UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DEPARTAMENTO ACADÊMICO DE INFORMÁTICA

CURSO DE TECNOLOGIA EM DESENVOLVIMENTO DE SISTEMAS DISTRIBUÍDOS

PEDRO HENRIQUE MODESTO DEGUCHI

FRANCISCO BITTENCOURT DOS SANTOS

REESTRUTURAÇÃO DAS CAMADAS 2 E 3 (ENLACE E REDE) DA UTFPR CÂMPUS CURITIBA

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2012

PEDRO HENRIQUE MODESTO DEGUCHI

FRANCISCO BITTENCOURT DOS SANTOS

REESTRUTURAÇÃO DAS CAMADAS 2 E 3 (ENLACE E REDE) DA UTFPR CÄMPUS CURITIBA

Trabalho de Conclusão de Curso apresentado à UTFPR como requisito parcial para obtenção do título de Tecnólogo em Desenvolvimento de Sistemas Distribuídos.
Orientador: Prof. Ms. Luiz Augusto Pelisson
Co-orientador: Prof. Ms. Wilson Horstmeyer Bogado

CURITIBA

2012

AGRADECIMENTOS

A todos os professores do curso de Tecnologia em Desenvolvimento de Sistemas Distribuídos que nos guiaram pelos semestres letivos sempre com total dedicação, em especial nossos orientadores Pelisson e Wilson.

Aos nossos pais, que sempre nos deram apoio, compreensão e acima de tudo um exemplo.

E por fim, a todas as pessoas que nos ajudaram a percorrer essa longa jornada, tais como colegas, amigos e principalmente pessoas companheiras que nos deram luz durante um caminho nem sempre tão claro.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

ACL	Access Control List
ARPANET	Advanced Research Projects Agency Network
ASCII	American Standard Code for Information Interchange
ASICs	Application-specific integrated circuit
BPDU	Bridge Protocol Data Units
CCITT	Comité Consultatif International Telegraphique et Telephonique
DARPA	Defense Advanced Research Projects Agency
DHCP	Dynamic Host Configuration Protocol
EBCDIC	Extended Binary Coded Decimal Interchange Code
EIGRP	Enhanced Interior Gateway Routing Protocol
IAB	Internet Architecture Board
IBM	International Business Machines
ID	Identificação
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPX	Internetwork Packet Exchange
ISL	Inter-*Switch Link*
ISO	International Organization for Standardization
ITU-T	Telecommunication Standardization Sector of the International Telecommunications Union

LACP	*Link* Aggregation Control Protocol
LLC	Logical *Link* Control
MAC	Media Access Control
MTBF	Mean Time Between Failures
MTTR	Mean time to recovery
MVS/ESA	Multiple Virtual System / Enterprise Systems Architecture
NAT	Network Address Translation
NetBIOS	Network Basic Input/Output System
NFS	Network File System
NTP	Network Time Protocol
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PAgP	Port Aggregation Protocol
QoS	Quality of Service
RFC	Request for Comment
RIP	Routing Information Protocol
RJ-45	Registered Jack - 45
RPVST+	Rapid Per VLAN Spanning Tree Plus
SFP	Small Form-factor Pluggable
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SQL	Structured Query Language

SSH	Secure Shell
STP	Spanning Tree Protocol
Syslog	System Log
TCP/IP	Transmission Control Protocol/Internet Protocol
TTL	Time-to-live
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
VTP	VLAN Trunk Protocol
WWW	World Wide Web

RESUMO

Em virtude da complexidade da rede de dados do Cämpus Curitiba da UTFPR, surgiu à necessidade de aquisição de vários equipamentos para a reestruturação desta nas camadas 2 e 3 do modelo OSI – respectivamente enlace e rede - para adequar-se não somente aos novos equipamentos, mas também lhe atribuindo características como escalabilidade, disponibilidade e controle. O principal objetivo desse projeto é apresentar propostas para essa nova rede, assim como as ferramentas necessárias para dar-lhe as características desejadas.

Palavras-chave: Reestruturação. Camada de enlace. Camada de rede.

ABSTRACT

Because of the complexity of the data network UTFPR Curitiba came the need to purchase various equipment for the restructuring of layers 2 and 3 – respectively the data-*link* and network layers of the OSI model, based on the principles of scalability, availability and management. The main goal of this project is to present and suggest a new topology for UTFPR, as well the tools needed to achieve them.

Key words: Restructure. Data-*link* layer. Network layer.

SUMÁRIO

1 INTRODUÇÃO

Desde que a tecnologia de redes surgiu cerca de 30 anos atrás, este é um ramo da informática que continua evoluindo. Dia após dia novos padrões são lançados que buscam torna antigos padrões mais robustos, seguros ou até substituí-los por completo, introduzindo novas formas de executar tarefas conhecidas e resolver problemas que surgiram da crescente demanda por serviços.

Apesar deste contexto dinâmico, a situação que costumamos encontrar na maioria das corporações que iniciaram sua inclusão digital juntamente com a popularização da Internet há 15 anos são ambientes que cresceram sob demanda, buscando sempre se adequar as necessidades dos usuários, nem sempre levando em consideração boas práticas de projeto de rede.

As redes de informação exercem hoje um papel muito importante dentro das organizações, não são apenas ferramentas que permitem conectar a outras pessoas como o telefone foi antes dela, mas muito mais. Ela pode funcionar como repositório de informações e até como provedora de serviços. Tendo isso em mente, percebe-se o quão importante é o projeto de uma rede computacional. Um projeto bem elaborado pode nos fornecer informações como quais equipamentos são mais adequados para o ambiente, quais tecnologias que poderão ser implementadas, quais métodos serão usados para garantir a segurança da informação da organização e ainda lidar com questões como gerenciamento de recursos, escalabilidade e períodos de inatividade.

O escopo deste projeto trata da reestruturação e melhorias para a atual infraestrutura de redes, levando em consideração as camadas 2 e 3 do modelo OSI, da sede do Cämpus Curitiba da UTFPR.

1.1 Justificativa

A rede atual da UTFPR Cämpus Curitiba foi estruturada de uma maneira funcional, adicionando-se equipamentos conforme a necessidade da mesma. Com o tempo essa solução apresenta alguns problemas, como a utilização de poucas VLANs estendidas por toda a rede, criando um grande domínio de *broadcast*; descentralização da rede, complicando o gerenciamento; criação de gargalos de rede através do cascateamento sob demanda dos *switches*; baixa disponibilidade devida à falta de redundância e baixa escalabilidade.

Com a compra de 49 *switches* de camada 2 e 2 de camada 3 com o intuito de reestruturar a rede de computadores da UTFPR Cämpus Curitiba, notou-se que esses equipamentos não poderiam simplesmente substituir os equipamentos empregados atualmente, mas também reestruturar a topologia lógica da rede. Devido às capacidades dos novos equipamentos, uma simples troca seria levá-los a uma subutilização extrema, o que não justificaria o custo de aquisição dos mesmos. Essa reestruturação leva em conta a otimização e valorização dos conceitos de alta disponibilidade (através de redundância), alto desempenho (através da utilização de *link*s mais rápidos e agregados, utilização de uma topologia otimizada), e escalável (através de uma topologia bem planejada).

1.2 Objetivo geral do trabalho

O objetivo do trabalho é propor uma nova estrutura lógica para a rede da UTFPR para que possa ser implantada nos novos equipamentos que foram adquiridos, permitindo assim que esta se torne mais robusta e escalável.

1.3 Objetivos específicos do trabalho

- Analisar a infraestrutura física e lógica da UTFPR Câmpus Curitiba.
- Verificar quais os equipamentos adequados para a implementação.
- Levantar os problemas decorrentes da topologia atual.
- Localizar possíveis pontos de falha e seus impactos.
- Elaborar uma nova maneira de estruturar o Câmpus utilizando os novos equipamentos.
- Sugerir novas funcionalidades e tecnologias que possam melhorar a administração e organização da rede.
- Explicar os benefícios da nova estrutura e das tecnologias envolvidas.

1.4 Conteúdo

Este trabalho apresentará uma nova topologia para as camadas 2 e 3 para a UTFPR Cämpus Curitiba, assim como os protocolos e funcionalidades selecionados juntamente à Coordenadoria de Gestão de Tecnologia da Informação (COGETI). As configurações e exemplos mostrados terão como base as recomendações da Cisco Systems, tendo em vista que todos os equipamentos adquiridos são fabricados pela mesma.

Porém, devido a uma necessidade de se utilizar outros fabricantes num futuro e à utilização de padrões que podem ser estudados e compartilhados por todos, sempre

que possível será abordado o paradigma proposto por um protocolo aberto ao invés de um proprietário.

Alguns protocolos amplamente conhecidos não serão abordados neste trabalho. O VTP não será utilizado, pois apresenta um risco muito grande para a rede. Protocolos de segurança proibitivos, como o Port Security ou 802.1x não serão implementados, pois inviabilizaria a administração da rede, uma vez que eles bloqueariam o acesso de vários usuários.

Não serão abordados assuntos pertinentes à camada física, bem como cabeamento estruturado, conversões de mídia e demais tópicos referentes à camada 1. Também não serão abordados assuntos da camada de aplicação, como DHCP (apenas a questão de *relay* das solicitações e detecção e bloqueio de clandestinos), bloqueio de protocolos como *torrent* e outros.

2 LEVANTAMENTO BIBLIOGRÁFICO E ESTADO DA ARTE

A seguir será discutida a criação das redes de computadores, seus modelos lógico-didáticos e as tecnologias empregadas neste trabalho.

2.1 Surgimento de tecnologias IP e da Internet

Por volta das décadas de 1960 e 1970, diversas redes estavam operando usando protocolos e implementações próprias. Compartilhar informações entre tais redes logo se tornou um problema. Surgiu dai a necessidade do desenvolvimento de um protocolo comum, que permitisse a comunicação entre máquinas de diferentes fabricantes, falando uma mesma “linguagem”. A *Defense Advanced Research Projects Agency* (DARPA) financiou então a pesquisa deste protocolo comum e do conjunto de protocolos da ARPANET, iniciativa que se destacou por introduzir o conceito de camadas em seu desenho. A pilha de protocolos TCP/IP evoluiu a partir dos protocolos da ARPANET e tomou forma em 1978. Com a adoção do TCP/IP, uma rede foi criada para o uso de agências governamentais e institutos de pesquisa com o proposito de compartilhar informações e colaboração de pesquisas.[30]

No *início* da década de 1980 TCP/IP se tornou o principal protocolo em redes que faziam uso de equipamentos de vários fabricantes, como no caso da ARPANET. A pilha de protocolos TCP/IP foi então integrada ao sistema Unix da Universidade da Califórnia em Berkeley e isso a tornou acessível para o publico. Deste ponto em diante TCP/IP se tornou amplamente utilizado devido a sua disponibilidade em ambientes Unix e espalhou-se para outros sistemas operacionais.

Hoje, TCP/IP fornece a habilidade para corporações integrarem redes físicas diferentes e ainda proporciona aos usuários um conjunto comum de funções. Permite a

interoperabilidade entre equipamentos fornecidos por diferentes fornecedores e diferentes plataformas e prove acesso a Internet.

A Internet hoje consiste em um conjunto de redes *backbones* que permitem que Cämpus, redes locais e acessos individuais acessem recursos globais. O uso da Internet cresceu exponencialmente nos últimos anos, principalmente apos a adoção pelo mercado consumidor.

Os motivos para o uso do TCP/IP ter crescido em um ritmo tão acelerado incluem a disponibilidade de funções comuns através de diferentes plataformas e a habilidade de acessar a Internet, mas o motivo principal é o de propiciar interoperabilidade. Os padrões abertos do TCP/IP permitem a corporações interconectar ou combinar plataformas diferentes. Um exemplo é o simples ato de permitir a transferência de um arquivo entre um usuário IBM MVS/ESA e estações Apple Macintosh. O TCP/IP também prove transporte para outros protocolos como IPX ou NetBIOS. Esses protocolos podem fazer uso de uma rede TCP/IP para se conectar em outras redes com protocolos similares.

Outra razão pelo crescimento do TCP/IP é a popularidade da interface de programação através de sockets, que é a interface entre a camada de transporte do protocolo TCP/IP e aplicativos. Um grande número de aplicativos hoje foi escrito para a interface de sockets TCP/IP. O processo RFC (Request for Comments), supervisionado pela Internet Architecture Board (IAB) e pela Internet Engineering Task Force (IETF), gerencia o trabalho constante de atualização e extensão da pilha de protocolos.

2.2 O modelo OSI

Durante a época que a DARPA pesquisava por um conjunto de protocolos que permitisse interligar redes heterogêneas, que eventualmente levou ao TCP/IP e a Internet, outra abordagem alternativa para o mesmo problema estava sendo conduzida

pelo CCITT (Comité Consultatif International Telegraphique et Telephonique, ou Consultative Committee on International Telegraph and Telephone) e ISO (International Organization for Standardization). O CCITT desde então tornou se o ITU-T (International Telecommunication Union - Telecommunication).[31]

O padrão que resultou foi o Modelo de *Referência* OSI (Open System Interconnection ou ISO 7498), que definia um modelo de 7 camadas para comunicação de dados, como pode ser visto na figura 1.

Figura 1 - Modelo OSI

(VSTRABELLO, 2010)

Cada camada do modelo de *referência* OSI fornece um conjunto de funções para a camada superior, e da mesma forma, depende de um conjunto de funções fornecidas pela camada imediatamente abaixo. Embora mensagens passem apenas verticalmente pela pilha de uma camada para a próxima, de um ponto de vista lógico, cada camada se comunica diretamente com sua equivalente em outros nós, ou seja, a camada 4 do usuário A se comunica com a camada 4 do usuário B e assim por diante.

A abordagem em camadas foi escolhida como base para promover a flexibilidade através do uso de interfaces definidas. As interfaces permitem que uma camada seja alterada sem que as outras sejam afetadas. A principio, desde que as

interfaces adjacentes das camadas ainda estejam funcionando, a implementação deve funcionar.

2.2.1 Camada de Aplicação

A camada de aplicação do modelo OSI é onde o usuário se relaciona/comunica com o computador. A camada de aplicação é responsável por identificar e estabelecer a disponibilidade de comunicação e determinar se recursos suficientes para a comunicação existem.[31]

Embora aplicativos geralmente precisem apenas de recursos locais, alguns podem fazer uso de componentes de comunicações de mais de um aplicativo de rede; por exemplo, transferências de arquivo, *e-mail*, acesso remoto, gerenciamento de rede, processos cliente/servidor e localização de informações. Muitos aplicativos permitem que a comunicação através de redes corporativas ocorra, mas diariamente seus limites são testados e a demanda por maior desempenho exigido. Hoje, transações e trocas de informação entre organizações continuam se expandindo e requerem interligar entre aplicativos como:

- *World Wide Web*(WWW): conecta incontáveis servidores apresentando diversos formatos. A maioria é multimídia e incluem alguns ou todos os seguintes: gráficos, texto, vídeo e ate som. Firefox, Internet Explorer, Google Chrome e outros navegadores simplificam o acesso e visualização de *sites Web*.
- *E-mail gateways*: são versáteis e podem utilizar *Simple Mail Transfer Protocol* (SMTP) ou outros padrões para entregar mensagens entre diferentes aplicativos de *e-mail*.
- *Bulletin Boards*: incluem muitas salas de bate-papo onde pessoas podem se conectar e comunicarem-se umas com as outras postando mensagens ou

digitando conversas interativamente. Embora tenha sido muito popular durante a época de acesso discado, hoje tem caído em desuso.

- Ferramentas de navegação da Internet: incluem aplicativos como Gopher e WAIS, assim como ferramentas de busca como YAHOO! e Google, que ajudam usuários a localizarem recursos e informações na Internet.
- Serviços de transações financeiras: voltados à comunidade financeira. Eles acumulam e vendem informações a respeito de investimentos, mercado de trocas, commodities, taxas de troca de moedas e informação de credito para seus assinantes.

2.2.2 Camada de Apresentação

A camada de apresentação possui este nome devido a seu proposito: ela apresenta os dados para a camada de aplicação. Ela é essencialmente um tradutor e prove codificação e funções de conversão. Uma técnica de transferência de dados bem sucedida é adaptar a informação transmitida em um formato padrão antes de transmiti-la. Computadores são configurados para receber dados neste formato genérico e então convertê-lo de volta para seu formato nativo para ser utilizado (por exemplo, EBCDIC para ASCII). Ao prover serviços de tradução, a camada de apresentação assegura que a transmissão de dados da camada de aplicação de um sistema pode ser lida pela camada de aplicação de outro *host*.

