

Faculdade de Engenharia da Universidade do Porto



Laboratório Remoto de Redes de Computadores

Pedro César Bessa Magalhães Oliveira

Dissertação realizada(o) no âmbito do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores
Major em Telecomunicações

Orientador: Prof. Dr. Ricardo Santos Morla

Julho de 2008

© Pedro Oliveira, 2008

Resumo

Nesta dissertação, foi realizado um estudo sobre o estado da arte de laboratórios remotos com o objectivo de avaliar qualitativamente o Laboratório de Redes da Faculdade de Engenharia da Universidade do Porto.

O estudo permitiu definir quais os caminhos a seguir para o futuro do eCassiopeia, com a finalidade de melhorar o serviço prestado à comunidade de alunos que utilizam o laboratório.

Com base nessa análise e depois de conhecidas as funcionalidades do eCassiopeia, foram propostas soluções para as fragilidades detectadas, tanto a nível de melhorias da solução actual como a proposta de uma nova arquitectura, com o intuito de desenvolver um serviço, capaz de possibilitar aos alunos o acesso remoto ao laboratório, para prepararem previamente os exercícios que lhes são propostos, semanalmente, durante o programa das disciplinas que cobrem os conteúdos das redes de computadores.

O acesso remoto permite eliminar as barreiras temporais e geográficas inerentes à utilização física do laboratório, fazendo deste uma ferramenta útil a professores e alunos, que apenas passam a necessitar de um computador com Java instalado, um browser e acesso à Internet.

Palavras-chave: Laboratório Remoto, Laboratório de Redes, Acesso Remoto.

Página em branco

Abstract

In this thesis, a study was conducted on the state of the art of remote laboratories to evaluate qualitatively the Network Laboratory of the Faculdade de Engenharia da Universidade do Porto.

The study allowed the definition of the path to take in the future of eCassiopeia, with the end of improving the service provided to the student community that uses the laboratory.

Based on that analysis, and knowing of the features of eCassiopeia, solutions were proposed and weaknesses identified both to the existing solution as to the new architecture proposed, in order to develop a service capable of providing the students remote access to the laboratory, so they can prepare, in advance, the weekly exercises proposed in computer network courses.

The remote access eliminates both the time and geographical barriers inherent in the use of a physical laboratory, making this a useful tool for teachers and students. For this access, a computer with a browser installed, a Java plugin and Internet access is the only thing needed.

Keywords: Remote Laboratory, Network Laboratory, Remote Access.

Página em branco

Agradecimentos

Agradeço a contribuição do João Taveira, com as suas opiniões sobre soluções a utilizar na implementação e todo o apoio técnico prestado na utilização do laboratório.

Deixo também uma palavra de agradecimento ao Eng. Paulo Correia da PT Prime, que me permitiu conciliar a frequência deste mestrado integrado, com o estágio profissional que desenvolvo na mesma empresa.

Página em branco

Índice

Resumo.....	iii
Abstract.....	v
Agradecimentos	vii
Índice.....	ix
Lista de figuras	xiii
Lista de tabelas.....	xv
Glossário e Abreviaturas.....	xvii
1. Introdução	1
1.1 Contexto	1
1.2 Objectivos.....	1
1.3 Estrutura do Dissertação	2
1.4 Principais contribuições	2
2. A web e o Ensino de Redes	5
2.1 Introdução	5
2.2 Porquê um laboratório remoto?.....	5
2.3 Estado da Arte	6
2.3.1VITELS	6
2.3.2University of Colorado.....	7
2.3.3Escola Politécnica da Universidade de São Paulo	9
2.3.4University of Massachusetts Amherst	10
2.3.5O ambiente laboratorial virtual	12
2.4 O eCassiopeia e as barreiras à sua utilização na FEUP.....	12
2.5 Conclusão	13
3. Solução eCassiopeia existente	15
3.1 Arquitectura Funcional	15
3.1.1Infra-estrutura da rede do laboratório	17
3.1.2Endereçamento do Laboratório eCassiopeia	17

3.2	Ligação ao exterior.....	18
3.2.1	A Firewall.....	18
3.2.2	O acesso ao laboratório via web	18
3.2.3	Controlo de acessos	19
3.3	Manipulação das topologias de rede	20
3.4	Autenticação dos utilizadores	21
3.5	Servidor Web apache	22
3.5.1	Módulo Autenticação	22
3.5.2	Agenda	22
3.5.3	Aplicação Web eCassiopeia.....	22
4.	Melhorias à Solução Existente	25
4.1	Arquitectura Funcional	25
4.1.1	Sistema Operativo	27
4.1.2	O Servidor Apache HTTP	27
4.1.3	PHP	28
4.1.4	Base de Dados MySQL.....	28
4.1.5	O phpMyAdmin.....	29
4.2	Interacção Utilizador/Consola de equipamento	30
4.2.1	A Consola Telnet	30
4.3	Painéis de repartição automáticos	31
4.4	Cálculo de Recursos Disponíveis	34
4.4.1	Equipamentos.....	34
4.4.2	Ligações patch panel	35
4.5	Adição da segunda interface ethernet ao Tux	35
4.6	Avaliação.....	35
4.6.1	Arquitectura funcional	36
4.6.2	Consola Java	38
4.6.3	Patch Panel	39
4.6.4	Segunda interface no Tux3	40
4.6.5	Recursos Ocupados	42
5.	Proposta de Nova Arquitectura	45
5.1	Patch Panel	45
5.1.1	Arquitectura Plana.....	47
5.1.2	Arquitectura Hierárquica	48
5.1.3	Arquitectura com três Patch Panel	49
5.2	Laboratórios Virtuais.....	51
5.2.1	Remote Desktop.....	52
5.2.2	Acesso Telnet / Ssh.....	53
5.2.3	Integração do Laboratório Virtual e Remoto	53
5.3	Laboratório Global	54
5.3.1	VLAN over Internet	55
5.3.2	GRE / PPTP.....	57
5.3.3	L2F / L2TP.....	59
5.3.4	IPSec.....	60
5.3.5	Conclusão.....	61
5.4	O Moodle	61
5.4.1	Autenticação e Controlo de Acesso	62
5.4.2	Calendarização	62
5.4.3	Segurança	62
5.4.4	Ferramentas Colaborativas.....	62
5.4.5	Ferramentas de Organização pessoal e de grupo	63

5.5	Interface Web	63
5.5.1	A Base de Dados	64
5.5.2	Web Design versus PHP Scripting.....	64
5.5.3	Módulos a Integrar	65
5.6	Acesso remoto.....	68
5.6.1	Servidor de Consolas e Servidor de Terminais	68
5.6.2	Principais características do servidor de consolas	69
5.6.3	Telnet ou SSH ?	69
5.7	Servidor de Configurações	70
5.8	Microsoft ConferenceXP	70
5.9	Conclusões	71
6.	Conclusão	73
	Referências Bibliográficas	75
	Apêndice A - Consola Patch Panel.....	79

Página em branco

Lista de figuras

Figura 3.1 - Arquitectura da Solução eCassiopeia	15
Figura 3.2 - Fluxo de pedidos na Solução eCassiopeia existente	16
Figura 3.3 - Módulos Funcionais eCassiopeia	17
Figura 3.4 - Acesso do Laboratório à Internet.....	19
Figura 4.1 - Implementada na nova revisão do eCassiopeia	26
Figura 4.2 - Fluxo de pedidos na Solução Implementada na nova revisão do eCassiopeia	26
Figura 4.3 - Servidor HTTP.....	27
Figura 4.4 - Aparência gráfica do phpMyAdmin	29
Figura 4.5 - Consola de equipamento lançada através de clique com rato	31
Figura 4.6 - CLI para controlo do patch panel	32
Figura 4.7 - Topologia de teste das funcionalidades implementadas	36
Figura 4.8 - Agenda do Laboratório e reserva de horário para realizar um exercício.....	37
Figura 4.9 - Página típica de um exercício de redes.	37
Figura 4.10 - Configuração de IP nas interfaces do TUX13 através da Consola Java.....	38
Figura 4.11 - Informação sobre as interfaces do TUX13.....	39
Figura 4.12 - Alteração da Matriz de Comutação de portos no Patch Panel.....	40
Figura 4.13 - Traceroute do tux11 para o tux12 passando pelo tux13.....	41
Figura 4.14 - Traceroute do tux12 para o tux11 passando pelo tux13.....	42
Figura 4.15 - Mensagem de alerta se não existir material disponível para realizar um exercício.	43
Figura 4.16 - Mensagem de alerta se não existir ligações no patch panel suficientes para implementar a topologia do exercício	44
Figura 5.1 - Proposta Arquitectura Plana	47

Figura 5.2 - Proposta Arquitectura Hierárquica	48
Figura 5.3 - Arquitectura com 3 Patch Panel.....	50
Figura 5.4 - Arquitectura 3 Patch Panel em Anel.....	50
Figura 5.5 - Interface Gráfica do GNS3.....	52
Figura 5.6 - Consola do Router Virtual R0 (exercício da figura 5.5)	52
Figura 5.7 - Interligação de laboratórios.....	54
Figura 5.8 - Serval - Relação entre o cliente e o driver ethernet	56
Figura 5.9 - Serval troca de mensagens entre cliente e servidor.....	57
Figura 5.10 - Túnel GRE encapsulando PPTP	58
Figura 5.11 - Túnel L2TP.....	59
Figura 5.12 - Túnel IPSEC	60
Figura 5.13 - Esquema ligação de um exercício	65
Figura 5.14 - Arquitectura para o Servidor de Consolas.....	70
Figura 5.15 - Arquitectura do Serviço ConferenceXP	71
Figura 5.16 - Visão Macro da Arquitectura Proposta.....	72

Lista de tabelas

Tabela 4.1 — Associação dos Portos do Patch Panel e Equipamentos do Laboratório.	39
---	----

Página em branco

Glossário e Abreviaturas

Base de dados

Conjunto de dados com uma estrutura regular que representam informação, utilizada normalmente para um determinado fim.

Browser

Ferramenta utilizada para ver ficheiros na World Wide Web, tipicamente em formato html (*HyperText Markup Language*).

Chat

Refere-se à utilização de ferramentas que permitem a conversação, por via de texto, entre indivíduos. Esta conversação decorre de forma síncrona, isto é, a emissão e a recepção ocorrem simultaneamente.

CLI - Command Line Interface

Permite ao utilizador escrever texto que representa comandos perceptíveis por um equipamento.

DMZ - Demilitarized Zone

Zona de acesso público, acessível através da Internet.

Encriptação

É a transformação da informação na sua forma original, para outra forma ininteligível, a menos que seja conhecida a chave de descriptação, o que torna difícil de ser lida por alguém que não possua essa chave. Garante-se assim, que só o receptor da mensagem, que possui a chave, pode ler a informação.

Firewall

É um dispositivo de uma rede de computadores que tem por função regular o tráfego entre redes distintas e impedir a transmissão e/ou recepção de dados nocivos ou não autorizados.

FEUP

Faculdade de Engenharia da Universidade do Porto

Fórum

Ferramenta que permite a afixação, com carácter permanente e sequencial, de mensagens de texto (que, tipicamente, podem ser personalizadas ou anónimas) que ficam visíveis para todos os outros indivíduos. O acesso a um fórum pode ser restrito.

Java

Java é uma linguagem de programação orientada aos objectos. Ao contrário das linguagens convencionais, que são compiladas para código máquina, a linguagem Java é compilada para um "bytecode", que é executado por uma máquina virtual.

Laboratório remoto

Um laboratório real, mas que pode ser acedido através da Internet.

HTML - HyperText Markup Language

Formato típico em que são gravados os conteúdos visualizáveis nos browsers e mantidos na World Wide Web. Tem como aspecto mais característico a possibilidade de incluir links entre a restante informação.

HTTP - HyperText Transfer Protocol

Protocolo que permite transferência de ficheiros html.

IP - Internet Protocol

Protocolo que permite a comunicação na Internet.

ISP - Internet Service Provider

Operador que permite o acesso de um cliente à internet

LAN - Local Area Network

Rede privada de computadores de uma empresa ou escola

Perl

Perl é uma linguagem de programação estável e independente do sistema operativo. Foi largamente adoptada pelas suas potencialidades em processamento de strings (variáveis de texto) e por ser resistente a falhas arbitrárias, destacando-se o seu uso no desenvolvimento de aplicações web de todos os tipos.

PHP - Hypertext Pre-Processor (PHP)

É uma linguagem de programação, livre e muito utilizada para gerar conteúdo dinâmico na Web, orientada aos objectos.

Servidor

Um servidor é uma máquina que oferece serviços a uma rede de computadores. A rede é assim do tipo cliente-servidor.

SQL - Structured Query Language

Linguagem que permite fazer consulta de informação num base de dados

SSH - Secure Shell

Protocolo para comunicação segura de um cliente e um servidor através da Internet

Unix

Unix é um sistema operativo, multitarefa e multiutilizador originalmente criado por Ken Thompson, que trabalhava nos Laboratórios Bell (Bell Labs) da AT&T. O sistema Unix é constituído por vários componentes como o Kernel, o ambiente de desenvolvimento, bibliotecas, documentos e comandos. O grande sucesso deste tipo de sistema operativo levou ao aparecimento de distribuições gratuitas baseadas em Unix, entre elas o Linux.

URL - Uniform Resource Locator

É um endereço de um recurso disponível numa rede, seja a internet ou uma rede empresarial, a intranet. O URL tem a seguinte estrutura: protocolo://máquina/caminho/recurso

VLAN - Virtual LAN

Permite a co-existência de várias lan num mesmo equipamento identificadas por uma tag que se coloca nas tramas

VPN - Virtual Private Network

Ligação de várias redes remotas que do ponto de vista de endereçamento formam uma única rede virtual

VRML - Virtual Reality Modeling Language

Linguagem utilizada para tornar virtual um equipamento real

WAN - Wide Area Network

Rede de computadores de nível nacional

WIKI

Permite a partilha de conteúdo recorrendo ao formato html. Os utilizadores com permissão de acesso podem alterar e corrigir os conteúdos da informação da página.

WWW - World Wide Web

A World Wide Web (que significa "rede de alcance mundial", em inglês; também conhecida como Web e WWW) é um sistema de documentos hipermédia interligados usando a Internet. Para visualizar a informação, pode-se usar um programa de computador chamado navegador (web browser) para descarregar informações de servidores web e mostrá-los no monitor do utilizador.

Capítulo 1

1. Introdução

1.1 Contexto

A constante evolução do mundo das telecomunicações, nomeadamente na área das redes, leva à necessidade de dotar os alunos de conceitos teóricos sobre arquitectura e configuração de redes. A existência de um laboratório para a consolidação desses conceitos, torna-se uma exigência que a Faculdade de Engenharia da Universidade do Porto tenta satisfazer.

Neste sentido e com o intuito de tornar mais eficiente a utilização do laboratório desta instituição, foi no passado construído um protótipo, o eCassiopeia, para disponibilizar o acesso remoto ao laboratório, permitindo a alunos e professores utilizarem os equipamentos do laboratório, quebrando barreiras temporais e geográficas.

Este protótipo está no entanto fora de produção desde o ano lectivo de 2004/2005.

Na sequência de uma renovação do laboratório e dos equipamentos que o compõem, sabendo-se à partida da mais valia que o acesso remoto significa para os seus utilizadores, levantou-se a questão da não utilização do eCassiopeia. Com esta questão surgiram outras de âmbito mais lato, que questionavam a possibilidade do protótipo estar temporalmente inadequado às novas exigências, bem como saber a possibilidade de integração dos novos equipamentos no sistema.

Assim surgiu a ideia de propor um estudo capaz de responder a estas questões, que culmina com a apresentação desta tese, a qual pretende dar resposta às perguntas levantadas, desde logo comparando as suas funcionalidades com o que de melhor se encontra em laboratórios remotos existentes e deixando em aberto propostas para o futuro do eCassiopeia.

Como aluno desta instituição e conhecendo o conceito do eCassiopeia, apesar de nunca o ter utilizado, abracei a ideia ciente da importância que o meu trabalho terá, para criar uma ferramenta que permita sem dúvida, aumentar o nível da qualidade de ensino na área das redes, na Faculdade de Engenharia da Universidade do Porto.

1.2 Objectivos

O objectivo desta tese é analisar a solução de laboratório de redes remoto eCassiopeia, estudar e propor melhorias à sua actual implementação e arquitectura.

Dado que o laboratório de redes remoto eCassiopeia, tem como principal objectivo proporcionar aos alunos o acesso a equipamento real à distância, para apoio às aulas práticas na área das redes de computadores, foi necessário estudar o estado da arte de outros laboratórios remotos e, com base nesse estudo, avaliar qualitativamente o eCassiopeia do modo como estava implementado. Sendo o eCassiopeia o objecto de comparação, foi fundamental garantir a sua funcionalidade e perceber as suas mais valias e limitações.

A arquitectura que o suportava foi à partida um ponto a melhorar. Dadas as capacidades de processamento das actuais máquinas, não fazia sentido a utilização de três servidores para garantir o serviço, pelo que esse foi logo um foco de atenção, implementar o serviço apenas numa máquina garantindo a segurança no acesso. O segundo ponto passava por permitir realizar exercícios com topologias fixas predefinidas e com topologias definidas pelo utilizador, dinamicamente e de modo automático, sem a intervenção manual do técnico do laboratório, como acontecia no passado.

Além do trabalho que se pretendia desenvolver, era necessário dar respostas às perguntas que estiveram na base deste trabalho de mestrado, pelo que foi necessário estudar e documentar as respostas encontradas.

1.3 Estrutura do Dissertação

A dissertação estrutura-se em seis capítulos.

No capítulo 2, apresenta-se o conceito de laboratório remoto e o resultado do estudo efectuado sobre o estado da arte em termos das tecnologias de rede para suportar a investigação e o ensino à distância na área de redes. Aborda-se também o estado do eCassiopeia na FEUP e a possibilidade de integrar um ambiente virtual na arquitectura do eCassiopeia.

No capítulo 3, faz-se uma descrição detalhada do laboratório de redes remoto eCassiopeia, implementado na FEUP, descrevendo-se a sua arquitectura e serviços.

No capítulo 4, descrevem-se as melhorias que se realizaram na actual arquitectura do eCassiopeia e que visaram resolver alguns problemas detectados. Demonstra-se também a operacionalidade das melhorias, com o recurso a um exercício de teste que aborda todas as alterações implementadas.

O capítulo 5 descreve uma proposta detalhada para uma arquitectura futura do eCassiopeia, tendo em conta as actuais tecnologias de redes e serviços, prevendo que o trabalho seja feito por módulos. Assim permitirá que a implementação seja efectuada por várias pessoas, no âmbito de mestrados, sem que os trabalhos se sobreponham e adaptando-se os objectivos à duração da disciplina de dissertação. A arquitectura prevê também a integração do eCassiopeia com ferramentas já utilizadas pela Faculdade e geridas pelos seus técnicos.

O último capítulo, o capítulo 6, refere as principais conclusões retiradas e a minha visão do trabalho.

1.4 Principais contribuições

As principais contribuições desta dissertação podem ser sumariadas como:

- Recolha sistematizada de aplicações e funcionalidades de laboratórios remotos, aplicáveis na investigação e no ensino da engenharia de redes;

- Implementação de uma nova arquitectura funcional para o sistema, adequada ao processamento dos dias de hoje;
- Introdução do conceito de topologia de rede a pedido, com o respectivo hardware associado;
- Implementação de uma consola para configuração do Patch Panel da APCON
- Algoritmo de verificação de recursos de rede e do Patch Panel disponíveis
- Identificação de funcionalidades a corrigir e de novas necessidades a satisfazer na implementação actual.
- Proposta de uma nova arquitectura para laboratórios remotos, baseada num estudo das limitações da actual arquitectura para desenvolvimento futuro;

Capítulo 2

2. A *web* e o Ensino de Redes

2.1 Introdução

Neste capítulo, faz-se uma introdução aos conceitos envolvidos na definição de laboratório remoto, salientando-se também as vantagens deste tipo de laboratórios para utilizadores e instituições.

Expõe-se por fim, o resultado de um estudo realizado sobre o estado da arte em termos de laboratórios de redes remotos, com características similares às do modelo em que incide o estudo e a proposta que resulta do trabalho realizado.

2.2 Porquê um laboratório remoto?

As vantagens da existência de um laboratório remoto poder-se-ão enumerar como:

1. Permite flexibilidade de horários no acesso ao laboratório a todos os intervenientes, desde professores, alunos e técnicos.
2. Não é necessária uma sala própria para se efectuarem os trabalhos práticos, pois é possível a partir de casa efectuar a componente prática de um curso de redes.
3. Não é necessário a definição de horários com a presença de um técnico ou professor para supervisionar o acesso ao equipamento
4. Satisfaz as necessidades dos estudantes à distância sem a condicionante do tempo.
5. Diminuição de gastos com pessoal para assistência técnica ao laboratório.
6. Rentabilização do custo do equipamento pois pode ser acedido 24h por dia.
7. Permite a interacção dos alunos com equipamento real, ao invés de software de simulação.

2.3 Estado da Arte

A pesquisa que realizei com o objectivo de conhecer o que existe actualmente sobre Laboratórios Remotos de Apoio ao Ensino na área das Redes, levou-me de encontro a cinco projectos. Estes são as versões Web-Based dos laboratórios de redes das seguintes universidades:

- VITELS - Virtual Internet Telecommunications Laboratory of Switzerland - laboratório partilhado pelas variadas universidades desse país.
- University of Colorado
- Polytechnic School of University of Sao Paulo
- University of Massachusetts Amherst - R-LAB
- University of Massachusetts Amherst - IV-LAB

Existe também um outro conceito de apoio ao ensino de redes, não com a utilização de equipamento real, mas sim com o uso de softwares de emulação que suportam os sistemas operativos dos equipamentos de redes, possibilitando assim a criação de um ambiente laboratorial virtual. Também será estudado este conceito de apoio à realização de trabalhos de redes.

2.3.1 VITELS

A informação sobre este laboratório é limitada, dado apenas representar uma pequena parte de um projecto mais extenso a longo prazo [14].

É um laboratório que pretende ser partilhado por todas as universidades da Suíça, e desenvolvido em módulos, estando ao em cargo de cada universidade determinado módulo. Assim, o objectivo final deste projecto pretende abordar os seguintes temas:

8. Instalação e Configuração do Sistema Operativo Linux
9. Simulação Redes IP
10. Configuração e Avaliação da Performance de uma Rede IP Real
11. Programação Cliente/Servidor
12. Análise de Protocolos
13. Segurança IP
14. Gestão de Firewall

A informação disponível apenas se refere ao primeiro ponto “Linux System Installation and Configuration” e diz ser possível instalar e configurar um computador com o sistema operativo Linux, a partir de uma máquina virgem. Isto permite aprender a trabalhar com o kernel do Linux e a instalar os seus subsistemas. O aluno pode aceder à máquina através de bootstrap e fazer

todos os passos necessários, desde preparar o sistema de ficheiros, instalar o kernel e os subsistemas. O aluno tem que configurar e personalizar todos os componentes necessários localmente e nos servidores remotos, de modo a que a máquina tenha uma configuração que a permita trabalhar na rede.

O laboratório é composto por seis computadores, um servidor, o servidor de bootstrap Rembo5 e configuração de apoio. Os exercícios remotos práticos, são semelhantes aos exercícios elaborados no contexto do laboratório físico. É feito o agendamento das tarefas de modo a otimizar a utilização do equipamento.

2.3.2 University of Colorado

Os autores do projecto nesta universidade justificam a sua necessidade atendendo aos seguintes pontos:

15. Focar o trabalho na aprendizagem de redes
16. Possibilitar múltiplos métodos de acesso a material educativo
17. Fornecer uma matriz de configuração, para apoiar em tempo real reconfigurações de elementos reais de uma rede
18. Controlo cuidadoso do acesso ao ambiente de aprendizagem
19. Criar um ambiente que reproduza (não apenas emule) a experiência laboratorial.
20. Desenvolver um ambiente padrão para o desenvolvimento de experiências laboratoriais, assistido por professores e/ou auxiliares no processo pedagógico.
21. Criar um sistema de gestão de acessos ao sistema agendado e autorizado, evitando o acesso aos equipamentos por vários alunos em simultâneo
22. Impossibilitar o acesso aos recursos do laboratório a utilizadores não autorizados
23. A utilização do browser para acesso à aplicação, evitando que os alunos necessitem de instalar ou comprar qualquer tipo de software ou hardware para aceder ao RELI
24. Disponibilizar material de apoio aos alunos, quando não dispõem de assistente técnico que os possa ajudar
25. Capacidade de supervisionar o aluno durante a sua actividade
26. Ambiente virtual baseado em simulações
27. Capacidade de o aluno agendar dinamicamente os recursos do laboratório e seleccionar qual o exercício que pretende resolver.

O documento data de 2005 e revela que desenvolveram o ReLI - Remote Laboratory Infrastructure [15] durante três anos e teria acesso via Internet pelo endereço <http://ReLI.colorado.edu>, utilizando um Browser, ao qual tentei aceder mas nesse momento o link não estava activo.

O primeiro trabalho surgiu no âmbito do conceito e utilização de instrumentos virtuais, de ambientes experimentais simulados e por fim, o acesso dos alunos ao equipamento físico remoto, melhorando a qualidade da sua aprendizagem no sector das telecomunicações.

Com o decorrer do trabalho tornaram-se possíveis funcionalidades como:

- Agendar as actividades de modo a evitar contenção dos recursos.
- Possibilidade de configuração remota de equipamentos de redes
- Focar o nível da aprendizagem nos sistemas de redes
- Possibilidade de criar um ambiente experimental reconfigurado remotamente
- Avaliar o resultado da actividade experimental
- Permitir o acesso ao laboratório nas suas três vertentes: virtual, remotas e real.
- Permitir reiniciar os equipamentos sem perda de conectividade
- Possibilita usar scripts para recolocar o equipamento em determinado estado.

Os autores do documento consideravam que em 2005, o tema dos laboratórios remotos estava ainda na fase da “adolescência” das experiências, pelo que não foi possível avaliar, com resultados precisos, o nível de envolvimento destes no processo educativo.

Caracterizam assim o estado da arte nessa época, com a utilização de ambientes simulados limitados, que por norma não permitem simular plenamente um ambiente real, que necessitam de um grande esforço para se manterem actualizados, devido à constante evolução dos ambientes reais e que por fim nunca são totalmente precisos na sua intenção de simular um sistema verdadeiro. Existem outras ferramentas como routers virtuais que possibilitam ao aluno compreender o comportamento dos protocolos.

Mesmo assim, foi utilizado o software Network Simulator, com o objectivo de os alunos utilizarem este recurso como introdução à actividade laboratorial, enquanto não souberem utilizar correctamente equipamentos reais.

Outras ferramentas como os geradores de tráfego e sniffers também são utilizados, sendo que o acesso a estas ferramentas é via remote desktop utilizando o REALVNC.

Acabam por concluir, com a afirmação que já existem laboratórios remotos com acesso via Web, mas que não permitem ao utilizador reconfigurar os elementos da Rede.

Quanto à avaliação do projecto por parte dos seus utilizadores, constatou-se que:

- Foram mais tolerantes a falhas do que se esperava
- Mostraram-se extremamente satisfeitos pelo novo método de acesso ao laboratório
- Recomendavam a outros colegas essa disciplina
- Toleravam falhas técnicas, esperando a disponibilidade de um técnico ou do professor para corrigir o problema.
- Consideraram valioso o vídeo como material de apoio
- A capacidade de repetir as experiências era muito valorizada

As deficiências apontadas ao RELI foram:

- As falhas técnicas resultavam em horas de atraso para os alunos
- A necessidade de melhorar a GUI

- Apresentação do esquema de ligações

2.3.3 Escola Politécnica da Universidade de São Paulo

Os autores deste projecto [16] [17] propõem-se, logo à partida, a implementar um laboratório onde cada grupo de trabalho, tem a possibilidade de criar e implementar a sua própria topologia, ao invés das topologias estáticas. Para o suporte do trabalho colaborativo, integraram no projecto a ferramenta de e-learning TIDIA - AE [18] desenvolvida em São Paulo no Brasil.

As ferramentas de auxílio ao professor permitem organizar os alunos em grupos, bem como criar pequenas tarefas laboratoriais que em conjunto formam um exercício. Assim, com base nessas tarefas e na sua combinação, é possível criar uma infinidade de exercícios. Esta facilidade tem como objectivo ajudar o professor na sua tarefa educativo e permitir a reutilização de tarefas já definidas por outros professores, rentabilizando o tempo na preparação das aulas. Cada tarefa pressupõe a utilização de hardware e software.

