UNIVERSIDADE ESTADUAL DO PIAUÍ - UESPI
CAMPUS PROFº. ALEXANDRE ALVES DE OLIVEIRA
BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO

PAULO RODRIGUES DOS SANTOS FILHO

PROPOSTA DE SERVIDOR LTSP NA BIBLIOTECA DA UESPI CAMPUS PARNAÍBA

PARNAÍBA
2011

PAULO RODRIGUES DOS SANTOS FILHO

PROPOSTA DE SERVIDOR LTSP NA BIBLIOTECA DA UESPI

Monografia apresentada ao Curso de Bacharelado em Ciências da Computação da Universidade Estadual do Piauí – UESPI, Campus Prof. Alexandre de Oliveira, como parte das exigências da disciplina de Estágio Supervisionado, requisito parcial para obtenção do título de Bacharel em Computação.

Orientador: Mayllon Veras da Silva

PARNAÍBA
2011

Ficha Catalográfica elaborada pela Bibliotecária
Christiane Maria Montenegro Sá Lins CRB/3 - 952

S237p

SANTOS FILHO, Paulo Rodrigues dos

Proposta de servidor LTSP na biblioteca da UESPI Campus Parnaíba/ Paulo Rodrigues dos Santos Filho. – Parnaíba: UESPI / Universidade Estadual do Piauí, 2011.

62 f.

Orientador: Esp. Mayllon Veras da Silva

Trabalho de Conclusão de Curso (TCC) – Universidade Estadual do Piauí, UESPI, Curso de Bacharelado em Ciências da Computação, 2011.

1. Redes de computadores. I. Silva, Mayllon Veras da. II. Universidade Estadual do Piauí. III. Título

CDD 004.6

GOVERNO DO ESTADO DO PIAUÍ
UNIVERSIDADE ESTADUAL DO PIAUÍ – UESPI
CAMPUS PROF. ALEXANDRE ÁLVES DE OLIVEIRA
COORDENAÇÃO DO CURSO DE CIÊNCIA DA COMPUTAÇÃO

Ata de Apresentação de Trabalho de Conclusão de Curso

Aos vinte dias do mês de agosto de dois mil e onze, às 9h30, no Laboratório de Informática do Campus Prof. Alexandre Alves Oliveira (Parnaíba) – UESPI, na presença da banca examinadora, presidida pelo professor Mayllon Veras da Silva e composta pelos seguintes membros: Alessandro Saraiva Mendonça e Turiano José Ribeiro dos Santos Neto, o aluno **Paulo Rodrigues dos Santos Filho** apresentou o Trabalho de Conclusão de Curso de Graduação em Ciência da Computação como elemento curricular indispensável à colação de grau, tendo como título: **Proposta de Servidor LTSP na Bihlioteca da UESPI.** A banca examinadora reunida em sessão reservada deliberou e decidiu pelo resultado de **aprovado** ora formalmente divulgado ao aluno e aos demais participantes. Nada mais havendo a tratar, eu professor Mayllon Veras da Silva na qualidade de presidente da banca lavrei a presente ata que será assinada por mim, pelos demais membros e pelo aluno apresentador do trabalho. Parnaíba (PI), 20 de agosto de 2011.

OBS.:	

Banca Examinadora

Prof. Esp. Mayllon Veras da Silva
Orientador, UESPI

Prof. Esp. Alessandro Saraiva Mendonça
Avaliador, PVP

Prof. Esp. Turiano José Ribeiro dos Santos Neto
Avaliador, SENAI

Aluno

Paulo Rodrigues dos Santos Filho

DEDICATÓRIA

Dedico primeiramente a Deus, que tem me capacitado em todos os momentos de minha vida e, com sua mão forte, ajudou-me na conclusão de mais esta etapa; em segundo lugar à minha família: mulher e filha, pais e irmãs, bem como a todos meus os amigos que ao longo desta caminhada compartilharam comigo momentos especiais.

AGRADECIMENTOS

Agradeço ao meu Senhor Jesus, pois, através de sua morte, hoje tenho vida e vida com abundância, e todo aquele que nele crê não perecerá, mas terá vida eterna. Agradeço à minha querida mãe Adelaide Maria e ao meu pai Paulo Rodrigues que sempre acreditaram em mim e me deram toda educação para que hoje eu possa me regozijar e dizer: "eu venci mais uma". Agradeço minha esposa Andréia e à minha filha Andreza, que são as minhas duas jóias preciosas, e também as minhas duas irmãs. Agradeço ao meu orientador Mayllon Veras, pois juntos tivemos êxito na conclusão deste trabalho.

"Elevo meus olhos para os montes: de onde me virá o socorro? O meu socorro vem do SENHOR, que fez o céu e a terra." Salmos 121 : 1-2.

RESUMO

Este trabalho visa ao desenvolvimento de um protótipo de servidor LTSP (*Linux Terminal Server Project* ou Projeto Linux de Servidor de Terminais) - uma ferramenta de código aberto muito utilizada em redes de computadores, de baixo desempenho, também denominados *thin client*, bem como à demonstração da viabilidade de inserção dessa tecnologia na biblioteca da Universidade Estadual do Piauí – UESPI, campus Parnaíba, onde se observou a existência de alguns computadores de mesa (Desktops) obsoletos e de outras máquinas novas. Assim, teve-se como foco de estudo alguma ferramenta que proporcionasse aos usuários da biblioteca a utilização de todos os computadores de forma equivalente quanto à velocidade de processamento visualizada pelos mesmos. No decorrer deste trabalho será demonstrado a configuração de um servidor, assim como dois clientes terminais executando o LTSP 5.0. Através desta tecnologia, pode-se agilizar a execução dos processos nos computadores clientes com baixo poder de processamento, por meio de um servidor de alto desempenho.

Palavras-chave: LTSP. Thin Client. UESPI. Linux. Terminais. Servidor. Tecnologia.

ABSTRACT

This work aims to development of a prototype of LTSP (Linux Terminal Server Project or Project Linux Terminal Server) - an instrument of open source, widely used in networks of computers underperforming, also called of thin client, as well as feasibility's demonstration of insertion of this technology in the library of the State University of Piauí – UESPI, Parnaíba campus, where it was observed that there are some desktop computers (desktops) obsolete and other new machines, so, it has to focus on the study, an instrument has given for library's user, the use of all computers in an equivalent way as the processing speed displayed by them. Throughout this work will be demonstrate the configuration of a terminal server and two clients running LTSP 5.0. Through this technology we can streamline the processes from running on client computers with low processing power through a high performance server.

Keywords: LTSP. Thin Client. UESPI. Linux. Terminals. Server. Technology.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

BIOS- *Basic Input/Output System*
CD- *Compact Disc*
CPD- Centro de processamento de dados
DHCP- *Dinamic Host Configuration Protocol.*
FIFO- *First-in, First-out*
FTP- *File Transfer Protocol*
GB- *Gigabyte*
GDM- *GNOME Display Manager*
HD- *Hard Disc*
ID- Identificador
IDE- *Integrated Development Environment*
IETF- *Internet Engineering Task Force*
IP- *Internet Protocol*
KB- *Kilobyte*
KDM- *KDE Display Manager*
LDM- *LTSP Display Manager*
LTSP- *Linux Terminal Server Project*
MAC- *Media Access Control*
NFS- *Network File System*
PC- *Personal Computer*
PXE- *Preboot Execution Environment*
RAM- *Random Access Memory*
RARP- *Reverse Address Resolution Protocol*
RDP- *Remote Desktop Protocol*
RFC- *Request For Comments*
ROM- *Ready Only Memory*
RPC- *Remote Procedure Call*
SO- Sistema Operacional
SSH- *Secure Shell*
TI- Tecnologia da Informação
TFTP- *Trivial File Transfer Protocol*
UESPI- Universidade Estadual do Piauí
XDMCP- *X Display Manager Control Protocol*
XDR- *External Data Representation*

SUMÁRIO

1 INTRODUÇÃO

A Tecnologia da Informação (T.I.) é uma área que está em constante evolução. Novas soluções em algoritmos são implementadas, como as técnicas de otimização, reduzindo inúmeras linhas de códigos a poucos comandos complexos, bem como novos dispositivos físicos que compõem o hardware, a exemplo os novos processadores com dois ou mais núcleos de processamento, que estão sendo fabricados e aprimorados a cada ano. Não diferente das demais áreas, as Redes de Computadores vêm ganhando destaque principalmente nas grandes empresas, ambientes domésticos e laboratórios de informática. Em conformidade com o pensamento de Tanembaum (2003), o velho modelo de um único computador atendendo a todas as necessidades computacionais da organização foi substituído pelas chamadas redes de computadores, nas quais os trabalhos são realizados por um grande número de computadores separados, mas interconectados.

Em busca de maior velocidade e estabilidade na troca de pacotes entre computadores em rede, também conhecidos como estações de trabalho, encontramos usuários cada vez mais exigentes e tendo em vista as estações que não possuem tanto poder de processamento e consequentemente não têm grande velocidade no tratamento das informações; surge, portanto, a necessidade de potencializar o processamento dessas máquinas obsoletas. Assim, observamos várias possibilidades que se conciliam com os anseios dos usuários, dentre elas: a atualização do *hardware* através de *upgrades* (troca de componentes, para aumentar o desempenho dos computadores), ou até mesmo a substituição das máquinas antigas por outras mais modernas. Essas possibilidades, entretanto vêm a angariar custos, que, em tese, são bastante consideráveis. Frente a isso, dispõe-se de outra solução, a qual se denomina de projeto de rede com terminais leves. Mais a frente serão observados alguns benefícios adquiridos pela adesão a esta tecnologia.

Baseando-se na velocidade do compartilhamento, tanto de informações, como de periféricos, por meio de enlaces de comunicação em uma rede local, viu-se que a Biblioteca da Universidade Estadual do Piauí (UESPI) dispunha de máquinas de desempenho inferior quanto à velocidade de processamento e de compartilhamento de informações, precisando, assim, de alguma alternativa para que os computadores atendam às exigências dos usuários, e, é nessa idéia, que este trabalho vem defender a elaboração de um protótipo que comprove a eficácia de inserção de uma rede com terminais leves baseadas no Linux Terminal Server

Project ou LTSP, viabilizando a futura instalação na Biblioteca da UESPI. Todavia, antes de explanar sobre os demais assuntos, deve-se primeiramente evidenciar a conjuntura na qual este se desenvolve.

1.1 Estrutura do Trabalho

Esta monografia é um trabalho prático que foi executado como proposta para futuramente implantar-se na biblioteca da UESPI um servidor de terminais. Através de uma visita técnica, percebeu-se a existência de alguns computadores de baixo desempenho e outros que possuem uma configuração de alto desempenho. Dessa forma pode-se utilizar o poder de processamento de um computador central e potencializar os inferiores, proporcionando o aumento da utilização de todos os computadores da biblioteca, conforme será demonstrado no protótipo desenvolvido no presente trabalho.

Prosseguindo no raciocínio anteriormente descrito, falaremos, nos capítulos a seguir, sobre alguns termos fundamentais para compreensão do leitor, como *Thin Client*, servidor de terminais, tecnologia LTSP. Veremos ainda os principais conceitos em que se fundamenta este trabalho, em seguida detalhes da configuração do protótipo e sua análise.

O capítulo 2 aborda as características do processamento e suas duas definições, quanto às arquiteturas dos computadores. Nesse capítulo são demonstrados os dois tipos de computação, a centralizada e a distribuída, para definirem-se os conceitos sobre o tratamento da informação a nível lógico. Finalizando este capítulo são citados todos os protocolos que envolvem o funcionamento do LTSP e suas funções exercidas no mesmo.

O capítulo 3 trata da implementação do protótipo, desde a escolha do Sistema Operacional até os equipamentos que o compõem. As instalações dos serviços que o servidor necessitará para disponibilizar aos clientes suas requisições e a configuração dos clientes para possibilitar a inicialização destes.

O capítulo 4 detalha sobre a análise realizada no protótipo, que foi dividida em três fases onde atuam diferentes protocolos, resumindo os níveis de comportamento da rede. A suportabilidade da rede frente às requisições dos clientes na execução de aplicativos muito utilizados em ambientes de estudo, como numa biblioteca. Após cada teste é feita uma análise de desempenho da rede quanto à utilização do processador, memória e tráfego de rede, aferindo na viabilidade de utilização do LTSP em cada um deles.

O capítulo 5 explana as conclusões obtidas no decorrer deste trabalho e trata de trabalhos futuros.

1.2 Thin Clients (Clientes Magros).

Os terminais leves são assim denominados por serem desprovidos de hardwares de alto desempenho e pouca ou nenhuma mídia de armazenamento em massa. "Os *Thin Clients,* também denominados terminais leves, surgiram aproximadamente na década de 40 com os *mainframes* ou computadores de grande porte, que suportavam inúmeros terminais" (RUSCHEL, 2009 pg. 09). Segundo Richards (2007), quando os *Thin Clients* são utilizados, todos os processos de software são executados a partir dos servidores, se mostrando extremamente confiáveis para manipulação, e os vírus são praticamente inexistentes. É distribuído sob a licença GNU *General Public License*, o que significa que é gratuito e sempre será (LTSP, 2011). Já os terminais leves que não possuem mídias como os *HD's* são chamados de Terminais *Diskless* - uma ramificação dos terminais leves. Podemos chamar de *Thin Client* qualquer equipamento que se conecta a um servidor de terminal, seja ele um hardware específico ou um simples PC executando um *boot* remoto, que é o processo de inicialização de um computador através de outro sistema (RUSCHEL, 2009).

Para que possam carregar seu ambiente gráfico, eles dão *boot* através de um ***Etherboot***, um *software* bem pequeno de apenas 35 ou 40 KB (dependendo da placa de rede), podendo ser gravado em um disquete ou em um *CD* de *boot* ou mesmo em um *chip* de *boot* conectado na placa de rede, por isso os terminais leves não precisam de grandes níveis de configuração, devido sua utilização na rede ser limitada, apenas para exibição da interface gráfica requisitada do servidor de aplicações. Conforme Morimoto (2006, p. 12), "É possível transformar micros a partir de antigos processadores 486 da Intel, em terminais leves, onde um servidor mais rápido executa os aplicativos e envia apenas a imagem para os terminais". Na figura 1 o exemplo de um computador *thin client* fabricado pela HP.

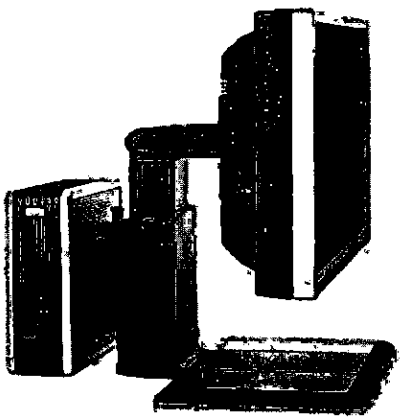

Figura 1 - Exemplo de um Terminal Thin client HP Compaq t5725.
Fonte: FARIAS (2009).

1.3 Servidor de Terminais.

Os Servidores de Terminais são também conhecidos por Servidores de Aplicações - uma ferramenta poderosa para instalar, distribuir e suportar aplicativos a partir de uma localização centralizada Scrimger et. al. (2002). Os servidores de terminais, em tese, oferecem através de um ambiente multiusuário do tipo UNIX (Sistema Operacional multitarefa que deu origem ao Linux). Todo processamento de aplicativos e de serviços acontecem centralmente nesses servidores. Quando um computador é referido como um servidor de terminais, na realidade significa que existe uma configuração lógica para que as aplicações executadas no servidor e distribuídas pelos terminais aconteçam. De acordo com Scrimger et. al. (2002), um servidor de terminais consiste em três componentes: o núcleo de servidor multiusuário, o software de cliente de servidor de terminais e o protocolo utilizado para comunicação cliente-servidor.