O Modelo OSI possui protocolos que definem como dados padrões devem ser formatados. Tarefas como compressão, descompressão, encriptação e decriptação estão associadas com esta camada. Alguns padrões da camada de apresentação estão envolvidos em operações multimídia, apresentações de imagens e gráficos.

2.2.3 Camada de Sessão

A camada de sessão é responsável por estabelecer, gerenciar e encerrar sessões entre entidades da camada de apresentação. A camada de sessão também fornece controle de conversação entre dispositivos e nós. Ela coordena comunicação entre sistemas e serve para organizar sua comunicação oferecendo 3 modos diferentes: simplex, half-duplex e full-duplex. A camada de sessão basicamente mantem dados de aplicativos diferentes separados de aplicativos de outros dados. A seguir estão alguns exemplos de protocolos da camada de sessão:

- *Network File System* (NFS): foi desenvolvido pela Sun Microsystems e usado com TCP/IP e estações Unix para permitir acesso remoto transparente a recursos remotos.
- *Structure Query Language* (SQL): foi desenvolvido pela IBM para fornecer a usuários um método mais simples para definir seus requisitos de informação tanto em sistemas locais como remotamente.
- *Remote Procedure Call* (RPC): é uma ampla ferramenta de redirecionamento cliente/servidor usada para uma variedade de ambientes de serviços. Suas rotinas são criadas nos clientes e executadas nos servidores.
- X *Window*: é amplamente usado por terminais inteligentes para comunicar com computadores Unix remotamente, permitindo que operem como se estivessem localmente no monitor.

2.2.4 Camada de Transporte

Serviços localizados na camada de transporte realizam o trabalho de segmentar e reagrupar dados de aplicativos da camada superior e atrelar em um mesmo *stream*

de dados. Ela fornece serviços de transporte de dados fim-a-fim e pode estabelecer uma conexão logica entre o *host* de origem e o de destino em uma rede.

Ao trabalhar com protocolos TCP/IP, desenvolvedores tem a escolha de trabalhar com 2 protocolos que possuem a mesma função, embora ofereçam benefícios diferentes. São eles TCP e UDP, sendo que o primeiro possui a vantagem de fornecer um serviço confiável, enquanto o segundo não tem esta preocupação, deixando tal controle para as camadas superiores.

A camada de transporte é responsável por fornecer mecanismos para multiplexação de aplicativos das camadas superiores, estabelecimento de sessões e encerramento de circuitos virtuais. Ela também esconde detalhes de qualquer informação dependente da rede das camadas superiores ao fornecer transferência de dados transparente.

2.2.5 Camada de Rede

A camada de rede é responsável pelo roteamento através das redes e pelo endereçamento de rede. Isso significa que a camada de Rede é responsável pelo transporte de trafego através de dispositivos que não estão conectados de forma local. *Routers* ou outros dispositivos de camada 3 são especificados na camada de rede e provêem serviços de roteamento e interligação de redes.[31]

Quando um *router* recebe um pacote em uma de suas interfaces, o endereço IP de destino é checado. Se o pacote é destinado ao *router* então este ira olhar a rede de destino em sua Quadro de roteamento. Uma vez que uma interface de saída for escolhida, o pacote será então enviado para essa interface para ser enquadrado (*framed*) e em seguida enviado para a rede local. Caso a entrada para a rede de destino não se encontre em sua Quadro de roteamento, o *router* ira descartar o pacote.

Dois tipos de pacotes são utilizados na camada de rede: pacotes de dados e atualizações de rota.

- Pacotes de dados: são usados para transferir informação através da Internet, e protocolos usados para suportar o trafego de dados também chamados de protocolos roteados;
- Pacotes de Atualização de Rotas: são usados para atualizar *routers* vizinhos a respeito de redes conectadas ao *router*. Protocolos que enviam estes pacotes são chamados de protocolos de roteamento e alguns exemplos desses podem ser RIP, EIGRP e OSPF. Pacotes de atualizações de rotas são pacotes utilizados para ajudar a construir e manter Quadros de roteamento em cada *router*;

O quadro de roteamento em um *router* contem as seguintes informações:

- Endereço de Rede: endereços de rede específicos para cada protocolo. Um *router* deve manter uma Quadro de roteamento individual para cada protocolo, pois cada protocolo de roteamento mantem registros de rede com um esquema de endereçamento particular;
- Interface: a interface de saída que um pacote irá tomar quando for destinado a uma interface específica.
- Métrica: A distância até a rede remota. Diferentes protocolos de roteamento fazem uso de métodos diferentes para calcular a distância. Protocolos de roteamento podem utilizar, por exemplo, como métrica número de nós até o destino (*hop count*), outros utilizam largura de banda disponível, atraso na linha e ainda há aqueles que usam ate mais de um valor para definir sua métrica.

Routers e *switches* multicamada quebram os chamados domínios de *broadcast* – ou difusão. Isso significa, por padrão, que *broadcasts* não são repassados por *routers*. Eles também quebram domínios de colisão, mas isso também pode ser alcançado através do uso de um *switch* de camada 2. Cada interface de um *router* está

em uma rede diferente e por isso deve receber um endereço de rede único. Cada *host* na rede conectada àquele *router* deve usar o mesmo endereço de rede.

Alguns pontos a respeito de *routers* que devem ser lembrados:

- *Routers* por padrão não irão encaminhar pacotes de *broadcast* ou *multicast*;
- *Routers* utilizam o endereço lógico localizado no cabeçalho da camada de rede para determinar o próximo *router* para onde encaminhar o pacote;
- *Routers* podem usar listas de acesso, criadas pelo administrador para controlar a segurança de pacotes tentando entrar ou sair de uma interface;
- *Router* podem fornecer funções de comutação da camada 2 e, se necessário, podem simultâneamente rotear através da mesma interface;
- Dispositivos de camada 3 (*routers* no caso) fornecem conectividade entre Virtual LANs (VLANs);
- *Routers* podem fornecer *Quality of Service* (QoS) para tipos específicos de tráfego de rede.

2.2.6 Camada de Enlace

A camada de enlace garante que a mensagem seja entregue ao dispositivo correto e traduz mensagens da camada de Rede em *bits* para a camada física transmitir. Ela formata as mensagens em quadros e adiciona um cabeçalho personalizado contendo o endereço de *hardware* de destino e de origem. Esta informação adicionada envolve a mensagem original como uma capsula.

É preciso entender que *routers*, que trabalham na camada de rede, não se importam com o local onde um *host* está localizado, mas apenas onde as redes estão localizadas. Eles também mantêm registros dos melhores modos de chegar a uma rede remota. A camada de acesso ao meio é responsável por identificar unicamente cada dispositivo em uma rede local.

Para um *host* enviar pacotes para outros *hosts* individualmente e através de *routers*, a camada de enlace usa o endereço de *hardware*. Cada vez que um pacote é enviado através de *routers* ele é enquadrado com informação de controle na camada de enlace, mas essa informação é destacada no *routers* recebendo a mensagem e apenas o pacote original é mantido intacto. Esse enquadramento de pacotes continua para cada salto do pacote, até que este alcance seu destino, sendo finalmente entregue ao *host* de destino. É importante entender que o pacote nunca foi alterado ao longo de sua rota, apenas foi encapsulado com o tipo de informação de controle para ser passado para os diferentes tipos de meio. A camada de acesso ao meio *Ethernet* do IEEE possui duas subcamadas:

- *Media Access Control* (MAC) 802.3: esta define como pacotes são inseridos no meio. A ordem de acesso ao meio é "*first come, first served*" onde todos compartilham o mesmo meio e a mesma largura de banda. Endereçamento físico também é definido aqui, assim como a topologia lógica. Topologia lógica é o caminho através de uma topologia física. Notificação de erros (não

correção) entrega ordenada de quadros e opcionalmente controle de fluxo são funcionalidades providas por esta subcamada;

- *Logical Link Control* (LLC) 802.2: esta subcamada é responsável por identificar os protocolos da camada de rede e encapsula-los. Um *header* LLC informa a camada de acesso ao meio o que fazer com o pacote, uma vez que um quadro é recebido. Por exemplo, um *host* vai receber um quadro e então olhar no cabeçalho LLC para entender que o pacote é destinado ao protocolo IP na camada de rede. O LLC também pode prover controle de fluxo e *bits* de controle sequenciais;

Switches e *bridges* trabalham ambas na camada de enlace e filtram a rede utilizando endereços de *hardware* (MAC). Comutação na camada 2 é considerada baseada em *hardware*, pois faz uso de um *hardware* especializado chamando *Application-Specific Integrated Circuits* (ASICs). ASICs podem chegar a rodar em velocidades da ordem de *gigabits* com uma latência muito baixa.

Bridges e *switches* lêem cada *frame* que passa através de rede. O dispositivo de camada 2 coloca então o endereço de *hardware* de destino em uma Quadro e também a porta de onde ele foi recebido. Isso ira informar o *switch* onde o dispositivo esta localizado.

Após terminar de construir a Quadro, o dispositivo de camada 2 irá apenas encaminhar quadros para o segmento onde o endereço de destino está localizado. Caso o dispositivo de destino esteja no mesmo segmento que o *frame*, o dispositivo de camada 2 irá bloquear o quadro de ir para qualquer outro segmento. Caso o destino esteja localizado em outro segmento então o quadro somente será enviado para este segmento. Isto é chamado *bridging* transparente.

Quando a interface de um dispositivo de camada 2 recebe um quadro e o endereço de *hardware* de destino é desconhecido da Quadro do dispositivo, ele irá encaminhar o quadro para todos os segmentos conectados. Caso o dispositivo desconhecido responda a este encaminhamento do quadro, o *switch* irá então atualizar sua Quadro com a localização daquele novo dispositivo. No entanto, o endereço de

destino do quadro transmitido pode ser um endereço de *broadcast*, nesse caso o *switch* irá, por padrão, encaminhar para todos os segmentos conectados.

Todos os dispositivos em que o *broadcast* é encaminhado são considerados pertencentes ao mesmo domínio de *broadcast*. Dispositivos de camada 2 propagam tempestades de *broadcasts* de camada 2. Um dos modos de impedir uma tempestade de *broadcast* de se propagar através da rede é com o uso de um dispositivo de camada 3.

O maior benefício de utilizar *switches* ao invés de *hubs* em sua rede é o fato de cada porta do *switch* representar um único domínio de colisão, ao passo que o *hub* cria um grande domínio de colisão. No entanto, *switches* e *bridges* não quebram domínios de *broadcast*, ao invés disso encaminham todos os *broadcasts*.

Outro beneficio de LAN *switches* sobre implementações com *hubs* é que cada dispositivo em cada segmento plugado ao *switch* pode transmitir simultaneamente por que cada segmento é um domino de colisão. *Hubs* permitem que apenas um dispositivo se comunique por vez.

2.2.7 Camada Física

A camada física possui duas responsabilidades: envia *bits* e recebe *bits*. *Bits* possuem apenas 2 valores 1 ou 0 - um Código Morse com valores numéricos. A camada física comunica-se diretamente com os vários tipos de meios de comunicação. Tipos de mídia diferentes representam esses valores de *bits* de forma diferente. Alguns usam tons de áudio, enquanto outros empregam mudanças de estado - variações na voltagem de alta para baixa e de baixa para alta. Protocolos específicos são necessários para cada tipo de mídia para descrever o padrão correto para designar um *bit*, como os dados são codificados em sinais para o meio e as diversas características da interface física do meio.

As propriedades da camada física especificam requisitos elétricos, mecânicos, procedurais e funcionais para ativar, manter e desativar um *link* físico entre 2 sistemas.

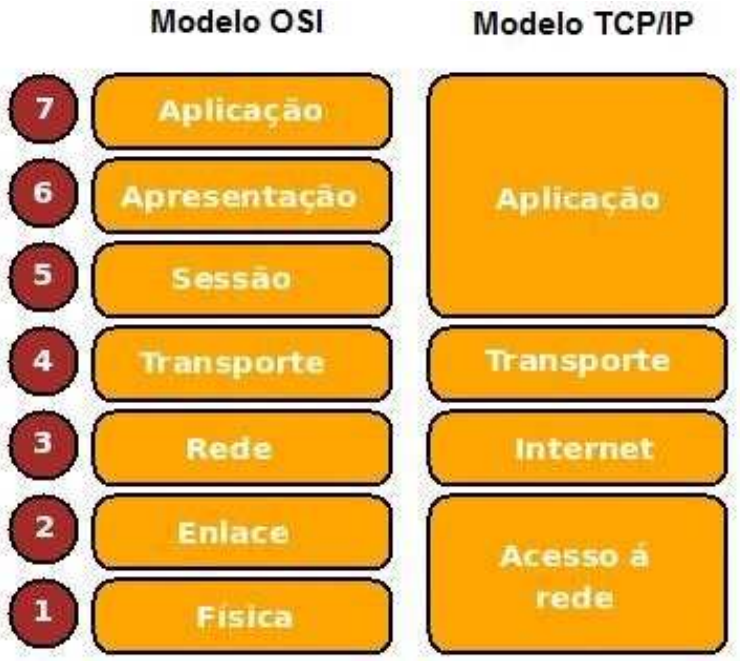
Os diferentes conectores e topologias físicos da camada física são definidos pelos padrões OSI, permitindo que sistemas heterogêneos possam se comunicar.

2.3 O Modelo TCP/IP

Ao contrário dos protocolos OSI que se desenvolveram de forma lenta, devido principalmente à sua abordagem de planejamento formal, através de comitês, os protocolos TCP/IP evoluíram e amadureceram rapidamente. Com sua política pública de RFCs para a melhoria e atualização da pilha de protocolos, ele conseguiu se estabelecer como o protocolo de escolha para redes de dados.

Como no modelo OSI, e na maioria dos demais protocolos de comunicação, TCP/IP consiste em uma pilha de protocolos, composta de 4 camadas. A figura 2 mostra essas camadas em comparação ao modelo OSI:

Figura 2 - Modelo OSI e TCP/IP



(VSTRABELLO, 2010)

Como podemos observar na figura 2, as camadas 5, 6 e 7 do modelo OSI foram condensadas em somente uma camada no modelo TCP/IP, chamada de Aplicação. As camadas 1 e 2 do modelo OSI também foram condensadas e chamadas de Acesso à rede.

2.4 A necessidade para o desenho de uma Rede IP

A facilidade para interconectar através do uso do TCP/IP pode acarretar problemas. Por exemplo, a falta de um planejamento cuidadoso no uso de endereços de rede pode gerar serias limitações no número de *hosts* podem se conectar em uma rede. A falta de uma coordenação centralizada pode levar ao uso de nomes ou endereços de recursos duplicados, o que pode impedi-lo conectar redes isoladas. O

uso inadvertido de endereços pode impedi-lo de conectar-se à Internet e outros problemas que podem surgir é a impossibilidade de traduzir nome de recursos para seus endereços, pois não foi possível estabelecer uma conexão com o servidor de nomes. É muito comum para uma rede ser conectada desta maneira, e isso pode até funcionar bem para pequenas redes. Porém, problemas ocorrem quando é necessário realizar alguma mudança na rede e nenhuma documentação é encontrada. [30]

Alguns problemas que surgem de uma rede sem planejamento ou mal planejada podem ser corrigidos facilmente. Outros, no entanto, requerem uma quantidade elevada de tempo e esforço para serem corrigidos. Por exemplo, configurar manualmente cada *host* em uma rede com aproximadamente 3 mil estações devido a um esquema de endereçamento que não se encaixa mais nas necessidades atuais.

Quando confrontado com a tarefa de desenhar uma nova rede ou interligar redes existentes, existe uma série de questões de desenho que deverão ser resolvidas. Por exemplo, como alocar endereços para recursos de rede, como alterar endereços existentes, se será usado roteamento dinâmico ou estático, como configurar servidores de nome e como proteger sua rede são todas questões que devem ser respondidas. Ao mesmo tempo questões como confiabilidade, disponibilidade e *backup* deverão ser considerados junto também como a forma como a rede será gerenciada e administrada.