O laboratório é constituído pelos seguintes componentes:

- A. Software
 1. Network Analyzer
 2. Performance Monitors
 3. Traffic Generator
 4. Network Management Tool
- B. Simuladores de Redes
 1. Collab - a sua utilização tem o objectivo de testar a topologia. Se for viável é implementada no laboratório para testes reais.
- C. Robôs e Câmaras
 1. Robô com uma câmara de vídeo anexada que pode ser controlada para obter mais detalhe nas ligações dos cabos.
 2. Robôs com braço, que permitem aos alunos o controlo para corrigir, por exemplo, uma ligação de um cabo errado.
- D. Hardware

A simulação tem apenas o objectivo de testar a solução pretendida. Após um teste válido os alunos enviam por e - mail, todas as ligações para o laboratório para que um técnico as reproduza manualmente, agendando uma data. Com os robôs apenas é possível corrigir algumas situações e não configurar todo o exercício.

Este laboratório esteve em fase experimental no 1º semestre do ano 2006, ao serviço da disciplina de redes do curso de Electrónica e Computadores daquela Universidade.

Na segunda fase do NetLab, pretendem construir um painel virtual (switch panel), que irá fazer praticamente todas as ligações desejadas pelos alunos.

O ambiente NetLab não tem em consideração o perfil do estudante, pois o LMS - Learning Management escolhido, o TIDIA - AE, ainda não considera esta opção.

2.3.4 University of Massachusetts Amherst

O trabalho realizado por esta faculdade divide-se em duas fases diferentes, pelo que agindo em coerência com o trabalho, o resumo estará também dividido.

2.3.4.1 R - LAB: WEB ENABLED REMOTE LABORATORY

A instalação e manutenção de um laboratório tem custos elevados, não só em equipamento mas também em pessoal técnico que os assista, de modo a permitir o acesso diário aos alunos. Devido às limitações orçamentais, muitas universidades e faculdades apenas permitem um acesso limitado a esses equipamentos físicos. Portanto, é imperativo permitir o acesso remoto ao laboratório físico.

Dado este facto, foi proposto no início dos anos noventa nos EUA a elaboração de um laboratório controlado remotamente e partilhado pelas universidades. Os autores do R-LAB [19] [20] [21] aceitaram o desafio e fizeram uma série de testes de actividades remotas.

À data o laboratório virtual tinha as funcionalidades:

- Permite a colaboração em tempo real
- Os equipamentos do laboratório podem ser acedidos e controlados remotamente
- Single Point Of Entry (SPOE) de acesso às ferramentas de controlo e programação dos equipamentos, que se faz via Web Browser com suporte para Java Applet
- A Java Applet é descarregada no portal do laboratório pelos utilizadores autorizados.
- O utilizador necessita de se autenticar para aceder ao laboratório, para isso existe uma base de dados dos utilizadores, com um nome de utilizador e uma senha. Depois de autenticado, o acesso interno R - Lab é concedido e o utilizador, tem a possibilidade de reservar os recursos para determinada data à sua escolha.
- Disponibiliza fundamentos teóricos para as questões levantadas no âmbito do exercício.
- Tem uma funcionalidade que permite dar dicas ao utilizador, em determinadas situações consideradas mais críticas para a conclusão do exercício.
- Permite aos utilizadores descarregar os resultados experimentais para futura análise e conclusões.
- Com a Applet Java é possível visualizar a topologia do laboratório. Cada elemento da rede é clicável e acessível, pelo que os alunos podem começar a configurar o equipamento após clicar sobre este.
- Firewall para garantir a segurança no acesso ao laboratório. Uma vez que o utilizador está autenticado, o seu IP é conhecido e apenas o tráfego gerado por este utilizador passa no filtro de endereço IP. Após ter expirado o tempo de exercício definido e alocado ao utilizador, o filtro de endereço IP será actualizado e passará a bloquear o acesso do utilizador ao sistema.
- Ferramenta de colaboração baseada em texto. É possível comunicar com outros estudantes e professores em tempo real.

É garantido ao utilizador a conectividade aos equipamentos, mesmo quando a sua configuração esteja incorrecta. Isto é possível dado que os equipamentos estão ligados via porta série, a um router dedicado para essa funcionalidade. Os utilizadores para acederem aos equipamentos, terão que iniciar uma sessão telnet a esse router numa determinada porta, que o indexa à consola de configuração do equipamento.

Cada vez que um aluno realiza uma operação num equipamento é proibido o acesso aos outros membros do grupo. Esses elementos do grupo apenas poderão ver a execução do companheiro ou realizar operações noutros equipamentos.

A ferramenta de colaboração baseada em texto, permite a troca de mensagens instantâneas (chat) e participação em fóruns de discussão possibilitando aos membros do grupo participar activamente na resolução da tarefa laboratorial.

Tem como limitação a utilização de Java que torna demorado o arranque da janela terminal, a falta de ferramentas de colaboração multimédia e a impossibilidade de corrigir problemas nas ligações físicas.

2.3.4.2 IV - LAB: InteractiveVirtual - LAB

Este segundo projecto [21] [22] data de 2005, dois anos após o término do R-LAB. Não se trata de um novo projecto de raiz, mas sim de um aumento das funcionalidades de acordo com as lacunas detectadas no R-LAB.

O Microsoft ConferenceXP permite que alunos dispersos em diferentes áreas geográficas, possam trabalhar juntos em grupos virtuais no laboratório e aprender efectivamente a usar o equipamento laboratorial. Esta ferramenta de colaboração permite a utilização não só de texto, mas voz e vídeo, maximizando o poder da interacção.

O mecanismo de colaboração do R-lab era suportado unicamente por mensagens de texto. Com esta nova ferramenta é possível criar um ambiente semelhante a uma turma, com interacção áudio e vídeo. É também certo que os alunos saem beneficiados com o seu envolvimento em pequenos grupos de trabalho e dizem-se mais motivados quando têm contacto frequente com o professor.

O ConferenceXP [23] oferece uma arquitectura flexível e extensível à infra - estrutura de banda larga, pelo que o seu uso é viável à grande parte dos alunos, dada a proliferação dos acessos à Internet em banda larga. Existe também a possibilidade de o pessoal técnico e alunos verificarem o estado do laboratório através de equipamento de áudio e vídeo aí instalado.

O Servidor ConferenceXP, permite arquivar informação de apoio aos alunos e suportar o mecanismo de colaboração áudio e vídeo, através de tráfego multicast para os clientes ConferenceXP. O ConferenceXP Apresentador, permite ao professor utilizar diapositivos para dar as suas aulas e todos os alunos recebem os slides no seu terminal em tempo real. Os alunos poderão interagir com o professor e questioná-lo sobre a matéria.

Assim o IV - Lab é nada mais nada menos que o R-LAB com funcionalidades acrescidas. O acesso ao equipamento e suas configurações, faz-se via Web Browser do mesmo modo que em R-LAB.

Todos os equipamentos estão ligados fisicamente ao Communication Server (Cisco 2511 Access Router), que está interligado ao Servidor ConferenceXP através de um hub Ethernet. O

Communication Server é utilizado para fornecer acesso à consola dos equipamento laboratorial, como em R-LAB.

Após o login, é invocado o software ConferenceXP Client que se liga à “sede da conferência” e possibilita a colaboração em tempo real, com áudio e vídeo. Os alunos que aderirem à conferência podem ver-se mutuamente.

Além do áudio e vídeo em tempo real, também é possível a colaboração baseada em texto, como mensagens instantâneas e fóruns de discussão, que também foram integrados no site do laboratório. Assim, todos os membros do grupo podem participar activamente na resolução das tarefas laboratoriais e beneficiar activamente do equipamento laboratorial.

2.3.5 O ambiente laboratorial virtual

A criação de ambientes laboratoriais virtuais, permite simular as funções essenciais de uma experiência laboratorial num computador. Neste cenário, as experiências laboratoriais são substituídas por modelos computacionais, com base em software de simulação [1]. Recentemente, emergiram na web muitos laboratórios deste tipo, em formatos Java, VRML (Virtual Reality Modeling Language) ou Shock-wave, que permitem emular o sistema operativo de equipamentos activos de rede, permitindo a simulação de experiências laboratoriais. O conhecimento apreendido pelo aluno num ambiente deste tipo, depende principalmente da autenticidade, dos constrangimentos e das capacidades do software de simulação. Estas restrições, impostas pelo software de simulação, limitam assim a criatividade do aluno, na medida em que o inibem de realizar experiências reais.

Esta possibilidade foi sempre abordada nos casos estudados, o que me leva a olhar para este tipo de solução como uma hipótese a ter em consideração na análise que farei à actual implementação do laboratório remoto de redes da Faculdade de Engenharia da Universidade do Porto.

2.4 O eCassiopeia e as barreiras à sua utilização na FEUP

O eCassiopeia, segundo contacto com ex-alunos da Faculdade de Engenharia da Universidade do Porto e alguns registos de blogs pessoais na Internet, chegou a estar em produção e a suportar a realização de exercícios das disciplinas de redes da instituição. No entanto durante a minha passagem como estudante pelo laboratório desde há três anos atrás, nunca este método de trabalho foi usado. Algumas conversas revelam que, desde a saída do seu promotor, Raul Oliveira, a ideia caiu em declínio pela falta de actualização de conteúdos, de tecnologia e de suporte para revisão de erros e falhas, que se foram entretanto detectadas. A saída sucessiva dos técnicos do laboratório que acompanharam a sua criação, deixou a Faculdade sem o conhecimento do sistema.

O disco da máquina que suportava o eCassiopeia sofreu problemas e nessa altura foi contratada uma pessoa com o objectivo de reconstruir o eCassiopeia, que concluiu não ser possível recuperar todas as suas funcionalidades.

Estes acontecimentos inviabilizaram a utilização pelos docentes da ferramenta web como, uma mais valia dos seus cursos e a reorganização das suas aulas centrou-se na utilização física do laboratório. Deixou por isso de haver pressão para o desenvolvimento e recuperação do eCassiopeia, razão pela qual nunca mais foi utilizado.

2.5 Conclusão

O estudo que fiz do laboratório eCassiopeia e dos restantes laboratórios enumerados neste capítulo, permitem-me definir caminhos de orientação para uma nova abordagem ao eCassiopeia e que este trabalho pretende discriminar no seu decorrer.

O estado de desenvolvimento dos restantes laboratórios, levam-me a reportar a necessidade de integrar ferramentas colaborativas no projecto, desde a possibilidade de comunicar em tempo real com utilização de texto, áudio e vídeo e comunicação assíncrona como fóruns e wiki's. Será necessário dotar o sistema de funcionalidades como salvaguardar as configurações efectuadas pelos alunos, possibilitando-lhes retomar o trabalho do ponto em que o terminaram aquando do último acesso ao laboratório.

Uma segunda necessidade, esta com base no laboratório da Escola Politécnica da Universidade de São Paulo e no ReLI, é a implementação de um mecanismo de reconfiguração dinâmica das configurações, não apenas para correcção, mas também para definir a topologia de trabalho para cada exercício.

Por último integrar no sistema a possibilidade de implementar virtualmente um dado exercício, permitindo aos alunos praticar e ajustar as configurações do exercício, sem que para isso ocupem o equipamento real do laboratório. Assim, apenas passarão à recolha de resultados depois de terem as configurações ajustadas aos objectivos do exercício, minimizando a ocupação remota do laboratório.

Capítulo 3

3. Solução eCassiopeia existente

Este capítulo discrimina o levantamento efectuado do sistema eCassiopeia de acordo com o sistema implementado no laboratório de redes da Faculdade de Engenharia da Universidade do Porto.

3.1 Arquitectura Funcional

A figura 3.1 representa a arquitectura física que encontrei no laboratório. O acesso remoto ao laboratório era suportado por três máquinas.

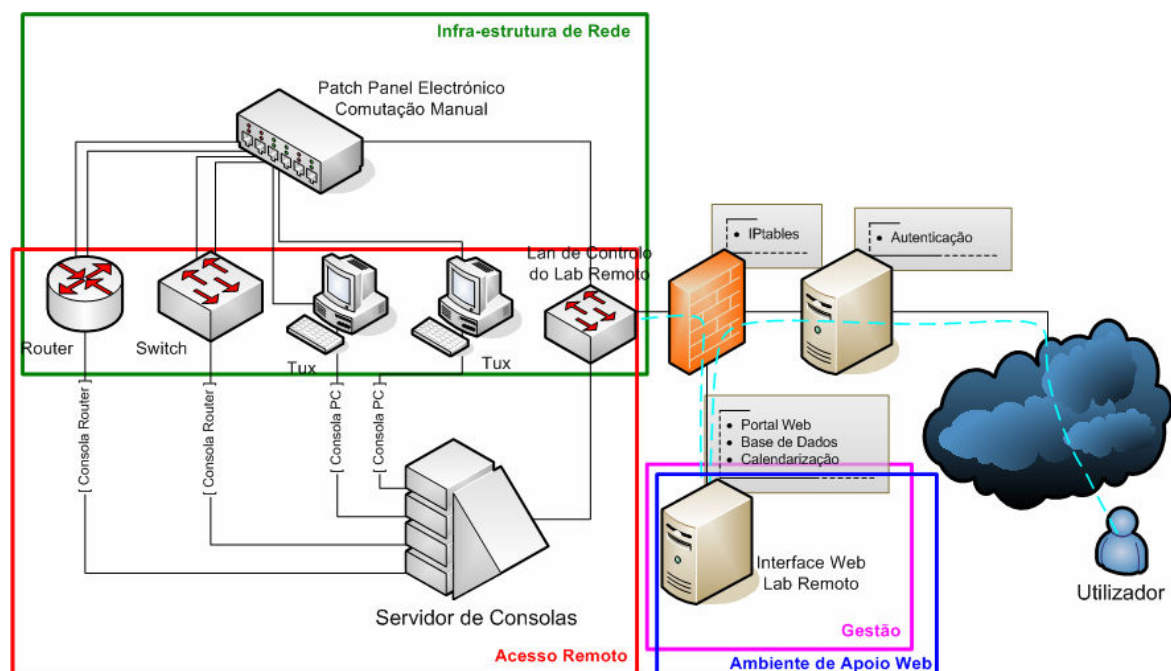


Figura 3.1 - Arquitectura da Solução eCassiopeia

Existia uma máquina acessível directamente da Internet, cuja função era permitir ao utilizador remoto autenticar-se no sistema para aceder aos serviços. Uma outra máquina tinha a única função de firewall, com o iptables configurado, que permitia apenas transitar tráfego

vindo da máquina acessível da Internet. A terceira máquina implementava o portal de interface com o laboratório.

No caso da autenticação ser válida, era adicionada nova permissão na firewall que permitiria o tráfego com origem na máquina do utilizador remoto. O tráfego permitido era apenas o destinado à interface web. O portal por sua vez fazia pedidos à rede privada de gestão do laboratório, de modo a executar as instruções que o cliente dava por browser.

Assim os pedidos eram sempre realizados no portal, e a rede só respondia a pedidos vindos do portal de acordo com a figura 3.2.

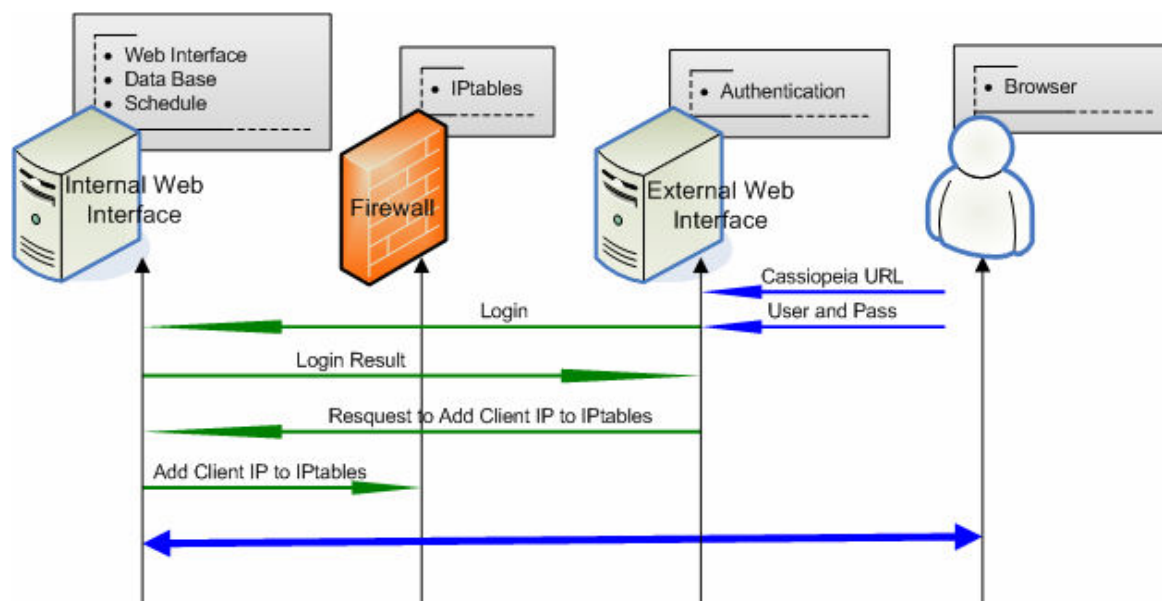


Figura 3.2 - Fluxo de pedidos na Solução eCassiopeia existente

A arquitectura apresentada era suportada por quatro módulos funcionais que se salientam na figura 3.1:

- **Infra-estrutura de rede** - representa a rede que suporta o laboratório remoto e os equipamentos de ensino. Existem dois tipos de rede: a rede de ensino e a rede de produção. A rede de ensino interliga os equipamentos de ensino onde são feitas as experiências e a rede de produção faz a ligação entre os equipamentos que dão suporte ao laboratório.
- **Gestão** - é o módulo centralizador da actividade de gestão do laboratório. Caracteriza-se por um servidor http, que disponibiliza uma interface de gestão, baseada em formulários html, processa os dados recebidos dos utilizadores, sob a forma de scripts PHP e executa as alterações pedidas nos sistemas envolvidos.
- **Acesso remoto** - facilita o acesso remoto aos equipamentos e à rede de ensino, através de sessões de telnet directamente para as portas das consolas.
- **Ambiente de apoio web** - Constitui o ponto de acesso, via browser, dos utilizadores ao laboratório e, consequentemente, aos equipamentos de ensino. Este módulo é fisicamente suportado na mesma máquina que suporta o módulo de gestão.

A figura 3.3 apresenta a arquitectura do laboratório remoto eCassiopeia, mostrando as relações entre os vários módulos funcionais e os protocolos de comunicação utilizados.

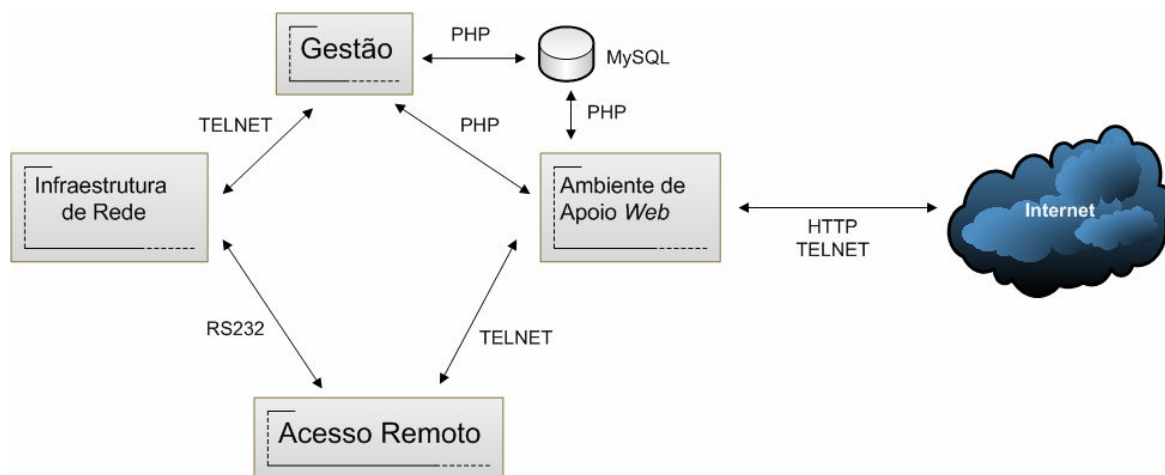


Figura 3.3 - Módulos Funcionais eCassiopeia

3.1.1 Infra-estrutura da rede do laboratório

O laboratório de redes local Cassiopeia, é constituído essencialmente por seis postos de trabalho. Cada um destes é formado por três computadores com sistema operativo Linux e por um bastidor com equipamentos próprios. Cada bastidor inclui um switch, um HUB e painéis de repartição. Existem ainda um router/switch para protocolos de lan e um outro router que suporta protocolos de wan em três dos seis bastidores que são partilhados por duas bancadas, durante a realização de exercícios.

Uma das máquinas Linux, com denominação Tux, fisicamente possui duas interfaces ethernet. Os exercícios laboratoriais têm em consideração este facto, pelo que frequentemente era necessário utilizar essa máquina nos exercícios que se pretendiam fazer. No entanto, o sistema considerava os Tux com uma única interface ethernet, pelo que não era possível realizar nenhum exercício que necessitasse da máquina Tux com duas interfaces ethernet.

3.1.2 Endereçamento do Laboratório eCassiopeia

O laboratório remoto eCassiopeia é suportado por duas redes. As estações de trabalho e todo o material activo da rede encontram-se na rede de ensino 172.16.1.0/24, enquanto que as estações de trabalho que dão suporte à gestão do laboratório e o material da rede de produção estão na rede de produção 172.16.100.0/24. Estas redes estão logicamente separadas pelo que não é possível trocar tráfego entre as duas. Esta abordagem é justificada pela imposição de segurança, garantindo que não é possível entrar na rede de gestão do laboratório.

3.2 Ligação ao exterior

3.2.1 A Firewall

Uma firewall é um equipamento utilizado para implementar políticas de segurança em redes de computadores. Na sua forma mais simples, uma firewall é um computador colocado na fronteira de uma rede, cujo principal objectivo é controlar o acesso a essa rede, eliminando o tráfego não desejado [2]. Os firewalls podem mediar o acesso de uma rede privada à Internet e vice-versa, ou então poderão ser usados para mediar o acesso entre duas redes privadas entre si. Por exemplo mediar o acesso da rede de trabalhadores à rede de administradores numa empresa.

3.2.1.1 Tipos de firewalls

Existem três tipos de firewalls divididos nas seguintes categorias:

- Filtro de pacotes (Packet filters)
- Servidores de proxy (Proxies de aplicações e proxies de rede)
- Firewalls baseados no estado (Stateful inspection)

A firewall usada no eCassiopeia é uma firewall de filtro de pacotes que decide o encaminhamento com base nos endereços IP de origem e de destino, no protocolo utilizado, nos portos TCP/UDP e nas interfaces. As listas de controlo de acessos (Access Control Lists) podem ser configuradas para permitir um ou mais serviços em particular, tal como http, telnet e SSH.

3.2.2 O acesso ao laboratório via web

Os próximos parágrafos descrevem a forma como o laboratório se ligava à rede do campus da FEUP (FEUPNet) e à Internet [24]. O laboratório estava isolado da Internet, sendo os acessos controlados por um sistema de firewall. Assim a arquitectura de acesso baseava-se numa DMZ combinada com uma Firewall. O servidor externo permitia a autenticação de utilizadores e obtinha o seu IP, assim se a autenticação fosse válida era adicionada uma nova regra na firewall permitindo o tráfego com origem no IP do cliente e destino à DMZ. A firewall tinha a capacidade de analisar os endereços IP de origem e destino, o protocolo e portos utilizados. Só o tráfego com base no porto http do servidor externo era permitido inicialmente, após a autenticação, era possível o trânsito de outros protocolos. A configuração contemplava uma zona desmilitarizada (Demilitarized Zone - DMZ), onde estava colocado o servidor HTTP interno de acesso público ao laboratório, actuando a Firewall como encaminhador de tráfego de e para essa máquina.

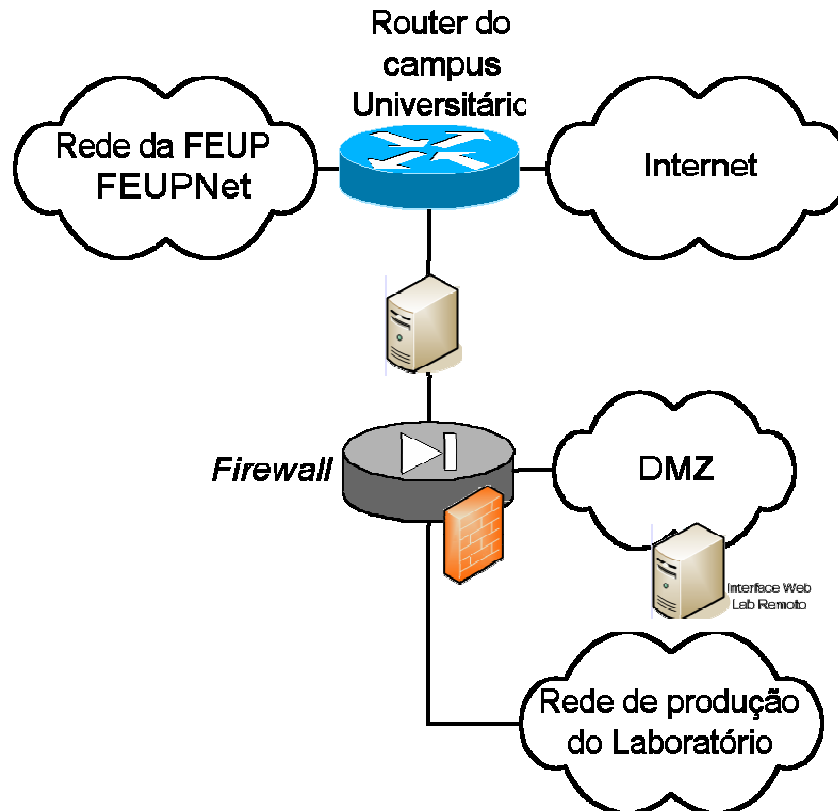


Figura 3.4 - Acesso do Laboratório à Internet

Na rede DMZ, ficava o servidor http de acesso ao laboratório, e apenas este servidor podia aceder aos recursos internos da rede do laboratório. Os utilizadores externos, após se terem autenticado nesse servidor, poderiam então aceder ao servidor de consolas do laboratório e posteriormente aos recursos de hardware.

O firewall possuía três interfaces, uma para acesso à Internet através da rede FEUPNet, outra para acesso à rede de produção do laboratório e uma outra onde comunica com a DMZ.

3.2.3 Controlo de acessos

A segurança era uma preocupação do sistema implementado, dado estar acessível através da Internet.

Uma vez identificado o utilizador, pelo seu login e password, o mecanismo de controlo de acessos detectava qual o IP com que o utilizador se apresentava no sistema e inseria uma regra no iptables que pretendia permitir o tráfego com origem nesse IP. Este mecanismo era implementado recorrendo ao firewall representado na Figura 3.4 como foi descrito.

A firewall foi implementada recorrendo a uma máquina linux onde foi configurado o filtro de pacotes iptables.

3.2.3.1 Filtro de pacotes: iptables

Todo o tráfego que atravessa uma rede é enviado sob a forma de pacotes. Um filtro de pacotes, é um software que inspeciona o cabeçalho dos pacotes, à medida que estes passam, e decide o destino a dar-lhes. Pode decidir fazer o DROP do pacote (isto é, descarta o pacote como se nunca tivesse sido recebido), como pode fazer o ACCEPT do pacote (isto é, deixa passar o pacote), ou ainda fazer o REJECT do pacote (idêntico ao DROP, mas notifica o nó de origem de que o pacote foi rejeitado). No Linux, o filtro de pacotes é implementado ao nível do kernel. A partir da versão 2.3.15 do kernel, o filtro de pacotes passou a ser o iptables [3] em substituição do ipchains.

O mecanismo de controlo de acessos representado na Figura 3.4, utilizava o firewall iptables que, com base na definição dinâmica de regras, aceitava ou rejeitava o acesso telnet e SSH ao servidor de terminais.

As regras eram inseridas e removidas dinamicamente no iptables pelo serviço Labkeeper (programa desenvolvido nas versões anteriores do eCassiopeia) de modo a abrir ou fechar os portos de telnet e SSH, controlando o acesso a partir do exterior ao servidor de terminais. Note-se que estas regras eram voláteis sendo inserida uma por cada utilizador ligado ao laboratório, variando apenas o endereço IP remoto.