Núcleo de servidor multiusuário: Fornece a capacidade básica para hospedar diversas sessões simultâneas de cliente. Além disso, inclui ferramentas de administração para gerenciar ambos, o servidor e as várias sessões de cliente.

Software de cliente de servidor de terminais: Precisa ser instalado em todos os nós que acessam o servidor de terminais para vários serviços e aplicativos. Utilizá-lo pode ser tão simples como trabalhar com *telnet,* um protocolo que permite a comunicação entre computadores de uma rede.

Protocolo: Utilizado para facilitar a comunicação entre o servidor de terminais e vários clientes. Pode-se citar como exemplo de servidor de terminais o *Remote Desktop Protocol* (RDP), utilizado pela *Microsoft Terminal Server.*

1.4 Tecnologia LTSP

O *Linux Terminal Server Project* (LTSP) ou Projeto de Servidor de Terminais Linux foi criado por James McQuillan e Ron Colcernian nos Estados Unidos, em 1999. É um projeto *Open Source,* ou seja, projeto de código aberto, não-proprietário, que disponibiliza de ferramentas administrativas para criação de terminais leves com Gnu/Linux, utilizando a combinação dos seguintes protocolos numa rede de computadores:

- DHCP – Protocolo que fornece aos clientes conectados ao servidor endereços automaticamente.
- TFTP – Protocolo que possibilita a transferência de arquivos.
- NFS- Protocolo que realiza montagem de diretório remotamente.
- XDMCP- Protocolo de gerenciamento de interfaces gráficas.
- SSH- Protocolo de transferência de dados criptografados.

O LTSP que faz com que máquinas obsoletas tornem-se estações de alto desempenho, é uma tecnologia que aproveita a capacidade de processamento de um computador poderoso de maneira distribuída entre terminais leves (MENDES, 2006). O *Linux Terminal Server Project* adiciona suporte de thin client para servidores Linux, o mesmo é uma solução flexível e de custo eficaz que está capacitando as escolas, empresas e organizações em todo o mundo para instalar facilmente e implantar estações de trabalho (LTSP, 2011).

O LTSP visa sanar problemas, como: a obsolência de algumas máquinas que apesar do pouco tempo de utilização tornam-se rapidamente defasadas, os altos custos de atualizações de hardware em grandes empresas, universidades, órgãos públicos ou privados, a proteção ao meio ambiente que no descarte de máquinas antigas é completamente agredido, devido à existência de materiais como plástico e metais pesados em sua composição que levam anos para se degradarem, além de serem materiais tóxicos. Por último citamos os benefícios que o LTSP pode levar a comunidades carentes na inclusão digital. Nas palavras de (CAMPOS, 2009 p. 01):

> Esta solução, baseada em software livre foi implementada na sala de uma escola indígena na Aldeia Córrego do Meio, Sidrolândia – MS, onde disponibilizou o acesso à informática básica a professores, alunos e comunidade, tornando possível a conclusão digital a partir do uso da informática.

Tendo em vista as soluções através da inserção do LTSP nas questões anteriores, pode-se vislumbrar a eficácia da implantação da tecnologia LTSP, possibilitando grande avanço ao meio no qual é inserida. Segundo Morimoto (2006, p. 376), "[...] o LTSP é uma solução mais usada para a criação de terminais leves [...]". Na figura 2 é demonstrado um esquema básico de rede LTSP, onde se têm três clientes conectados a um servidor através de um *switch*.

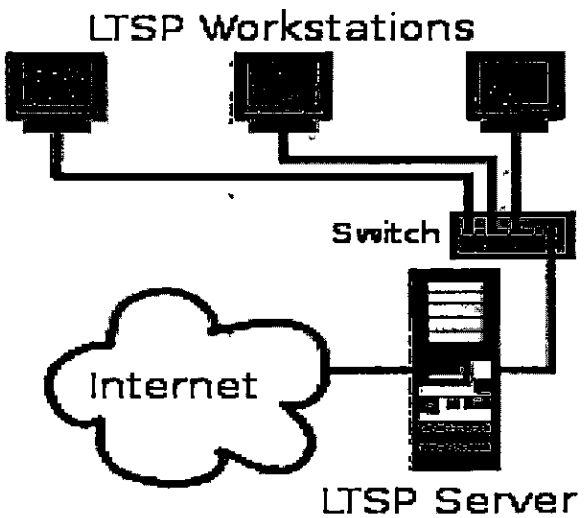

Figura 2 - Diagrama de uma Rede com LTSP.
Fonte: LTSP (2011).

1.5 Objetivos do Trabalho

Para a aquisição de conhecimento sobre o problema, orientamo-nos a partir de objetivos esclarecedores, priorizando uma pesquisa determinada.

1.5.1 Objetivo Geral

Demonstrar a viabilidade da inserção de uma rede de terminais Leves na biblioteca da Universidade Estadual do Piauí baseada no LTSP.

1.5.2 Objetivos Específicos

a) Elaborar um protótipo de rede que demonstre a utilização da tecnologia LTSP.
b) Inferir sobre os benefícios de inserção da tecnologia LTSP.
c) Demonstrar o ganho de desempenho nos computadores obsoletos do protótipo.
d) Criar uma solução que torne a velocidade de processamento equivalente nos

computadores na biblioteca da UESPI.

e) Mostrar a análise de desempenho do ambiente *thin client*, levando em consideração o consumo de memória RAM o desempenho do processador e o tráfego da rede.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Tipos de Processamento

A partir da evolução tecnológica, especificamente na década de 60, os computadores da terceira geração, que, em sua arquitetura, utilizavam microcircuitos integrados, em substituição às válvulas, proporcionaram um grande avanço na área de processamento de dados, passaram a ser construídos para atender tanto ao processamento comercial, quanto ao processamento científico. Conforme o pensamento de Rodriguez et. al. (2000, p.48), "O IBM/360, por exemplo, foi assim denominado por ter sido projetado para cobrir "180 graus" de aplicações comerciais e 180 de aplicações científicas, totalizando 360 graus".

A partir dessa evolução tecnológica, a ênfase quanto à classificação dos computadores foi dada à arquitetura, deixando de lado o parâmetro de classificação de computadores através de gerações. Ainda na década de 60, a utilização dos computadores resumia-se na centralização, ficando suas operações restritas fisicamente em uma grande sala, denominada CPD ou Centro de Processamento de Dados, onde usuários faziam suas aplicações via *batch* ou aplicações em lote, que consistiam em execuções de diversos trabalhos coletados em lote sem qualquer interação com o usuário (BROOKSHEAR, 2003). Com o surgimento dos microcomputadores, a ideia de nenhuma interação com o usuário do processamento *batch* foi extinta; a computação, aproximada dos usuários na década de 80 com os *PC's* (*Personal Computer*) ou computadores pessoais. De forma reduzida, pode-se afirmar que os avanços tecnológicos na área de processamento de dados, nos últimos 30 anos, foram exorbitantes, conforme mostra a figura 3.

Ano	Custo (US$)	Tempo de processamento (minutos)	Equipamento
1970	500.000	1020	IBM/360, 128KB RAM, 28MB HD
1980	80.000	120	PDP11/44, 512KB RAM, 216 MB HD
1995	3.000	2	PC486-66MHZ, 8MB RAM, 480 MB HD

Figura 3 - Crescimento do Poder Computacional
Fonte: (Bookshear, 2003)

A seguir serão descritos os dois principais modelos de processamento criados a partir da evolução da Tecnologia da Informação: os processamentos centralizado e distribuído.

2.1.1 Processamento Centralizado

Desde o início do processamento eletrônico até final da década de 70, a alternativa disponível na manipulação de dados era o uso de centros de processamento de dados centralizados, equipados com grandes computadores e basicamente para aplicações comerciai. Segundo Brookshear (2003, p. 48):

> A utilização dos centros de processamento de dados dependia de uma equipe altamente especializada que trabalhava na operação dos computadores e no desenvolvimento de aplicativos, particularmente os de caráter corporativo. Esta equipe era percebida como uma elite privilegiada, que estava mais próxima dos computadores, sempre disponíveis para eles. Os usuários, por sua vez, normalmente estavam muito longe do centro de computação, e na maioria dos casos o processamento era realizado no modo batch.

Os computadores centrais realizavam todo o processamento das informações, porém a falta de interação dos usuários com a máquina propiciou que os computadores centrais fossem equipados por terminais nos quais poderiam ser realizadas as tarefas que anteriormente só poderiam ser executadas de forma única, em sistemas de organização, os quais eram ordenados de forma que o primeiro a entrar seria o primeiro a ser executado. Esse método de organização é denominado **FIFO** (***first in, first out***). Dessa maneira, disseminava-se o processamento centralizado, contudo, devido à falta das facilidades adequadas ao processamento interativo com a máquina, uma nova arquitetura de processamento seria implementada, o processamento distribuído.

2.1.2 Processamento Distribuído

A partir da década de 80, os microcomputadores pessoais foram inseridos no mercado, determinando certa independência da então denominada elite do centro de computação. Os usuários agora poderiam implementar suas próprias soluções independentemente dos CPD's. Organizações indagaram a respeito de qual arquitetura adotar, algumas investiram na aquisição e utilização dos computadores pessoais; outras ficaram satisfeitas por terem alguns usuários longe dos centros de processamento, especificamente aqueles que possuíam grandes quantidades de pendências a serem satisfeitas. Somando essas indagações com a queda do custo dos centros de processamento de dados, o ambiente computacional começou a descentralizar-se.

Os computadores pessoais eram considerados adequados à edição de textos e à transmissão de dados para o computador central, proporcionando muitas requisições dessas tecnologias. Logo o crescimento da computação, além das grandes CPD's, as quais eram equipadas por inúmeros terminais, tornou-se um fato de grande ênfase nesse período devido à melhoria de serviço dos usuários. Com relação a isso, Rodriguez et. al. (2000 p. 50) afirma que:

> À medida que mais usuários tinham acesso aos ambientes computacionais, maiores recursos foram sendo necessários a cada centro de computação. O crescimento das facilidades computacionais solicitadas fazia com que, entre o planejamento, aprovação, aquisição, instalação e liberação para o S.O, um tempinho de 6 meses fosse normalmente requerido, chegando em alguns casos a mais de 1 ano. Neste processo, a demanda de usuários para capacidade de processamento disponível tinha um crescimento contínuo, enquanto as facilidades computacionais eram sempre insuficientes.

À medida que o final da década de 80 chegava, os especialistas em Tecnologia da Informação concentravam suas atenções nos computadores de grande porte, enquanto os usuários implementavam suas soluções nos PC's, visando sanarem suas necessidades pessoais. Todas essas investidas tecnológicas ocasionaram uma descentralização do ambiente computacional Rodriguez et. al. (2000).

Apesar dos avanços, muitas deficiências foram encontradas nesses processos de desenvolvimento de soluções independentes para os microcomputadores pessoais e os mainframes, criando algumas "ilhas" de informação, nos parâmetros da troca de informações,

exceto entre os recursos disponíveis nos computadores de grande porte, cita-se como exemplo a desintegração dos sistemas de pequeno porte implementados para os micros como os editores de textos que não possuíam portabilidade entre si, ou seja, o que era digitado em um aplicativo numa máquina não poderia ser lido por outro. Sendo assim o usuário deveria digitar todo o conteúdo novamente, caso não dispusesse do mesmo editor de texto.

Frente a essas problemáticas, criaram-se padrões de software, objetivando a integração entre os computadores de grande porte com os microcomputadores pessoais, como a transferência de arquivos entre eles, emulação de terminais em microcomputadores, o processamento *task-to-task* no qual permitia a sincronização de aplicações de diferentes fornecedores. A necessidade da troca de dados, motivou a interconexão de diversos micro computadores entre si e, por conseguinte, com os mainframes iniciando a filosofia de processamento distribuído.

2.2 Computação Distribuída

O modelo de computação distribuída é baseado na utilização de diversos processamentos ou sistemas de multiprocessamento, visando à execução de um processo em comum, de forma transparente e coerente, possibilitando aos usuários uma visão de sistema homogêneo, denotando-se um sistema centralizado. A heterogeneidade presente nos dispositivos que executam os sistemas distribuídos implementa a ideia de diversas interfaces que são interconectadas a diferentes placas mãe, que processam suas informações em diferentes Sistemas Operacionais. Os sistemas distribuídos, em consonância com Coulouris et. al. (2007 p. 15), "[...] são aqueles sistemas nos quais os componentes localizados em computadores interligados em rede, se comunicam e coordenam suas ações.".

Um sistema de computação distribuído possui no mínimo um dos seguintes requisitos (GHOSH, 2007):

- **Múltiplos processos** – mais de um processo sequencial, seja ele de sistema ou de usuário; podem ser executados possuindo um caminho e um controle independentes.

- **Comunicação inter processos** – um processo pode se comunicar com outro por mensagens através de um canal em um período finito de tempo.
- **Separação dos espaços de endereçamento** – Os processos não devem possuir espaços de memória compartilhados.
- **Metas coletivas** – os processos devem interagir uns com os outros a fim de se obter um resultado comum. Por exemplo, se um processo calcula a largura de um quadrado e outro a altura, os dois processos podem interagir para se obter a área do quadrado.

Alguns Sistemas Distribuídos aplicados em redes são bastante familiares e difundidos, como a internet, que consiste em um conjunto de dispositivos interligados em uma rede de computadores. Essa rede permite que interajam entre si por diversos protocolos que são norteados pelo *internet protocol* ou protocolo de internet (IP) e pelo envio de mensagens através de um meio de comunicação comum. Outro exemplo de sistema distribuído são as *intranets* – redes que possuem as mesmas funcionalidades e conceitos da internet, diferenciando-se por possuir níveis de acesso privado, restrito às necessidades dos administradores. Os usuários de uma intranet podem se conectar a internet acessando serviços armazenados em outras mídias remotamente.

Os Sistemas Distribuídos possuem algumas metas fundamentais a serem atendidas, como a conectividade, através da qual o sistema deve conectar de forma ágil os usuários aos seus recursos. Essa conectividade reforça a ideia de distribuição de componentes de uma rede, visando ao compartilhamento de recursos. Outra meta é a Transparência, que visa ocultar dos usuários a distribuição dos recursos pela rede de hardwares e softwares heterogêneos, contextualizando uma visão unicista de sistema. Segundo os pensamentos de Coulouris et. al. (2007), têm ainda a heterogeneidade que se aplica aos seguintes aspectos: redes, hardware, sistemas operacionais, linguagens de programação, implementações de diferentes desenvolvedores. Um exemplo prático da heterogeneidade é o fato de computadores interligados no mundo inteiro de diferentes fabricantes comunicam-se através de protocolos internet sem maiores complicações.

2.3 Compùtação Centralizada

Conforme Turban et. al. (2002 p. 299), "A computação Centralizada surgiu na década de 70 e tem sido a base da computação empresarial há mais de 30 anos [...]". A base de funcionamento desse tipo de computação está na presença de um computador central, também conhecido por **Mainframe**, com um alto poder de processamento e capacidade de armazenamento, por meio do qual os clientes a ele conectados, fazem requisições para alocarem os recursos solicitados pelos mesmos. Tanembaum (2007), coloca que os sistemas centralizados eram denominados de sistemas de tempo compartilhado, neles cada terminal possuía um *time slice* ou fatia do tempo do processador para que seus processos fossem executados, possibilitando, assim, um melhor aproveitamento da capacidade de processamento dos grandes *mainframes* existentes.

