



**Faculdade de Tecnologia de Americana
Curso de Segurança da Informação**

UMA ABORDAGEM DE IMPLANTAÇÃO DA TECNOLOGIA THIN CLIENT

EDUARDO ARAÚJO SILVA

**Americana, SP
2011**



**Faculdade de Tecnologia de Americana
Curso de Segurança da Informação**

UMA ABORDAGEM DE IMPLANTAÇÃO DA TECNOLOGIA THIN CLIENT

EDUARDO ARAÚJO SILVA

duasti@gmail.com

Trabalho de conclusão de curso desenvolvido em cumprimento à exigência curricular do Curso de Análise de Sistemas e Tecnologia da Informação: Habilitação em Segurança da Informação da Faculdade de Tecnologia de Americana, sob orientação do Prof. Msc. Rossano Pablo Pinto.

**Americana, SP
2011**

BANCA EXAMINADORA

Prof. Msc. Rossano Pablo Pinto (Orientador)

Prof. Msc. José Aparecido Carrilho (Convidado)

Prof. Irineu Ambrozano Filho (Presidente da banca)

AGRADECIMENTOS

Em primeiro lugar agradeço a Deus e a minha família pela força e apoio que sempre me deram.

Ao meu orientador prof. Msc. Rossano Pablo Pinto que me indicou o caminho pelo qual deveria seguir para a realização deste trabalho.

Ao meu colega de trabalho Júnior, pois sua ajuda foi fundamental para a implementação da parte prática deste trabalho.

RESUMO

Este trabalho apresenta a tecnologia *thin client* utilizada em ambientes Linux e Windows, bem como as configurações de hardware necessárias e alguns benefícios que esta tecnologia pode trazer em uma rede corporativa. O trabalho é finalizado com o estudo de caso de uma escola que resolveu implantar o sistema *thin client* em sua biblioteca, usando o WTS em conjunto com a distribuição Linux Thinstation.

Palavras chave: *thin client*, LTSP, WTS.

ABSTRACT

This essay presents the thin-client technology in both Windows and Linux platforms, the necessary hardware configuration and the benefits in using this technology in corporate computer networks as well.

Also, this work shows a study case about a school that deployed the WTS thin-client technology and the Thinstation Linux distribution in it's library.

Keywords: thin client, LTSP, WTS.

LISTA DE SIGLAS

ICA – Independent Computing Architecture
LTSP – Linux Terminal Server Project
PXE – Preboot Execution Environment
RDP – Remote Desktop Protocol
TFTP – Trivial File Transfer Protocol
VNC – Virtual Network Computing
WTS – Windows Terminal Server
XDMCP – X Display Manager Control Protocol

LISTA DE FIGURAS

Figura 1: <i>Mainframe</i> acessado por terminais via rede.	13
Figura 2: Funcionamento do sistema X11.....	21
Figura 3: Exemplo do funcionamento dos canais RDP.	24
Figura 4: Rede formada por servidor WTS, com os respectivos serviços, sendo acessado pelos <i>thin clients</i> via protocolo PXE.	35
Figura 5: Tela de inicialização de um <i>thin client</i>	36
Figura 6: Área de trabalho padrão do Thinstation.....	38
Figura 7: Trecho do script <code>xtdesk.functions</code> configurado.	39
Figura 8: Trecho do arquivo <code>preferences</code> configurado.	39
Figura 9: Tela final, depois de feita todas as configurações necessárias.	40
Figura 10: Ferramenta de configuração do servidor.....	41
Figura 11: Sistema de licenciamento dos clientes do WTS.....	42
Figura 12: Ferramenta de configuração do servidor de licença.	43
Figura 13: Ativando a ferramenta de licenciamento do WTS.	44
Figura 14: Servidor DHCP.....	45
Figura 15: Pasta <code>tftp</code> com os arquivos necessários para <i>boot</i>	47
Figura 16: Servidor TFTP.....	48
Figura 17: Ferramenta de configuração do Terminal Server.	50
Figura 18: Configuração das permissões.	51

SUMÁRIO

1	INTRODUÇÃO	9
2	PROCESSAMENTO CENTRALIZADO X PROCESSAMENTO NÃO CENTRALIZADO	12
2.1	MAINFRAME.....	12
2.2	ESTAÇÃO DE TRABALHO.....	13
2.3	MICROCOMPUTADORES (PC's)	14
2.4	DOWNSIZING.....	14
2.5	THIN CLIENT	16
3	A TECNOLOGIA THIN CLIENT	17
3.1	ARQUITETURA.....	18
3.1.1	SERVIDOR DE TERMINAIS	18
3.1.2	CLIENTE	19
3.1.3	PROTOCOLOS DE ACESSO A AMBIENTES REMOTOS	20
3.1.3.1	X11	20
3.1.3.2	CITRIX ICA	22
3.1.3.3	VNC	22
3.1.3.4	RDP	23
3.2	LTSP – LINUX TERMINAL SERVER PROJECT	24
3.2.1	SERVIDOR LTSP	26
3.2.2	CLIENTE LTSP	27
3.3	WTS – WINDOWS TERMINAL SERVER.....	28
3.3.1	SERVIDOR WTS.....	30
3.3.2	CLIENTE WTS	31
4	CENÁRIO	32
5	DETALHES DA IMPLANTAÇÃO	35
5.1	THINSTATION.....	35
5.2	HARDWARE E SOFTWARE DO SERVIDOR	40
5.3	HARDWARE E SOFTWARE DOS TERMINAIS	48
5.4	PERMISSÕES E SEGURANÇA	49
6	CONCLUSÃO.....	52
7	REFERÊNCIAS BIBLIOGRÁFICAS	54

1 INTRODUÇÃO

Nas duas últimas décadas o uso da Internet e de computadores ligados em rede têm aumentado de forma explosiva. Cada vez mais instituições como escolas, universidades, comércios, empresas e os mais variados aspectos de negócio têm feito uso da informática, o que tem feito dessa indispensável para o bom andamento dessas instituições no mundo atual.

Essa tendência fez com que o parque computacional dessas instituições crescesse consideravelmente, trazendo novos problemas.

Um desses problemas está no fato de cada novo produto lançado no mercado, como no caso de softwares, exigir computadores com poder de processamento e quantidade de memória cada vez maiores, obrigando as instituições que fazem uso de computadores a estarem sempre trocando de equipamento.

Essa situação tem levado as instituições do mundo inteiro a se perguntar: O que fazer com os computadores antigos? Um investimento vultoso em computadores pode se tornar inútil dentro de dois ou três anos pelo fato de os softwares exigirem cada vez mais recursos?

Outro problema que surgiu com o aumento do parque computacional nas instituições foi a dificuldade em se administrar e gerenciar um número cada vez maior de computadores, exigindo das instituições a contratação de mão-de-obra técnica para manutenção, formatação, instalação de sistemas operacionais e aplicativos, assim como as devidas configurações desses computadores em rede.

Uma solução encontrada por algumas instituições para resolver esses problemas, é a volta de um conceito muito utilizado na década de 70, que é a utilização de um sistema de processamento centralizado, onde se faz necessário o uso de uma única máquina mais robusta, com alto poder de processamento e maior capacidade de memória. Antigamente cabia ao *Mainframe* o papel desta máquina mais robusta. Nos dias de hoje, são utilizadas máquinas menos potentes que os *Mainframes*, mas com poder de processamento e capacidade de memória superiores às demais máquinas da rede.

A ideia consiste em fazer dessa máquina um servidor de terminais, onde ficariam instaladas todas as aplicações utilizadas pelo cliente, que por sua vez, acessaria esses serviços remotamente através da rede.

Devido ao fato das aplicações rodarem no servidor de terminais, as máquinas cliente não necessitam de um alto poder de processamento, nem de uma grande capacidade de memória e nem mesmo precisa de um HD, o que torna possível a utilização de máquinas mais antigas e consideradas obsoletas.

Por exemplo, aplicações mais modernas que exigem alto poder de processamento e capacidade de memória e que não conseguiriam ser executados em computadores com mais de cinco anos de uso, podem ser executadas através do servidor e ter apenas o resultado mostrado nos clientes.

Essas máquinas clientes são chamadas de *thin client*, terminais leves ou, até mesmo, terminais burros, devido ao fato de necessitarem apenas de pequenos programas para poderem inicializar (dar *boot*) através da rede. Esses programas podem ser instalados e executados por disquetes, CD/DVD ou pela própria interface de rede, desde que esta disponha de uma ROM com o programa necessário instalado.

Além disso, o uso de um servidor de terminais pode resolver um problema bastante comum em instituições que possuem muitas máquinas ligadas em rede, que é instalação e atualização de softwares. Uma vez que se tem todas as aplicações instaladas em uma única máquina, não se tem a preocupação em fazer instalações e atualizações em todas as máquinas da rede. Dessa forma, cabe aos administradores e gerentes da rede se preocupar somente com as licenças, pois cada máquina conectada no servidor de terminais necessita de uma licença. A manutenção, também, ficaria restrita somente ao hardware e à infra estrutura da rede.

O objetivo desse trabalho é analisar a tecnologia *thin client*. Do surgimento da ideia de se usar o processamento centralizado até os tipos de sistemas existentes hoje em dia para implantação de *thin clients*. Para isso, o trabalho foi dividido da seguinte forma:

No Capítulo 2 é apresentada a evolução que ocorreu nos computadores, desde a década de 70, com o *Mainframe*, até os dias atuais com os *thin clients*.

O Capítulo 3 mostra o que é necessário para a implantação de um sistema de *thin clients*. Descreve os tipos de protocolos existentes, os protocolos mais utilizados, as configurações de hardware necessárias e os dois sistemas mais utilizados: LTSP e o WTS, dando-se uma ênfase maior ao WTS.

O Capítulo 4 faz um estudo de caso da Escola Polivalente de Americana, a qual está implantando o sistema de *thin clients* em sua biblioteca. Sendo assim, o capítulo descreve o cenário e os problemas que levaram a adoção de tal solução.

O Capítulo 5 detalha a implantação do sistema de *thin clients* no estudo de caso descrito no Capítulo 4.

E, por fim, no Capítulo 6 é feita a conclusão do trabalho, onde são apresentados alguns benefícios, trazidos com a implantação desta tecnologia, para o ambiente em questão, assim como alguns pontos negativos encontrados devido à escolha feita.

2 PROCESSAMENTO CENTRALIZADO X PROCESSAMENTO NÃO CENTRALIZADO

Embora pareça ser uma solução nova para questões como reutilização de equipamentos antigos e facilidade de gerenciamento de equipamentos e usuários, a ideia de processamento centralizado é antiga e teve início juntamente com a difusão do uso de computadores por parte das grandes corporações, instituições financeiras e universidades, com o intuito de tornar a resolução de problemas mais rápida e eficaz.

Porém, as transformações ocorridas na forma de gestão das grandes corporações e a rápida e constante evolução dos computadores, fizeram com que a ideia de processamento centralizado fosse esquecida por um determinado tempo, aparecendo novamente nos dias de hoje de forma ampliada.

Esse capítulo faz uma breve descrição do ciclo sofrido pelo uso do processamento centralizado nas últimas décadas, bem como das máquinas que fizeram parte deste ciclo.

2.1 MAINFRAME

Durante os anos 60 e 70, os CPD's (Centros de Processamento de Dados) reinaram absolutos, sendo esta época conhecida como a Era do CPD.

Esses CPD's eram ambientes aclimatizados e preparados para abrigarem computadores que ocupavam toda uma sala, conhecidos como *Mainframes*.

Mainframe é um computador de grande porte, dedicado normalmente ao processamento de um grande volume de informações e capaz de oferecer serviços de processamento a milhares de usuários através de milhares de terminais conectados diretamente a ele, ou através de uma rede [6].

Naquela época, todos os programas e dados ficavam centralizados nessas super-máquinas, as quais eram acessadas por terminais, conhecidos como terminais burros. Isso porque possuíam pouca ou nenhuma capacidade de processamento e de armazenamento, e somente executavam o sistema administrativo da empresa. A Figura 1 mostra o esquema de um *Mainframe* acessado por terminais [10].

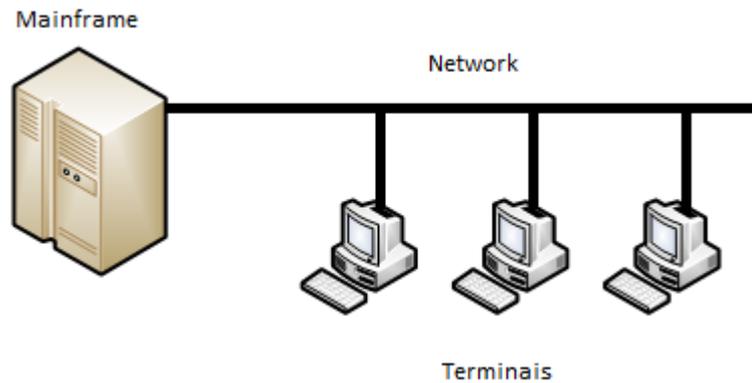


Figura 1: *Mainframe* acessado por terminais via rede.

