfiguren mediante DHCP. Los clientes se identificarán con su GUID/UUID.

Un paquete DHCPOFFER extendido contiene:

- Un campo *PXE Discovery Control* para indicar si se debe utilizar Multicasting, Broadcasting, o Unicasting para contactar con los servidores de arranque PXE
- Una lista con las direcciones IP de los servidores de arranque PXE.
- Un menú en el que cada entrada representa un servidor de arranque PXE.

- Un prompt que indica al usuario que pulse <F8> para ver el menú de arranque.
- Un tiempo de espera que lanza la primera opción del menú de arranque cuando expira.

El servicio de proxy DHCP debe ejecutarse sobre el mismo servidor que el servicio estándar de DHCP. Puesto que ambos servicios no pueden compartir el puerto 67/UDP, el Proxy DHCP se ejecuta sobre el puerto 4011/UDP y espera que los paquetes DHCPDISCOVER extendidos de los clientes PXE sean paquetes DHCPREQUEST. El servicio estándar DHCP debe enviar una combinación especial de opciones PXE en su paquete DHCPOFFER, de forma que los clientes PXE sepan que deben buscar un proxy DHCP en el mismo servidor, en el puerto 4011/UDP.

Para contactar con el servidor de arranque PXE el firmware debe obtener una dirección IP y el resto de información de un único paquete DHCPOFFER extendido. Tras elegir el servidor de arranque PXE y la capa de arranque PXE, permitiendo ejecutar múltiples tipos de servidores de arranque mediante un único daemon (o programa) de arranque. El paquete DHCPREQUEST extendido también puede ser un paquete DHCPINFORM.

Si un servidor de arranque PXE recibe un paquete DHCPREQUEST extendido como el descrito anteriormente y si está configurado para el tipo de servidor de arranque PXE y la arquitectura cliente solicitados, debe responder devolviendo un paquete DHCPACK extendido que son opciones específicas de PXE.

El contenido más importante de un paquete DHCPACK extendido es:

- El file path completo para descargar el NBP vía TFTP.
- El tipo de servidor de arranque PXE y la capa de arranque PXE.
- La configuración multicast TFTP, si debe utilizarse multicast TFTP.

Un servidor de arranque PXE debe soportar Boot Integrity Services (BIS). BIS permite al cliente PXE verificar los NBPs descargados mediante un archivo checksum que es descargado desde el mismo servidor de arranque que el NBP.

Tras recibir el paquete DHCPACK solicitado, el Network Bootstrap Program comienza a descargarse vía TFTP con los parámetros recibidos y una vez descargado se carga en la RAM del cliente y se pasa a su ejecución.

Network Bootstrap Program (NBP)

El NBP es el programa que arranca un sistema operativo por red, es un bootloader (cargador de arranque), es decir, un programa capaz de cargar la imagen de un sistema operativo en la RAM de la máquina y pasarle el control de ejecución, en este caso por red.

Protocolos para Sesiones de Escritorio Remoto

Un escritorio remoto es una tecnología que permite a un usuario trabajar en una computadora a través de su escritorio gráfico desde otro terminal ubicado en otro lugar.

La tecnología de escritorio remoto permite la centralización de aquellas aplicaciones que generalmente se ejecutan en entorno de usuario. Es por esto que este servicio es la base de la tecnología Thin Client, ya que los clientes se transforman en máquinas de entrada y salida. Los eventos de pulsación de teclas y movimientos de ratón se transmiten a un servidor central donde la aplicación los procesa como si se tratase de eventos locales. La imagen en pantalla de dicha aplicación es retornada al terminal cliente.

La tecnología de escritorio remoto se basa en cualquier implementación en su protocolo de comunicaciones entre el cliente y el servidor que permite el envío y recepción de la entrada y salida de eventos y pantallas gráficas.

Para el desarrollo correcto del escritorio remoto, el protocolo debe de cumplir una serie de requisitos:

- Latencia: el retardo de las comunicaciones es más importante que el ancho de banda.
- Balanceo de funcionalidad entre cliente y servidor.
- Representación gráfica: Indica el tipo de visualización del escritorio remoto. Existen cuatro posibilidades: gráficos de alto nivel, gráficos de bajo nivel, primitivas de dibujo en 2D y píxels en bruto.
- Compresión: de los datos relativos a las actualizaciones de la pantalla.
- Sincronización.

Existen diversas implementaciones de escritorio remoto con sus protocolos de comunicaciones correspondientes, vamos a ver en detalle cada una de las siguientes:

- XDMCP utilizado por X-Window
- VNC (Virtual Network Computing)
- RDP utilizado por Terminal Services
- ICA utilizado por MetaFrame.
- AIP utilizado por Secure Global Desktop
- Tecnología NX

X Window System / XDMCP

X Window System (sistema de ventanas X) es un software que fue desarrollado a mediados de los años 1980 en el MIT para dotar de una interfaz gráfica a los sistemas Unix. Este protocolo permite la interacción gráfica en red entre un usuario y una o más computadoras haciendo transparente la red para éste. Generalmente se refiere a la versión 11 de este protocolo, X11, el que está en uso actualmente. X es el encargado de mostrar la información gráfica de forma totalmente independiente del sistema operativo.

