

LUCAS NASCIMENTO MARTINS CAMARGO DA SILVA

ANÁLISE DE CUSTO DE ESCALABILIDADE EM AMBIENTE LINUX TERMINAL  
SERVER PROJECT (LTSP)

Trabalho de Conclusão de Curso apresentado  
para obtenção do Grau de Bacharel em Ciência  
da Computação da Universidade do Extremo  
Sul Catarinense.

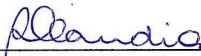
Orientador: Prof. MSc. Rogério Antônio  
Casagrande

CRICIÚMA, JUNHO DE 2010

**LUCAS NASCIMENTO MARTINS CAMARGO DA SILVA**

**Análise de Custo de Escalabilidade em  
Ambiente *Linux Terminal Server Project***

Submetido ao corpo docente do Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.




**Profa. MSc. Ana Claudia Garcia Barbosa**  
Coordenadora do Curso de Ciência da Computação


Banca Examinadora:



**Prof. MSc. Rogério Antônio Casagrande (UNESC)**  
Orientador



**Prof. MSc. Paulo João Martins (UNESC)**  
Membro da banca



**Prof. Esp. Sérgio Coral (UNESC)**  
Membro da banca

Dedico este trabalho as pessoas  
que direta ou indiretamente contribuíram  
no meu crescimento profissional.

“...porque o Espírito Santo vos inspirará naquela hora o que deveis dizer.”

São Lucas 12, 12.

## RESUMO

A utilização de terminais leves mostra-se viável em ambientes comerciais, de produção ou de ensino, pois ela permite a reutilização de computadores que seriam descartados por motivo de obsolescência. O *Linux Terminal Server Project* (LTSP) é um *software* livre que lança mão dessa tecnologia e seu emprego justifica-se quando a redução do impacto econômico precede a necessidade de desempenho. Para o emprego do LTSP em um laboratório de informática é necessário estipular uma métrica que permita definir o dimensionamento de um servidor a ser adquirido. Para obter esse dado foi estudado o comportamento de um servidor conforme o número de estações ligadas simultaneamente. Em um ambiente com dez computadores, foram realizados testes e, posteriormente, analisou-se o servidor LTSP quando o número de clientes aumentou de maneira escalonar de um até dez *hosts*. O resultado aqui obtido funciona como ferramenta para o emprego do LTSP em outros ambientes onde procura-se implementar *thin clients*.

**Palavras-chave:** *Linux Terminal Server Project* (LTSP), Redes de Computadores, Clientes magros (*thin clients*).

## **ABSTRACT**

The use of thin clients appears to be viable in commercial, production, teaching environment, as it enables the reuse of computers that would be discarded due to obsolescence. The Linux Terminal Server Project (LTSP) is a free software that makes use of this technology and its use is justified when reducing the economic impact is above the need for performance. For the use of LTSP in a computer lab is necessary to set a metric that allows the definition of the configuration of a server that will be acquired To obtain this data the performance of a server was observed according to the number of stations connected simultaneously was studied. In a researching room ten computers were tested then the server was analyzed while the number of terminals were increased scaling up from one to ten hosts. The result obtained here serves as a reference source to implement a LTSP in other cases that thin client architecture is sought.

**Keywords: Linux Terminal Server Project (LTSP), Computer Networks, Thin Clients.**

## LISTA DE ILUSTRAÇÕES

Figura 1. Wide Area Network.....	18
Figura 2. Local Area Network.....	19
Figura 3. Topologia em Barra.....	21
Figura 4. Topologia em Estrela.....	21
Figura 5. Topologia em Anel.....	22
Figura 6. Cabo de Par Trançado.....	24
Figura 7. Cabo Coaxial.....	25
Figura 8. Fibra Óptica.....	26
Figura 9. Pilha de Protocolos TCP/IP.....	29
Figura 10. Demultiplexação Baseada em Portas.....	30
Figura 11. Formato dos campos em um datagrama UDP.....	30
Figura 12. Fluxo no modelo TCP/IP.....	31
Figura 13. Disposição da rede LTSP.....	37
Figura 14. Memória principal disponível no primeiro dia.....	42
Figura 15. Memória swap disponível no primeiro dia.....	42
Figura 16. Consumo de memória principal.....	43
Figura 17. Memória principal disponível no segundo dia.....	44
Figura 18. Memória swap disponível no segundo dia.....	44
Figura 19. Inatividade de CPU no primeiro dia de testes.....	45
Figura 20. Inatividade de CPU no segundo dia de testes.....	45
Figura 21. Consumo médio de swap %.....	46

## LISTA DE TABELAS

Tabela 1. Quadro comparativo WAN/LAN.....	20
Tabela 2. Picos de consumo no dia 24 de abril.....	43
Tabela 3. Média do consumo de RAM.....	43
Tabela 4. Picos de consumo do dia 15 de maio – primeira rodada de testes.....	44
Tabela 5. Picos de consumo do dia 15 de maio – segunda rodada de testes.....	45

## LISTA DE ABREVIATURAS E SIGLAS

CPU	Central Processing Unit
CSMA/CD	Carrier Sense Multiple Access/Collision Detection
DARPA	Defense Advanced Research Projects Agency
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
FTP	File Transfer Protocol
GB	Gigabyte
GHZ	Gigahertz
GPL	General Public License
GUI	Graphic User Interface
HD	Hard Disk
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IEEE	Institute Electronic and Electrical Engineers
IP	Internet Protocol
ISO	International Organization for Standardization
LAN	Local Area Network
LTSP	Linux Terminal Server Project
MBPS	Megabits Per Second
MHZ	Mega Hertz
NAT	Network Address Translation
OSI	Open Systems Interconnect
PDU	Protocol Data Units

PSN	Packet Switch Node (router)
PXE	Pre-eXecution Environment
RAM	Random Access Memory
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SO	Sistema Operacional
STP	Shielded Twisted Pair
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
UDP	User Datagrama Protocol
UTP	Unshielded Twisted Pair
WAN	Wide Area Network

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>13</b>
1.1 OBJETIVO GERAL.....	14
1.2 OBJETIVOS ESPECÍFICOS.....	14
1.3 JUSTIFICATIVA.....	14
1.4 ESTRUTURA DO TRABALHO.....	15
<b>2 INTRODUÇÃO A REDES.....</b>	<b>17</b>
2.1 WIDE AREA NETWORK.....	18
2.2 LOCAL AREA NETWORK.....	18
2.3 COMPARATIVO WAN/LAN.....	19
2.4 TOPOLOGIA DE REDE.....	20
<b>2.4.1 Topologia Física.....</b>	<b>20</b>
2.4.1.1 Topologia em Barramento.....	21
2.4.1.2 Topologia em Estrela.....	21
2.4.1.3 Topologia em Anel.....	22
<b>2.4.2 Topologia Lógica.....</b>	<b>22</b>
2.4.2.1 Ethernet.....	22
2.5 MEIOS DE TRANSMISSÃO.....	23
<b>2.5.1 Par Trançado.....</b>	<b>24</b>
<b>2.5.2 Cabo Coaxial.....</b>	<b>24</b>
<b>2.5.3 Fibra óptica.....</b>	<b>25</b>
2.6 ARQUITETURA OSI e TCP/IP.....	26
2.7 ARQUITETURA CLIENTE / SERVIDOR.....	32
<b>3 SISTEMAS OPERACIONAIS.....</b>	<b>33</b>
3.1 LINUX.....	33

3.2 BOOT REMOTO.....	35
<b>4 LINUX TERMINAL SERVER PROJECT.....</b>	<b>37</b>
4.1 METODOLOGIA DOS TESTES.....	38
<b>5 TRABALHOS CORRELATOS.....</b>	<b>40</b>
<b>6 TRABALHO DESENVOLVIDO.....</b>	<b>41</b>
6.1 RESULTADOS OBTIDOS.....	41
<b>7 CONCLUSÃO.....</b>	<b>48</b>
<b>REFERÊNCIAS.....</b>	<b>49</b>

## 1 INTRODUÇÃO

No modelo cliente/servidor, a comunicação ocorre por meio de uma mensagem de solicitação do cliente enviada para o servidor, pedindo para que alguma tarefa seja executada. Em seguida, o servidor executa a tarefa e envia a resposta (TANENBAUM, 1997).

Nesse contexto é que funcionam também os *thin clients*. Os clientes leves são computadores sem disco rígido que recebem, via rede durante a inicialização, o sistema operacional, que é executado remotamente no servidor (AIEX; DEZIDERÁ; KAMINSKI, 2006).

No ano passado, o 28º Grupo de Artilharia de Campanha recebeu uma doação de dez microcomputadores com configuração idêntica da Caixa Econômica Federal com objetivo de implantar um laboratório de informática e promover inclusão digital. O referido laboratório será destinado ao público em geral e a utilização dos computadores será para atividades triviais como navegação na Internet e edição de textos. No entanto os computadores possuem uma configuração de *hardware* defasada, sendo assim, justificável a aquisição de um servidor que dará suporte às máquinas doadas.

Dentre as opções pesquisadas para o reaproveitamento dos equipamentos, uma alternativa foi o *Linux Terminal Server Project*. O LTSP utiliza-se do conceito dos *thin clients*, e, por ser uma arquitetura que permite reaproveitar computadores antigos e que normalmente seriam descartados. Este modelo transcendeu o ambiente onde foi originalmente destinado (escolas e laboratórios de informática), e hoje figura como solução em instituições importantes como o Departamento de Física de Zurich e a Escola Naval pertencente à Armada Nacional Uruguaia.

## 1.1 OBJETIVO GERAL

Estabelecer métricas para determinar a configuração mínima para aquisição de um servidor LTSP.

## 1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desta pesquisa consistem em:

- a) levantar dados bibliográficos;
- b) estudar de redes;
- c) obter dos recursos necessários;
- d) implementar a rede de dados;
- e) instalar e configurar dos clientes e servidor LTSP;
- f) executar testes de saturação no ambiente LTSP;
- g) analisar o comportamento e configuração do servidor.

## 1.3 JUSTIFICATIVA

Segundo o *site* oficial do *Linux Terminal Server Project*, é um recurso de baixo custo que oferece um ambiente com aplicações básicas, utilizando-se *thin clients*.

Aiex, Deziderá e Kaminski (2006) formularam um questionário que ilustra essas aplicações básicas. Respondido por 180 usuários de um laboratório de informática, o resultado da pesquisa possibilitou que se definissem os programas que deveriam ficar disponíveis no servidor. De acordo com as pesquisas, os programas julgados essenciais devem possuir as funcionalidades de edição de textos, descompactador de arquivos, visualizador de

pdf, navegador web e visualizador de vídeos.

Estabelecida a tecnologia a ser empregada e os aplicativos que ficarão a disposição dos usuários no servidor, é determinado o cenário e serão realizados testes para estudo do comportamento do servidor conforme o número de estações ligadas simultaneamente.