Apesar da defasagem desse tipo de arquitetura, segundo Turban et. al. (2002 p. 299), "[...] muitos especialistas concordam que ela continuará existindo por muitos anos, pois a computação centralizada oferece a possibilidade de multiprocessamento paralelo, possibilitando que usuários usem a capacidade de processamento do servidor central.".

Abaixo estão algumas vantagens da Computação centralizada:

- **Custo-Benefício:** Ao invés de se investir em computadores de alto poder de processamento em quantidades equivalentes ao número de funcionários, far-se-á isso apenas no servidor central com alto poder de processamento, o qual executará as requisições dos clientes.
- **Baixo Consumo de energia:** essa característica é definida pela baixa potência dos terminais.
- **Backups centralizados:** Como todos os arquivos estarão armazenados no computador central será necessário apenas a realização de um backup central dos dados.

2.4 Terminais Leves com LTSP

Como foi visto, os *thin clients* são computadores de baixo desempenho, e estão cada vez mais entrando em desuso, devido à rápida evolução da informática (MICHELON,

2009). O hardware é rapidamente depreciado, pois os *softwares* exigem níveis de processamento e de memória RAM cada vez mais elevados. Citam-se como exemplo os requisitos mínimos e máximos de uma máquina para rodar o Windows 7 (MICROSOFT, 2011):

- Processador de 1 gigahertz (GHz) ou superior de 32 bits (x86) ou 64 bits (x64);
- 1 gigabyte (GB) de RAM (32 bits) ou 2 GB de RAM (64 bits);
- 16 GB de espaço em disco disponível (32 bits) ou 20 GB (64 bits);
- Dispositivo gráfico DirectX 9 com driver WDDM 1.0 ou superior.

No exemplo do Windows 7, notamos que a disponibilidade de máquinas antigas que suportem o novo sistema operacional da Microsoft é praticamente nula, contudo deve-se observar a disponibilidade de ferramentas que possibilitem o reaproveitamento de máquinas obsoletas, como o LTSP. Essa ferramenta é um projeto para criação de um servidor de terminais, utilizando um sistema operacional GNU/Linux. Segundo Rezende (2008), podemos ter as seguintes opções de terminais gráficos para um sistema baseado em LTSP:

- utilização de *Thin Client* - máquina projetada especificamente para funcionar como terminais gráficos. Possui como características baixo poder de processamento, baixo consumo de energia, tamanho e custo de manutenção reduzidos;
- estações de trabalho montadas com o propósito de serem servidores gráficos. Por esse motivo, geralmente são máquinas desprovidas de dispositivos de armazenamento e com poder de processamento reduzido e ainda capazes de realizarem o boot via rede;
- estações de trabalho obsoletas, aquelas que possuem um poder de processamento insuficiente para executar os *softwares* atuais no modelo tradicional devido ao aumento das necessidades de processamento que foi ocasionado pela evolução desses sistemas.

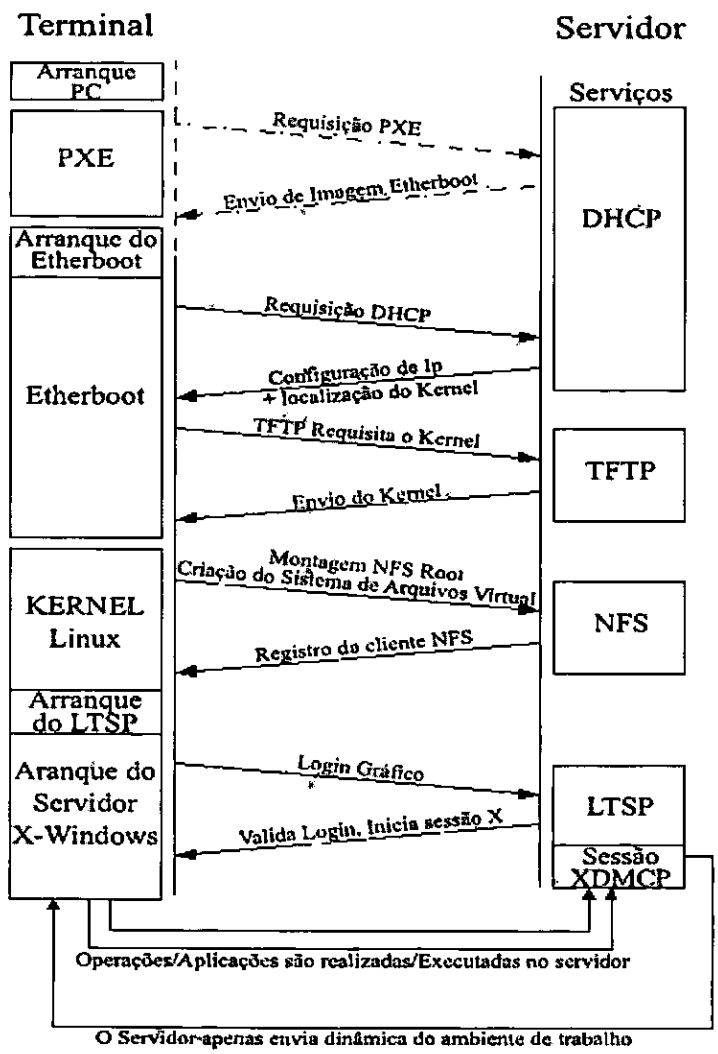

Figura 4 - Seqüência de Processos na Inicialização de um Terminal. Fonte: REZENDE (2008).

2.5 Bootstrap Protocol (BOOTP)

Os dispositivos computacionais que são gerenciados por um Sistema Operacional necessitam das informações sobre o sistema na inicialização. Nos computadores, essas informações ficam basicamente localizadas nos disco rígido ou *HD*, entretanto, nos computadores que não possuem disco rígido, como no caso dos terminais *diskless*, os dados

não estão disponíveis em um ambiente de rede. Esses terminais necessitam ainda de um endereço IP único. Como resultado dos desfalques na arquitetura, os *thin client's* precisam carregar suas informações em fontes externas. Assim, com a utilização *Reverse Address Resolution Protocol* (RARP), recebem um IP válido via DHCP e as suas informações iniciais a partir de um servidor de inicialização, contudo o RARP possui algumas deficiências que o tornam inadequado para disponibilizar as informações de inicialização, Scrimger et. al. (2002, p. 193) destaca duas delas:

> O pacote que é trocado entre o servidor e o cliente contém apenas o endereço IP de quatro bytes do cliente. O cliente também precisa das informações iniciais para inicializar, as quais não são fornecidas pelo pacote RARP; o RARP utiliza um endereço de MAC do host para sua identificação. Como resultado, não pode ser utilizado em redes onde os endereços de hardware são atribuídos dinamicamente.

O BOOTP foi desenvolvido para sanar as desvantagens encontradas no RARP, pois, além de conter o endereço IP, as mensagens enviadas pelo protocolo contêm as informações requisitadas por um terminal *diskless* para sua inicialização com sucesso. Outro conteúdo importante da mensagem é o endereço de servidor de BOOTP, o roteador ou gateway padrão da rede.

Para Scrimger et. al. (2002), a inicialização do BOOTP consiste em duas fases. A primeira é chamada de fase de determinação de endereço e seleção de arquivo de inicialização, onde o cliente obtém o endereço IP e os arquivos exigidos na inicialização são selecionados. Essa primeira fase começa quando o terminal *diskless* em seu processo de inicialização envia uma solicitação de endereço IP para o servidor DHCP e de um arquivo da imagem de inicialização ao servidor de BOOTP, através da porta 67. Após a identificação do endereço MAC do cliente pelo servidor, o qual é enviado juntamente com a solicitação, o servidor disponibiliza as informações solicitadas. A segunda é chamada de fase de transferência de arquivos de inicialização. Depois da fase inicial, o cliente utiliza um protocolo de transferência de arquivo para copiar os arquivos iniciais do servidor de inicialização, começando a 2ª fase depois que o servidor de BOOTP identifica o cliente e seleciona seu arquivo de imagem específico. Então o cliente copia, através de um protocolo de transferência de arquivo, como o *Trivial Fiel Transfer Protocol* (TFTP) ou o *File Transfer Protocol* (FTP), o arquivo de inicialização para sua memória.

2.6 Dynamic Host Configuration Protocol (DHCP)

O DHCP ("*Dynamic Host Configuration Protocol*" ou "protocolo de configuração dinâmica de endereços de rede"), conforme Morimoto (2006), permite que todos os micros da rede recebam suas configurações de rede automaticamente a partir de um servidor central, sem que você precise ficar configurando os endereços manualmente em cada um. O protocolo DHCP (uma ampliação do protocolo BOOTP que entrou em desuso devido a exigência da configuração manual das tabelas que mapeiam os endereços IP para endereços *Ethernet* ou endereços para computadores de redes cabeadas). Em consonância com Tanembaum (2003), o DHCP permite a atribuição manual ou automática de endereços IP, ele é descrito nas RFCs 2131 e 2132. O DHCP usa um modelo cliente servidor, no qual o servidor gerencia de forma centralizada os endereços IP usados numa rede. Em seguida será demonstrada a descrição do funcionamento do DHCP:

1. Um cliente envia um pacote (DHCP DISCOVER) para todos os computadores da rede (*broadcast*), endereçado ao IP "255.255.255.255".
2. Quando o servidor DHCP captura o pacote, reenvia outro pacote (DHCP OFFER) em *broadcast* endereçado ao IP "0.0.0.0".
3. Só o cliente que fez a requisição ao servidor DHCP receberá temporariamente o pacote com o endereço IP, máscara, gateway e servidores DNS, pois o pacote enviado está associado ao endereço MAC do cliente solicitante.
4. Após um determinado tempo configurado no servidor (*lease time*), o cliente envia ao servidor DHCP uma mensagem de renovação de IP (DHCP REQUEST). Se não estiver disponível, o cliente perde conexão com a rede até que o servidor DHCP volte a responder, confirmando as informações ao cliente (DHCP ACK).

2.6.1 Utilização do Protocolo DHCP no LTSP

Um servidor DHCP instalado num servidor LTSP é configurado para atender a requisições dos clientes, enviando as configurações adequadas para o ingresso do host à rede.

Ele é o primeiro a ser acessado pelas estações, entregando as informações sobre o Kernel ou cliente PXE - *Preboot Execution Environment.* A estação deve estar apta a encontrar no compartilhamento de rede, especificamente no servidor, o sistema a ser carregado por ela. Ao ser ligada, a estação executa o *Etherboot* e envia uma mensagem a todos os terminais, chamada de *broadcast,* informando o endereço físico *Media Access Control* (MAC). O servidor DHCP recebe a mensagem e verifica se localmente o endereço físico está localizado no arquivo de configuração *dhcp.conf* ou se há outra resposta padrão a ser enviada. Caso alguma das alternativas seja atendida, o servidor retorna uma mensagem informando um endereço IP, uma máscara de rede, o caminho do núcleo do sistema chamado de Kernel, o caminho do sistema de arquivos raiz e outras configurações dispostas no arquivo *dhcp.conf.*

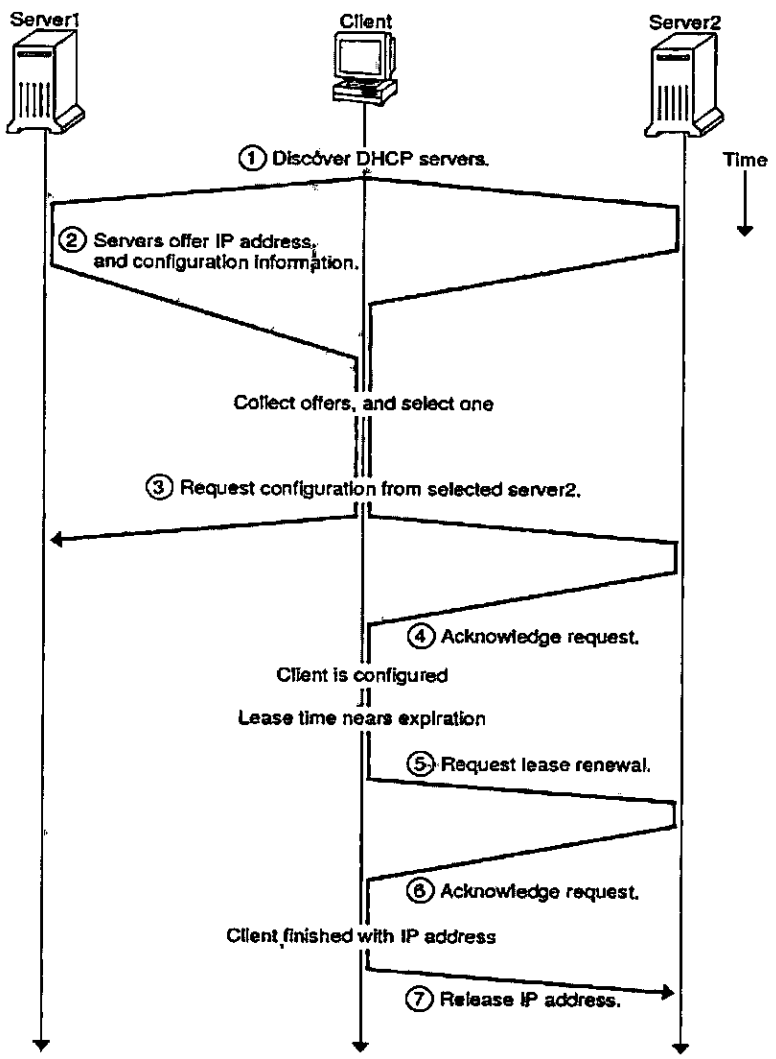

Figura 5 - Seqüência de Eventos do DHCP.
Fonte: http://www.shrubbery.net/solaris9ab/SUNWaadm/SYSADV3/p26.html

2.7 Trivial File Transfer Protocol (TFTP)

O *Trivial File Transfer Protocol* (TFTP) ou protocolo trivial de transferência de arquivos é uma versão simplificada do *File Transfer Protocol* (FTP), que é baseado no *Transmission Control Protocol* (TCP), portanto é orientado à conexão. Já o TFTP utiliza portas UDP (*User Datagram Protocol* ou Protocolo de Datagrama de Usuário) para a transferência dos dados, sendo assim não inclui suporte à correção de erros. Ele também não permite autenticação de usuário, os conteúdos dos diretórios remotos não podem ser mostrados na tela, e só a transferência de arquivo é suportada. Embora algumas dessas características façam o TFTP parecer um protocolo inferior, sua utilização é fundamental para que sistemas como o LTSP sejam implementados com sucesso, cita-se o exemplo da utilização dos terminais *Diskless*. Para Morimoto (2006), ele pode ser usado para transferência de arquivos em geral, porém é mais freqüentemente usado em sistemas de *boot* remoto. O TFTP é um protocolo simples e de fácil implementação, seu tamanho reduzido é muito importante para aplicações em *Thin clients,* os softwares baseados nesse protocolo podem ser gravados em dispositivos de armazenamento somente de leitura (ROM), para que sejam utilizados na obtenção de uma imagem na inicialização da máquina.