El sistema de ventanas X distribuye el procesamiento de aplicaciones especificando enlaces cliente-servidor. El servidor provee servicios para acceder a la pantalla, teclado y ratón, mientras que los clientes son las aplicaciones que utilizan estos recursos para interacción con el usuario. De este modo mientras el servidor se ejecuta de manera local, las aplicaciones pueden ejecutarse remotamente desde otras máquinas, proporcionando así el concepto de transparencia de red.

Debido a este esquema cliente-servidor, se puede decir que X se comporta como una terminal gráfica virtual. El hecho que exista un estándar definido para X permite que se desarrollen servidores X para distintos sistemas operativos y plataformas, lo que hace que el código sea muy portable. Por ejemplo, permite tener clientes X ejecutándose en un potente servidor UNIX mientras los resultados son visualizados en un PC de escritorio con cualquier otro sistema operativo funcionando.

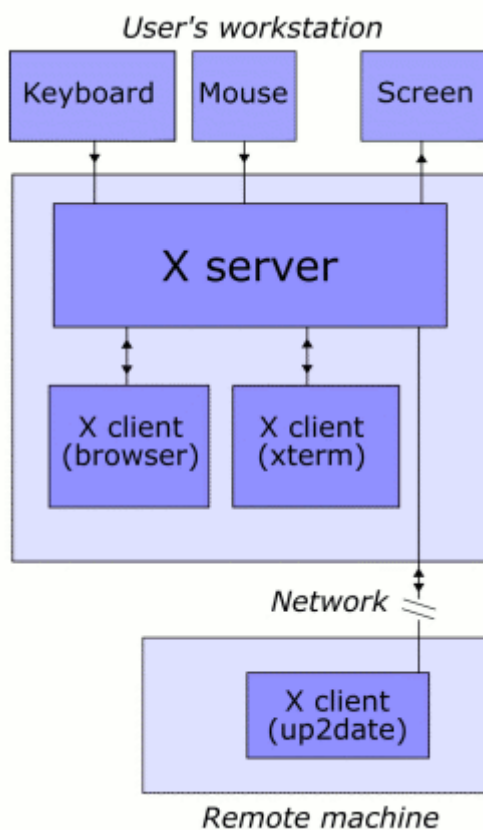
X no es un gestor de ventanas, necesita de uno para controlar el manejo de ventanas. Esto trae la ventaja de que permite al usuario instalar uno o más administradores de ventanas de su preferencia. También trae la ventaja de que hace de X estrictamente un sistema gráfico, de tal modo que un cliente X podría estar enviando un gráfico a una pantalla, a una impresora o a cualquier otro hardware sin darse cuenta, flexibilizando la salida gráfica.

X usa el modelo cliente-servidor: un servidor X se comunica con varios programas cliente. El servidor acepta los pedidos para la salida gráfica (ventanas) y devuelve la entrada del usuario (desde el teclado, del ratón, o de la pantalla táctil). El servidor puede funcionar así:

- Una aplicación exhibiendo hacia una ventana de otro sistema de visualización
- Un programa del sistema controlando la salida vídeo de una PC
- Una pieza de hardware dedicada

Esta terminología de cliente servidor - el terminal de usuario siendo el servidor y las aplicaciones siendo los clientes - a menudo confunde a nuevos usuarios de X, porque los términos parecen invertidos. Pero X toma la perspectiva de la aplicación, en vez de la del usuario final: X proporciona la exhibición por pantalla y los servicios de entrada/salida a las aplicaciones, así que es un servidor; las aplicaciones usan estos servicios, por lo tanto son los clientes.

El protocolo de comunicaciones entre el servidor y el cliente opera transparente a la red: el cliente y el servidor pueden correr en la misma o en diferentes máquinas, posiblemente con diferentes arquitecturas y sistemas operativos. Un cliente y un servidor pueden incluso comunicarse con seguridad sobre Internet haciendo una conexión de túnel sobre una sesión cifrada de la red.



Un cliente X puede emular un servidor X proporcionando servicios de exhibición a otros clientes. Esto es conocido como "X nesting" (anidado X). Los clientes de código abierto tales como Xnest y Xephyr soportan el X nesting.

Para utilizar un programa de cliente X sobre una máquina remota, el usuario hace lo siguiente:

- En la máquina local, abrir una ventana de terminal
- Usar telnet o ssh para conectarse con la máquina remota
- Solicitar el servicio local de pantalla/entrada (ej., export DISPLAY=[user's machine]:0 si no se está usando SSH con X tunneling activado)

El cliente X remoto entonces hará una conexión al servidor X local del usuario, proporcionando la exhibición por pantalla y la entrada. Alternativamente, la máquina local puede correr un pequeño programa que se conecte con la máquina remota y comience la aplicación cliente.

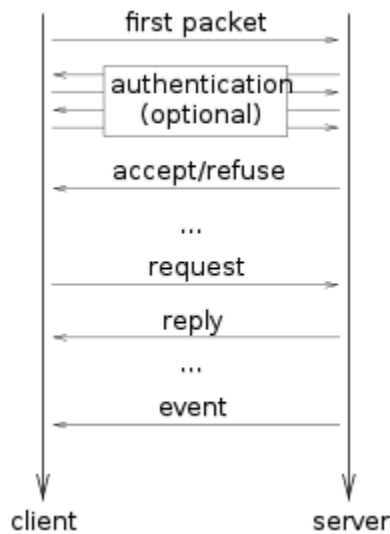
Ejemplos prácticos de clientes remotos incluyen:

- Administrando una máquina remota gráficamente.
- Corriendo una simulación de cómputo intensivo en una máquina Unix remota y mostrando los resultados por pantalla en una de escritorio Windows.
- Correr software gráfico en varias máquinas al mismo tiempo, controlados por una sola pantalla, teclado y ratón.