Morimoto (2008) sugere que um servidor com 1 GB de memória RAM, dividido entre 20 terminais, executa em geral os aplicativos com um desempenho muito melhor que um desktop com 256 MB usado por um único usuário. Complementa, ainda, que uma sugestão mínima para dar suporte a dez máquinas seria de um servidor Pentium III com 512 MB de memória RAM. Outros *sites* também orientam qual deve ser a razão da quantidade de memória RAM que deve ser investida versus o número de máquinas clientes na expansão de terminais LTSP.

Os dados apresentados acima são antigos e imprecisos. Além disso, não existe nenhum estudo específico no sentido de acrescentar terminais, bombardeando o servidor LTSP com requisições, procurando saturar seu poder de processamento. Por esses motivos não há um denominador comum no que tange a configuração adequada ao servidor que dará suporte as máquinas da inclusão digital.

Para quantificar é que se torna necessário um estudo mais detalhado e que dê valor científico aos dados. O trabalho consistirá em aumentar de maneira escalonar as requisições destinadas ao servidor LTSP num ambiente previamente determinado quando o número de *thin clients* é acrescido de maneira sistemática.

#### 1.4 ESTRUTURA DO TRABALHO

A seguir uma breve descrição do conteúdo contido em cada capítulo deste

trabalho.

Primeiro capítulo apresenta a visão geral do trabalho, definindo o objeto de estudo. São também definidos os objetivos específicos que conduzirão até a resolução do tema proposto.

Para fornecer conhecimento básicos, o segundo capítulo introduz um histórico sobre redes de computadores e seus conceitos, estruturas existentes para sua exploração e meios físicos usados para interligação. Logo a arquitetura cliente/servidor é apresentada no terceiro capítulo.

Aprofundando o assunto, os Capítulos 4 e 5 apresentam o Sistema Operacional Linux, abordando ainda, o funcionamento do *boot* remoto explorado com o *Linux Terminal Server Project*.

As ferramentas e técnicas utilizadas no presente trabalho são oriundas de pesquisas apresentadas no Capítulo 6 – Trabalhos Correlatos.

Para dar fechamento ao trabalho, os resultados das análises estão dispostas no Capítulo 7 – Trabalho Desenvolvido.

## 2 INTRODUÇÃO A REDES

As redes de computadores são parte do cotidiano de muitas pessoas e empresas de todo mundo. Desde as primeiras experiências, na década de 60, até os dias de hoje, as tecnologias de comunicação entre computadores evoluíram substancialmente (TANEMBAUM, 2003).

Uma rede de computadores se define por um conjunto de dispositivos capazes de efetuar qualquer tipo de troca de mensagens através de um meio de transmissão, sendo este por meio de fios de cobre, fibras ópticas, ar, entre outros (SOARES; LEMOS; COLCHER, 1995).

No início da década de 1980, o ambiente de informações das empresas era baseado nos grandes mainframes controlados por especialistas técnicos e programadores. Ao ser solicitado alguma informação, levaria um tempo indefinido, dependendo de sua complexidade (THOMAS, 1997). Os computadores pessoais mudaram essa realidade e com apenas um deles, era possível entrar com as informações, manipulá-las e produzir o documento necessitado, por esse motivo o mercado de computadores pessoais explodiu, e muitas distribuidoras de *hardware* e *software* participaram de uma competição intensa para aproveitar a oportunidade de grandes lucros. A concorrência estimulou ainda mais o desenvolvimento tecnológico e o aumento da capacidade dos computadores pessoais (THOMAS, 1997).

Apesar da velocidade e da capacidade proporcionadas pelo ambiente dos computadores pessoais, este ainda era bastante fraco no elemento mais importante: a comunicação entre as pessoas na empresa (SOARES; LEMOS; COLCHER, 1995). A solução para isto foi conectar os computadores pessoais e interligar a um repositório de informações central compartilhado, este repositório foi denominado servidor.

As redes de computadores podem existir dentro de uma única sala, em todo um prédio, em uma cidade, em um país e no planeta. Existem redes constituídas de redes, e redes que se acessam umas às outras de acordo com a necessidade de acesso às informações. (PISONI, 2003)

## 2.1 WIDE AREA NETWORK

A *Wide Area Network* (WAN), ou redes de longa distância transmitem dados, voz, imagens e vídeos por grandes extensões, podendo compreender um país, um continente ou todo o mundo.

“Para conectar com efetividade vastas distâncias, as WAN’s utilizam-se de redes públicas, redes sob concessão ou alugadas, equipamentos privados de comunicação ou combinações desses para atingir distâncias ilimitadas na superfície do planeta.” (FOROUZAN, 2006, p. 43).

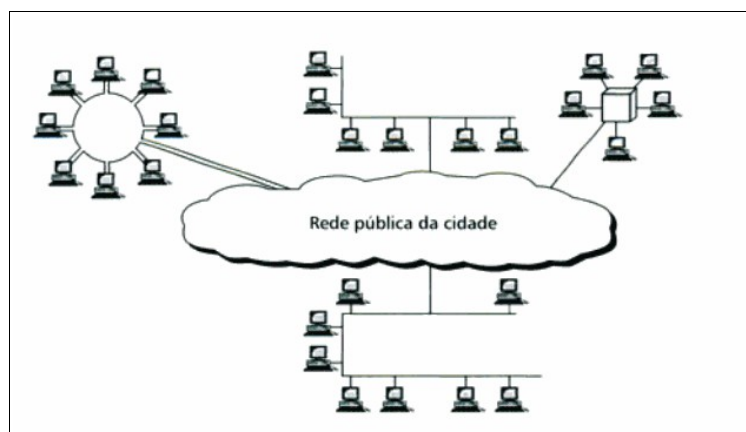


Figura 1. *Wide Area Network*  
Fonte: FOROUZAN, B. A. (2006)

## 2.2 LOCAL AREA NETWORK

De maneira geral uma *Local Area Network* (LAN), é administrada de maneira privada e os dispositivos encontram-se numa mesma área geográfica como uma sala,

escritório, apartamento ou edifício. A LAN pode ser constituída por no mínimo dois computadores e pode se estender a milhares. “Uma LAN depende essencialmente da infraestrutura de uma organização ou de uma empresa e do tipo de tecnologia utilizada. Atualmente o tamanho aceitável para uma LAN está limitada a poucos quilômetros.” (FOROUZAN, 2006, p. 42).

As LAN's são projetadas para permitirem compartilhamento de recursos de dispositivos ligados a ela. Estes dispositivos podem incluir *hardware* (impressoras ou gravadores), como *software* (programas e aplicativos). Como descrito na Figura 2, esta composição de rede pode ser facilmente encontrada em empresas cujos diversos departamentos trabalham interligados entre si (FOROUZAN, 2006).

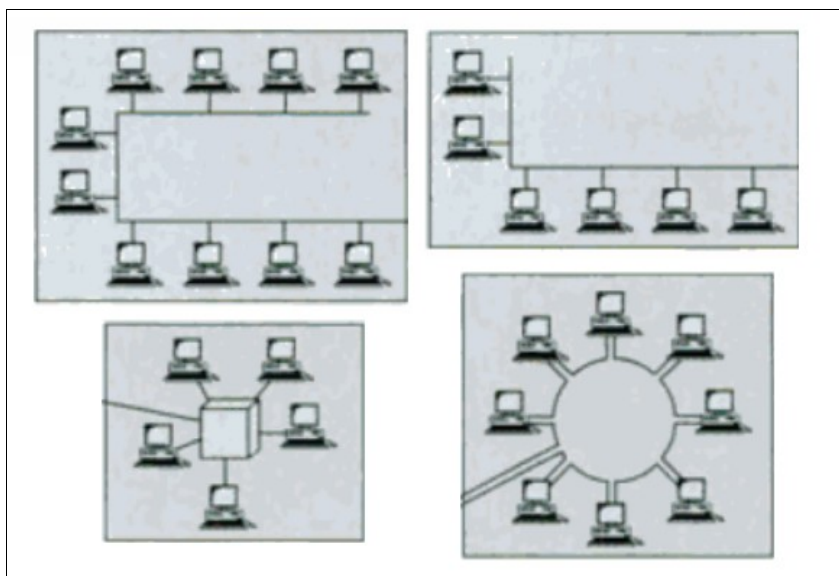


Figura 2. *Local Area Network*  
Fonte: FOROUZAN, B. A. (2006)

### 2.3 COMPARATIVO WAN/LAN

Por definição WAN's e LAN's diferem devido a distância por onde estabelecem comunicação. Em linhas gerais WAN's abrangem maiores distâncias que LAN's. A Tabela 1 traça as nomenclaturas empregadas em relação as distâncias percorridas entre os terminais de rede.

Tabela 1. Quadro comparativo WAN/LAN

<b>Distância entre as CPUs</b>	<b>Localização das CPUs</b>	<b>Nome</b>
100 m	Prédio	Redes Locais (LANs)
1000 m 1 km	Campus	Redes Locais (LANs) Universidade
100.000 m 100 km	País	Rede de longa Distância (WANs), Empresas
1000.000 m 1.000 km	Continente	Rede de longa Distância (WANs), África
10.000,000 m 10.000 km	Planeta	Rede de longa Distância (WANs), A Internet
100.000.000 m 100.000 km	Sistema Terra-Lua	Rede de longa Distância (WANs), Terra e satélites artificiais.

Fonte: SOARES, L.F. ; LEMOS G.; COLCHER S. (1995).

## 2.4 TOPOLOGIA DE REDE

Os sistemas de comunicação eletrônicos trocam mensagens quando estão interligadas a algum meio físico. A distribuição e modo como são conectados os componentes dessa rede é definido como topologia de rede. Variadas topologias influenciam no desempenho dos dados que por ali trafegam, especialmente quando se tratando de velocidade (TITTEL, 2002).

### 2.4.1 Topologia Física

Descreve-se como o cabeamento e componentes de *hardware* do meio físico serão dispostos na rede, como é feita a distribuição da mídia de conexão (cabeamento de cobre, fibra óptica, *wireless*, entre outros). São exemplos: a topologia em barramento, em estrela e em anel.(TANEMBAUM, 1997)

#### 2.4.1.1 Topologia em Barramento

Em uma rede onde se emprega uma topologia em barramento, tem-se um meio de transmissão onde todos os nós se conectam, podendo cada nó capturar as mensagens que estão trafegando na rede. Com uma topologia muito parecida com o anel, se diferencia desta no fato de se um nó parar de funcionar o circuito poderá continuar funcionando. Na Figura 3 pode-se visualizar a topologia em forma de barra (SOARES; LEMOS; COLCHER, 1995).

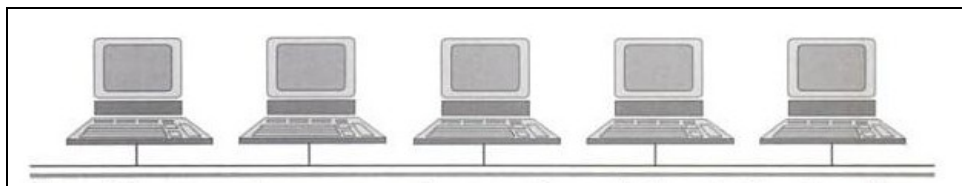


Figura 3. Topologia em Barra

Fonte: GALLO, M. A.; HANCOCK, W. M. (2003)

#### 2.4.1.2 Topologia em Estrela

Quando os computadores estão interligados em estrela, esta rede possui um nó central, chamado de mestre, onde todas as mensagens que trafegam na rede devem passar. Este nó central pode executar diversas funções como chaveamento e processamento. Nada impede que haja comunicações simultâneas, desde que as estações envolvidas sejam diferentes (SOARES; LEMOS; COLCHER, 1995).