Uma rede IP que não foi desenhada de uma forma sistemática invariavelmente irá se deparar com problemas desde o começo do estágio de implementação. Quando se esta atualizando uma rede existente, geralmente existem redes legadas que precisam ser conectadas. A introdução de uma nova tecnologia sem o estudo das limitações da rede atual pode levar a problemas imprevistos. O administrador pode acabar tentando resolver um problema que foi criado desnecessariamente.

O desenho da rede deve ser realizado antes que qualquer implementação venha a ser feito. O desenho de uma rede IP também deve ser constantemente revisado, pois os requisitos mudam com o tempo. Um bom desenho de uma rede IP também inclui

uma documentação detalhada da rede para *referência* futura. Uma rede IP bem desenhada deve ser fácil de implementar, com poucas surpresas.

Algumas considerações de desenho que representam pontos essenciais são:

- Escalabilidade: Uma rede bem desenhada deve ser escalável, para que possa crescer conforme o aumento de requerimentos. Introdução de novos *hosts*, servidores ou redes não devem exigir que a topologia inteira seja redesenhada. A topologia escolhida deve ser capaz de acomodar a expansão resultante dos requerimentos do negócio;
- Disponibilidade/Confiabilidade: As exigências de negócios demandam um grande nível de disponibilidade e confiabilidade da rede;
- Modularidade: Um conceito importante a ser adotado é a abordagem de desenho modular ao construir uma rede. Modularidade permite dividir um sistema complexo em outros menores e mais gerenciáveis, tornando a implementação muito mais fácil. Modularidade também garante que a falha em determinada parte da rede possa ser isolada para que esta não traga a rede inteira abaixo. A modularidade tem benefícios monetários também, pois a adição de um novo segmento de rede ou aplicativo a rede não implicará em mudanças em todos os demais;
- Segurança: A segurança da rede de uma organização é um importante aspecto do desenho, especialmente quando a rede irá interagir com a Internet. Considerar os aspectos de segurança em estágios posteriores deixa a rede aberta para ataques até que todos os riscos de segurança sejam lidados, uma abordagem reativa ao invés da proativa pode custar muito caro. Embora novas falhas de segurança possam surgir, eles podem ser adicionados ao desenho;
- Desempenho: Existem duas medidas a serem consideradas ao planejar uma rede. A primeira delas é o *throughput* e a segunda delas é o tempo de resposta. *Throughput* corresponde à quantidade de dados que pode ser enviado no menor tempo possível, enquanto o tempo de resposta é o quanto um usuário deve esperar antes que o sistema lhe retorne um resultado.

- Economia: Uma rede que adereça todas as necessidades da organização, mas custa duas vezes o que deveria, não é uma rede bem planejada.

2.5 Modelo Hierárquico Cisco

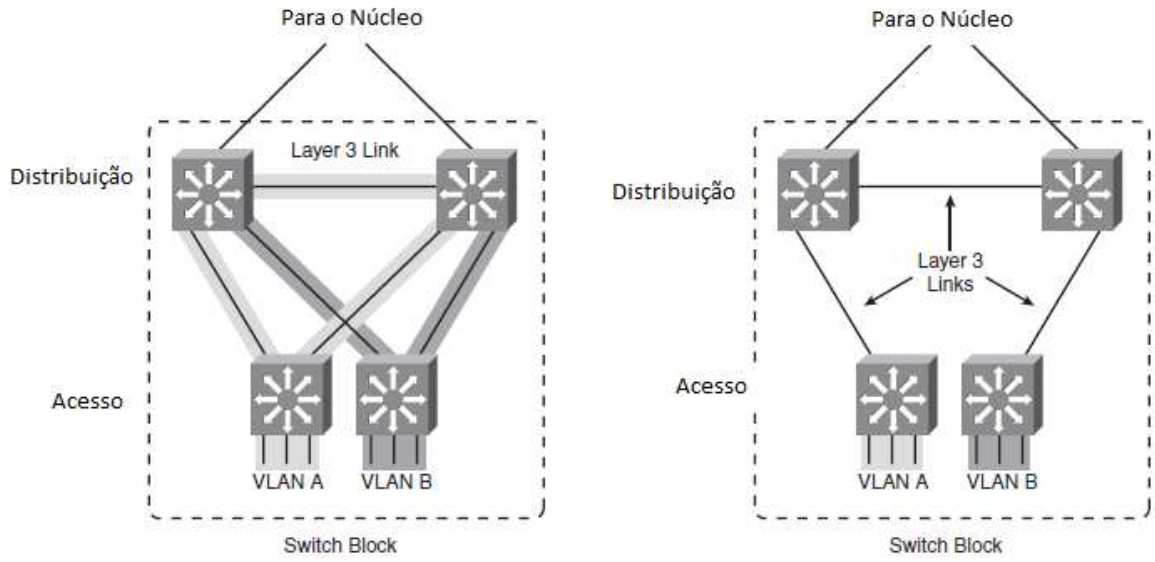
Para suprir as necessidades de se moldar uma rede que demande baixa manutenção, alta disponibilidade, escalável e eficiente a Cisco System desenvolveu um modelo hierárquico em camadas distintas de dispositivos. A topologia resultante é eficiente, inteligente, escalável e de fácil gerenciamento. As camadas são:

- Acesso - Presente onde os usuários finais se conectam na rede. *Switches* de acesso normalmente provêem conectividade à camada 2 (VLAN) entre os usuários. Dispositivos nessa camada devem ter as seguintes capacidades:
  - Baixo custo por porta.
  - Alta densidade de portas.
  - Troncos escaláveis para as camadas superiores.
  - Funções de acesso ao usuário, como entrada às VLANs, filtros de tráfego e protocolo e Qualidade de Serviço (QoS).
- Distribuição – Provê interconectividade entre os dispositivos de acesso e núcleo. Dispositivos nessa camada devem ter as seguintes capacidades:
  - Agregação de múltiplos dispositivos de acesso.
  - Alta vazão (*throughput*) para manejo de pacotes.
  - Funções de segurança e políticas de conectividade através de listas de acesso ou filtros de pacotes.
  - QoS
  - *Link*s de alta velocidade escaláveis e resilientes para as camadas de acesso e núcleo.
  - Comutação multicamada, criando um limite de camada 3.
- Núcleo – Provê conectividade para todos os dispositivos de distribuição. O Núcleo, também chamado de *Backbone*, deve ser capaz de comutar o tráfego da maneira mais eficiente possível. Dispositivos do *backbone* devem ter as seguintes capacidades:
  - Alta vazão de dados na camada 3.

- Nenhuma manipulação desnecessária ou de alto custo de pacotes (listas de acesso e filtros).
- Redundância e resiliência para alta disponibilidade. Funções de QoS avançadas.
- Otimização para alto-desempenho e manejo de grande quantidade de dados.

Apesar de serem apresentadas 3 camadas – Acesso, Distribuição e Núcleo – esse modelo pode ser colapsado ou simplificado em certos casos. Por exemplo: Redes de Cämpus pequeno ou médio podem ter as camadas de Distribuição e Núcleo colapsadas para melhor custo-benefício. A figura 3 ilustra o modelo em 3 camadas e o modelo colapsado.[1]

Figura 3 - Modelo Hierárquico



(Hucaby, 2010)

2.6 VLAN

Em uma topologia de rede simples onde existam apenas *switches* ethernet nível 2 possuímos apenas um domínio de *broadcast*. Isso significa que todos os dispositivos

conectados aos *switches* receberão os pacotes de *broadcast*. Isso em uma rede com poucos dispositivos não é problema, mas quando é aumentada a quantidade de dispositivos conectados, passa a ser um. Para melhorar a segmentação da rede e aumentar o nível de segurança são utilizadas as VLANs ou LANs Virtuais para criar mais domínios de *broadcast*, separando a comunicação por função, área, setor ou quaisquer outras características necessárias. [34]

Uma rede de camada 2 comutada é considerada plana. Cada pacote de *broadcast* transmitido pode ser visto por todos os dispositivos na rede, independentemente se o dispositivo precisa receber a informação.

Pelo fato de *switches* de camada 2 criarem segmentos individuais de domínios de colisão para cada dispositivo conectado nele, as limitações de distância impostas pelo padrão Ethernet podem ser removidas, o que significa que redes maiores podem ser construídas. Quanto maior o número de usuários e dispositivos, maior será o número de pacotes e *broadcasts* que cada dispositivo deverá lidar. Outro problema com redes camada 2 planas é a segurança, por todos os usuários poderem ver todos os dispositivos. Não é possível impedir que dispositivos utilizem *broadcasts* e nem impedir usuários tentando responder a *broadcasts*. *Broadcasts* são utilizados em todos os protocolos, mas a frequência em que ocorrem depende do protocolo, das aplicações rodando na rede e como esses serviços são usados.

Alguns aplicativos mais antigos foram reescritos para diminuir sua necessidade de largura de banda. No entanto, existe uma nova geração de aplicativos que são gananciosos no uso de largura de banda, consumindo toda que possam encontrar. Estes são aplicativos multimídia que usam *broadcasts* e *multicasts* intensivamente. Equipamentos problemáticos, segmentação inadequada e *firewalls* mal desenhados também podem ser adicionados aos problemas de aplicativos que utilizam intensivamente *broadcasts*. Isso adicionou um novo capítulo ao desenho de rede, pois *broadcasts* podem se propagar através de uma rede comutada. *Routers*, por padrão, enviam *broadcasts* apenas para a rede de onde este se originou, mas *switches* encaminham *broadcasts* para todos os segmentos. Isto é chamado de "rede plana" por que possui apenas um domínio de *broadcast*.

Como um administrador, é preciso ter certeza que a rede está segmentada de maneira eficiente para evitar que os problemas de um segmento se propaguem para a o restante da rede. A maneira mais eficaz de fazer isso é através de roteamento e *switching*. Nos últimos tempos o custo-benefício de *switches* se tornou o mais atraente, por isso a maioria das empresas tem substituído redes planas por redes comutadas e VLANs. Todos os dispositivos em uma VLAN são membros do mesmo domínio de *broadcast* e recebem todos os *broadcasts*. Os *broadcasts,* por padrão, são filtrados em todas as portas que não são membros da mesma VLAN. *Routers* ou *switches* camada 3 devem ser usados juntamente com *switches* para promover a conexão entre redes (VLANs), o que pode impedir *broadcasts* de propagar através de uma rede inteira.

Um problema com redes planas é que a segurança é implementada através da conexão de *hubs* e *switches* junto com *routers*. A segurança é mantida no *router*, mas qualquer um conectado à rede física pode ter acesso a recursos da rede local. Outra falha é que um usuário poderia plugar um dispositivo de captura de pacotes em um *hub* e ter acesso a todo o trafego da rede.

Através do uso de VLANs, e criando múltiplos grupos de *broadcasts*, administradores podem ter o controle sobre cada usuário e cada porta. Usuários não podem mais ter acesso aos recursos da rede simplesmente plugando em uma porta do *switch*. O administrador controla cada porta e qual recurso é permitido de usar. Como grupos podem ser criados de acordo com recursos de rede que o usuário necessita, *switches* podem ser configurados para informar uma estação de gerenciamento de rede qualquer acesso não autorizado a recursos da rede. Caso deva existir comunicação entre as VLANs, restrições podem ser adicionadas. Restrições também podem ser inseridas no endereço de *hardware*, protocolos e aplicativos.

Switches camada 2 apenas lêem quadros para filtrá-los; eles não olham o protocolo da camada de Rede. Isso pode fazer com que um *switch* encaminhe todos os *broadcasts*. No entanto, ao criar VLANs, estarão sendo criados domínios de *broadcast*. *Broadcasts* enviados de um nó em uma VLAN não serão encaminhados para portas em uma VLAN diferente. Ao atribuir portas ou usuários para grupos VLAN em um *switch* ou grupo de *switches* conectados, é possível obter a flexibilidade para adicionar

o usuário que quiser ao domínio de *broadcast* independente de sua localização física. Isso pode impedir tempestades de *broadcast* causadas por interfaces de rede defeituosas ou aplicativos de se propagarem através da rede inteira.

Quando uma VLAN fica muito grande, é possível criar mais VLANs para manter *broadcasts* de consumir toda a largura de banda. Quanto menos usuários houver em uma VLANs, menor é o número de usuários afetado por *broadcasts*.

Para entender como uma VLAN é vista por um *switch*, é possível usar um *router* como analogia. Cada rede conectada a um *router* possui seu próprio endereço de rede. Cada nó conectado a uma rede física deve ter um endereço de rede que combine com o número da rede para serem capazes de se conectar com as demais. *Switches* funcionam de forma semelhante. *Switches* criam maior flexibilidade e escalabilidade do que *routers* poderiam alcançar sozinhos. É possível agrupar usuários em comunidades de interesse, que são conhecidas como VLANs organizacionais.

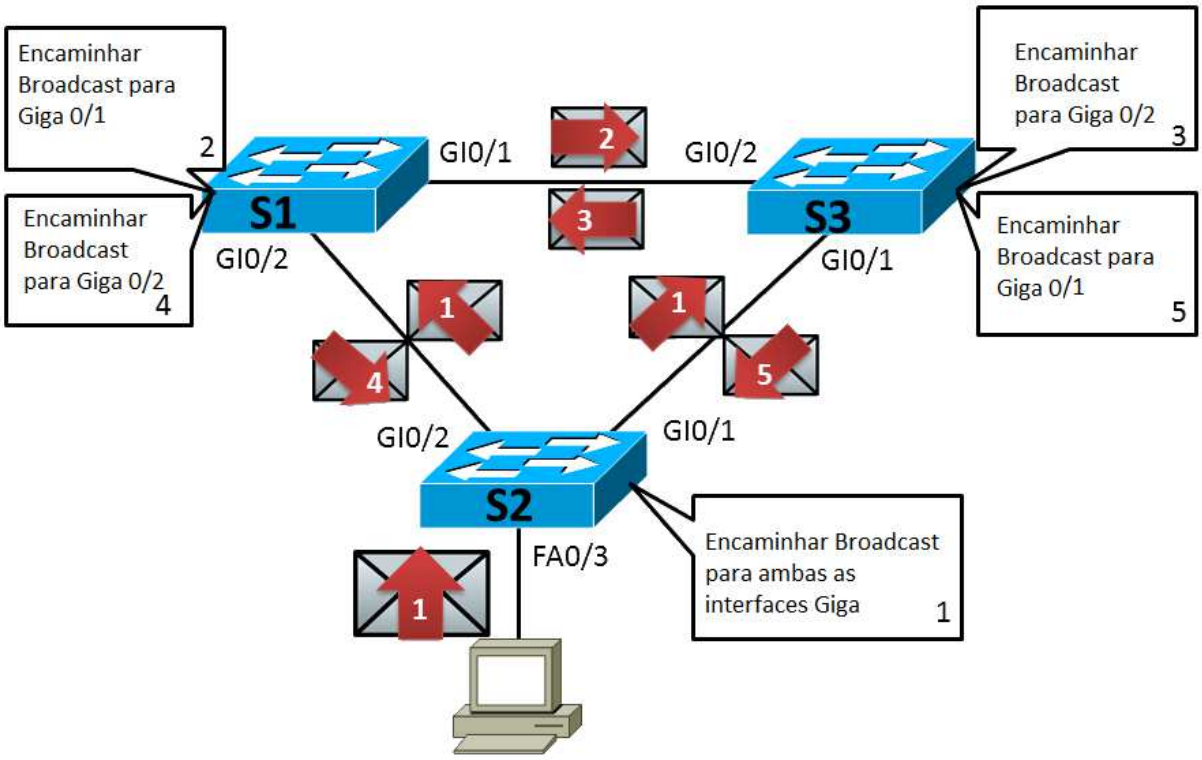
É um equívoco imaginar que graças a *switches*, não há necessidade de utilizarem-se *routers*. Apesar dos *hosts* de uma VLAN poderem se comunicar-se com todos os demais *hosts* de seu segmento, eles ficam impossibilitados de comunicar-se com *hosts* localizados em outras VLANs. Para permitir que esse tipo de comunicação aconteça é necessário que um dispositivo de camada 3 faça o roteamento entre as VLANs da mesma forma com LANs físicas, deve passar através de um *router*.[1]

2.7 *Spanning Tree*

Uma rede robusta não somente transfere pacotes ou quadros de uma maneira eficiente, mas também leva em consideração como se recuperar rapidamente de quaisquer falhas. Caminhos e equipamentos redundantes oferecem uma excelente prática para recuperação e convergência em caso de falha em alguma conexão, equipamento, rompimento de cabos, etc.