Fonte: <http://arcanjo.org/voltaremos-a-era-dos-mainframes>

Vale lembrar que durante o ciclo evolutivo da informática, ainda se tem o minicomputador que fora desenvolvido e utilizado paralelamente aos *Mainframes*, embora fosse um pouco mais novo.

Os minicomputadores foram os primeiros na escala evolutiva da informática a serem beneficiados com o avanço da eletrônica, pois com o advento dos circuitos integrados e dos chips, tornou-se possível a construção de máquinas menores e com poder de processamento igual ou até mesmo superior ao de alguns computadores de grande porte.

Também eram mais baratos que os *Mainframes*, de modo que cada departamento de uma empresa ou universidade poderia ter seu próprio minicomputador.

Para a história da arquitetura do computador os minicomputadores tiveram uma importância significativa, pois os microcomputadores, os atuais PC's, tiveram sua arquitetura baseada na arquitetura dos minicomputadores, ou seja, a arquitetura dos microcomputadores se assemelha mais com a arquitetura dos minicomputadores do que com a arquitetura dos *Mainframes* [13].

2.2 ESTAÇÃO DE TRABALHO

Seguindo a evolução, no início dos anos 80, os minicomputadores foram sendo substituídos pelas estações de trabalho (*Workstations*), tornando-se menores, ocupando cada vez menos espaço e, ao mesmo tempo, mais poderosos no que se refere ao aumento do poder computacional e memória.

Essas máquinas eram otimizadas para a visualização e manipulação de diferentes tipos de dados complexos e por isso eram voltadas para a execução de tarefas pesadas, em geral na área científica ou industrial, como computação matemática complexa, projetos de engenharia, processamento de imagem etc.

Em nível de processamento, as estações de trabalho encontravam-se abaixo dos *Mainframes* e dos minicomputadores e acima dos atuais PC's.

Começou, nesse período, a se pensar na descentralização do processamento, passando esta tarefa para as estações de trabalho.

Hoje, o termo estação de trabalho é entendido de uma forma mais genérica, ou seja, pode ser usada para se referir a qualquer tipo de computador ligado em rede, desde microcomputadores até terminais conectados ao *Mainframe*.

2.3 MICROCOMPUTADORES (PC's)

A derrocada do processamento centralizado e, conseqüentemente, do *Mainframe* se deu, de fato, com a popularização do microcomputador ou PC (*Personal Computer* - Computador Pessoal), no início dos anos 80.

Aos poucos, os grandes *Mainframes* que exigiam pessoal altamente especializado para operá-los e para fazer sua manutenção, começaram a ser substituídos por máquinas servidoras de aplicação.

As redes de terminais burros ligadas ao *Mainframe* foram substituídas pelo que se conhece hoje como modelo cliente-servidor, onde pode-se ter vários microcomputadores como clientes ligados à computadores mais potentes, os servidores, os quais podem fornecer diversos tipos de serviços.

Este processo de substituição das redes de terminais burros conectados aos *Mainframes* pelo modelo de rede formado por máquinas clientes conectadas à máquinas servidoras, foi batizado de *Downsizing* [6].

2.4 DOWNSIZING

“O *downsizing* é uma das técnicas empregadas para tornar a empresa ágil e competitiva, e normalmente é a primeira ferramenta utilizada para iniciar processos de horizontalização nas empresas e reestruturação dos recursos humanos” [17].

Tendo seu início entre o final dos 80 e começo dos anos 90, o *downsizing* tinha como principal objetivo enxugar e desburocratizar as empresas.

Na área de TI o *downsizing* se deu com a substituição do processamento centralizado, composto pelos *Mainframes* e terminais burros, pelo modelo cliente-servidor [6].

Os *Mainframes* passaram a ser vistos como um excesso de burocracia da infra estrutura das empresas, pesando contra eles o fato de ocuparem grandes áreas dentro da empresa, terem um alto consumo de energia, além de necessitarem de pessoal altamente especializado. Esse pessoal não era bem visto pelos executivos das empresas, pois acabavam tendo um papel importante nas tomadas de decisão, além de serem vistos como um gasto a mais para as empresas.

Sobre a alegação por partes dos executivos e administradores das grandes corporações de que quanto menor o nível hierárquico em que as divisões são tomadas, melhores serão seus resultados, ou seja, quanto menor o número de pessoas envolvidas nas tomadas de decisão, mais fácil se torna a chegada a um ponto comum, começou-se a procura por soluções que tornassem a manipulação da informação menos complicada aos olhos dos administradores das empresas e que não necessitassem de tantas pessoas para manipulá-las [4].

Para que isso ocorresse era essencial a criação de sistemas que fossem pequenos em termos de tamanho, porém grandes em termos de desempenho e capacidade.

Com hardwares cada vez menores, mas dotados de processadores mais velozes e memórias com maior capacidade de armazenamento, os microcomputadores que compunham o modelo cliente-servidor, se tornaram capazes de executar aplicativos cada vez mais robustos e menos complicados para os usuários comuns. Isso fez com que o modelo cliente-servidor tivesse grande aceitação por parte das instituições, decretando, momentaneamente, a extinção do *Mainframe*.¹

¹ Segundo a revista Isto É Dinheiro de setembro de 2010, estima-se que atualmente no mundo deve haver em torno de dez mil *Mainframes* em uso, sendo as instituições financeiras seus maiores usuários. Ainda segundo a revista, no Brasil os *Mainframes* representaram 30% do mercado de servidores durante o ano de 2009. Em países como EUA, China, Alemanha e Canadá esta proporção esteve entre 10% e 15% no mesmo período, e segue caindo a cada ano.

2.5 THIN CLIENT

“Um *thin client* (cliente magro) é um computador cliente em uma rede de modelo cliente-servidor de duas camadas que tem poucos ou nenhum aplicativo instalados, de modo que depende primariamente de um servidor central para o processamento de atividades. A palavra "thin" refere-se a uma pequena imagem de *boot* que tais clientes tipicamente requerem - talvez não mais do que o necessário para fazer a conexão com a rede e iniciar um navegador web dedicado ou uma conexão de "Área de Trabalho Remota" tais como X11, Citrix ICA ou Microsoft RDP” [19].

A tecnologia *thin client* foi projetada para facilitar o trabalho do administrador de TI, o qual pode conectar os microcomputadores a um servidor remoto e rodar todas as aplicações a partir de um único ponto. O principal objetivo dessa tecnologia é centralizar os dados, o gerenciamento e a administração do ambiente de microcomputadores, reduzir a complexidade do parque de TI e reaproveitar equipamentos considerados obsoletos.

São muito úteis em instituições que buscam facilidade no gerenciamento dos aplicativos e usuários do sistema, manutenção fácil e reaproveitamento de equipamentos com um custo total relativamente baixo.

A ideia por trás da tecnologia *thin client* nada mais é do que uma volta aos tempos do processamento centralizado. Porém, não mais fazendo uso do *Mainframe* e de terminais burros, mas sim de servidores de terminais os quais podem ser acessados por microcomputadores.

3 A TECNOLOGIA *THIN CLIENT*

Conforme mencionado no Capítulo 1 a ideia por trás do *thin client* não é nova.

Sendo baseada nos antigos sistemas de processamento centralizado, trata-se de uma rede formada por computadores antigos ou com pouco poder de processamento e baixa capacidade de memória, os quais enviam requisições via teclado e/ou mouse para o servidor que, por sua vez, irá processar as informações e retornar o resultado para que estes computadores somente o exibam no monitor.

A principal diferença entre o sistema usado há vinte ou trinta anos atrás e o sistema usado hoje em dia está no fato de não mais se usar o *Mainframe* para a execução de todos os serviços e tarefas dos usuários, mas sim no uso de máquinas menos potentes que os *Mainframes*, chamadas servidores. Esses são dotados de sistemas operacionais multiusuário capazes de suportar vários usuários conectados à máquina através de sessões independentes (modelo conhecido como cliente-servidor).

Mas afinal, por que mudar de computação centralizada para descentralizada e voltar novamente?

Segundo Minasi [12], apesar do crescimento do uso dos microcomputadores ter sido impulsionado pelo avanço tecnológico, principalmente no que diz respeito a hardware e aplicativos comerciais, o aumento dos parques computacionais e da complexidade das redes deixou claro que o modelo baseado em processamento centralizado tinha algumas vantagens a oferecer que uma rede baseada em microcomputadores no modelo cliente-servidor não oferece:

- “Agrupamento de recursos de computação para certificar-se de que nenhum é desperdiçado.
- Distribuição e manutenção de aplicações centralizadas.
- Clientes que não necessitam executar o sistema operacional mais recente e melhor com o hardware mais recente e melhor para suportá-lo.
- Máquinas clientes que não requerem proteção para economia de energia porque não executam nenhuma aplicação localmente” [12].

Esses são alguns motivos que fizeram com que o processamento centralizado nunca deixasse de existir e começasse a ganhar força novamente nos dias atuais, tornando-se mais uma opção para as instituições, as quais podem ajustar seus

modelos de computação de acordo com suas necessidades ambientais e administrativas.

3.1 ARQUITETURA

Antes de continuar o estudo da tecnologia *thin client*, é preciso fazer uma observação. Muitos autores de livros e sites especializados no assunto tratam o *thin client* como sendo um console (hardware) específico para determinadas atividades. Porém, o uso de microcomputadores com pouco poder de processamento e baixa capacidade de memória, como máquinas mais antigas, por exemplo, usadas para os mesmos fins que os consoles, também podem ser considerados *thin clients*.

A arquitetura de uma rede com *thin clients* é composta por:

- Servidor de terminais;
- Máquinas clientes usadas como *thin client*;
- Protocolos de comunicação.

3.1.1 SERVIDOR DE TERMINAIS

O servidor de terminais é um computador com bom poder de processamento e boa capacidade de memória que possibilita o acesso remoto à área de trabalho do usuário através de um software *thin client*. Esse software permite o compartilhamento de recursos do servidor com computadores clientes, e os faz funcionar como emuladores de terminal.

Em circunstâncias normais de uso, um computador passa a maior parte do tempo ocioso. Devido a isso, um servidor de terminal consegue compartilhar o uso do processador com varias máquinas, pois dificilmente todas as máquinas clientes executarão aplicativos que exijam muito processamento por muito tempo, tendo apenas alguns instantes de picos de processamento.

A memória RAM também é compartilhada. Um aplicativo é carregado uma única vez na memória, mesmo que esteja sendo usado por vários usuários ao mesmo tempo. O que o servidor de terminais faz é abrir diversas sessões do mesmo aplicativo, o que ainda torna o carregamento de programas mais rápido.

Quando um *thin client* solicitar uma conexão com o servidor de terminais e este aceitar, será criada uma sessão entre ambos. Através desta sessão, o servidor de terminais transmitirá a interface de usuário ao *thin client*, de modo que cada usuário que faça logon no sistema tenha acesso apenas à sua sessão individual. As

sessões são gerenciadas de forma transparente pelo sistema operacional do servidor de terminais e independente de qualquer outra sessão de *thin client*. Isto faz com que o usuário tenha a impressão de estar logado na máquina local [12] [14].

Com o uso de um servidor de terminais, todas as execuções, processamento e armazenamento de dados ocorrem no servidor, de modo a se centralizar a implantação dos programas. Isso assegura que todos os clientes poderão acessar a mesma versão de um programa, pois o software é instalado apenas uma vez no servidor, e não em todas as máquinas da rede [20].

3.1.2 CLIENTE

Uma rede formada por computadores que quase não necessitam de manutenção, fáceis de gerenciar e que consomem pouca energia seria o ideal para reduzir o tamanho, complexidade e o custo com o parque computacional das instituições.

Essa rede, quase ideal, pode ser fácil de se conseguir utilizando máquinas clientes conhecidas como *thin client* ou clientes leves.

O *thin client* envia entradas de teclado e mouse para o servidor de terminais que, por sua vez, fará todo o trabalho pesado. Após isso, o servidor de terminais devolve ao *thin client* todas as tarefas processadas para que este somente mostre o resultado no monitor.

Devido ao fato das aplicações rodarem no servidor de terminais, as máquinas usadas como *thin clients* não necessitam de um alto poder de processamento, nem de uma grande capacidade de memória e nem mesmo de um HD, o que faz com que sejam conhecidos como terminais leves.

Necessitam apenas de pequenos programas para poderem inicializar (dar *boot*) através da rede, conhecidos como Etherboot, os quais podem ser instalados e executados por disquetes, CD/DVD ou *pen drive*. Também é possível carregar o sistema pela própria interface de rede, desde que esta disponha de uma ROM que possua o Etherboot. Placas-mãe mais recentes possibilitam o *boot* pela rede usando um protocolo desenvolvido pela Intel conhecido como PXE (*Preboot Execution Environment* – Ambiente de Execução Pré-inicialização), bastando apenas que se configure a BIOS do *thin client* [14].

3.1.3 PROTOCOLOS DE ACESSO A AMBIENTES REMOTOS

Existem diversos protocolos que permitem que microcomputadores se conectem a um servidor de terminais via rede, sendo os mais conhecidos e usados: X11, Citrix ICA, VNC e Microsoft RDP.