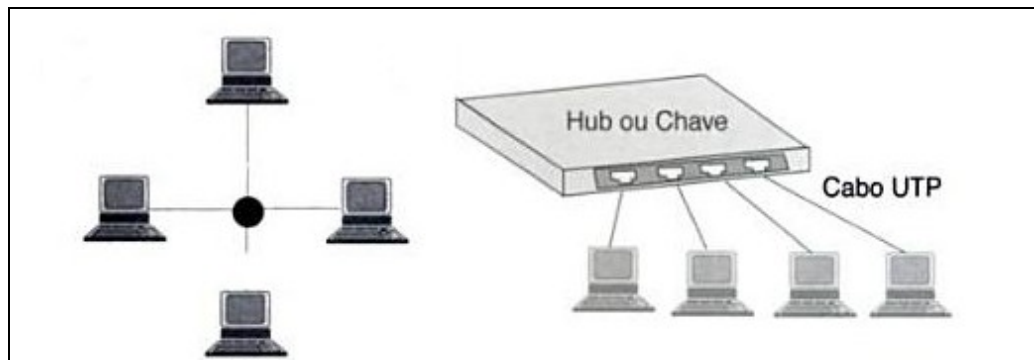


Figura 4. Topologia em Estrela

Fonte: GALLO, M. A.; HANCOCK, W. M. (2003)

### 2.4.1.3 Topologia em Anel

A topologia em anel, representada pela Figura 1, requer que cada nó seja capaz de verificar as mensagens que chegam. Caso estas mensagens não sejam destinadas a ele, deverão ser passadas adiante. Na topologia em anel existe um circuito fechado entre as estações, sendo cada uma delas um repetidor, para passar adiante a mensagem. Porém se um nó destes para de transmitir informação, todo o circuito de comunicação fica comprometido (SOARES; LEMOS; COLCHER, 1995).

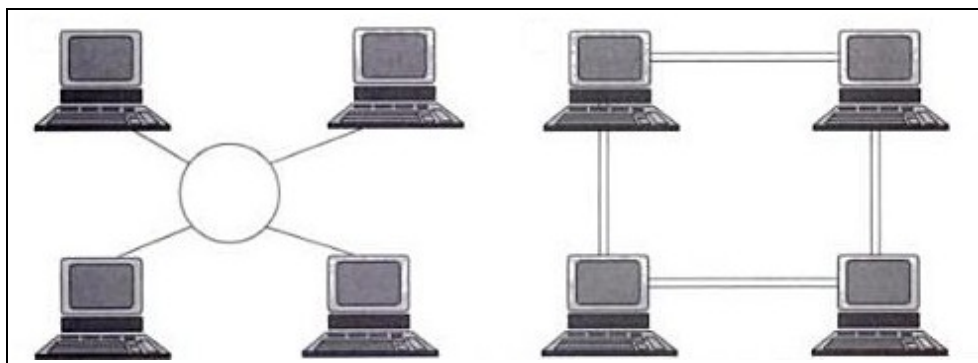


Figura 5. Topologia em Anel  
Fonte: GALLO, M. A.; HANCOCK, W. M. (2003)

### 2.4.2 Topologia Lógica

Descreve a maneira como a rede transmite informações de um equipamento para outro, o formato dos dados, o método de transferência, dentre outros. É a forma como os protocolos operam no meio físico. O exemplo mais conhecido e mais usado é a topologia lógica Ethernet (TANENBAUM, 1997).

#### 2.4.2.1 Ethernet

No início, a transmissão de dados por uma rede de computadores não era muito eficiente. Os computadores ligados a uma rede não conseguiam evitar o envio simultâneo de

sinais elétricos pelo fio que os ligava (THOMAS, 1997). A Ethernet, inventada em 1973 por Bob Metcalfe (fundador da 3Com), foi uma maneira de contornar as limitações das primeiras redes. Baseava-se em um padrão do IEEE (*Institute Electronic and Electrical Engineers*), chamado 802.3 CSMA/CD, e oferecia maneiras de conciliar a transmissão simultânea por um único fio (HAYDEN, 1999).

A base da Ethernet é o *Carrier Sense Multiple Access/Collision Detection* (CSMA/CD). Em uma rede Ethernet todos os computadores compartilham um único segmento de rede, chamado domínio de colisão. Quando dois computadores transmitirem pacotes ao mesmo tempo, ocorre uma condição chamada de colisão. Quanto maior for o domínio de colisão, mais provável será a ocorrência de colisões, por essa razão deve-se manter o número de computadores de um segmento o mais baixo possível (HAYDEN, 1999).

Em CSMA/CD, cada computador aguarda por um momento em que o fio está “quieto”. Quando isso ocorre, o computador que tiver dados para transmitir os enviará pelo fio da rede. Se um segundo computador tiver de transmitir ao mesmo tempo que o primeiro, ambos sentirão a presença um do outro e ambos interromperão o envio de dados, aguardando assim por um número aleatório de milissegundos e transmitir novamente. Na topologia Ethernet somente um computador pode transmitir dados de cada vez (HAYDEN, 1999).

## 2.5 MEIOS DE TRANSMISSÃO

É importante entender que os dispositivos de comunicação utilizam sinais para representar dados. Todos estes sinais são transmitidos por meio de pacotes de energia eletromagnética propagando-se através de um meio de transmissão. Se fossemos discutir onde se encontra o meio de transmissão no modelo de camadas OSI, é como se estivesse abaixo da camada física (FOROUZAN, 2006).

Por exemplo, a corrente elétrica pode ser usada para transferir dados através de um fio, ou as ondas de rádio podem ser usadas para transferir dados através do ar. Uma vez que os dispositivos de hardware conectados a um computador executam a codificação e decodificação dos dados, os programadores e os usuários não precisam conhecer os detalhes da transmissão. (COMER, 2007, P. 51).

### 2.5.1 Par Trançado

O cabo de par trançado é formado por pelo menos dois pares de fios condutores entrelaçados separados por material isolante. Cada par é composto por um positivo e um negativo, isso é o que garante a proteção contra interferências eletromagnéticas. Os cabos de par trançado também são classificados de acordo com a existência ou não de blindagem.

Nos cabos onde o condutor é isolado por um dielétrico como polietileno são chamados de *Unshielded Twisted Pair* (UTP). Os cabos *Shielded Twisted Pair* (STP), cada par de condutores é embrulhado em uma blindagem trançada de cobre ou latão. As blindagens metálica e trançada tornam os cabos menos suscetíveis a ruído e interferência elétrica (GALLO; HANCOCK, 2003).

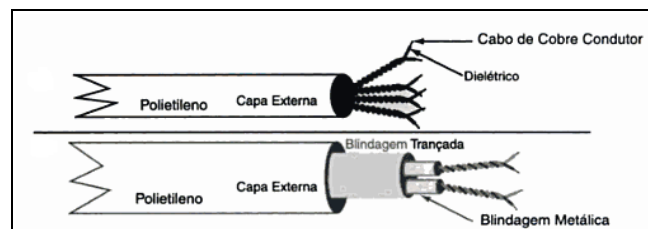


Figura 6. Cabo de Par Trançado  
Fonte: GALLO, M. A.; HANCOCK, W. M. (2003)

### 2.5.2 Cabo Coaxial

O cabo coaxial é formado por dois condutores separados e envoltos por um material isolante. O primeiro condutor, normalmente o cobre, é mais rígido e está envolto pelo segundo condutor, este em forma de malha. Este segundo condutor, além de ajudar na transmissão é também responsável por proteger o primeiro condutor contra interferências magnéticas. O cabo coaxial pode ser classificado de duas formas dependendo do material do

condutor em malha (FOROUZAN, 2003).

O diâmetro interno do cabo coaxial pode variar de 0,96 cm a 1,04cm dependendo do fabricante. Cabo coaxial grosso recebe a designação RG-8 com impedância 50 ohms. Coaxial fino tem diâmetro variando entre 0,448cm e 0,5cm e recebe a designação RG-58 também com impedância de 50 ohms. Embora tenha sido empregada na década de 80 o cabo coaxial perdeu espaço para o UTP e dificilmente é instalado hoje em dia (GALLO;HANKOK, 2003).

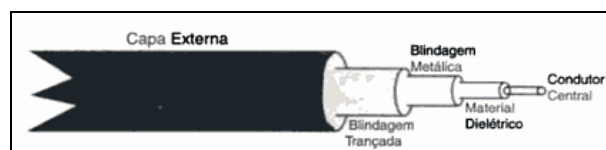


Figura 7. Cabo Coaxial

Fonte: GALLO, M. A.; HANCOCK, W. M. (2003)

### 2.5.3 Fibra óptica

Os cabos de fibra óptica são filamentos de vidro com capacidade de transmitir sinais digitais sob a forma de sinais luminosos. Tal filamento pode apresentar diâmetros variáveis, dependendo da aplicação, indo desde diâmetros ínfimos, da ordem de micrômetros (mais finos que um fio de cabelo) até vários milímetros. Devido a essa característica, são cabos que conseguem ter uma velocidade superiores aos cabos elétricos. Também torna seu uso desejável quando existe a necessidade de transmitir dados a grandes distâncias. Outra característica interessante destes tipos de cabos é que eles não sofrem interferência de campos eletromagnéticos. São cabos com custo mais alto, e com certa dificuldade de manuseio. Entretanto, seu uso vem se disseminando cada vez mais, com a necessidade cada vez maior de velocidades mais altas. Recebem duas designações de acordo com o diâmetro do núcleo. O monomodo que possui diâmetro de núcleo geralmente de sete micrômetros. A outra designação é o cabo multimodo, neste o diâmetro do núcleo varia entre cinquenta e cem micrômetros (FOROUZAN, 2003).

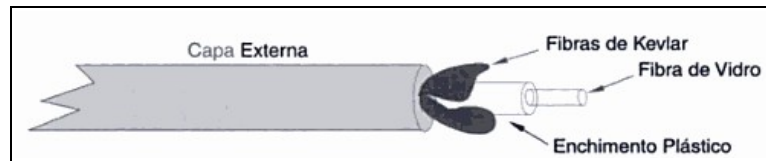


Figura 8. Fibra Óptica  
 Fonte: GALLO, M. A.; HANCOCK, W. M. (2003)

## 2.6 ARQUITETURA OSI e TCP/IP

A arquitetura *Open Systems Interconnection* (OSI), foi lançada pela *International Organization for Standardization* (ISO), e escolhida para ser padrão de comunicação entre os diversos sistemas computadorizados, este padrão acabou sendo amplamente empregado no mercado (FREER, 1996, tradução nossa).