La comunicación entre el servidor y los clientes se hace mediante el intercambio de paquetes sobre un canal. La conexión se establece por el cliente (la forma en que el cliente se inicia no se ha especificado en el protocolo). El cliente también envía el primer paquete, que contiene el orden del byte a ser utilizado y la información sobre la versión del protocolo y el tipo de autenticación que el cliente espera que el servidor usará. La respuesta del servidor mediante el envío de un paquete de vuelta indica la aceptación o el rechazo de la conexión, o con una solicitud de una autenticación adicional. Si la conexión es aceptada, el paquete de aceptación contiene los datos que el cliente debe usar en la interacción posterior con el servidor.

Después de que se establezca la conexión, cuatro tipos de paquetes son intercambiados entre el cliente y el servidor sobre el canal de comunicación:

1. **Petición:** El cliente pide información al servidor o solicita que éste realice una acción.
2. **Respuesta:** El servidor responde a una petición. No todas las peticiones generan respuestas.
3. **Evento:** El servidor informa al cliente de un acontecimiento, tal como la entrada del teclado o del ratón, que una ventana está siendo movida, redimensionada, expuesta, etc.
4. **Error:** El servidor envía un paquete de error si una petición es inválida. Puesto que las respuestas están en cola, los paquetes de error generados por una petición pueden no enviarse inmediatamente.



Los paquetes de la petición y de la respuesta tienen una longitud diversa, mientras que los paquetes de eventos y de error tienen una longitud fija de 32 bytes.

Los paquetes de petición son numerados secuencialmente por el servidor tan pronto como los recibe: la primera petición de un cliente se numera 1, la segunda 2, etc. Los 16 bits menos significativos del número secuencial de una

petición se incluyen en los paquetes de la respuesta y del error, si los hay, generados por la petición. También son incluidos en los paquetes de eventos, para indicar el número secuencial de la petición que el servidor está actualmente procesando o acaba de terminar de procesar.

X es primariamente una definición de primitivas de protocolo y gráficas, y deliberadamente no contiene especificaciones de diseño de interfaz de usuario, como estilos de botón, menú, barra de título para las ventanas. En vez de eso, un software de aplicación, tal como los manejadores de ventana, Widget toolkits de GUI y ambientes de escritorio, o las interfaces gráficas de usuario específicas de una aplicación -definen y proporcionan tales detalles. Como resultado, no hay interfaz X típica y varios ambientes de escritorio han sido populares entre los usuarios.

Un manejador de ventana controla la colocación y la apariencia de las ventanas de aplicación. Esto puede resultar en interfaces semejantes a las de Microsoft Windows o Macintosh (los ejemplos incluyen Metacity en GNOME, KWin en KDE, Xfwm en Xfce, o Compiz) o tener controles radicalmente diferentes (tales como tiling window manager, como wmii o Ratpoison). Los manejadores de ventana abarcan en sofisticación y complejidad desde los más simples (ej., twm, el manejador de ventana básico suministrado con X, o evilwm, un manejador de ventana extremadamente ligero) hasta los ambientes de escritorio más completos tales como Enlightenment.

Muchos usuarios usan X con un ambiente de escritorio, que, independientemente del manejador de ventana, incluyen varias aplicaciones usando una interfaz de usuario consistente. GNOME, KDE y Xfce son los ambientes de escritorio más populares. El ambiente estándar de Unix es Common Desktop Environment (CDE).

XDMCP es el “X Display Manager Control Protocol – Manejador del Protocolo de control de X”. Es un protocolo utilizado en redes para comunicar un ordenador servidor que ejecuta un sistema operativo con un gestor de ventanas basado en X-Window con el resto de clientes que se conectarán a éste con propósitos interactivos.

Un **X Display Manager** (administrador de pantalla X) mantiene los procesos del servidor X activos en él, conectándolo a una pantalla física y proporcionando un cuadro de diálogo que permita iniciar sesión en el mismo a todos los ordenadores que lo requieran. Escucha el User Datagram Protocol (UDP) en el puerto 177 y responde a peticiones de tipo *QUERY* y *BROADCAST_QUERY* enviando un paquete tipo *WILLING* al equipo que le contactó.

Nuestro cliente marca una llamada al XDMCP del Display, el cual lo obtiene de la siguiente manera: X-query 192.168.x.x:display donde quiere activarse. Al ingresar tal comando, nuestro cliente inicia la llamada de conexión y el servidor de XDMCP le envía el prompt de Login (un display como XDM) de respuesta, donde nuestro cliente ingresa su cuenta con su usuario y password, tal y como si estuviera en la misma máquina servidora validándose en su propio escritorio.

A continuación vemos una captura del display del cliente, se trata del login de XDM



X proporciona XDM (**X Window Display Manager**) como administrador de pantalla primario. Otros ejemplos de administradores de pantalla son GDM (GNOME), KDM (KDE), Entrance (Enlightenment), etc.