Esta seção faz uma análise destes protocolos. Porém, por este trabalho ser dedicado ao estudo de uma implantação de um sistema *thin client* em ambiente Windows, o protocolo RDP será visto com mais detalhes, pois o mesmo é usado pelo Terminal Server do Windows Server 2003.

3.1.3.1 X11

O X11, X Windows ou simplesmente X é um sistema desenvolvido pelo MIT (*Massachusetts Institute of Technology* – Instituto de Tecnologia de Massachusetts) usado para controlar dispositivos mapeados em bits como monitor, teclado e mouse. Ele permite a exibição de gráficos por parte dos aplicativos e possibilita a esses desenhar na tela pixels, linhas, textos, imagens etc. Pode ser visto como parte do sistema operacional que controla o terminal.

“O sistema é formado por um núcleo (núcleo X) que contém todos os drivers de dispositivos específicos de terminal, o que o torna totalmente dependente de hardware, sendo o responsável por controlar a tela e capturar os eventos oriundos do teclado e do mouse. Para isso, o núcleo X disponibiliza às aplicações uma interface no formato de uma biblioteca denominada Xlib” [18].

Este esquema é mostrado na Figura 2.

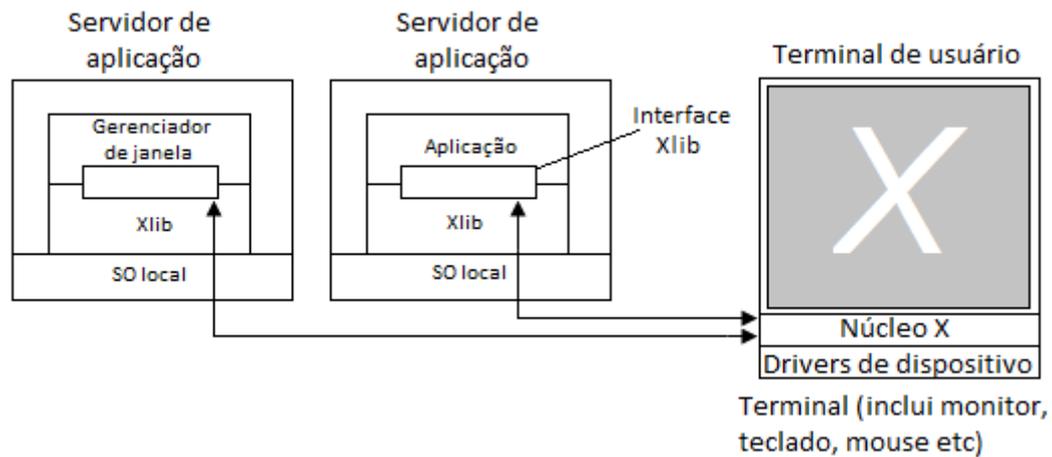


Figura 2: Funcionamento do sistema X11.

Fonte: Tanenbaum;Steen [18]

Um ponto importante, principalmente para a ideia de *thin client*, é que o núcleo X e as aplicações X não precisam necessariamente residir na mesma máquina. O núcleo X fornece o protocolo X, que é um protocolo de comunicação da camada de aplicação pelo qual uma instância de Xlib pode trocar dados e eventos com o núcleo X.

“Por exemplo, Xlib pode enviar requisições ao núcleo X para criar ou encerrar uma janela, estabelecer cores e definir o tipo de cursor a exibir, entre muitas outras requisições. Por sua vez, o núcleo X reagirá a eventos locais como entrada de teclado e mouse devolvendo pacotes de eventos a Xlib” [18].

O núcleo X também permite que várias aplicações se comuniquem com ele ao mesmo tempo. Mas existe uma aplicação específica que recebe direitos especiais, conhecida como gerenciador de janela, que é quem determina a aparência geral do visor, ou seja, como esse se apresenta ao usuário. Por exemplo, como os botões aparecerão na janela, como as janelas devem ser colocadas no visor etc.

Tudo isso faz com que o X11 seja caracterizado como um modelo cliente-servidor, onde um cliente X se conecta a um servidor X e envia seus pedidos utilizando o protocolo da biblioteca X (Xlib).

Porém, um aspecto interessante do protocolo X11 é que o núcleo X, que faz o papel de servidor, reside na máquina utilizada como *thin client*, enquanto a

aplicação, que faz o papel de cliente, reside na máquina usada como servidor de terminais.

O X11 é padrão para o sistema gráfico no mundo UNIX, fazendo parte do Linux, BSD e da grande parte dos sistemas UNIX comerciais. Alguns de seus ambientes de trabalho mais usados são: Gnome, KDE e Xfce.

3.1.3.2 CITRIX ICA

O Citrix ICA (*Independent Computing Architecture* – Arquitetura de Computação Independente) é um protocolo proprietário projetado pela Citrix System para passar dados entre servidores e estações de trabalho cliente. O ICA captura os eventos do teclado e do mouse do lado cliente e envia para o servidor que, por sua vez, processa as informações e as manda de volta para que sejam exibidas no monitor do lado cliente. Dessa forma, ele permite transferir o processamento das aplicações do cliente para o servidor [15].

Os clientes remotos ICA interagem com servidores que permitem aos usuários ver e trabalhar com todas as interfaces de aplicação, mas com as aplicações sendo executadas no servidor.

O software que faz a função de servidor do *thin client* é o MetaFrame, que pode trabalhar tanto com a plataforma Windows quanto com a plataforma UNIX, fornecendo uma extensão para o Terminal Server baseado no ambiente Windows da Microsoft e serviços remotos para clientes UNIX.²

3.1.3.3 VNC

O VNC (*Virtual Network Computing* – Rede de Computação Virtual) é um software de acesso remoto que permite visualizar e interagir totalmente com um computador (o "servidor VNC") usando um programa simples (o "VNC Viewer") em outro computador que se encontra em qualquer lugar na Internet ou rede local.

Os dois computadores não precisam estar rodando, necessariamente, o mesmo sistema operacional. Assim, por exemplo, pode-se usar o VNC para visualizar uma máquina que esteja rodando Windows em um outro computador que esteja rodando Linux ou Mac. Para maior simplicidade, há ainda um visualizador em

²"A versão atual do Microsoft Terminal Server é um subconjunto dos serviços terminais oferecido pelo Citrix MetaFrame. A Microsoft co-desenvolveu seu servidor de terminais com a Citrix (com a Citrix fazendo a parte do leão). A solução da Citrix é mais abrangente como resultado do negócio elaborado entre Microsoft e a Citrix". (Rob Scrimger; Paul Lasalle; Mridula Parihar, 318)

Java, de modo que qualquer computador pode ser controlado remotamente de dentro de um navegador sem ter que instalar o software [3].

Embora o VNC permita o acesso remoto entre computadores distintos através de uma rede, ele necessita de um sistema operacional para que possa ser executado.

3.1.3.4 RDP

O RDP (*Remote Desktop Protocol* – Protocolo de Área de Trabalho Remota) é o protocolo de exibição nativo do Windows Terminal Server da Microsoft. Assim como o protocolo X e o Citrix ICA, o RDP é o responsável por entregar as imagens gráficas do servidor de terminais para o cliente, e transferir a entrada feita pelo teclado e mouse do cliente para o servidor.

O RDP permite que mais de uma aplicação, a partir de um cliente, se conecte ponto a ponto com o servidor. Desse modo, pode-se ter múltiplas sessões a partir de um único cliente e até mesmo múltiplas sessões para um único usuário.

“Um recurso usado pelo RDP denominado *client-side caching*, permite ao cliente “lembrar-se” de imagens que já foram baixadas durante a sessão. Com o *caching*, somente as partes modificadas de tela são baixadas para o cliente durante cada renovação. Por exemplo, se o ícone do Microsoft Word já tiver sido baixado para o cliente, não haverá necessidade de ele ser baixado novamente, uma vez que a imagem da área de trabalho será atualizada. Quando o cache fica cheio, ele se desfaz dos dados que não foram utilizados há mais tempo, em favor de novos dados” [12].

A imagem na tela é atualizada cerca de 20 vezes por segundo quando uma sessão está ativa. Se o servidor ficar sem receber entradas de teclado ou mouse, esta taxa de renovação é diminuída para 10 vezes por segundo, até que o envio de entradas seja retomado.

Entendendo os canais RDP

Segundo Minasi [12], as aplicações de uma sessão se comunicam com o computador cliente através de canais, que são como links, interligando o cliente ao servidor de terminais.

São os canais que determinam como os clientes interagem com o servidor de terminais e, por isso, em ambos os lados os canais devem estar ativados para que o protocolo de exibição, no caso o RDP, possa funcionar.

O RDP tem espaço para um grande número de canais fazerem diferentes coisas, e cada versão do RDP possibilita um número maior deles. A capacidade do protocolo RDP é diretamente proporcional à quantidade de canais que ele suporta, de modo que quanto mais canais ele suportar, maior o número de informações que ele irá transportar entre o cliente e o servidor de terminais. Um exemplo do funcionamento dos canais RDP pode ser visto na Figura 3.

Porém, quanto mais canais ele usa, mais largura de banda da rede ele consome. Em uma rede com muitos computadores e com um tráfego intenso de pacotes, o uso de um número grande de canais pode se tornar inviável.



Figura 3: Exemplo do funcionamento dos canais RDP.

Fonte: Minasi [12]

3.2 LTSP – LINUX TERMINAL SERVER PROJECT

O LTSP (*Linux Terminal Server Project* – Projeto de Servidor de Terminal Linux) é um conjunto de softwares que transforma uma distribuição Linux normal em um servidor de terminais.

Alguns serviços são essenciais para que o LTSP possa funcionar. Portanto, segue uma breve descrição destes serviços:

- DHCP – o servidor DHCP é o responsável por fornecer as informações da rede, indicar o kernel ou cliente PXE e indicar em qual local compartilhado está o sistema a ser carregado pelo *thin client*.

- TFTP – o servidor TFTP é o responsável por transferir a imagem inicial do *boot* e o kernel para que estes possam ser carregados no *thin client*.
- NFS – o servidor NFS é o responsável por disponibilizar aos *thin clients* o diretório onde encontram-se as informações necessárias ao LTSP, de modo que estas sejam carregadas durante o processo de *boot*.
- XDMCP – o servidor XDMCP é o responsável por mostrar no monitor do *thin client* as informações oriundas do servidor depois de processadas, e por levar as entradas de teclado e mouse do cliente para o servidor.

É a combinação de todos estes serviços que torna possível ao LTSP executar aplicativos, processar informações e mostrar os resultados nos *thin clients*.

Segundo Morimoto [14], depois de todos os serviços essenciais estarem instalados e funcionando, o LTSP cria um sistema de arquivo dentro do diretório `/opt/ltsp/i386`, o qual conterá um conjunto de pacotes que será compartilhado na rede e acessado, via NFS, pelos *thin clients* como se fosse uma partição local.

Depois disso, todo o processo ocorre da seguinte forma:

- O *thin client* inicializa dando *boot* pela imagem de *boot* instalada na ROM (ou da placa-mãe, no caso de placas de rede *onboard*, ou pela própria interface de rede, caso esta a possua ou, ainda, pelo próprio HD, caso se tenha optado por instalar um sistema operacional e o configurado para funcionar como *thin client*), ou pelo CD/DVD.
- O software de *boot* envia um pacote de *broadcast* pedindo a configuração da rede. O servidor DHCP responde, enviando as configurações juntamente com as informações do kernel e do diretório onde se encontra a instalação do LTSP no servidor.
- O kernel, então, é carregado, via TFTP, para o *thin client* dando início ao *boot* real, onde será feita a substituição do TFTP pelo NFS.
- será montado, via NFS, o diretório `/opt/ltsp/i386` do servidor como diretório raiz do *thin client*, o qual irá carregar o sistema LTSP que se encarrega de detectar o hardware do *thin client* e abrir uma sessão do X.
- Terminado o *boot*, o XDMCP envia ao *thin client* a tela de login.

Deste ponto em diante o servidor de terminais roda os aplicativos e o *thin client* apenas mostra as imagens geradas na tela.

3.2.1 SERVIDOR LTSP

A configuração de hardware de um servidor LTSP dependerá da quantidade de *thin clients* conectados a ele, e dos tipos de aplicativos que serão usados pelos *thin clients*.

Por exemplo, se os *thin clients* usarem o servidor apenas para acessar a internet ou trabalhar com editores de texto, a configuração de hardware do servidor não necessitaria ser tão robusta quanto se fosse usado para jogos 3D ou softwares manipuladores de imagem e desenho.

A seguir, algumas diretrizes que devem satisfazer à maioria dos casos:

Memória

“A fórmula que é usada para adicionar memória em um servidor de terminais com LTSP é:

$$256 + (192 \times N) \text{ MB}$$

onde N é o número de *thin clients* que se deseja conectar ao servidor de terminais” [11].

Por exemplo, se o objetivo é montar uma rede com vinte máquinas *thin client* conectadas ao servidor de terminais, a quantidade de memória requerida pelo servidor será:

$$256 + (192 \times 20) = 4096 \text{ MB} \sim 4.1 \text{ GB.}$$

Porém, Morimoto [14] afirma que para atender a dez máquinas *thin clients* são necessários no mínimo 512 MB de memória RAM, mas que o ideal seria começar com um 1GB.