Em um ambiente de camada 2 caminhos redundantes porém não são desejados, muito menos permitidos. Um quadro *ethernet* não tem nenhum parâmetro que evite que o pacote fique eternamente percorrendo vários caminhos e equipamentos, como o TTL do cabeçalho IP. O funcionamento permissivo de um *switch* também define que caso um quadro ingresso tenha como destino um endereço egresso que ainda não foi aprendido ou configurado ele deverá ser propagado para todas as outras portas, menos a ingressa. *Switches* e *bridges* também não filtram *broadcasts* e/ou *multicasts*, sendo assim, um quadro cujo destino é um *broadcast* ethernet (ffff.ffff.ffff) pode ficar eternamente percorrendo a rede e sendo replicado para todas as outras portas, causando sobrecarga nos equipamentos de rede e por fim paralisação da mesma.[1] Um exemplo simples é mostrado na figura 4.[2]

Figura 4 - Tempestades de Broadcast



(Cisco Skills, 2012)

Para utilizar as vantagens de caminhos redundantes e evitar os problemas supracitados, foi definido pelo IEEE o padrão 802.1D, mais conhecido como *Spanning Tree Protocol* (STP). O protocolo implementa o algoritmo 802.1D do IEEE através de trocas de mensagens BPDU entre os *switches* para detecção de *loops*, e então remove o *loop* bloqueando as portas selecionadas nos equipamentos. Assim, o algoritmo garante que exista somente um, e somente um, caminho ativo entre 2 equipamentos de rede. Caso o caminho existente sofra algum acidente, o algoritmo trata de disponibilizar a porta, antes bloqueada para evitar o *loop* de camada 2, para tráfego, realizando assim a alteração para um caminho disponível automaticamente, diminuindo o tempo de indisponibilidade para os usuários e garantindo comunicação fim-a-fim.

No STP tradicional, porém, existem alguns problemas. Em caso de queda, a rede leva aproximadamente 30 segundos para atualizar os estados das portas (considerando o padrão dos temporizadores), tempo que pode ser crucial ao usuário. Para isso, o IEEE definiu uma “evolução” do STP tradicional e desenvolveu o padrão 802.1w, conhecido como *Rapid Spanning Tree Protocol* (RSTP).

Nos equipamentos adquiridos pela UTFPR, é utilizado um padrão híbrido conhecido como *Rapid Per-VLAN Spanning Tree Protocol* + (RPVST+). Esse padrão cria uma instancia para cada VLAN e é compatível com o STP tradicional.[1]

A maneira como o STP opera é encontrando todos os *link*s da rede e fechando caminhos redundantes entre eles. Para isso é feita uma eleição entre os *switches* da rede onde um se torna o *root bridge* (raiz); é então a partir dele que é montada a topologia da rede. Em uma rede irá existir somente um *root bridge*. Todas as portas deste *switch* são chamadas de *designated-ports* e se encontram em estado de encaminhamento, ou seja, elas podem enviar e receber pacotes.

Após a eleição da *root bridge*, uma nova eleição terá *início* para definir a função que será desempenhada pelas portas dos demais *switches*, também chamados de *nonroot bridge*. A primeira definição será quais serão as portas *root* em cada *switch*, esta é a porta com a menor métrica até a *root bridge*. Em seguida é feita a definição das *designated ports*, estas são as portas em um dado segmento de rede que possuem

permissão para enviar e receber. Por fim, as portas que não forem definidas nem como *root ports*, nem como *designated ports* serão bloqueadas.

As informações de controle utilizadas por *switches* para suas eleições e definições de funções e estado de portas são feitas através de um quadro chamado *Bridge Protocol Data Units* (BPDU), que são enviados para um endereço *multicast* de camada 2 do STP.

As portas dos *switches* possuem 5 estados de operação e para participar dos processos de eleição e participar do STP devem passar por estes estados:

- *Disable* - é o estado em que se encontram portas que foram desativadas pelo administrador da rede ou pelo sistema devido a algum problema técnico. Ela não faz parte da progressão normal do STP.
- *Blocking* - após inicializar, este é o estado em que as portas se encontram para evitar a formação de *loops*. Nesse estado a porta não envia nem recebe quadros e não adiciona os endereços MAC à sua Quadro.
- *Listening* - caso exista a possibilidade da porta se tornar uma porta *root* ou *designated* o *switch* irá promovê-la para este estado. Ainda incapaz de enviar e receber quadros, a porta pode receber BPDUs para poder fazer parte do processo de definição da topologia *spanning tree*. Caso não venha a fazer parte da topologia, a porta retornará ao estado *blocking*.
- *Learning* - após um tempo atuando no estado *listening*, a porta pode ser promovida a este estado, cuja única diferença ao anterior é o fato de poder aprender os endereços MAC dos quadros e inseri-los em sua Quadro de endereços.
- *Forwarding* - depois de mais um período atuando no estado *learning,* a porta pode chegar a este estado, em que poderá enviar e receber quadros e BPDUs e atualizar a Quadro de endereços MAC.

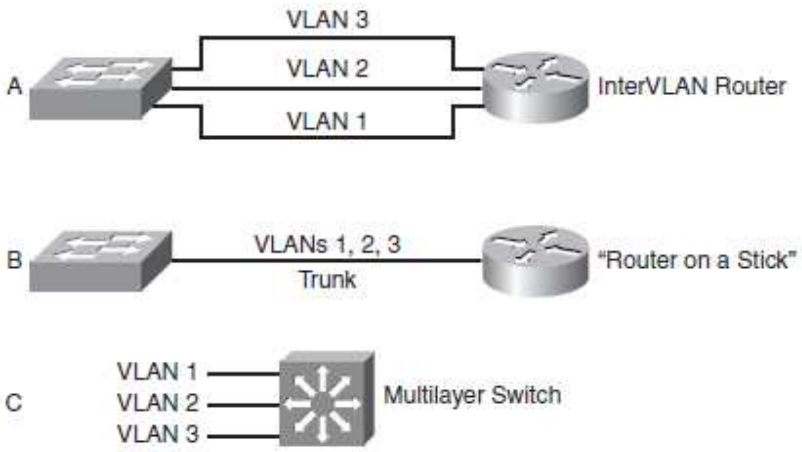
2.8 Comutação Multicamada

Uma rede de camada 2 pode ser definida como um domínio de difusão. Ela também pode existir como uma VLAN dentro de um ou mais *switches*. VLANs são essencialmente isoladas umas das outras, para que pacotes de uma VLAN não façam o cruzamento para outra.

Para transportar quadros entre VLANs, é necessário algum dispositivo de camada 3. Tradicionalmente, esta tem sido uma tarefa atribuída ao *router*, onde o mesmo precisa ter uma conexão lógica ou física para cada VLAN, para assim encaminhar pacotes entre elas, em um processo conhecido como Roteamento InterVLAN.

O Roteamento InterVLAN pode ser feito por um *router* externo que conecta cada VLAN em um *switch*. Podem-se utilizar conexões físicas separadas ou um único tronco. A figura 5 mostra essas conexões, bem como a utilização de um *switch* multicamada, que combina as funções de roteamento e comutação em um único dispositivo, dispensando assim a necessidade de um *router* externo.

Figura 5 - Roteamento InterVLAN



(Hucaby, 2010)

Assim como um *router*, um *switch* multicamadas pode designar um endereço IP para uma interface específica. Também pode ser designado um endereço IP para uma interface lógica que representa toda a VLAN. Essa interface é conhecida como *switch*ed *virtual interface (*SVI). Essa interface é utilizada como default gateway para quaisquer usuários conectados a essa VLAN. Eles também utilizarão a interface de camada 3 para se comunicar com dispositivos que estão fora do seu domínio de difusão.

Como dito anteriormente, um *switch* multicamada agrega ambas as funções de um *switch* de camada 2 e *router* camada 3. Sendo assim, as funções de roteamento são feitas em *hardware* em paralelo às funções de comutação, sem que haja perda de desempenho, inclusive na aplicação de listas de acesso e QoS. [1]

Este trabalho abordará o conceito de listas de acesso. As ACLs, como também são conhecidas, são filtros IP utilizados em equipamento para permitir, ou não, a passagem de pacotes de uma determinada rede para outra, assim como VLANs e portas da camada de transporte. [10]

2.9 Agregação de *link*s

Em uma rede comum, vários usuários podem estar conectados a um *switch* de acesso e solicitar um serviço externo às suas respectivas VLANs. Nesse ambiente, também é comum uma conexão única entre os *switches*. Quando se juntam as duas condições, pode ocorrer uma condição conhecida como gargalo de rede. Em uma analogia, imagine uma rodovia com 5 pistas, sendo que em um determinado momento todas as pistas são “afuniladas” em somente uma, causando assim engarrafamento em fila e lentidão. O mesmo pode ocorrer se vários usuários tiverem de passar pelo tronco entre os *switches* ao mesmo tempo.

Em uma rede *ethernet* as portas podem ter várias velocidades: 10 Mbps, 100 Mbps, 1 Gbps e 10 Gbps. Porém, nem sempre a velocidade máxima de uma porta pode ser o suficiente.

Para resolver essa questão e evitar que seja necessária a troca de todo o equipamento, foi desenvolvida a capacidade de agregar portas de um *switch*. Assim torna-se possível expandir a capacidade do *link* entre 2 *switches* sem ter de comprar um novo equipamento com a próxima magnitude de velocidade.

Os *link*s podem ser agregados de 2 a 8, sendo que todos os *link*s agregados deverão ter as mesmas configurações e capacidades. Assim um *link* EtherChannel – como também é chamado – é visto somente como um único *link* lógico. Se um dos *link*s porém sofrer uma queda, o tráfego é desviado automaticamente para o *link* adjacente em questão de milissegundos. Assim, quando os *link*s são restaurados o tráfego voltará a ser distribuído entre eles. [1]

Existem 2 protocolos para negociação de *EtherChannel*:

- *Port Aggregation Protocol* (PAgP) – Protocolo proprietário Cisco.
- *Link Aggregation Control Protocol* (LACP) – Definido pelo padrão IEEE 802.3 Cláusula 43 “*Link Aggregation*”.

São poucas as diferenças entre os protocolos. O LACP porém define que um *switch*, o que tiver a MENOR prioridade é responsável pelos parâmetros do *link*, sendo que o PAgP muda automaticamente os parâmetros de todas as portas caso uma seja alterada. O PAgP aceita no máximo 8 portas agregadas e o LACP 16, sendo 8 ativas e 8 em espera. [3]

Por ser um padrão aberto e compatível com outros fabricantes, o protocolo utilizado nesse trabalho será o LACP.

2.10 DHCP clandestino e DHCP *Snooping*

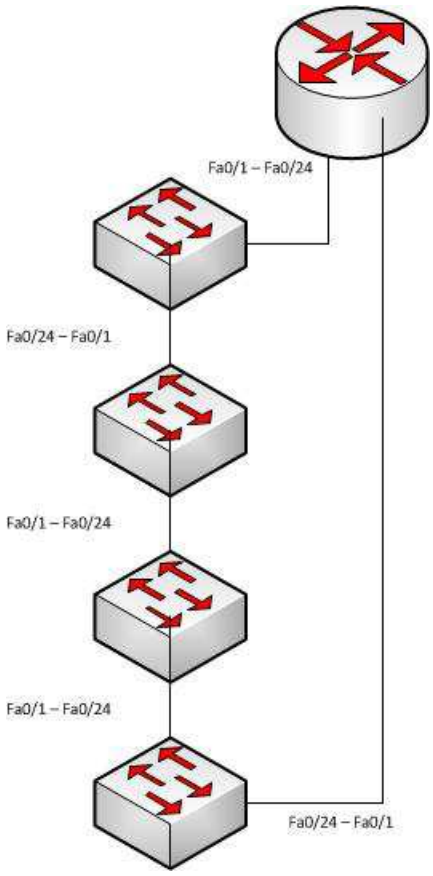
Normalmente utilizado para prevenir ataques onde uma entidade maliciosa se passa por um servidor DHCP legítimo na rede, porém, com endereços utilizados por ele para realizar um ataque do tipo *man-in-the-middle* onde essa entidade intercepta os pacotes enviados pelo usuário antes que eles cheguem ao destinatário correto. [1]

DHCP *Snooping* é uma funcionalidade de segurança que monta e mantém uma Quadro de associações DHCP e filtra mensagens DHCP não confiáveis. Uma mensagem não confiável pode vir de fora da rede ou *firewall* como forma de ataque, ou quando algum usuário, inadvertidamente, adiciona outro servidor DHCP sem permissão, como *routers* sem fio e outros dispositivos que têm integrados um servidor DHCP. O protocolo age como um *firewall* entre *hosts* não-confiáveis e servidores DHCP. [4]

2.11 Cascateamento e empilhamento

Cascateamento é o método de conexão linear onde cada *switch* se conecta bi-direcionalmente através de uma ou mais conexões em outro *switch*. [5] A figura 6 exemplifica o conceito.

Figura 6 – Cascateamento



(Autoria própria)

Apesar de ser um método não recomendado, muitas redes o utilizam, pois assim torna-se possível adicionar mais portas de acesso com o menor custo de cabeamento.[6] Caso necessário, recomenda-se ligar o último *switch* da pilha ao equipamento de distribuição, para aumentar a redundância.

Por ser desencorajada essa topologia não pôde ser verificado em nenhuma bibliografia. Porém a mesma foi testada através do simulador Packet Tracer 5.3.

Existe uma técnica alternativa ao cascateamento que é recomendada e suportada pelos *switches* Cisco 3750 que se chama empilhamento, ou *stack*. Através do *stack*, até 9 *switches* podem ser empilhados, sendo interconectados por *link*s específicos de 32 Gbps. Quando são empilhados, os equipamentos se comportam como uma única unidade lógica. A cada *switch* é atribuído um papel, sendo um mestre

por pilha que serve de centro de controle. A cada pilha é atribuída uma única configuração, que é distribuída aos demais dispositivos. Em caso de queda do *switch* mestre, qualquer outro membro pode assumir as suas funções, evitando assim interrupções na rede. [7]

2.12 Gerenciamento

Para o funcionamento correto de uma rede, não basta somente implementar protocolos e *link*s redundantes e rápido transporte de dados. Suponha que um *link* redundante por alguma razão sofra algum acidente. Se o administrador não estiver ciente não poderá tomar nenhuma ação para realizar a correção necessária. Se não for realizada a correção, a rede não terá mais *link*s redundantes para garantir a disponibilidade.[1] Existem várias formas de se monitorar e analisar uma rede: Através do acesso local ao console do equipamento, acesso remoto via SSH, mensagens de *Syslog* e SNMP.

Para realizar acesso físico ao console do equipamento, assim como realizar sua configuração inicial, é necessária a utilização de um cabo *rollover*, onde a interface no equipamento é RJ-45 e no terminal que realizará a emulação uma interface serial DB-9 e as seguintes configurações para a conexão serial: 9600 baud; 8 data *bits*; Sem paridade; 1 *stop bit* [8]. Através da console, é possível ver mensagens de erro e realizar todas as configurações necessárias em um equipamento Cisco através da *command-line interface*, ou CLI, desde que o usuário tenha permissão de acesso. [9]

Após a configuração inicial do equipamento, é possível realizar o acesso à CLI remotamente, através de 2 protocolos: Telnet e SSH. Por não ser seguro, utilizaremos somente acesso via SSH, utilizando a interface virtual do *switch* – SVI. [1]

Os *switches Catalyst* também podem ser configurados para gerar mensagens que descrevem eventos importantes. Essas mensagens, do inglês *system message*

logs, ou Syslogs, podem ser coletadas e analisadas para determinar o que ocorreu, quando e quão severo o evento foi. As mensagens podem ser coletadas via console, SSH (através do comando terminal monitor) e enviadas para um servidor *Syslog*.

Por fim, o *switch* também pode compartilhar informações através do protocolo SNMP. Por questões de segurança, será tratado o padrão SNMPv3. Através do SNMP, o administrador pode coletar informações do equipamento pelas OIDs contidas na MIB do *switch*, assim como receber alertas disparados devido a algum evento e definir algumas variáveis remotamente. [1]

3 METODOLOGIA

A seguir, os protocolos e temas abordados nos itens anteriores serão aplicados à necessidade da UTFPR propostas por este trabalho.