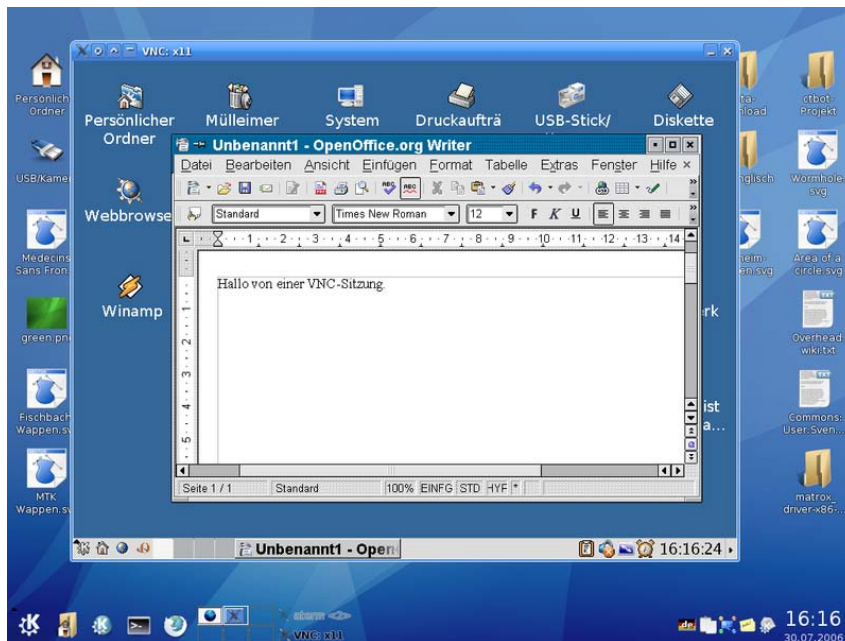
VNC

VNC (Virtual Network Computing – Computación en Red Virtual). Es un programa de software libre basado en una estructura cliente-servidor el cual nos permite tomar el control del ordenador servidor remotamente a través de un ordenador cliente. VNC permite que el sistema operativo en cada computadora sea distinto: Es posible compartir la pantalla de una máquina de "cualquier" sistema operativo conectando desde cualquier otro ordenador o dispositivo que disponga de un cliente VNC portado.

La versión original del VNC se desarrolló en Reino Unido, concretamente en los laboratorios AT&T, en Cambridge. El programa era de código abierto por lo que cualquiera podía modificarlo y existen hoy en día varios programas para el mismo uso.

El programa servidor suele tener la opción de funcionar como servidor HTTP para mostrar la pantalla compartida en un navegador con soporte de Java. En este caso el usuario remoto (cliente) no tiene que instalar un programa cliente de VNC, éste es descargado por el navegador automáticamente.

A continuación vemos una pantalla en la que tenemos un cliente VNC bajo Linux en un escritorio KDE.



Terminal Services / RDP

Los **Servicios de Escritorio Remoto (Remote Desktop Services)**, formalmente conocido como **Servicios de Terminal (o Terminal Services)** son un componente de los sistemas operativos Windows que permite a un usuario acceder a las aplicaciones y datos almacenados en otro ordenador mediante un acceso por red.

Basado en el protocolo de escritorio remoto (Remote Desktop Protocol (RDP)) aparece por primera vez en Windows NT 4.0 (Terminal Server Edition).

Los productos Windows 2000 Server, Windows 2000 Advanced Server, Windows 2000 Datacenter Server y Windows Server 2003 han introducido algunas mejoras y funcionalidades nuevas.

Microsoft proporciona el software cliente para todas las versiones de Windows 32 bits y para Mac OS X de Apple. El uso de los servicios de terminal requiere de tres componentes:

- 1-Servidor de Terminal Server.
- 2-Cliente de Terminal Server.
- 3-Protocolo de escritorio remoto.

La instalación de dicho componente no supone mayor problema ya que se incorpora en los sistemas operativos, aunque sí que es algo diferente en Windows 2000 y 2003. Podemos distinguir dos tipos de instalación:

- 1. Modo Administración remota: proporciona acceso remoto a los servidores por parte de los administradores. Soporta, además de la sesión de consola, dos sesiones más, sin tener que pagar ninguna licencia extra.
- 2. Modo Servidor de Aplicaciones: permite el acceso simultáneo por parte de varios clientes remotos. En este caso sí será necesario adquirir licencias de terminal.

Remote Desktop Protocol (RDP) es el protocolo propietario desarrollado por Microsoft que permite la comunicación en la ejecución de una aplicación entre un terminal (mostrando la información procesada que recibe del servidor) y un servidor Windows (recibiendo la información dada por el usuario en el terminal mediante el ratón ó el teclado).

El modo de funcionamiento del protocolo es sencillo. La información gráfica que genera el servidor es convertida a un formato propio RDP y enviada a través de la red al terminal, que interpretará la información contenida en el paquete del protocolo para reconstruir la imagen a mostrar en la pantalla del terminal. En cuanto a la introducción de órdenes en el terminal por parte del usuario, las teclas que pulse el usuario en el teclado del terminal así como los movimientos y pulsaciones de ratón son redirigidos al servidor, permitiendo el protocolo un cifrado de los mismos por motivos de seguridad. El protocolo también permite que toda la información que intercambien cliente y servidor sea comprimida para un mejor rendimiento en las redes menos veloces. Pues es la única de las soluciones de clientes ligeros analizadas que nos permite utilizar este protocolo para que los terminales puedan actuar como clientes de servidores Windows, lo

que puede ser interesante en multitud de ambientes de trabajo en los que se utilizan servidores Microsoft.