Processador

A escolha do processador também dependerá dos tipos de aplicativo que serão utilizados pelos *thin clients*. Por exemplo, aplicativos baseados em Java e Flash, consomem mais processamento do que simples aplicativos editores de texto.

Para uma rede onde se fará uso de diversos aplicativos, como programas editores de imagem, navegadores de Internet, editores de texto etc., um processador

de 2 GHz deve ser capaz de suportar vinte máquinas *thin client*, mas com pequenos atrasos. O ideal é utilizar um processador de 3 GHz ou superior.

Para redes maiores, com trinta máquinas *thin client* ou mais, um servidor com processador dual-core Xeon ou superior seria mais apropriado. Ou ainda um servidor com múltiplos processadores [9].

Para Morimoto [14], um servidor com processador Pentium III ou Athlon é capaz de atender a dez máquinas *thin client*.

HD

No que diz respeito ao HD, o ideal é ter uma combinação de vários discos formando um RAID, pois isso aumentaria a performance no acesso e gravação dos dados.

Dependendo do tipo de RAID a segurança dos dados também estaria garantida, uma vez que os dados são guardados em todos os discos que compõem o RAID. No caso de se haver alguma falha em um dos discos, basta que se substitua somente o disco avariado [11] [14].

3.2.2 CLIENTE LTSP

Embora os *thin clients* utilizem os recursos do servidor de terminais, como processador e memória, a configuração do hardware destas máquinas não deve ser deixada de lado.

Memória

A quantidade de memória para que um *thin client* funcione é de 48 MB. Porém, esta quantidade de memória o tornaria consideravelmente lento, sendo que a quantidade de memória recomendada é de 128 MB a 256 MB para um bom desempenho [11].

Já Morimoto [14] afirma que a capacidade de memória do *thin client* poder ser de 8 MB e a quantidade ideal de 32 MB ou superior.

Processador

O clock recomendado para o processador do *thin client* é de 553 MHz ou superior [11].

Porém, Morimoto [14] afirma que é possível utilizar um processador 486 DX-100 (50 MHz ~ 100 MHz) sendo recomendável um Pentium 100 ou superior.

HD

O uso de um HD no *thin client* não é obrigatório, mas pode ser usado caso se queira instalar um sistema operacional e fazer o acesso ao servidor de terminais usando algum aplicativo de acesso remoto. Neste caso, o HD deve ter a capacidade necessária para a instalação do sistema operacional.

3.3 WTS – WINDOWS TERMINAL SERVER

Como surgiu

O Windows Terminal Server (WTS) da Microsoft é um serviço que permite que mais de um usuário use o sistema operacional ao mesmo tempo [7].

A tecnologia por trás do WTS começou a ser desenvolvida na década de 80 pela Citrix, sendo tal tecnologia conhecida como Citrix MultiWin, que era conjunto de extensões para o Windows NT.

O primeiro produto usando tal tecnologia foi o WinFrame para Windows NT, sendo que muitos recursos usados no Windows Server 2003, incluindo impressoras e acessos a drivers, foram originalmente desenvolvidos no WinFrame [7].

Em 1997 a Microsoft licenciou a tecnologia MultiWins da Citrix, passando a incorporá-la nas versões do Windows desde então. O Windows Server 2003 foi a primeira versão do Windows a trazer a tecnologia MultiWin como parte do sistema operacional.

É atualmente conhecido como Terminal Server, e se tornou um dos serviços disponíveis no Windows Server 2003, ganhando uma série de novos recursos, tanto do lado cliente quanto do lado servidor [12].

Como funciona

Segundo Minasi [12], no Windows Server 2003 e seu sucessor Windows Server 2008, o Terminal Server, quando habilitado, é inicializado juntamente com o sistema operacional, passando a escutar a porta TCP 3389 para saber se há pedidos de conexão de clientes chegando.

Ainda segundo o autor, o funcionamento do Terminal Server pode ser explicado da seguinte forma:

Quando um *thin client* solicita uma conexão com o servidor de terminais e este aceita o pedido, é aberta uma sessão. Cada sessão possui um identificador chamado *SessionID*, o qual é usado para diferenciar os processos que rodam em sessões diferentes no mesmo computador.

Para que as sessões sejam abertas o mais rápido possível, o Terminal Server cria duas sessões inativas.

Quando um cliente estabelece uma conexão com as sessões inativas são criadas mais sessões inativas, de modo que sempre existirão sessões prontas para conectar.

Após estabelecida a conexão entre o servidor de terminais e o *thin client*, o *Virtual Memory Manager* irá gerar uma nova *SessionID* para a sessão recém criada. Esta *SessionID* será, então, repassada para o *Session Manager* tão logo o *SessionSpace*³ para essa sessão seja criado.

Independente da quantidade de aplicações usadas, toda sessão executa os seguintes processos:

- *Win32 Subsystem* (Subsistema Win32) – necessário para executar aplicações Win32.
- *User Authentication Module* (Módulo de Autenticação do Usuário) – processo de logon responsável por capturar a informação de nome do usuário e a senha, e repassá-la para o subsistema de segurança para autenticação.
- *Executable Environment for Applications* (Ambientes Executáveis para Aplicações) – todas as aplicações de usuários Win32 e máquinas DOS virtuais rodam no contexto do Shell do usuário.

Cada sessão aberta possui sua própria cópia do subsistema Win32, que criará uma única área de trabalho e instâncias únicas que a suportam. Possui também sua própria cópia da aplicação WinLogon, que autentica a identidade do usuário.

As sessões são independentes entre si, de modo que os processos executados em uma determinada sessão não interferem nos processos que estão sendo executados em outra sessão. Porém, todas as sessões utilizam os mesmos recursos da máquina como tempo de processador, memória, funções do sistema

³ O termo *SessionSpace* se refere ao espaço de memória alocado para uma determinada sessão.

operacional etc. Por isso é necessário que o sistema operacional divida o uso desses recursos ao mesmo tempo em que os mantém separados.

Para fazer tal divisão o servidor de terminais usa o identificador de processos e o identificador da respectiva sessão onde está sendo executado o processo.

A quantidade de sessões que um servidor de terminais é capaz de suportar depende apenas das limitações de hardware, como capacidade de memória, processador, largura de banda da rede e acesso a disco.

Quando um cliente efetua logout da sessão, os canais virtuais para sua máquina se fecham e os recursos alocados para a sessão são liberados.

Para o lado do *thin client*, uma sessão se torna ativa quando um usuário efetua logon de maneira bem sucedida, ou seja, quando o usuário consegue executar as aplicações desejadas usando o servidor de terminais.

Durante uma sessão, as entradas feitas pelo usuário com cliques do mouse e toques no teclado é atualizada no servidor por intermédio de um canal virtual. Os comandos para transferir os bitmaps que exibem a interface são baixados para o cliente por outro canal virtual.

3.3.1 SERVIDOR WTS

Para a configuração do hardware de um servidor WTS, vale as mesmas observações feitas para o servidor LTSP, ou seja, depende da quantidade de máquinas *thin client* que irão se conectar ao servidor e do tipo de aplicativo que os usuários irão precisar.

Seguem algumas diretrizes gerais que servem como base para uma configuração.

Memória

Um servidor de terminal WTS exige no mínimo 128 MB de memória RAM para funcionar, porém a Microsoft recomenda um mínimo de 256 MB, além de memória adicional para cada usuário. Para saber a quantidade de memória necessária para cada usuário, recomenda-se adicionar 10 MB a mais de memória para usuários que utilizam somente uma aplicação por vez; para usuários que fazem uso de três ou mais aplicativos por vez, recomenda-se o acréscimo de 21 MB a mais para cada usuário [8].

Segundo Minasi [12], a quantidade de memória depende das aplicações que usadas, do número de sessões simultâneas e das demandas por memória dos arquivos abertos nessas sessões.

Ainda, segundo o autor, pelo fato da memória ser quem limita o poder do servidor, e não o processador, elas devem ser de alta velocidade e com correção de erros. Quanto mais memória deste tipo o servidor possuir, melhor será seu desempenho, sendo recomendado o mínimo de 1 GB.

Processador

A Microsoft recomenda que o processador do servidor que irá rodar o WTS tenha um clock de 733 MHz ou superior [9].

Porém, Minasi [12] afirma que mais importante que a velocidade é a capacidade da memória cache do processador, de modo que ela seja suficiente o bastante para que o processador não necessite buscar informações na memória RAM, que é mais lenta.

O referido autor também afirma que o uso de multiprocessadores ajudaria a diminuir o gargalo no servidor.

HD

O HD deve ter a capacidade mínima para a instalação do Windows Server 2003 que é de 1,5 GB mais a capacidade requerida por cada aplicativo instalado no servidor.

3.3.2 CLIENTE WTS

A ideia inicial da Microsoft era fazer com que o acesso ao servidor WTS se desse através de um aplicativo de acesso remoto, o qual seria executado por uma máquina com um sistema Win32 instalado (Windows 98, NT, 2000, XP). Neste caso, a configuração de hardware do cliente deve ser a mesma necessária para a instalação do sistema operacional.

Porém, no caso de se usar uma versão Linux mais leve para que os *thin clients* acessem o servidor de terminais sem a necessidade de instalação de um sistema operacional, as mesmas considerações que foram feitas em relação à configuração de hardware dos clientes LTSP, são feitas para os clientes WTS.

4 CENÁRIO

Conforme mencionado nos capítulos anteriores a tecnologia *thin client* pode ser utilizada tanto para o reaproveitamento de computadores antigos quanto para facilitar o gerenciamento do uso de computadores, uma vez que todo o processamento estará centralizado em uma única máquina.

Com o objetivo de demonstrar as utilidades de um sistema baseado em *thin clients*, este capítulo utilizará como exemplo a escola Polivalente de Americana, a qual pretende adotar este sistema para solucionar alguns problemas, que são descritos a seguir.

A escola Polivalente de Americana conta hoje com aproximadamente quatrocentos computadores, os quais estão distribuídos entre diversos departamentos, como laboratórios, secretaria, salas de professores, salas da administração (direção, coordenação e RH), gráfica, guarita e biblioteca.

Devido a problemas de depredação e de mau uso dos computadores, o uso dos laboratórios passou a ser restrito somente às aulas e com a presença de um professor responsável, cabendo aos alunos o uso dos computadores da biblioteca, quando não estiverem em horário de aula.

Isto fez da biblioteca o mais crítico de todos os departamentos da escola no que se refere ao uso de computadores devido aos seguintes problemas:

- Uso inadequado dos computadores por parte de alguns alunos que ao invés de fazer uso dessas máquinas para realização de pesquisas e trabalhos didáticos, as utilizam para jogar e acessar páginas de relacionamento como Orkut, Facebook, Twitter etc., sendo isto o motivo de várias reclamações por parte de professores e de outros alunos que necessitam utilizar os computadores para realização de trabalhos.
- Máquinas que param de funcionar devido a problemas no sistema operacional, sendo muitos destes problemas causados por vírus, que são instalados nas máquinas através de pen drives dos alunos.
- Reclamação por parte dos alunos que precisam fazer seus trabalhos escolares sendo que para isto necessitam fazer uso de alguns programas específicos como Corel Draw e que não foram instalados devido à configuração do hardware das máquinas não os suportarem ou não os executarem corretamente.

Junta-se a todos estes problemas o fato da equipe responsável por manter todo o parque computacional da escola funcionando ser composta por um número pequeno de pessoas, cabendo a estes, também a parte de suporte aos usuários, o que torna ainda mais difícil fazer o controle e gerenciamento de toda a rede.

Diante do cenário exposto acima, coube aos responsáveis pelo parque computacional da escola buscar uma solução que resolvesse todos os problemas relacionados ao uso dos computadores da biblioteca.

Primeiramente, pensou-se na instalação de programas (como o *Deep Freeze*) que “congelam” o estado da máquina em um ponto que esta esteja funcionando corretamente. Desse modo, mesmo que se fizesse alguma alteração na configuração da máquina, após a reinicialização tudo voltaria ao mesmo estado de antes, onde foi feito o “congelamento”. Porém, essa ideia não foi concretizada porque não resolveria o problema de mau uso no que se refere ao acesso a páginas de relacionamento e jogos, sem contar no problema que se poderia ter no futuro quanto a licenciamento de software, uma vez que a maioria dos softwares que realizam esta tarefa e que atenderia as necessidades neste caso, são proprietários e necessitam que se compre suas respectivas licenças.

Outra solução proposta foi a implantação de um servidor de gerenciamento de rede como o Nagios. Porém, esta ideia também foi descartada pelo administrador da rede, que temia por um aumento no tráfego da rede, o que poderia atrapalhar no funcionamento da secretaria e nas aulas do laboratório.