3.1 Questionário

O seguinte questionário foi submetido à COGETI com o intuito de averiguar quais as necessidades, exigências e recursos do projeto:

- Pergunta: Quais equipamentos foram adquiridos?
  - Resposta: Foram adquiridos 49 *switches* de camada 2 e 2 *switches* de camada 3.
- P: Os equipamentos deverão apenas substituir os atuais ou toda a rede será reestruturada (o que envolveria refazer todo o cabeamento)?
  - R: Para efeito deste trabalho, a rede deverá ser reestruturada seguindo os conceitos de *best practice* recomendados pelo fabricante.
- P: Como será feita a divisão de VLANs (departamentos, blocos)?
  - R: As VLANs serão divididas por departamentos e setores.
- P: Como será feito o roteamento interVLAN e para fora da rede?
  - R: Como um departamento não se comunica com outro, as VLANs NÃO devem se comunicar. A única comunicação da VLAN será entre ela e a rota default.
- P: Qual será a faixa de IPs designados para cada VLAN?
  - R: O desenho atual possui uma rede 172.17.0.0 /16 que pode ser dividida em várias redes /24, pois não existe a necessidade de termos mais de 254 *hosts* em cada VLAN. Alguns departamentos têm uma faixa de IP

válida, são eles: 200.134.25/24 (Geral), 200.134.26/24 (Ecoville), 200.134.10/24 (DAINF), 200.134.9/24 (CITEC), 200.17.96/24 (CPGEI).

- P: Qual é e quantas saídas existem para a Internet?
  - R: Existe apenas uma saída para a *Internet*, que também é a rota *default* citada na pergunta anterior. Esse *link* é entregue pela Reitoria e não faz parte da administração da COGETI.
- P: Qual será a política de segurança (restritiva ou permissiva)?
  - R: Devido à grande quantidade de usuários, perfil dos mesmos, mobilidade, rotatividade entre eles, será adotada uma política PERMISSIVA, que cause o menor impacto possível ao usuário.
- P: Existe algum problema causado pelo usuário, tal como ataque ou mau uso?
  - R: Os problemas que mais ocorrem são causados devido a *loop* na rede e conexão de *routers* sem fio com servidor DHCP habilitado.
- P: A configuração do DHCP será feita nos *switches* camada 3 ou em um servidor dedicado?
  - R: Para deixar a solução mais “limpa”, visto que o DHCP é um protocolo referente à camada de aplicação, seria mais coerente separá-lo em um servidor dedicado.

3.2 Migração

Pelas definições da COGETI, a migração da rede atual para a proposta não requer nenhuma ação específica, a não ser a garantia da comunicação do departamento migrado com a *Internet* já que não existe, e não é desejável, a comunicação interVLAN. Para tanto é necessário apenas duplicar o *link* da Reitoria através de um *switch* simples, sendo que uma ponta se conecta à rede antiga e outra à

rede nova. A migração deverá ser feita sempre em um departamento, migrando-o todo, para a rede nova.

A seguir, será abordada a realidade atual da rede da UTFPR Cämpus Curitiba, bem como a proposta de uma nova topologia, suas funcionalidades e como configurar nos equipamentos adquiridos.

Para todos os exemplos de configuração adiante, seguiremos os padrões adotados pela editora Cisco Press: [1].

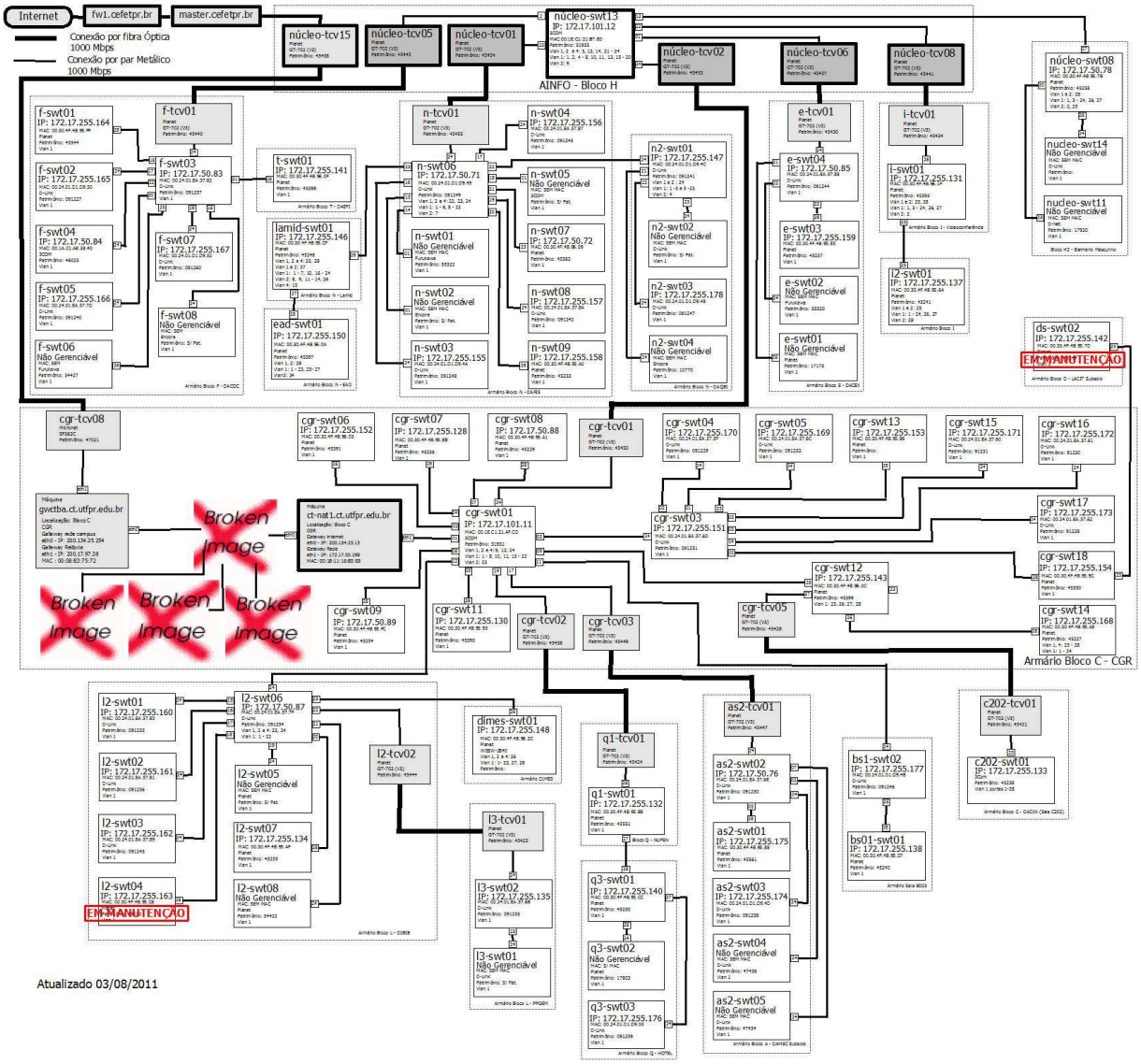
- Negrito indica comandos e palavras-chave que são digitados literalmente como são mostrados.
- Itálico indicam argumentos para os quais devem ser fornecidos valores desejados.
- Barras verticais (|) separam alternativas e elementos mutualmente exclusivos.
- Colchetes ([ ]) indicam um elemento opcional.
- Chaves ({ }) indicam uma opção obrigatória.
- Colchetes com chaves ([{ }]) indicam uma opção obrigatória com um elemento opcional.

3.3 Topologia atual

A maneira como a rede do Cämpus Curitiba está estruturada atualmente é extremamente plana. Não existe uma distinção hierárquica das funções desempenhadas pelos equipamentos que compõem a topologia. Em sua grande maioria possui *switches* e equipamentos de acesso para que os usuários possam se conectar a rede. A maioria dos equipamentos está disposta em uma única VLAN, a VLAN1, e embora existam outras VLANs (VLAN 2 e VLAN 4) configuradas elas são interfaces utilizadas para controle e testes. Este tipo de arranjo pode ser problemático

pelo fato de possuir um domínio de *broadcast* permitindo que ocorram tempestades de *broadcast* e tornando mais difícil o diagnóstico de problemas (como a inserção de servidores DHCP clandestinos na rede). Outra carência que pode ser ressaltada é o uso de um número considerável de equipamentos não gerenciáveis. Estes equipamentos apresentam algumas carências em comparação com seus equivalentes gerenciáveis. A figura 7 foi disponibilizada pela COGETI e representa a atual rede da UTFPR Cämpus Curitiba.

Figura 7 - Topologia Atual



(COGETI, 2012)

3.4 Nova topologia

A Cisco Systems define um modelo hierárquico de desenho de rede dividido em 3 camadas: Núcleo, Distribuição e Acesso.[1] Dentro deste modelo, destacam-se:

- As VLANs e domínios de difusão devem convergir na camada de distribuição.
- O modelo pode ser simplificado e colapsado, combinando as camadas de distribuição e núcleo em uma, chamada de Núcleo Colapsado.

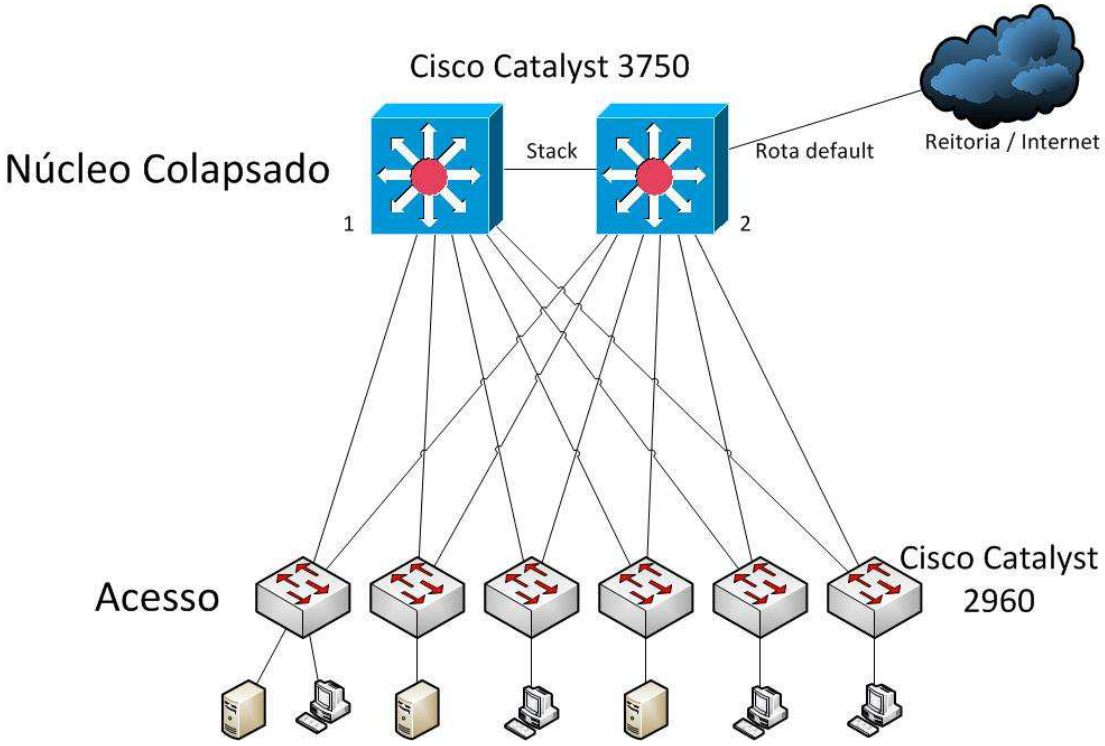
Outro fato que devemos atentar à formulação do novo desenho são os novos equipamentos e suas capacidades. O Quadro 1 detalha os *switches* e suas características [11] [12] [13]. As portas de dupla finalidade são portas RJ-45 ou SFP, porém apenas um tipo pode ser usado por vez:

Quadro 1 - Os Equipamentos e suas descrições (Autoria própria)

Modelo	**Quantidade**	**Camada**	**Descrição**	**Up*link*s**
Cisco Catalyst 2960-24TC-L	4	2	24 Portas Ethernet 10/100	2 portas *dupla finalidade*
Cisco Catalyst 2960-24TT-L	7	2	24 Portas Ethernet 10/100	2 Portas Ethernet 10/100/1000
Cisco Catalyst 2960-48TC-L	9	2	48 Portas Ethernet 10/100	2 portas *dupla finalidade* (10/100/1000 ou SFP)
Cisco Catalyst 2960-48TT-L	26	2	48 Portas Ethernet 10/100	2 Portas Ethernet 10/100/1000
Cisco Catalyst 2960G-24TC-L	1	2	24 portas Ethernet 10/100/1000, das quais 4 são *dupla finalidade* (10/100/1000 ou SFP).	4 portas *dupla finalidade* (10/100/1000 ou SFP)
Cisco Catalyst 2960S-24TS-L	2	2	24 portas Ethernet 10/100/1000 Stack	4 portas 1 Gigabit Ethernet SFP
WS-C3750G-24TS-S1U	2	3	24 portas Ethernet 10/100/1000 Stack	4 portas SFP-based Gigabit Ethernet

Analisando os dados acima, foi sugerido o desenho abaixo para a nova topologia, ilustrado pela figura 8:

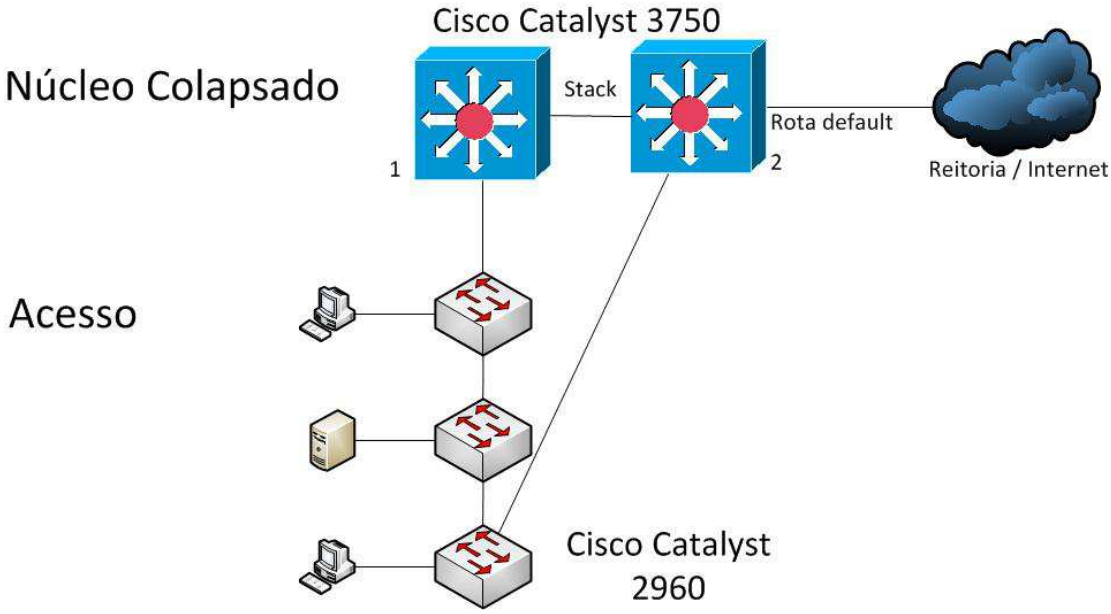
Figura 8 - A Nova Topologia



(Autoria própria)

Devido à quantidade de portas disponíveis no núcleo colapsado, e também a restrições como cabeamento e espaço físico, pode-se utilizar uma topologia em cascata na camada de acesso. Para garantir a redundância, é ligada uma porta do primeiro *switch* e uma porta do último. A figura 9 exemplifica essa topologia.