Este servicio utiliza por defecto el puerto TCP 3389 en el servidor para recibir las peticiones. Una vez iniciada la sesión desde un punto remoto el ordenador servidor mostrará la pantalla de bienvenida de windows, no se verá lo que el usuario está realizando de forma remota.

A continuación presentamos las características del protocolo RDP:

- Permite el uso de colores de 8, 15, 16, 24 y 32 bits.
- Cifrado de 128 bits utilizando el algoritmo criptográfico RC4. Los clientes más antiguos pueden utilizar cifrados más débiles.
- Permite seguridad a nivel de transporte Transport Layer Security.
- El redireccionamiento del audio permite al usuario ejecutar un programa de audio en una ventana remota y escuchar el sonido en el ordenador local.
- El redireccionamiento del sistema de ficheros permite a los usuarios utilizar sus ficheros locales en una ventana remota.
- Permite al usuario utilizar su impresora local al estar conectado al sistema remoto.
- El redireccionamiento de puertos permite utilizar los puertos serie y paralelo directamente.
- El portapapeles puede compartirse entre los ordenadores local y remoto.
- Programas remotos: Aplicaciones con ficheros del lado del cliente.
- Las aplicaciones (seamless windows) remotas pueden funcionar en una máquina cliente servida por una conexión remota.
- Permite utilizar un servidor IIS de manera que acepte conexiones en el puerto 443 para servidores de respaldo de Terminal Services mediante

conexiones HTTPS, similar a como las llamadas remotas RCP sobre HTTPS permiten a los clientes Outlook conectar a un servidor de copias Exchange 2003. Se necesita Windows Server 2008.

- Soporte remoto de Aero Glass Thema (o Composed Desktop), incluyendo tecnología de suavizado de fonts ClearType
- Soporte para aplicaciones Windows Presentation Foundation compatibles con clientes .Net Framework 3.0 y que sean capaces de tener efectos en la máquina local.
- Revisado para que el redireccionamiento de dispositivos sea más general, permitiendo una mayor variedad de dispositivos.
- Todos los servicios de terminal serán totalmente configurables y suscritos vía Windows Management Instrumentation.
- Ancho de banda ajustado para clientes RDP.
- Soporte para Transport Layer Security (TLS) 1.0 en los lados cliente y servidor.
- Soporte de varios monitores. La sesión puede mostrarse en dos monitores.

Tecnología NX

NX es un protocolo que realiza las conexiones X11 muy rápidas, lo que permite acceso a escritorios remotos sobre Linux o Unix mediante conexiones lentas como por ejemplo por modem.

La mejora en la velocidad se basa en la compresión directa del protocolo X11, lo que permite una mayor eficiencia que VNC y que XDMCP. Además la información se envía mediante SSH, por lo que toda la información que se intercambian servidor y cliente está cifrada mejorando también la seguridad en la conexión.

NX está desarrollado por la empresa italiana NoMachine, que ha liberado el código. Existe una implementación libre llamada FreeNX.

La tecnología NX ofrece una mayor eficiencia que otras debido principalmente a las características que se listan a continuación:

- Realiza una eficiente compresión del tráfico X. La compresión del tráfico de forma eficiente es necesaria para lograr ejecutar aplicaciones sobre medios de poco ancho de banda y también para permitir ejecutar múltiples sesiones de usuario en redes LAN.
- Utiliza mecanismos de cache para almacenar y reutilizar la información transferida entre cliente y servidor. NX utiliza un método de cache innovador que divide el mensaje X en dos partes, uno de identificación y otro de datos. La tecnología mantiene en cache únicamente los datos de los últimos mensajes enviados, clasificados por protocolo. A este cache se le conoce como MessageStore y hace que el número de peticiones para mostrar los elementos de pantalla disminuya notoriamente.
- El tiempo consumido en realizar roundtrips (par de petición-respuesta entre cliente y servidor) es prácticamente nulo. La reducción de roundtrips es fundamental, al igual que el estricto control del flujo de datos que viaja por la red.
- Utiliza un algoritmo de codificación perezoso para realizar actualizaciones de pantalla. NX posee mecanismos de adaptación para ajustarse a las propiedades de la red (latencia y velocidad de conexión), lo que permite pasar de métodos estrictos de codificación a métodos perezosos que retrasen la actualización de pantalla cuando la red está congestionada.

El principal objetivo de esta tecnología es el de ejecutar a través de internet aplicaciones que logren tener el mismo aspecto gráfico que cuando se ejecutan en cualquier PC. Generalmente cuando esto ocurre es necesario deshabilitar todo aquello que pueda consumir demasiados recursos, como por ejemplo, menús desplegados, fondos de pantalla, iconos o animaciones gráficas. NX fue diseñado especialmente para soportar estas condiciones y no hacer que usuarios y desarrolladores deban cambiar sus hábitos o su código.

