S. A. M. Menezes, M.O. Neres and T.B. Santos

Thin Clients on Linux Terminal Server Project

***Abstract—* [[1]](#footnote-1) This article seeks study the LTSP' technology. Through of the analysis about protocols of communication, like the DHCP (*Dynamic Host Manager Control Protocol*), NFS (*Network File System*), TFTP (*Trivial File Transfer Protocol*) and XDMP (*X Display Manager Protocol*) available for the Linux distributions. These technologies made many older computers or low performer computers use these concurrently the resources available. In the server, there by the architecture is network client/server. The goal of this paper is studying this technology, analyzing the functioning of all the protocols and therefore prove the efficiency of yours implementing in environments that requires low power of processing.**

*Keywords—* LTSP, Protocols e PXE.

# I. INTRODUÇÃO

O

computador é uma importante ferramenta de trabalho na grande maioria das empresas. Entretanto, sua aquisição e a necessidade de atualização do parque tecnológico, por vezes, demandam um grande investimento da empresa. Estas chegam a gastar e investir 7,5% da sua receita em TI dentro do mercado brasileiro, com tendência de aumento anual, afinal esse valor dobrou nos últimos 14 anos [1]. Porém, por ser um artigo eletrônico, o seu descarte é muito frequente e sua reutilização, muito baixa. A existência de 136 de milhões de computadores em uso, somados a de cerca de 272 milhões de telefones fixos e móveis, é motivo de comemoração no país, contudo o destino de tais aparelhos, não é discutido com devida atenção. O que totaliza mais de 200 milhões de toneladas de resíduo eletrônico por ano. Do total, 17% desse lixo são armazenados nas próprias residências, fazendo com que leve mais tempo para ser descartado. Considerando que o ciclo de vida útil dos computadores é de cinco anos e caso não seja adequadamente descartado, toneladas de substâncias nocivas como chumbo, mercúrio, cádmio e berílio, serão despejadas na natureza. [2]

Essas máquinas, por estarem obsoletas, deixam de ter utilidade para empresa e, na maioria das vezes, são descartadas uma vez que essas poderiam ser utilizadas por comunidades carentes. Afinal, há uma grande parcela de excluídos digitais no Brasil, sendo os dois principais fatores responsáveis: a falta de recursos financeiros para comprar computadores e pagar uma mensalidade de provedor de internet; e a ausência de uma infraestrutura de acesso à internet em inúmeras regiões do país, o que inviabiliza o uso a rede mesmo àqueles que têm condições de comprar um computador. [3]

Uma solução encontrada para solucionar o problema da defasagem dessas maquina foi o Linux Terminal Server Project (LTSP). James McQuillan conta que iniciou o projeto LTSP em 1999 para suprir a necessidade de um de seus clientes em Detroit, Michigan USA, Binson's Home Health Care Centers[4]. Além disso, segundo o LTSP ORG os benefícios são diversos e entre eles estão: redução de custos e segurança [5].

Dessa forma, o presente trabalho busca estudar o funcionamento de cada protocolo dentro da tecnologia LTSP. Tendo em vista que essa solução tecnológica se apresenta muito palpável, porém pouco popular. Buscaremos mostrar o desempenho desse sistema em relação ao modelo de organização tradicional a fim de torná-la mais aceitável em ambientes que disponha de poucos recursos.