Figura 9 - A nova topologia com cascateamento



(Autoria própria)

Não foi encontrada nenhuma recomendação quanto a quais portas utilizar em cada conexão. Sendo assim, foram definidas as conexões com o intuito de padronizar a rede. O Quadro 2 lista os padrões:

Quadro 2 - Padrões de conexão entre os *switches* (Autoria própria)

Modelo do *Switch*	**Tipo de Up*link***	**Ligação normal**	**Ligação em cascata**
Cisco Catalyst 2960-24TC-L	2 portas *dupla finalidade*	Pelas portas de dupla finalidade – caso SFP através de conversor de mídia. Porta 1 para núcleo 1. Porta 2 para núcleo 2.	Pelas portas de dupla finalidade – Caso SFP para o núcleo, através de conversor de mídia. *Switch* 1 porta 1 → núcleo *Switch* 1 porta 2 → *Switch* 2 porta 1 *Switch* 2 porta 2 → núcleo
Cisco Catalyst 2960-24TT-L	2 Portas Ethernet 10/100/1000	Pelas de portas Up*link* RJ-45. Porta 1 para núcleo 1. Porta 2 para núcleo 2.	Pelas portas de Up*link* RJ-45. *Switch* 1 porta 1 → núcleo *Switch* 1 porta 2 → *Switch* 2 porta 1 *Switch* 2 porta 2 → núcleo
Cisco Catalyst 2960-48TC-L	2 portas *dupla finalidade* (10/100/1000 ou SFP)	Pelas portas de dupla finalidade – caso SFP através de conversor de mídia. Porta 1 para núcleo 1. Porta 2 para núcleo 2.	Pelas portas de dupla finalidade – Caso SFP para o núcleo, através de conversor de mídia. *Switch* 1 porta 1 → núcleo *Switch* 1 porta 2 → *Switch* 2 porta 1 *Switch* 2 porta 2 → núcleo
Cisco Catalyst 2960-48TT-L	2 Portas Ethernet 10/100/1000	Pelas de portas Up*link* RJ-45. Porta 1 para núcleo 1. Porta 2 para núcleo 2.	Pelas portas de Up*link* RJ-45. *Switch* 1 porta 1 → núcleo *Switch* 1 porta 2 → *Switch* 2 porta 1 *Switch* 2 porta 2 → núcleo
Cisco Catalyst 2960G-24TC-L	4 portas *dupla finalidade* (10/100/1000 ou SFP)	Pelas portas de dupla finalidade – caso SFP através de conversor de mídia. Porta 1 para núcleo 1. Porta 2 para núcleo 2.	Pelas portas de dupla finalidade – Caso SFP para o núcleo, através de conversor de mídia. *Switch* 1 porta 1 → núcleo *Switch* 1 porta 2 → *Switch* 2 porta 1 *Switch* 2 porta 2 → núcleo
Cisco Catalyst 2960S-24TS-L	4 portas 1 Gigabit Ethernet SFP	Caso pelas portas SFP, através de conversor de mídia. Porta 1 para núcleo 1. Porta 2 para núcleo 2. Caso portas RJ-45 utilizar porta 1 para núcleo 1 e porta 24 para núcleo 2.	Caso SFP para o núcleo, através de conversor de mídia. *Switch* 1 porta 1 → núcleo *Switch* 1 porta 2 → *Switch* 2 porta 1 *Switch* 2 porta 2 → núcleo Caso portas RJ-45: Up*link*s para núcleo sempre pela porta 1. Up*link*s entre *switch* da cascata pela porta 24

Para os equipamentos do núcleo colapsado os padrões adotados serão:

- Ligação Normal
  - *Switch* acesso porta 1 → Núcleo 1 – porta RJ-45 indiferente.
  - *Switch* acesso porta 2 → Núcleo 2 – porta RJ-45 indiferente.

- Ligação em cascata
  - Primeiro *switch* de acesso → Núcleo 1 – porta RJ-45 indiferente.
  - Segundo *switch* de acesso → Núcleo 2 – porta RJ-45 indiferente.

Apesar das portas nos *switches* do núcleo colapsado serem indiferentes, recomenda-se utilizar as portas de mesmo número em ambos os *switches* para facilitar o gerenciamento, sendo o número da porta do *switch* de acesso referenciado não pela porta dos *switches* do núcleo, e sim pelo número do mesmo. Por exemplo:

Switch de acesso 1 porta de up*link* 1 → Núcleo 1 porta 1

Switch de acesso 1 porta de up*link* 2 → Núcleo 2 porta 1

3.5 VLANs UTFPR

Redes comutadas podem ser subdivididas em VLANs. Essas subdivisões oferecem a tecnologia para sobrepor às limitações de uma rede plana. Pela definição, cada VLAN é um único domínio de difusão. Portanto, todos os dispositivos conectados à VLAN recebem *broadcasts* enviados por qualquer outro membro da VLAN. Membros de VLANs diferentes não receberão esses mesmos *broadcasts*.[1]

Na topologia proposta, cada departamento acadêmico, diretoria e coordenadoria farão parte de uma VLAN específica.[21][22] De acordo com o questionário submetido à COGETI algumas VLANs terão faixas de IP válido. Para todas as outras, foi utilizada a faixa privada 172.17.0.0 /16. Essa faixa foi subdividida em várias redes /24, uma para cada VLAN. As faixas de IP privados, ou seja, que não se aplicam à Internet, são definidas pela RFC 1918.[23]

Para as redes privadas, o terceiro octeto do endereço de rede foi referenciado no número de identificação da VLAN. Por exemplo, a rede 200.134.10.0 /24, pertencente ao DAINF, foi atribuída à VLAN 10. A mesma lógica se aplica às redes privadas. Porém para evitar uma sobreposição entre as identificações da VLAN e entre

os IPs privados e públicos, o identificador da VLAN é igual ao terceiro octeto da rede somado ao número 200, iniciando pela VLAN 60. Assim conseguimos endereçar todos os departamentos, diretorias e coordenadorias necessários, deixando a faixa de VLANs 2 a 255 para redes com IP verdadeiro e uma faixa de subdivisões de 9 VLANs, caso necessário. A única exceção é a VLAN administrativa, que foi reservada o endereço de rede 172.17.1.0 com a VLAN ID 1 – que é a VLAN padrão nos equipamentos.

O Quadro 3 mostra como ficou essa separação, assim como a faixa de IPs reservada para cada VLAN, seu endereço de rede, faixa de IPs reservados para endereços fixos, como impressoras e servidores, faixa de IPs dinâmicos disponíveis para leasing via DHCP e a SVI – que também é o *Default Gateway* de cada VLAN.

Quadro 3 - As VLANs e endereços IP privados (Autoria própria)

Depto. / Nome VLAN	VLAN ID	Endereço de Rede	Faixa de IPs Fixos	Faixa de IPs DHCP	Default Gateway (SVI)
ADMINISTRATIVA	1	172.17.1.0	1 - 50	51 - 253	172.17.1.254
DIRGE	260	172.17.60.0	1 - 50	51 - 253	172.17.60.254
DIRGRAD	270	172.17.70.0	1 - 50	51 - 253	172.17.70.254
DIRPPG	280	172.17.80.0	1 - 50	51 - 253	172.17.80.254
DIREC	290	172.17.90.0	1 - 50	51 - 253	172.17.90.254
DIRPLAD	300	172.17.100.0	1 - 50	51 - 253	172.17.100.254
COGERH	310	172.17.110.0	1 - 50	51 - 253	172.17.110.254
COGETI	320	172.17.120.0	1 - 50	51 - 253	172.17.120.254
DAELN	330	172.17.130.0	1 - 50	51 - 253	172.17.130.254
DAELT	340	172.17.140.0	1 - 50	51 - 253	172.17.140.254
DAMEC	350	172.17.150.0	1 - 50	51 - 253	172.17.150.254
DACOC	360	172.17.160.0	1 - 50	51 - 253	172.17.160.254
DADIN	370	172.17.170.0	1 - 50	51 - 253	172.17.170.254
DAFIS	380	172.17.180.0	1 - 50	51 - 253	172.17.180.254
DACEX	390	172.17.190.0	1 - 50	51 - 253	172.17.190.254
DALEM	400	172.17.200.0	1 - 50	51 - 253	172.17.200.254
DAQBI	410	172.17.210.0	1 - 50	51 - 253	172.17.210.254
DAESO	420	172.17.220.0	1 - 50	51 - 253	172.17.220.254
DAMAT	430	172.17.230.0	1 - 50	51 - 253	172.17.230.254
DAGEE	440	172.17.240.0	1 - 50	51 - 253	172.17.240.254
DAEFI	450	172.17.250.0	1 - 50	51 - 253	172.17.250.254

O Quadro 4 mostra os mesmos conceitos da Quadro anterior, porém aplicada às redes com IP válido.

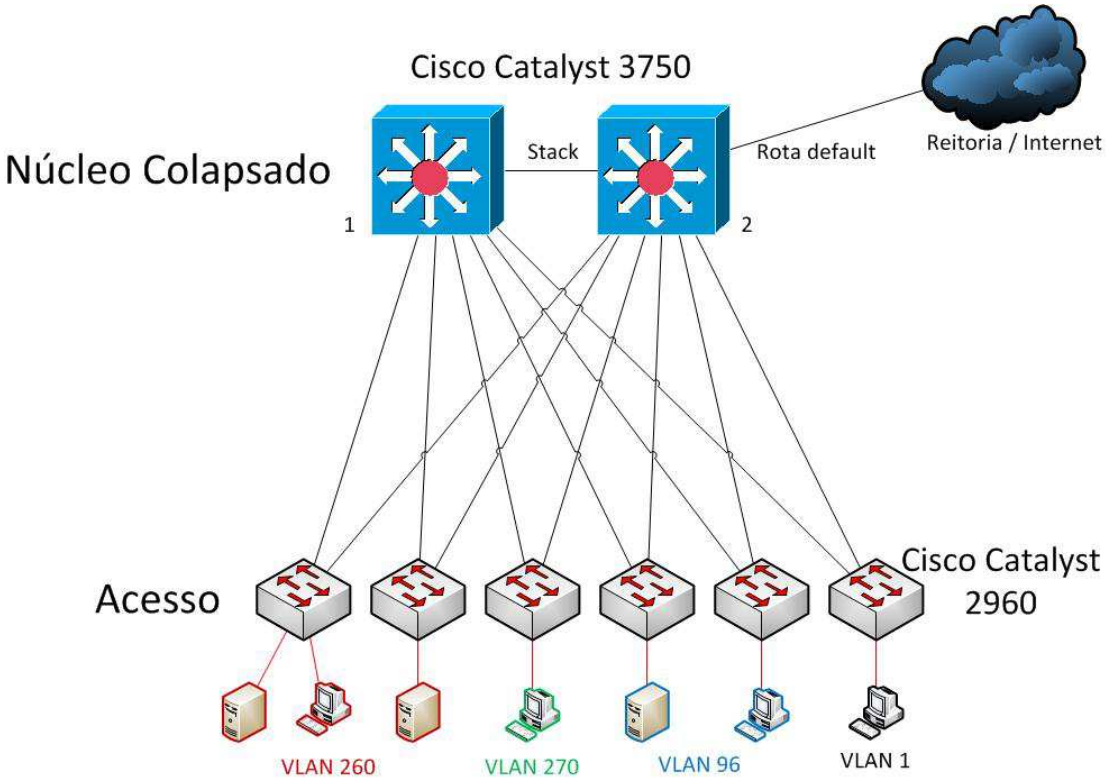
Quadro 4 - As VLANs e endereços IP públicos (Autoria própria)

Depto. / Nome VLAN	VLAN ID	Endereço de Rede	Faixa de IPs Fixos	Faixa de IPs DHCP	Default Gateway (SVI)
GERAL	25	200.X.X.X	1 - 50	51-253	200.134.25.254
DAINF	10	200.X.X.X	1 - 50	51-253	200.134.10.254
ECOVILLE	26	200.X.X.X	1 - 50	51-253	200.134.26.254
CITEC	09	200.X.X.X	1 - 50	51-253	200.134.9.254
CPGEI	96	200.X.X.X	1 - 50	51-253	200.17.96.254

Obs: Os endereços públicos foram omitidos a pedido da banca avaliadora por motivos de segurança.

A figura 10 mostra a topologia proposta com as conexões em destaque vermelho como os *link*s de acesso. Somente para ilustração, cada grupo de usuários finais contornados por uma cor também representam uma VLAN.

Figura 10 - A nova topologia com VLANs



(Autoria própria)

Para configurar as portas dos equipamentos como membro de uma determinada VLAN, é necessário definir que a porta em questão se trata de uma porta de acesso. O exemplo a seguir demonstra essa configuração:

Switch(config)# **interface** *tipo módulo/porta*

Switch(config-if)# ***switch*port**

Switch(config-if)# ***switch*port mode access**

Switch(config-if)# ***switch*port access vlan** *id-da-vlan*

As 3 primeiras linhas representam, respectivamente, a escolha da interface física que será configurada como membro de uma VLAN, a configuração da porta como

camada 2 (que é o padrão em *switches*) e a associação porta à VLAN desejada. O VLAN ID pode variar entre 1 a 1005, ou 1 a 4094.

Para cada VLAN configurada é necessário também configurar a SVI. Ela é uma interface lógica que representa toda a VLAN com um endereço IP de camada 3. É através desse endereço que é possível realizar acesso remoto e verificações de conectividade, como ping. A configuração é exemplificada a seguir:

Switch(config)# **interface vlan** *id-da-vlan*

Switch(config-if)# **ip address** *endereço_ip máscara_de_subrede*

Switch(config-if)# **no shutdown**

Por fim, é necessário desabilitar o *VLAN Trunking Protocol*. Essa funcionalidade permite replicar a VLANs em um grupo de *switches*, onde os equipamentos configurados como servidores passam suas VLANs para os outros, assim se uma VLAN for criada ou apagada é acrescentado um número de revisão e a informação replicada nos *switches* com números de revisão menores. Porém, ela apresenta um problema inerente, pois caso algum *switch* seja conectado na rede com um número de revisão maior, mas com as VLANs erradas, todos os *switches* irão replicar essa configuração erroneamente. Devido a essa possibilidade, a COGETI decidiu por não utilizar essa funcionalidade. Os comandos para desabilitar essa funcionalidade são demonstrados a seguir:

Switch(config)# **vtp mode transparent**

O Quadro 5 mostra os comandos utilizados para verificação das configurações, bem como sua explicação.[1]

Quadro 5 - Comandos de verificação de VLANs (Autoria própria)

Comando	Função
Switch#**show interface** *tipo módulo/porta* ***switch*port**	Mostra configuração da porta na camada 2, bem como a qual VLAN está associada.

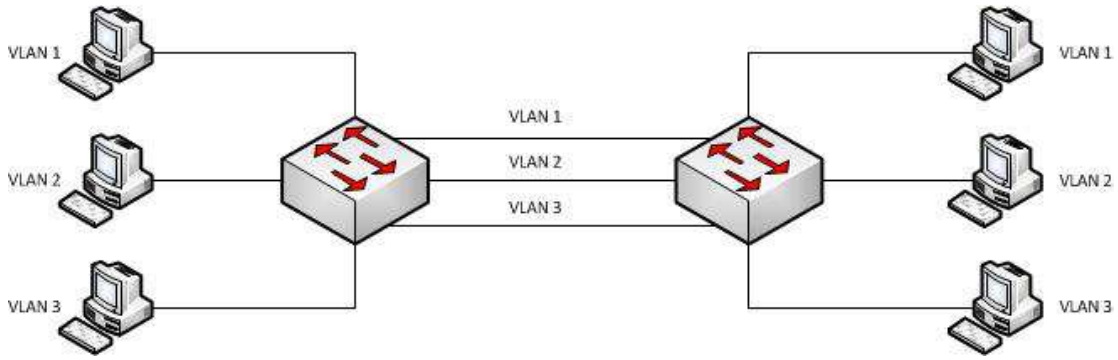
Switch#**show vlan**	Mostra a lista de todas as VLANs, juntamente com as portas associadas a cada VLAN.
Switch#**show ip interface brief**	Exibe todas as interfaces, estado e endereço IP das mesmas.

3.6 Troncos

No nível de acesso, cada porta do *switch* fornece conectividade para uma única VLAN. Os dispositivos conectados nessas portas não têm qualquer conhecimento sobre a estrutura da VLAN, dando a impressão de estarem em um segmento físico comum da rede. Vale lembrar que a comunicação de um usuário em uma VLAN para outra só é possível através da intervenção de um dispositivo adicional: um *router* ou um *switch* multicamada.

As VLANs também podem estar distribuídas em vários *switches*, tornando usuários que não estão conectados no mesmo equipamento membros do mesmo segmento de rede. Porém, para interconectar esses equipamentos seria necessária uma conexão para cada VLAN existente. Embora possa parecer viável à primeira vista, percebe-se o problema quando escalamos essas conexões para um *switch* que trabalhe como um concentrador de outros *switches* como os utilizados nas camadas de distribuição e núcleo, onde provavelmente passam várias VLANs oriundas de vários outros *switches* de acesso. A figura 11 ilustra essas conexões.