Esta arquitetura é composta por um conjunto de protocolos sendo que os dois principais são o *Transmission Control Protocol* (TCP) e o *Internet Protocol* (IP) de onde se originou o nome TCP/IP, o protocolo utilizado pela Internet (MURHAMMER et al, 2000).

A arquitetura TCP/IP é largamente utilizada para interconexão e inter-operação de sistemas computacionais heterogêneos. Este conjunto de protocolos foi desenvolvido para permitir aos computadores compartilharem recursos numa rede. Sendo assim, esta arquitetura define um conjunto de padrões específicos para realizar a comunicação entre computadores e também convenções para interconectar redes e rotear o tráfego.

Denominado oficialmente como Pilha de Protocolos de interligação em redes TCP/IP, essa pilha pode ser utilizada para comunicação em qualquer conjunto de redes interligadas. Os protocolos TCP e IP fornecem as regras para a comunicação, eles contem os detalhes e formatos de mensagens, descrevem e especificam como o computador trata o recebimento e envio dessas mensagens (COMER, 1999).

O protocolo TCP é orientado a conexão e foi projetado especificamente para proporcionar um fluxo de dados ponto a ponto, ou seja, um transporte confiável de dados entre dois *hosts* (TANENBAUM, 1997). Com o uso de retransmissão o protocolo TCP

funciona corretamente mesmo que os pacotes sofram retardos, sejam duplicados, perdidos, entregues fora de ordem ou com dados danificados ou truncados. Esta é a confirmação de que o pacote foi realmente entregue e sem afetar sua integridade. O TCP pode ser caracterizado pelas possibilidades que oferece aos aplicativos que o usam (MURHAMMER et al, 2000):

- a) transferência em fluxo (*stream*) de dados: certifica-se de que todos os dados passados ao TCP foram realmente transmitidos ao destino;
- b) confiabilidade: o TCP indica um número sequencial para cada segmento transmitido e espera um reconhecimento positivo do TCP receptor. Este por sua vez usa os números sequenciais para reorganizar os segmentos quando os mesmos chegam fora de ordem, e elimina os segmentos duplicados;
- c) controle de fluxo: o TCP receptor indica ao remetente o número de *bytes* adicionais que ele pode receber além do último segmento para não causar sobrecarga;
- d) multiplexação: é alcançado por meio de portas;
- e) conexões lógicas: é o conjunto de informações que o TCP inicializa e mantém como status para cada fluxo de dados;
- f) *full-duplex*: são fluxos simultâneos de dados em ambas as direções. O IP foi projetado para permitir a interconexão de redes que utilizam a tecnologia de comutação de pacotes e é um serviço não orientado a conexão.

O protocolo IP tecnicamente é um serviço definido como um sistema de transmissão sem conexão e não confiável. Este protocolo oferece três definições importantes. A primeira, o IP define a unidade básica de transferência de dados. A segunda, o *software* IP desempenha a função de roteamento. E a terceira, o IP inclui um conjunto de regras que concentram a ideia de entrega não confiável de pacotes (COMER, 1999).

O IP é o protocolo que oculta a rede física, criando uma visão virtual da rede. É

um protocolo de entrega de pacotes não confiável, o que significa que esses pacotes enviados pelo IP podem ser perdidos, mas o IP não se responsabiliza por isso quem trata dessas situações são os protocolos de camada mais alta (MURHAMMER et al, 2000).

O desenvolvimento do protocolo IP foi voltado para condução de pacotes através da rede de dados analisando-se o melhor percurso. Apesar desta característica ele não contempla mecanismo que garantam a qualidade de serviço. Existem outros protocolos que buscam atender essas necessidades:

- a) *Internet Protocol (IP)*: aceita dados segmentados, denominado *Protocol Data Units (PDUs)*, por meio de um *host* e os envia pela Internet até seu destino. Porém deve-se lembrar que por ser um sistema não confiável alguns PDU's não chegam a seu destino;
- b) *Transmission Control Protocol (TCP)*: oferece um sistema de transporte que garante a entrega sem perdas, erros ou duplicação, e ainda reorganiza os pacotes de acordo com a ordem enviada;
- c) *User Datagram Protocol (UDP)*: atua no mesmo nível que o TCP, oferecendo serviço para troca de mensagens. Exclui recursos de confiabilidade do TCP, por isso é mais rápido e é apropriado para aplicações orientadas para transações;
- d) *File Transfer Protocol (FTP)*: é projetado para transferir arquivos ou parte deles de um sistema para outro. Transfere dados pelo TCP por meio da interface do sistema operacional. Esses arquivos podem utilizar um conjunto de caracteres ASCII, ou incluir opções de compressão de dados e mecanismos de identificação de senhas para controle de acesso;
- e) *Simple Mail Transfer Protocol (SMTP)*: não oferece a interface gráfica, mas oferece o mecanismo para correio eletrônico de rede;
- f) *Telnet*: é o protocolo padrão TCP/IP para serviço de terminal remoto. Permite a

conexão de terminais e o controle de aplicações operando em host remoto;

g) *Simple Network Management Protocol* (SNMP): é o protocolo padrão utilizado para monitorar hosts, roteadores e as redes às quais eles se conectam. O SNMP foi projetado para atuar sobre o protocolo UDP;

h) *HyperText Transfer Protocol* (HTTP): é o protocolo de transferência de hipertexto que define como os programas de navegação e servidores Internet devem interagir na transferência de arquivos.

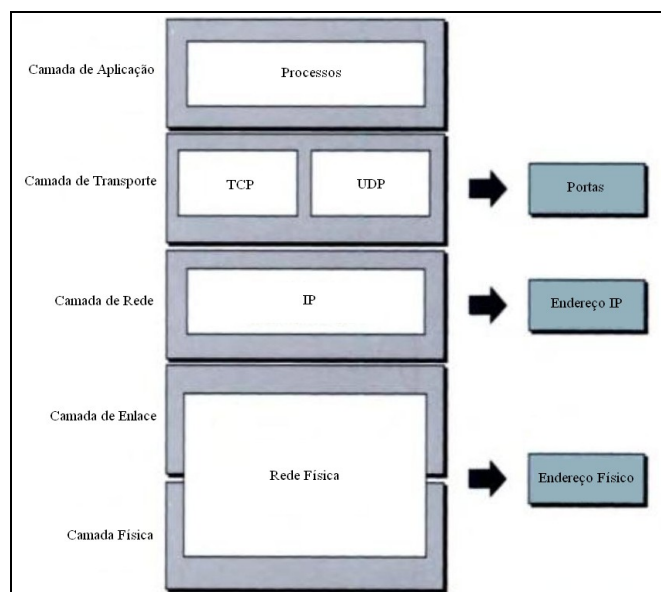


Figura 9. Pilha de Protocolos TCP/IP  
Fonte: FOROUZAN, B. A. (2003)

Como pode se observar na Figura 9, assim como o TCP, o outro protocolo da camada de transporte é o UDP, um protocolo de envio de dados não orientado a conexão. Este protocolo tem como característica a ausência de controle de fluxo e nem de manutenção de mensagens enviadas. Desta maneira, os controles devem ser realizados em nível de aplicação, o que pode se tornar oneroso ao desenvolvedor de *software*. Mesmo não contendo controle de dados confiável o UDP é vantajoso quando a velocidade de transmissão é mais importante que integridade. Seu uso está associado aos protocolos DNS, SNMP e RIP (SCRIMGER et al, 2002).

O protocolo UDP por suas características constitui um canal direto de

comunicação com a camada de rede, agindo como multiplexador e demultiplexador para a camada inter redes. Por não estabelecer conexão e não ficar associado a um processo no *host* destino, o UDP associa um processo a uma determinada porta. O receptor ao identificar a porta a qual o pacote é endereçado encaminha ao aplicativo a que lhe é de direito (MURHAMMER et al, 2000). A Figura a seguir ilustra o direcionamento destas portas.

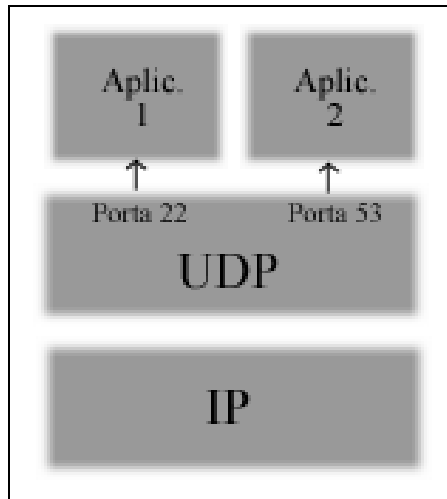


Figura 10. Demultiplexação Baseada em Portas  
Fonte: FOROUZAN, B. A. (2003)

Scrimger (2002) define que o protocolo UDP trabalha com portas que orientam adequadamente o tráfego de informações a cada aplicação de nível superior. Munido do conceito de portas, o UDP torna-se útil às aplicações que necessitam velocidade. Ao observar a Figura 11 torna-se intuitivo entender a simplicidade do cabeçalho dos datagramas UDP ocasionando um baixo *overhead*.

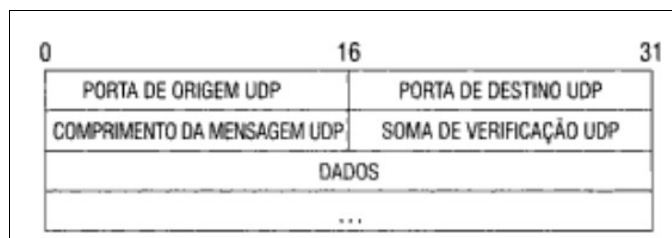


Figura 11. Formato dos campos em um datagrama UDP  
Fonte: SCRIMGER, R. et al (2002)

De acordo com Murhammer et al (2000) a pilha de protocolos TCP/IP é modelada em quatro camadas (Figura 12):

- a) camada de aplicação: é fornecida pelo programa que utiliza o TCP/IP para comunicação. Uma aplicação é um processo de usuário que coopera com outro processo no mesmo servidor ou em outro diferente;
- b) camada de transporte: fornece a transferência de dados de uma ponta a outra. Pode suportar aplicações múltiplas simultaneamente. Esta camada é responsável pelo intercambio de informações confiável. O principal protocolo desta camada é o TCP que é orientado a conexão, porém, o UDP também atua na camada de transporte e oferece um serviço não orientado a conexão;
- c) camada de inter-redes: conhecida também como camada de Internet ou camada de rede, fornece a imagem de rede virtual. O IP é o protocolo mais importante desta camada, é sem conexão e não oferece confiabilidade que são fornecidas pelas camadas mais altas;
- d) camada de interface de rede: também chamada de camada de enlace é a interface com o verdadeiro *hardware* de rede. Pode ou não oferecer entrega confiável, podendo ser orientada para pacotes ou fluxo.