COMPARATIVA DE SOFTWARE DE ESCRITORIO REMOTO

Software	Protocol	License	Client/Server	Built-in encryption	File transfer	Audio support	Multiple sessions	Seamless window	FreeBSD client	Linux client	Mac OS X client	Microsoft Windows client	Blackberry client	Windows Mobile client	Java client	iPhone/iPod client	Android client
AnywhereTS	RDP, ICA	Proprietary	Client Only	SSL, TLS	✓	✓	✓	X	X	X	X	✓	X	X	X	?	?
Apple Remote Desktop	RFB(VNC)	Proprietary	Client & Server	AES-128	✓	X	✓	X	X	X	✓	X	X	X	X	?	?
Apple Screen Sharing (iChat)	Proprietary,RFB(VNC)	Proprietary	Client & Server	AES-256	X	X	✓	X	X	X	✓	X	X	X	X	?	?
Bomgar	Proprietary,X11	Proprietary	Client & Server	SSL, AES-256	✓	X	✓	✓	✓	✓	✓	✓	✓	✓	X	?	?
Cendio ThinLinc	RFB(VNC)	Proprietary	Client & Server	SSH	✓	✓	✓	✓	X	✓	✓	✓	X	X	✓	?	?

Chicken of the VNC	RFB(VNC)	GPL	Client Only	X	X	X	X	X	X	X	X	✓	X	X	X	X	?	?
Citrix XenApp	RDP, ICA	Proprietary	Client & Server	SSL, TLS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Crossloop	RFB(VNC)	Proprietary	Client & Server	✓	✓	X	X	X	X	X	X	✓	✓	X	X	X	X	X
DameWare Mini Remote Control	RDP,Proprietary	Proprietary	Client & Server	SSL, AES-256	✓	✓	✓	X	X	X	X	✓	✓	X	X	X	?	?
EchoVNC	RFB(VNC)	GPL	Client & Server	✓	✓	X	✓	X	X	✓	X	✓	✓	X	X	X	?	?
Ericom PowerTerm WebConnect	RDP,RFB(VNC)	Proprietary	Client & Server	✓	✓	✓	✓	✓	X	✓	✓	✓	✓	X	X	X	?	?
Fog Creek Copilot	RFB(VNC)	Proprietary	Client & Server	SSL	✓	X	X	X	X	X	X	✓	✓	X	X	X	?	?
GO-Global	Proprietary	Proprietary	Client & Server	SSL, DES-56, DES-168, AES-256	✓	X	✓	✓	X	✓	✓	✓	✓	X	✓	✓	?	?
FreeNX	NX, RDP,RFB(VNC)	GPL	Client & Server	AES	✓	✓	✓	✓	✓	✓	✓	✓	✓	X	X	X	?	?

GoToMyPC	Proprietary	Proprietary	Client & Server	AES-256	✓	✓	✓	X	X	✓	✓	✓	X	X	✓	?	?
HOB HOBLINK JWT	RDP	Proprietary	Client Only	SSL, TLS	✓	✓	✓	✓	✓	✓	✓	✓	X	X	✓	?	?
HOB HOBLINK UWT	RDP	Proprietary	Client Only	SSL, TLS	✓	✓	✓	✓	✓	✓	✓	X	X	X	X	?	?
I'm InTouch	Proprietary	Proprietary	Client & Server	SSL, AES-256	✓	✓	✓	X	X	✓	✓	✓	X	X	✓	?	?
ISL Light	Proprietary,RFB(VNC)	Proprietary	Client & Server	SSL, AES-256	✓	✓	✓	X	X	✓	✓	✓	X	X	X	?	?
iTALC	RFB(VNC)	GPL	Client & Server	SSH	X	X	✓	X	X	✓	X	✓	X	X	X	?	?
IBM Director Remote Control	Proprietary	Proprietary	Client & Server	AES, Triple DES, DES	✓	X	✓	X	X	X	X	✓	X	X	✓	?	?
KRDC	RFB(VNC),RDP	GPL	Client Only	X	X	X	✓	X	✓	✓	X	X	X	X	X	?	?
LogMeIn Free	Proprietary	Proprietary	Client & Server	SSL	X	X	X	X	X	X	✓	✓	X	X	✓	✓	?
LogMeIn Pro	Proprietary	Proprietary	Client &	SSL	✓	✓	✓	X	X	X	✓	✓	X	X	✓	✓	?

		ry	Server														
Mac HelpMate	RFB(VNC)	Proprietary	Client & Server	AES-128	X	X	✓	X	X	X	✓	✓	X	X	✓	?	?
Mikogo	Proprietary	Freeware	Client & Server	AES-256	✓	X	✓	X	X	X	✓	✓	X	X	X	X	X
Netman	Proprietary	Proprietary	Client & Server	DES	✓	✓	✓	✓	X	X	X	✓	X	X	X	?	?
Netsupport Manager	Proprietary	Proprietary	Client & Server	✓	✓	✓	✓	X	✓	✓	✓	✓	X	✓	✓	?	?
Netviewer	Proprietary	Proprietary	Client & Server	AES-128	✓	✓	✓	X	X	X	X	✓	X	X	X	?	?
Open Text Exceed onDemand	Proprietary	Proprietary	Client & Server	SSL, SSH	✓	X	✓	✓	✓	✓	X	✓	X	X	X	X	X
rdesktop	RDP	GPL	Client Only	✓	✓	✓	✓	✓	✓	✓	✓	✓	X	X	✓	?	?
RealVNC Free	RFB(VNC)	GPL	Client & Server	X	X	X	✓	X	✓	✓	X	✓	X	X	✓	X	X
RealVNC Personal	RFB(VNC)	Proprietary	Client & Server	AES-128	✓	X	✓	X	X	X	X	✓	X	X	✓	X	X