Uma vantagem segundo Comer (2006), em se utilizar o TFTP é que ele permite que o BOOTP utilize os mesmos protocolos básicos do conjunto TCP/IP que o sistema operacional irá utilizar quando for totalmente inicializado. A inicialização de comunicação com o servidor TFTP pelo cliente é estabelecida através do envio de um pacote de inicialização, que pode ser de leitura ou escrita. Caso a requisição seja aceita pelo servidor, estabelecerá uma comunicação com o cliente. O TFTP realiza suas operações através do envio de arquivos em blocos com tamanho fixo de 512 bytes, eles recebem uma numeração em ordem crescente inicializada pelo bloco de número um. O bloco $n+1$ só será transmitido após a chegada de confirmação do recebimento do bloco n, tanto o servidor como o cliente são implementados de forma que um contador de tempo limite controle a transmissão dos pacotes de dados e de confirmação.

2.7.1 Utilização do Protocolo TFTP no LTSP

Depois de receber a mensagem enviada pelo servidor DHCP, informando a configuração da placa de rede, o terminal executa o TFTP possibilitando carregar nos

registros da memória dos terminais leves o kernel do servidor, o qual passará a poder controlá-los. O kernel carregado pelo TFTP é responsável por fazer a detecção dos periféricos, carregar o módulo e as configurações finais da placa de rede (REZENDE, 2008).

Na figura 6 é demonstrado a seqüência de eventos entre a comunicação entre cliente e servidor.

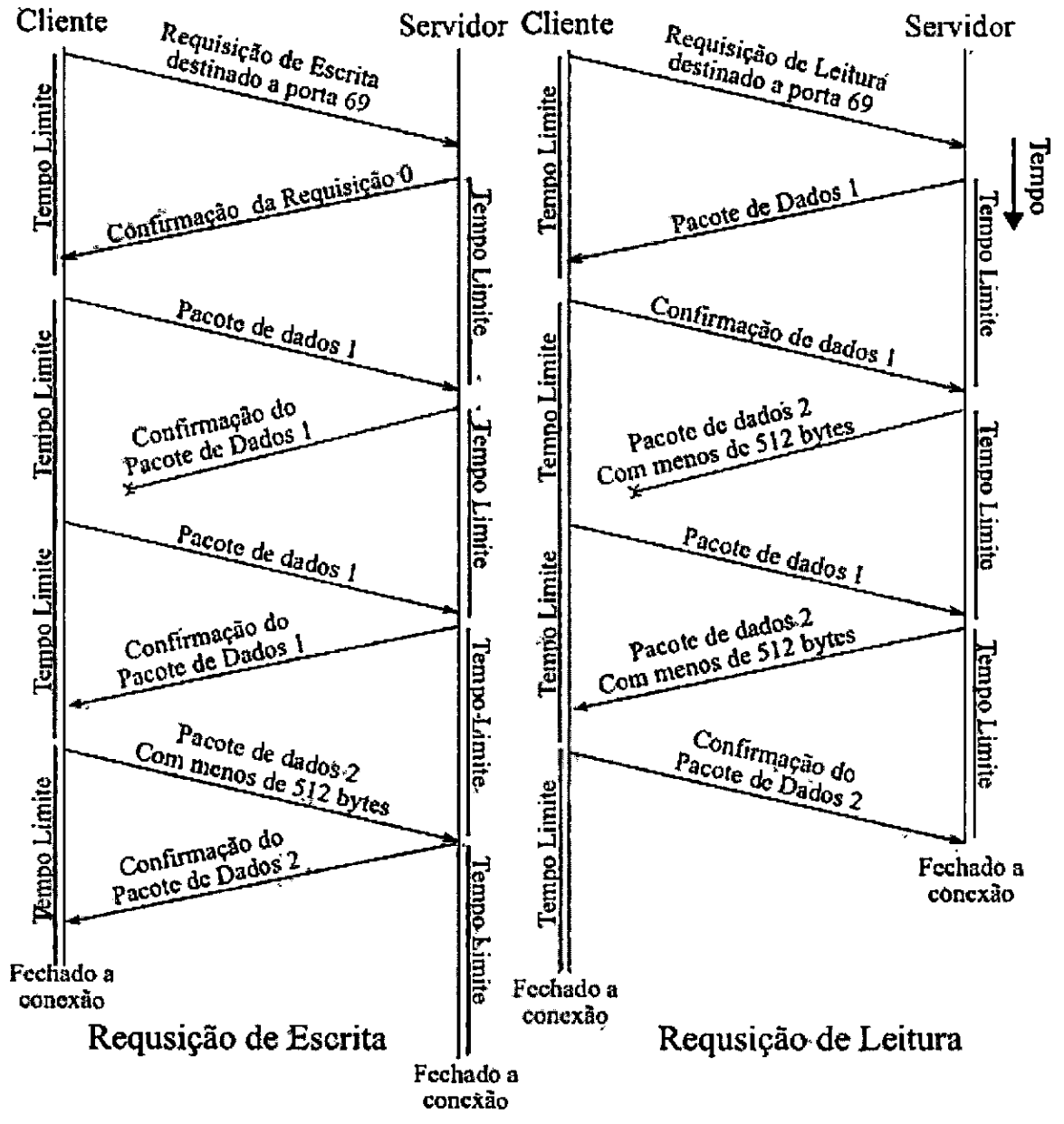

Figura 6 - Seqüência de Eventos do TFTP.
Fonte: REZENDE (2008).

2.8 Network File System (NFS)

Criado pela Sun *Microsystems Incorporated* em 1984, o Network File System ou sistema de arquivos em rede (NFS), tornou-se um padrão da *Internet Engineering Task Force* (IETF), para acesso de arquivos compartilhados, de forma transparente e integrado (TANENBAUM, 2001). O NFS é um protocolo que disponibiliza aos usuários o acesso a recursos num arquivo compartilhado em uma rede TCP/IP. A sua forma de acesso transparente aos recursos possibilita aos usuários a manipulação de arquivos e diretórios remotamente, de forma como se estivèssem localizados em um sistema local, sem haver conexão remota.

Os servidores NFS são do tipo sem informação de estado, ou seja, não dispõem de nenhuma informação do estado de protocolo dos clientes NFS. Scrimger et. al. (2002 p. 279) cita que, "Essa é uma vantagem caso haja falha no servidor, pois o cliente não percebe que o servidor caiu, somente solicita os serviços disponibilizados pelo mesmo". A arquitetura do NFS é constituída de três componentes básicos: o protocolo NFS, um mecanismo de chamada remoto (*Remote Procedure Call* – RPC) e uma representação externa de finalidade geral (*eXternal Data Representation* – XDR). RPC é o responsável pela comunicação entre o servidor e o cliente, e o XDR converge a representação dos dados, possibilitando a comunicação entre máquinas heterogêneas.

Para melhor compreensão do acesso transparente entre máquinas, suponha-se que uma determinada empresa possui um computador com o nome Estoque_1 e que o mesmo armazena todos os relatórios de estoque de produtos no diretório /estoques/reports. Ainda nessa empresa, existe o departamento de vendas, o qual possui um computador chamado de Vendas_1, que é utilizado na efetuação das vendas. Ambos estão configurados numa rede TCP/IP. O departamento de vendas tem a necessidade de acessar o computador Estoque_1 verificando a disponibilidade do produto. Para isso, utiliza-se o protocolo NFS, que interliga o diretório /estoques/reports com outro diretório no PC Vendas_1, através do comando: ***mount -t nfs.*** Essa operação em sistemas UNIX é chamada de montagem, o comando *mount* irá tentar se comunicar com o processo *rpc.mount* no servidor Estoque_1 via RPC. O servidor verificará se a requisição é permitida e retornará um arquivo de confirmação denominado *handle.* Se não for permitida, será exibida uma mensagem de erro correspondente. Caso

contrário, depois de montado o diretório, o departamento de vendas pode acessar os arquivos de estoque, como se fossem arquivos locais.

2.8.1 Utilização do NFS no LTSP

Conforme foi explanado, as estações ou terminais leves *diskless* não possuem mídias de armazenamento local. Devido a esse fato, o LTSP utiliza o NFS para que os terminais tenham seu sistema de arquivos. O NFS irá substituir os arquivos carregados pelo TFTP montando uma nova raiz no servidor, especificamente no diretório */opt/ltsp/i386,* o qual é compartilhado por todas as estações da rede. Segundo Morimoto (2006, p. 259), "[...] o NFS é uma opção para compartilhar sistemas de arquivos entre máquinas Linux, de uma forma prática e estável.".

2.9 X Display Manager Control Protocol (XDMCP)

O *X Display Manager Control Protocol* ou Protocolo de Controle de Gerenciamento de Janelas é um protocolo de acesso remoto que permite rodar aplicativos remotamente. Ele é a base para sistemas que são configurados com servidores centrais e terminais leves, como o LTSP. O X do início da sigla do protocolo faz menção ao sistema *X Window* que, segundo Ferrari (2007, p. 26), "[...] é uma interface gráfica que permite operar no Linux utilizando a mesma filosofia de outros sistemas operacionais baseados em interfaces gráficas como o Windows da Microsoft e o OS/X da Apple, para computadores Macintosh [...]".

O XDMCP é um protocolo bastante simples e rápido, oferecendo um ótimo desempenho à rede local. Outra característica favorável à sua utilização é o baixo consumo de recursos, tanto em nível de servidor, como em de cliente. Segundo Morimoto (2006), ele é um recurso nativo do X, de forma que não necessita de nenhum software adicional, basta ativar o recurso na configuração do KDM ou GDM (os gerenciadores de login usados nas distribuições atuais). Entretanto o XDMCP é um protocolo considerado antigo, que não possui encriptação ou compressão, simplesmente transmite os dados da forma mais simples e rápida possível, de modo que a possibilidade de ocorrer *overhead* (processamento ou armazenamento em excesso, gerando perda de desempenho) é muito pequena. Contudo

devido sua falta de segurança, o XDMCP deve ser usado somente dentro da rede local, executando um *firewall*, para bloquear conexões advindas da internet. O protocolo disponibiliza serviços que possibilitam a autenticação da janela no gerenciador. Outra permissão conforme Packard (2005), é que as informações de configuração possam ser centralizadas no gerenciador; isso diminui o trabalho do administrador de sistemas em grandes redes.

Uma importante definição sobre a operabilidade dos pacotes do protocolo XDMCP é descrita por Packard (2005), citamos abaixo as definições preditas:

Consulta, DifusãoDeConsulta, ConsultaIndireta: pacotes que possuem esses códigos de operação são transmitidos da janela para o gerenciador. Um pacote de *Consulta* é transmitido de uma janela para um cliente específico perguntando se está apto a fornecer um serviço de gerenciamento, este deverá responder enviando um pacote *Disposto* se estiver apto, ou um pacote *Indisposto*, caso contrário. O funcionamento do pacote *DifusãoDeConsulta* se destina a qualquer cliente conectado à rede. Um pacote *ConsultaIndireta* é transmitido a um gerenciador conhecido, que encaminha a requisição a uma coleção de gerenciadores secundários usando um pacote *EncaminhamentoDeConsulta.* Deste modo, a coleção de gerenciadores pode estar agrupada em outra rede; o uso de um gerenciador central reduz a sobrecarga do sistema administrativo.

Encaminhamento de Consulta: é um pacote enviado entre gerenciadores. Quando um primeiro recebe um pacote *ConsultaIndireta*, ele fica responsável por enviar um pacote *EncaminhamentoDeConsulta*, para uma máquina da lista de gerenciadores disponíveis, fornecendo o serviço para a janela usando o mesmo caminho na rede que o pacote de *ConsultaIndireta* original realizou. Os campos de porta e endereço do cliente devem conter um endereço que o gerenciador secundário pode usar para localizar a janela na rede. Cada gerenciador secundário envia um pacote de *Disposto* para a janela se estiver apto a fornecer o serviço. Semelhante ao pacote de *DifusãoDeConsulta,* o de *EncaminhamentoDeConsulta* não requer um pacote de *Indisposto* como resposta quando a máquina não está apta a fornecer o serviço.

Disposto: um pacote de *Disposto* é sempre enviado do gerenciador para a janela. É enviado pelos gerenciadores que podem fornecer serviços de conexão para esta janela e emitido em resposta aos pacotes *Consulta, DifusãoDeConsulta, ou EncaminhamentoDeConsulta,* mas não implica em garantia do fornecimento do serviço (por

exemplo, o gerenciador pode posteriormente decidir que já aceitou muitas conexões).

Indisposto: um pacote de *Indisposto* é sempre enviado no sentido do gerenciador para a janela.

Requisição: um pacote de *Requisição* é sempre enviado da janela para os gerenciadores. Ele é enviado por uma janela para uma máquina específica requisitando um ID de sessão na preparação para um estabelecimento de uma conexão. Se o gerenciador está pronto para fornecer o serviço de conexão para a janela, ele deve retornar um pacote *Aceito* com um ID de sessão válido e estar pronto para a requisição do pacote de *Gerenciar* subseqüente. Caso contrário, ele deve retornar um pacote de *Rejeitado.*

Aceito: um pacote de *Aceito* é sempre enviado do gerenciador para a janela, em resposta a um pacote de *Requisção,* quando o gerenciador está pronto para aceitar a conexão. O ID da sessão é usado para identificar essa conexão e as que a precedem, identificará também a janela em seus pacotes *Gerenciar* subseqüentes. O ID de sessão é um número de 32 bits que é incrementado a cada vez que um pacote *Aceito* é enviado, sendo único durante um longo período de tempo. Se a informação de autenticação é inválida, um pacote de *Rejeitado* é retornado com uma mensagem do seu status.

Rejeitado: um pacote de *Rejeitado* é sempre enviado do gerenciador para a janela, em resposta a um pacote de *Requisição,* quando não pode estabelecer a conexão com a mesma, isto é permitido mesmo que o gerenciador tenha previamente respondido com um pacote de *Disposto* a uma consulta.

Gerenciar: Um pacote de *Gerenciar* é sempre enviado da janela para os gerenciadores, para solicitar que comece uma sessão na janela. Se a identificação está correta, o gerenciador deve abrir uma conexão. Caso contrário, deve responder com um pacote *Recusado* ou *Falhou,* a menos que o ID da sessão seja encontrado em funcionamento ou em uma sessão que ainda não abriu a conexão da janela. Neste último caso, o pacote *Gerenciar* deve ser ignorado.

Recusado: Um pacote de *Recusado* é sempre enviado do gerenciador para a janela. Um pacote *Recusado* é enviado por um gerenciador quando o ID da sessão recebida no pacote *Gerenciar* não corresponde ao ID da sessão corrente. A janela deve assumir que ela recebeu um pacote de *Aceito* antigo e deve reenviar seu pacote de *Requisição.*

Falhou: um pacote de *Falhou* é sempre enviado do gerenciador para a janela, quando ele tem problemas para estabelecer a conexão X inicial em resposta a um pacote *Gerenciar*.

FiqueAtivo: um pacote de *FiqueAtivo* é sempre enviado da janela para o gerenciador, ele pode ser enviado a qualquer tempo durante a sessão por uma janela para descobrir se o gerenciador está funcionando, que deve responder com um pacote *Ativo* sempre que receber esse tipo de arquivo. Isso permite à janela descobrir quando o gerenciador não está mais ativo. Não é exigido que uma janela envie pacotes *FiqueAtivo*. E na falta do recibo de pacotes *Ativos*, não se exige a execução de nenhuma ação específica. A utilidade prevista para esse pacote é terminar uma sessão ativa quando a máquina na qual o gerenciador está instalado ou a ligação de rede falharem. A janela deve manter-se a par do tempo decorrido desde o recebimento do último pacote enviado pelo gerenciador e usar um pacote *FiqueAtivo* quando esse intervalo de tempo for muito grande.