# II. A TECNOLOGIA LTSP

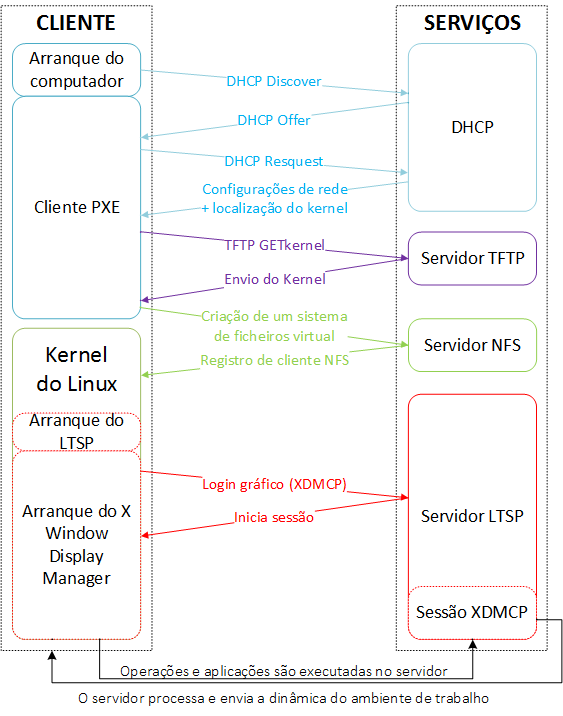
Linux Terminal Server Project é um projeto de código aberto. Criado por James McQuillan, nos Estados Unidos da América, em 1999 e, atualmente, conta com desenvolvedores em todo o mundo. É um conjunto de software que transforma uma instalação normal GNU/Linux em um servidor de terminais. Trata-se de um conjunto de protocolos de comunicações que, em conjunto, funciona a maneira que utiliza a máxima capacidade de processamento do servidor. Esta irá ser o responsável por receber todos os processos e tratá-los. Com isso, o LTSP torna-se uma solução atrativa para projetos que visam a diminuição de gastos. Por não demandar o alto desempenho dos terminais, os custos caem bastante com hardware, o mesmo acontece com o software porque essa tecnologia dispensa o custo de licenças proprietárias.  
 Os protocolos de comunicações em suas especificidades, são de extrema importância no funcionamento dessa tecnologia. Entre eles, podemos citar o DHCP, NFS, TFTP e XDMCP.

ARQUITETURA CLIENTE-SERVIDOR

Numa arquitetura cliente-servidor, teremos um terminal que será o servidor esse, por sua vez, atenderá às requisições de outros terminais. Uma característica dessa arquitetura é que ela não permite que *hosts* se comuniquem diretamente sem que, antes, estabeleça uma comunicação com o servidor.

Muitas vezes, em uma aplicação cliente-servidor, apenas um servidor é incapaz de acompanhar todos os pedidos de clientes. Por exemplo, um site de rede social popular pode rapidamente tornar-se sobrecarregado, se tiver apenas um servidor para processar todos pedidos. Por esse motivo, um centro de dados, abrigando um grande número de *hosts*, é muitas vezes usado para criar um poderoso servidor virtual.

Os existem alguns serviços de Internet mais populares como as ferramentas de busca, baseado na Web (por exemplo, Google e Bing), comércio Internet (por exemplo, Amazon e e-Bay) e-mail (por exemplo, Gmail e Yahoo Mail), redes sociais (por exemplo, Facebook e Twitter) - empregam um ou mais centros de dados [6].

**

III. PROTOCOLOS UTILIZADOS

O protocolo é o responsável pelo controle de uma conexão entre sistemas computacionais. Pode ser definido como a "linguagem" utilizada pelos dispositivos de uma rede, a qual é baseada em regras e possuí objetivos definidos. Então, para que todos os dispositivos de uma rede consigam conversar entre si, todos eles deverão estar utilizando a mesma linguagem, isto é, um mesmo protocolo. Ou sucintamente, pode ser definido, como um conjunto de regras que governa a comunicação. Alguns dos protocolos utilizados na tecnologia LTSP são: TCP/IP, DHCP, NFS, TFTP, XDMCP, que iremos estudar adiante. [7] [8]

TCP/IP

O TCP/IP é atualmente um dos protocolos mais bem equipados e robustos. Ele possui quatro camadas que serão denominadas de: Aplicação, Transporte, Internet e Interface com a Rede.

DHCP

*Dynamic Host Configuration Protocol* é o protocolo responsável por automatizar as configurações do protocolo TCP/IP nos dispositivos de rede. Esse protocolo atua na camada de aplicação da pilha de protocolos que compõe o TCP/IP. Um dos motivos pelos quais esse protocolo é utilizado na tecnologia LTSP é porque ele se baseia na arquitetura cliente-servidor. Sem o uso do DHCP, o administrador da rede e a sua equipe teriam que configurar, manualmente, as propriedades do protocolo TCP/IP em cada dispositivo de rede. Com o uso do DHCP esta tarefa pode ser completamente automatizada. [9]

Figura 1. Diagrama temporal do arranque remoto do LTSP.