RealVNC Enterprise	RFB(VNC)	Proprietary	Client & Server	AES-128	✓	X	✓	X	✓	✓	✓	✓	X	X	✓	✓	X
Remote Administrator	Proprietary	Proprietary	Client & Server	AES-256	✓	✓	✓	✓	X	✓	X	✓	X	X	X	X	X
Remote Desktop Connection/Terminal Services	RDP	Proprietary	Client & Server	✓	✓	✓	✓	✓	X	✓	✓	✓	X	✓	✓	✓	?
Remote Graphics Software	Proprietary	Proprietary	Client & Server	✓	✓	✓	✓	X	X	✓	X	✓	✓	✓	✓	?	?
RemoteVNC	RFB(VNC)	GPL	Client & Server	SSL,TLS	✓	X	✓	X	X	X	X	✓	X	X	X	?	?
SimpleHelp	Proprietary	Proprietary	Client & Server	448-bit	✓	X	✓	X	✓	✓	✓	✓	X	X	✓	?	?
SSH with X forwarding	X11	BSD	Client & Server	✓	X	X	✓	✓	✓	✓	✓	✓	X	X	X	?	?
SoftRay	ALP	GPL	Client only	X	X	X	X	X	X	X	X	X	X	X	✓	?	?
Sub7	Proprietary	GPL	Client & Server	X	X	X	X	X	X	X	X	X	X	X	✓	?	?
Sun Secure Global	AIP	Proprietary	Client &	SSL, TLS(AE	✓	✓	✓	✓	✓	✓	✓	✓	X	X	✓	?	?

Desktop Software		ry	Server	S-256)													
SymantecpcAnywhere	Proprietary	Proprietary	Client & Server	AES-256	✓	X	✓	X	X	✓	✓	✓	✓	✓	✓	?	?
TeamViewer	Proprietary	Proprietary	Client & Server	AES-256	✓	X	✓	X	X	X	✓	✓	X	X	✓	✓	X
TightVNC	RFB(VNC)	GPL	Client & Server	X	✓	X	✓	X	✓	✓	✓	✓	X	X	✓	?	✓
Timbuktu	Proprietary	Proprietary	Client & Server	✓	✓	X	✓	X	X	X	✓	✓	X	X	X	?	?
UltraVNC	RFB(VNC)	GPL	Client & Server	With plugin	✓	X	✓	Partial	X	X	X	✓	X	X	✓	?	?
Webex	Proprietary	Proprietary	Client & Server	✓	✓	✓	✓	X	X	✓	✓	✓	X	X	✓	?	?
WebTrain	Proprietary	Proprietary	Client & Server	SSL, TLS	✓	✓	✓	X	X	X	X	✓	X	X	N/A	?	?
XDMCP	X11	MIT	Client & Server	X	X	X	✓	X	✓	✓	✓	X	X	X	X	?	?
xpra	Bencode-based	GPL	Client & Server	SSH	X	X	✓	✓	✓	✓	X	X	X	X	X	?	?

xrdp	RDP	GPL	Server Only	✓	X	X	✓	X	✓	✓	X	X	X	X	N/A	?	?
X11vnc	RFB(VNC)	GPL	Server Only	SSL	✓	X	✓	X	✓	✓	✓	X	X	X	✓	?	?
x2vnc	X11	BSD	Client Only	X	X	X	✓	N/A	✓	✓	✓	✓	X	X	X	?	?
x2x	X11	BSD	Client Only	X	X	X	✓	N/A	✓	✓	✓	✓ ¹	X	X	X	?	?
Software	Protocol	License	Client/Server	Built-in Encryption	File Transfer	Audio Support	Multiple Sessions	Seamless Window	FreeBSD Client	Linux Client	Mac OS X Client	Microsoft Windows Client	Blackberry Client	Windows Mobile Client	Java Client	iPhone Client	Android Client

Software Libre en la tecnología Thin Client

Arranque por red

Etherboot/gPXE. Es un proyecto open source (GPL) para el arranque por red, que proporciona el firmware PXE ROM, con algunas características extra como DNS, HTTP, iSCSI, etc. <http://etherboot.org/wiki/index.php>

Netboot. Proyecto Open Source GPL que permite el arranque remoto de una máquina sobre una red IP sin acceder al disco duro. Ideal para el arranque de Thin Clients usando un procesador x86. <http://netboot.sourceforge.net/english/index.shtml>

DHCP. Como hemos visto DHCP (Dynamic Host Configuration Protocol) es un servicio de red que permite a las máquinas obtener su configuración de red automáticamente, recibiendo esta configuración del servidor DHCP, siendo esta transparente al usuario. Existen varias implementaciones de servidores DHCP, como por ejemplo dhcp3-server, que es la versión 3 de la implementación del ISC (Internet Software Consortium) de DHCP <http://www.isc.org>

TFTP. Tenemos también varias implementaciones de servidores TFTP como por ejemplo el proyecto **atftpd** es una implementación del servidor TFTP multi-threaded con todas las opciones y extensiones especificadas en RFC1350, RFC2090, RFC2347, RFC2348 y RFC2349. Atftpd soporta además multicast también conocido como mftft, definido en la especificación PXE. Otra opción para PXE es **tftpd-hpa**, se trata de un servidor tftp derivado de OpenBSD tftp con algunas opciones extra añadidas. Es útil para usar el protocolo PXE ya que implementa las características necesarias no definidas en el protocolo tftp.