A regra aceitava tráfego com origem no porto 80 do servidor http que se encontrava na rede DMZ e se destinava ao serviço Labkeeper (porto 3123). Estavam também implementadas regras que impediam uma das técnicas de ataque a firewalls conhecida: o ip-spoofing. Este tipo de ataque consiste em enviar pacotes para outro sistema, simulando que os mesmos são provenientes de um sistema autorizado [4].

Depois da regra que pretendia eliminar o spoofing, era aceite todo o tráfego que tinha origem na rede de produção do laboratório e no próprio firewall.

Para que o firewall permitisse a comunicação da rede interna para o exterior, existia uma regra que deixava passar os pacotes TCP de resposta vindos do exterior e destinados à rede do laboratório nos portos 1025-65534. Eram permitidos também pacotes UDP provenientes de servidores DNS externos e pacotes ICMP, para respostas ao ping e mensagens de erro Destination Unreachable. Todo o restante tráfego era bloqueado.

3.3 Manipulação das topologias de rede

Numa rede de computadores, os painéis de repartição ou de patching seriam dispensáveis se cada tomada tivesse sempre o mesmo serviço e a mesma utilização. Porém, na prática, todas as redes estão sujeitas a alterações, a movimentações de pessoas, ou ainda à evolução dos sistemas de comunicação. No caso do laboratório de redes, essas alterações ainda são mais frequentes, porque cada experiência ou exercício diferente que é realizado, envolve normalmente alterações na topologia da rede, uma vez que é indispensável o uso de equipamentos e topologias para a realização dos exercícios laboratoriais. Estas topologias podem-se apresentar de forma fixa (definidas previamente), ou dinâmica (definidas a pedido).

Eram utilizadas topologias fixas, em que a segmentação física da rede era efectuada previamente pelo assistente do laboratório. Este, alterava as ligações nos painéis de repartição, de acordo com os objectivos dos exercícios marcados na agenda do laboratório.

Dada esta limitação a realização de exercícios estava condicionada ao horário semanal útil, definido entre as 8h e as 23h. Este horário, justificava-se pelo facto de não ser possível configurar a arquitectura do exercício de forma automática.

Apesar da implementação, física utilizar os patch panel electrónicos, estes eram comutados manualmente. Na realidade, não existia o software de controlo do patch panel pelo que nunca tinha sido possível configurá-lo de modo electrónico. Assim, eram agendados os exercícios no portal para determinada hora, e o técnico responsável pela manutenção do laboratório deslocava-se à sala, para configurar manualmente a interligação dos portos do patch panel.

Para permitir topologias a pedido, era necessário implementar o software de controlo ou contactar a Apcon para que o disponibilize, dado não existir qualquer suporte de armazenamento no laboratório com esta aplicação.

Com efeito sendo matrizes de comutadores, possibilitam a comutação remota do equipamento de ensino de um circuito para outro, dando liberdade aos utilizadores do laboratório para manipular a topologia física da rede de ensino, adicionando e retirando equipamentos ou estações de trabalho, de forma automática.

Com esta solução, o utilizador consegue alterar a pedido a topologia física da rede de ensino, podendo preparar remotamente o seu ambiente de teste. Neste cenário, a rede de ensino é então segmentada através do sistema de painéis de repartição automáticos.

3.4 Autenticação dos utilizadores

Para se garantir que apenas as pessoas autorizadas podiam aceder aos recursos do laboratório, o serviço de autenticação obrigava os utilizadores a se autenticarem verificando se a origem dos dados recebidos coincidia com o computador remoto que esse utilizador dizia estar a usar. Este serviço aumentava a garantia de que não se teria ataques por imitação, spoofing attacks.

A autenticação dos utilizadores era feita na aplicação web que corria no servidor http exterior, usando o par login e password. Ao utilizador era fornecido um login e password que deveria usar para se autenticar na aplicação do laboratório.

A segurança no controlo do acesso aos equipamentos do laboratório era assegurada pelo iptables. Assim, as sessões de telnet ou SSH só seriam aceites depois do utilizador se ter autenticado no servidor http no entanto existia um problema de implementação e nunca era apagada essa entrada no iptables, logo sempre que um utilizador se ligava ao sistema o seu IP ficaria para sempre registado e permitido no sistema.

3.5 Servidor Web apache

O servidor Apache tinha uma função dupla: a de comunicar com os clientes através da Internet e a de interagir com os outros componentes da gestão do laboratório. O servidor estava implementado sobre o sistema operativo Linux a correr o Apache [5] utilizando a linguagem web scripting PHP e Perl.

3.5.1 Módulo Autenticação

O módulo de autenticação era responsável pela credenciação dos utilizadores e permitia ao administrador do laboratório, manter informação relativa às contas dos utilizadores e gerir os acessos. O mecanismo de autenticação interagia com a base de dados, que permitia consultar e validar a credenciação do utilizador.

3.5.2 Agenda

A agenda permitia gerir a utilização do laboratório e efectuar a reserva de recursos. Este serviço servia-se da base de dados do laboratório para manter os dados.

Quando os alunos recebiam autorização do professor para realizar determinada experiência ou exercício, tinham que previamente reservar na agenda os recursos necessários. A interface da agenda consistia num calendário que, para uma determinada semana seleccionada, mostrava os períodos disponíveis e ocupados. O aluno poderia reservar a hora mais conveniente, dentro dos períodos não reservados. Estes períodos eram monitorizados e não ficariam disponíveis para marcações posteriores. As marcações efectuadas eram mantidas na base de dados e essa informação era usada pelo controlo de acessos, para que, na hora marcada, fosse concedido ao utilizador o acesso exclusivo aos equipamentos reservados para a experiência ou para o exercício.

3.5.3 Aplicação Web eCassiopeia

O ambiente de apoio web foi implementado através de uma aplicação web, que designaram de Cassiopeia@FEUP [24]. Esta aplicação funcionava no servidor Apache e incluía um conjunto de sub-componentes, implementados em PHP, que processavam os pedidos dos utilizadores. O acesso dos utilizadores ocorria através de ligações http para o servidor apache, que interagia com o código PHP, que, por sua vez, fazia o acesso aos dados armazenados na base de dados Mysql. O código PHP estava embutido em páginas HTML e era processado pelo servidor apache, não sendo, no entanto, visível pelo utilizador. A aplicação permitia três níveis de acesso distintos, de acordo com o tipo de utilizador (professor, aluno e administrador). As funções acessíveis aos utilizadores variavam consoante o seu tipo de acesso.

Esta aplicação permitia aos alunos acederem ao laboratório a qualquer hora, a partir de qualquer lugar, utilizando apenas o browser, apesar de algumas limitações é certo.

3.5.3.1 Registo de Utilizadores

Para aceder ao laboratório [24], era necessário um registo prévio, realizado através de um formulário que registava os utilizadores numa tabela da base de dados. Para além dos dados pessoais do utilizador, era registado também o tipo de utilizador, uma vez que existiam três tipos com funcionalidades distintas: o administrador, os professores e os alunos. O primeiro tinha uma password que lhe permitia aceder a um módulo da aplicação para administração. Neste módulo, o administrador podia fazer a gestão, parametrização e manutenção do laboratório, incluindo a manutenção do cadastro de professores. Tendo os dados pessoais dos docentes, realizava esta tarefa através da opção "Novo professor". Devia ainda introduzir os dados relativos à instituição, definir um login/password e, posteriormente, enviar essa informação ao professor.

O professor, depois de fazer o login, tinha acesso a uma página de administração com opções para manutenção dos cadastros dos alunos, das disciplinas, dos grupos e uma área para alteração de dados pessoais. Este tinha ainda privilégios que lhe permitiam criar novos exercícios.

Outro tipo de utilizador é o aluno, que da primeira vez que acedia ao URL do laboratório, registava o seu cadastro através da opção "Novo aluno", inserindo os dados pessoais, disciplina e grupo a que pretendia pertencer. Depois, aguardava que lhe fosse enviado um e-mail de resposta com a password para acesso ao laboratório. Após o auto-registo, a aplicação gerava uma password aleatória e enviava automaticamente um e-mail para o professor, contendo os dados principais do aluno e a disciplina a que se associou. Esta informação também era armazenada na base de dados. No caso de o professor aceitar a inscrição do aluno para aquela disciplina, informava o administrador para tornar o utilizador activo, e este agia de acordo a que as credenciais fossem correctamente validas no sistema informando o utilizador da sua password.

Esta solução tinha a vantagem de os alunos poderem fazer o auto-registo. Contudo, eles só acediam efectivamente ao laboratório, depois do professor os autorizar. Para evitar que o administrador ficasse a saber a password do aluno, este deveria alterá-la da primeira vez que efectuasse o login no laboratório.

3.5.3.2 Interacção dos alunos com a Aplicação

O aluno depois de se autenticar, deveria seleccionar a disciplina, isto no caso de estar inscrito a várias. No caso de estar inscrito apenas a uma disciplina, esta era assumida automaticamente.

De seguida, a interface web apresentava ao aluno uma lista com os exercícios existentes para a disciplina seleccionada. Os exercícios poderiam estar activos ou inactivos, dependendo da época do semestre. Quando o aluno seleccionava um exercício activo, tinha duas opções: tentar iniciar o exercício de imediato ou reservar na agenda do laboratório um dia e uma hora para o realizar. Na primeira opção, o sistema verificava na agenda se existiam recursos disponíveis para suportar a realização imediata daquele exercício. Se realmente existissem, o exercício era reservado automaticamente na agenda, durante o tempo estimado para a sua realização. Se, por outro lado, não existissem recursos disponíveis para começar o exercício

naquele momento, o aluno tinha a possibilidade de reservar um dia e uma hora na agenda do laboratório, para a realização do exercício. Em ambos os casos, o aluno só poderia começar o exercício depois deste ter sido reservado na agenda e só tinha acesso aos equipamentos, durante o período reservado. Uma vez iniciado, o exercício não poderia ser interrompido.

Quando o aluno iniciava um exercício, a aplicação interagiu com o controlo de acessos, adicionando uma regra no iptables, com o fim de permitir o acesso via telnet ao servidor de terminais. Depois destas operações estarem concluídas, a interface do utilizador mostrava a página do exercício que tinha as hiperligações para vários recursos, nomeadamente: o enunciado do exercício, com uma descrição dos objectivos e das tarefas propostas, uma imagem com a topologia de rede disponível para o exercício (consola do exercício) e hiperligações para outra documentação on-line, que fosse considerada relevante para o exercício.

Depois de ler o enunciado, o aluno podia entrar na consola do exercício e começar o seu trabalho. A imagem com a topologia da rede, continha ícones activos, que quando activados com um clique do rato, abriam uma ligação por telnet para o equipamento respectivo. Poderiam existir também hiperligações para outra documentação

O aluno tinha um tempo limitado para completar o exercício, que começava a decrementar a partir do momento que o iniciava. O tempo disponível era previamente definido pelo professor, de acordo com o grau de dificuldade do exercício criado. Quando o tempo estimado para o exercício terminava, era pretendido que o mecanismo de controlo de acessos cortasse a conexão do aluno com as consolas dos equipamentos, sendo este obrigado a terminar o exercício, no entanto fruto dos problemas que o eCassiopeia foi sujeito, esta funcionalidade não estava a trabalhar correctamente, pelo que o aluno ficaria sempre com o seu acesso registado no iptables. Esta metodologia pretendia garantir que os equipamentos ficassem livres para outros alunos iniciarem também os seus trabalhos.

3.5.3.3 Metodologia para a realização dos exercícios

Para realizar os exercícios, os alunos acediam ao laboratório utilizando um web browser. Depois de passarem pelo processo de autenticação, através de login e password, chegavam à página dos exercícios, onde seleccionavam o exercício pretendido e finalmente teriam acesso à página HTML do exercício, que era construída dinamicamente pela aplicação. Esta página continha os objectivos, a introdução teórica, as actividades do exercício, bem como uma imagem com ícones activos da topologia da rede para o exercício. Ao fazerem um clique sobre uma área activa da imagem (por exemplo, sobre um router), o utilizador ganhava acesso, via servidor de terminais, à CLI desse router. Verifica-se no entanto, que os browsers não têm definida uma acção directa para um link telnet pelo que apenas esta funcionalidade estava disponível no Konqueror, browser da interface gráfica KDE para o sistema operativo Linux.

Utilizando a CLI os alunos configuravam as interfaces de rede, atribuíam os endereços IP, definiam as rotas de encaminhamento, activavam um protocolo de encaminhamento, gravavam a configuração e finalmente verificavam o funcionamento das configurações efectuadas, fazendo testes de conectividade, usando os comandos ping e traceroute. Todos os exercícios seguiam a mesma sequência, variando essencialmente o equipamento, a topologia e as actividades a realizar.

Capítulo 4

4. Melhorias à Solução Existente

Após a reflexão a que procedi durante o estudo descrito no capítulo anterior, encontrei algumas limitações que pretendia resolver no percurso deste trabalho de mestrado. Dado que o tempo foi limitado para o trabalho que era necessário fazer, efectuei algumas correcções à solução existente que achei de base, o mais importante e de maior urgência.

Assim, este capítulo pretende salientar esse trabalho de implementação efectuado.

4.1 Arquitectura Funcional

Considerando as actuais capacidades de processamento dos computadores e conferenciando com o técnico do laboratório onde se encontra o eCassiopeia, chegamos à conclusão que não faria sentido a utilização de três máquinas para aquela arquitectura. Assim decidimos alterar a arquitectura descrita no capítulo 3, preservando todos os serviços e funcionalidades que o trabalho anterior suportava, utilizando apenas uma máquina.

A nova máquina possui duas interfaces de rede, uma pública, acessível directamente através da Internet com IP público na gama de endereços atribuída à FEUP, e uma privada com IP na rede de produção do laboratório, com endereço na gama 172.16.100.0/24.

Foi configurado no seu interior o iptables para actuar do mesmo modo que acontecia com a solução anterior, pelo que a máquina só responde a tráfego com destino à porta 80. No caso de o aluno pretender realizar um exercício e estar autenticado, o seu IP é adicionado à tabela de permissões de tráfego, permitindo que o aluno aceda remotamente, via telnet, às máquinas do laboratório.

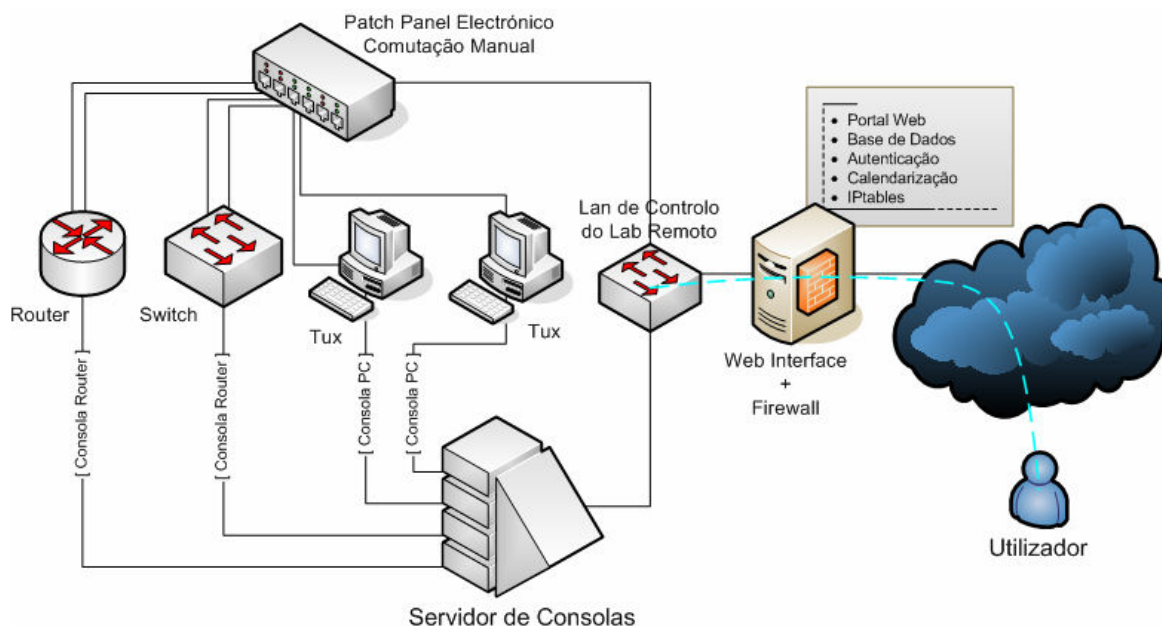


Figura 4.1 - Implementada na nova revisão do eCassiopeia

A nova arquitectura não introduziu alterações, do ponto de vista lógico do fluxo de mensagens, apenas o tratamento dos procedimentos necessários passou a ser realizado internamente de acordo com a figura 4.2.

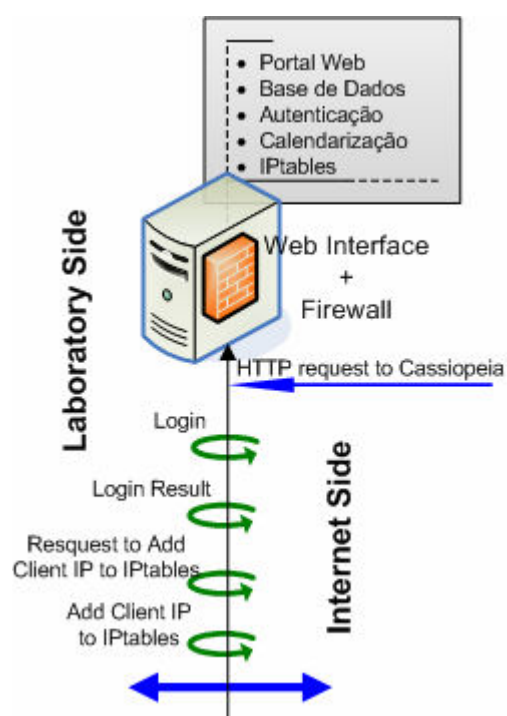


Figura 4.2 - Fluxo de pedidos na Solução Implementada na nova revisão do eCassiopeia

Esta nova máquina foi configurada com as seguintes ferramentas:

4.1.1 Sistema Operativo

O primeiro passo na instalação e configuração do servidor, foi sem dúvida a instalação do sistema operativo.

Com o objectivo de aproveitar algum do trabalho anteriormente feito, porque os sistemas Linux, baseados em Unix, são tendencialmente mais eficientes a nível de processamento e evitando a necessidade de licenciar software, optou-se por instalar o Linux no servidor. A distribuição utilizada foi o Debian etch, pelo facto de ter algum conhecimento com esta distribuição.

4.1.2 O Servidor Apache HTTP

Era necessário instalar um interpretador de http no servidor, para suportar a interface web do eCassiopeia. Deste modo, instalei o servidor HTTP da Apache (Apache Server) [5] dado ser o servidor web livre mais bem sucedido. Foi criado em 1995 por Rob McCool, então funcionário do NCSA (National Center for Supercomputing Applications), Universidade de Illinois. Na última pesquisa efectuada pelo site www.netcraft.com, em Dezembro de 2005, foi comunicado que a utilização do Apache supera 60% de servidores activos no mundo, dado ser uma aplicação de código fonte aberto. Descrevem-se de seguida os comandos usados para configurar e instalar este serviço cuja versão é 2.0.55:

```
$ ./configure --prefix=/wwwroot --enable-so
$ make
$ make install
$ /wwwroot/bin/apachectl start
```

Depois de instalado e iniciado o serviço como se descreve na última linha acima, poder-se-á verificar o seu funcionamento:

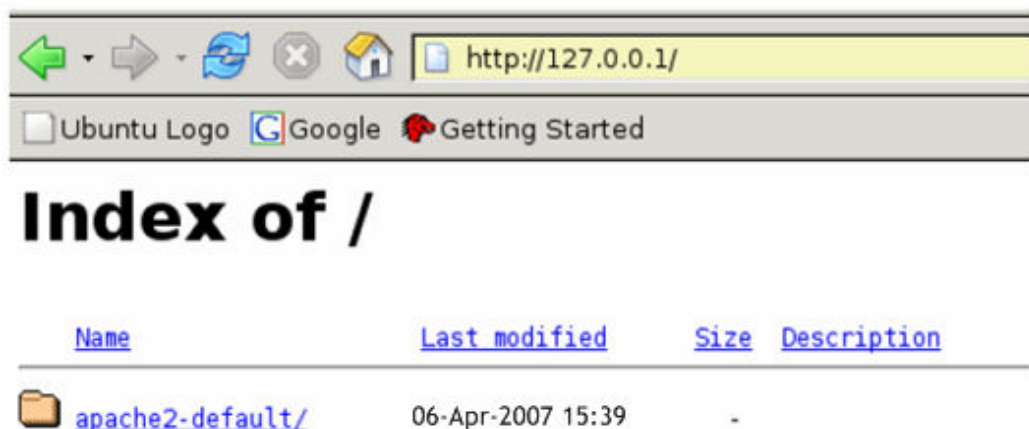


Figura 4.3 - Servidor HTTP

4.1.3 O PHP

O suporte à interactividade do utilizador com o laboratório é implementado recorrendo a scripts de PHP. O servidor Apache HTTP não implementa por si só a interpretação desta linguagem.

PHP é um acrónimo recursivo para "PHP: Hypertext Preprocessor", caracterizado por uma linguagem de programação de computadores interpretada, livre e muito utilizada para gerar conteúdo dinâmico na Web. Apesar de ser uma linguagem de fácil aprendizagem e utilização em scripts dinâmicos, o PHP é uma linguagem poderosa, orientada a objectos cuja sintaxe é similar à linguagem C/C++ e PERL. A justificação para o uso do PHP, ao invés de outras linguagens com objectivos idênticos, como Java Applets, deveu-se a experiências passadas com o PHP e C/C++.

Foi necessário, então, adicionar o módulo de processamento PHP ao servidor de Apache http, cuja configuração e instalação se apresenta:

```
$ ./configure --prefix=/wwwroot/php --with-apxs2=/wwwroot/bin/apxs --  
with-config-file-path=/wwwroot/php --with-pgsql  
$ make  
$ make install
```

Foram também adicionadas as seguintes linhas ao ficheiro `/var/www/conf/httpd.conf`:

```
(...)  
LoadModule php5_module modules/libphp5.so  
AddType application/x-httpd-php .php  
(...)
```

```
$ /wwwroot/bin/apachectl restart
```

4.1.4 Base de Dados MySQL

O MySQL [6] é um sistema de gestão de base de dados (SGBD), inicialmente desenvolvido pela MySQL AB, empresa com cerca de 70 programadores.

Recentemente o projecto foi adquirido pela Sun Microsystems, num negócio que envolveu o maior montante até agora pago no sector das OpenSource.

O sucesso desta Base de Dados é comprovado pelos 11 milhões de sistemas que o usam em todo o mundo. Empresas como a NASA, Banco Bradesco, HP, Nokia, Sony, U.S Army, US. Federal Reserve Bank, Alcatel e Cisco Systems acreditaram no seu sucesso, equipando os seus sistemas com este tipo de base de dados.

O MySQL tornou-se popular devido à compatibilidade com qualquer sistema operativo e por ser livre.

A base de dados suportará informação sobre alunos, equipamentos, exercícios, topologias de rede. No fundo será o ponto de interacção de todos os módulos que compõem o sistema.

A instalação desta ferramenta, a qual já está configurada para trabalhar com o PHP, foi conseguida com o comando:

```
$ apt-get install php5-mysql
```


4.1.5 O phpMyAdmin

O phpMyAdmin [7] traduz-se como uma interface gráfica para a gestão da base de dados, remotamente, via protocolo HTTP e apresentação dinâmica via PHP. Deste modo é possível realizar todas as funções sobre a base de dados de maneira intuitiva. A instalação e configuração deste serviço implicam o uso dos seguintes comandos:

```
$ apt-get install phpmyadmin  
$ ln -s /usr/share/phpmyadmin /var/www/
```

A aparência gráfica que esta ferramenta apresenta poder-se-á ver na imagem seguinte:

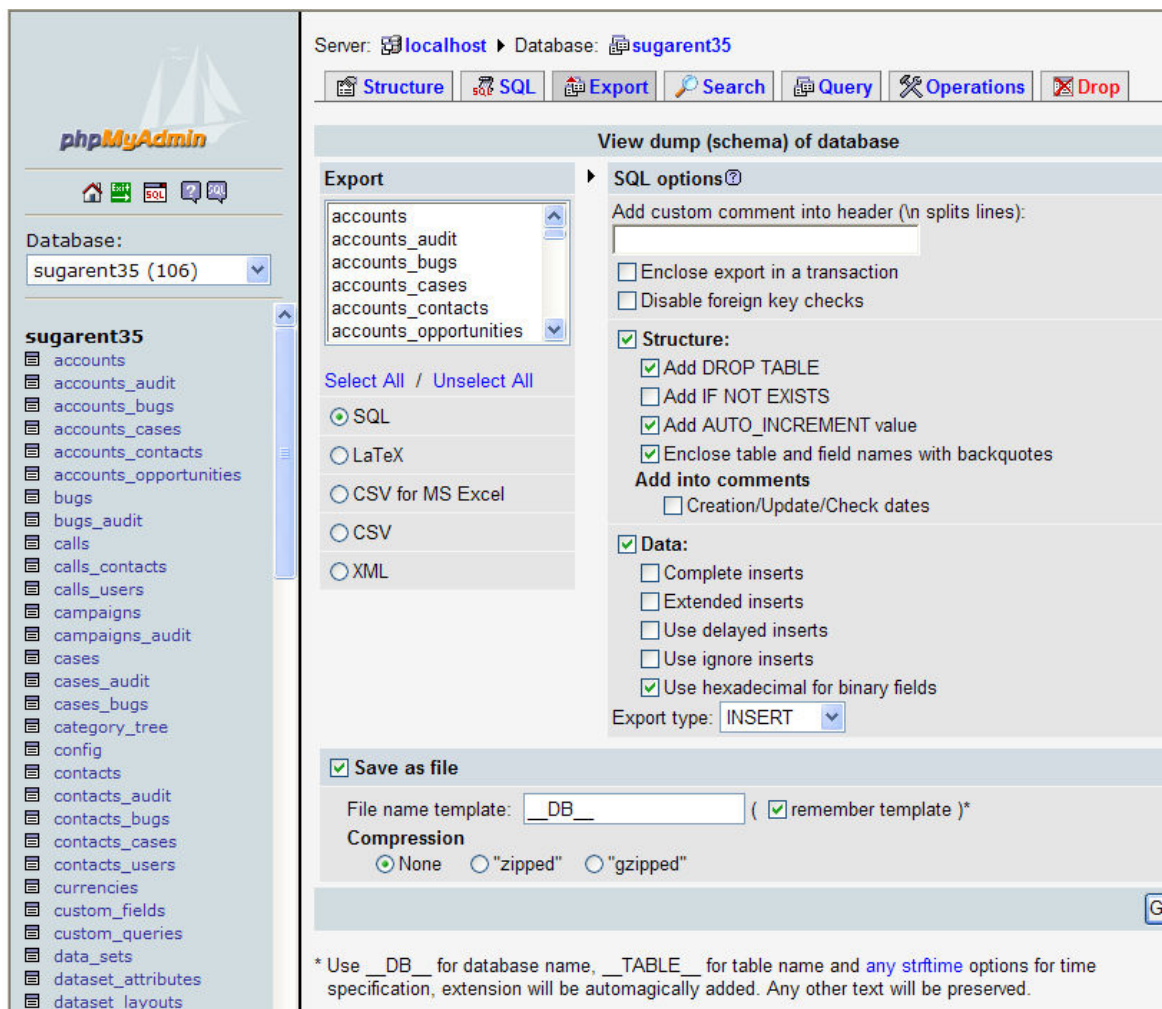


Figura 4.4 - Aparência gráfica do phpMyAdmin

4.2 Interacção Utilizador/Consola de equipamento

Com vista a resolver a limitação existente, no tratamento do acesso telnet à consola dos equipamentos, pelo browser, já descrita na secção 3.5.3.3, adicionei ao sistema uma applet em Java que permite executar ligações telnet a uma máquina. Isto permite que o utilizador aceda, com um clique sobre o equipamento, à consola do mesmo. É mandatório, no entanto, que o utilizador tenha instalado um interpretador de Java, libertando-o da necessidade de possuir um qualquer programa para executar telnet.

4.2.1 A Consola Telnet

O programa que suporta esta funcionalidade é um programa opensource, conhecido como JTA - Telnet/SSH for the JAVA(tm) platform [8]. JTA são as iniciais de Java Terminal Applet.