Ativo: um pacote de *Ativo* é sempre enviado do gerenciador para a janela, em resposta a uma requisição *FiqueAtivo*. Se a sessão estiver ativa na janela naquele momento, o gerenciador inclui a ID da sessão no pacote. A janela pode usar esta informação para determinar o status do gerenciador.

2.9.1 Utilização do XDMCP no LTSP

Finalizado o processo de montagem do sistema de arquivos e o mapeamento dos hardwares encontrados nos terminais da rede, o servidor *X Window*, localizado em cada terminal, faz uma requisição XDMCP para o servidor, que em seguida reenvia uma mensagem para o *X Window* exibir a tela de *login* nos terminais, permitindo que os clientes pós logados executem seus aplicativos remotamente pelo servidor através do SSH.

2.10 Secure Shell Host (SSH)

O sistema SSH, escrito por Ttu Ylönen, é um substituto seguro do TELNET. Ele usa autenticação criptografada para confirmar a identidade dos usuários e criptografa toda a comunicação entre duas máquinas (NEMETH et. al., 2002). Já o TELNET, além de muito antigo - pois sua primeira demonstração foi realizada no ano de 1969 - transmite via rede

informações, como senha e *login* em texto puro, que podem facilmente ser rastreadas e utilizadas para acessos indesejáveis ao servidor, possibilitando a uma máquina ser acessada via linha de comando por usuários que tenham o *login* e a senha de uma das contas do sistema. Em relação ao uso do SSH, é permitido o acesso completo ao sistema via terminal, seja via rede ou via Internet, só que se limitam os privilégios do *login* usado. Mendonça (2005, p. 74) afirma que "[...] o SSH é um serviço de *login* remoto, um método muito seguro, pois permite a autenticação e transmissão de dados criptografados [...]". Ele trabalha como servidor e cliente. Sendo assim, ao iniciar o serviço, você pode se conectar a uma máquina remota ou permitir uma conexão externa.

O SSH é um protocolo muito utilizado na administração remota em servidores Linux. Ele possibilita login remoto seguro entre outros serviços de rede sobre uma conexão insegura como a TCP/IP Javvin (2005). Inicialmente esse protocolo só possibilitava a execução de comandos de texto remotamente; depois, foi aprimorado e passou a executar também aplicativos gráficos. No contexto de segurança do SSH, a encriptação dos pacotes é realizada através de chaves assimétricas para fazer a autenticação. Segundo Morimoto (2006), o SSH utiliza chaves (um par de chaves) assimétricas para fazer a autenticação. As chaves assimétricas são um sistema muito interessante. Uma (a chave pública) permite apenas encriptar dados; enquanto a segunda (a chave privada), permite desencriptar as informações embaralhadas pela primeira. Quando um terminal cliente se conecta a um servidor SSH, ambos trocam suas chaves públicas, havendo a troca de informações de forma segura. O SSH, descrito pela *Request For Comments* (RFC) 4251, para Ylonen et. al. (2006a) é composto de três componentes:

- **O protocolo da camada de transporte** fornece autenticação de servidores, confidencialidade e integridade de dados com privacidade garantida pela criptografia dos dados. Opcionalmente, pode oferecer compactação dos dados a serem transportados. A proposta para padronização desse padrão é descrita por Ylonen et. al. (2006b) na RFC 4253.
- **O protocolo de autenticação do usuário** é o componente do SSH responsável por autenticar o usuário para o servidor. Ele funciona sobre a camada de transporte e fornece um único túnel para a conexão SSH. A RFC 4252 possui uma proposta de padronização desse protocolo Ylonen, et. al. (2006c).

- **O protocolo de conexão** com proposta de padronização encontrada na RFC 4254, ylonen, et. al. (2006d), é projetado para executar sobre a camada de autenticação do usuário possibilitando login interativo das sessões, execução remota de comandos e a transmissão de conexões X11 e TCP/IP. Cada um desses serviços é transmitido em um canal de comunicação lógico, que será multiplexado em uma única conexão SSH.

2.10.1 Utilização do SSH no LSTP

A utilização do SSH no LTSP se deu a partir da versão 5.0, o que possibilitou uma maior segurança na realização das autenticações dos usuários dos terminais via rede e o tunelamento dos comandos enviados ao servidor LTSP. O SSH ainda possibilita a compactação dos dados para transmissão, bem como a importação da interface gráfica do servidor pelos terminais.

3 IMPLEMENTAÇÃO DO PROTÓTIPO

Foi implementado um protótipo de servidor LTSP, para que seja demonstrada a viabilidade de inserção da tecnologia na Biblioteca da Universidade Estadual do Piauí, que possui em sua estrutura alguns computadores considerados como de alto desempenho e outros, obsoletos, que deixam os usuários a desejar quanto à velocidade no processamento dos aplicativos operacionalizados pelos mesmos. No protótipo utilizaram-se três computadores: um será utilizado como o servidor, e os outros dois serão os terminais leves ou *thin client's*.

3.1 Arquitetura da Rede

Para a implementação da rede, aplicamos a topologia estrela, onde os computadores clientes estarão conectados ao servidor através de um switch. A figura 7 demonstra a arquitetura do protótipo LTSP, em que se verificam dois terminais clientes e um servidor interligados por um Switch.

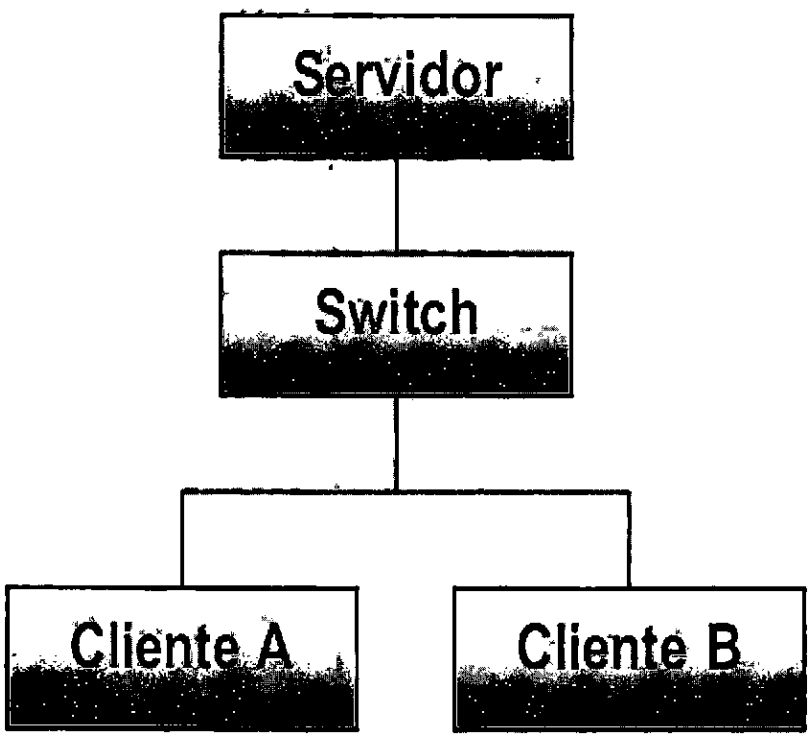

Figura 7 - Arquitetura do Protótipo de Servidor LTSP.
Fonte: FARIA (2009).

3.2 Configuração de Hardware

Tabela 1 – Configuração de Hardware dos Computares do Protótipo.

SERVIDOR	
Processador	Pentium Dual Core
Memória RAM	1 GB
HD	160 GB
TERMINAL 01	
Processador	Celeron D
Memória RAM	256 MB
HD	Não utilizado
TERMINAL 02	
Processador	Celeron D
Memória RAM	256 MB
HD	Não utilizado

3.3 Sistema Operacinal

O Sistema Operacional utilizado na implementação foi o Ubuntu 10.10, o qual se baseia no Debian GNU/Linux. Segundo o site oficial da comunidade Ubuntu no Brasil, ele é um sistema operacional desenvolvido pela comunidade, e é perfeito para laptops, desktops e servidores. Seja para uso em casa, na escola ou no trabalho, o Ubuntu contém todas as ferramentas de que você necessita: desde processador de texto e leitor de e-mails a servidores web e ferramentas de programação (UBUNTU, 2011). O Ubuntu encerra uma metodologia de desenvolvimento em que a cada 18 meses, são disponibilizadas atualizações de segurança gratuitas para desktops e servidores; possui, ainda a melhor infra estrutura de tradução e acessibilidade que a comunidade de Software livre tem a oferecer, tornando o Ubuntu usável para tantas pessoas quanto for possível.

3.4 Configuração do servidor

Antes de abordarmos a configuração do servidor, devemos, relatar que em um servidor LTSP, todos os aplicativos são executados no mesmo servidor por todos os clientes, garantindo o compartilhamento de recursos. Já em desktops tradicionais, o processador fica na maioria do tempo ocioso. Exemplificando o que foi dito, executamos o comando *top* no servidor que será utilizado na rede sem o LTSP estar configurado:

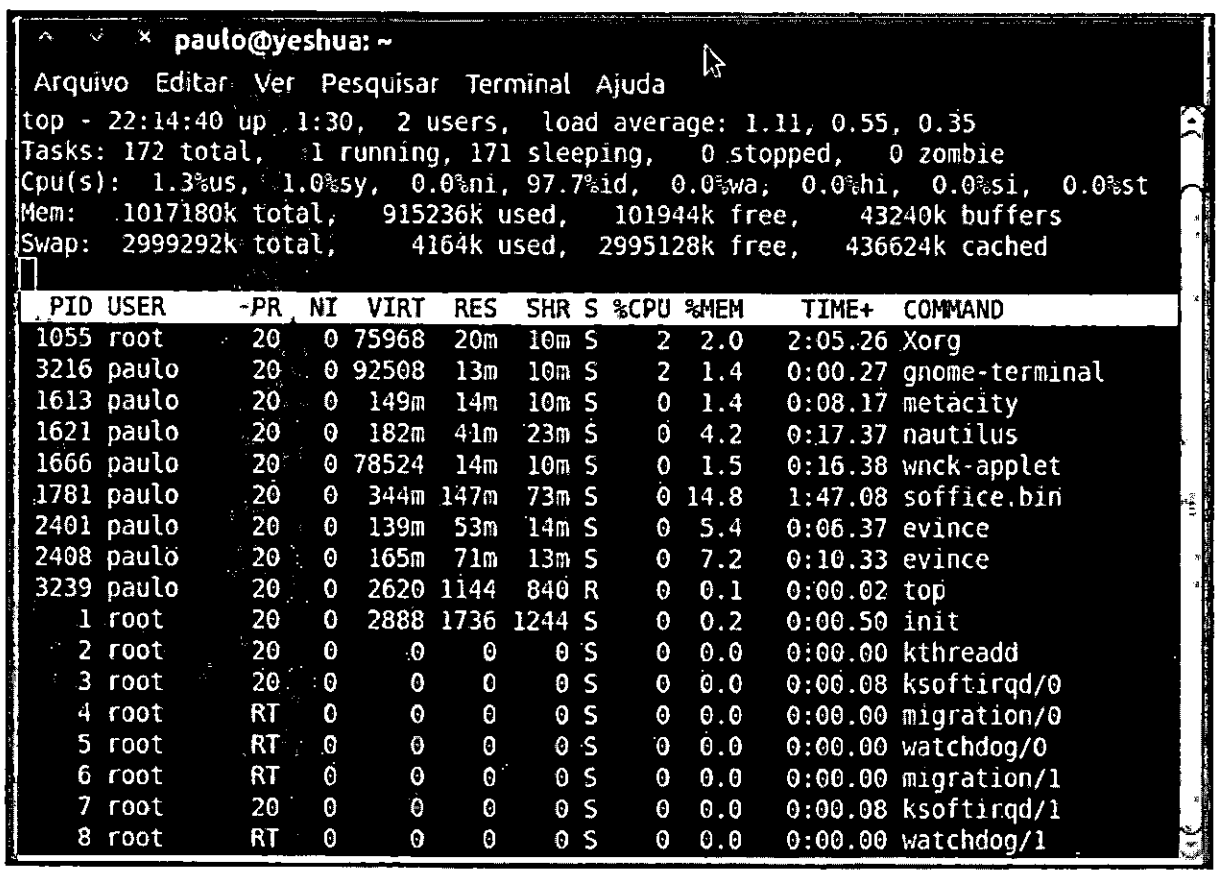

Figura 8 - Visualização do Comando top no Servidor.
Fonte: Primária

Podemos verificar que o comando top retornou algumas informações referentes ao sistema, como, por exemplo, na terceira linha, em que a utilização do processador varia entre 1% e 2%, com oscilações rápidas entre 2% a 6%, apenas 1.3% dos ciclos do processador estão sendo usados para os programas abertos e 1.0% para processos internos, relacionados ao sistema.

Baseando-se nessas informações têm-se uma noção sobre a utilização dos processadores na maioria dos Desktops. Morimoto (2006), afirma que na maioria do tempo a utilização dos processadores ficará sempre abaixo de 5% ou 10%. Mesmo nos servidores de

terminais, a utilização do processador continua sendo baixo devido às requisições dos usuários serem para executar tarefas simples, como navegar na internet e editar textos. Deve-se ainda frisar que a memória RAM é outro dispositivo compartilhado, de forma que os aplicativos são carregados na memória do servidor independente do número de usuários. O servidor carrega o aplicativo somente uma vez e, conforme as requisições forem acontecendo, ele abre novas sessões como se fossem novas janelas, tornando ágil o carregamento dos aplicativos e otimizando a utilização da memória.

3.5 Instalação os pacotes do LTSP no servidor

A primeira etapa a ser exécutada é a instalação dos pacotes do LTSP que compõem a configuração do servidor. Ainda no LTSP 3.0 estavam disponíveis para download os pacotes para várias distribuições: Fedora, Mandrake, Debian, etc. Depois do LTSP 4.0, especificamente no LTSP 4.2, os pacotes de instalação foram unificados, possibilitando a instalação através de um instalador, o *ltsp-utils*, encontrado no link http://ltsp.mirrors.tds.net/pub/ltsp/utils/, que se encarrega de baixar e instalar os pacotes. No caso de nosso protótipo, utilizou-se o LTSP 5.0, pois sua configuração é bem mais prática que a das versões anteriores, pois exigiam modificações nos diretórios do servidor. Na figura 9, o exemplo de configuração do arquivo # */etc/dhcp3/dhcpd.conf,* que define no servidor o protocolo DHCP, para servir um endereço de IP estático para um cliente no LTSP 4.2.

```
# /etc/dhcp3/dhcpd.conf – Configurado para o LTSP 4.2
    shared-network WORKSTATIONS {
   subnet 192.168.0.0 netmask 255.255.255.0 {
    default-lease-time          21600;
    max-lease-time              21600;
    option subnet-mask          255.255.255.0;
    option broadcast-address    192.168.0.255;
    option routers              192.168.0.1;
    option domain-name-servers  192.168.0.1;
    deny unknown-clients;
    option root-path            "192.168.0.10:/opt/ltsp/i386";
    next-server                 192.168.0.10;
   }
}
group  {
    use-host-decl-names     on;
    # terminal 1:
   host ws001 {
    hardware ethernet   00:01:03:2A:90:7B;
    fixed-address       192.168.0.11;
    filename "lts/2.6.17.3-ltsp-1/pxelinux.0";
   }
# Fim da configuração.
}
```

Figura 9 – Configuração do Arquivo /etc/dhcp3/dhcpd.conf para o LTSP 4.2.
Fonte: Primária

Há, portanto a necessidade de configuração manual de cada serviço exigido pelo LTSP, e seus respectivos arquivos de configuração no sistema:

- DHCP - /etc/dhcp3/dhcpd.conf.
- TFTP - /etc/default/tftpd-hpa.
- NFS – (Instalação manual dos pacotes: *portmap, nfs-common* e *nfs-kernel-server* e edição do arquivo: */etc/hosts*).
- XDMCP - */etc/gdm/custom.conf.*
- lts.conf (Configuração de hardware dos terminais utilizados na rede).