Então, foi proposto a implantação de um servidor de terminais, o qual poderia ser acessado via terminais leves, ou seja, que não precisavam ser muito potentes e nem necessitariam de instalação de sistema operacional. Esta ideia foi a mais plausível, pois atenderia às principais necessidades da biblioteca:

- Facilidade no gerenciamento de uso dos computadores, uma vez que os acessos se dariam via servidor, o controle de acesso à Internet seria feito somente nesta máquina.
- Problemas de sistema operacional corrompido por causa de vírus, também deixariam de ocorrer, pois o uso de pen drives não mais seria possível, de modo que os alunos acessariam seus arquivos somente do servidor de arquivos.
- O uso de programas específicos considerados pesados para a configuração dos computadores da biblioteca seria possível já que todos os programas

seriam instalados e executados no servidor, sendo que este deve possuir uma configuração mais robusta. Atualmente, o uso destes softwares só pode ser feito nos laboratórios, os quais possuem máquinas mais novas.

A partir daí, foi escolhido qual sistema seria implantado, tendo-se no início duas opções: o LTSP, baseado em ambiente Linux e WTS, baseado em ambiente Windows.

Pelo fato da maioria dos programas usados nos diversos cursos da escola ser baseado em ambiente Microsoft Windows e devido a grande parte dos alunos e professores não estarem habituados com o ambiente Linux, optou-se pela implantação do WTS. Pesou, também, para esta escolha o fato da escola ter um convênio com a Microsoft, sendo que desta forma, todos os computadores que acessarem o servidor, estariam operando de forma legal e licenciada.

A ideia principal desde o início era fazer com que as máquinas acessassem o servidor sem a necessidade de um sistema operacional instalado. Porém, o WTS normalmente é acessado via conexão remota, o qual é um serviço disponível somente em sistemas operacionais.

Para que a ideia inicial fosse mantida, foi escolhida uma distribuição Linux extremamente leve e desenvolvida para *thin clients* conhecida como Thinstation. Com ele é possível que os computadores da biblioteca da escola dêem *boot* pela rede e carreguem os arquivos necessários do sistema operacional via servidor, onde estão alocados.

No próximo capítulo, será apresentado um detalhamento de como foi feita a implantação do sistema *thin client* na biblioteca da escola, baseados no WTS e no Thinstation, assim como as configurações necessárias.

5 DETALHES DA IMPLANTAÇÃO

Este capítulo descreve como a solução WTS foi implantada no cenário descrito no Capítulo 3, bem como a configuração do hardware das máquinas utilizadas como servidor WTS e *thin clients*, e o processo de instalação e configuração dos serviços necessários para o funcionamento do sistema, conforme mostra a Figura 4.

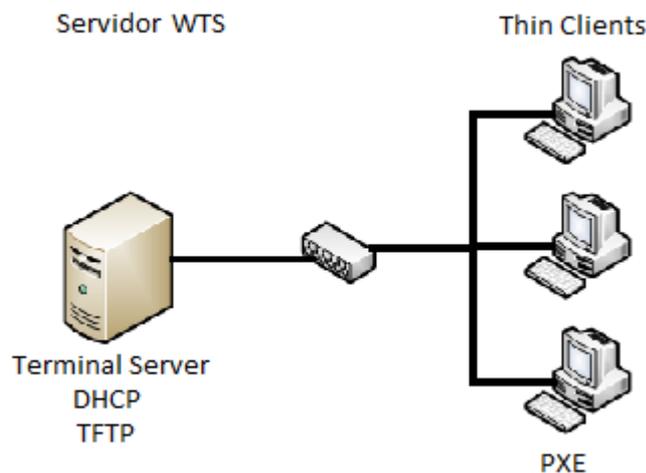


Figura 4: Rede formada por servidor WTS, com os respectivos serviços, sendo acessado pelos *thin clients* via protocolo PXE.

Fonte: Autor.

5.1 THINSTATION

Conforme explicado no Capítulo 3, a ideia inicial quando se pensou em implantar um sistema de *thin clients* para solucionar os problemas encontrados na biblioteca do Polivalente, era fazer com que as máquinas daquele local operassem sem a necessidade de instalação de um sistema operacional, e a solução encontrada foi a implantação do Thinstation para que este pudesse trabalhar junto com o WTS.

Thinstation⁴ é uma distribuição Linux extremamente leve desenvolvida para ser usada em terminais leves. A versão escolhida e que foi usada na implantação foi o Thinstation-2.2.d.

No caso em estudo, o Thinstation em conjunto com outros serviços como DHCP e TFTP rodando no servidor, possibilitam que as máquinas clientes acessem o servidor de terminais, sem a necessidade de instalação de um sistema operacional.

⁴ Thinstation pode ser baixado do site: <http://www.thinstation.org>

Para isto, são necessários quatro arquivos básicos:

- pxelinux.0 → é o responsável por carregar o kernel e o arquivo initrd.
- pxelinux.cfg/default → possui as instruções que serão executadas pelos terminais ao carregarem o arquivo pxelinux.0 [14].
- vmlinuz → kernel.
- initrd → sistema de arquivos usado pelo kernel.

O funcionamento ocorre da seguinte forma:

- Quando a máquina cliente é inicializada, o protocolo PXE envia pacotes de requisição na rede, por broadcast. O servidor DHCP responde, enviando um endereço IP e o nome do arquivo pxelinux.0 à máquina cliente.
- O protocolo PXE envia outra requisição, desta vez solicitando ao servidor TFTP a localização do arquivo pxelinux.0 e do kernel.
- O servidor TFTP responde, enviando os arquivos solicitados.
- Ao ser carregado, o kernel solicita o arquivo initrd ao servidor TFTP, que o envia [5].

Esta sequência pode ser verificada na Figura 5.

```
CLIENT MAC ADDR: 08 00 27 26 34 8F  GUID: FAC9E105-4DB4-4AA3-AAE3-965664676423
CLIENT IP: 172.16.101.28  MASK: 255.255.255.0  DHCP IP: 172.16.101.24

PXELINUX 3.09 2005-06-17  Copyright (C) 1994-2005 H. Peter Anvin
UNDI data segment at: 0009C8D0
UNDI data segment size: 1830
UNDI code segment at: 0009E100
UNDI code segment size: 19BB
PXE entry point found (we hope) at 9E10:0104
My IP address seems to be AC10651C 172.16.101.28
ip=172.16.101.28:172.16.101.24:0.0.0:255.255.255.0
TFTP prefix:
Trying to load: pxelinux.cfg/01-08-00-27-26-34-8f
Trying to load: pxelinux.cfg/AC10651C
Trying to load: pxelinux.cfg/AC10651
Trying to load: pxelinux.cfg/AC1065
Trying to load: pxelinux.cfg/AC106
Trying to load: pxelinux.cfg/AC10
Trying to load: pxelinux.cfg/AC1
Trying to load: pxelinux.cfg/AC
Trying to load: pxelinux.cfg/A
Trying to load: pxelinux.cfg/default
Loading vmlinuz.....
Loading initrd.....
```

Figura 5: Tela de inicialização de um *thin client*.

Fonte: Autor.

Após todo esse processo, os módulos referentes aos drivers da máquina cliente são carregados.

Os arquivos pxelinux.0 e default se encontram dentro da pasta Thinstation-2.2.d/boot-images/pxe, após esta ser descompactada. Já os arquivos initrd e vmlinuz

só estarão disponíveis dentro da mesma pasta após a compilação do Thinstation. Esta compilação se dá digitando o comando `./build` dentro da pasta `Thinstation-2.2.d`.

Porém, antes da compilação, alguns arquivos devem ser configurados para se adaptar o Thinstation ao uso que se necessita fazer e de acordo com a rede.

No caso do Polivalente, primeiramente foi necessário configurar o arquivo `Thinstation-2.2.d/build.conf` para que os módulos referentes às placas de rede das máquinas clientes ficassem disponíveis, e configurar o layout do teclado para o padrão brasileiro (`pt_br`) `abnt2`.

A segunda configuração necessária foi a forma com que os usuários acessariam a tela de logon do Terminal Server. Depois de carregado, o Thinstation tem a aparência do Windows XP com alguns ícones instalados na área de trabalho, conforme ilustrado na Figura 6. Estes ícones quando clicados é que dão a opção de se conectar a algum servidor ou outro computador via endereço IP, sendo que este deve ser digitado pelo usuário.

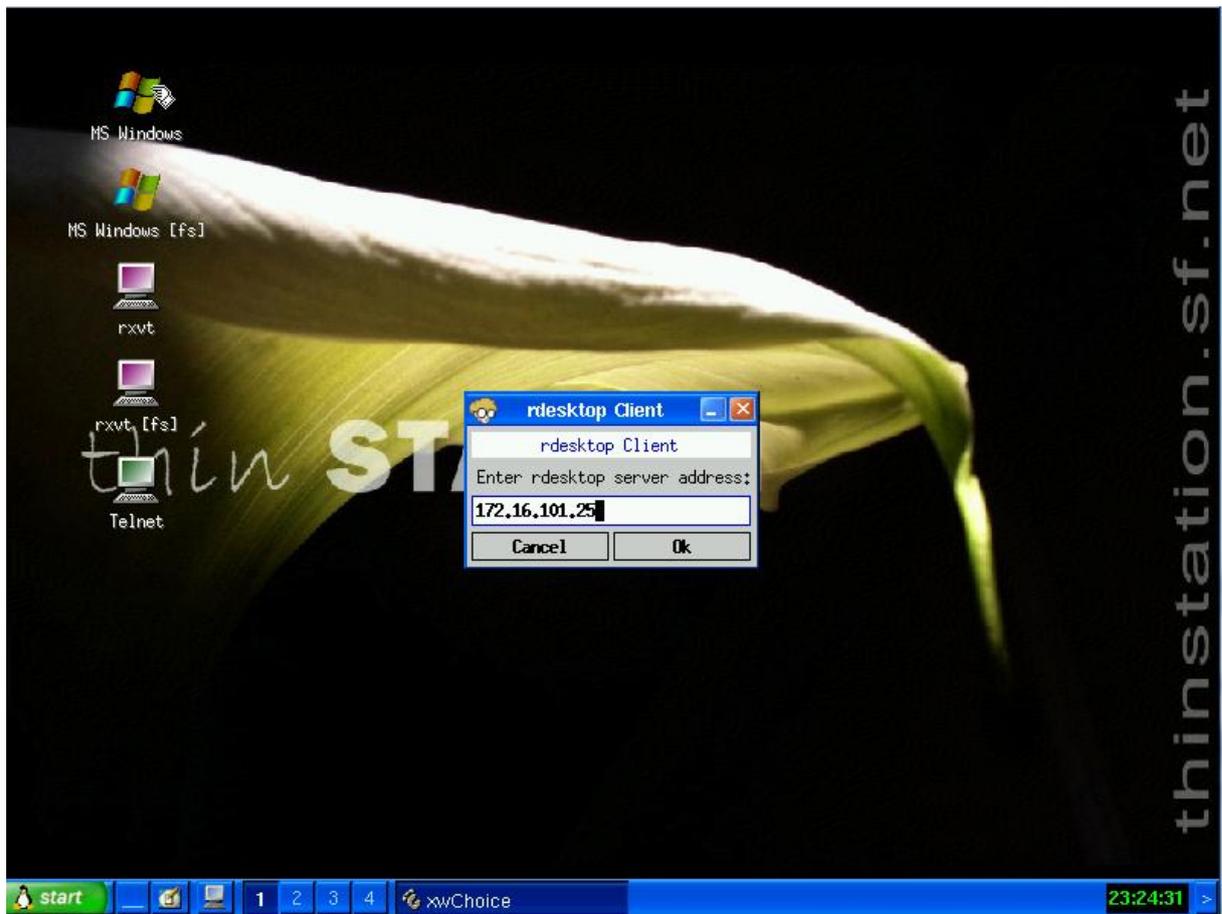


Figura 6: Área de trabalho padrão do Thinstation.

Fonte: Autor.

Como a maior parte dos usuários que utilizam os computadores da biblioteca são usuários comuns, ou seja, com conhecimentos básicos em informática, essa forma padrão do Thinstation de acessar o servidor de terminal não seria viável. Sem contar que alguns ícones permitem o acesso a outros tipos de serviço como Telnet, o que poderia trazer problemas futuros.

Para resolver esse problema o script `Thinstation-2.2.d/packages/xtdesk/etc/xtdesk.functions` foi configurado de forma que somente um ícone ficasse visível na área de trabalho e com o nome desejado, de modo que bastasse clicar sobre este ícone para que o usuário fosse levado à tela de logon do Terminal Server [16].

A Figura 7 mostra a configuração feita.

```

do
  case "$entry" in
    \#*|"") continue ;;
  esac
  echo $entry > /tmp/entry
  . /tmp/entry
  ICON=/usr/icons/`replace_char $package " " "_`_32x32.xpm"
  if [ -e $ICON ] ; then
    echo "table Icon">$HOME/.xtdesktop/Logar.lnk
    echo " Type: Program">>$HOME/.xtdesktop/Logar.lnk
    echo " Caption: Logar">>$HOME/.xtdesktop/Logar.lnk
    echo " Command: pkg fullscreen rdesktop 172.16.101.24">>$HOME/.xtdesktop/Logar.lnk
    echo " Icon: /usr/icons/rdesktop_32x32.xpm">>$HOME/.xtdesktop/Logar.lnk
    echo " X: 132">>$HOME/.xtdesktop/Logar.lnk
    echo " Y: $y">>$HOME/.xtdesktop/Logar.lnk
    echo "end">>$HOME/.xtdesktop/Logar.lnk
    let y=y+64
    if [ $y -gt $maxy ] ; then
      let x=x+100
      let y=40
    fi
    echo "y=$y; x=$x" > /tmp/entry
  fi
done
rm $XTMENU
fi

```

Figura 7: Trecho do script xtdesk.functions configurado.