NFS

*Network File System* é um protocolo de distribuição de arquivos que foi criado em 1984 pela Sun Microsystems. Ele permite que o usuário acesse os arquivos via rede como se estivesse acessando localmente. Além disso é uma aplicação cliente/servidor que permite que opcionalmente o armazenamento e o *update* do arquivo seja feito de um computador remoto como se estivesse usando no próprio computador do usuário. O NFS permite que um sistema compartilhe seus diretórios e arquivos com outros sistemas através de uma rede. Usando NFS, os usuários e programas podem acessar arquivos em sistemas remotos quase como se fossem arquivos locais. [10]

TFTP

*Trivial File Transfer Protocol* é um simples protocolo de transferência de arquivos, semelhante ao FTP. Ele é geralmente usado para transferir pequenos arquivos entre um terminal e a rede, da mesma forma serve remotamente, a partir de um servidor. Outra caraterística desse protocolo, é pelo fato dele ser simples, não necessita de autenticação ou verificação de erros. A definição de TFTP foi modificado em 1995 para criar um mecanismo para negociação transferir os parâmetros. A opção de tamanho do bloco é um desses parâmetros e foi adicionado à definição do protocolo em 1998. Outros incluem a possibilidade de utilização de um pacote de TFTP padrão para obter uma notificação do tamanho total do arquivo a ser transferido. Antes desta opção, foi criado um programa de recepção, o qual sempre continuava a esperar mais blocos, sem saber quantos seriam transferidos. Como cada bloco tinha que ser de 512 bytes de comprimento, o recebimento de um bloco menor sinalizava o fim da transmissão. [11]

XDMCP

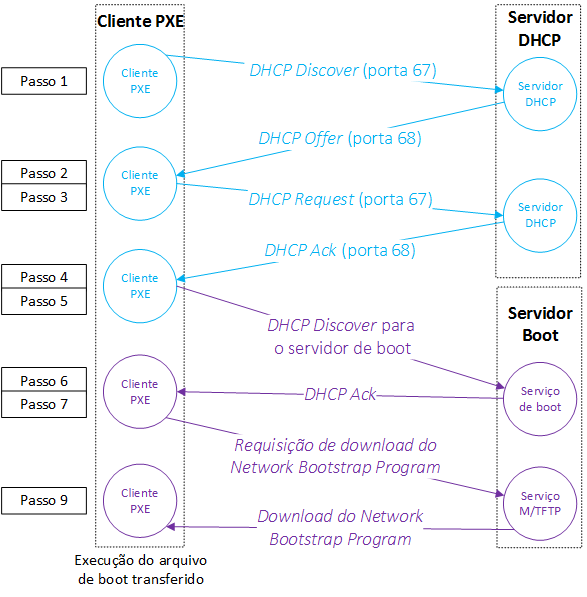
É um protocolo de compartilhamento de desktop utilizado pelo X. Originalmente o X foi desenvolvido para ser usado em mainframes rodando Unix, usados em conjunto com estações de trabalho que se limitavam a exibir as imagens de tela dos aplicativos executados no servidor. Na década de 80 o hardware necessário para rodar aplicativos gráficos e produzir efeitos era muito caro, por isso compartilhar um servidor caro entre vários clientes mais simples e baratos era o melhor custo benefício. Graças a isso o X foi desenvolvido sobre um protocolo bastante sólido e rápido de comunicação via rede. As imagens e gráficos são transmitidos na forma de comandos que consomem relativamente pouca banda da rede e são rapidamente processados pelo destinatário, fazendo com que apesar de rodar a distância, o usuário não perceba demora na atualização das imagens. [12]

IV. A INICIALIZAÇÃO DOS TERMINAIS

*Boot* é o termo utilizado para fazer referência ao processo de inicialização de um computador: do acionamento da máquina até o total carregamento do sistema operacional instalado. Nos terminais LTSP, esse processo também ocorre através da rede, baixando todos os *softwares* necessários diretamente do servidor, logo não é necessário possuir um disco rígido para o funcionamento dos terminais, tão pouco um sistema operacional instalado no terminal. Entretanto todos os servidores devem estar devidamente configurados, provendo os serviços de DHCP, M/TFTP e NFS para o cliente (Figura 1). [13] [14]