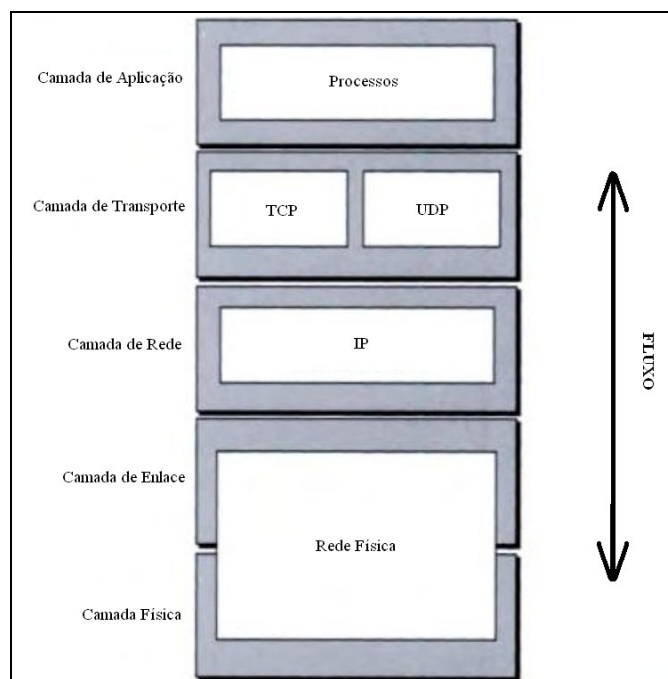


Figura 12. Fluxo no modelo TCP/IP  
 Fonte: FOROUZAN, B. A. (2003)

## 2.7 ARQUITETURA CLIENTE / SERVIDOR

O termo servidor aplica-se a qualquer programa que ofereça um serviço possível de ser alcançado através de uma rede. Um servidor aceita uma solicitação efetivada da rede, executa seu serviço e retorna o resultado ao solicitante. Para os programadores mais simples, cada solicitação chega em um datagrama IP único e o servidor retorna uma resposta em outro datagrama (COMER, 1999).

Um programa que estiver em execução torna-se um cliente quando envia uma solicitação ao servidor e espera a resposta. Como o modelo cliente/servidor é uma extensão conveniente e natural do processo de comunicação interno em uma única máquina, é simples criar programas que usem o modelo para interagir.

Os servidores podem executar tarefas simples e complexas. Por exemplo, um servidor de hora do dia simplesmente retorna a hora corrente sempre que um cliente enviar um pacote ao servidor. Um servidor de arquivo recebe solicitações para executar operações que armazenam ou recuperam dados de um arquivo. O servidor executa a operação e retorna o resultado.

Geralmente, os servidores são implementados como programas aplicativos. A vantagem de implementar os servidores como programas aplicativos é que eles podem ser executados em qualquer sistema de computação que suporte a comunicação TCP/IP. Portanto, o servidor de determinado serviço pode ser executado em um sistema de compartilhamento, juntamente com outros programas, ou pode ser executado em um computador pessoal. Vários servidores podem oferecer o mesmo serviço e podem executar na mesma máquina ou em várias máquinas. De fato, os gerenciadores geralmente duplicam as cópias de determinado servidor em máquinas fisicamente independentes para aumentar a confiabilidade ou melhorar o desempenho (COMER, 1999).

### 3 SISTEMAS OPERACIONAIS

Os sistemas operacionais possuem características de estarem situados como primeiro “elo” com o *hardware*. Inicialmente devemos ter em mente a ideia de operacionalizar o potencial que é oferecido pelo equipamento (POLLONI, 2003).

Percebe-se, assim, ao menos em teoria, a existência de tantos sistemas operacionais quantos forem os tipos de equipamentos a serem “operacionalizados”. Em outras palavras, existem diversos tipos de sistemas operacionais para que se possa atender aos diversos tipos de *hardware*. Dessa forma, vamos verificar como são classificados os sistemas operacionais (POLLONI, 2003).

O sistema operacional basicamente é que gerencia todo o *hardware* e interfaceia o sistema com o programa que utilizamos. É um *software*, mas uma categoria especial diferente dos aplicativos, e, como estes, é também composto de vários arquivos para seu correto funcionamento (arquivos executáveis diversos e drives lógicos, dentre outros).

Vários são os sistemas operacionais, mas podemos citar como exemplo o DOS, o OS/2, o Windows NT, Windows 95, 98, ME, XP e LINUX (POLLONI, 2003).

#### 3.1 LINUX

O Linux é um Sistema Operacional (SO) de livre distribuição baseado no sistema operacional UNIX.

Este SO foi originalmente desenvolvido na década de 1970, nos laboratórios Bell, como um sistema multitarefa para microcomputadores e computadores de grande porte. As primeiras versões do UNIX tiveram o seu código licenciando para diversas universidades e centros de pesquisa, onde gerou muitos desenvolvedores assíduos. A partir do momento em

que o código do UNIX não pode mais ser utilizado livremente, alguns programadores partiram para implementações próprias do UNIX, como é o caso do Linux (PITANGA, 2008).

O Linux foi criado por um jovem estudante de Ciência da Computação da Universidade de Helsinque, na Finlândia, chamado Linus Torvalds. Este sistema, foi concebido como alternativa aos caros sistemas UNIX. Seu desenvolvimento contou com a colaboração e ajuda de muitos programadores e especialistas em UNIX por meio da Internet. Inspirado no SO Minix um sistema UNIX de pequeno porte desenvolvido por Andrew Tanenbaum (1986), o linux foi projetado inicialmente para máquinas de pequeno porte que explorasse as novas funcionalidades disponíveis nos processadores 80386, como interface de modo protegido. Facilidades estas que não eram exploradas pelo Minix (PITANGA, 2008).

O criador Linus Torvalds detém os direitos autorais do *kernel* do Linux e o colocou sob *General Public License* de GNU. É um SO livre e gratuito possível de copiar, alterar e distribuir livremente, na condição de deixar o código fonte das implementações feitas livres para consultas (PISONI, 2003).

O Linux é um sistema preemptivo e multitarefa, ou seja, pode ter mais de um programa executando simultaneamente, cuidando de alternar entre os vários processos, sem que os mesmos tenham conhecimento do fato. Isso é o contrário dos sistemas operacionais multitarefa cooperativos, como o *Windows 3.1*, nesse sistema cada aplicativo é responsável por ceder tempo do processador, se um aplicativo nunca cede, outros aplicativos não receberão nenhum tempo de processador (TOBLER, 2001, tradução nossa).

Além disso, o Linux tem a característica de ser multiusuário, ou seja, ele permite que diversos usuários estejam conectados a uma máquina e utilizem seus serviços. Esse conceito é que possibilita a existência dos *Virtual Terminals*. Estes Terminais Virtuais permitem a abertura de mais de uma tela de ambiente de trabalho simultaneamente (TOBLER, 2001, tradução nossa).

Suporta também o conceito de memória virtual, utilizando uma parte dedicada de uma unidade de disco da máquina, chamado partição de troca (*swap*), trocando páginas entre a memória e o espaço de troca conforme necessário. Para o Linux e quaisquer aplicativos em execução, a máquina parece ter mais memória física do que realmente existe. É claro que existe um custo para utilização da SWAP, o custo do tempo de leitura/escrita em disco rígido. (TOBLER, 2001, tradução nossa).

A Interface Gráfica com o Usuário (GUI) localizada em um sistema UNIX é conhecida como *X Windows*. O Linux fornece um clone da GUI *X Windows* que é chamada *Xfree86*, essa interface gráfica é em janelas. Um gerenciador de janelas é responsável pelo gerenciamento e exibição gráfica, como a colocação e redimensionamento de janelas e aparência global da exibição gráfica. Podemos citar como exemplo de gerenciadores de janelas desde o simples TWM até o elegante KDE (TOBLER, 2001, tradução nossa).

### 3.2 *BOOT* REMOTO

O Linux é plenamente capaz de operar neste modelo, tanto como servidor, quanto como o sistema operacional a ser executado nos computadores sem disco. O *boot* remoto é a técnica necessária para permitir que um *thin client* faça todo o seu processo de inicialização via rede, sem a necessidade de um disco de *boot* local. Entretanto, para que este processo aconteça, é necessário que sua rede conte com os servidores dos protocolos TFTP, DHCP e NFS (SANTOS; CAMPOS, 2009).

De acordo com Balneaves *et al* (2009), o processo de *boot* remoto ocorre da seguinte forma:

- a) o *thin client* inicia o *boot* usando de um protocolo chamado *Pre-eXecution Environment* (PXE);

- b) PXE requisita um endereço IP a um servidor DHCP;
- c) o servidor DHCP informa os parâmetros adicionais ao *thin client* e este baixa um sistema de arquivos Linux via protocolo TFTP e armazena em memória RAM;
- d) o *thin client* efetua um *boot* baseado na imagem Linux recebida anteriormente e inicia a detecção de *hardware*;
- e) o *thin client* após detectar os componentes locais, conecta-se a sessão gráfica do servidor LTSP.

A partir daqui, todas as operações, tais como autenticação de usuário e senha, execução de aplicações, e a visualização *sites* são rodadas no servidor LTSP. O servidor LTSP transfere todos os gráficos e informações para o *thin client* através da rede local (BALNEAVES et al, 2009).

#### 4 LINUX TERMINAL SERVER PROJECT

Uma das principais tecnologias incluídas no GNU / Linux é o *Linux Terminal Server Project* (LTSP), que lhe permite inicializar *thin clients* partir de um servidor LTSP. Ele reduz os custos de *hardware*, permitindo o uso de máquinas com poder de processamento intermediário como *thin clients*. Reduz também a sobrecarga na administração de redes, tendo apenas que instalar e manter o *software* no servidor. Quando uma estação de trabalho falhar, ele pode simplesmente ser substituído, sem perda de dados ou reinstalação do sistema operacional.

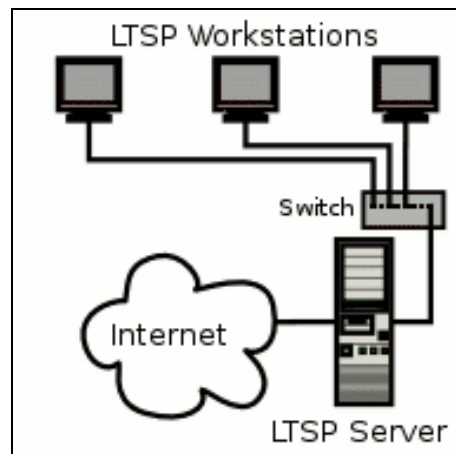


Figura 13. Disposição da rede LTSP  
Fonte: BALNEAVES, S. et al (2009)

Computação *thin client* tem sido tema de discussão há um longo tempo na arquitetura UNIX. Embora a implementação tenha evoluído um pouco, o conceito permanece o mesmo (BALNEAVES, 2009):

- a) o *thin client* cuida apenas das funções básicas, como monitor, teclado, mouse e som;
- b) o servidor faz trabalho de processamento. Todos os aplicativos são executados no servidor, e eles são exibidos no *thin client*.

Devido ao número limitado de tarefas para gerenciar, os *thin clients* não precisam conter componentes de alto custo, tornando-os baratos. Os componentes móveis (*hard disk*),

de um *thin client* são suprimidos, conseqüentemente a manutenção é quase nula. Outro ponto é que a ausência de *hard disk* (HD) no terminal representa segurança, se um *thin client* for roubado não há risco que os dados cheguem a destinos indesejáveis.