Fonte: Autor.

A terceira configuração necessária foi com relação a acessos a recursos do Thinstation. Foi verificado que na barra de tarefas havia uma opção disponível para acessar o terminal Shell do Thinstation. Isso poderia se tornar um problema caso alguns usuários, com um pouco mais de conhecimento em informática, tivessem acesso a este terminal e resolvessem fazer algo com intuito de prejudicar o funcionamento da rede da escola.

Este problema foi resolvido com a configuração do arquivo Thinstation-2.2.d/packages/icewm/etc/icewm/preferences, como mostrado na Figura 8.

```

# Workspace switches continuously when moving mouse to screen edge
# ContinuousEdgeSwitch=1 # 0/1

# Reload menu files automatically
# AutoReloadMenus=1 # 0/1

# Show task bar
ShowTaskBar=0 # 0/1

# Task bar at top of the screen
# TaskBarAtTop=0 # 0/1

```

Figura 8: Trecho do arquivo preferences configurado.

Fonte: Autor.

Isso fez com que a barra de tarefas fosse retirada, ficando a aparência final, depois de todas as configurações necessárias, como a mostrada na Figura 9.



Figura 9: Tela final, depois de feita todas as configurações necessárias.

Fonte: Autor.

5.2 HARDWARE E SOFTWARE DO SERVIDOR

Hardware

- Processador – Pentium Core 2 Duo E7500 2.93 GHz
- Memória RAM – 4 GB DDR2
- HD – 320 GB Sata
- Placa de rede – Realtek Gigabit Ethernet

Software

O sistema operacional instalado no servidor foi o Windows Server 2003 R2, o qual já dispõe do serviço de Terminal Server pré-instalado, bastando apenas configurá-lo.

Para que se atenda as necessidades que foram descritas no Capítulo 3, além do Terminal Server configurado e funcionando, é necessário que outros três serviços

estejam ativos no servidor para que os arquivos necessários, possam ser transferidos para as máquinas clientes . Esses serviços são:

- Servidor de licenças.
- Servidor DHCP.
- Servidor TFTP.

Terminal Server

Conforme já mencionado, embora o Terminal Server venha pré-instalado no Windows Server 2003 é necessário configurá-lo. A maneira mais fácil de se fazer essa configuração é utilizando a ferramenta “Gerenciar o servidor “ contida naquele sistema operacional, conforme mostrado na Figura 10.

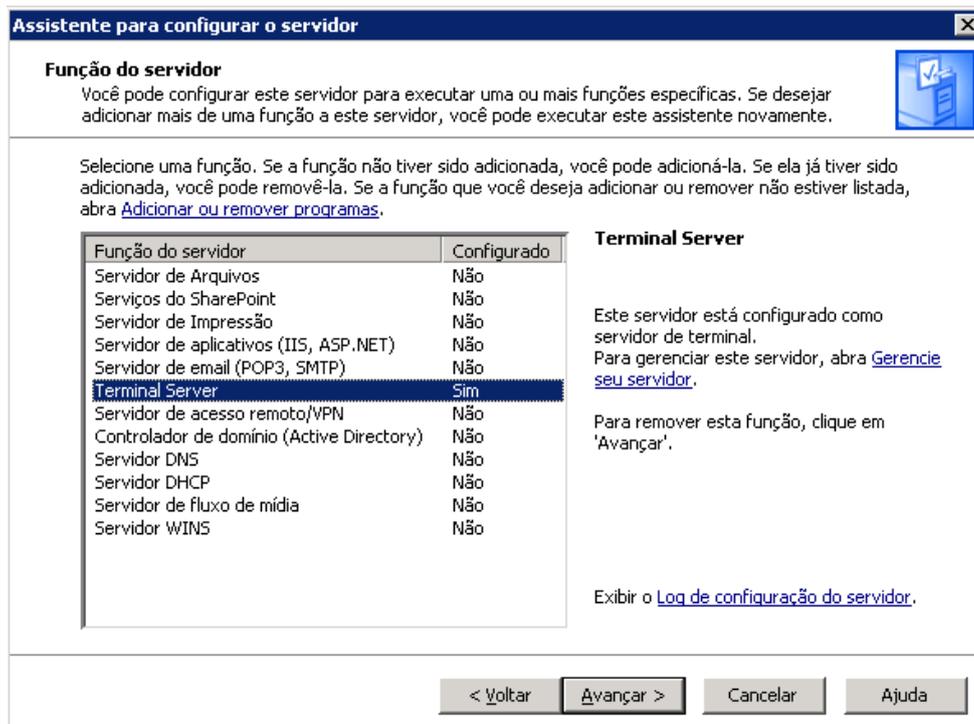


Figura 10: Ferramenta de configuração do servidor.
 Fonte: Autor.

Essa ferramenta oferece um assistente para configurar o servidor, bastando selecionar uma função e clicar em avançar para que o serviço escolhido seja configurado.

Servidor de licença

Pelo fato do Terminal Server funcionar como um emulador de área de trabalho e ser um componente do Windows Server 2003, sendo este último um software proprietário, ou seja, que necessita da compra de uma licença, para que o sistema de *thin clients* possa funcionar corretamente, é necessário que se tenha na rede um servidor de licenças.

Portanto, abaixo segue uma breve descrição de como funciona o licenciamento e a configuração do servidor que distribui as licenças.

Esse servidor de licença pode estar funcionando no próprio servidor onde esteja funcionando o Terminal Server ou em qualquer outra máquina da rede que tenha o Windows Server 2003 instalado.

Segundo Minasi [12], o licenciamento dos computadores clientes que irão acessar o servidor de terminais se dá conforme a Figura 11.

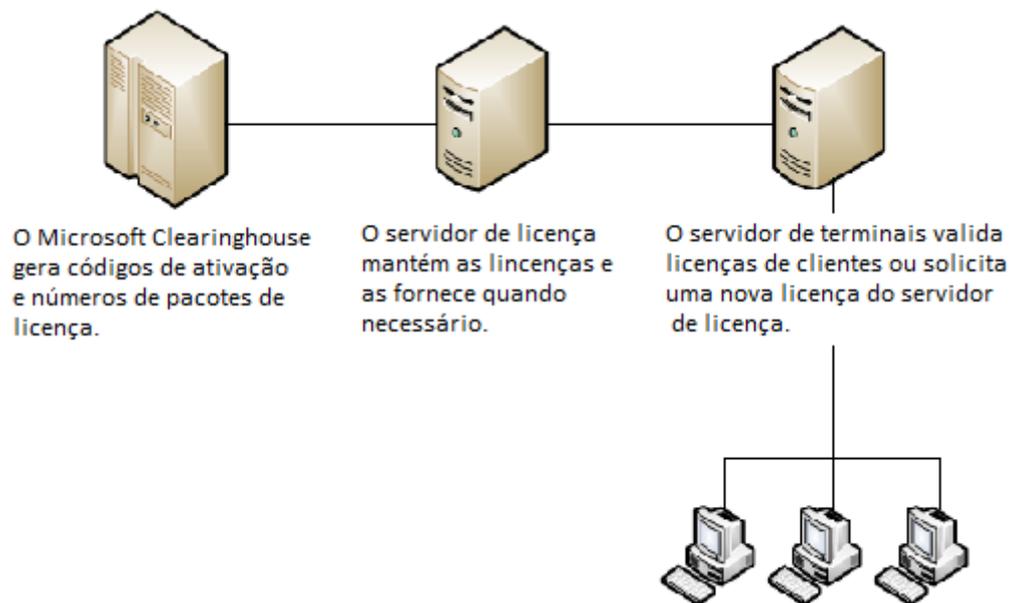


Figura 11: Sistema de licenciamento dos clientes do WTS.

Fonte: Minasi [12].

De acordo com a Figura 11, para se fazer o licenciamento são necessários:

- Servidor de terminal.
- Servidor de licença.
- Câmara de compensação da Microsoft que ativa o servidor de licença e fornece as licenças de acesso.

O Windows Server 2003, também, possui uma ferramenta para auxiliar no controle e gerenciamento de licenças, chamado Licenciamento do Terminal Server, ilustrado na Figura 12.

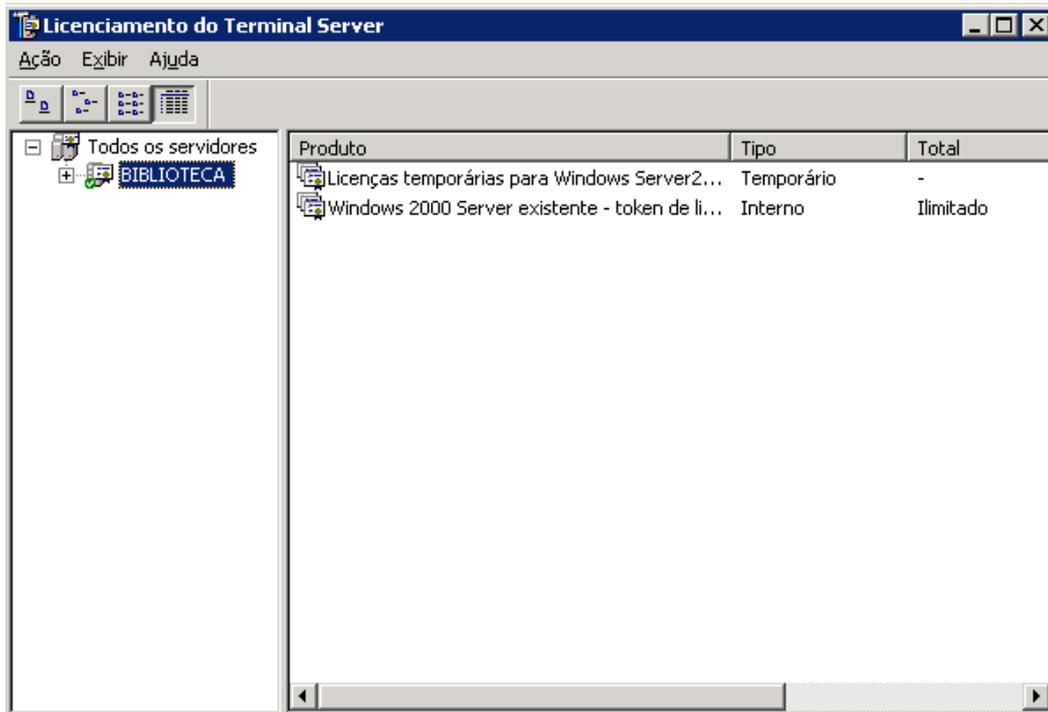


Figura 12: Ferramenta de configuração do servidor de licença.

Fonte: Autor.

Para utilizá-la é preciso adicioná-la através da opção “Adicionar ou remover programas” do Painel de Controle. Ela é encontrada clicando-se no botão Adicionar/remover componentes do Windows, conforme mostrado na Figura 13.

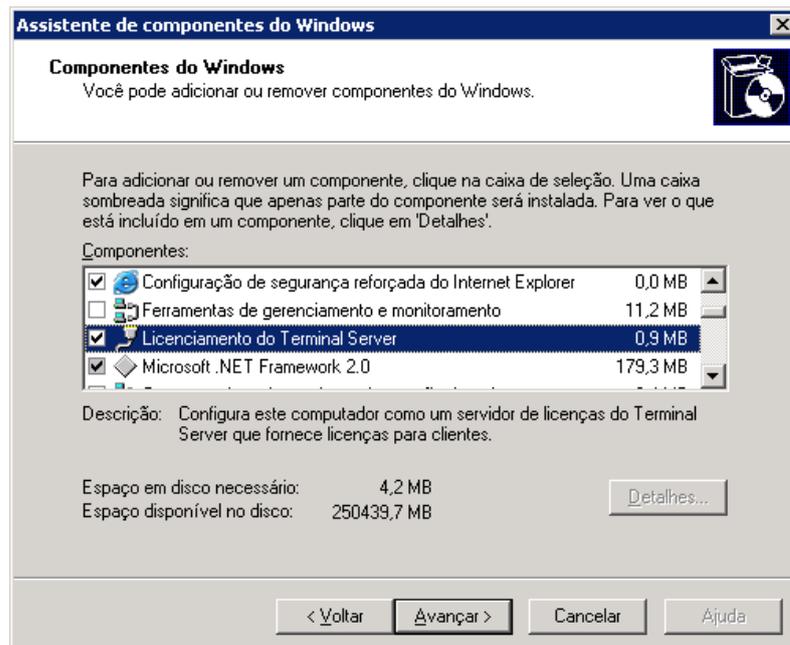


Figura 13: Ativando a ferramenta de licenciamento do WTS.

Fonte: Autor.

Após a instalação da ferramenta Licenciamento do Terminal Server, já é possível ativar o servidor de licenças. Isso é feito clicando-se com o botão direito no servidor que aparece e que se deseja ativar, e escolher a opção ativar servidor. Após isso, o assistente irá contatar a câmara de compensação de licenciamento da Microsoft e ativará o servidor (conforme a Figura 11).