Esta aplicação implementa em Java, os protocolos de telnet e ssh. O programa é executado, do lado do cliente, no seu browser e para isso é necessário possuir um interpretador de Java.

Após o cliente clicar num equipamento capaz de ser acedido por telnet, o browser descarrega o executável e as configurações associadas a esse equipamento e, de modo automático, o programa estabelece a ligação ao equipamento.

As configurações para acessos a cada equipamento são executadas implicitamente, pelo que o cliente não poderá usar a aplicação para se ligar a qualquer outra máquina.

É necessário então possuir um ficheiro html, por cada equipamento do laboratório que seja acedido remotamente. Este ficheiro faz associação do executável com as configurações que serão carregadas. No caso do equipamento tux11, teremos o ficheiro tux11.html:

```
<html>
<body>
  <applet CODEBASE="."
    ARCHIVE="jta26.jar" <!-- Executável -->
    CODE="de.mud.jta.Applet"
    WIDTH=590 HEIGHT=360>
    <!--Configurações para tux11 -->
    <param name="config" value="applet_tux11.conf">
  </applet>
</body>
</html>
```

De seguida apresento também o ficheiro das configurações no caso particular do tux11, semelhante no entanto a todos os outros equipamentos. O que varia de equipamento para equipamento é o porto público pelo qual é acedido e que terá que respeitar a seguinte configuração

```
(...)
plugins          =          Status,Socket,Telnet,ButtonBar(1),ButtonBar(2),Terminal

# connection target configuration
Socket.host       =          193.136.29.90
Socket.port       =          7001
(...)
```

Depois de configurada esta função, o equipamento terá uma hiperligação para o ficheiro html que lhe está associado. Ao clicar no equipamento, abrirá um popup em Javascript que lançará a consola como se verifica na figura 4.5.

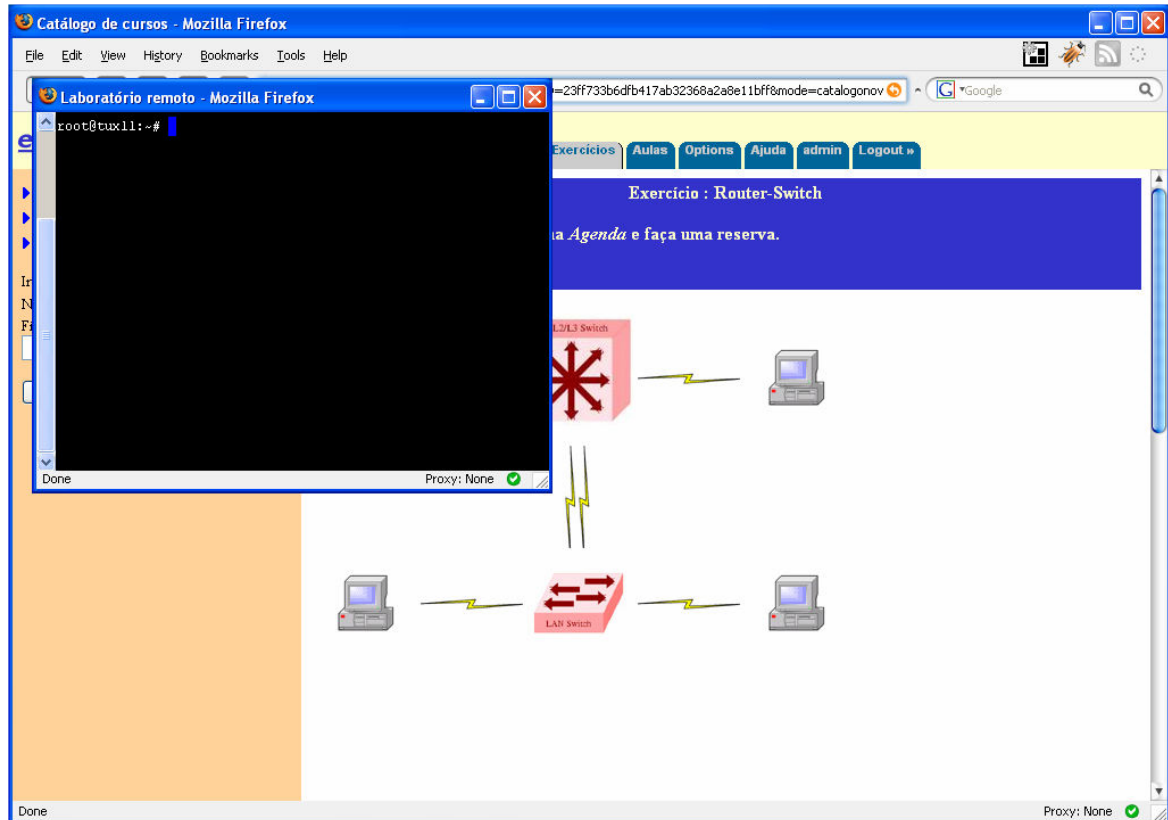


Figura 4.5 - Consola de equipamento lançada através de clique com rato

4.3 Painéis de repartição automáticos

A dificuldade fundamental para a aplicação, sem restrições do conceito de laboratório remoto, prendia-se de facto com a necessidade de existir um técnico no laboratório, para realizar as configurações da arquitectura do exercício que se pretendia executar. Era necessário assim colmatar essa falha fundamental.

Com a introdução no laboratório do sistema de painéis de repartição automáticos, conseguir-se-ia, de facto construir topologias de rede a pedido. Isto permite uma maior flexibilidade na utilização do laboratório, uma vez que ao utilizador, é dada a possibilidade de seleccionar um dos exercícios pré-configurados ou ainda seleccionar equipamentos, para construção da topologia pretendida.

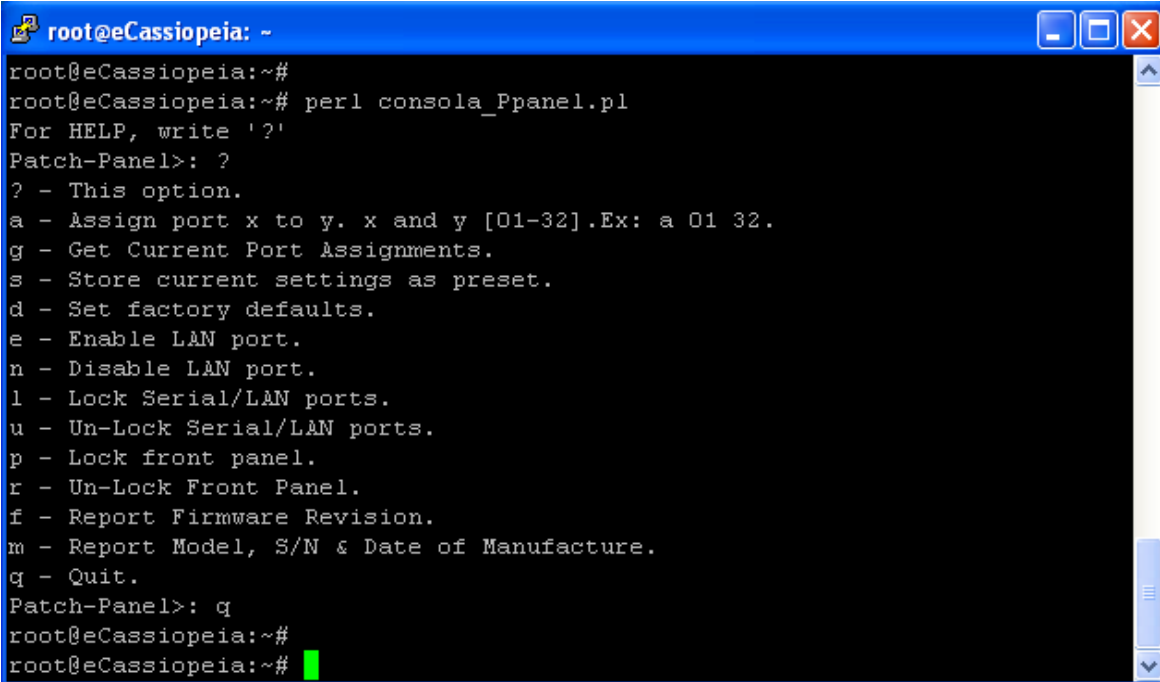
Os painéis já fazem parte da lista de equipamento do laboratório e, fisicamente, a arquitectura da rede de ensino encontra-se conectada a este tipo de painéis, no entanto a sua configuração automática não é, possível pois não existe software de controlo.

Contactei por isso, via e-mail, o fabricante do equipamento e o seu representante na Europa, com o objectivo de me disponibilizarem o software proprietário, para controlo electrónico do patch panel, sem sucesso. Vi-me sem outra alternativa que não resolver sozinho o problema.

Consultando o manual do equipamento, encontrei informação necessária sobre o protocolo de comunicação via telnet, o que me possibilitou desenvolver um programa em perl, uma CLI, para utilização do patch panel da APCON [9], ou de outro modo switch de layer um.

Esta consola funciona como uma interface de administrador.

O utilizador não terá acesso a esta interface, pois a configuração da arquitectura será transparente, isto é, com um clique sobre o exercício que pretende realizar, os scripts php executarão os procedimentos necessários para que isso aconteça. Assim, a consola será utilizada pelo código php ou em caso específico pelo administrador, para processos de administração do patch panel.



```
root@eCassiopeia: ~  
root@eCassiopeia:~# perl consola_Ppanel.pl  
For HELP, write '?'  
Patch-Panel>: ?  
? - This option.  
a - Assign port x to y. x and y [01-32].Ex: a 01 32.  
g - Get Current Port Assignments.  
s - Store current settings as preset.  
d - Set factory defaults.  
e - Enable LAN port.  
n - Disable LAN port.  
l - Lock Serial/LAN ports.  
u - Un-Lock Serial/LAN ports.  
p - Lock front panel.  
r - Un-Lock Front Panel.  
f - Report Firmware Revision.  
m - Report Model, S/N & Date of Manufacture.  
q - Quit.  
Patch-Panel>: q  
root@eCassiopeia:~#  
root@eCassiopeia:~#
```

Figura 4.6 - CLI para controlo do patch panel

O trabalho mais importante seria definir o conjunto de instruções essenciais para o controlo do patch panel. Após este levantamento, foram definidos quais os comandos que implementariam as acções de gestão. Com o intuito de minimizar a complexidade de implementação da CLI, foram utilizados comandos apenas com uma letra, que no entanto poderiam ser complementados com outra informação de configuração. Foi este o trabalho mais complexo. Transformar a codificação da CLI nos comandos que deveriam ser introduzidos no patch panel.

Essa função de tratamento de texto, pesquisa de caracteres e análise da coerência do comando é apresentada em baixo. Todo o algoritmo foi implementado em perl, pela sua conhecida facilidade para tratamento de texto.

```
(...)
case /^a\s\d\d\s\d\d$/
{
    $cmd =~ s/(.)(.)(.)(.)(.)(.)(.)/\3\4\6\7/;
    executeCmd("\n\\\/\|P".$cmd."\n"); }
case /^g$/
{
    $response = execCmdReturnResponse("\n\\\/\|S\n");
    @links = createLinkMatrix($response);
    foreach (@links) {print $_ . "\n";}
}
case /^s$/ { executeCmd("\n\\\/\|G01FFFFFFFF\n"); }
case /^d$/ { executeCmd("\n\\\/\|OF\n"); }
case /^e$/ { executeCmd("\n\\\/\|OE10\n"); }
case /^n$/ { executeCmd("\n\\\/\|OE00\n"); }
case /^l$/ { executeCmd("\n\\\/\|OL1234\n"); }
case /^u$/ { executeCmd("\n\\\/\|OU1234\n"); }
case /^p$/ { executeCmd("\n\\\/\|L1234\n"); }
case /^r$/ { executeCmd("\n\\\/\|U1234\n"); }
case /^f$/ { $response = execCmdReturnResponse("\n\\\/\|R\n");
            $response =~ s/(....)(.*)/\2/;
            print "Firmware Revision: $response\n";
        }
case /^m$/ { getModel(); }
case /^q$/ { $quit=1; }
```

O trabalho inverso, de percepção dos comandos de resposta enviados pelo patch panel para o utilizador, também foi alvo de alguma dedicação. Assim sendo, seria necessário interpretar esses códigos e torná-los legíveis para o utilizador. Dada a grande manipulação de texto necessária, mais uma vez justificava a utilização de perl. Desenvolvi então uma função, que apresentava de forma intuitiva a matriz de comutação do patch panel, facilitando ao administrador rapidamente perceber quais os portos conectados.

Esta função trabalha de forma cíclica, percorrendo toda a matriz de comutação e ordenando por ordem ascendente a apresentação dos resultados, o que facilita ao utilizador, rapidamente, localizar o porto que pretende verificar.

```
#----- ( createLinkMatrix )-----#
#
# FUNCTION:      createLinkMatrix      #
# PURPOSE:      send matrix of port linked to the user      #
# ARGS:        $matrix - the matrix response by patch panel  #
#              @links - return port linked in patch panel    #
#-----#
sub createLinkMatrix {
    local($response) = @_;
    $response =~ s/(....)(.*)/\2/;
    @matrix = split(/(..)/, $response);
    $i=0;

    for($index=0;$index<=31;$index++)
    {
        if ($matrix[2*$index+1] != "00")
        {
            $port_index=$index+1;
            if ($port_index < 10)
            { $links[$i]="0"."$port_index"."<"."-
".">".$matrix[2*$index+1]; }
            else
            { $links[$i]="$port_index"."<"."-
".">".$matrix[2*$index+1]; }
```

```

        $i++;
    }
}

for($index=0;$index<$i;$index++)
{
    $_ = $links[$index];
    /<->/;
    if ("$_" > "$'") { $links[$index] =~ s/(..)(...)(..)/\3\2\1/;}
}

for($index=0;$index<$i;$index++)
{
    for ($start=$index+1;$start<$i;$start++)
    {
        if ($links[$index] == $links[$start]) { delete
$links[$start];}
    }
}
$str=join(' ', @links);
@links = split(/\s+/, $str);
return @links;
}

```

A mais valia que a CLI proporcionou, foi a possibilidade de se configurar dinamicamente as ligações no patch panel de acordo com o exercício a realizar.

4.4 Cálculo de Recursos Disponíveis

Com o intuito de tornar a utilização do laboratório o mais rentável possível, permitindo assim a realização de exercícios simultaneamente por mais que um aluno, tornou-se necessário perceber se existiam recursos suficientes, para a realização do segundo e seguintes exercícios. Assim foi construído um algoritmo, para o cálculo de equipamentos livres para realização de determinado exercício e um outro, para o cálculo da disponibilidade de ligações no patch panel para construir a topologia do exercício.

4.4.1 Equipamentos

O algoritmo que contabiliza o número de equipamentos disponíveis para a realização de exercícios tem as seguintes funções:

- Consulta todos os equipamentos registados na base de dados
- Consulta os equipamentos reservados para o horário em causa
- Consulta o equipamento necessário para o novo exercício que se pretende realizar
- Após calcular a diferença entre as duas primeiras consultas à base de dados, são comparados os valores dos equipamentos disponíveis e pretendidos.

O resultado do último cálculo permitirá avançar ou não para a realização de um novo exercício laboratorial.

4.4.2 Ligações patch panel

O algoritmo que prevê a existência de portos livres no patch panel, para elaborar a topologia do exercício tem um comportamento semelhante:

- Consulta o número de portos que determinado equipamento possui no patch panel
- Consulta os portos do equipamento já utilizados
- Consulta qual a necessidade de portos para elaborar a topologia do exercício pretendido
- Calcula a diferença entre as duas primeiras consultas à base de dados e verifica a disponibilidade com a necessidade do exercício

Este método é iterativo e aplicado a cada equipamento. Se alguma das iterações não satisfizer as necessidades, não será possível realizar o exercício.

4.5 Adição da segunda interface ethernet ao Tux

A abordagem ao eCassiopeia implementada, não considera a segunda interface nas máquinas linux, os Tux. Fisicamente existe uma das máquinas, o TuxY3, em que Y é o número da bancada, ou de outro modo a 3 máquina linux de cada bancada, possui duas interfaces ethernet.

Os exercícios propostos pelos docentes utilizam esta particularidade. A abordagem remota implementada, como não considerava esta especificidade não era possível realizar todos os exercícios remotamente.

Dada esta situação desenvolvi um novo algoritmo em php, que considera este caso, pelo que na definição do novo exercício, é possível indicar que se pretende utilizar a máquinas Tux com estas característica. Foi necessário também redefinir o algoritmo de configuração da topologia, quando esta máquina é contemplada.

O algoritmo actualmente implementado, considera apenas a existência de duas interfaces no TuxY3, tendo sido parametrizado propositadamente para este caso. É pretendido, no entanto, que no futuro o algoritmo considere as outras máquinas, para que de forma transparente se possa adicionar ou retirar interfaces ethernet às máquinas do laboratório, sem que isso cause indisponibilidade do serviço.

4.6 Avaliação

Apesar das melhorias introduzidas no eCassiopeia e apresentadas nesta secção, não terem sido utilizadas por alunos em aulas, foram feitas várias experiências para validar a funcionalidade do trabalho realizado. O primeiro trabalho da disciplina de Planeamento e

Gestão de Redes (PGRE), serviu de base para estas experiências. Em particular, este trabalho intitulado LANs Virtuais, pretende demonstrar aos alunos o que é uma VLAN e provar que não é possível a comutação de nível dois entre VLANs. A primeira proposta que é apresentada aos alunos, é que configurem um endereço de gamas diferente nas máquinas TUX e testem a conectividade IP entre essas máquinas. Para a realização desta tarefa são necessárias as 3 máquinas TUX e apenas um switch. Esta tarefa do exercício, foi escolhida em particular por ser necessário o uso do TUX13 com duas interfaces.

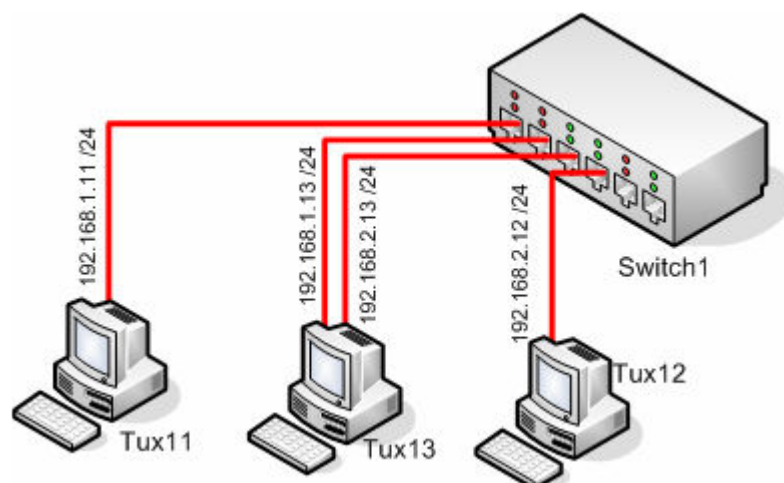


Figura 4.7 - Topologia de teste das funcionalidades implementadas

4.6.1 Arquitectura funcional

Usando a nova arquitectura funcional definida, foi possível entrar no portal de gestão para reservar o exercício de teste e pôr à prova as melhorias implementadas. É de notar que o horário disponível para alocar os exercícios está alargado a todas as horas do dia, ao invés das 8h-23h dos dias úteis.

Após o utilizador se ter cadastrado na página de entrada da aplicação, é confrontado com a agenda do laboratório.

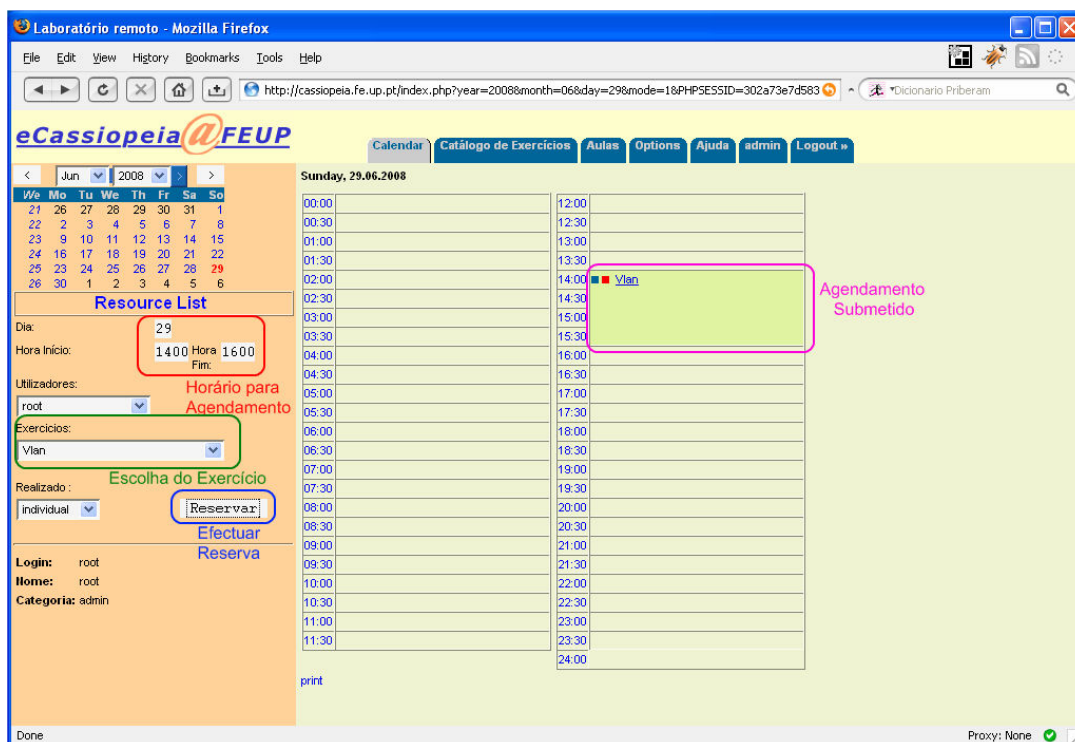


Figura 4.8 - Agenda do Laboratório e reserva de horário para realizar um exercício

Poderá assim escolher o horário livre mais conveniente e agendar o exercício que pretende. Após submeter os dados, a agenda do laboratório é actualizada em função do novo agendamento. Após ter inserido o agendamento e seguindo a ligação adicionada na agenda, ser-lhe-á apresentada a página do exercício de acordo com a figura 4.9.

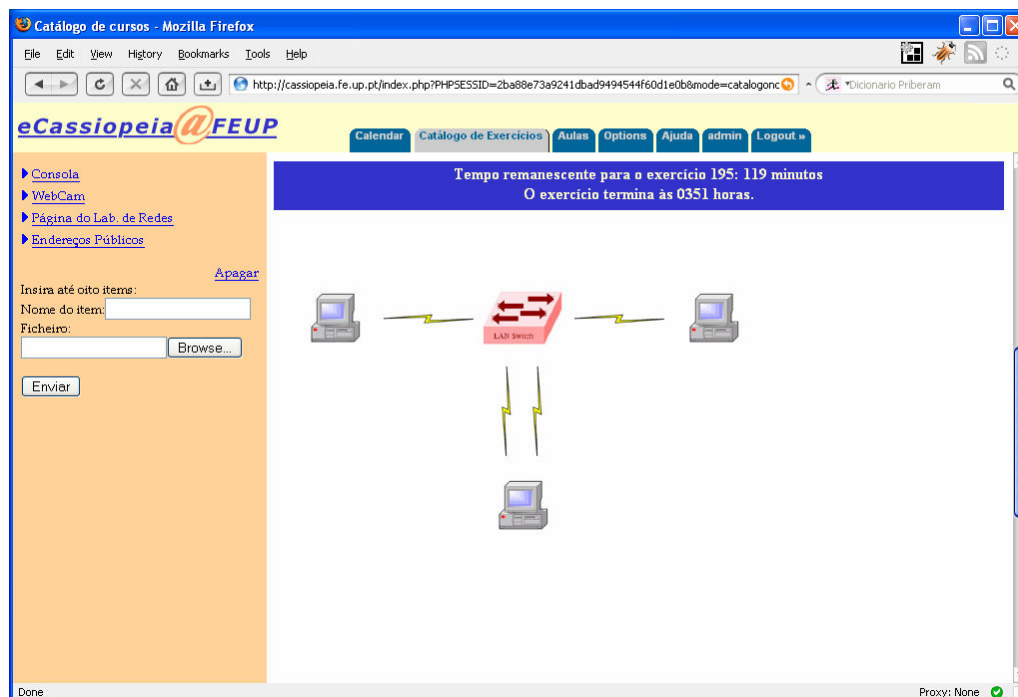


Figura 4.9 - Página típica de um exercício de redes.

A diferença entre abrir a página do exercício no horário agendado ou fora dele, está apenas na activação das ligações às consolas dos equipamentos. No horário de agendado, o aluno poderá aceder via clique do rato em cima dos equipamentos, às suas consolas. Fora do horário, apenas poderá consultar a topologia do exercício.

4.6.2 Consola Java

De acordo com o exercício que se pretende realizar, foram configuradas as interfaces ethernet do TUX13. O acesso à consola foi feito via applet Java instalada no eCassiopeia. Como já foi referenciado, o acesso à consola é feito através de clique do rato em cima do respectivo equipamento.

Foram configurados os endereços IP nas interfaces do TUX13, satisfazendo as necessidades do exercício, como se poderá verificar através da figura 4.10 e 4.11.

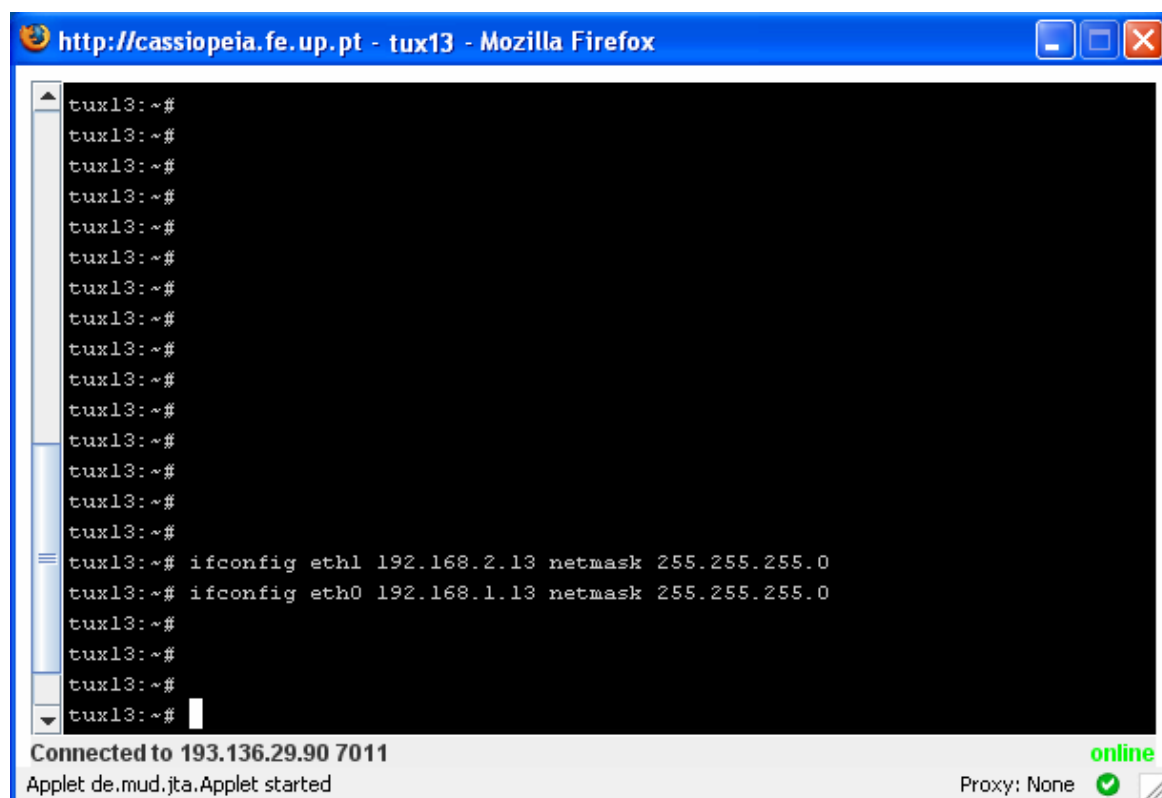


Figura 4.10 - Configuração de IP nas interfaces do TUX13 através da Consola Java.

```

tux13:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:c0:df:08:d5:b3
          inet addr:192.168.1.13  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::2c0:dfff:fe08:d5b3/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:657 errors:89 dropped:0 overruns:0 frame:0
          TX packets:400 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:60045 (58.6 KiB)  TX bytes:37522 (36.6 KiB)
          Interrupt:17 Base address:0x1000

tux13:~# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:01:02:9f:7f:fe
          inet addr:172.16.1.123  Bcast:172.16.1.255  Mask:255.255.255.0
          inet6 addr: fe80::201:2ff:fe9f:7ffe/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1934 errors:0 dropped:0 overruns:0 frame:0
          TX packets:520 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:131018 (127.9 KiB)  TX bytes:51939 (50.7 KiB)
          Interrupt:16 Base address:0x2000
  
```

Connected to 193.136.29.90 7011 online
 Applet de.mud.jta.Applet started Proxy: None ☒

Figura 4.11 - Informação sobre as interfaces do TUX13.