O servidor LTSP executa todas as aplicações e contém todos os dados. Toda a manutenção regular como *updates* ocorrem no servidor. O número de *thin clients* que um servidor pode suportar é proporcional ao seu poder de processamento. Como o uso dos recursos dentro de um sistema operacional Linux é eficiente, ele permite a sustentação de um número surpreendente de *thin clients*.

Em uma arquitetura LTSP, a estabilidade do servidor também deve ser analisada. Dependendo do grau de necessidade e disponibilidade de recursos, pode-se lançar mão de fontes de alimentação redundantes e *hard disks* para uso simultâneo em RAID, tudo para garantia de alta disponibilidade do servidor (BALNEAVES et al, 2009).

#### 4.1 METODOLOGIA DOS TESTES

Serão utilizadas as dependências dos laboratórios de informática da UNESC, assim como usuários voluntários a contribuir com o experimento. Os testes serão realizados em uma rede de computadores seguindo as normas técnicas para uma rede padrão ethernet tipo estrela. Para a realização deste projeto será utilizado o Ubuntu 9.10 (versão de kernel 2.6.31-20) em um servidor juntamente com o LTSP 5.0, que segundo TALAUI (2009), é a versão mais recente e que apresenta melhor desempenho. O servidor será configurado com os serviços nativos do LTSP por ele conter os programas básicos para navegação na Internet, execução de vídeos, editor de textos, imagens, entre outros.

Neste ambiente serão realizados uma sequência de comandos pré-estabelecidos nos terminais. A maneira escalonar que será adotada é de, inicialmente, inicializar com um

*thin client*, realizar a sequência de comandos pré-estabelecidos e após isso, desligar o *thin client*. Repetir-se-á esse processo acrescentado-se um terminal de cada vez, até chegar num total de dez *thin clients*.

O roteiro de comandos citado acima é basicamente:

- a) abrir o editor de texto (*Open Office 3.1.1*);
- b) abrir o *browser (Mozilla Firefox 3.5.9)*, copiar uma determinada figura da Internet e inseri-la no texto;
- c) salvar o texto na área de trabalho e fechar o editor de texto;
- d) abrir um determinado vídeo e deixá-lo tocar até o fim; e
- e) encerrar a sessão.

O roteiro descrito acima foi escolhido por ser o método utilizados nos diversos trabalhos científicos pesquisados que envolvam o mesmo assunto. É importante frisar que ele reflete o anseio exposto no objetivo geral desta pesquisa.

Para alcançar o objetivo geral que é de estabelecer métricas para determinar a configuração mínima para um servidor LTSP, durante a execução dos testes, serão obtidas leituras do servidor por meio de ferramentas capazes de registrar o consumo de memória RAM e taxa de utilização da Unidade Central de Processamento

De posse dos dados colhidos pela ferramenta no decorrer dos testes, é possível estabelecer a razão do custo computacional num ambiente escalonado. Esse processo viabiliza a criação de projeções para um parque de informática que abrigue um número diversificado de *thin clients*.

## 5 TRABALHOS CORRELATOS

Na Universidade Tecnológica Federal do Paraná, Marcos Talau e Marcelo Rodrigues Botão, autores do trabalho Sistema de Terminal GNU/Linux LTSP 4.2 para o LTSP 5.0: Uma Análise Prática de Implantação e Desempenho, concluído em 2008, propõem uma unidade de métrica de desempenho LTSP, e ainda, apresentam, por meio desta ferramenta os dados relativos a melhora de desempenho ocorrida deste sistema entre a versão 4.2 para 5.0.

Na Universidade Federal de Lavras, Bruno Arriel Rezende, autor do trabalho Análise de Desempenho de Terminais Leves em Laboratórios de Informática, concluído em 2008, traça um comparativo positivo em relação ao uso da tecnologia LTSP frente aos tradicionais *desktops* Windows naquele laboratório de informática. Comenta ainda sobre dificuldades encontradas durante a implementação do LTSP e apresenta a ferramenta Ganglia para visualização da taxa de processamento da CPU.

Na Universidade Federal de Santa Catarina, Guilherme Arthur Geronimo, autor do trabalho Estudo em Terminais Leves como nós de um *Cluster*, concluído em 2007, apresenta a ferramenta Zabbix para o auferimento do custo de processamento no servidor de *clusteres* que usa terminais leves como nós.

Na Universidade Federal de Lavras, Carlos Ferreti, autor do trabalho Implementação de uma Rede de Computadores Baseada no *Linux Terminal Server Project*, concluído em 2004, expõe de maneira detalhada como é feita a configuração dos diversos arquivos que compõe o LTSP.

John Houser publicou em abril de 2009 no *Library Technology Reports*, uma extensão da *American Library Association*, um artigo intitulado *Open Source Public Workstations and Today's Library World* versando sobre aumento da procura de informação via meio eletrônico. Cita o benefício do LTSP, do ponto de vista econômico e ambiental, quando usado em bibliotecas dispostas pelo mundo.

## 6 TRABALHO DESENVOLVIDO

Com o objetivo de estabelecer a métrica para a configuração do servidor que deve ser adquirido para dar suporte ao ambiente LTSP pretendido, foi estabelecido um cenário para testes nos laboratórios da UNESC composto dos seguintes itens de hardware:

- a) um servidor com processador AMD Athlon 64 *Dual Core* 6400+, 3.21 Ghz, com 2 GB de memória RAM, 1 GB de SWAP, 250 GB de capacidade no disco rígido, duas interfaces de rede integradas NVIDIA nForce 100 Mbps;
- b) dez estações com processador AMD Athlon 64 *Dual Core* 4200+, 2200 Mhz, com 1 GB de memória RAM, 160 GB de capacidade no disco rígido, uma interface de rede Broadcom NetXtreme 57xx 100 Mbps; e
- c) dois *ethernet switch* 10/100 Mbps.

Como ferramentas de apoio consta o TOP (versão 3.2.8), um aplicativo nativo do LINUX que registra em tempo real os processos em execução, consumo de memória principal e memória *SWAP*. Além deste, foi utilizado o ZABBIX (versão 1.8.1), uma ferramenta de monitoramento que transforma em gráficos os valores extraídos do TOP.

### 6.1 RESULTADOS OBTIDOS

Os testes foram realizados em dois dias não consecutivos para caracterizar intervalos de tempo e obtenção de dados o mais aleatório possível.

No primeiro dia foram realizados dez testes obedecendo a metodologia descrita na sessão 5.1. Inicialmente foi ligada uma estação de trabalho, executadas as tarefas determinadas e finalmente encerrada a sessão. O processo foi repetido com o acréscimo de uma estação por vez tal que, na décima vez, dez estações estavam ligadas simultaneamente.

Observou-se o comportamento do uso de memória principal e uso de memória *swap* em todos os testes, como mostrado nas Figuras 14 e 15.

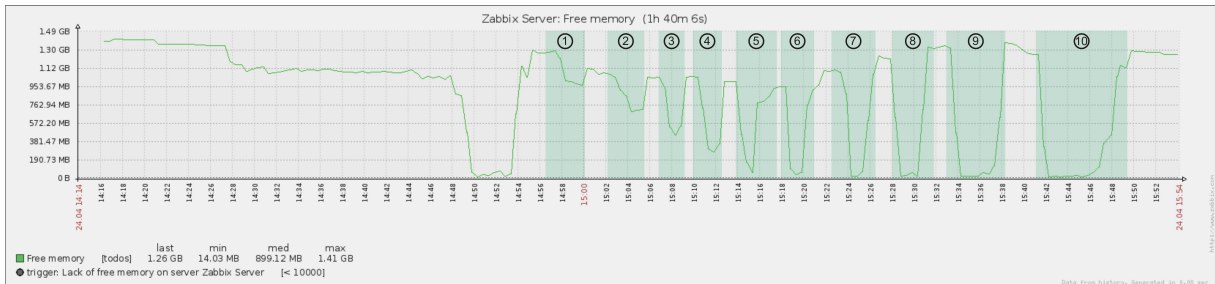


Figura 14. Memória principal disponível no primeiro dia

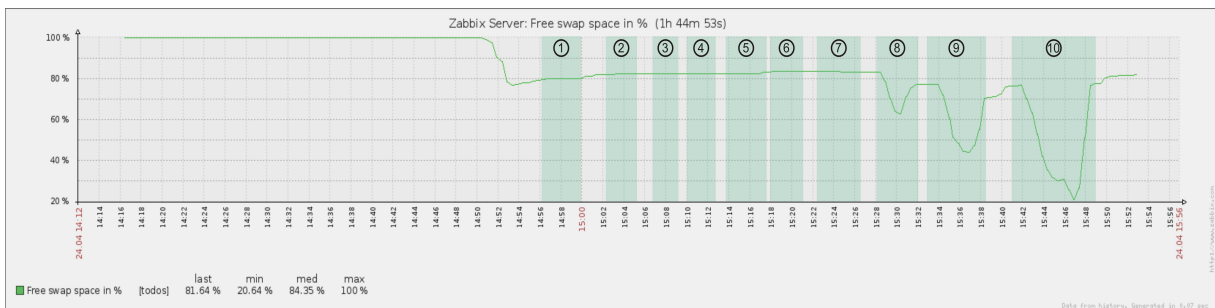


Figura 15. Memória *swap* disponível no primeiro dia

Nas Figuras 14 e 15 foram sombreados os períodos de execução dos testes. Cada fatia possui um número representativo do número de estações que estavam ligadas simultaneamente no servidor LTSP.

As figuras acima devem ser interpretadas juntas, pois refletem o comportamento de memórias complementares uma a outra. Nota-se que na Figura 14 o início de cada teste fica evidente, pois há a queda da linha representativa do total de memória livre. Quando é encerrada a sessão nos terminais, a linha do gráfico se eleva caracterizando a liberação de memória principal.

Na Figura 15 existem quedas nos testes com oito, nove e dez estações. Estas quedas refletem o término de memória principal (Figura 14), e alocação em *swap* pelo SO.

Os testes foram analisados separadamente e extraídos os picos de consumo de memória principal e memória *swap* (ambos em MB), em razão do número de terminais. Tais dados estão expostos na tabela a seguir.

Tabela 2. Picos de consumo no dia 24 de abril

	1	2	3	4	5	6	7	8	9	10
<b>RAM</b>	1042,55	1319,26	1564,1	1734,6	1948,88	1971,13	1989,91	1992,2	1993	1994,52
<b>SWAP</b>	196,08	172,35	170,91	170,62	169,38	161,92	164,4	357,23	537,04	759,44

Observou-se também a linearidade no aumento do consumo de RAM em cada teste. O acréscimo de um terminal aumenta, aproximadamente, 230 MB no consumo do servidor. Esse valor foi obtido pela média das diferenças de consumo de memória principal em razão do número de estações como apresentado na Tabela 3.