Com o servidor de licenças ativado e funcionando, quando uma máquina cliente tentar acessar o servidor, este irá verificar se esta máquina possui uma licença para que assim possa liberar o acesso. No caso da máquina cliente não possuir uma licença, o servidor de terminal irá solicitar uma licença ao servidor de licença. Este, por sua vez, possuindo uma licença que possa ser liberada, irá fornecê-la para o servidor de terminais que irá repassá-la à máquina cliente.

O acesso ao servidor de terminais será negado à máquina cliente caso esta não possua uma licença e o servidor de licença estiver desativado ou inacessível.

Servidor DHCP

O servidor DHCP é o responsável por disponibilizar para as máquinas clientes as informações da rede como endereço IP e máscara de rede assim como o nome

do arquivo que o protocolo PXE usará para dar início ao carregamento do Thinstation.

Embora o Windows Server 2003 possua o serviço de DHCP, foi escolhido um software de terceiro e gratuito que realiza este serviço, chamado Tftpd32⁵.

Este software foi escolhido por ser extremamente leve, não sendo necessário sequer sua instalação na máquina, por ser de fácil configuração e por fornecer, ao mesmo tempo, o serviço de TFTP.

Ao ser iniciado o referido programa, este detecta automaticamente o endereço IP do servidor.

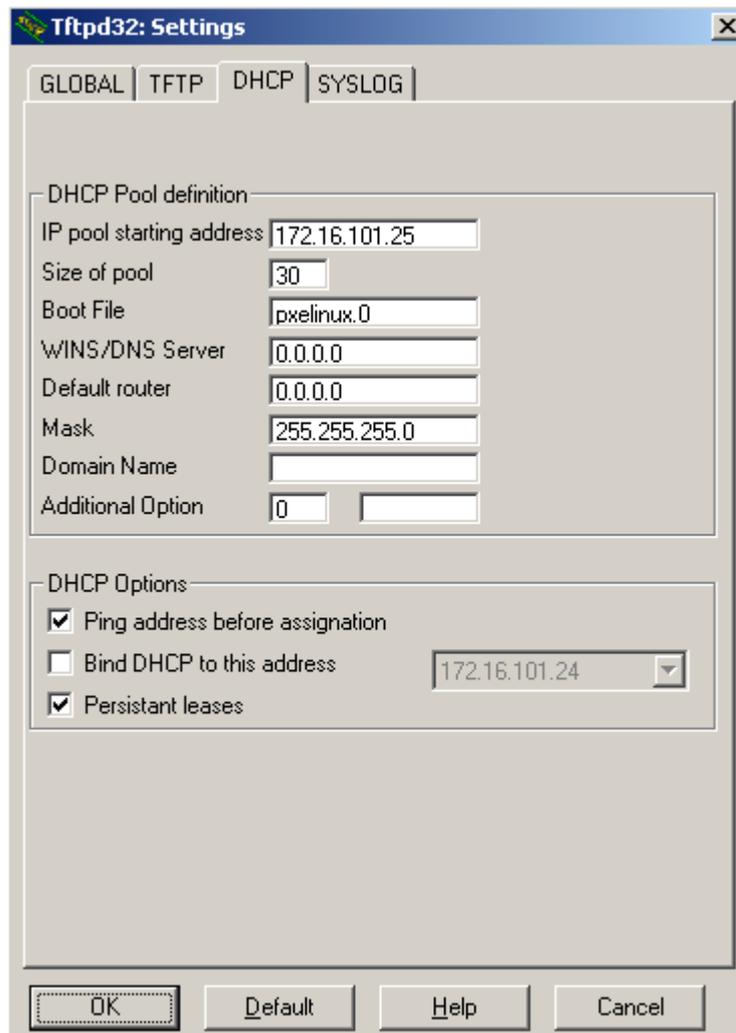


Figura 14: Servidor DHCP.

Fonte: Autor.

⁵ Tftpd32 pode ser baixado do site <http://tftpd32.jounin.net>

A Figura 14 mostra as informações que devem ser passadas ao programa para que ele repasse às máquinas clientes:

- *IP Pool Starting Address* → é o primeiro endereço da faixa de endereçamento IP que será disponibilizada para as máquinas clientes.
- *Size of Pool* → quantidade de endereços que será disponibilizada.
- *Boot File* → nome do arquivo que contém as informações referentes à imagem do sistema que será carregado pela máquina cliente, através do protocolo PXE.
- *Mask* → máscara da rede.

Servidor TFTP

Da mesma forma que no servidor LTSP, para que as máquinas clientes funcionem sem a necessidade de sistema operacional, será necessário que um servidor TFTP esteja funcionando, de forma que esse será o responsável por disponibilizar os arquivos de imagem e configuração do Thinstation para as máquinas clientes conectadas ao servidor de terminais.

Porém, como o Windows Server 2003 não dispõe deste tipo de serviço, assim como o servidor DHCP, também foi utilizado o Tftpd32.

Para a configuração do servidor TFTP, primeiramente é necessário criar uma pasta onde ficarão o arquivo executável do programa Tftpd32 e os arquivos de imagem e configuração do Thinstation, que serão enviados aos computadores *thin client*.

No caso do Polivalente de Americana, foi criada uma pasta com o nome tftp dentro de C:\, conforme mostra a Figura 15.

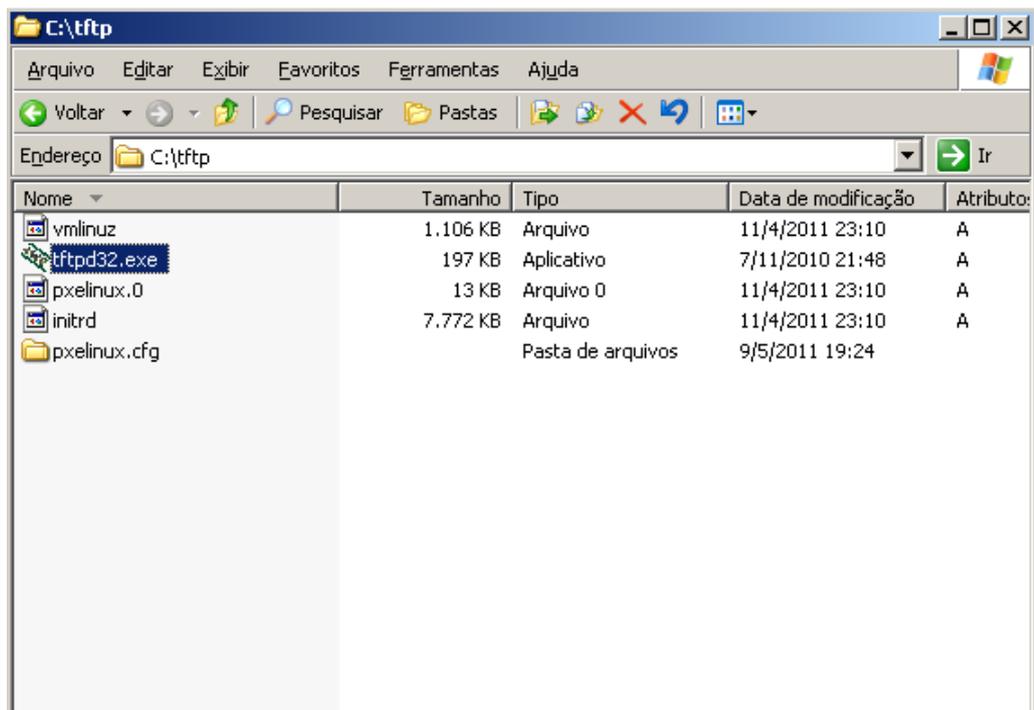


Figura 15: Pasta tftp com os arquivos necessários para *boot*.

Fonte: Autor.

Criada a pasta com o programa Tftpd32 e os arquivos necessários, é feita a configuração do servidor TFTP, com a execução do referido programa.

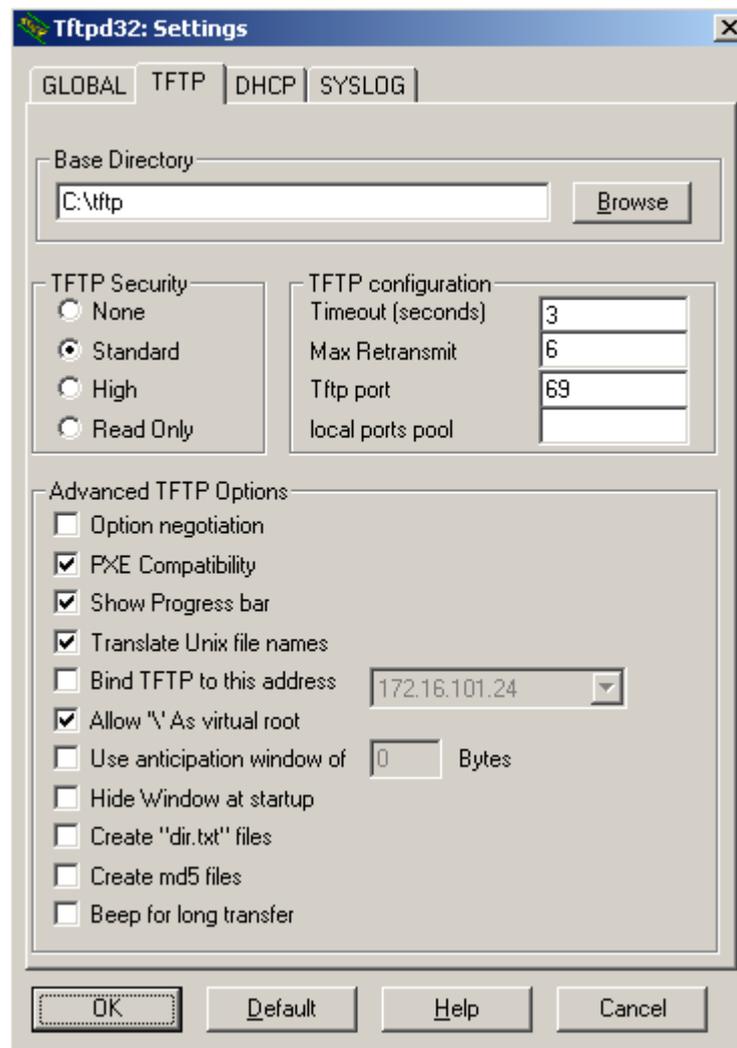


Figura 16: Servidor TFTP.

Fonte: Autor.

A Figura 16 mostra a configuração feita. No campo *Base Directory* deve-se colocar o caminho onde se encontra a pasta com os arquivos do Thinstation. No campo *Advanced TFTP Options* deve-se marcar a opção *PXE Compatibility* para que as máquinas clientes consigam acessar o servidor via PXE.

5.3 HARDWARE E SOFTWARE DOS TERMINAIS

Hardware

- Processador – Pentium 4 2.8 GHz – 1 MB de cache
- Memória RAM – 256 MB ~ 512 MB DDR2
- Placa de rede – Broadcom Gigabit Ethernet

Software

No caso em estudo, não foi necessária a instalação de nenhum tipo de software nas máquinas clientes.

As máquinas descritas acima já possuem a opção de *boot* via PXE, bastando apenas que fosse habilitada.

O PXE (*Pre-Boot Execution Environment* – Ambiente de Execução Pré-Inicialização) é um pequeno software desenvolvido pela Intel e instalado na ROM das interfaces de rede. É ele o responsável por ir buscar a imagem do Thinstation no servidor e carregá-la na memória RAM das máquinas clientes.

Segundo Morimoto [14], o PXE é capaz de carregar apenas arquivos pequenos, de no máximo 32KB e por isso precisa carregar um *Bootstrap*, que neste caso, é o arquivo pxelinux.0.

5.4 PERMISSÕES E SEGURANÇA

Depois que o Thinstation estiver carregado na memória dos *thin clients*, já é possível o acesso à tela de logon do Terminal Server.

Porém, para que se consiga logar, é necessário configurar as permissões de acesso para usuários.

O Windows Server 2003 possui uma ferramenta que possibilita fazer, de forma fácil, as configurações pertinentes ao Terminal Server, conforme mostra a Figura 17.