Há algumas placas de rede que suportam a tecnologia *Preboot Execution Environment (PXE),* uma especificação que padroniza a comunicação cliente/servidor para iniciar um conjunto de *software* e permitir o arranque do computador através da rede. Por ser uma especificação nativa no *firmware* das placas de rede desenvolvidas pela Intel facilita a configuração dos terminais, requisitando apenas a devida configuração no *Setup* da BIOS do terminal. [14] [15]

Entretanto, caso a placa de rede (NIC) não possua tal especificação, a inicialização do boot pode ser feita através de um *software* *etherboot –* programa de código aberto e de pequeno tamanho, gravado em um chip de boot PROM ou ROM na NIC, um disquete, um CD-ROM ou memória *flash*, cuja função é implementar os recursos do PXE. Geralmente é utilizado para carregar sistemas Linux, porém os protocolos e arquivos de boot possuem formatos genéricos, permitindo carregar outros sistemas operacionais. [16]

O processo de inicialização remoto PXE é baseada no protocolo DHCP (*Dynamic Host Configuration Protocol*). Por sua vez, criado com base no protocolo *Bootstrap*, para resolver limitações específicas, cuja finalidade é permitir que os sistemas obtenham configurações de rede e realizar o boot remoto através da rede. O cliente PXE utilizará o protocolo DHCP através das mesmas etapas de uma solicitação comum, contudo adicionará parâmetros extras. Uma sinopse do processo de *boot* por PXE, pode ser descrito através de nove etapas (Figura 2). [17]

**Passo 1.** Quando um computador conectado à rede é ligado, é enviado uma mensagem *DHCP Discover,* um pacote UDP para porta 67, com a finalidade de localizar servidores disponíveis e prontamente identifica-se sendo habilitado para PXE. No campo de opções do pacote está contido: uma *tag* com um identificador universal único (UUID) – um padrão identificador utilizado na construção de software; a interface de dispositivo de rede universal (UNDI) - uma API para placas de redes que utilizam PXE e uma *tag* para informar a arquitetura do cliente.

**Passo 2.** Quando o servidor DHCP receber o pacote *broadcast*, reservará os dados solicitados e enviará outro pacote *broadcast*, o *DHCP Offer,* um pacote UDP na porta 68 para o endereço de *broadcast* na porta 67, indicando disponibilidade e enviando dados básicos como o endereço IP que o servidor está oferecendo, a máscara de subrede, a duração da concessão e o endereço IP do servidor de DHCP que faz a oferta.

**Passo 3**. Um cliente pode receber ofertas de vários servidores DHCP, mas vai aceitar apenas uma oferta DHCP, através da *DHCP Request,* nela o cliente PXE solicita informações necessárias para inicializar remotamente um computador, como identificador de classe do fornecedor DHCP, o nome do servidor TFTP que hospeda a imagem de inicialização e o nome do arquivo de *boot* para realizar transferência.

**Passo 4.** A próxima mensagem será do servidor *DHCP*, que responde com um ACK, um acrónimo de *acknowledge* ou “reconhecimento” em uma tradução livre, podendo conceder os dados solicitados ou recusá-los, de acordo com a sua configuração, dentre eles a localização de um servidor de *boot*.

**Passo 5.** Após receber um IP válido, e com o endereçamento do servidor de inicialização, o cliente enviará outra mensagem *DHCP Request*, semelhante a *DHCP Discover*, para o servidor de Boot fornecido. No campo das opções estará contido: o IP atribuído a ele pelo servidor DHCP, uma *tag* com a versão do identificador universal único (UUID), uma *tag* identificando interface de dispositivo de rede universal (UNDI), uma *tag* informando a arquitetura do processador e o tipo de servidor de inicialização em um campo de opção do PXE.

**Passo 6.** O servidor de boot, então retornará com outro DHCP *Ack*, contendo o nome do arquivo de inicialização, os parâmetros de configuração do TFTP ou MTFTP e quaisquer outras opções necessárias para que o programa de *boot* remoto seja executado com sucesso.

**Passo 7.** O cliente faz o *download* do arquivo executável usando TFTP padrão (porta 69) ou MTFTP (porta atribuído no pacote servidor de inicialização ACK). O arquivo baixado e a colocação do código obtido na memória depende da arquitetura da CPU do cliente.