4.6.3 Patch Panel

Com o objectivo de demonstrar o correcto funcionamento do patch panel, apresento a tabela dos portos de equipamento, associados a cada porto no patch panel. São apresentados apenas os portos relevantes para o exercício em causa.

Equipamento / Interface	Patch Panel Port
Tux11 / eth0	2
Tux12 / eth0	3
Tux13 / eth0	4
Switch1 / eth1	5
Switch1 / eth2	6
Switch1 / eth3	7
Switch1 / eth4	8
Router1 / eth1	9
Router1 / eth1	10
Tux11 / eth1	11

Tabela 4.1 – Associação dos Portos do Patch Panel e Equipamentos do Laboratório.

Antes do acesso ao exercício que se pretendia realizar e imediatamente após se terem iniciado os procedimentos para o resolver, foi retirada informação sobre a matriz de comutação no patch panel, de acordo com a figura 4.12.

```
eCassiopeia:~# perl consola_Ppanel.pl
For HELP, write '?'
Patch-Panel>: ?
? - This option.
a - Assign port x to y. x and y [01-32].Ex: a 01 32.
g - Get Current Port Assignments.
s - Store current settings as preset.
d - Set factory defaults.
e - Enable LAN port.
n - Disable LAN port.
l - Lock Serial/LAN ports.
u - Un-Lock Serial/LAN ports.
p - Lock front panel.
r - Un-Lock Front Panel.
f - Report Firmware Revision.
m - Report Model, S/N & Date of Manufacture.
q - Quit.
Patch-Panel>: g
02<->09
03<->05
04<->10
06<->11
Patch-Panel>: g
02<->05
03<->06
04<->07
08<->11
Patch-Panel>: q
eCassiopeia:~#
eCassiopeia:~#
```

Figura 4.12 - Alteração da Matriz de Comutação de portos no Patch Panel.

Observando cuidadosamente os resultados, tomando a tabela 1 como referência e comparando com a topologia pretendia, poder-se-á confirmar que o teste revela a funcionalidade da solução, pois implementa correctamente a topologia pretendida para o exercício.

4.6.4 Segunda interface no Tux3

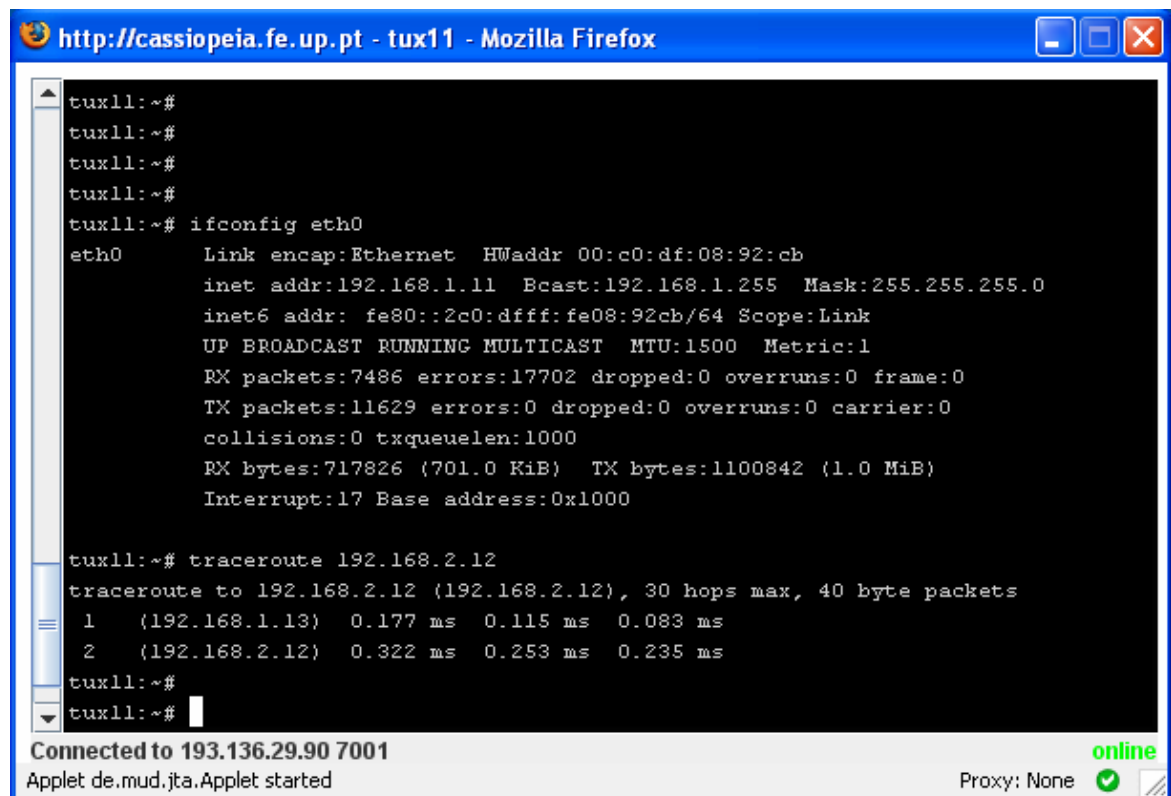
O trabalho acima escolhido, não foi fruto do caso pelo facto de ser necessário demonstrar a utilização de duas interfaces ethernet nos Tux3.

Dado que o tux11 e 12 estão em subnets diferentes, não teriam conectividade, apesar de fisicamente ligados ao mesmo switch. É esta a ideia que se pretende passar aos alunos com este exercício, no entanto, se usarmos o Tux13, que possui duas interfaces uma em cada rede, como router, teremos a partir desse momento conectividade. Foi esse o teste que se fez, de modo a provar que só existe conectividade entre as duas subnet através do tux13.

Assim, configurou-se a máquina tux13 para funcionar como router, de acordo com os seguintes comandos:

```
tux13:~# cat /proc/sys/net/ipv4/ip_forward
0
tux13:~# echo 1 > /proc/sys/net/ipv4/ip_forward
tux13:~# cat /proc/sys/net/ipv4/ip_forward
1
tux13:~#
```

Após esta configuração fizeram-se traceroute, que provam que os pacotes passam no tux13 antes de chegarem ao destino.



```
http://cassiopeia.fe.up.pt - tux11 - Mozilla Firefox

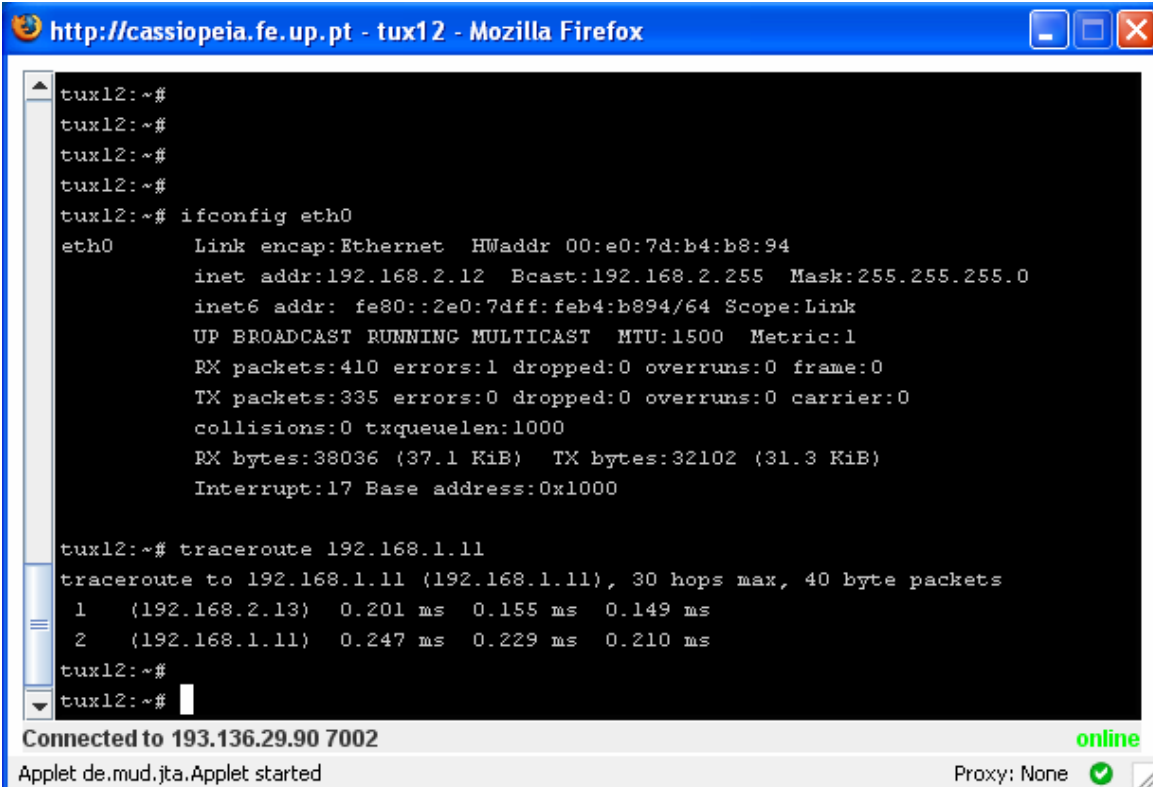
tux11:~#
tux11:~#
tux11:~#
tux11:~#
tux11:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:c0:df:08:92:cb
          inet addr:192.168.1.11  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::2c0:dfff:fe08:92cb/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7486 errors:17702 dropped:0 overruns:0 frame:0
          TX packets:11629 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:717826 (701.0 KiB)  TX bytes:1100842 (1.0 MiB)
          Interrupt:17 Base address:0x1000

tux11:~# traceroute 192.168.2.12
traceroute to 192.168.2.12 (192.168.2.12), 30 hops max, 40 byte packets
 1  (192.168.1.13)  0.177 ms  0.115 ms  0.083 ms
 2  (192.168.2.12)  0.322 ms  0.253 ms  0.235 ms
tux11:~#
tux11:~#
```

Connected to 193.136.29.90 7001
Applet de.mud.jta.Applet started

Proxy: None online

Figura 4.13 - Traceroute do tux11 para o tux12 passando pelo tux13.



```
tux12:~#
tux12:~#
tux12:~#
tux12:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:e0:7d:b4:b8:94
          inet addr:192.168.2.12  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::2e0:7dff:feb4:b894/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:410 errors:1 dropped:0 overruns:0 frame:0
          TX packets:335 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:38036 (37.1 KiB)  TX bytes:32102 (31.3 KiB)
          Interrupt:17 Base address:0x1000

tux12:~# traceroute 192.168.1.11
traceroute to 192.168.1.11 (192.168.1.11), 30 hops max, 40 byte packets
 1  (192.168.2.13)  0.201 ms  0.155 ms  0.149 ms
 2  (192.168.1.11)  0.247 ms  0.229 ms  0.210 ms
tux12:~#
tux12:~#
```

Connected to 193.136.29.90 7002 online
Applet de.mud.jta.Applet started Proxy: None

Figura 4.14 - Traceroute do tux12 para o tux11 passando pelo tux13.

4.6.5 Recursos Ocupados

Foi necessário também testar os algoritmos para o cálculo das limitações de recursos para a execução de um trabalho. Assim foram forçados a acontecer os dois casos previstos, o limite de material e o limite de ligações disponíveis no patch panel. Os resultados obtidos são apresentados nas duas seguintes secções.

4.6.5.1 Limitação de Material

Após ter agendado um exercício em determinado horário, tentei agendar um outro que necessitaria do algum material utilizado pelo primeiro exercício. A ocorrência deste erro dá-se durante a fase de agendamento do exercício, pois não é possível reservar para uma mesma hora o mesmo equipamento. Este caso reporta a página da figura 4.15.

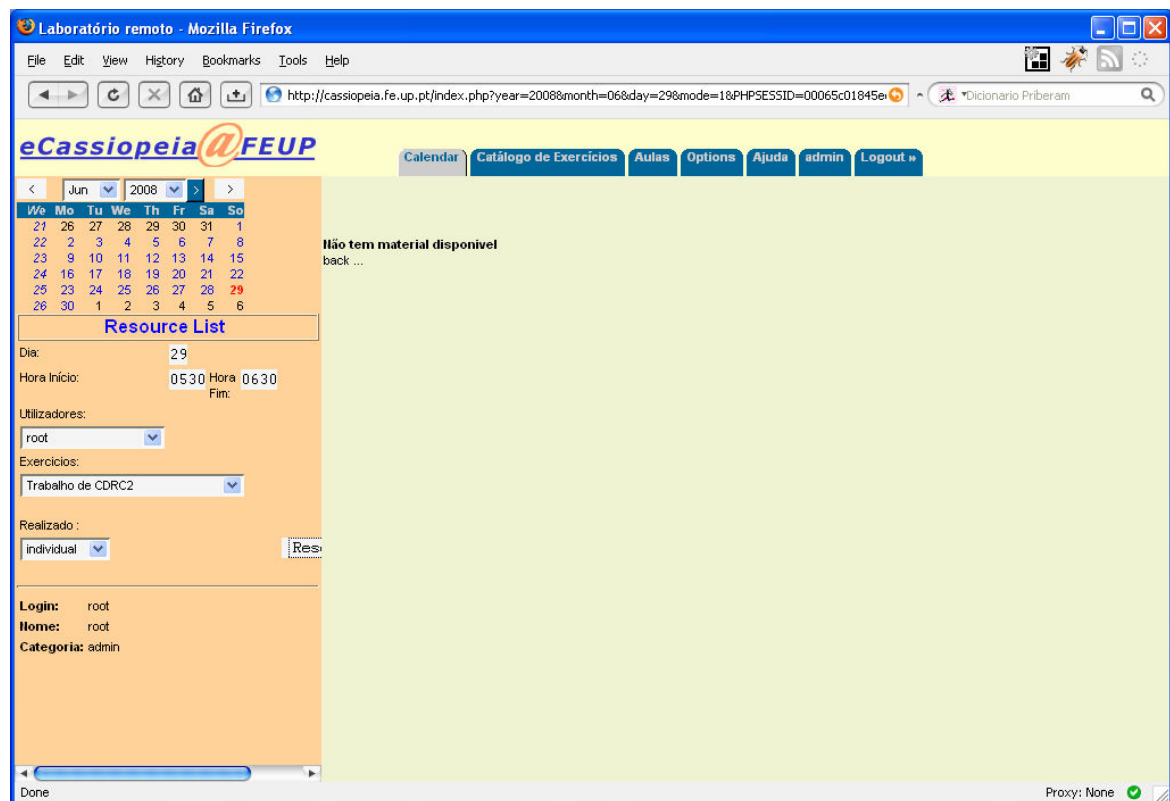


Figura 4.15 - Mensagem de alerta se não existir material disponível para realizar um exercício.

4.6.5.2 Limitação de Ligações no Patch Panel

A mensagem de erro despoletada nos casos em que o número de ligações no patch panel está condicionado, acontece apenas quando iniciamos a realização do exercício. Nesse momento, o sistema ao tentar implementar a topologia no patch panel, faz uma revisão às ligações detectando se a topologia é viável de implementar ou não.

A mensagem de erro ocorreu quando tentei realizar um exercício que previa a ligação das três máquinas Tux ao router1. De acordo com a tabela 1, poderemos verificar que apenas existem duas ligações possíveis ao router, pelo que a mensagem de erro é então despoletada.

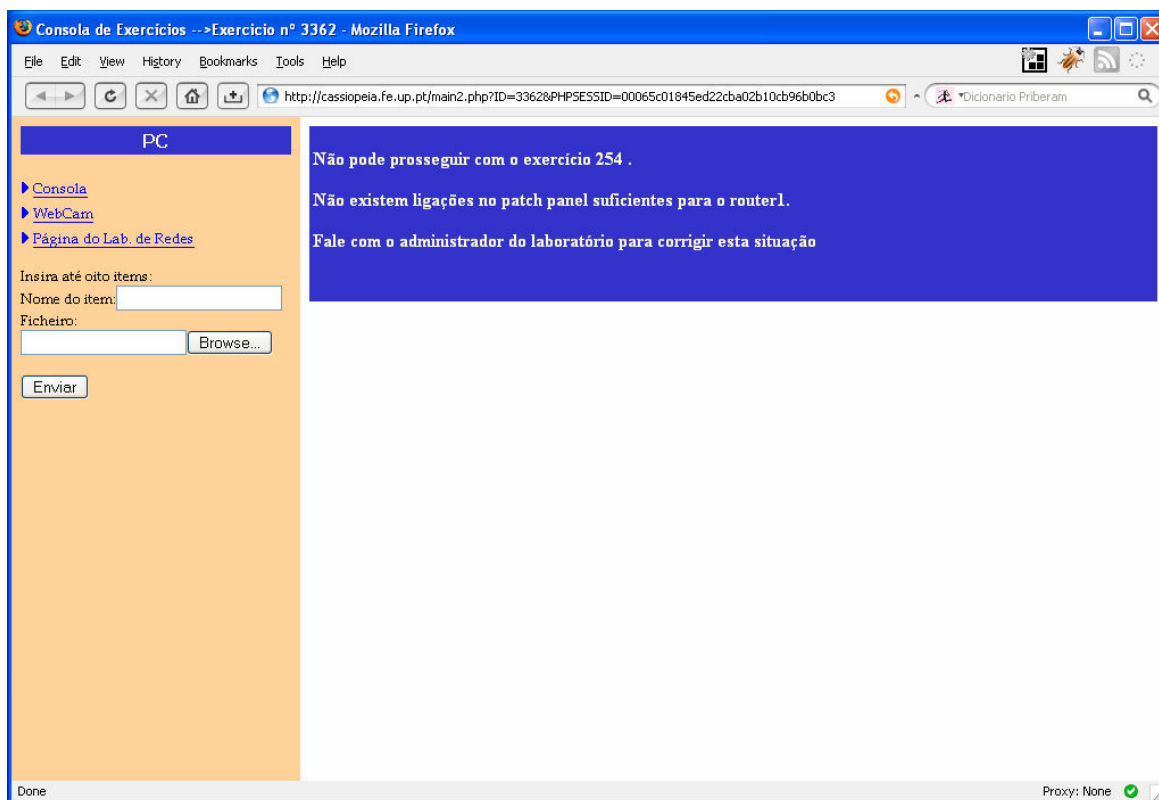


Figura 4.16 - Mensagem de alerta se não existir ligações no patch panel suficientes para implementar a topologia do exercício

Capítulo 5

5. Proposta de Nova Arquitectura

5.1 Patch Panel

O laboratório físico que suporta o projecto eCassiopeia, funciona como sala de aulas práticas para as disciplinas nas áreas das Redes de Computadores. Este laboratório divide-se em seis bancadas de trabalho que permitem a organização das turmas em 6 grupos de trabalho. O trabalho em pequenos grupos, possibilita que todos os alunos partilhem da mesma oportunidade de acesso aos equipamentos, pondo em prática as noções teóricas que assimilam durante as aulas práticas do curso.

Não sendo o objectivo deste trabalho remodelar a distribuição física das bancadas, é possível definir uma arquitectura lógica de ligações, que permita desenhar qualquer trabalho prático, independentemente da sua localização na sala, pelo que é necessário avaliar a arquitectura de ligações mais flexíveis e que melhor rentabilizem a utilização do trabalho em laboratório.

Os equipamentos que compõem a sala têm as seguintes interfaces ethernet:

- 6 bancadas, com 2 máquinas linux com 1 única interface ethernet perfazendo o total de 12 interfaces;
- 6 bancadas com 1 máquinas linux que possui 2 interfaces ethernet totalizando 12 interfaces;
- Existe 1 switch por cada bancada com 24 interfaces cada que perfaz 144 interfaces;
- Existem 3 router / switch layer 3, cada um partilhado por 2 bancadas, que possuem 24 interfaces ethernet lan, o que sumaria mais 72 interfaces;
- Associado a cada um dos routers acima descritos, existem mais 3 routers com interface wan/lan, do lado wan tem interfaces serial pelo que não serão contabilizadas, e uma interface ethernet lan que completa o cálculo com mais 3 interfaces.

Assim, temos um total de 243 portas ethernet a satisfazer no patch panel para termos o laboratório completamente autónomo.

Poderemos otimizar este número restringindo as ligações possíveis em cada exercício, no entanto, é de notar que os próprios equipamentos impõem já uma limitação intrínseca.

Vejamos a situação em que se pretende ter todas máquinas linux ligadas a um switch e por sua vez estas com acesso wan via router apropriado da nortel, que existe no equipamento. As máquinas linux necessitam de 24 portas perfazendo a capacidade disponível no switch, faltando satisfazer no entanto, a ligação da interface lan do router wan/lan ao switch.

Este caso específico, demonstra a necessidade de mais do que um switch para a realização de trabalhos mais complexos. Assim nasce uma nova proposta, sendo que as máquinas de cada bastidor/bancada apenas se ligam ao switch situado nesse bastidor. Pretendendo a comunicação de layer 2, temos que interligar todos os switches. Na melhor das hipóteses fazemos um anel de switches, ocupando 2 interfaces de cada switch (vizinho do lado esquerdo e do lado direito) e activamos o spanning-tree. Na pior das hipóteses, ligamos todos os switch a um central, perfazendo uma topologia em estrela. Deste modo, utilizamos 5 interfaces no switch que se localiza no centro da estrela. Se pretendermos saturar o switch com os restantes equipamentos, serão necessárias apenas mais 6 interfaces, sendo que 3 se justificam pela ligação de cada um dos routers de lan, e as 3 restantes pela ligação dos 3 routers de wan. Esta opção revela serem necessários apenas 15 portos em cada equipamento switch (3 máquinas linux que contabilizam 4 interfaces, 5 switches e 6 routers).

Utilizando o mesmo pensamento lógico, para contabilizar a saturação dos router, temos 6 switches, 2 router lan e 3 router wan, que somam 11 portos. No entanto teremos que entrar neste caso com uma nova variável, isto é, a presença de máquinas unix ligadas directamente ao router. Como cada router serve 2 bastidores (dois conjuntos de 3 máquinas que possuem 4 interfaces), são necessários no máximo mais 8 portos, totalizando assim a necessidade de 19 portos no router. Qualquer outra arquitectura em que se pretendam ligar mais máquinas a um único router, poderá ser elaborada fazendo a ligação dessas máquinas a um switch, o qual já está contemplado nas ligações, e a possível utilização de marcação de pacotes com o protocolo dot1q, mais conhecido por vlans.

Este exercício leva-nos a afirmar que se poderá alcançar toda a conectividade entre equipamentos do laboratório, desde que o patch panel contemple os seguintes portos:

- 15 portos por switch, o que nos leva ao número 90;
- 19 portos por router lan, que perfaz 57 portos;
- 12 portos para a ligação das 2 máquinas linux de cada uma das 6 bancadas que possuem apenas 1 interface ethernet;
- 12 portos para a ligação da máquina linux de cada uma das 6 bancadas que possuem 2 interfaces ethernet;

Contabiliza-se assim a necessidade de um patch panel com 171 portos.

Uma pesquisa de mercado deste tipo de produtos leva-nos ao encontro de 2 grandes fabricantes, a APCON [9], que comercializa equipamentos com o mesmo nome e a Gillaspay Associates, Inc. [10], que comercializa a marca LinkXchange.

Estes dois fabricantes apenas disponibilizam módulos de 288, 144, 64, 32 e 16 portos [9] [10] pelo que qualquer uma das opções apresentadas implica a necessidade do produto com maior número de portos.

Uma análise ao custo dos equipamentos, onde apenas foi possível aceder ao preço dos equipamentos de 16 e 32 portos, revela valores aproximados de \$4,000.00 e \$6,500.00 respectivamente. Pela análise efectuada e pelo facto de não serem públicos os valores de

equipamentos superiores, apenas mediante contacto por formulário, leva a pensar que os valores serão economicamente proibitivos.

O factor económico e a existência de 3 patch panel de 32 portos cada no laboratório, leva a considerar outro tipo de soluções para satisfazer as topologias necessárias aos exercícios

Apresento assim a solução para o caso mais dispendioso, com a utilização de um único patch panel, a arquitectura plana, e a solução alternativa, mais económica, e que prevê a utilização do material existente no laboratório, a hierarquia de patch panel.

5.1.1 Arquitectura Plana

Esta arquitectura define um mesmo nível hierárquico para todos os equipamentos. Poderemos olhá-la como uma sala totalmente ao dispor do aluno. Todas as combinações de ligações são possíveis, o que permite a utilização de qualquer equipamento do laboratório para definir a topologia de um exercício.

Esta estrutura tem a vantagem de não ter limitações físicas a qualquer combinação de ligações e a mais valia de se poderem utilizar todos os equipamentos disponíveis no laboratório, permitindo assim a realização de exercícios mais complexos, mais próximos de um cenário real.

A grande limitação desta arquitectura, prende-se com o custo dos recursos necessários para se criar a matriz de comutação, capaz de perfazer os pressupostos. Como é sabido, qualquer projecto está condicionado por questões de carácter monetário.

Atendendo a estas considerações, foi proposta uma nova arquitectura com o objectivo de colmatar as limitações da arquitectura plana.

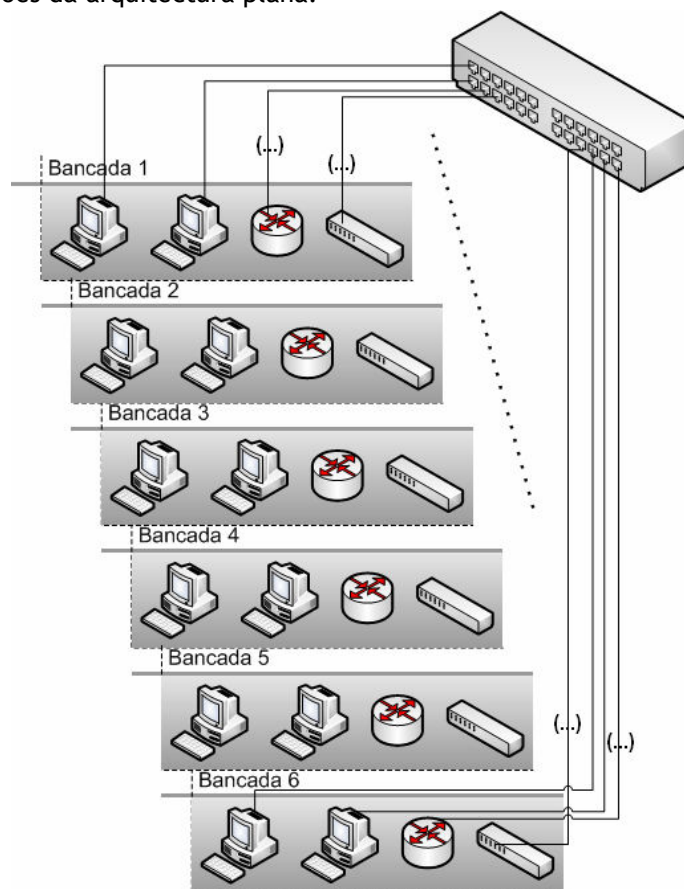


Figura 5.1 - Proposta Arquitectura Plana

5.1.2 Arquitectura Hierárquica

Este tipo de arquitectura define-se por estar orientada ao conceito de bancada, tal como está fisicamente o laboratório. As ligações entre bancadas, ou conjunto de bancadas serão hierárquicas.

Grande parte dos trabalhos são testados entre pares de bancadas, pelo que temos três grandes grupos de ligações. Teremos que, a um nível superior, garantir as ligações entre estes três agrupamentos de equipamento.

A orientação deste tipo de arquitectura, limita o número de combinações de ligações possíveis, o que do ponto de vista económico a torna atractivo, pois poderemos utilizar os recursos já existentes.