Assim, pode-se afirmar que o administrador da rede deverá conhecer toda estrutura dos arquivos e cada diretório, bem como a função dos comandos para que não haja erros durante a configuração do LTSP 4.2. Dessa forma optamos pelo LTSP 5.0, que além de ser uma versão mais atualizada, tem uma instalação mais prática, como será demonstrado a seguir. Através dos comandos *#apt-get install ltsp-server-standalone* e *#ltsp-build-client,*

instalou-se o servidor LTSP e o diretório que será utilizado no boot dos clientes.

Verificando-se, após a instalação dos pacotes, que as pastas presentes no diretório *#/opt/ltsp/i386* são o mesmo conjunto padrão de pastas presentes numa distribuição Linux e estão localizadas no diretório *opt,* que serve para armazenar programas que não utilizam o padrão do sistema e precisam compártilhar arquivos entre vários usuários. Algumas distribuições optam por deixar este diretório no modo "leitura/escrita" para todos os usuários, funcionando assim como um diretório compartilhado entre todos (MENDONÇA, 2005). Os arquivos ficaram dispostos no diretório *opt,* conforme verificamos na figura 8.

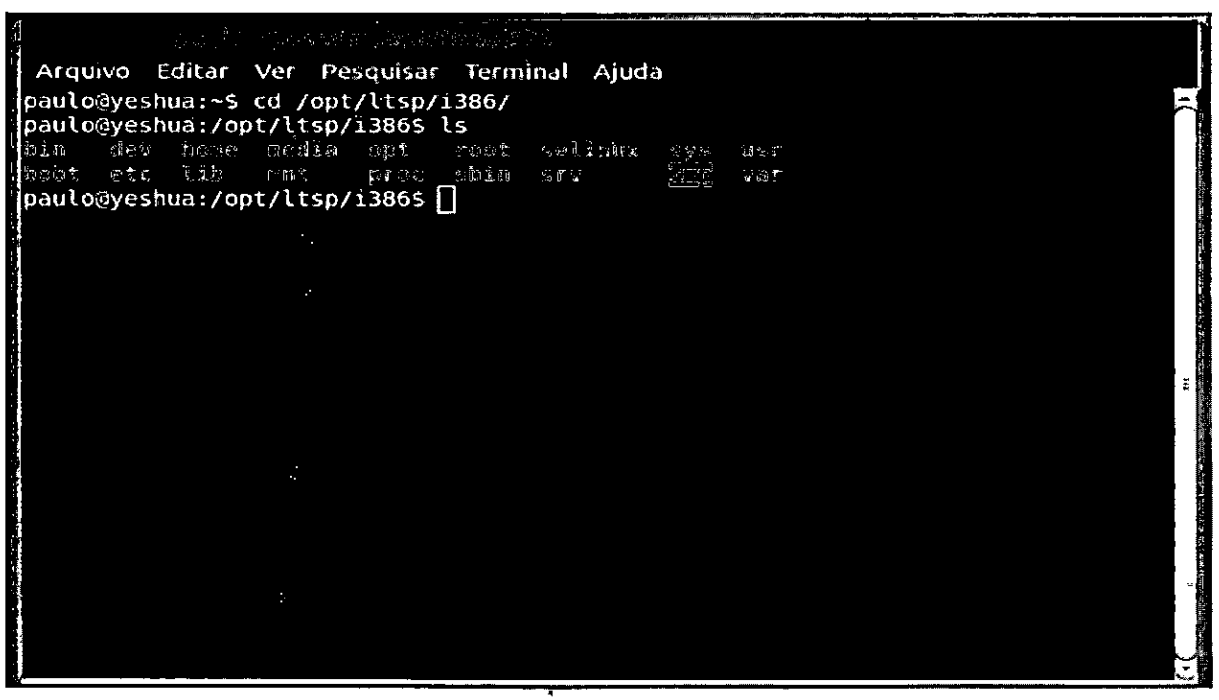

Figura 10 - Visualização dos Arquivos no Diretório /opt/ltsp/i386/.
Fonte: Primária

Em seguida, devem-se configurar os arquivos para preparação do servidor, que receberá as solicitações dos clientes.

3.6 Configuração do DHCP no servidor

O servidor DHCP é o primeiro serviço a ser requisitado pelo terminal, ele responderá aos terminais enviando as configurações da rede e o Kernel. Deve-se instalar o pacote de instalação do dhcp - *dhcp3-server* - automaticamente através do comando *#apt-get*

install ltsp-server-standalone. A configuração do servidor DHCP está localizada no arquivo *#/etc/ltsp/dhcpd.conf*, como segue na figura 11:

```
#Configuração do arquivo /etc/ltsp/dhcpd.conf para LTSP 5.0
 authoritative;
#endereço de rede e máscara de rede
subnet 192.168.0.0 netmask 255.255.255.0 {

  #intervalo de endereços IP pagos pelo servidor DHCP aos terminais
   range 192.168.0.20 192.168.0.250;
   option domain-name "example.com";

   #endereço de DNS
   option domain-name-servers 192.168.0.1;

   #endereço de Broadcast
   option broadcast-address 192.168.0.255;

   #endereço do servidor
   option routers 192.168.0.1;

   #   next-server 192.168.0.1;
   #   get-lease-hostnames true;

   #endereço de máscara de rede
   option subnet-mask 255.255.255.0;

   #Caminho para inicialização do sistema pelos clientes
   option root-path "/opt/ltsp/i386";
   if substring( option vendor-class-identifier, 0, 9 ) = "PXEClient" {
      filename "/ltsp/i386/pxelinux.0";
   } else {
      filename "/ltsp/i386/nbi.img";
   }
}
```

Figura 11 - Configuração do Arquivo /etc/ltsp/dhcpd.conf para LTSP 5.0
Fonte: Primária

Depois da configuração, reinicia-se o servidor DHCP para atualizar a nova configuração com o comando */etc/init.d/dhcp3-server restart.*

3.7 Configuração do TFTP e do NFS no servidor

Para se configurar o NFS e o TFTP, instalou-se o pacote *dnsmasq,* através do comando *#sudo apt-get install dsmasq.* Nesse arquivo encontram-se os parâmetros de configuração dos dois serviços fundamentais para o funcionamento do LTSP. O TFTP, que conforme já foi mostrado, servirá para que os terminais clientes obtenham as informações necessárias para o carregamento da imagem no processo de inicialização; e o NFS, para que o diretório */opt/ltsp/i386* seja virtualmente usado como local pelos clientes. Logo configura-se o arquivo */etc/dnsmasq.conf* da seguinte maneira:

```
#Configuração do arquivo /etc/dnsmàsq.conf para LTSP 5.0
port=0
dhcp-range=192.168.0.20,192.168.0.250,1h

dhcp-option=17,/opt/ltsp/i386

dhcp-vendorclass=etherboot,Etherboot
dhcp-vendorclass=pxe,PXEClient
dhcp-vendorclass=ltsp,"Linux ipconfig"

dhcp-boot=net:pxe,/ltsp/i386/pxelinux.0
dhcp-boot=net:etherboot,/ltsp/i386/nbi.img
dhcp-boot=net:ltsp,/ltsp/i386/lts.conf

dhcp-option=vendor:pxe,6,2b
dhcp-no-override

enable-tftp
tftp-root=/var/lib/tftpboot/
```

Figura 12 - Configuração do Arquivo /etc/dnsmasq.conf para LTSP 5.0
Fonte: Primária

3.8 Configuração do lts.conf no servidor

O próximo arquivo a ser configurado é o *lts.conf,* que fica localizado no diretório */opt/ltsp/i386/etc/.* Nesse arquivo, ainda no LTSP 4.2, eram especificadas as configurações dos terminais, como a resolução de vídeo, o tipo de mouse utilizado, o teclado, a opção de realizar o *swap* via rede, nos casos de terminais que tenham 32MB de memória RAM ou menos, dentre outras configurações de periféricos, dividido-se em duas partes. Na primeira

eram estabelecidas as configurações *default,* ou seja, em caso de não haver nenhuma configuração específica dos terminais da rede, estes a utilizam.

A segunda parte era dividida em sessões onde cada terminal podia ter configurada sua especificidade, como por exemplo, um tipo de teclado US Internacional e mouse serial, sendo que as demais estações estão configuradas no *default* com mouses PS/2 e teclados ABNT2.

A nova versão LTSP 5.0 torna as configurações de hardware dos terminais automáticas, sendo necessária a alteração do */opt/ltsp/i386/etc/lts.conf.* Caso haja necessidade de desabilitar o serviço SSH, como por exemplo, se o administrador da rede prefirir utilizar somente o ambiente gráfico via LDM *(LTSP Display Manager)* - um gerenciador de display semelhante ao GDM e KDM, adiciona-se ao final do lts.conf o comando *LDM_DIRECTX=true.* Após a modificação deve-se atualizar a imagem no servidor via comando *#sudo ltsp-update-image.*

3.9 Configuração do XDMCP no servidor

Todas as etapas até então configuradas possibilitam que os terminais possam dar boot via rede. E para que eles possam rodar os aplicativos, o serviço de gerenciamento remoto de interfaces gráficas deve estar devidamente configurado. Outra atualização no LTSP 5.0, foi o tunelamento via SSH para a importação das interfaces gráficas, automatizando o XDMCP, para que os clientes possam carregar sua tela de login, não havendo a necessidade de alterar-se o arquivo */etc/gdm/custom.conf.*

Após a tela de login ser carregada pelos terminais, é utilizado o *LTSP Display Manager* (LDM), como gerenciador de janela padrão. Por meio do LDM além das informações criptografadas, o login e a senha da sessão também passam a trafegar criptografados (LTSP, 2011). A título de informação, descreve-se na figura 13, um exemplo de configuração do XDMCP no Ubuntu 10.10, para o LTSP 4.2:

```
#/etc/gdm/custom.conf - Configurado para LTSP 4.2
[daemon]
  User= gdm
  Group= gdm
[security]
  DisallowTCP= true
[xdmcp]
  Enable= true
  DisplaysPerHost= 2
  HonorIndirect= false
  MaxPending= 4
  MaxSessions= 16
  MaxWait= 30
  MaxWaitIndirect= 30
  PingIntervalSeconds= 60
  Port= 177
[greeter]
[chooser]
  Multicast= false
[debug]
  Enable= false
```

Figura 13 - Configuração do Arquivo etc/gdm/custom.conf - LTSP 4.2
Fonte: Primária

3.10 Configuração do SSH no servidor

Para configuração do servidor SSH, deve-se instalar via *apt-get* o pacote *openssh-server,* habilitando os serviços do servidor ssh. Após a instalação do serviço deve-se verificar se o mesmo está sendo executado. Para isso executa-se o comando *#/etc/init.d/ssh start,* se tudo estiver dentro dos conformes, o comando retornará ok, como na figura 14:

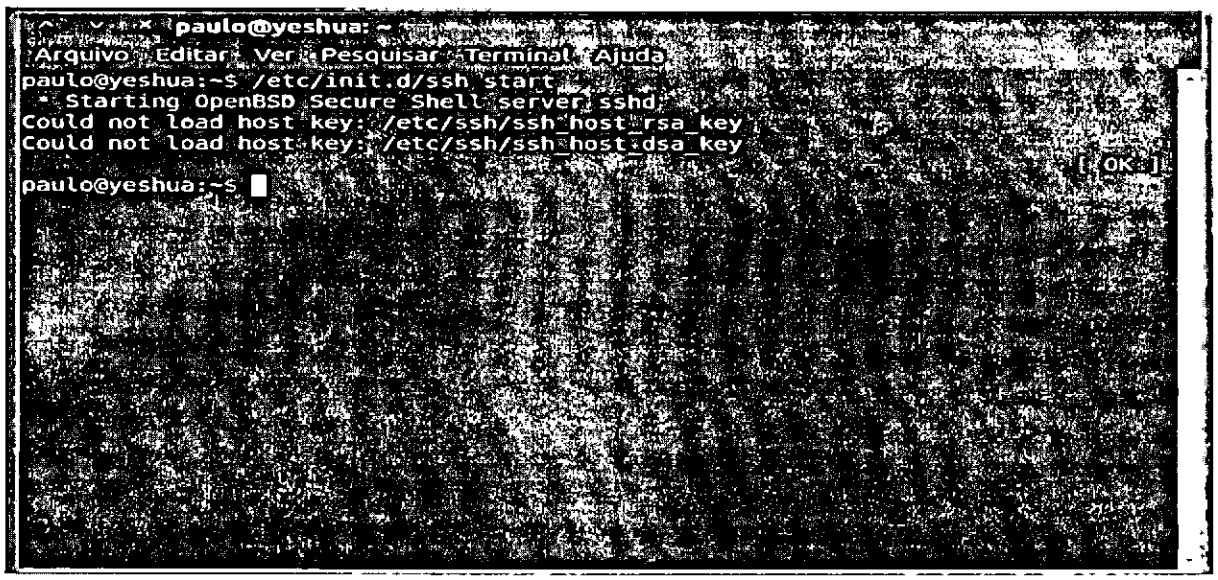

Figura 14 - Mensagem após a Inicialização do SSH LTSP 5.0.
Fonte: Primária

3.11 Configuração dos Terminais

O primeiro passo a ser realizado é a configuração do boot via rede nos terminais. Antes de abordar-se a configuração dos nossos clientes, cabe aqui citar duas maneiras de uma máquina dar boot via rede: uma é através do PXE, um protocolo de boot criado pela Intel. Várias placas de rede atualmente possuem essa tecnologia. Segundo Meier (2006), O tempo em que era preciso "queimar" os boot ROMs de placas de rede exóticas ou mesmo usar disquetes de inicialização para poder iniciar o sistema no terminal leve através da rede pertence à história. Hoje, o padrão da indústria é o PXE (verificar significado), da Intel, que está disponível em praticamente qualquer PC capaz de iniciar um sistema operacional sem uma mídia de boot local, como no caso do nosso protótipo, sua versão é a 2.0. A outra opção é a realização do boot de inicialização através da criação de discos do Etherboot. Para isso estão disponibilizados no site http://www.rom-o-matic.org/ 6 passos a serem seguidos como na figura 15:

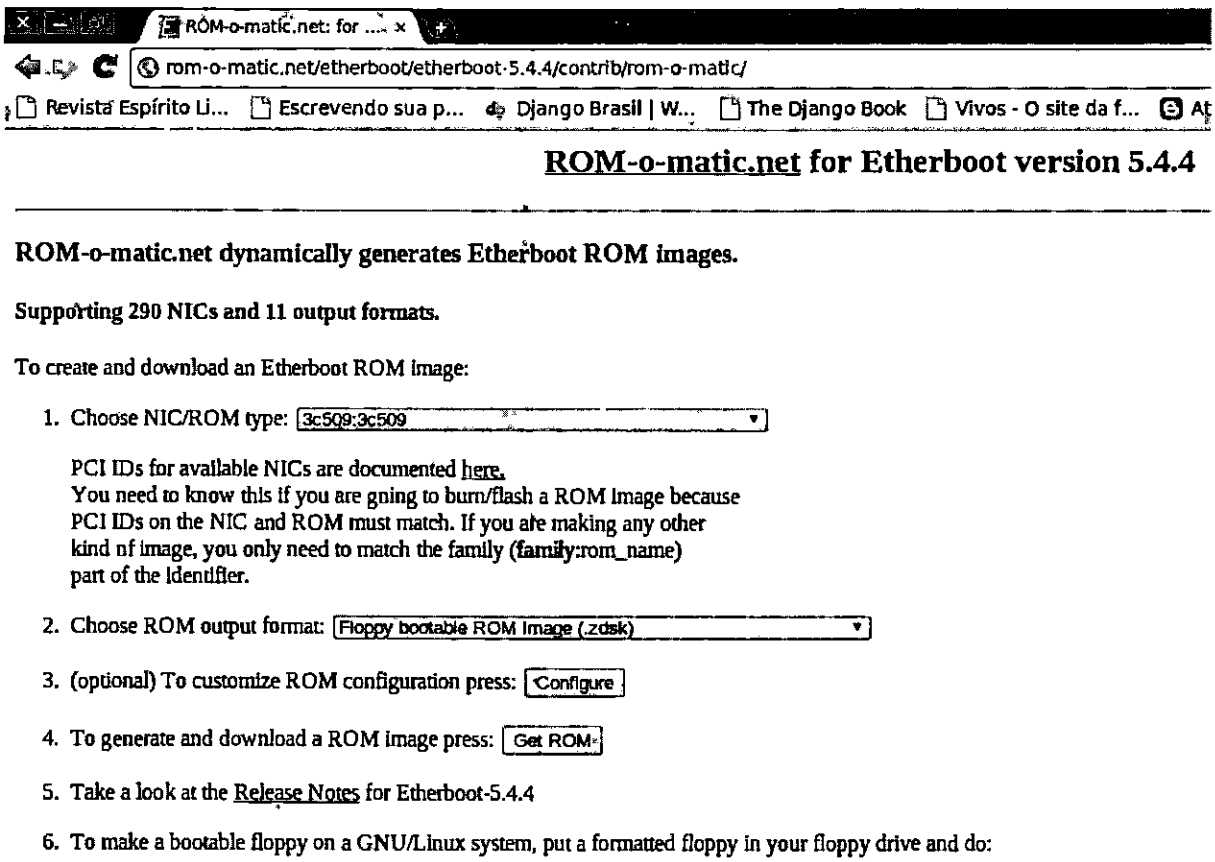

Figura 15 - Página WEB para Gerar o Software de Etherboot.
Fonte: http://rom-o-matic.net/.

A primeira opção *Choose NIC/ROM type* é para escolha do modelos da placa, a segunda, *Choose ROM output format*, é a escolha da extensão do arquivo a ser gravado. Dependendo do caso, é escolhida a opção. No exemplo acima o boot será via disquete, portanto a opção ideal é .zdsk. A terceira opção não é obrigatória e a quarta opção é para gerar o arquivo. Após gravado o disquete com o comando $ dd if=(nome do arquivo de imagem) of=/dev/fd0, configuramos os terminais para inicializarem pelo *driver* de disquete. Para a configuração dos clientes presentes no protótipo, setou-se a inicialização via PXE, através do menu de configuração denominado *setup,* do sistema integrado da placa-mãe ou *Basic Inpu/Output System* (BIOS) finalizando assim a configuração dos clientes.

4 ANÁLISE DO DESEMPENHO

No que se refere à análise e coleta de informações sobre o desempenho da rede, foi utilizado um software padrão do Linux, chamado Monitor do Sistema, versão 2.28.1, que monitora o sistema verificando o consumo de memória, tráfego de rede e desempenho do processador. Dessa maneira dividiu-se a análise em três fases, que serão estabelecidas logo na tabela 2 e descritas na tabela 3. Serão utilizados três computadores, um servidor LTSP e dois terminais clientes, executando alguns aplicativos mais usados em ambientes de estudo como *Browser Google Chromium 12.0,* editor de texto *OpenOffice Writer 3.2,* editor de imagens *Gimp 2.6 e o NetBeans IDE 6.9* para desenvolvimento de sistemas.

Tabela 2 Tabela de Definição das Fazes de Análise.

Fase 1	Após a inicialização das máquinas.
Fase 2	Após o login dos terminais.
Fase 3	Momento de utilização dos aplicativos.

Na tabela 3, tem-se um detalhamento das fases que serão levadas em consideração na análise:

Tabela 3 - Tabela de Descrição das Fazes de Análise.

Fase 1	Nesta fase, os dois clientes solicitarão do servidor os endereços para seu funcionamento, que serão entregues via DHCP. Após estarem endereçados, os terminais procuram o caminho de diretório onde estará o Kernel a ser carregado via TFTP, logo após essa localização os terminais carregarão seu diretório raiz através do NFS, em seguida é exibido a imagem de login via XDMCP.
Fase 2	Nesta fase, os clientes estarão logados no servidor e executando seus aplicativos através do LDM, ou LTSP Display Manager.
Fase 3	Nesta ultima fase os aplicativos: *Browser Google Chromium, OpenOffice Writer 3.212.0, Gimp 2.6* e *NetBeans IDE 6.9,* estarão sendo executados no servidor via SSH.

4.1 Desempenho na Primeira Fase

Na figura 16 são demonstrados os gráficos referentes ao desempenho do processador, consumo de memória RAM e o tráfego da rede, respectivamente. Nessa primeira etapa, como fora mencionado, o terminal 1 recebeu o endereço IP 192.168.0.169, e o terminal 2 o endereço IP 192.168.0.226. Na inicialização, as demais informações são padronizadas, como pôde-se observar na configuração do DHCP. Os mesmos carregam suas imagens de login, o que não requisita muito processamento de servidor. Nos dois núcleos de processamento menos de 7% foram utilizados, já a memória principal está em 25,9% de seu consumo não havendo a necessidade de utilizar a área de *swap,* que serve para suprir caso a mesma necessite. Quanto ao tráfego de rede, pode-se perceber que está existindo uma troca de pacotes entre o servidor e os clientes em bits, registrando, portanto: 0 o número de *bytes* recebidos e 0 o de *bytes* enviados.

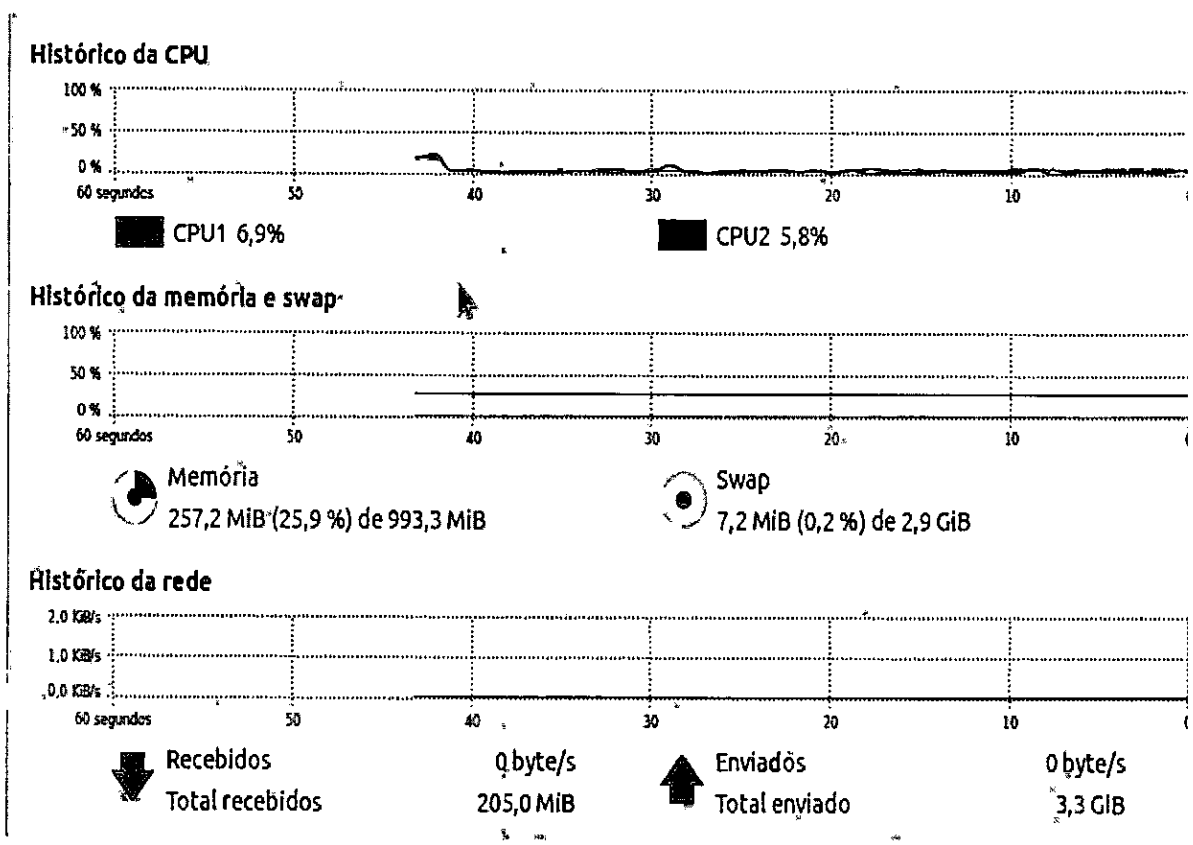

Figura 16 – Sistema de Monitoramento na Fase 1.
Fonte: Primária.

Com base nessas informações, observa-se que nessa primeira fase a instalação do LTSP não exigiu grande capacidade de processamento do processador e de armazenamento no servidor. Concluindo-se, portanto que para execução do LTSP 5.0, pode-se utilizar nessa primeira etapa um servidor com menos requisitos de configuração. Ao disponibilizar as interfaces gráficas do Ubuntu 10.10 aos terminais, através do protocolo XDMCP, não houve nenhuma defectibilidade apresentada, permitindo uma posterior inserção de novos clientes na rede.

4.2 Desempenho na Segunda Fase

A figura 17 demonstra os recursos do servidor utilizados pelos clientes na segunda fase, que consiste na utilização do gerenciador de *display* do LTSP, após a efetuação do login nos dois terminais. Pode-se notar que a porcentagem de utilização dos núcleos 1 e 2 do processador ficou em 4% e 3%, respectivamente, com picos entre 12% a 2%, alternando entre ambos. O consumo de memória RAM foi elevado para 467MB, a qual foi utilizada 21% a mais do que quando os clientes estavam em modo de login. Os pacotes enviados foram a 849 bytes/s, e os recebidos chegaram a 744 bytes/s, aumentado, assim, consideravelmente o tráfego de informações na rede. Nesta fase observou-se um aumento considerável no consumo da memória e do tráfego na rede. Porém, após a efetuação do login, os valores da memória ficaram oscilando em 400MB e 402 MB, e o tráfego da rede, em intervalos de 3 a 4 segundos. Os pacotes enviados e recebidos estabilizavam e 0 bytes, demonstrando a possibilidade de inserção de novas máquinas em nosso protótipo, havendo assim a escalabilidade de sistema.

Outra novidade observada nesta fase foi a capacidade de execução do Sistema Operacional Ubuntu 10.10, pelos terminais de forma rápida, pois segundo Novais (2010), o sistema exige uma configuração mínima de 512 MB, para que o usuário possa utilizar o mínimo de características visuais proporcionada pelo S.O. Dessa forma os clientes ganharam em desempenho e velocidade de execução do sistema à vista dos usuários.

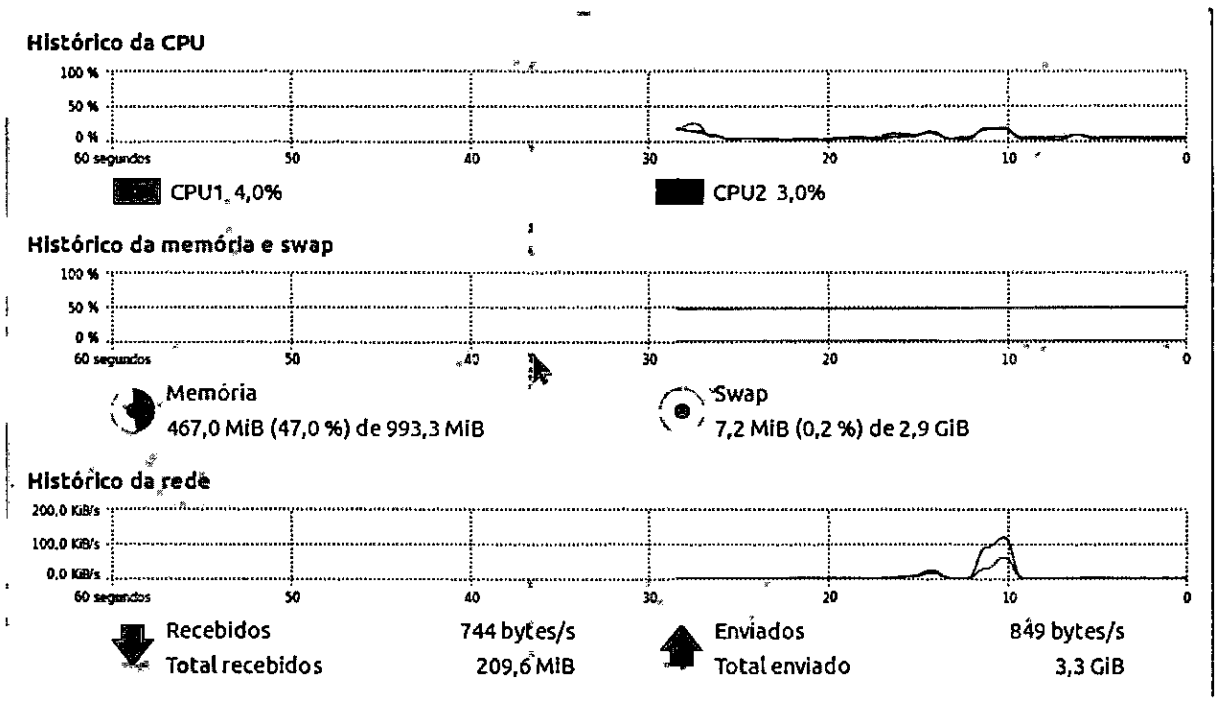

Figura 17 – Sistema de Monitoramento na Fase 2.
Fonte: Primária.

4.3 Desempenho na Terceira Fase

A figura 18 demonstra os recursos solicitados pelos clientes do servido na terceira fase. Nesta etapa de execução do LTSP, as informações estão sendo enviadas e recebidas via SSH. Todos os aplicativos foram executados nos clientes e no servidor. Aqui fase nota-se que o desempenho do primeiro núcleo do processador ficou com 5,9% de utilização, e o segundo, com 16,5%. O consumo da memória RAM foi para 793,3 MB, e a área de *SWAP* consumiu 266 MB. O número de pacotes enviados na rede foi 1,6 KiB/s e o número de pacotes recebidos 2,6 KiB/s. Percebe-se que nesta fase a troca de pacotes foi mensurada em KiB, ultrapassando os 1024 bytes. Os valores apresentados nesse teste foram realizados com todos os aplicativos listados na tabela nº 3, rodando nos dois clientes e na máquina do servidor, de forma que como podemos analisar na figura 18, o sistema não ficou sobrecarregado, os dois núcleos do processador ficaram a menos de 6% e 17%, sendo assim faltou muito para o 100% da capacidade total do processador. Na memória RAM com apenas 9,1% área de SWAP utilizada, pode-se afirmar que novos softwares poderiam ser utilizados localmente ou em outros hosts na rede. E quanto aos pacotes enviados e recebidos, não houve problema na velocidade de comunicação entre os terminais da rede, verificando-se, então assim a eficácia

do LTSP, na utilização dos aplicativos. Avaliando-se a capacidade dos terminais com apenas 256 MB de memória RAM, executarem o ambiente de desenvolvimento NetBeans que conforme NetBeans (2010), os requisitos mínimos para execução do *Integrated Development Environment* (IDE) para o Ubuntu, é 512 MB, comprovando mais uma vez o aumento de desempenho adquirido pelos terminais.