**Passo 8.** O cliente PXE determina se um teste de autenticidade sobre o arquivo baixado é necessária. Se for necessário o teste, o cliente envia outra mensagem *DHCP Request* ao servidor de inicialização solicitando um arquivo com credenciais para o arquivo de inicialização baixado anteriormente, realizando o download das credenciais via TFTP ou MTFTP, e efetuando o teste de autenticidade.

Figura 2. Etapas do Boot PXE. [17]

**Passo 9.** Finalmente, se o teste de autenticidade foi bem sucedido ou não era necessário, o cliente PXE inicia a execução do código baixado, ou seja, o kernel do Linux foi obtido e alocado na memória RAM.

Primeiramente o *kernel* efetua alguns testes para verificar a quantidade de memória RAM disponível e reserva um pouco de memória para algumas estruturas de dados internas Ao criar um *kernel*, é programado quais dispositivos ele deve encontrar e quando executado, ele tenta localizar e executar estes dispositivos. Embora alguns dispositivos não sejam localizados completamente, o *kernel* do Linux faz uma verificação sondando o barramento em busca de novos dispositivos. Os dispositivos que não respondem à sondagem são desabilitados, se algum dispositivo for conectado após este procedimento, o mesmo pode ser carregado ou habilitado durante a execução. [18]

Quando o kernel auto inicializa, ele finaliza as suas tarefas no sistema ao iniciar o primeiro processo do sistema, o *initramfs –* o sistema raiz de arquivos, ponto de partida para o processo de inicialização, que irá completar determinadas tarefas antes que o sistema de arquivos real é carregado. Assim, o *initramfs* precisa conter todos os drivers de dispositivos e ferramentas necessárias para montar o sistema de arquivos principal. Porém por não possuir um disco rígido por conta própria, será criado um sistema de arquivos virtuais em um endereço reservado no servidor, através do protocolo *NFS – Network File System* que permitirá a estação de trabalho realizar acesso remoto e transparente a arquivos, discos e diretórios. [19]

Finalmente, após o núcleo montar a raiz do sistema de arquivos, irá lançar os *scripts* de arranque do sistema, que serão responsáveis por colocar o terminal no modo configurado, neste caso, o modo gráfico. Este servidor de ambientes de trabalho em modo gráfico permite autenticar e executar uma sessão num servidor remoto, utilizando o protocolo *XDMCP- X Display Manager Connection Protocol,* que é utilizado para permitir que os terminais obtenham a tela de *login* do servidor e executem os aplicativos remotamente. E por não utilizar nenhum tipo de encriptação ou compressão, transmitirá os dados de forma mais simples e rápida possível. Caso o utilizador consiga autenticar, será lançado o seu ambiente de trabalho, como se o utilizador estivesse a trabalhar localmente no servidor.

A transmissão de toda informação gráfica do servidor para o terminal através da rede permitirá que computadores com baixo desempenho, utilizem o desempenho do servidor para o restante das operações. Quando há mais de um terminal utilizando o servidor LTSP, os aplicativos são carregados na memória RAM do servidor apenas uma vez, independentemente do número de usuários que utilizarem simultaneamente, pois o sistema carrega o aplicativo uma vez, e depois passa a abrir diferentes seções do mesmo programa e referenciados para cada sessão do usuário, o que faz que o carregamento passe a ser mais rápido e o uso de memória seja otimizado. Ou seja, caso 50 terminais estejam utilizando a suíte *LibreOffice*, geralmente o servidor LTSP necessitará de memória RAM para uma única instância do *LibreOffice*, excluindo a configuração por usuário, que é mínima.

V. CONSIDERAÇÕES FINAIS

Sabe-se que o crescimento, a modernização e as novas tecnologias sugerem aquisição e movimento de capital, ocasionando a efetivação do descarte para substituir o que se apresenta ultrapassado/saturado. Por outro lado, é emergente a reutilização e exploração do parque tecnológico.

A tecnologia LTSP apresenta-se uma ótima alternativa para reaproveitamento de hardware, afinal utiliza hardware obsoleto. Além de ser um projeto de baixo custo por utilizar software livre, livre de vírus, de manutenção mais rápida, pois reduz o tempo de configuração dos microcomputadores, e por fim, insere uma nova consciência nas pessoas sobre o processo de reaproveitamento de tecnologia.