A desvantagem desta arquitectura prende-se, no entanto, precisamente com o limite de combinações, que pode ser de algum modo constrangedor, quando se pretenderem implementar exercícios mais arrojados.

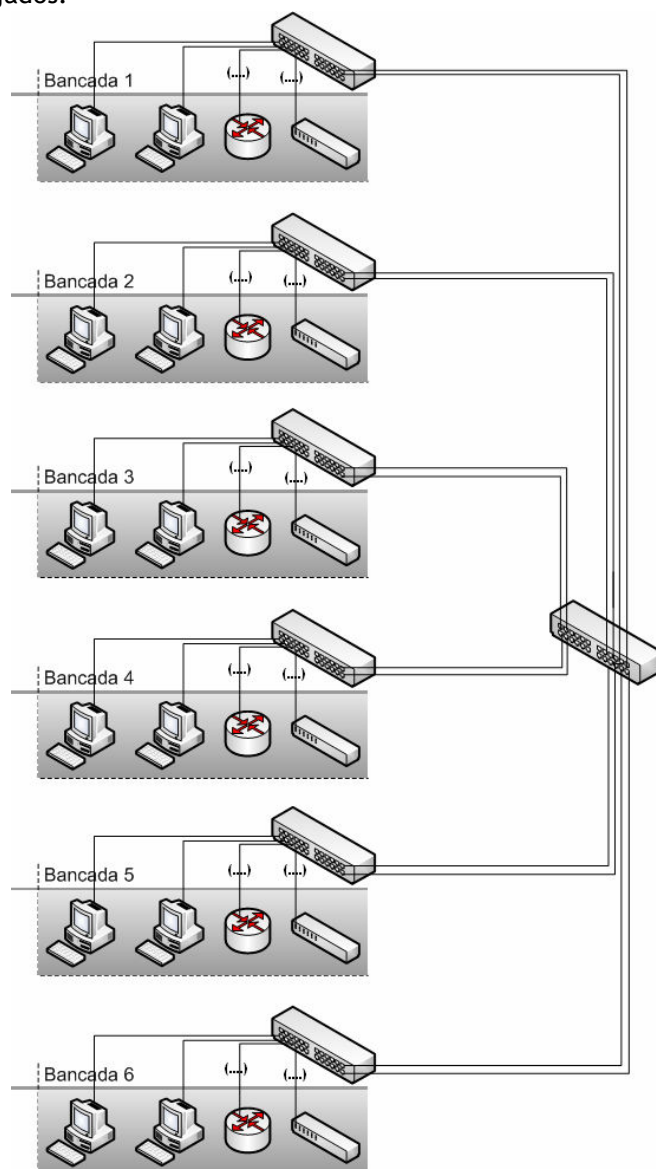


Figura 5.2 - Proposta Arquitectura Hierárquica

Esta disposição, em consonância com a organização do laboratório, poderá ser aproveitada no dia a dia da sala, como suporte às aulas, garantindo entre pares de bancadas as ligações essenciais para grande parte dos trabalhos laboratoriais.

A proposta especificada prevê a existência de dois níveis de comutadores. O nível II é responsável pelas ligações dos equipamentos da bancada, ou seja, prevê apenas a matriz de comutação dentro da bancada. O nível I tem a função de interligar bancadas, quando o exercício implicar a interação de bancadas, ou até mesmo os equipamentos de uma única bancada, caso não perfaçam os requisitos físicos do exercício e sejam necessário repescar esses recursos. Estas ligações entre bancadas são limitadas, obviamente, por razões económicas, mas terá de ser ponderado qual o número adequado de ligações tendo em conta os trabalhos realizados no laboratório.

Para potenciar a utilização do laboratório sem recorrer à compra bruta de equipamentos, seria aconselhável utilizar um patch panel por cada duas bancadas e adquirir um outro. Poder-se-ia até economizar os custos, com a compra de um patch panel com numero menor de recursos, que faria a interligação de nível I.

Para que seja possível a entrada imediata em produção do laboratório, com o número particular de três equipamentos, apresento de seguida a solução possível.

5.1.3 Architectura com três Patch Panel

No caso específico do laboratório, poderemos pensar numa arquitectura, para já, com apenas três comutadores, onde agruparemos várias bancadas, aproveitando os recursos existentes. Será aconselhável, assim, que cada patch panel sirva pelo menos duas bancadas e algum equipamento de uma terceira e entre eles sejam reservadas ligações, para comutarmos um equipamento de determinado patch panel para um porto de um outro patch panel.

Como se pode verificar na figura 5.3 se pretendêssemos ligar um novo equipamento no patch panel 3, por exemplo uma nova máquina, e pretendêssemos ligá-la directamente ao router1, estaríamos limitados pelas ligações entre o patch panel 1 e 2, apesar de entre o patch panel 2 e 3 existirem recursos livres.

As ligações a amarelo representam os cabos que interligam os equipamentos e o patch panel. Os traços azuis representam também cabos, que interligam portos de patch panel diferentes e que permitem ligar equipamento em sistemas diferentes.

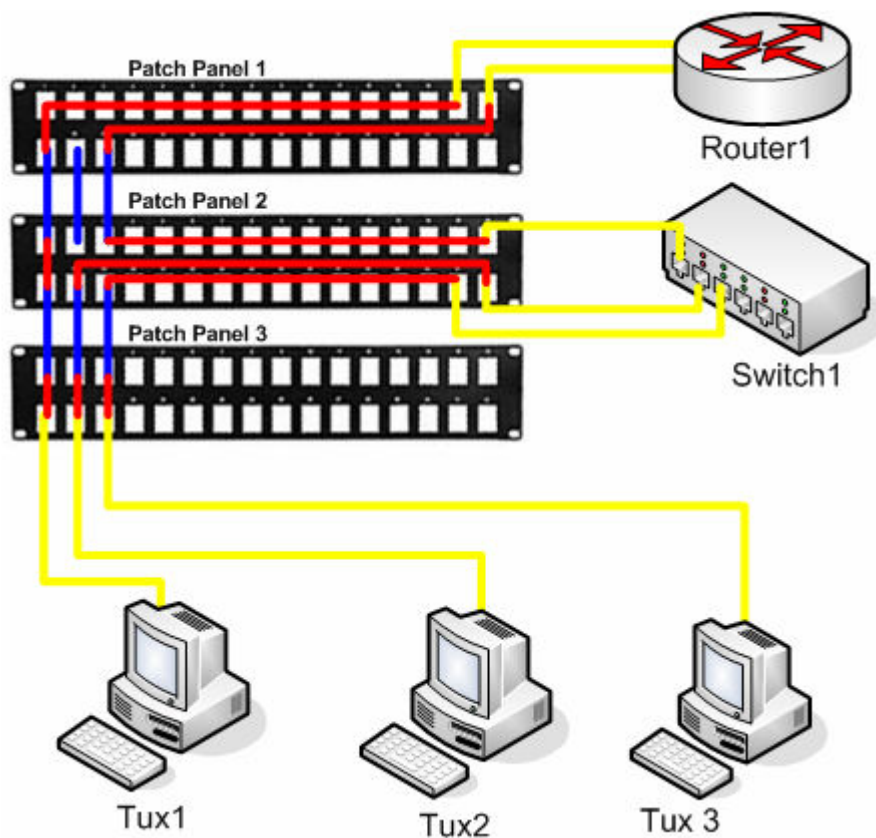


Figura 5.3 - Arquitectura com 3 Patch Panel

Esta é uma solução que poderá como vimos implicar restrições, no entanto serve mais como representação da necessidade que existe, em termos de ligações entre patch panel, possibilitando a comutação de equipamentos para definir topologias complexas para os exercícios.

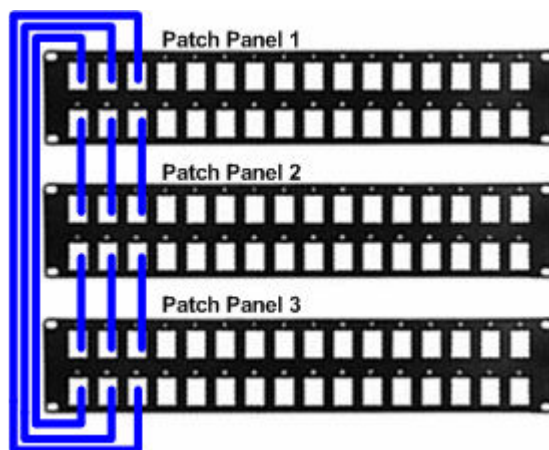


Figura 5.4 - Arquitectura 3 Patch Panel em Anel

A figura 5.4 permite resolver a limitação encontrada na solução anterior. Criamos com este novo modelo de interligação, um anel de ligação entre os patch panel.

Esta opção permite aumentar o número possível de combinações, para que se possam interligar equipamentos em diferentes patch panel, no entanto são necessárias logo à partida mais portas ocupadas para satisfazer este objectivo sem que isso possa ser útil.

Será necessário estudar a melhor distribuição possível para os equipamentos, de modo a permitir o maior número de topologias de exercícios.

5.2 Laboratórios Virtuais

A necessidade, anteriormente verificada de se incluir um ambiente virtual no eCassiopeia leva-me a integrar esta funcionalidade no sistema.

A pesquisa e teste de alguns softwares de virtualização de sistemas de redes, levou-me de encontro ao GNS3. É um software opensource baseado no NS, conhecidíssimo software de teste de redes com interface gráfica, daí GNS, graphical network Simulator.

O GNS3 [11] pode ser instalado em qualquer sistema operativo, no caso particular do debian, sistema operativo pelo qual o eCassiopeia é suportado, existe uma pacote .deb já compilado, que facilita a instalação. Depois de instalado, é possível definir a topologia do exercício que se pretende testar e carregar o ios (Internetwork Operating System) ou firmware dos equipamentos que definem o exercício. O software emula o processador utilizado pela cisco, pelo que é possível carregar qualquer ios deste fabricante nos routers e switchs e executar as configurações como se de um ambiente real se tratasse.

Esta solução parece-me particularmente interessante, pelo facto de no início deste ano lectivo terem sido comprados equipamentos cisco e os docentes adoptaram estes equipamentos como a ferramenta principal nas suas aulas práticas. Com a utilização de ambiente virtual, os alunos poderiam preparar a configuração dos equipamentos, remotamente, neste laboratório virtual e no horário da sua aula prática rapidamente concluíam o exercício, aproveitando a presença do docente, para elucidarem as dúvidas que lhes tivessem surgido durante a realização em casa.

A limitação deste software é que não é possível emular a presença de um cliente, ou seja um pc, no entanto tem a vantagem de suportar todos os ios da cisco.

O modo de acesso à consola dos equipamentos é via protocolo telnet ou ssh, a um porto do localhost, no qual o software escuta os comandos e reencaminha para o equipamento virtual, como se de um acesso remoto se tratasse. Existe, é claro, um porto de acesso diferente por cada equipamento

Este modo de implementação leva-nos a definir duas abordagens para a utilização desta solução, as quais apresento em secções distintas:

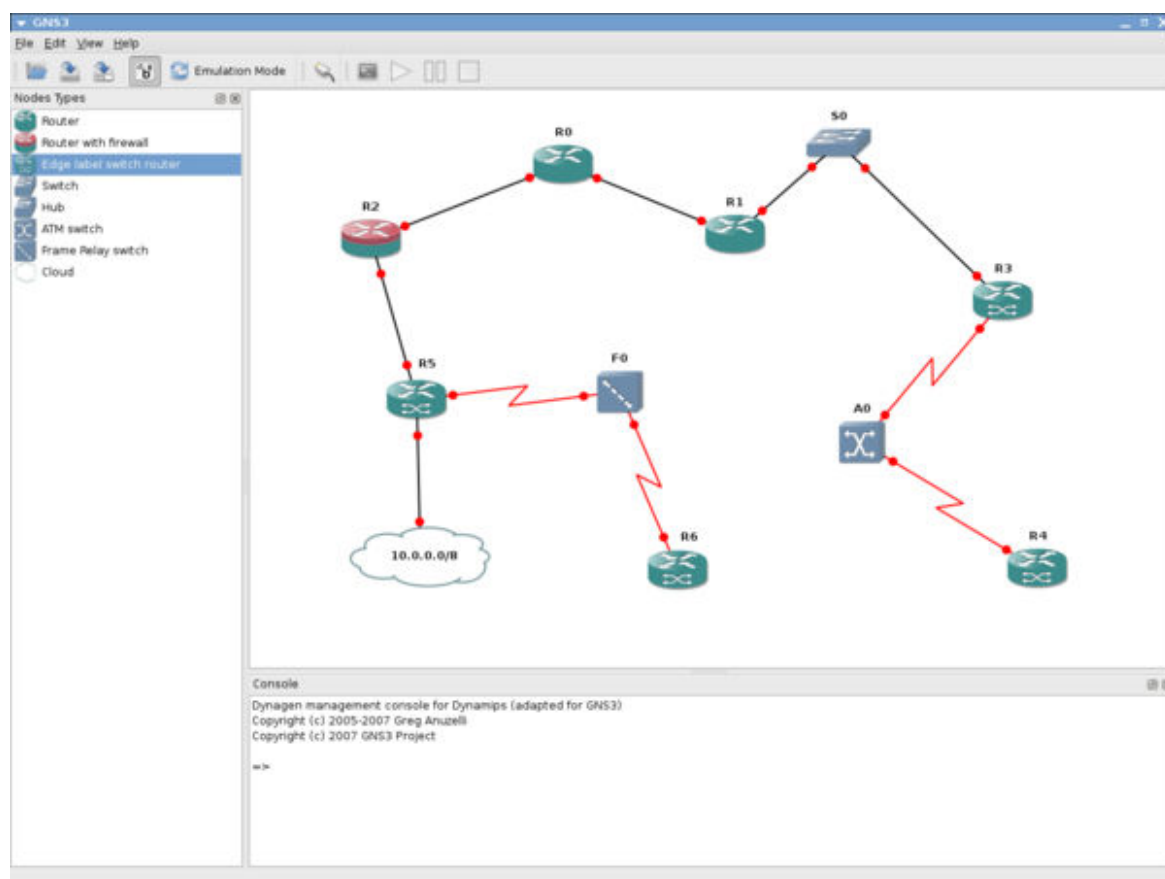


Figura 5.5 - Interface Gráfica do GNS3

```

R0
*Mar 1 00:00:00.440: %PA-3-NOTSUPPORTED: PA in slot1 (Unknown (type 65535)) is not supported on this image.
Please issue "show diag" in fully loaded IOS image
to get the PA's information and verify if it is supported
by this image, a newer version may be needed.
*Mar 1 00:00:02.431: %LINK-3-UPDOWN: Interface Ethernet0/0, changed state to up
*Mar 1 00:00:03.433: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/0, changed state to up
*Mar 1 00:00:06.162: %SYS-5-RESTART: System restarted --
Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-I03-M), Version 12.3(23), RELEASE SOFTWARE (fc5)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Tue 24-Jul-07 15:44 by stshen
*Mar 1 00:00:06.166: %SNMP-5-COLDSTART: SNMP agent on host Router is undergoing a cold start
*Mar 1 00:00:07.776: %LINK-5-CHANGED: Interface Ethernet0/0, changed state to administratively down
*Mar
Router> 1 00:00:08.778: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/0, changed state to down
Router#
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#

```

Figura 5.6 - Consola do Router Virtual R0 (exercício da figura 5.5)

5.2.1 Remote Desktop

A opção de acesso via remote desktop, implica que os utilizadores se liguem ao servidor eCassiopeia, executem o programa, preparem a topologia do exercício, carreguem os firmwares

e depois façam as configurações, executando telnet ou ssh aos portos locais respectivos da consola de cada equipamento.

Todo o processo pode também ser abreviado se existirem já topologias definidas. Assim o utilizador apenas executava o gns3, com uma topologia já definida e teria apenas de executar as configurações via telnet/ssh.

Esta opção necessita termos instalado um servidor para aceitar conexões remotas, via protocolo remote desktop.

5.2.2 Acesso Telnet / Ssh

Uma outra solução e que penso ser mais elegante, será aproveitar o facto do gns3 abrir localmente um porto para acesso à consola de configuração. Neste caso, não necessitaríamos de utilizar remote desktop, mas sim definir um nat para o IP público do eCassiopeia e utilizando a solução por mim já apresentada, com a consola de telnet/ssh em Java, aceder ao equipamento remotamente.

Aprofundando mais esta opção a qual penso ser a que traz mais vantagens, teríamos exercícios tipo, com topologias pré-configuradas. Através do portal, o aluno decidia realizar determinado exercício de modo virtual De seguida executaríamos o gns3 com as configurações respectivas ao exercício escolhido e, após este ter iniciado, abriria então portos locais para acesso à consola dos equipamentos. Nessa altura, no iptables, acrescentávamos uma regra para nat desse porto local num porto público e configurar-se-ia um ficheiro html com as configurações para acesso ao ip e porto onde a consola do equipamento virtual era acedida. Poderíamos de seguida apresentar uma página web com a topologia que o aluno iria resolver, na qual, ao clicar sobre o equipamento, era possível aceder à consola do equipamento, à semelhança do que acontece com o equipamento real.

5.2.3 Integração do Laboratório Virtual e Remoto

Uma abordagem complementar, é utilização simultânea dos recursos virtuais e físicos do laboratório, de modo a que os elementos simulados, possam interagir com os elementos físicos, presenciais ou remotos. A vantagem seria de expandir, a baixo custo, a rede do laboratório e a complexidade dos problemas, mantendo uma componente física de interacção com o equipamento que é importante para os alunos. Neste caso, o módulo de laboratório virtual teria que permitir esta interacção, possivelmente através do redireccionamento de tráfego dos elementos de rede simulados, para os elementos de rede físicos.

Esta abordagem é possível, utilizando o Mircrosoft Loopback [25], estando já testada esta solução. Dada a utilização do Sistema Operativo Linux, é necessário encontrar uma alternativa compatível com o GNS3 para a plataforma Linux. Se os resultados da pesquisa forem infrutíferos, existe sempre a possibilidade de reutilizar código do NS2 [26], que permite já esta funcionalidade e integrar no GNS3. Uma outra alternativa, seria programar esta funcionalidade no GNS3, com o uso de tap. Relembro que o GNS3 e NS2 são programas de código aberto, o que facilita a sua reutilização e alteração.

5.3 Laboratório Global

Dadas as limitações orçamentais com que se debatem todas as universidades e instituições de ensino público, poucas vezes os fundos são canalizados para financiarem a modernização dos laboratórios, e dada a mais valia do conceito de laboratório remoto, fui levado a considerar a existência de uma solução global, com a partilha de sinergias e custos a nível nacional.

É com este objectivo que proponho uma arquitectura, onde se possam interligar vários laboratórios e, por exemplo para um dado exercício, utilizar um equipamento de um laboratório remoto de uma outra instituição pública, que no laboratório de origem não se encontra disponível.

Assim, estudei a possibilidade de se fazer uma VPN sobre a Internet, entre os laboratórios associados, independente do ISP, de modo a que seja possível conectar equipamentos entre sítios distintos. A figura pretende realçar a ideia que tento passar, onde se utiliza uma máquina de um outro laboratório remoto, para auxiliar o exercício do laboratório principal.

A filosofia que está na essência desta proposta, prevê a criação de um túnel entre os routers de bordo do laboratório, à semelhança de uma VPN, que permita ligar o equipamento que se pretende recrutar ao laboratório auxiliar.

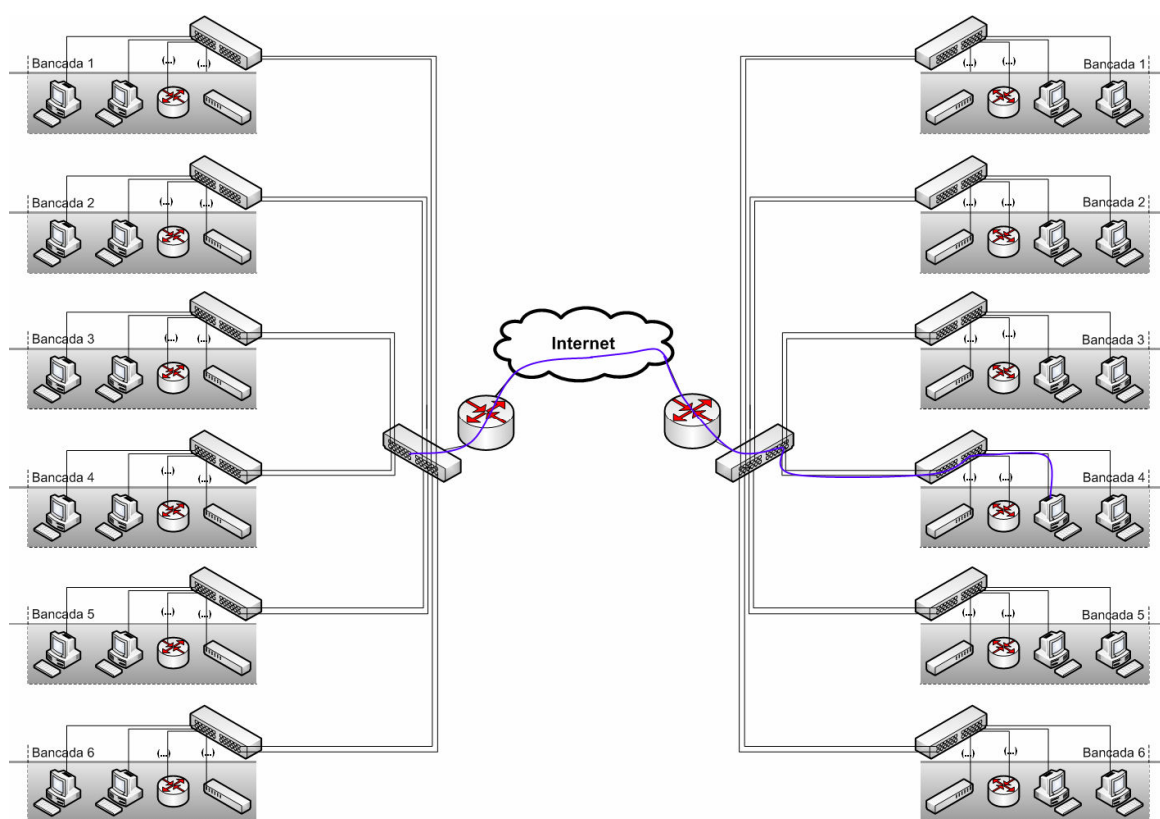


Figura 5.7 - Interligação de laboratórios

As seguintes tecnologias foram estudadas como possíveis soluções para a implementação desta funcionalidade:

- VLAN over Internet
- PPTP/GRE
- L2F
- L2TP
- IPSec

5.3.1 VLAN over Internet

A implementação ideal para a proposta que se apresenta, seria o uso de VLANs sobre a Internet. Com o conceito de VLANs, poderíamos associar os equipamentos, do laboratório remoto que pretendemos usar, a uma VLAN com a mesma tag da VLAN onde o pretendemos inserir no nosso laboratório. O desafio passaria então por fazer com que os dois routers de bordo dos laboratórios, falassem VLAN (802.1Q) sobre a Internet. Uma pesquisa detalhada revelou que actualmente não existe uma solução comercial que implemente esta funcionalidade.

No entanto, existe um projecto de âmbito académico de nome SERVAL, desenvolvido no Departamento de Ciências da Computação da Universidade da Corunha, que apresenta uma proposta curiosa para resolver este problema.

O objectivo passa por construir num software cliente, ao nível da camada de aplicação, o pacote que se pretende fazer chegar à VLAN, e de seguida, esses dados serão processados de modo natural na pilha do modelo OSI.

O pacote fluirá na rede até um servidor que, ao nível da camada de aplicação corre um programa capaz de analisar os dados e tratá-los como um switch, comutando então as VLANs.

A proposta é então do tipo cliente/servidor, em que cada equipamento se conecta ao servidor com uma ligação PPP, transportando na camada da aplicação a emulação do pacote que percorreria a rede, no caso de uma rede local. O servidor funcionará como um switch, recebendo os pacotes, processando-os e comutando-os entre os variados clientes.

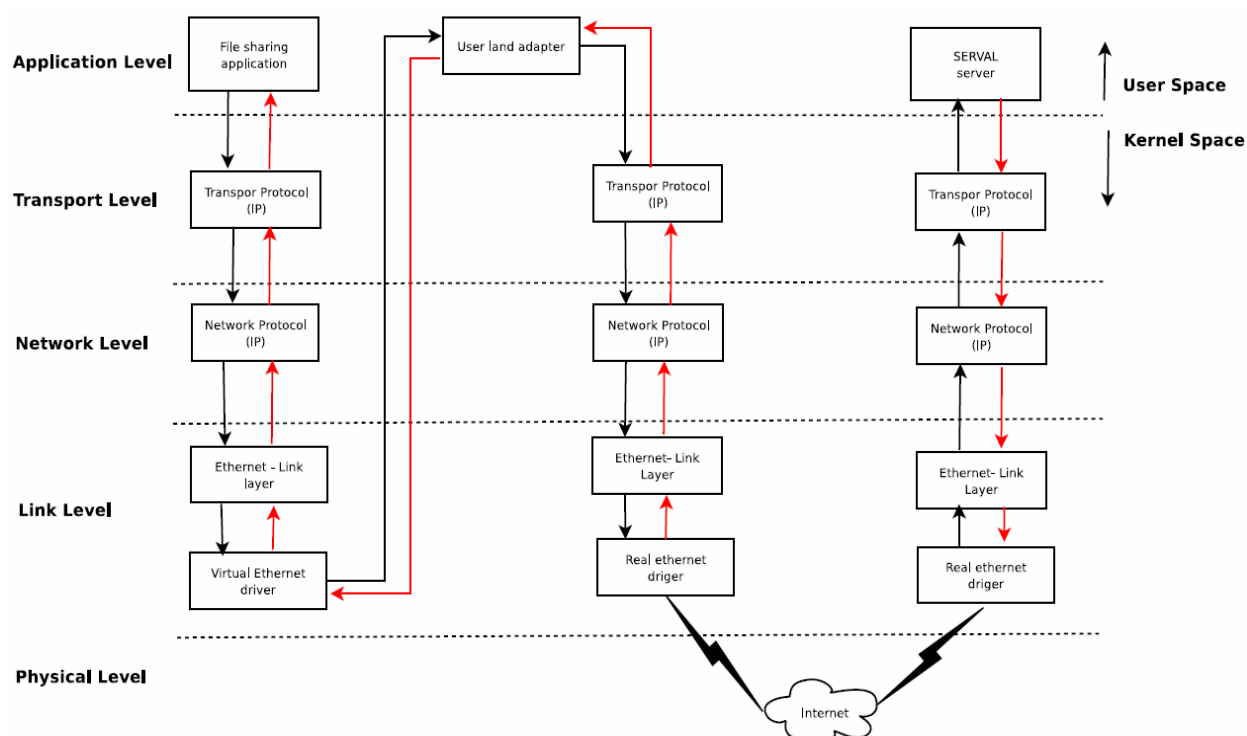


Figura 5.8 - Serval - Relação entre o cliente e o driver ethernet

Este esquema traduz o fluxo do pacote tal como descrito anteriormente.

Cada cliente deste esquema de VLANs sobre Internet, no início da sessão negocia em qual VLAN pretende ser inserido como demonstra a figura 5.9.

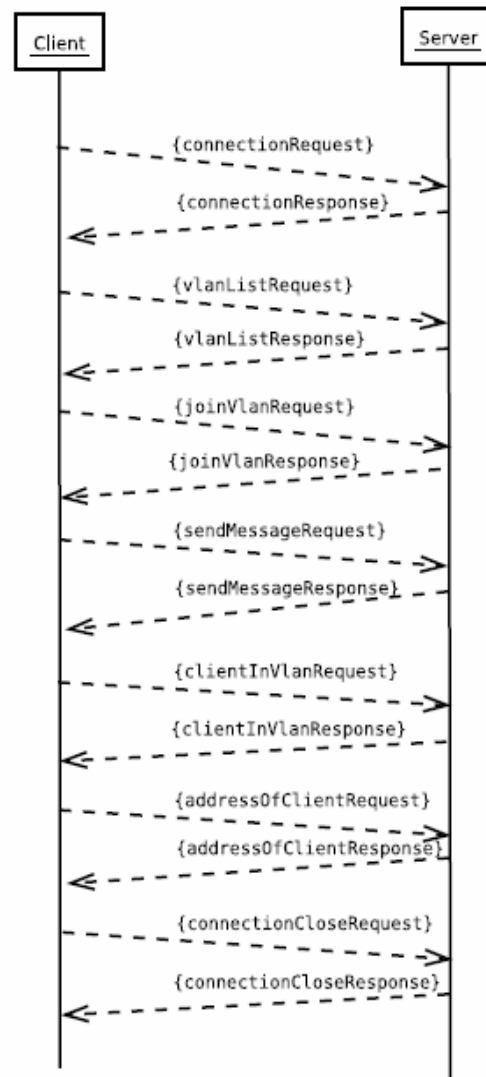


Figura 5.9 - Serval troca de mensagens entre cliente e servidor

Decorre em paralelo a este projecto, na mesma faculdade, um trabalho intitulado WebSwitch.