Tabela 3. Média do consumo de RAM

Nr Termimais	RAM	Diferença
1	1042,55	-
2	1319,26	276,71
3	1564,1	244,84
4	1734,6	170,5
5	1948,88	214,28

Abaixo o gráfico demonstra a linearidade do aumento do consumo de memória RAM. No eixo das abscissas está o número de terminais ligados e no eixo das ordenadas a quantia de memória principal consumida, em MB.

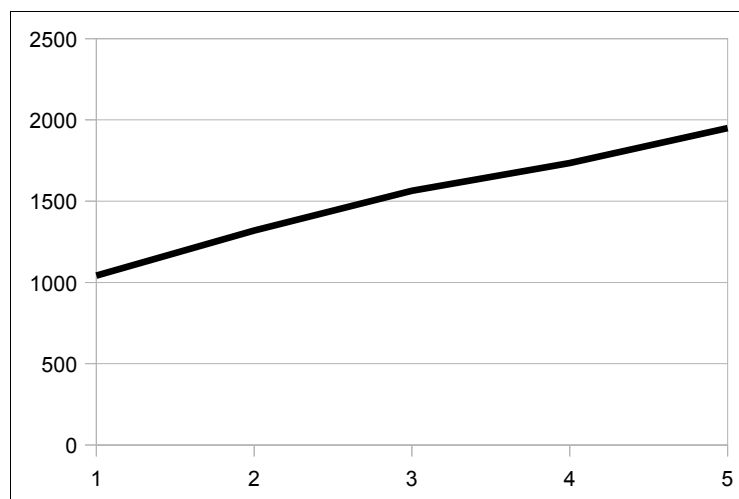


Figura 16. Consumo de memória principal

No segundo dia de testes a metodologia empregada foi a descrita na sessão 5.1 onde uma sequência de atividades deveria ser executadas em cada terminal. Desta vez os

testes foram realizados partindo-se de seis estações e terminando com nove. Além disso, os testes foram repetidos duas vezes. Nos gráficos abaixo, foram sombreados os períodos de execução de cada teste. Os números no topo de cada fatia representam o número de estações ligadas simultaneamente no servidor LTSP.

As variáveis memória principal e *swap* foram novamente monitoradas. Obtiveram-se os gráficos apresentado nas Figuras 17 e 18.

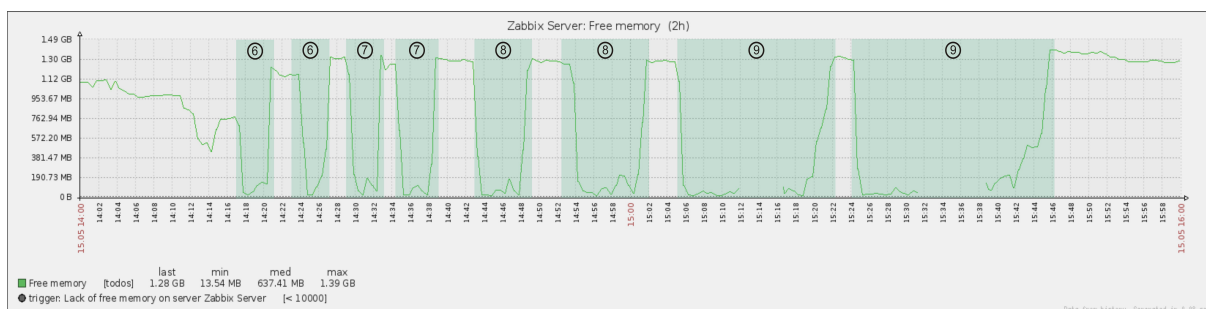


Figura 17. Memória principal disponível no segundo dia

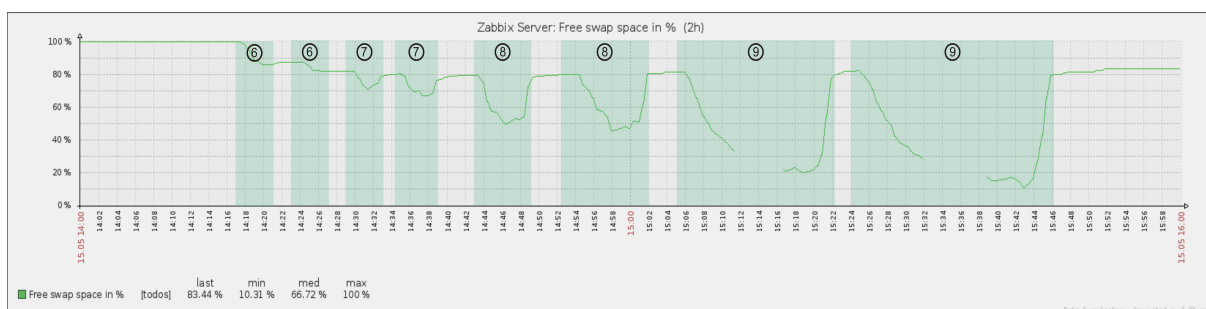


Figura 18. Memória *swap* disponível no segundo dia

Similarmente ao primeiro dia de testes, foram extraídos dos gráficos acima os picos de consumo de memória principal e memória *swap* em razão do número de terminais. Com base nesses dados foram formuladas as tabelas 4 e 5.

Tabela 4. Picos de consumo do dia 15 de maio – primeira rodada de testes

	6	7	8	9
<b>RAM</b>	1992,42	1991,94	1994,96	1995,01
<b>SWAP</b>	137,71	280,86	486,51	769,01

Tabela 5. Picos de consumo do dia 15 de maio – segunda rodada de testes

	6	7	8	9
<b>RAM</b>	1992,87	1992,29	1984,98	1989,85
<b>SWAP</b>	177,04	318,19	522,3	858,29

Seguem as taxas de inatividade de CPU registradas nos dois dias de testes.

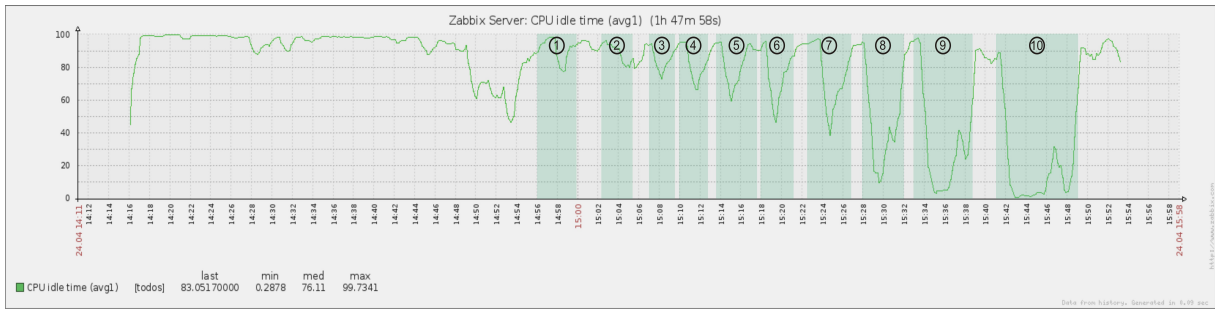


Figura 19. Inatividade de CPU no primeiro dia de testes

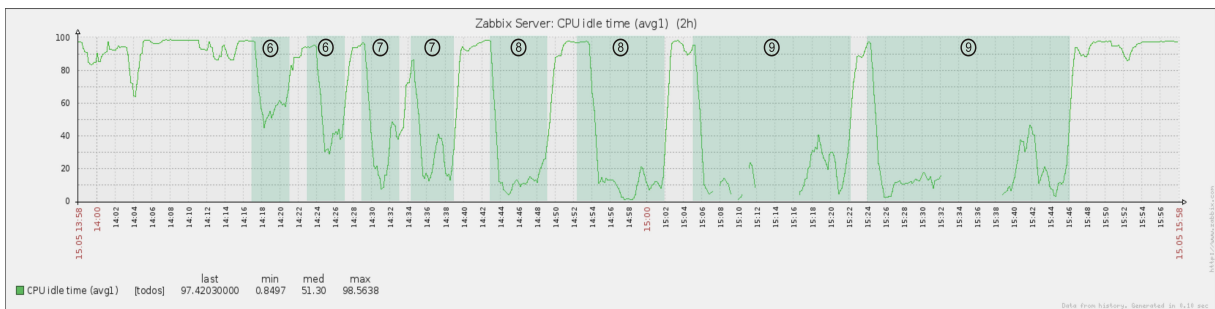


Figura 20. Inatividade de CPU no segundo dia de testes

Pôde-se verificar que o desempenho do servidor atendeu satisfatoriamente quando havia até seis estações de trabalho ligadas simultaneamente. A característica principal evidenciada com um número superior a seis foi uma tela de advertência informando o travamento de aplicativo e perguntando se o usuário desejava continuar processando ou forçar seu término. Nestas ocasiões sempre se optou por continuar processando, ou seja, nunca se optou por forçar o término de execução de qualquer aplicativo. Partindo desta escolha, os programas voltavam a responder com brevidade. Outro fato importante é que apesar das mensagens supra citadas, o servidor LTSP em nenhuma ocasião travou ou apresentou mensagens de alerta, ou seja, demonstrou-se estável durante todo o tempo. É fato que a taxa de processamento registrava-se elevada nestas ocasiões, conforme Figuras 19 e 20.

Devido ao bom desempenho apresentado pelo servidor com até seis estações de trabalho, no segundo dia o processo iniciou-se com seis terminais. O objetivo principal foi encontrar o fator crítico de travamento de aplicativos quando abertos simultaneamente nas estações.

Na Figura 21, estão representados os picos de consumo de memória *swap* em razão do número de terminais ligados ao servidor. Para cada número de estações ligadas existem três barras representando a repetição dos testes.

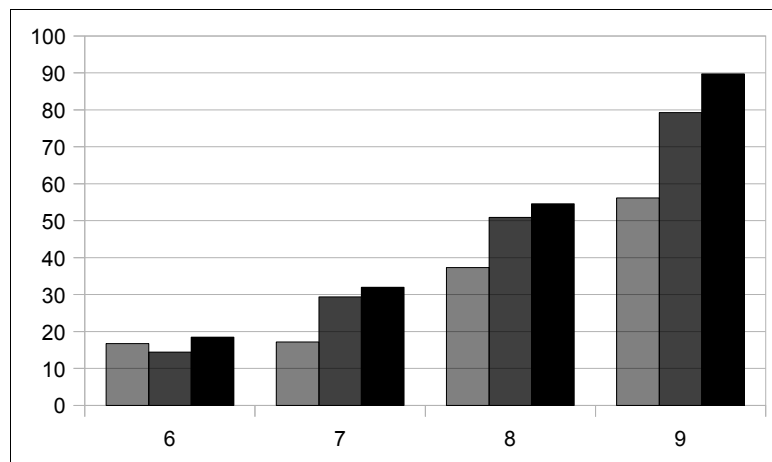


Figura 21. Consumo médio de *swap* %

Foi observada uma relação próxima entre consumo de *swap* e taxa de atividade de CPU. No momento que há mais de 30% de *swap* alocada, a taxa de atividade do CPU se eleva a índices muito próximo a 90% (Figura 20). Não só o fato da taxa de CPU encontrar-se elevada, quando este ocorre por muito tempo o desempenho cai acentuadamente, pois a tarefa de recuperar dados em disco rígido é muito mais lenta que na memória principal.