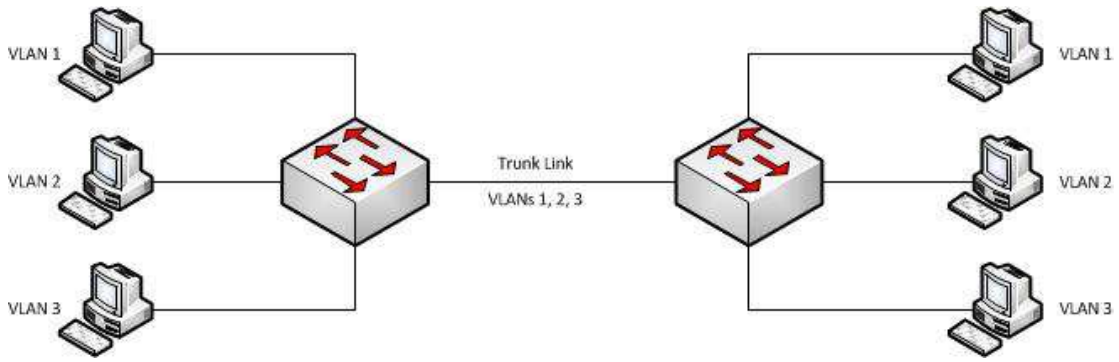
Figura 11 - Conexões entre VLANs sem tronco



(Autoria própria)

Como podemos observar na figura 11, essa é uma solução não-escalável. Com novas VLANs sendo adicionadas à rede, o número de conexões necessárias pode crescer rapidamente. Para resolver esse problema, foi desenvolvida uma técnica para "agrupar" todas as VLANs em um único *link*, chamada de *Trunking* – do inglês "tronco". Um tronco pode transportar mais de uma VLAN através de uma única porta. Esses *link*s são mais utilizados quando *switches* se conectam a outros *switches* ou a *routers*. Esses *link*s não são designados a nenhuma VLAN específica; ao invés disso uma, várias ou todas as VLANs ativas podem ser transportadas entre os dispositivos através de uma única conexão física. A figura 12 ilustra essa conexão.

Figura 12 - Conexões entre VLANs com tronco



(Autoria própria)

Como podemos perceber, novas VLANs podem ser adicionadas à rede utilizando a mesma ligação física entre os *switches*, aumentando consideravelmente a escalabilidade da mesma.

Como transportam várias VLANs, um *switch* deve conseguir associar os quadros às suas respectivas VLANs. Essa identificação é chamada de *tagging* e é feita através de uma identificação única da VLAN que é adicionada ao cabeçalho do quadro. Por causa dessa alteração no cabeçalho, quadros que têm a informação de *trunking* não podem ser lidos por computadores ou equipamentos que não tenham suporte a esses protocolos em suas placas de rede e/ou interfaces. Existem 2 métodos de realizar essa identificação nos equipamentos da Cisco Systems:

- Inter-*Switch Link* (ISL)
- IEEE 802.1Q

Como proposto, iremos analisar o padrão aberto, neste caso sendo o IEEE 802.1Q, também conhecido como dot1q. Para realizar a configuração de uma porta como *trunk* e utilizando o protocolo 802.1Q, utilize os seguintes comandos:

Switch(config)# **interface** *tipo módulo/porta*

Switch(config-if)# ***switch*port**

Switch(config-if)# ***switch*port trunk encapsulation dot1q**

Switch(config-if)# ***switch*port trunk native vlan** *id-da-vlan*

Switch(config-if)# ***switch*port trunk allowed vlan** {*lista-de-vlans*| **all |**

{add | except | remove*} lista-de-vlans }*

Switch(config-if)# ***switch*port mode trunk**

As 3 primeiras linhas representam, respectivamente, a escolha da interface física em que será configurado o *trunk*, a configuração da porta como camada 2 (que é o padrão em *switches*) e a escolha do protocolo 802.1Q para o *link*.

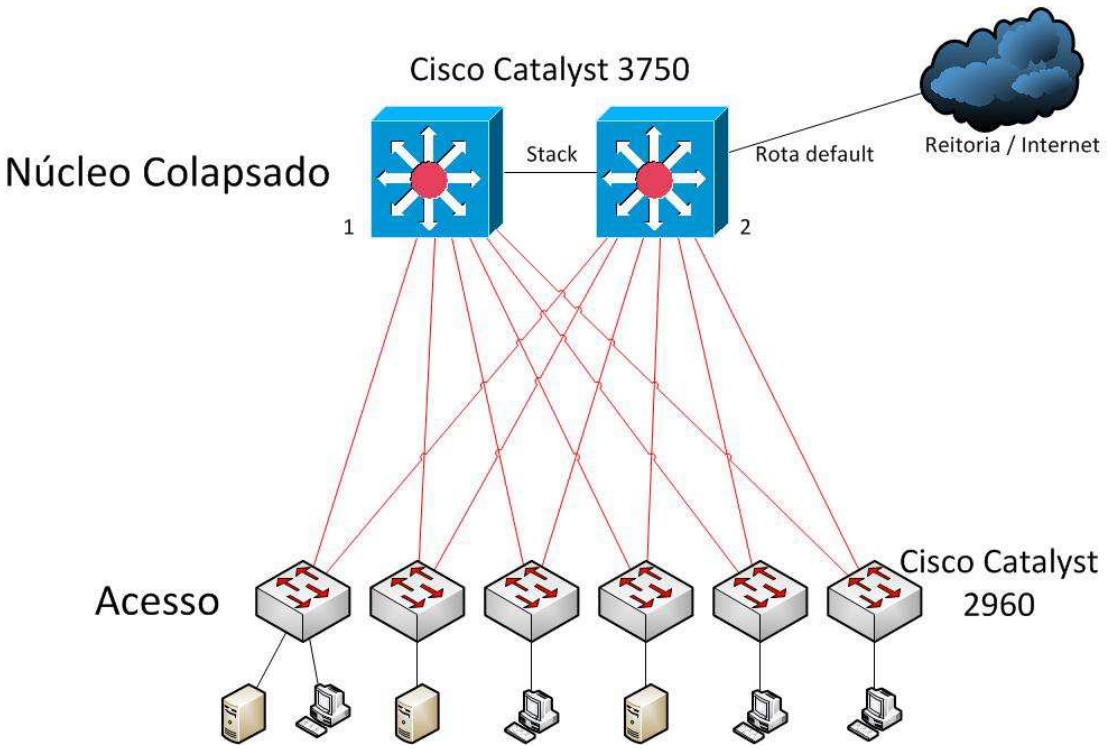
O comando *switchport trunk encapsulation dot1q* define qual será a VLAN nativa, por padrão VLAN 1, em um *trunk*. Quadros pertencentes à VLAN nativa não são encapsulados com a informação de tagging. Assim, se uma estação de trabalho for conectada a uma interface configurada pra tronco ela poderá “entender” somente os quadros da VLAN nativa.

A penúltima linha de configuração escolhe quais VLANs serão permitidas no tronco. Por padrão, um *switch* permite todas as VLANs (1 a 4096) ativas pelo *link*:

- lista-de-vlans é uma lista explícita contendo os números das VLANs, separadas por vírgulas (,) ou traços (-).
- all permite todas as VLANs.
- add lista-de-vlans adicionam às VLANs permitidas a uma lista já configurada.
- remove lista-de-vlans remove da lista VLANs permitidas a uma lista já configurada.
- remove lista-de-vlans permite todas as VLANs, com exceção das listadas no comando.

Na topologia proposta, todas as ligações entre os *switches* serão feitas como troncos. A figura 13 ilustra a topologia proposta com destaque, em vermelho, dos *link*s utilizados como *trunk*.

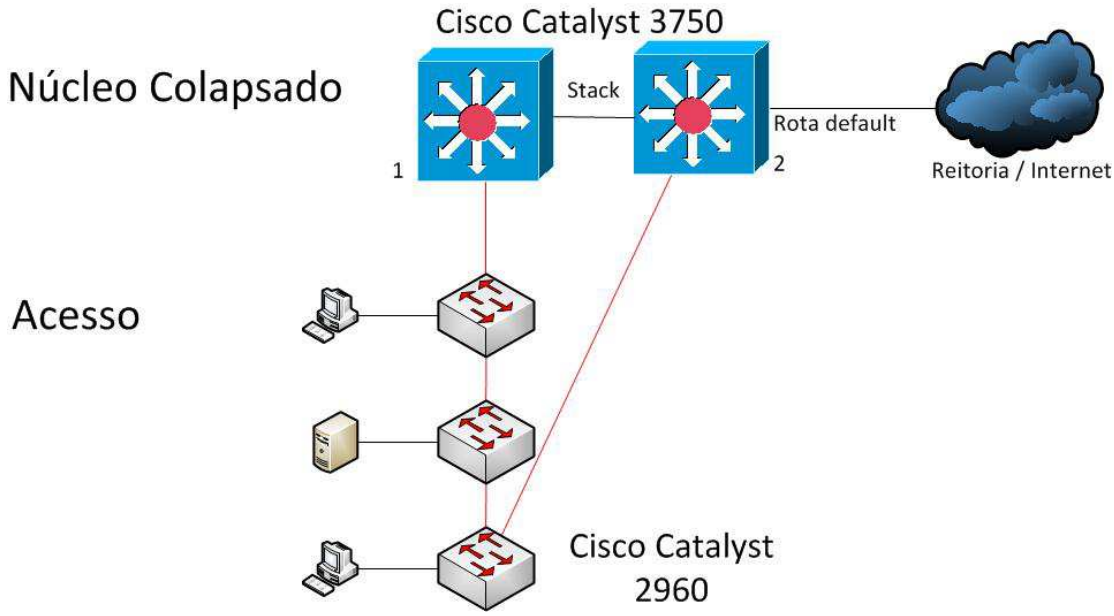
Figura 13 - A nova topologia e os links de tronco



(Autoria própria)

A figura 14 mostra o mesmo paradigma, aplicado ao cascateamento de *switches*.

Figura 14 - Troncos com cascateamento



(Autoria própria)

O Quadro 6 mostra os comandos utilizados para verificação das configurações, bem como sua explicação.[1]

Quadro 6 - Comandos de verificação de troncos (Autoria própria)

Comando	Função
*Switch#***show interface** *tipo módulo/porta* ***switch*port**	Mostra configuração da porta na camada 2, bem qual encapsulamento foi configurado, qual é a VLAN nativa e quais VLANs são permitidas no tronco.
Switch# **show interface** *tipo módulo/porta* **trunk**	Exibe o estado do tronco, bem qual encapsulamento foi configurado, qual é a VLAN nativa e quais VLANs são permitidas no tronco.

3.7 *Spanning Tree Protocol* e RPVST+

Apesar de suportar outras versões de STP, o tipo recomendado pela Cisco para seus equipamentos é o chamado de *Rapid* Per-VLAN *Spanning Tree Protocol Plus*, ou RPVST+. Este tipo de STP alia o padrão 802.1w ou *Rapid Spanning Tree Protocol* à implementação da Cisco que permite que uma instância de Spanning Tree seja usada para cada VLAN, o Per-VLAN *Spanning Tree*.

As vantagens do RSTP é o fato do protocolo possuir um desempenho muito superior ao *Spanning Tree* tradicional. Por exemplo, enquanto o STP comum pode levar ate 50 segundos para reagir e realizar as mudanças de topologia em uma rede, o 802.1w pode obter o mesmo resultado no tempo necessário para 2 *hellos* (6 segundos), isso pode ser obtido graças a duas características: ele trabalha com apenas 3 estados para as portas evitando assim o intervalo necessário para a transição entre alguns estados; uma vez definido quem será a *root bridge* ele permite que as portas em um dado segmento de rede negociem e definam qual será sua função e estado, isso retira da *root bridge* o ônus de ter que organizar toda a topologia sozinha e atrasos resultantes da troca de BPDUs para a definição dos papeis de cada porta.

O Per-VLAN é uma funcionalidade desenvolvida pela Cisco para seus equipamentos que permite a criação de uma instancia de STP para cada VLAN. Esse tipo de habilidade torna possível aos equipamentos realizar tarefas que de outra forma não seriam, como por exemplo, utilização de topologias redundantes sem o risco de *loops* e também configurações que permitam o balanceamento de carga entre *link*s redundantes quando estes *link*s estão em VLANs diferentes.[1]

Por apresentar as vantagens descritas acima, neste projeto seguiremos as recomendações da Cisco e utilizaremos o *Rapid* PVSTP. Para a configuração do RPVST+ nos equipamentos, devemos indicar a estes equipamentos que o *Spanning Tree Protocol* será utilizado e qual tipo. Para isso devemos entrar no modo de configuração global e inserir o comando:

switch# **configure terminal**

switch(config)# **spanning-tree mode rapid-pvst**

No desenho proposto, os *switches* 3750 desempenharão o papel de *root bridge*. A eleição da *root bridge* pode é feita baseada em um cálculo que utilizada o MAC e um valor de prioridade. Para não deixar esta eleição à mercê de chance, é importante definir a *root bridge* manualmente, para isso devemos configurar quem será o *root bridge* para cada VLAN na topologia. No modo de configuração global deve ser usado o seguinte comando para cada VLAN:

switch# **configure terminal**

switch(config)# **spanning-tree vlan** número_*da_vlan* **priority 0**

Existem duas configurações importantes que devem ser executadas em todas as portas de acesso. A primeira define que a porta é uma porta de acesso para o STP, assim quando o usuário desliga ou liga o computador o *switch*, sabendo que se trata de uma porta de acesso, coloca a porta automaticamente no estado de forwarding. Ele também não faz a descarga da Quadro de MACs e não avisa o *root bridge* sobre a alteração no estado da porta. O protocolo RPVST+ define esse tipo de porta como *edge port*. O comando a seguir mostra como devem ser configuradas todas as portas de acesso do usuário: [1]

Switch(config-if)# **spanning-tree portfast**

A outra medida é habilitar a funcionalidade BPDU Guard. Essa funcionalidade bloqueia a porta, colocando-a no estado de *errdisable* caso algum BPDU entre pela mesma; como será configurada nas portas de acesso, não é esperado nenhum BPDU nessas portas. A configuração abaixo habilita o BPDU Guard em todas as portas configuradas como *edge ports*: [1]

Switch(config)# **spanning-tree portfast bpduguard default**

Quando um BPDU for recebido por essa interface, o *switch* irá desabilitar essa porta e mesma deverá ser habilitada manualmente, através da configuração: [1]

Switch(config)# **interface** *tipo módulo/porta*

Switch(config-if)# **no shut**

O Quadro 7 mostra os comandos utilizados para verificação das configurações, bem como sua explicação.[1]

Quadro 7 - Comandos de verificação de STP (Autoria própria)

Comando	Função
Switch# **show spanning-tree**	Mostra os parâmetros do STP para todas as VLANs
Switch# **show spanning-tree inconsistentports**	Exibe as portas consideradas como inconsistentes.

3.8 Agregação de *link*s - *Etherchannel*

Em uma rede típica, como a da UTFPR, vários usuários se conectam a um *switch* de acesso, e esse se conecta a outros *switches* através de um único tronco. Dependendo do tráfego desses usuários, o tronco pode ficar sobrecarregado pois precisa passar o tráfego de vários usuários, conectados a várias interfaces e VLANs através de uma única conexão, criando o chamado “gargalo” na rede. Lembrando que uma rede de camada 2 não pode ter *loops*, mesmo que se conecte outro *link* entre os *switches*, o STP irá bloquear uma das portas, deixando somente um caminho ativo por vez, garantindo nesse caso somente a redundância.

Os equipamentos da Cisco System adquiridos oferecem outro método de escalar a largura de banda agregando *link*s paralelos. Essa tecnologia é conhecida como EtherChannel. Essa tecnologia provê uma maneira fácil de expandir a rede sem a necessidade da compra de um equipamento com a próxima magnitude de vazão de dados. Por exemplo, um *link* FastEthernet que tem a vazão total de 200 Mbps (entrante e sainte) pode ser expandido a até 8 *link*s aumentando a vazão 1600 Mbps; o mesmo

pode ser aplicado a um *link* GigabitEthernet, criando um Gigabit Etherchannel de até 16 Gbps. A questão da redundância continua sendo garantida. Se uma das conexões agregadas falhar o tráfego da mesma é automaticamente transferido para o *link* adjacente, em um processo que leva milissegundos e totalmente transparente para o usuário. Caso mais conexões falhem, o tráfego continua a ser transferido para outros *link*s, assim também caso eles voltem a funcionar o tráfego voltará a ser redistribuído entre eles automaticamente.