O WebSwitch, funciona de modo equivalente ao SERVAL, no entanto visa processar na camada de aplicação, apenas os protocolos HTTP, HTTPS, FTP e RTP. Uma nova abordagem no futuro do WebSwitch poderá ter em conta esta nova funcionalidade que se descreve à imagem do SERVAL, sendo integrado no laboratório eCassiopeia, alargando as suas capacidades.

5.3.2 GRE / PPTP

Generic Routing Encapsulation (GRE) é um protocolo que implementa um túnel entre dois locais remotos de uma rede IP.

Os pacotes destinados ao túnel, tipicamente encapsulados com o protocolo IP, são novamente encapsulados por um cabeçalho, GRE, e colocados no túnel com destino ao endereço onde termina o túnel GRE.

Depois de chegarem ao destino, os pacotes são desencapsulados, retirando-se-lhe o cabeçalho GRE e percorrem o caminho com destino ao endereço determinado pelo cabeçalho original, no protocolo de rede, no caso mencionado acima, IP.

Este protocolo permite encapsular protocolos da camada de rede, dentro da camada de rede IP, sendo usado sobre a Internet para dar suporte a VPNs.

O GRE foi desenvolvido pela Cisco e não é orientado ao estado, isto é, os pontos de terminação do túnel não se monitorizam, o que permite que os cliente configurem o túnel sem a necessidade de saber qual a arquitectura do operador, nem o estado do serviço no destino do túnel. A diferença fundamental entre um túnel GRE encriptado e um túnel IPsec é a capacidade do GRE suportar Multicast. Um exemplo seria a utilização de OSPF ao longo de um túnel GRE.

O uso mais frequente destes túneis, é no entanto, associado ao PPTP.

O protocolo PPTP visa criar um túnel, que ligue um cliente remoto a uma VPN. Sendo este túnel a interface com uma rede privada, logo endereçada ao nível da camada 3, e tendo que viajar numa rede pública também endereçada na camada de rede, é necessário esconder da rede pública os endereços privados, onde o tráfego tem origem e se destina. O modo de esconder o endereçamento privado é encaminhar o túnel PPTP dentro de um túnel GRE, fazendo valer a máxima do GRE: “encapsular protocolos da camada de rede dentro da camada de rede IP”.

O PPTP implementa as funcionalidades do protocolo PPP (Point-to-Point Protocol), possibilitando o encaminhamento remoto de pacotes originários de redes como o IP, IPX (Internet Packet Exchange) ou NetBEUI (Network Basic Input/Output System Extended User Interface).

Desenvolvido pela Microsoft, vem desde a versão Windows 95 nos sistemas operativos deste fabricante, pelo que a sua utilização se tornou massiva.

A grande desvantagem desta solução, deve-se à necessidade de iniciar duas sessões por cada cliente remoto, uma primeira no porto 1723 para o túnel GRE e posteriormente a sessão para o estabelecimento do túnel PPTP.

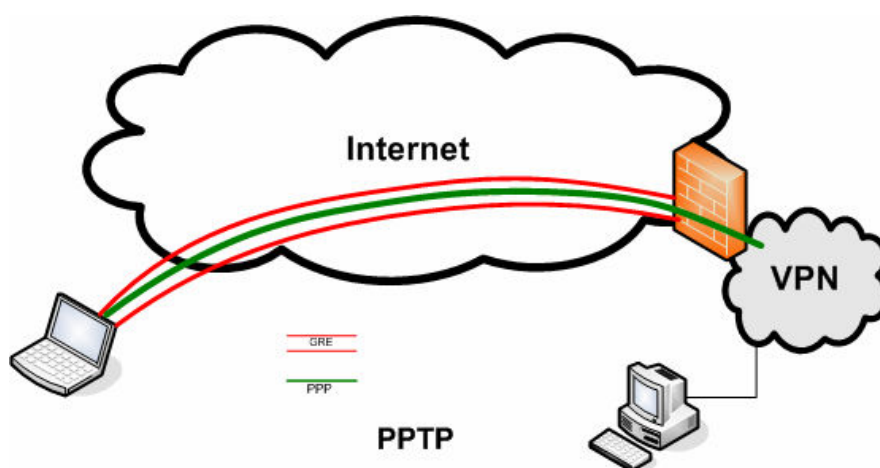


Figura 5.10 - Túnel GRE encapsulando PPTP

5.3.3 L2F / L2TP

O protocolo L2F (Layer 2 Forwarding), foi um dos primeiros protocolos utilizados para implementar VPNs. Tal como o PPTP, o L2F foi desenhado para estabelecer um túnel entre um utilizador remoto e uma rede privada, VPN.

Este protocolo é também proprietário sendo desenvolvido pela Cisco.

A grande diferença entre o PPTP e o L2F, é o facto de o mesmo não depender de IP e, por isso, é capaz de trabalhar directamente sobre tipos de acesso como Frame Relay ou ATM.

Este protocolo utiliza conexões PPP para a autenticação dos utilizadores remotos, sendo a sua grande vantagem a possibilidade de suportar várias sessões de rede, dentro de um único túnel. Com o PPTP, além de ser impossível esta característica, é necessário estabelecer duas sessões por túnel.

O L2F também suporta protocolos de rede diferentes de IP, como o IPX ou o NetBEUI.

O protocolo L2TP (Layer 2 Tunneling Protocol) foi criado pela IETF (Internet Engennering Task Force), com o objectivo de colmatar as falhas do PPTP e do L2F. Utiliza o conceito do L2F e, tal como este, foi desenvolvido para transportar pacotes por diferentes meios como X.25, frame-relay ou ATM, sendo também capaz de suportar protocolos de rede IP, IPX ou NetBEUI.

O túnel L2TP pode ser iniciado de quatro modos diferentes:

1. *voluntary tunnel*
2. *compulsory tunnel – incoming call*
3. *compulsory tunnel – remote dial*
4. *L2TP multi-hop connection*

Os modos 2, 3 e 4 são mediados pelo provedor do acesso, que verifica as credenciais para o estabelecimento do túnel L2TP. De seguida é estabelecida uma ligação PPP entre o utilizador remoto e a VPN onde este se pretende ligar.

O modo 1, voluntário, é o que faz sentido para o caso em estudo, permitindo o estabelecimento de um túnel L2TP entre um router em modo LAC (L2TP Access Concentrator) e um outro em modo LNS (L2TP Network Server) sem a necessidade da intervenção do ISP. O LAC funciona como um cliente que inicia o túnel L2TP e efectivamente reside no sistema como o cliente remoto. O túnel estabelece-se entre o cliente L2TP e o LNS permitindo de seguida o início de uma sessão PPP.

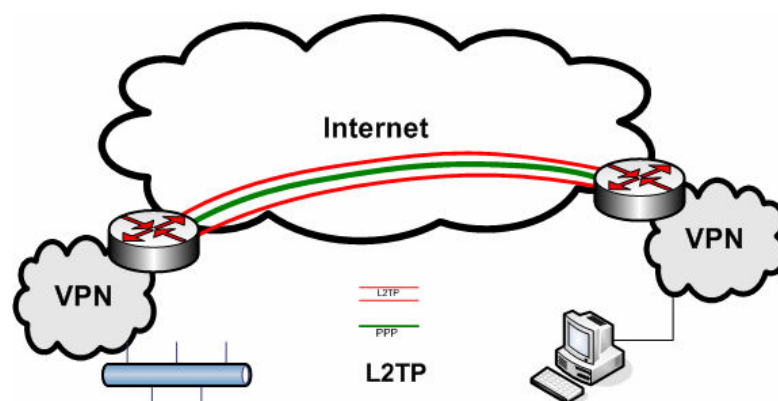


Figura 5.11 - Túnel L2TP

5.3.4 IPSec

Os protocolos PPTP, L2F e L2TP não prevêem qualquer tipo de criptografia ou processamento para tratar chaves criptográficas, o que actualmente é visto como uma falha de segurança.

Para resolver o problema, surgiu um dos mais importantes protocolos já criados, para garantir a segurança da próxima geração IP (IPv6) e que, nos dias de hoje, vem sendo utilizado já com IPv4.

O IPSec permite autenticar e criptografar a comunicação sobre a Internet, entre um utilizador e a sua interface para a VPN.

O protocolo IPSEC permite a utilização em dois modos distintos: no modo transporte, em que apenas a camada de transporte de um pacote IP é autenticado e criptografado; a outra abordagem, permite a autenticação e criptografia de todo o pacote IP, é chamado o modo túnel.

Em modo transporte o IPSec prima por ser mais eficiente do ponto de vista da rede, em modo túnel, promove uma maior protecção contra ataques de monitorização de tráfego que podem ocorrer na Internet, vulgarmente conhecidos como man in the middle.

O IPSec utiliza métodos criptográficos que fomentam a integridade e segurança dos dados como:

- Diffie-Hellman-Key-exchanges, para negociar as chaves criptográficas entre as partes na rede pública.
- Public-key-cryptography, para sinalizar trocas do tipo Diffie-Hellman e garantir a identificação das duas partes evitando, assim, ataques man in the middle.
- DES e outros algoritmos, para criptografar os dados.
- Algoritmos para a autenticação dos pacotes, que utilizam funções de hash.
- Certificados digitais, para validar chaves públicas.

Existem duas formas de lidar com a troca de chaves numa arquitectura IPSec: chaves manuais (manual keying) e Internet Key Exchange (IKE), para manutenção automática de chaves. Enquanto a utilização de chaves manuais pode ser utilizado em VPNs com um número pequeno de sites, o IKE deve ser obrigatoriamente usado em VPNS que suportam um grande número de sites e utilizadores remotos.

O IPSec teve um grande impacto na evolução dos ambientes IP, por implementar modelos de segurança - criptografia, autenticação e troca de chaves - mas a limitação prende-se com o facto de apenas suportar endereçamento IP.

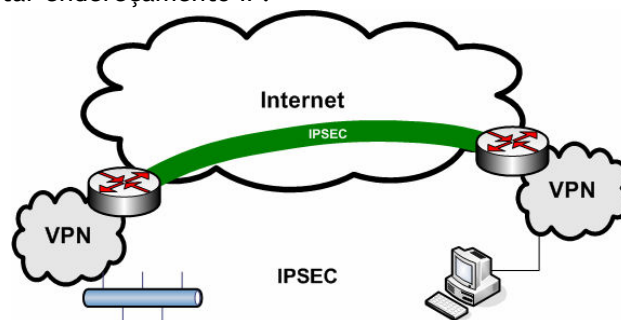


Figura 5.12 - Túnel IPSEC

5.3.5 Conclusão

Tendo em conta as propostas apresentadas e os seus prós e contras, na minha opinião seria a opção mais viável a utilização de túneis IPSEC na arquitectura de um Laboratório Global.

Esta opção levantaria novos desafios como, por exemplo, a negociação das de PSK entre os vários equipamentos de bordo, de modo a iniciarem os túneis IPSEC ou, na perspectiva mais lata, considerar-se a utilização de um core para este tipo de arquitectura, gerido a nível nacional entre os vários associados.

5.4 O Moodle

O Moodle [12] - Modular Object-Oriented Dynamic Learning Environment, é um software opensource de apoio à aprendizagem, executado num ambiente virtual.

Criado em 2001 por Martin Dougiamas, permite fazer a gestão de actividades inerentes ao processo de formação e é cada vez mais uma ferramenta essencial no apoio ao ensino. Utilizado por agentes educativos desde o 3º Ciclo de escolaridade até ao Ensino Superior, permite a criação de ambientes virtuais, que facilitam a aprendizagem colaborativa, permite, de maneira simplificada, que um estudante ou professor, estude ou leccione, uma disciplina on-line.

As actividades mais frequentes são:

- Disponibilizar material para apoio ao estudo;
- Criar Processos de Avaliação: questionários, exercícios, testes, etc.
- Disponibilizar Ferramentas Colaborativas: chat, fóruns, etc.
- Criação de LogBooks Electrónico ou Wiki para organização pessoal.

Torna-se essencial que qualquer uma destas facilidades façam parte do laboratório remoto, alargando as ferramentas de trabalho a disponibilizar aos utilizadores. É lógica a necessidade de consultar informação, durante a realização de trabalhos, bem como no final da sua realização, os alunos sejam avaliados sobre os temas cobertos pela actividade laboratorial.

As ferramentas colaborativas, serão sem dúvida um método de partilha de experiências e opiniões, para a resolução do desafio que é proposto, e as formas de anotar resultados, configurações e explicações, sejam elas wiki ou logbook electrónico são, também uma mais valia.

Assim, com o intuito de economizar tempo na implementação do laboratório remoto e aliando sinergias com a comunidade que desenvolve o moodle, penso que seria importante utilizar estas ferramentas que o moodle disponibiliza. Além de minimizar o tempo de implementação do projecto, usaremos ferramentas testadas globalmente e suportadas por uma grande comunidade de programadores, o que permitirá concentrar os esforços na parte de redes do projecto. Os serviços serão suportados pelos recursos da Faculdade de Engenharia, alocados ao desenvolvimento do projecto moodle, sem que se tornem uma preocupação para o administrador do eCassiopeia.

5.4.1 Autenticação e Controlo de Acesso

A autenticação de utilizadores para controlo de acesso ao laboratório, seria uma ferramenta a aproveitar do moodle. Este, implementa já políticas de permissões de acesso, o que evitaria que os alunos tivessem mais um username e uma password. A ideia seria utilizar as credenciais utilizadas no SIFEUP, e através de uma política de permissões, era barrada ou permitida a entrada do utilizador na ferramenta de acesso ao laboratório. Poder-se-ia utilizar esta mesma funcionalidade para, semana a semana, definir que trabalhos cada utilizador ou grupo de utilizadores poderia realizar, temporizar a realização de questionários sobre a actividade laboratorial ou até mesmo temporizar a realização do trabalho.

5.4.2 Calendarização

O moodle permite uma arquitectura modular, na qual grande parte dos módulos são do domínio público, permitindo facilmente adicionar funcionalidades a um sistema já em produção.

Uma pesquisa a esses módulos, permitiu encontrar aplicações que disponibilizam um calendário e a possibilidade de agendar eventos no dia seleccionado. Esta funcionalidade é essencial no laboratório remoto que se pretende implementar, pois possibilita a reserva de equipamentos para determinado dia e hora, com vista a garantir exclusividade no acesso ao material, pelo aluno ou grupo de alunos que realiza a reserva.

Este módulo, mais uma vez, permite libertar o sistema desta tarefa e possibilita outras funcionalidades como alertar, com um pop-up, o utilizador do agendamento se este se encontrar ligado ao sistema. Se por sua vez o utilizador não estiver ligado ao moodle, é enviado um e-mail a notificar esse utilizador.

5.4.3 Segurança

As rotinas de segurança, relacionadas com acessos indevidos a conteúdos, questionários ou até falsificação de identidade, são preocupações constantes da comunidade de desenvolvimento, pelo que, adoptando o moodle como ponto de acesso ao laboratório, estas questões ficam resolvidas e sempre acompanhadas pela equipa de suporte. Mais uma vez esse trabalho não necessita de ser objecto de análise regular da equipa de desenvolvimento do laboratório remoto, pois está à partida assegurado pelo moodle.

5.4.4 Ferramentas Colaborativas

Grande parte dos trabalhos executados no laboratório são realizados por grupos de alunos e, por vezes, é necessária a realização de testes entre grupos distintos que pretendem comunicar. Esta interacção de vários elementos, deverá ser suportada por ferramentas que permitam a

comunicação em tempo real, para a parametrização e sincronização de procedimentos de teste ou até para partilha de soluções, para a resolução do trabalho.

As ferramentas para apoio ao trabalho de grupo e à partilha de opinião e experiências, entre membros que executam as mesmas tarefas, são também abordadas pelo moodle. Deste modo, de forma simples, é possível implementar estes serviços de comunicação síncrono e em tempo real, os chats, ou então fóruns que permitem a comunicação assíncrona e a consulta de soluções para problemas, que outros elementos possam já ter resolvido e cuja experiência pretendam partilhar.

A integração do moodle com laboratório, permite adicionar de modo prático ambientes de comunicação virtual ao aCassiopeia.

5.4.5 Ferramentas de Organização pessoal e de grupo

Um trabalho laboratorial, é por norma precedido por uma exposição dos conhecimentos adquiridos, pelo que, maioritariamente, é necessário realizar testes que comprovem os conceitos teóricos que estão na base da actividade laboratorial, confrontando-se resultados esperados e alcançados. Por norma, é necessário apelar à crítica dos alunos para justificarem discrepâncias analisadas, realçando o domínio da matéria pelos executantes. Este registo de experiências e resultados, das configurações ou cálculos teóricos efectuados, bem como o resumo crítico de resultados, terá que ser anotado pelos alunos ao longo do trabalho.

A utilização de cadernos de notas, mais conhecidos por logbooks, é sempre aconselhada pelos docentes, apelando a adopção de metodologias de trabalho correctas. Deste modo também no moodle se podem implementar Wiki's e LogBooks Electrónicos, que poderão apoiar os alunos nas suas tarefas.

5.5 Interface Web

A actual interface web que permite o acesso ao laboratório, foi desenhada em 2003.

O tempo que entretanto decorreu, permitiu o aparecimento de novos paradigmas de programação web e, acima de tudo, novas tecnologias para serviço web. Na realidade o estudo que efectuei sobre esta interface e a minha experiência em tecnologias web, leva-me a propor uma nova interface, criada de raiz, de modo a não condicionar a evolução futura do projecto do laboratório remoto.

Segundo o meu pensamento, é necessário criar uma interface capaz de ser desenvolvida por módulos, isto porque a intenção de pôr em produção o laboratório remoto, será no âmbito de trabalhos futuros de dissertação das próximas gerações de alunos.

Sendo este trabalho complexo e trabalhoso, que apela ao uso de várias tecnologias web, HTML, SMARTY, PHP, JAVASCRIPT, JAVA APPLET, Bases de Dados e conhecimentos de Redes IP, poderá ser difícil encontrar pessoas com todas estas valências e, mais preocupante, será definir a quantidade de trabalho adequado ao tempo que as dissertações de Bolonha implicam.

Proponho então que o trabalho seja desenvolvido em torno da base, isto é, os módulos que se venham a implementar serão como uma espécie de função matemática, cujo input ou output, ou ambos sejam a base de dados.

5.5.1 A Base de Dados

O modelo da base de dados actualmente implementado é bastante completo. Foram definidos campos que actualmente não são usados, no entanto poderão, no futuro, ser convenientes. Poderá ser caso a caso, conforme a evolução do projecto, necessário adicionar novos campos. Durante o desenvolvimento que fiz do trabalho senti essa necessidade, mas penso que de modo geral, grande parte dos campos necessários estão já definidos no modelo implementado.

A actual base de dados, usa a massificada tecnologia mysql, e considero que não haverá necessidade de migrar para outro tipo de base de dados. É opensource, globalmente usada, continua a ter suporte de uma comunidade de desenvolvimento e existe uma ferramenta de administração também opensource que facilita a sua gestão, o phpmyadmin.

Apesar da sua estrutura ser boa e consistente, é possível definir “vistas” que possibilitam diminuir a necessidade de “queries” mysql e que por sua vez permitirá um código php menos complexo. Assim, apelo à utilização deste tipo de tabelas na implementação da proposta que descrevo com o intuito de minimizar a complexidade dos algoritmos php.

Dada a consistência da Base de Dados, proponho que se possam criar módulos de trabalho que interajam sempre com esta, por exemplo, um módulo que interprete o pedido do utilizador e o escreva na base de dados. No futuro poderemos criar mais um módulo de trabalho, que consulte a base de dados e implemente o pedido. Este tipo de fluxo permitirá o desenvolvimento dos módulos que incluirei na minha proposta.

5.5.2 Web Design versus PHP Scripting

A actual implementação da interface web, é ineficiente do ponto de vista de organização do código PHP e do design. Deste modo é impossível pensar em desenhar uma nova interface, mais actual, sem ter que reprogramar os scripts php. Também utiliza frames, solução que actualmente não é aconselhada e que saiu mesmo de moda.

O que proponho para o novo trabalho a implementar, é a separação, à partida, destas duas vertentes do trabalho. Assim, poderemos pensar em renovar a interface, independentemente do código que suporta a aplicação do laboratório remoto e, mais importante que isso, será possível implementar novos módulos, sem a preocupação de conceitos como web design e usabilidade.

Esta diferenciação de linguagens de programação que possibilitam ao programador libertar-se do pensamento estético, é há muito objecto de estudo, existindo já tecnologias que permitem esta diferenciação, funcionando elas como interface entre php e html.

Utilizado largamente por profissionais da web, o Smarty permite colocar tags no código HTML, que depois serão substituídas por valores que resultam da execução de um script PHP. O

Smarty tem também um pequeno interpretador de comandos, que permite implementar ciclos condicionais em função do resultado PHP que lhe é passado.

A utilização desta tecnologia, permitirá também programar trabalhos a desenvolver por pessoas distintas, em trabalhos de dissertação diferentes, desde que, à partida, se definam quais serão as tags partilhadas pelos dois trabalhos.

Assim sendo, poder-se-á atribuir estas duas vertentes de trabalho, a pessoas cujas orientações académicas se adequem ao trabalho que se pretende alcançar, de modo a tornar o laboratório remoto robusto.

5.5.3 Módulos a Integrar

O estudo que desenvolvi sobre o trabalho eCassiopeia, permitiu-me detectar algumas deficiências no código que implementa as rotinas, que suportam o funcionamento do laboratório remoto e que seria de bom grado corrigir. Também me permitiu organizar mentalmente a sequência de operações necessárias à criação, montagem e configuração de um exercício laboratorial. De acordo com a minha proposta, para desenvolver a nova arquitectura como um conjunto de módulos que interagem sempre com a base de dados, e conhecendo a sequência de operações necessárias, proponho a criação de três módulos que serão a base de suporte do laboratório remoto.

5.5.3.1 Módulo: Novo Exercício

Este módulo permitirá ao administrador, professor ou até mesmo aos alunos, criarem um novo exercício. Assim, é necessário desenvolver um applet que permita ao utilizador desenhar o exercício, isto é, de uma lista de equipamentos disponíveis, carregados da base de dados e utilizando o rato no conceito “drag’n’drop”, largar os equipamentos na área de desenho e definir o modo como se ligam. De seguida, existirá um interpretador do desenho que decifrará o esquema de ligações e o guardará na base de dados.

Teremos então uma função com a seguinte macro *function new_exercise (input array equipment_on_lab[], output array links_to[])* de acordo com a figura 5.13.

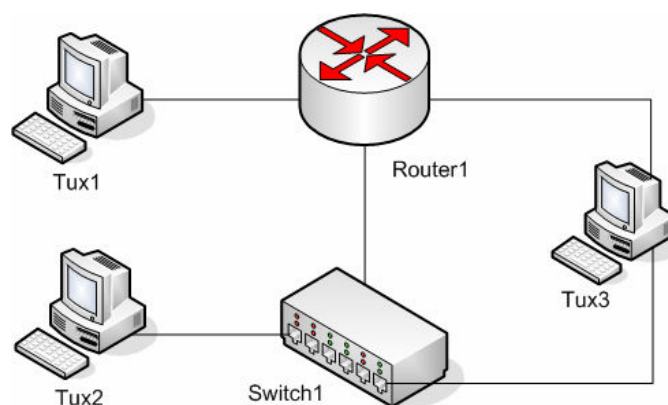


Figura 5.13 - Esquema ligação de um exercício

O caso que a figura representa teria como valores de entrada e saída:

- *array equipment_on_lab[]* = {Tux1, Tux2, Tux3, Router1, Switch1}
- *array links_to[]* = {{Tux1, Router1}, {Tux2, Switch1}, {Tux3, Router1}, {Tux3, Switch1}, {Router1, Switch1}}

Uma pesquisa sobre tecnologias que permitem o desenvolvimento de uma interface web para drag and drop, demonstrou que se poderá usar applets JAVA, ou então DHTML, uma combinação de HTML e JavaScript.

As seguintes referências, disponibilizam exemplos interessantes opensource, cujo código poderá ser reutilizado e servir de base a este módulo:

- http://demos.openrico.org/demos/drag_and_drop_simple
- <http://www.javaworld.com/javaworld/jw-03-1999/jw-03-dragndrop.html>
- <http://opencomponentry.com:8080/tacos/ajax/DragAndDropExample.html>

5.5.3.2 Módulo: Matriz Ligações PatchPanel

O módulo que descrevo, foi desenvolvido por mim nas melhorias que implementei ao sistema já em vigor. No entanto, programei este processo como uma rotina executada em série, para o procedimento que estava em produção. A minha proposta é que este trabalho seja um módulo independente e que se possa melhorar continuamente, em função dos equipamentos e alterações que se possam vir a adoptar.

Uma das limitações que a rotina que implementei tem, é permitir apenas o suporte de duas interfaces ethernet, no Tux3 de cada bancada e apenas uma interface nos Tux1 e Tux2. É necessário assim tornar este módulo global e não orientando apenas às condições do laboratório actual. É essencial dotar também esta rotina, da análise aos routers de wan, para no futuro poder suportar o equipamento da nortel, que simula ambientes wide área network.

A macro terá então uma aparência do tipo *function patchpanel_matrix (input array links_to[], output array PatchPanel_Port_Matrix[])* e terá que procurar para cada equipamento, quais os portos associados no patch panel, verificar se estão livres, quais os equipamentos que se ligam ao equipamento em análise, procurar também esses portos no patch panel e então criar a matriz de comutação de portos a inserir no patch panel, para que remotamente se possam implementar as ligações para o exercício em causa.

5.5.3.3 Módulo: Carregar exercício

O módulo que descrevo, será o mais complexo e responsável pelas rotinas essenciais à realização da actividade laboratorial remota.

Sempre que um utilizador invocar a realização de um exercício, este módulo será responsável por:

- Invocar o módulo da matriz de ligações do patch panel, para que no patch panel se implemente o esquema de ligações, de acordo com o exercício pretendido
- Carregar, da base de dados, a arquitectura e esquema de ligações com endereçamento à consola dos equipamentos, permitindo que o utilizador clique na imagem do equipamento e despolete uma aplicação de telnet a esse equipamento.

De um modo geral implementará o efeito inverso do que se pretendia com o módulo de novo exercício. Assim a macro será *function load_exercise (input array links_to[])* cujo output será a informação a passar ao smarty, para criação do esquema de ligações do exercício.

Uma sugestão pessoal que no entanto ficará à responsabilidade dos próximos intervenientes, seria adicionar uma nova tabela na base de dados, a qual guardaria informação sobre a localização dos objectos e a topologia das ligações de cada exercício. No entanto, dependendo da tecnologia que se vá utilizar, ter-se-á que avaliar a viabilidade desta sugestão, de modo a reverter o processo e permitir desenhar o exercício através da informação retirada da base de dados.

5.5.3.4 Módulo: Controlo de Sessão

O controlo de sessão, é uma técnica utilizada nos dias de hoje para monitorizar a actividade do utilizador nomeadamente nas chamadas web vpn. O utilizador liga-se a uma vpn, através de um browser que inicia um túnel https e que de modo encriptado comunica com a vpn. Quando se utilizam aplicações dedicadas a essa finalidade, o controlo da sessão é feito por essas mesmas aplicações. No caso de um browser, a maneira de se saber, do lado do servidor, quando o utilizador pretende terminar a vpn ou não, é o uso de javascript que abre um pop-up de controlo. O utilizador terá nessa janela de controlo a opção de logout, ou no caso de fechar a janela de controlo, é alertado que ao fechar a janela terminará a ligação à vpn.

Com vista à implementar o controlo de acessos de utilizadores aos equipamentos, garantindo que o acesso é exclusivo a esse utilizador ou grupo de utilizadores, é utilizado o iptables do linux, para permitir apenas tráfego vindo do IP do utilizador ou grupo de utilizadores que tem a reserva no momento. Se não tivermos controlo sobre a sessão do utilizador, este pode abandonar o exercício sem efectuar o logout e ficará o registo do seu ip no iptables, o que se torna um problema grave de segurança.