A execução deste teste visa obter parâmetros de configuração de um servidor para atender dez estações de trabalho, refletindo o objetivo da pesquisa. As máquinas são doadas, ou seja, busca-se uma solução de servidor que se mostre econômica e ao mesmo tempo de fácil aquisição no comércio local. Ao longo da análise houve linearidade entre os dados colhidos durante o aumento escalonar de requisições ao servidor LTSP. Também foi encontrado o limite de consumo de *swap* para que haja comprometimento da CPU no

ambiente de *hardware* e *software* testado.

Depreende-se, que baseado numa configuração de *hardware* comercial - neste caso 3,2 GHz e dois núcleos - observou-se a necessidade de aproximadamente 2,25 GB de memória principal para dez *thin clients*, além de 1 GB necessários para o SO.

## 7 CONCLUSÃO

A utilização de terminais leves mostra-se viável em ambientes comerciais, produtivos ou de ensino, pois ela permite a reutilização de computadores que seriam descartados por motivo de obsolescência. O *Linux Terminal Server Project* (LTSP) lança mão dessa tecnologia e seu emprego justifica-se quando redução do impacto financeiro precede a necessidade de desempenho.

Valendo-se da sugestão de trabalhos futuros apresentada em pesquisa passada, aqui foi analisado o servidor conforme o número de estações. Esta prática foi necessária para o dimensionamento de um servidor que dará suporte a dez *thin clients* instalados em um ponto de acesso para inclusão digital. Trata-se de um laboratório onde os usuários irão utilizar os computadores para digitação de documentos e acesso a Internet, portanto sem a necessidade de um sistema robusto capaz de suportar aplicativos de grande vulto.

Explorou-se os aspectos técnicos para implementação de redes de computadores até a instalação do LTSP. Os resultados obtidos foram satisfatórios, pois houve um aumento gradual uniforme do consumo do servidor ou longo do aumento do número de estações ligados a ele. Com estes números concluiu-se que é possível empregar um computador *desktop* de fácil aquisição no comércio local obedecendo-se uma recomendação de configuração mínima para que ele dê suporte aos dez *thin clients* pretendidos.

Os dados colhidos na análise oferecem um padrão que podem servir como parâmetro àqueles que desejam implementar um laboratório de informática com LTSP.

Sugere-se complementarmente a este trabalho que no futuro se estude o servidor aumento-se gradualmente de memória principal com o mesmo número de estações. Assim como o descrito anteriormente, sugere-se o mesmo estudo aumentando apenas a capacidade do CPU ao invés da memória principal.

## REFERÊNCIAS

- AIEX, Antônio Emílio W.; DEZIDERÁ, Ariane Gomes Da Silva; KAMINSKI, Maria Carolina. **Implementação e análise de desempenho de terminais Thin Clients nos computadores da Rede Local de Ensino**. 2006. 56 f. Trabalho de Conclusão do Curso (Tecnólogo em Informática) - Curso de Tecnologia Em Informática, Universidade Tecnológica Federal Do Paraná, Curitiba, 2006.
- BALNEAVES, Scott et al. **Linux Terminal Server Project Administrator's Reference Guide**. Disponível em: <<http://www.ltsp.org/~sbalneav/LTSPManual.html>>. Acesso em: 19 out. 2009.
- COMER, Douglas; STEVENS, David L. **Interligação em rede com TCP/IP**. Rio de Janeiro: Ed. Campus, 1999. 2.v
- COMER, Douglas E.. **Redes de Computadores e Internet: Transmissão de Dados, Ligações Inter-Redes, Web e Aplicações**. 4. ed. Porto Alegre: Bookman, 2007. 720 p. Disponível em: <<http://books.google.com.br/books?id=KDR4teVK0JAC>>. Acesso em: 20 out. 2009.
- FERRETTI, Carlos, **IMPLEMENTAÇÃO DE UMA REDE DE COMPUTADORES BASEADA NO LINUX TERMINAL SERVER PROJECT**. 2004. Monografia – Curso de Ciência da Computação, Universidade Federal de Lavras, Lavras, 2004. Disponível em <<http://bazar.ginix.ufla.br/index.php/MonosARL/article/viewDownloadInterstitial/75/70>>, acesso em 29 de outubro de 2009.
- FOROUZAN, Behrouz A.. **TCP/IP Protocol Suite**. 2 ed. New York: McGraw-Hill Companies, 2003. Disponível em <[http://books.google.com/books?id=HsCjH\\_V04tUC](http://books.google.com/books?id=HsCjH_V04tUC)> acesso em 22 Nov 2009.
- FOROUZAN, Behrouz. ; **Comunicação de Dados e Redes de Computadores**. 3. ed. Porto Alegre: Bookman, 2006. 840 p. Disponível em: <<http://books.google.com/books?id=C9ZN-jYKHpMC>>. Acesso em: 20 out. 2009.
- FREER, John R.; **Computer Communications and Networks**. 2 ed. Londres: Taylor & Francis, 1996. 394 p. Disponível em: <<http://books.google.com.br/books?id=fBJKfP1meKQC>>. Acesso em 20 out. 2009.
- GALLO, Michael A.; HANCOCK, Willian M. **Comunicação entre Computadores e Tecnologias de Rede**. São Paulo: Pioneira Thompson Learning, 2003. 674 p. Disponível em <<http://books.google.com/books?id=EPqgxOGYrTMC>> acesso em 10 Out 2009.
- GERONIMO, Guilherme Arthur. **Estudo em Terminais Leves como nós de um Cluster**. 2007. 50 p. Trabalho de Conclusão de Curso - Universidade Federal de Santa Catarina, Florianópolis, 2007. Disponível em < [http://projetos.inf.ufsc.br/arquivos\\_projetos/projeto\\_631/TCC.Guilherme.Arthur.Geronimo.pdf](http://projetos.inf.ufsc.br/arquivos_projetos/projeto_631/TCC.Guilherme.Arthur.Geronimo.pdf)>, acesso em 29 de outubro de 2009.
- HAYDEN, Matt. **Aprenda em 24 Horas Redes**. Rio de Janeiro: Campus, 1999. 461 p.
- HOUSER, John. Open Source Public Workstations and Today's Library World. **Library**

**Technology Reports**, Chicago, p. 5-7. 22 abr. 2009. Disponível em: <<http://alatechsource.metapress.com/content/n6726q514h186486/fulltext.pdf>>. Acesso em: 07 jun. 2010.

MORIMOTO, Carlos E. **Servidores Linux: Guia Prático**. São Paulo, GDH Press e Sul Editores, 2008, 736p.

MURHAMMER, Martin W.; GAERTNER, Jussara Licinia Souza. **TCP/IP: tutorial e técnico**. São Paulo: Makron Books, 2000. 690 p.

PISONI, Andréia Silva. **ANÁLISE DE DESEMPENHO DA SOLUÇÃO DE BOOT REMOTO DISKLESS EM REDES UTILIZANDO LINUX**. 2003. 118 f. Dissertação (Graduação) - Curso de Ciência Da Computação, Universidade Do Extremo Sul Catarinense, Criciúma, 2003.

PITANGA, Marcos. **Construindo Supercomputadores com Linux**. 3. ed. Rio de Janeiro: Brasport, 2008. 400 p. Disponível em: <<http://books.google.com.br/books?id=PbMKWMXeZgsC>>. Acesso em: 15 maio 2010.

POLLONI, Erico Giulio Franco; FEDELI, Ricardo Daniel; PERES, Fernando Eduardo. **Introdução à Ciência da Computação**. 1. Ed. São Paulo: Thomson Pioneira, 2003. 238 p. Disponível em <<http://books.google.com/books?id=eTKb86SwCQYC>>. Acesso em 29 de outubro de 2009.

REZENDE, Bruno Arriel; **ANÁLISE DE DESEMPENHO DE TERMINAIS LEVES EM LABORATÓRIOS DE INFORMÁTICA**. 2008. 77 p. Monografia de graduação- Curso de Ciência Da Computação, Universidade Federal de Lavras, Lavras, 2003. Disponível em <[http://www.bcc.ufla.br/monografias/2008/Analise\\_de\\_desempenho\\_de\\_terminais\\_leves\\_em\\_laboratorios\\_de\\_informatica.pdf](http://www.bcc.ufla.br/monografias/2008/Analise_de_desempenho_de_terminais_leves_em_laboratorios_de_informatica.pdf)>, acesso em 29 de outubro de 2009.

SANTOS, Moises Jose G. Dos; CAMPOS, Augusto. **Economize com Software Livre**. Disponível em: <<http://augustocampos.net/revista-do-linux/023/capa.html>>. Acesso em: 19 de outubro de 2009.

SCRIMGER, Rob et al. **TCP/IP a Bíblia**. Rio de Janeiro: Elsevier, 2002.

SOARES, Luiz Fernando G; LEMOS, Guido; COLCHER, Sérgio. **Redes de Computadores: das LANs, MANs e WANs às Redes ATM**. Rio de Janeiro: Campus, 1995.

TALAU, Marcos; BOTÃO, Marcelo Rodrigues. **SISTEMA DE TERMINAL GNU/LINUX LTSP 4.2 PARA O LTSP 5.0: UMA ANÁLISE PRÁTICA DE IMPLANTAÇÃO E DESEMPENHO**. Disponível em: <[http://sites.google.com/site/marcostalau/document/paper/sistema\\_de\\_terminal\\_gnu\\_linux\\_ltsp\\_4.2\\_para\\_o\\_ltsp\\_5.0\\_uma\\_analise\\_pratica\\_de\\_implantacao\\_e\\_desempenho.pdf](http://sites.google.com/site/marcostalau/document/paper/sistema_de_terminal_gnu_linux_ltsp_4.2_para_o_ltsp_5.0_uma_analise_pratica_de_implantacao_e_desempenho.pdf)>. Acesso em: 29 out. 2009.

TANENBAUM, Andrew S. **Redes de Computadores**. 3 ed. Rio de Janeiro: Campus, 1997.

THOMAS, Rosert M. **Introdução às Redes Locais**. Rio de Janeiro: Makron Books, 1997. 287 p.

TITTEL, Ed. **Rede de Computadores**. Porto Alegre: Artmed, 2002. 264p. (Chaum).  
Disponível em: <<http://books.google.com.br/books?id=21I-3aa7MAQC>>. Acesso em 20 de outubro de 2009.

TOBLER, Michel J. **Inside Linux**. Indianapolis: New Riders Publishing, 2001. 778 p.  
Disponível em: <<http://books.google.com/books?id=K0iOxlCfPdgC>>. Acesso em 19 de outubro de 2009.

UNIVERSIDADE DO EXTREMO SUL CATARINENSE  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

LUCAS NASCIMENTO MARTINS CAMARGO DA SILVA

ANÁLISE DE CUSTO DE ESCALABILIDADE EM AMBIENTE LINUX TERMINAL  
SERVER PROJECT (LTSP)

CRICIÚMA, JUNHO DE 2010