Kernel y Cargadores

Kernel Linux. En la página web <http://www.kernel.org/> tenemos el código fuente de todas las versiones del núcleo del sistema operativo Linux, para adaptarlo y configurarlo para nuestras necesidades de sistema empujado.

PXELinux. Proyecto derivado de Syslinux para arrancar Linux por red mediante el protocolo PXE. Más información en: <http://syslinux.zytor.com/wiki/index.php/PXELINUX>

Escritorio Remoto

Xnest. <http://www.xfree86.org/4.2.0/Xnest.1.html> Es un servidor X Window System que direcciona su salida a una ventana. Se puede usar para crear un escritorio virtual de otro equipo, dentro de una ventana. Puede mostrar un escritorio remoto de otro equipo localmente usando XDMCP, siempre que la máquina remota acepte conexiones XDMCP entrantes. De forma alternativa, Xnest puede ser ejecutado en la máquina remota mientras se muestra su ventana localmente.

Xvncviewer. Es un cliente de una VNC para X, lo cual permite ver un entorno de escritorio no solo en la máquina donde está corriendo sino en cualquier máquina en internet y una gran variedad de arquitecturas, este cliente en concreto es para X, con este cliente nos podemos conectar a vncserver en una red y mostrar el contenido en una ventana. Vncserver está disponible para X y para Win/NT.

FreeNX. <http://freenx.berlios.de/> Es una implementación libre de la tecnología NX, las librerías base son proporcionadas por NoMachine bajo GPL. FreeNX es una implementación GPL de los componentes NX Server y el NX Client.

RDesktop. <http://www.rdesktop.org/>. Es un Cliente Open Source para Windows Terminal Service, capaz de hablar nativamente el protocolo RDP para presentar al usuario escritorios Windows.

XRDP. <http://xrdp.sourceforge.net/>. Basado en el trabajo de RDesktop es un servidor de escritorio remoto que utiliza el protocolo RDP. El objetivo de este proyecto es proporcionar una entera funcionalidad de Linux Terminal Server, capaz de aceptar conexiones desde rdesktop y desde los clientes escritorios remoto de Microsoft. Xrdp usa Xvnc o X11rdp para manejar la sesión X.

Tsclient. (Terminal Server Client) es una aplicación gnome que hace de front-end para rdesktop.

XWindow. En este enlace <http://www.xfree86.org/> tenemos la página web del proyecto X Window System, mediante este sistema cliente-servidor de ventanas podemos instalar los diversos entornos gráficos de escritorio al cuál se conectan los clientes ligeros, como puede ser por citar algunos los siguientes:

GNOME. GNU Object Model Environment. <http://www.gnome.org/>. Es un entorno de escritorio e infraestructura de desarrollo para sistemas operativos Unix y derivados Unix como GNU/Linux, BSD o Solaris, compuesto enteramente por software libre.

KDE. <http://www.kde.org/> Comunidad internacional de un equipo tecnológico dedicado a crear un escritorio gráfico avanzado libre. Es un proyecto de software libre dedicado para la creación de un entorno de escritorio e infraestructura de desarrollo para diversos sistemas operativos como GNU/Linux, Mac OS X, Windows, etc.

LXDE. <http://lxde.org/> Es un entorno de escritorio libre para Unix y otras plataformas POSIX, como Linux o BSD. El nombre corresponde a “Lightweight X11 Desktop Enviroment”, es decir, Entorno de escritorio X11 ligero. Es un proyecto que apunta a entregar un nuevo entorno de escritorio ligero y rápido, ni está diseñado para ser tan complejo como KDE o Gnome, pero es bastante usable y ligero y mantiene una utilización baja de recursos.

Xfce. <http://www.xfce.org/> Es un entorno de escritorio ligero para sistemas tipo Unix como Linux, BSD, Solaris y derivados. Diseñado para la productividad, las aplicaciones se cargan y ejecutan rápidamente, mientras conserva los recursos del sistema.

Conclusiones

Con el desarrollo teórico de este trabajo se ha conocido la tecnología Thin Client y se ha estudiado las ventajas que presenta.

Con el trabajo práctico realizado se ha profundizado y se han puesto en común, muchos de los conocimientos adquiridos a lo largo de los estudios de la carrera Ingeniería en Informática, puesto que se ha entrado en diversos temas como: Sistemas Operativos, Sistemas Operativos empotrados, Linux, Sistemas de Ficheros, Hardware, Drivers, Virtualización, Redes, Shellscripts, Herramientas de compilación, Protocolos de Comunicaciones, Instalación y configuración de Servidores, Interfaces de red, etc. por lo que su realización ha resultado ser interesante y atractiva.

Bibliografía

Para el desarrollo del trabajo se ha extraído información de diversas fuentes en internet:

<http://en.wikipedia.org>

<http://es.wikipedia.org>

<http://www.google.com>

<http://www.monografias.com>