Referências

1. MEIRELLES, F.S. Pesquisa Anual CIA. FGV-EAESP. 25ª edição. 2014.
2. Cartilha E-Lixo no Brasil. TI Rio-Riosoft. 2014. Disponível na internet via http://www.tirio.org.br/media/cartilha-e-lixo-ti-rio-versao-site.pdf
3. THIAGO, Heitor Blum S. Exclusão Digital e o mercado de trabalho. 2014.
4. BEZERRA, Arthur Lopes. Thin Clients com LTSP – Linux Terminal Server Project: Uma Alternativa de Sucesso em Ambientes Corporativos. Fortaleza, 2009.
5. LTSP. Keys Benefits of LTSP. Disponível na internet via <http://www.ltsp.org/benefits>
6. KUROSE, James F. ROSS, W. Keith. Redes de Computadores e a Internet: uma abordagem top-down. – 3. Ed. – São Paulo: Pearson Addison Wesley, 2006.
7. TORRES, Gabriel. Redes de Computadores Curso Completo. São Paulo, Axel Books, 2011.
8. FOROUZAN, Behrouz A. A Comunicação de Dados e Redes de Computadores – 6. Ed. Mcgraw Hill – Artmed, 2006.
9. BATTISTI, Júlio. Tutorial de TCP/IP – Introdução ao DHCP. Disponível em < <http://juliobattisti.com.br/artigos/windows/tcpip_p9.asp>>. Acessado em 17 de março de 2015.
10. SWINGLE, Bill. RHODES, Tom. NFS. Disponível em <<http://www.openit.com.br/freebsd-hb/network-nfs.html>>. Acessado em 17 de março de 2015.
11. GAVIDIA, Jorge J. Zavaleta. Finalidade Trivial File Transfer Protocol. Disponível em <http://penta2.ufrgs.br/rc952/trab1/tftp.html>. Acessado em 16 de abril de 2015.
12. MORIMOTO, Carlos E. XDMCP. Disponível em < <http://www.hardware.com.br/termos/xdmcp>>. Acessado em 17 de março de 2015.
13. LTSP – Página oficial do projeto. Disponível em <<http://wiki.ltsp.org/wiki/Concepts>>. Acessado em 16 de março de 2015.
14. VALICHESKI, Roberto Carlos. LTSP - Reaproveitamento de Hardware em laboratório de Informática. Disponível em <http://www.ginux.ufla.br/node/285>. Acessado em 16 de março de 2015.
15. MORIMOTO, Carlos E. Guia: Terminais Leves com LTSP 4.2. Disponível em < <http://www.hardware.com.br/guias/terminais-leves-ltsp/>>. Acessado em 16 de março de 2015.
16. ETHERBOOT PROJECT – Página oficial do projeto. Disponível em <<http://etherboot.org/>>. Acessado em 16 de março de 2015.
17. Preboot Execution Environment (PXE) Specification. Disponível em <<http://download.intel.com/design/archives/wfm/downloads/pxespec.pdf>>. Arquivo capturado em 16 de março de 2015.
18. SALES, Tadeu Araújo. O Processo de Boot no Linux. (Viva ao Linux). Disponível em <<http://www.vivaolinux.com.br/artigo/O-Processo-de-Boot-no-Linux>>. Acessado em 16 de março de 2015.
19. BOER, Adriano. O programa INIT. Disponível em <<http://www.vivaolinux.com.br/artigo/O-programa-INIT/?pagina=1>>. Acessado em 16 de março de 2015.

1. S. A. M. Menezes, Instituto Federal de Sergipe (IFS), Lagarto, Sergipe, Brasil, andrade\_sm1972@yahoo.com.br

   M. de O. Neres, Instituto Federal de Sergipe (IFS), Lagarto, Sergipe, Brasil, mikaelneres@gmail.com

   T. B. dos Santos, Instituto Federal de Sergipe (IFS), Salgado, Sergipe, Brasil, [teofilobs@gmail.com](mailto:teofilobs@gmail.com) [↑](#footnote-ref-1)