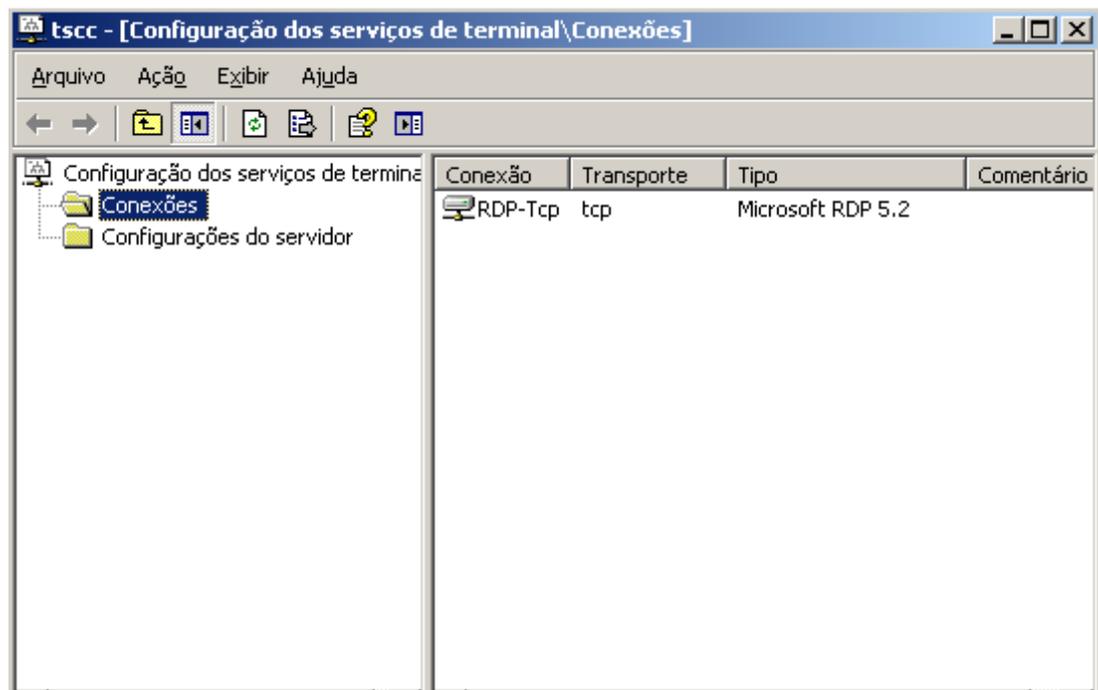


Figura 17: Ferramenta de configuração do Terminal Server.

Fonte: Autor.

Pela Figura 17 é possível visualizar a conexão RDP-Tcp. Em “propriedades” dessa conexão estão todas as configurações que podem ser feitas para controlar a forma que se darão os acessos, assim como o comportamento das sessões RDP.

Para o caso em estudo, em um primeiro momento foram configuradas somente as permissões, ficando as demais opções com a configuração padrão.

Como a máquina que abriga o Terminal Server pertence ao domínio da escola, as permissões foram configuradas de modo que todos os usuários que possuem um login consigam ter acesso ao servidor, mas com as restrições aplicadas pelo controlador de domínio.

A Figura 18 mostra como ficou a configuração das permissões.

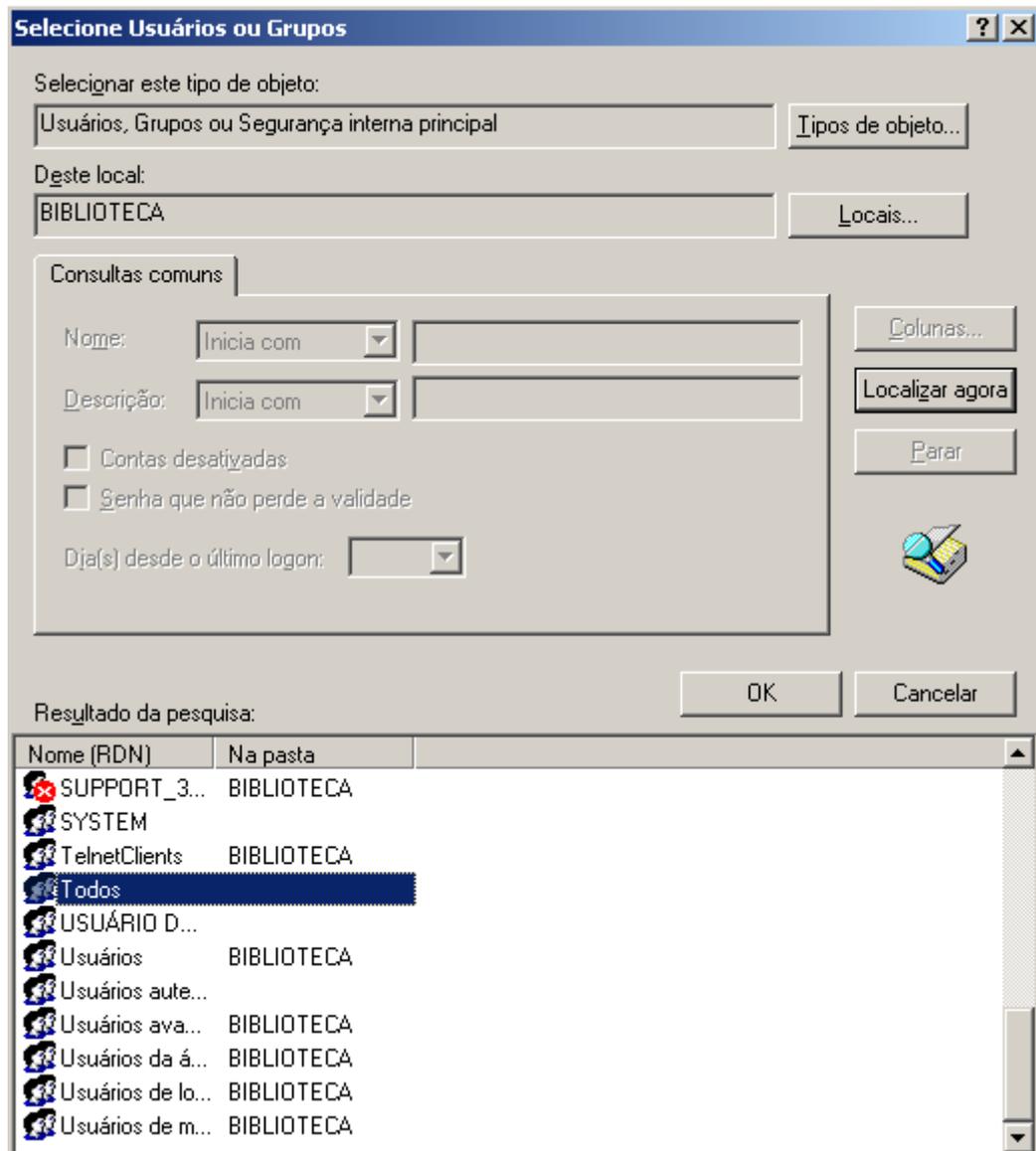


Figura 18: Configuração das permissões.

Fonte: Autor.

No que diz respeito à segurança, como o servidor faz parte do domínio, ele irá receber todas as regras e restrições já definidas no controlador de domínio para acesso e uso dos computadores da rede da escola. Deste modo, somente quem possui privilégios de administrador pode fazer alterações na configuração do servidor, assim como instalar e remover programas.

6 CONCLUSÃO

Apesar da constante evolução da tecnologia no campo da informática, onde aparecem equipamentos e softwares novos todos os dias, algumas ideias originadas e utilizadas há trinta anos, ainda estão em uso e se mostram úteis nos dias atuais.

Conforme visto, a ideia de processamento centralizado, muito utilizada na década de setenta com os famigerados *Mainframes*, ainda hoje é utilizada com os *thin clients*, os quais se mostram uma boa solução para vários problemas em redes corporativas, facilitando a administração, gerenciamento e manutenção do parque computacional das instituições.

A implantação do sistema de *thin clients* na biblioteca da escola Polivalente de Americana já apresentou uma diminuição na quantidade de computadores com problemas devido a sistemas operacionais corrompidos, o que antes era motivo de reclamações quase diárias. Também, tornou possível o uso de alguns computadores que estavam parados por falta de HD's.

Outro problema resolvido foi com relação ao uso de alguns programas como Corel Draw, que antes não podiam ser usados nos computadores da biblioteca por serem pesados demais para a configuração de hardware daqueles computadores. Porém, alguns programas não permitem serem executados em sistemas de processamento centralizado como é o caso do Auto CAD.

Apesar do Terminal Server da Microsoft se mostrar uma boa solução, alguns pontos negativos encontrados durante os testes devem ser considerados.

Um destes pontos é o fato do Terminal Server ser desenvolvido especificamente para ser acessado remotamente através de um sistema operacional Windows. Isto dificulta seu uso com *thin clients* por parte de algumas instituições como as escolas públicas, que vêm na ideia de processamento centralizado uma boa solução para reaproveitarem máquinas antigas. Porém, para que isto aconteça, deve-se lançar mão de alternativas de terceiros, como no referido estudo de caso do Polivalente de Americana, onde foi utilizada a distribuição Linux Thinstation.

Outro ponto negativo encontrado durante os testes realizados foi com relação à execução de arquivos de vídeo. Tanto arquivos de vídeo localizados no próprio servidor quanto os localizados na Internet são, praticamente, impossíveis de serem vistos nos *thin clients* devido à lentidão na taxa de atualização das imagens, as quais são apresentadas quadro a quadro.

Mas, como já mencionado, o Terminal Server ainda se mostra uma boa solução, pois os benefícios alcançados com ele acabam sendo maiores. Ainda são esperados mais benefícios com o uso de *thin clients* na biblioteca da referida escola, mas isso só será conseguido com o tempo, pois o trabalho diário com essa tecnologia tem mostrado que para que tudo funcione corretamente é preciso que se faça uma série de ajustes no servidor de terminais, e isso somente se dá com as necessidades que aparecem no dia-a-dia.

7 REFERÊNCIAS BIBLIOGRÁFICAS

[1] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Citação:** NBR-10520/ago - 2002. Rio de Janeiro: ABNT, 2002.

[2] _____ . **Referências:** NBR-6023/ago. 2002. Rio de Janeiro: ABNT, 2002.

[3] Applications of VNC, Disponível em: <http://www.realvnc.com/vnc/how.html>, acesso em: 12 março 2011.

[4] **BARBOSA, A. P.**, Disponível em: <http://recantodasletras.uol.com.br/artigos/1346172>, acesso em 25 março 2011.

[5] **BASTOS, D.; ALVES, R.; MACIEL, W.** Sistema Operacional Saber-Linux, Disponível em: <http://www.slideshare.net/campuspartybrasil/terminal-server-6611499>, acesso em 23 abril 2011.

[6] **CHIES, A.** Complexidade da Tecnologia, Disponível em: http://achies.com.br/home/index.php?option=com_content&view=article&id=62:1-complexidade-da-tecnologia&catid=43:gerenciamento-de-ti&Itemid=69, acesso em 05 março 2011.

[7] **CRUZ, L. F.** da, Disponível em: http://www.msitecnologia.com/blog/terminal_service.html, acesso em: 05 março 2011.

[8] Função do Terminal Server: Configurando um Terminal Server, Disponível em: [http://technet.microsoft.com/pt-br/library/cc779334\(WS.10\).aspx](http://technet.microsoft.com/pt-br/library/cc779334(WS.10).aspx), acesso em: 16 abril 2011.

[9] http://www.microsoft.com/brasil/servidores/windowsserver2003/requisitos_sistema.mspx, acesso em 16 abril 2011.

- [10] **LIMA, A. Q.** Ambiente de trabalho remoto com LTSP – Desenvolvimento, Disponível em: <http://www.webartigos.com/articles/27236/2/Ambiente-de-Trabalho-Remoto-com-LTSP/pagina2.html>, acesso em 08 março 2011.
- [11] Linux Terminal Server Project Administrator's Reference A Guide to LTSP Networks, Disponível em: <http://ufpr.dl.sourceforge.net/project/ltsp/Docs-Admin-Guide/LTSPManual.pdf>, acesso em 19 março 2011.
- [12] **MINASI, M.** Dando Suporte a Clientes com o Windows Terminal Services. In: **Dominando o Windows Server 2003**. 4ª edição. São Paulo: Pearson, 2008. P. 986-1046.
- [13] Minicomputer, Disponível em: <http://www.alanclements.co.uk/History/Minicomputer.htm>, acesso em 07 março 2011.
- [14] **MORIMOTO, C. E.** Terminais Leves com o LTSP. In: **Redes e Servidores Linux**. 2ª edição. Porto Alegre: Sul Editores, 2006. P. 359-401.
- [15] Otimizando Citrix com Expand Networks, Disponível em: http://www.optiwan.com/downloads/expand/artigos/Citrix%20White%20Paper_portuguese_V2.pdf, acesso em: 12 março 2011.
- [16] **PEREIRA F. F.** Inicialização remota do Thinstation para ambiente Windows Terminal Service, Disponível em: <http://www.ppgia.pucpr.br/~jamhour/RSS/TCCRSS08A/Fabio%20Fernando%20Pereira%20-%20Artigo.pdf>, acesso em: 01 abril 2011.
- [17] **SILVA, L. A.** Downsizing, Disponível em: <http://www.guialog.com.br/ARTIGO481.htm>, acesso em 25 março 2011.
- [18] **TANENBAUM, A. S.; STEEN, M. V.** Processos. In: **Sistemas Distribuídos Princípios e Paradigmas**. 2ª edição. São Paulo, 2008. P. 49.

[19] *Thin client*, Disponível em: http://pt.wikipedia.org/wiki/Cliente_magro, acesso em: 25 fevereiro 2011.

[20] Visão geral sobre os Serviços de Terminal, Disponível em: [http://technet.microsoft.com/pt-br/library/cc775715\(WS.10\).aspx](http://technet.microsoft.com/pt-br/library/cc775715(WS.10).aspx), acesso em: 11 março 2011.