Assim este módulo implementará as seguintes funções:

- Testar se o JavaScript está activo no browser do cliente. Se estiver, deixará prosseguir o exercício, se não estiver, pede ao utilizador para activar, sob pena de não efectuar o exercício
- Se o teste de JavaScript der positivo, o cliente será redireccionado para a página do exercício e será iniciada uma janela popup de controlo. Esta janela terá a opção de logout do utilizador e o tempo restante para terminar a reserva agendada, ou seja o tempo restante para terminar o exercício.

- Ao esgotar-se o tempo, o JavaScript invocará automaticamente o logout.
- Nos casos em que o cliente pretenda fechar a janela de controlo, será alertado que se continuar com a operação terminará a sessão, ou seja o logout é invocado durante o processo do fecho da janela. Todas estas funcionalidades são possíveis com o uso de JavaScript.
- A opção de logout, sendo a ordem dada pelo utilizador ou não, invocará no servidor um url, que implementa uma rotina que limpará o IP do cliente do iptables e bloqueará o tráfego vindo desse utilizador.

Garantir-se-á assim a segurança e exclusividade no acesso directo aos equipamentos.

5.6 Acesso remoto

O acesso remoto aos equipamentos, actualmente faz-se via um servidor de terminais, isto é, todos os equipamentos estão ligados via porta série RS232, ao servidor de terminais que suporta telnet, assim faz-se telnet a um porto específico do servidor de terminais, que despoleta uma sessão à consola do equipamento pendurado nesse porto. Este servidor apenas suporta uma única sessão a cada porto, pelo que apenas um único aluno poderá efectuar as configurações.

Alargando o pensamento e permitindo o trabalho em grupo, proporcionando a que todo o grupo acompanhe a realização do exercício e das suas configurações, torna-se necessário implementar a possibilidade dos elementos visualizarem a progressão do trabalho. Existe para isso, uma software opensource para trabalho colaborativo, que suporta múltiplas sessões Telnet ou SSH a uma mesma consola, o Conserver [13].

Este serviço, que a partir deste momento trataremos por servidor de consolas, permite que os utilizadores ganhem acesso, de forma transparente, à CLI (Command Line Interface) dos equipamentos reais, para os configurar.

5.6.1 Servidor de Consolas e Servidor de Terminais

O objectivo destes dois servidores, é possibilitar o acesso partilhado e transparente da CLI do equipamento a um grupo de alunos. Deste modo, o servidor de consolas inicia uma sessão de telnet ao servidor de terminais, no porto do equipamento que se pretende aceder, pelo que existirá, a partir desse momento, uma consola aberta para esse equipamento no servidor de consolas. O servidor de terminais apenas suporta uma sessão a cada porto, pelo que nesse momento temos a sessão ocupada e não se poderia ligar mais ninguém ao equipamento.

Os alunos poderão no entanto ligar-se ao servidor de consolas e acederem à consola aberta para o equipamento. Em cada momento apenas um dos utilizadores poderá aceder à consola em modo escrita, todos os outros apenas poderão visualizar o que o detentor dos direitos de escrita está a executar.

Este software permite partilhar, de forma transparente, as consolas dos equipamentos activos que estejam ligados ao servidor de terminais.

5.6.2 Principais características do servidor de consolas

O servidor de consolas apresenta duas características importantes, para o contexto do laboratório remoto: a possibilidade de ter múltiplas sessões na mesma consola e o registo em ficheiros de log, toda a informação que sai pelas portas das consolas dos equipamentos ligados.

Estas características são importantes, uma vez que facilitam o trabalho colaborativo

Como o sistema aceita múltiplas sessões para a mesma consola e o controlo da sessão pode ser comutado entre as várias sessões, vários utilizadores podem colaborar, em tempo real, para completarem o mesmo exercício. É possível, por exemplo, a um aluno, que tem uma dúvida, pedir ajuda ao seu colega, que entra na mesma consola e ganha o controlo da sessão.

Os ficheiros de log dos vários dispositivos, também podem ser usados como uma ferramenta de aprendizagem, pois os alunos podem aceder ao registo das várias sessões, para ver exemplos de comandos utilizados em configurações anteriores.

Estes mesmos logs poderão ser usados para que o aluno tire dúvidas. Imaginemos um caso em que o aluno faz uma configuração mas não chega ao resultado pretendido. Poderá recolher os logs e apresentar ao docente, para que este possa analisar e detectar o problema da configuração.

Poderá também ser uma ferramenta essencial, para que o administrador do sistema possa detectar a razão para determinadas falhas e perceber se estas foram premeditadas ou fortuitas, perceber quando existe acesso indevido aos terminais e quais as falhas de segurança.

5.6.3 Telnet ou SSH ?

O acesso remoto é sempre um ponto crítico em questões de segurança. A vulnerabilidade aumenta se estiverem a ser utilizadas aplicações como o telnet, que não utiliza comunicação encriptada.

Ao contrário do Telnet, o SSH possibilita a comunicação segura através de canais inseguros, encriptando todo o tráfego e fornecendo vários níveis de autenticação.

O servidor de terminais apenas suporta o protocolo telnet, pelo que na implementação actual se utiliza o telnet através da Internet, para configurar os equipamentos. A solução para este problema também passa pelo Conserver, além das funcionalidades que até aqui já caracterizei.

O servidor de consolas suporta os protocolos telnet e ssh, assim funcionará como uma gateway aos dois protocolos. Internamente a máquina fará, através da rede privada de gestão do laboratório, telnet ao servidor de terminais. Os utilizadores remotos utilizarão o protocolo ssh para aceder à consola partilhada.

Aplicando este novo conceito na próxima arquitectura, chamo na mesma a atenção, que se poderá manter a utilização da applet ssh em Java, libertando os alunos de instalarem uma aplicação ssh para realizar os trabalhos.

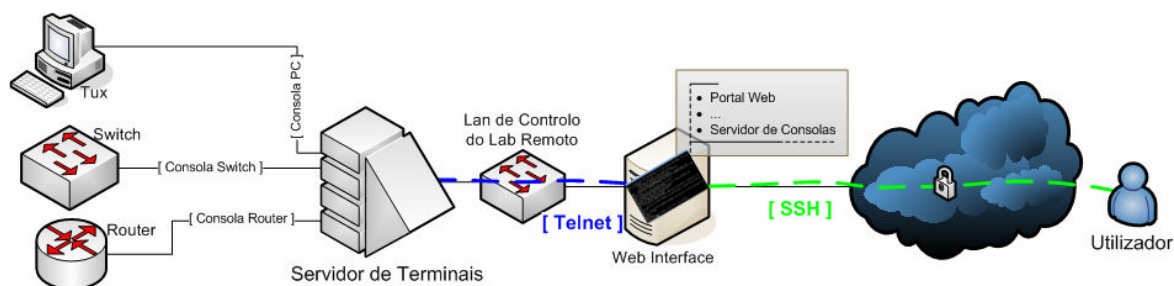


Figura 5.14 - Arquitectura para o Servidor de Consolas

5.7 Servidor de Configurações

Sempre que um aluno pretende iniciar um trabalho, seria interessante que o equipamento estivesse limpo de configurações, no entanto não é o que acontece com o sistema actual.

Neste capítulo pretendo resolver este problema.

Antes de começar um novo exercício, os equipamentos devem ser preparados com a configuração inicial default ou então, se pretendido pelo docente, com alguma configuração específica que facilite a realização do trabalho aos alunos.

Será necessário, para isso, desenvolver um algoritmo a instalar no servidor web, que invoque comandos nos equipamentos, através do servidor de consolas ou de terminais, para os reiniciar. A configuração de arranque depende do que for pretendido pelo docente, existindo ou não um ficheiro com essas configurações. Assim, após o pedido para realizar um exercício, os equipamentos sofrem reinicialização e poderão ou não receber comandos para as configurações de base.

De um modo geral, será um programa que poderá ser invocado via php, que permitirá configurar e invocar comandos a aplicar aos equipamentos ou grupos de equipamentos, possibilitando executar acções em instantes definidos;

5.8 Microsoft ConferenceXP

O ConferenceXP é uma ferramenta desenvolvida para suportar o trabalho colaborativo. Desenvolvido pela Microsoft com licenciamento OpenSource, este serviço além das já conhecidas ferramentas de conversação através de texto, Chats e Instant Messaging, permite a comunicação

áudio e vídeo entre os participantes. Este tipo de tecnologia potencia sem dúvida a interacção dos grupos de trabalho.

Esta ferramenta além do suporte de conferência, é dotada de uma denominada ConferenceXP Apresentator. O ConferenceXP Apresentator, permite aos utilizadores partilhar diapositivos que possam apoiar a resolução dos exercícios, sendo que os membros do grupo de conferência recebem os slides em tempo real na sua máquina.

Esta ferramenta foi desenvolvida a pensar num ambiente de ensino remoto, em que o professor utiliza diapositivos para leccionar as aulas e os alunos recebem essa mesma informação, no entanto poder-se-á usar como partilha de informação, para a resolução do exercício.

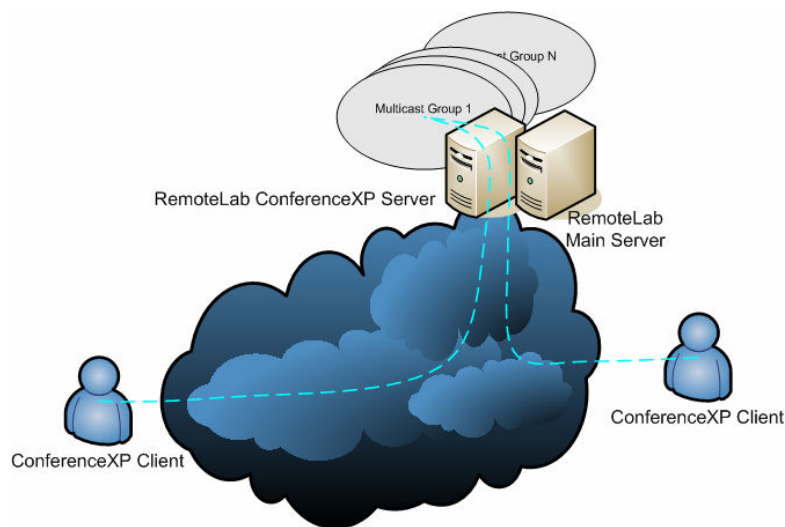


Figura 5.15 - Arquitectura do Serviço ConferenceXP

Adicionar esta nova funcionalidade ao Laboratório, como proponho, não implica qualquer reorganização da arquitectura proposta, apenas será necessário adicionar uma nova máquina com Windows Server à solução, e instalar o ConferenceXP Venue Service, de acordo com a figura 5.15.

5.9 Conclusões

As propostas desenvolvidas neste capítulo, permitem-nos formar uma macro arquitectura, que disponibilizará os serviços estudados e pretendidos para o laboratório remoto que se deseja implementar. Assim, poderemos visualizar como todos os módulos se interligarão, constituindo a arquitectura que proponho na figura 5.16.

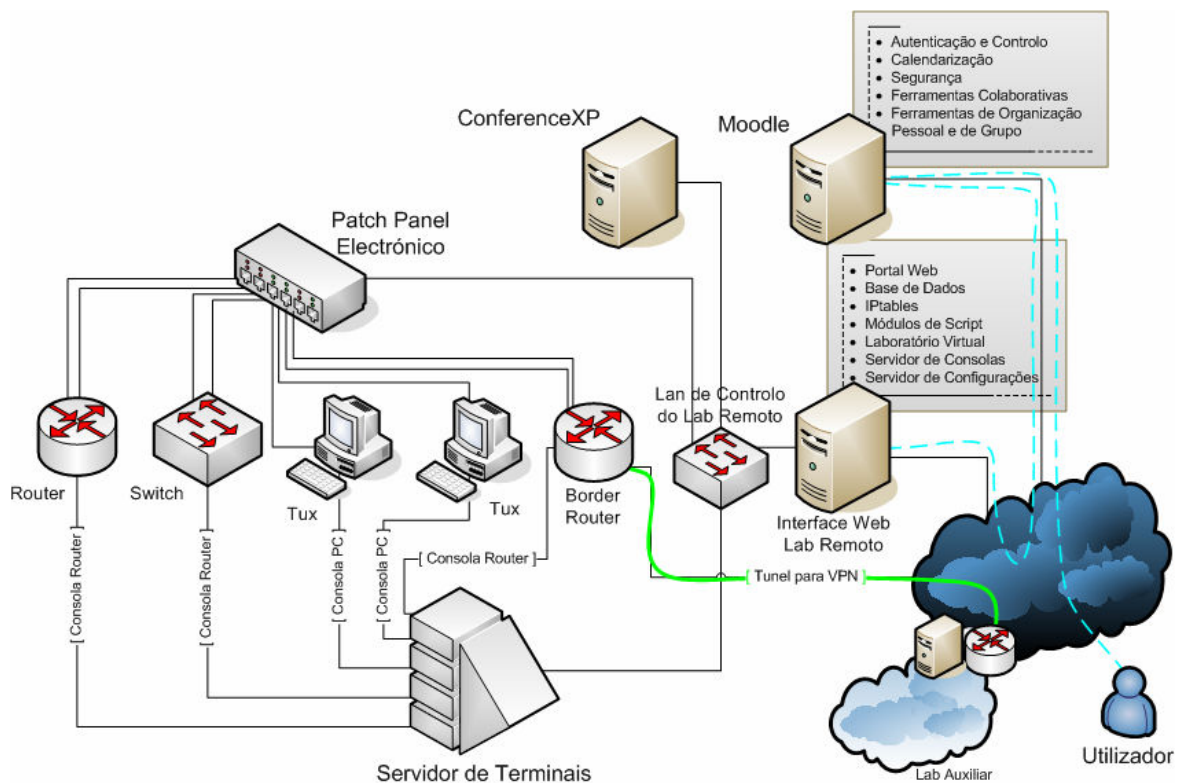


Figura 5.16 - Visão Macro da Arquitectura Proposta

Capítulo 6

6. Conclusão

Vindo da área de redes, na qual já exerço a minha actividade profissional, fui atraído a este trabalho pelo título sugestivo de “Laboratório Remoto de Redes”. Com o culminar do trabalho constato, que de redes pouco trabalho tem. Na realidade este trabalho tem a sua essência na área aplicacional e nos serviços Web.

Motivado pelo conceito do acesso remoto e pela possibilidade de desenvolver um trabalho que dará uma nova força ao ensino de redes na FEUP, da qual fui aluno, continuei afecto ao trabalho, apesar de ver defraudadas as minhas perspectivas iniciais de um trabalho de redes.

Encontrei muito por fazer, numa área do ensino que pessoalmente considero a tendência futura, para a qual devem as nossas universidade caminhar. Nos dias de hoje os conceitos de mobilidade são reais, pelo que o ensino terá que caminhar também nesse sentido,

O eCassiopeia, quando foi pensado, em 2003, teve como fundamento, com certeza, uma visão muito futurista do ensino, no entanto o conceito “à distância” é nos dias de hoje já uma realidade, e é pena que em cinco anos tudo esteja igual. A implementação não saiu do papel que alguém elaborou, até mesmo regrediu. O estudo que fiz da implementação actual e os contactos que efectuei com a pessoa que acompanhou a sua criação, deram-me a conhecer que muito do trabalho feito se perdeu, pois não havia backup de um trabalho com duração de 2 anos.

Assim, o objectivo que se pretendia inicialmente com a minha abordagem ao eCassiopeia, que seria alargar as suas funcionalidades a um segundo laboratório de redes, que se estava a criar para a faculdade, tiveram que se ajustar ao estado em que o encontrei a solução.

Os novos objectivos permitiram-me porém conhecer variadas ferramentas opensource, que possibilitarão implementar novos serviços e funcionalidade ao eCassiopeia. Permitiram-me também conhecer equipamentos, desenhados para criar o dinamismo que se pretende num projecto como o eCassiopeia, como foram o servidor de terminais e patch panel electrónico. Este último cativou o meu interesse, dada a flexibilidade que as suas funcionalidades proporcionam, numa rede de computadores.

O fruto deste trabalho para a faculdade, será a consola que programei para controlar o patch panel electrónico e a arquitectura que defini, pensando já na sua integração com moodle no qual a faculdade tem já apostado. Ficará também algum trabalho de programação, alguns algoritmos que implementei e que poderão ser utilizados na arquitectura que proponho.

Serei sem dúvida o maior beneficiário do know-how que adquiri sobre este tipo de sistema para acesso remoto, no entanto para que não se perda o estudo que realizei e a faculdade

possa também ganhar com isso, disponibilizo-me a acompanhar futuros alunos que venham a implementar a minha proposta. Será um desafio pessoal, acompanhar a realização de algo que idealizei e um agradecimento para o que a faculdade fez por mim, como engenheiro e como pessoa.

Referências Bibliográficas

- [1] Alhalabi, B. et al., "Remote Labs: An Innovative Leap in Engineering Distance Education", Proceedings of the ACE 2000, IFAC/IEEE Symposium on Advances in Control Education, Gold Coast, Australia, December 2000.
- [2] MONTEIRO, Edmundo, Fernando Boavida, Engenharia de redes informáticas, FCA - Editora de Informática, 2000.
- [3] RUSSELL, Rusty, "Linux iptables HOWTO". Documento disponível em <http://netfilter.samba.org>
- [4] PEREIRA, Fernando, Linux, curso completo, FCA - Editora de Informática, 2000.
- [5] APACHE, Servidor Web. Informação e software disponível em <http://www.apache.org>
- [6] MySQL, Base de Dados. Informação, software e Manual de Utilização disponível em <http://www.mysql.com>
- [7] phpMyAdmin, Administrador Web para Bases de Dados MySQL. Software disponível em http://www.phpmyadmin.net/home_page/index.php
- [8] JTA - Telnet/SSH for the JAVA(tm) platform. Software e Manual de Instalação e Configuração disponível em <http://www.javassh.org/space/start>
- [9] APCON, Patch panels automáticos produzidos pela Apcon. Informação sobre preços gama baixa e características dos equipamentos disponível em <http://www.apcon.com/patchpanels.html>
- [10] LinkXchange, Patch panels automáticos produzidos pela Gillaspay Associates, Inc. Informação sobre equipamentos disponível em <http://www.networktesttools.com>

- [11]GNS3, Simulador de ambientes de rede virtuais. Software para download, Informação de instalação, configuração e integração com Microsoft LoopBack disponível em <http://www.gns3.net>
- [12]Moodle, Gestão para E-learning. Informação sobre blocos de software disponíveis e informação sobre configuração disponível em <http://moodle.org/>
- [13]Conserver, Servidor OpenSource para CLI. Software e Informação de Instalação disponível em <http://www.conserver.com>
- [14]Vitels, Virtual Internet and Telecommunications Laboratory of Switzerland. Informação disponível em <http://www.vitels.ch/events/VID2003/index.php>. Acesso em 2/Março/2008.
- [15]ReLI, Remote Laboratory Infrastructure
Sicker, D.C et al. Assessing the Effectiveness of Remote Networking Laboratories
Documento disponível em
<http://ieeexplore.ieee.org/iel5/10731/33854/01612279.pdf?arnumber=1612279>
- [16]Stiubiener, I. et al. NETLAB: A Framework for Remote Network Experiences. Documento disponível em fie.engrng.pitt.edu/fie2006/papers/1057.pdf
- [17]Stiubiener, I. et al. NETLAB: A Framework for Remote Network Experiences. Documento disponível em
<http://ieeexplore.ieee.org/iel5/4141586/4084515/04141704.pdf?isnumber=4084515&prod=CNF&arnumber=4141704&arSt=749&ared=755&arAuthor=Stiubiener%2C+I.%3B+Ruggiero%2C+W.V.%3B+Silveira%2C+R.M.%3B+Korolkovas%2C+I.%3B+Skopp%2C+S.%3B+Meiler%2C+C.>
- [18]TIDIA-AE. Informação disponível em <http://tidia-ae.incubadora.fapesp.br/portal>
- [19] *Ji Hua*, WEB ENABLED REMOTE LABORATORY (R-LAB) FRAMEWORK. Documento disponível em fie.engrng.pitt.edu/fie2003/papers/1359.pdf
- [20]Stiubiener, I. et al. WEB ENABLED REMOTE LABORATORY (R-LAB) FRAMEWORK. Documento disponível em doi.ieeecomputersociety.org/10.1109/FIE.2003.1263345
- [21]Deniz, D.Z et al. A novel approach to remote laboratories. Documento disponível em ieeexplore.ieee.org/iel5/8925/28250/01315161.pdf?tp=&isnumber=&arnumber=1315161
- [22]IV - LAB: Interactive Virtual Lab. Informações disponíveis em research.microsoft.com/conferencexp/research_relatedpapers.aspx

[23]ConferenceXP. Software e Informações de Instalação e Configuração disponíveis em <http://research.microsoft.com/conferencexp>

[24]Coelho, Paulo, Tese Arquitectura do Laboratório de Redes Remoto: eCassiopeia, FEUP, 2001

[25]Microsoft LoopBack. Informações de Instalação e Configuração para Windows XP em <http://support.microsoft.com/kb/839013>

[26]Network Simulator 2, Informações disponíveis em <http://www.isi.edu/nsnam/ns/>

Apêndice A – Consola Patch Panel

```
#!/usr/local/bin/perl
#
# Program to open the password file, read it in,
# print it, and close it again.

use Net::Telnet;
use Switch;
$quit=0;

#-----#
#                               #
#                               #
#-----#

print "For HELP, write '?'\n";

while ($quit == 0)
{
    $cmd = &promptUser("Patch-Panel>");

    switch ($cmd)
    {
        case '?' { print "? - This option.\n".
            "a - Assign port x to y. x and y [01-32].Ex: a 01
32.\n".
            "g - Get Current Port Assignments.\n".
            "s - Store current settings as preset.\n".
            "d - Set factory defaults.\n".
            "e - Enable LAN port.\n".
            "n - Disable LAN port.\n".
            "l - Lock Serial/LAN ports.\n".
            "u - Un-Lock Serial/LAN ports.\n".
            "p - Lock front panel.\n".
            "r - Un-Lock Front Panel.\n".
            "f - Report Firmware Revision.\n".
            "m - Report Model, S/N & Date of Manufacture.\n".
            "q - Quit.\n";
        }

        case /^a\s\d\d\s\d\d$/
        {
            $cmd =~
s/(.)(.)(.)(.)(.)(.)(.)(.)/\3\4\6\7/;

            executeCmd("\n\\\/\|P".$cmd."\n");
        }

        case /^g$/ { $response =
execCmdReturnResponse("\n\\\/\|S\n");

            # $response=
            "///|S0201000013000000000000014051200002100000017000000000000000000000";
            @links =
createLinkMatrix($response);

            foreach (@links) {print $_ .
"\n";}

        }

        case /^s$/ { executeCmd("\n\\\/\|G01FFFFFFFF\n"); }
        case /^d$/ { executeCmd("\n\\\/\|OF\n"); }
        case /^e$/ { executeCmd("\n\\\/\|OE10\n"); }
        case /^n$/ { executeCmd("\n\\\/\|OE00\n"); }
```

```

        case /^l$/      { executeCmd("\n\\\/\\\/OL1234\n"); }
        case /^u$/      { executeCmd("\n\\\/\\\/OU1234\n"); }
        case /^p$/      { executeCmd("\n\\\/\\\/L1234\n"); }
        case /^r$/      { executeCmd("\n\\\/\\\/U1234\n"); }
        case /^f$/      {
                                $response =
execCmdReturnResponse("\n\\\/\\\/R\n");
                                # $response = "//|R2050123";
                                $response =~ s/(....)(.*)/\2/;
                                print "Firmware Revision:
$response\n";
                                }
        case /^m$/      { getModel(); }
        case /^q$/      { $quit=1; }
        else            { print "For HELP, write '?'\n" }
    }

#----- (  getModel  )-----#
#                                     #
# FUNCTION:      getModel                #
# PURPOSE:      get Model, Serial  and Date of Manufacture    #
#                                     #
#-----#
#
sub getModel {
    $response = execCmdReturnResponse("\n\\\/\\\/? \n");
    # $response = "//|?ACI-2050-C32  5001003  04/04/2001";
    $response =~ s/(....)(.*)/\2/;
    @matrix = split(/\s+/, $response);
    print "Model: $matrix[0]". "\n";
    print "Serial Number: $matrix[1]". "\n";
    print "Date of Manufacture: $matrix[2]". "\n";
}

#----- (  createLinkMatrix  )-----#
#                                     #
# FUNCTION:      createLinkMatrix        #
# PURPOSE:      send matrix of port linked to the user        #
# ARGs:         $matrix - the matrix response by patch panel   #
#               @links - return port linked in patch panel     #
#-----#
#
sub createLinkMatrix {
    local($response) = @_ ;
    $response =~ s/(....)(.*)/\2/;
    @matrix = split(/(..)/, $response);
    $i=0;

    for($index=0;$index<=31;$index++)
    {
        if ($matrix[2*$index+1] != "00")
        {
            $port_index=$index+1;
            if ($port_index < 10)
            { $links[$i]="0"."$port_index"."<". "-"
".">". "$matrix[2*$index+1]"; }
            else
            { $links[$i]="$port_index"."<". "-"
".">". "$matrix[2*$index+1]"; }
            $i++;
        }
    }

    for($index=0;$index<$i;$index++)
    {
        $_ = $links[$index];
        /<->/;
        if ("$_" > "$'") { $links[$index] =~ s/(..)(...)(...)/\3\2\1/; }
    }
}

```

```

    }

    for($index=0;$index<$i;$index++)
    {
        for ($start=$index+1;$start<$i;$start++)
        {
            if ($links[$index] == $links[$start]) { delete
$links[$start];}
        }
        $str=join(' ', @links);
        @links = split(/\s+/, $str);
        return @links;
    }
}

#----- ( executeCmd )-----#
#
# FUNCTION:      executeCmd                      #
# PURPOSE:      send command in arg to telnet session      #
# ARGS:        $cmd - the command to execute on telnet session      #
#
#-----#

sub executeCmd {
    local($cmd) = @_;
    $telnet = new Net::Telnet ( Timeout=>1, Errmode=>'return', Port=>3001);
    $telnet->open('172.16.100.150');
    $telnet->cmd($cmd);
    $telnet->close;
}

#----- ( execCmdReturnResponse )-----#
#
# FUNCTION:      execCmdReturnResponse          #
# PURPOSE:      send command in arg to telnet session      #
#                and get the response          #
# ARGS:        $cmd - the command to execute on telnet session      #
#                $response - the response from the command      #
#
#-----#

sub execCmdReturnResponse {
    local($cmd) = @_;
    $telnet = new Net::Telnet ( Timeout=>1,
                                Errmode=>'return',
                                Port=>3001);

    $telnet->open('172.16.100.150');
    $telnet->buffer_empty;
    $telnet->cmd($cmd);
    $response = $telnet->get;
    $telnet->buffer_empty;
    $telnet->close;
    return $response;
}

#----- ( promptUser )-----#
#
# FUNCTION:      promptUser                      #
# PURPOSE:      Prompt the user for some type of input, and      #
#                return the input back to the calling program.#
# ARGS:        $promptString - what you want to prompt the user#
#                with $defaultValue - (optional) a default value      #
#                for the prompt                      #
#
#-----#

```

```

#-----#
#
sub promptUser {

    #-----#
    # two possible input arguments - $promptString, and $defaultValue #
    # make the input arguments local variables.                        #
    #-----#

    local($promptString,$defaultValue) = @_;

    #-----#
    # if there is a default value, use the first print statement; if   #
    # no default is provided, print the second string.                 #
    #-----#

    if ($defaultValue) {
        print $promptString, "[", $defaultValue, "]: ";
    } else {
        print $promptString, ": ";
    }

    $| = 1;                # force a flush after our print
    $_ = <STDIN>;          # get the input from STDIN (presumably the keyboard)

    #-----#
    # remove the newline character from the end of the input the user  #
    # gave us.                                                           #
    #-----#

    chomp;

    #-----#
    # if we had a $default value, and the user gave us input, then    #
    # return the input; if we had a default, and they gave us no     #
    # no input, return the $defaultValue.                             #
    # if we did not have a default value, then just return whatever   #
    # the user gave us. if they just hit the <enter> key,             #
    # the calling routine will have to deal with that.               #
    #-----#

    if ("$_") {
        return $_ ? $_ : $defaultValue;    # return $_ if it has a value
    } else {
        return $_;
    }
}

```