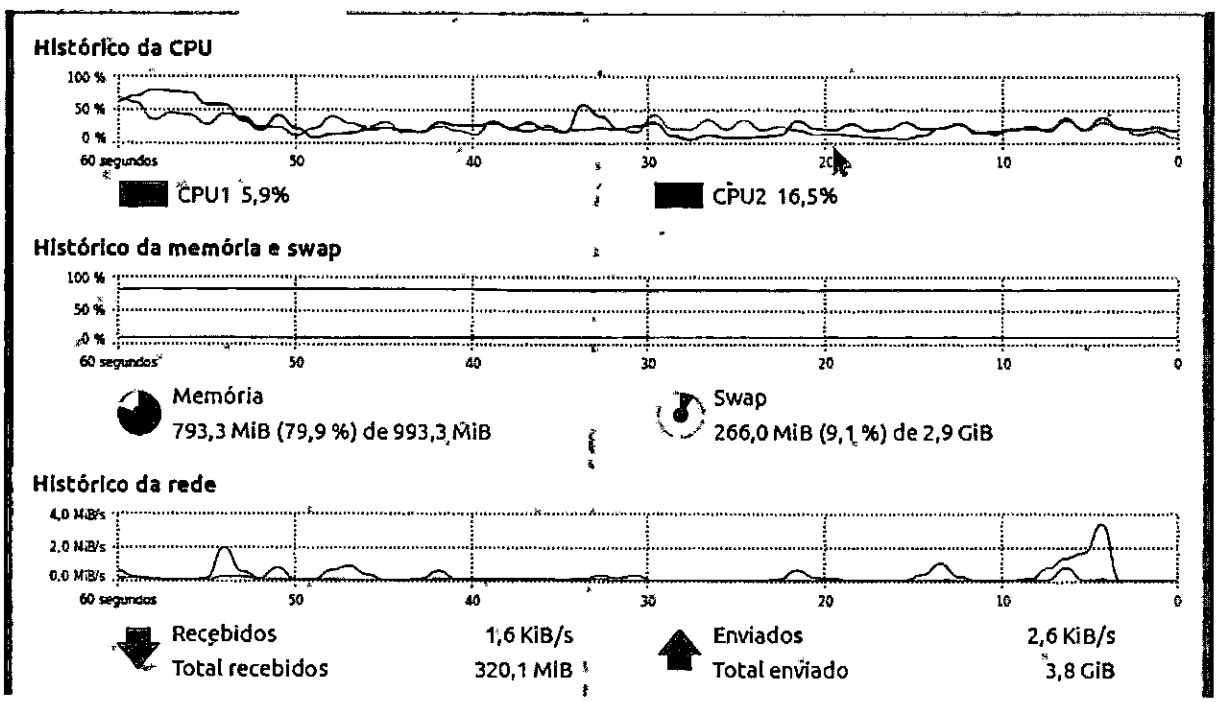

Figura 18 – Sistema de Monitoramento na Fase 3.
Fonte: Primária.

5 CONCLUSÕES

Este trabalho propôs criar um protótipo de uma rede de terminais leves com LTSP, demonstrando a viabilidade de inserção dessa tecnologia na biblioteca da Universidade Estadual do Piauí campus Parnaíba, tornando a possibilidade de utilizar máquinas que tenham baixo desempenho em estações de trabalho com usabilidades viáveis. Através do teste realizado, concretizou-se a proposta de demonstração de viabilidade de utilização da tecnologia LTSP.

A principal contribuição deste trabalho é criar uma alternativa de reaproveitamento de máquinas obsoletas, utilizando-as como terminais leves aumentando, assim, sua viabilidade de uso. Ao longo deste, algumas etapas foram caracterizadas por diferentes aspectos, dentre eles analisou-se: a versatilidade de instalação e configuração do LTSP, a disponibilização de novos programas através do servidor aos terminais, problemas com hardwares obsoletos e, finalizando, fez-se uma avaliação do desempenho do sistema com uma demonstração de uso.

A versatilidade de instalação e configuração do LTSP: esse aspecto foi escolhido devido a facilidade que se teve após a migração do LTSP 4.2 para o LTSP 5.0, onde foram encontradas muitas dificuldades no anterior, por exemplo, quando se encontrou um erro nos primeiros testes de conexão de um terminal ao servidor, ao ligar o cliente, carregava as informações através do DHCP. Na hora de encontrar o caminho diretório via TFTP para carregar o kernel, o cliente apresentava o seguinte erro *TFTP error 1 (File not found).* Após várias tentativas frustradas para corrigir o erro, decidiu-se migrar o servidor para o LTSP 5.0, por meio do qual se obteve sucesso na instalação no GNU/Linux Ubuntu 10.10. Através de alguns *scripts,* instala-se no servidor a imagem de kernel a ser utilizada pelos terminais, como está presente na configuração do servidor.

A disponibilização de novos programas através do servidor aos terminais: o sistema escolhido, Ubuntu 10.10, possui a funcionalidade de instalação de softwares gratuitos, e, dependendo das necessidades dos usuários, pode-se configurar diversos aplicativos a serem utilizados pelos clientes, de forma que a instalação e desinstalação só ocorre no servidor. Os demais clientes apenas rodarão os aplicativos via rede, não precisando de instalações locais em cada host.

Problemas com hardwares obsoletos: antes de se chegar aos clientes configurados em nosso protótipo, houve a tentativa de reativar máquinas muito antigas, como por exemplo, a configuração de um terminal com 16 MB de memória RAM e processador Pentium MMX, 166 Mhz. O mesmo não suportava carregar a imagem de Kernel via servidor. Outra dificuldade foi tentar trabalhar com a mídia de disquetes, que, em sua grande maioria apresentava falhas nos cilindros de gravação. Cita-se a dificuldade de trabalhar com terminais com memórias SIMM, muitas se encontravam queimadas e outras queimaram no decorrer da pesquisa.

Avaliação do desempenho: pôde-se verificar, através da avaliação feita no protótipo, que o LTSP é uma ferramenta rápida e de grande impacto, mas que ao se optar por instalá-la deve-se atentar para qual sistema se deve utilizar, pois, dependendo do sistema, alguns terminais não poderão carregar o kernel, ou por conta de memória insuficiente ou devido o processador não suportar tal configuração. Portanto, em caso de terminais muito antigos deve-se optar por sistemas com menos exigências de hardware. Também foi observado que para o Ubuntu 10.10, 256 MB de memória RAM dos clientes é suficiente para atender aos requisitos do sistema. Assim, concluiu-se que a ferramenta LTSP pode trazer grandes benefícios aos que decidirem aderir a tal ferramenta, sabendo que é preciso fazer um estudo de viabilidade, pois, dependendo da finalidade, para qual será empregada, as implementações irão variar de acordo com as mesmas.

Como trabalhos futuros sugerem-se: (a) A implantação de uma rede de terminais leves com LTSP na biblioteca de UESPI; (b) A utilização do LTSP como forma de inclusão digital; (c) Um comparativo entre o LTSP 4.2 e o LTSP 5.0, (d) Rede LTSP: como proposta de redução de gastos empresariais.

REFERÊNCIAS

BROOKSHEAR, J. Glenn. **Ciência da Computação:** Uma Visão Abrangente. 7ª ed.São Paulo: Editora Bookman, 2003. 515 p.

CAMPOS, Eliziel Gabriel. **LTSP-Linux Terminal Server Project:** Implantação de Telecentro em uma Escola Indígena. Campo Grande – MS, 2009. 09 p. Projeto de Pesquisa apresentado no Curso de Tecnologia em Redes de Computadores, Universidade Católica Dom Bosco- UCDB. Disponível em: < http://www.rededesaberes.org/3seminario/anais/textos/RELATOS_POSTERS_BANNERS%20PDF/GT%207C-08%20-%20Eliziel%20Gabriel20Campos.pdf>. Acesso em 02 de fevereiro de 2011.

COMER, Douglas. **Interligação de Redes com TCP/IP**. 5. ed. Rio de Janeiro: Editora Campus, 2006. 468 p.

COULOURIS, George; DOLLIMORE, Jean; KINDBERG Tim. **Sistemas Distribuídos – Conceitos e Projeto.** 4. ed. São Paulo: Editora Bookman, 2007.

FARIA, Clístenes P. de Sousa. **Thin Clients:** Soluções, Implementação e Desempenho. Teresina – Piauí, 2009 54 p. Monografia de Graduação (Bacharelado em Ciências da Computação), Universidade Estadual do Piauí – UFPI. Disponível em <http://www.ufpi.br/subsiteFiles/eml/arquivos/files/tcc/clistenes.pdf>. Acesso em 23 de dezembro de 2010.

FERRARI, Fabrício A. **Curso Prático de Linux.** São Paulo: Digerati Books, 2007. 128 p.

GHOSH, Sukumar. **Distributed Systems**: an algorithmic approach. Boca Raton: Chapman & Hall, 2007.

JAVVIN TECHNOLOGIES. **Network Protocols Handbook**. Saratoga: Javvin Technologies, 2005. 340 p.

LTSP. **Linux Terminal Server Project.** Disponível em: <http://www.ltsp.org/>. Acesso em 02 de janeiro de 2011.

MEIER, Wilhelm; DWORSCHAK, Jan; MÜLLER, Markus. **Rede de Terminais Leves**: Economia de Recursos. *Revista Linux Magazine, São Paulo, nº 20, 46-53, jun. 2006.*

MENDES, Jorge M. Dos Santos. **Cidade Digital com Terminais Leves.** Guaretinguetá – SP, 2006. 12 p. Artigo apresentado na Faculdade de Tecnologia de Guaretinguetá, FATEC-GT. Disponível em: <http://www.portallivre.org/blog_fatecti/jorge_ltsp_fatec.pdf>.Acesso em 28 de janeiro de 2011.

MENDONÇA, Tales Araújo. **Manual de Sobrevivência:** Dicas e comandos do mundo linux. Santa Cruz do Rio Pardo, SP: Editora Viena, 2005. 112p.

MICHELON, Gisane A.; HILD Giancarlo F. **Reutilização de Computadores Osoletos com a implementação de um servidor de Terminais GNU/Linux.** *Revista IP Informática Pública, Belo Horizonte, v. 11. nº 01, 89-105, jun. 2009.*

MICROSOFT. **Requisitos de Sistema do Windows 7.** Disponível em: <http://windows.microsoft.com/pt-BR/windows7/products/system-requirements>. Acesso em 12 de janeiro de 2011.

MORIMOTO, Carlos E.. **Linux Redes e Servidores, Guia Prático. 2 ed.** Porto Alegre: GDH Press e Sul Editores, 2006.

NEMETH, Evi; SNYDER, Garth; HEIN, Trent. **Linux Administration Handbook.** Upper Saddle River: Prentice Hall PTR, 2002. 890 p.

NETBEANS. **Notas de Versão da versão 6.9.1 do NetBeans IDE .** ORACLE, 2010 Disponível em : <http://netbeans.org/community/releases/69/relnotes_p t_BR.html#system_r equirements>. Acesso em: 12 Jul. 2011.

NOVAIS, Cláudio. UBUNTUED: **Como Instalar o Ubuntu 10.10 Maverick Meerkat.** Braga – Portugal, 2010. <http://ubuntued.info/como-instalar-o-ubuntu-10-10-maverick-meerkat>. Acesso em: 04 Jun. 2011.

PACKARD, Keith. X Consortium. "**X Display Manager Control Protocol**". Laboratory for Computer Science Massachusetts Institute of Technology. Disponível em: <http://www.x.org/docs/XDMCP/>. Acesso em: 05 de fevereiro de 2011.

RICHARDS, David. **Linux Thin Client Networks Design and Deployment**: A quick guide for System Administrators. Birminghan – Munbai: Packt Publiching, 2007. 173 p.

REZENDE, Bruno A. **Análise de Desempenho de Terminais Leves em Laboratórios de Informática.** Lavras – MG, 2008. 77 p. Monografia de Graduação (Bacharelado em Ciências da Computação), Universidade Federal de Lavras – UFLA. Disponível em <http://www.bcc.ufla.br/monografias/2008/PrimeiraTurma.html>. Acesso em 12 de janeiro de 2011.

RODRIGUEZ, Martins V.; FERRANTE, AGUSTIN J. **Tecnologia de Informação e Gestão Empresarial.** Tradução: Washington Luiz Salles e Louise Anne N. Bonitz. Rio de Janeiro: E-Papers, 2000. 448 p.

RUSCHEL, André. **Terminais sem HardDisk: aprenda a configurar um servidor de boot remoto no Windows Server.** Rio de Janeiro: Brasport, 2009.

SCRIMGER, Bob; LASALLE, Paul; PARIHAR, Mridula. **TCP/IP:** A Bíblia; Tradução de Edson Furmankievicz, DocWare Traduções Técnicas. Rio de Janeiro: Elsevier, 2002.

TANENBAUM, Andrew S. **Modern operating systems**. 2nd ed. Upper Saddle River: Prentice Hall, 2001. 951 p.

TANENBAUM, Andrew S. **Organização estruturada de computadores**. 5. ed. São Paulo: Pearson Prentice Hall, 2007.

TANENBAUM, Andrew S. **Redes de Computadores**. 4ª ed. São Paulo: Campus, 2003.

TURBAN, Efraim; WETHERBE, James C.; MCLEAN, Ephraim. **Tecnologia da Informaçao para Gestao: Transformando os Negócios na Economia Digital.** 3 ed. São Paulo: Editora Bookman, 2002.

UBUNTU. **O que é o Ubuntu?.** Disponível em: <http://www.ubuntu-br.org/ubuntu>. Acesso em 18 de janeiro de 2011.

UBUNTU DOCUMENTATION. **UbuntuLTSP.** Desenvolvido por: CANONICAL, 2011. Disponível em: <https://help.ubuntu.com/community/UbuntuLTSP>. Acesso em: 19 Jul. 2011.

YLONEN, T.; LONVICK, C.; ED.; **The Secure Shell (SSH) Authentication Protocol.** Internet Engineering Task Force (IETF), 2006b. (Request for Comments: 4253). Disponível em:<http://www.ietf.org/rfc/rfc4253.txt>. Acesso em: 15 Jul. 2011.

YLONEN, T.; LONVICK, C.; ED.; **The Secure Shell (SSH) Authentication Protocol.** Internet Engineering Task Force (IETF), 2006c. (Request for Comments: 4252). Disponível em: <http://www.ietf.org/rfc/rfc4252.txt>. Acesso em: 15 Jul. 2011.

YLONEN, T.; LONVICK, C.; ED.; **The Secure Shell (SSH) Connection Protocol.** Internet Engineering Task Force (IETF), 2006d. (Request for Comments: 4254). Disponível em: <http://www.ietf.org/rfc/rfc4254.txt>. Acesso em: 15 Jul. 2011.

YLONEN, T.; LONVICK C.; ED.; **The Secure Shell (SSH) Protocol Architecture.** Internet Engineering Task Force (IETF), 2006a. (Request for Comments: 4251). Disponível em: <http://www.ietf.org/rfc/rfc4251.txt>. Acesso em: 15 Jul. 2011.