Existem 2 protocolos disponíveis para o *EtherChannel*:

- Port Aggregation Protocol (PAgP) – Proprietário Cisco.
- *Link* Aggregation Control Protocol (LACP) – IEEE 802.3ad

Devido ao caráter padronizado, será abordado o protocolo LACP. Esse protocolo define que até 16 portas podem ser agregadas, sendo que somente 8 podem ser utilizados ao mesmo tempo (menor prioridade), deixando o restante em um modo de espera caso algum dos ativos falhe. Para agrupar interfaces é necessário que todas tenham exatamente as mesmas configurações (velocidade, duplex, modo, VLANs e Spanning-tree.

Switch(config**)# interface** *tipo módulo/porta*

Switch(config-if)# **channel-protocol lacp**

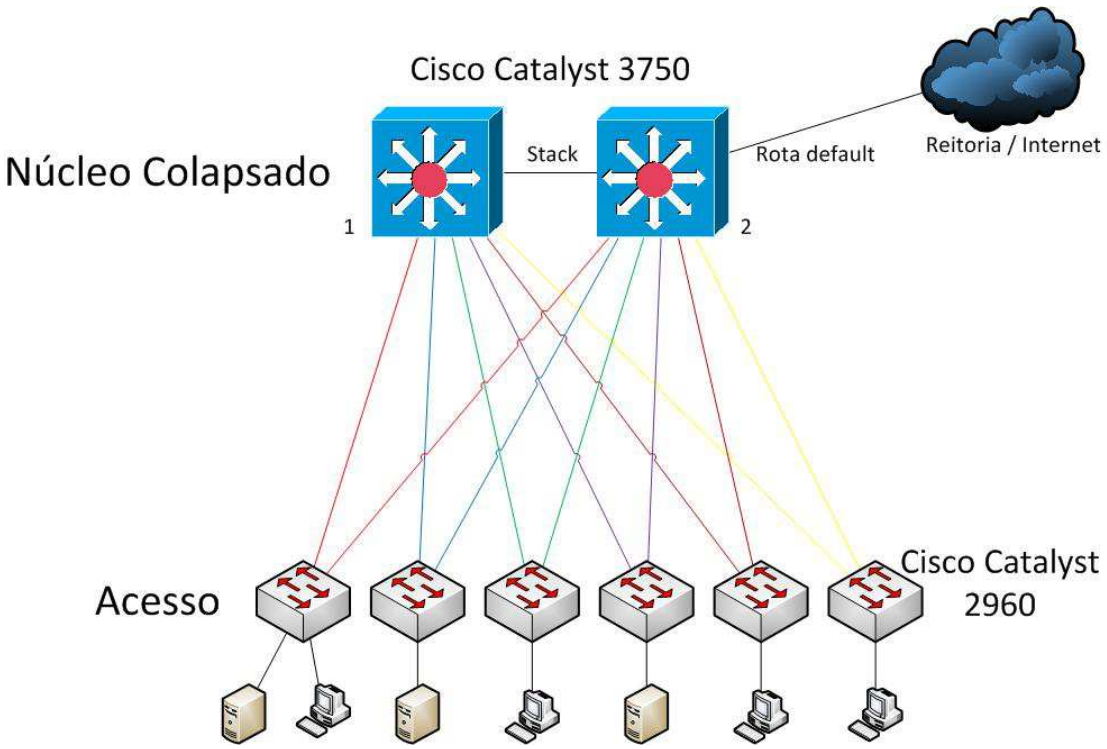
Switch(config-if)# **channel-group** *número-do-grupo* **mode** {**on | passive | active**}

Todas as interfaces inclusas em um grupo devem ter o mesmo identificador único do mesmo, que é o número-do-grupo (que vai de 1 a 64). O modo on define que o canal é incondicional, sobrepujando a negociação do LACP; *passive* fica escutando e espera uma negociação e *active* pergunta ativamente à outra ponta do *link* para negociar o canal.[1]

Na topologia proposta, os *link*s redundantes entre os *switches* de acesso e do núcleo colapsado podem ser configurados como *EtherChannel*, caso necessário. A

figura 15 mostra os canais como 2 *link*s da mesma cor formando um canal. Também dependendo da necessidade, mais *link*s podem ser adicionados posteriormente aos canais.

Figura 15 - A nova topologia com links agregados



(Autoria própria)

O modelo em cascata não contempla a utilização de agrupamento de *link*s, pois somente um up*link* é feito entre cada *switch*.

O Quadro 8 mostra os comandos utilizados para verificação das configurações,

Quadro 8 - Comandos de verificação de EtherChannel (Autoria própria)

Comando	Função
Switch# **show etherchannel summary** *Switch#* **show etherchannel port**	Mostra um sumário dos canais, assim como as portas membro

3.9 Roteamento em *Switch* Multicamada

O roteamento InterVLAN inicialmente requer que o serviço de roteamento seja habilitado. Após isso, pode-se configurar rotas estáticas ou algum protocolo de roteamento. Por padrão, o roteamento está habilitado nos equipamentos, caso seja necessário ele pode ser desabilitado com o comando:[14]

Switch(config)# **no ip routing**

Por padrão, as portas de um *switch*, pelo menos da maioria dos *switches* Catalyst, estão configuradas como interfaces de camada 2, ou seja, não pode ser configurado um endereço IP direto nessa interface e todo o tráfego de camada 3 irá passar pela SVI da VLAN da mesma.

Para configurar uma interface física em camada 3, entre o com o seguinte comando:

Switch(config)# **interface** *tipo módulo/porta*

Switch(config-if**)# no *switch*port**

Switch(config-if**)# ip address** *endereço_ip máscara_de_subrede*

Switch(config-if**)# no shutdown**

A palavra chave nessa configuração é *switch*port. Sempre que aparecer a essa palavra, pense em camada 2. Na configuração acima, a palavra *no* é entrada antes, negando a condição. Assim como junto do comando shutdown ele nega o estado de fechado da porta, em outras palavras, habilita a mesma.[1]

De acordo com a COGETI, as VLANs não devem se comunicar entre si, somente com o servidor DHCP e o *Default Gateway*. Para tanto, algumas configurações extras devem ser adicionadas. O roteamento para a rota *default* é

realizado através da configuração de uma única rota estática no *stack* formado pelos *switches* do núcleo colapsado:

Switch(config**)# ip route 0.0.0.0 0.0.0.0** *ip_do_default_gateway*

O IP 172.17.50.198 é o *default gateway* da rede da UTFPR. Esse comando define a rede 0.0.0.0 no primeiro bloco com a máscara 0.0.0.0 no segundo, o que enquadra todas os destinos possíveis nessa rota, lembrando que a decisão é feita sempre da rota mais para a menos específica.[15][16][17]

O roteamento InterVLAN é feito automaticamente caso seja configurado uma SVI na mesma, o que é necessário para rotear os pacotes para a rota *default*. Pelo mesmo motivo o roteamento IP não pode ser desabilitado. Para evitar que uma VLAN se comunique com outra, serão utilizadas então Listas de Acesso, em inglês *Access Control List* ou ACL, para filtrar o tráfego entre as VLANs. Para configurar uma ACL, primeiramente devem-se colocar as cláusulas da mesma:

Switch(config)# **access-list** *número_da_lista* **deny ip any** *ip_da_rede_destino máscara_coringa*

Os parâmetros da configuração definem:

- número_da_lista identifica a lista e o tipo da mesma. As listas de 1 a 99 são chamadas de ACL Padrão, são mais antigas e definem somente o destino. As listas de 101 a 199 são chamadas de ACLs Estendidas, que analisam a origem, destino e protocolo. Será utilizada somente a ACL Estendida.
- *deny* define que a lista irá bloquear, ou filtrar, caso o pacote corresponda às cláusulas propostas.
- IP define qual o protocolo será avaliado; no caso, todos os pacotes IP.
- *any* define qualquer endereço como origem.
- ip_da_rede_destino máscara_coringa definem o destino, que pode ir de um único *host* até uma rede inteira.

A máscara “coringa” representa um diferencial na configuração. Nela é feita uma comparação de quais *bits* do endereço serão analisados, sendo que 0 define que o *bit*

correspondente do endereço é avaliado e 1 não. A máscara “coringa” pode ser comparada como um inverso da máscara de rede. O parâmetro *any* então pode também ser escrito como 0.0.0.0 255.255.255.255.

Para permitir o tráfego para a Internet, devemos adicionar mais uma linha no final, e somente nele, da ACL. O motivo para tanto é que uma ACL é avaliada linha-a-linha, da primeira para a última cláusula entrada, sendo que em caso de uma correspondência seja feita, a ação de permitir ou negar é tomada automaticamente e nenhuma outra cláusula é avaliada, tornando assim a ordem das cláusulas importantes. O fator que exige a configuração extra é uma cláusula implícita no fim de todas as ACLs que nega todo o tráfego de qualquer origem para qualquer destino, ou seja, uma cláusula *deny any any*. Sendo assim iremos negar o tráfego desejado entre as VLANs com a configuração anterior e permitir todo o tráfego para outras redes, ou seja, para a Internet. Para permitir esse tráfego, execute a seguinte configuração:[18][10][19]

Switch(config)# **access-list** *número_da_lista* **permit ip any any**

Por fim, a lista de acesso deve ser aplicada a uma interface e definida se a análise será feita quando um pacote entra ou sai pela mesma. Quando um pacote é roteado por um *router*, ele passa “através” do mesmo, pois entra em uma interface, analisado o destino e as rotas conhecidas e encaminhado para a interface de saída. Em um *switch* multicamada o mesmo processo ocorre, porém uma interface de acesso em camada 2 não possui seu próprio endereço IP, e sim uma interface lógica descrita anteriormente como SVI. O pacote, portanto, não passa somente através do *switch*, mas sim através da SVI.

Para a necessidade da UTFPR, a ACL deverá ser aplicada para o tráfego entrante em cada SVI de cada VLAN nos *switches* do núcleo colapsado, assim evita-se que a análise seja feita somente quando o tráfego estiver voltando, o que criaria tráfego desnecessário na rede. Para executar essa configuração, siga os seguintes comandos:[18][10][19]

Switch(config)# **interface vlan** *id_da_vlan*

Switch(config-if)# **ip access-group** *número_da_acl* **in**

Como mencionado anteriormente, o parâmetro in define que que a ACL será avaliada quando pacotes estiverem entrando na SVI. Para a VLAN de administração, não será aplicada nenhuma ACL, pois ela necessita de acesso total à rede.

Outro fator importante é permitir e direcionar requisições de DHCP para o servidor. Os *switches* do núcleo colapsado têm a capacidade de fornecer esse serviço[1], porém não será implementada essa solução devido ao DHCP ser um protocolo da camada de aplicação. Sendo assim, para deixar a solução mais limpa e elegante, a COGETI definiu que o servidor DHCP será feito em um máquina separada, pertencente à VLAN administrativa.

Para permitir o tráfego para o servidor DHCP, a seguinte linha deverá ser inclusa na ACL:

Switch(config)# **access-list** *número_da_lista* **permit ip any host** *ip_servidor_DHCP*

O parâmetro *host* define que somente um *host*, que é definido pelo endereço logo após esse parâmetro, será avaliado. Esse parâmetro é idêntico à máscara “coringa” 0.0.0.0. Lembrando que a ordem das cláusulas faz diferença, essa cláusula deverá entrar antes da negação da rede das outras VLANs, considerando que o DHCP estará em uma VLAN negada posteriormente.

O segundo e último ponto é o redirecionamento das mensagens DHCP para o servidor. Isso é realizado através da seguinte configuração na SVI:[1]

Switch(config)# **interface vlan** *vlan_id*

Switch(config-if)# **ip helper-address** *ip servidor_DHCP*

Essa configuração deverá ser efetuada em todas as SVIs de todas as VLANs nos *switches* do núcleo colapsado, relembrando que elas serão o default gateway de todos os usuários da VLAN. Juntamente com o redirecionamento, o equipamento que o faz adiciona o parâmetro GIADDR, definido na RFC 2131 pela IETF.[20] Nele, o agente

que realiza o redirecionamento adiciona o endereço IP de sua interface, no caso a SVI. Assim o servidor pode escolher de qual grupo ele irá disponibilizar um endereço IP para o solicitante. Assim um único servidor DHCP pode fornecer esse serviço para toda a rede.

O Quadro 9 mostra os comandos utilizados para verificação das configurações, bem como sua explicação.[1]

Quadro 9 - Comandos de verificação de roteamento, ACLs e IPs (Autoria própria)

Comando	Função
Switch# **show ip route**	Mostra a Quadro de roteamento, assim como a rota padrão.
Switch# **show ip interface brief**	Exibe todas as interfaces, estado e endereço IP das mesmas.
Switch# **show access-lists**	Exibe as listas de acesso configuradas, seu número ou nome e suas cláusulas.
Switch# **show ip interface** *tipo módulo/porta* *Switch#* **show ip interface vlan** *vlan_id*	Mostra quais ACLs estão aplicadas para tráfego entrante e sainte.

3.10 DHCP Clandestino

De acordo com a COGETI, um dos problemas mais comuns na rede é a adição de servidores DHCP clandestinos. Em outras palavras, usuários trazem *routers* próprios e ligam na rede com o serviço de servidor DHCP habilitado; quase sempre não querendo degradar a rede propositalmente, mas por falta de conhecimento. Isso causa um enorme problema, pois as requisições de IP são respondidas pelo servidor clandestino, com outra faixa de endereços.

Para resolver tal problema, será implantado uma funcionalidade de segurança chamada DHCP *Snooping*. Quando habilitada, as portas do *switch* são categorizadas como “confiáveis” ou “não-confiáveis”. Servidores legítimos podem ser conectados às

portas confiáveis, enquanto todos os outros usuários ficam por trás das portas não-confiáveis.

O *switch* intercepta todos as solicitações de DHCP oriundas de portas não confiáveis antes de enviar por toda a VLAN. Todas as respostas que entram em portas não-confiáveis são descartadas, pois provavelmente foram provavelmente geradas por servidores clandestinos. A porta ofensora então é derrubada no estado de *errdisable* e deve ser manualmente “erguida”. Para configurar o DHCP *Snooping*, é necessário executar a seguinte configuração:[1]

Switch(config)# **ip dhcp snooping**

Em seguida, é necessário identificar em quais VLANs ele será implementado, com o comando:

Switch(config)# **ip dhcp snooping vlan** *número_da_vlan*

Por padrão, todas as portas são consideradas não confiáveis. É necessário configurar a porta na qual o servidor DHCP está ligado como confiável, através da configuração a seguir:

Switch(config)# **interface** *tipo módulo/porta*

Switch(config-if)# **ip dhcp snooping trust**

Por ultimo, existe uma configuração que é habilitada por padrão, chamada de opção-82, que adiciona detalhes que serão comparados pelo *switch*. Essa opção é descrita em detalhes na RFC 3046.[1] Para deixar o DHCP *Snooping* com um caráter mais permissivo, como solicitado pela COGETI, essa função deverá ser desabilitada com o comando a seguir:

Switch(config)# **no ip dhcp snooping information option**

O Quadro 10 mostra os comandos utilizados para verificação das configurações, bem como sua explicação.[1]

Quadro 10 - Comandos de verificação de DHCP *snooping* (Autoria própria)

Comando	Função
Switch# **show ip dhcp snooping**	Exibe o estado do DHCP snooping, para quais VLANs está habilitado e quais interfaces.

3.11 Stacking

Switches da Cisco possuem a funcionalidade de operarem em modo *Stack* ou empilhados. Essa pilha pode ser definida como um conjunto de 1 a 9 *switches* conectados por suas portas *StackWise*. Em uma pilha um dos *switches* irá funcionar como o *Stack Master*, funcionando como um ponto centralizado para gerenciamento da pilha, mas todos são *stack members* e todo membro é elegível para se tornar *Stack Master*. Os *stack members* usam a tecnologia *StackWise* para trabalhar junto, agregar recursos e se comportar como uma única entidade para o restante da rede.

O *switch* opera em *stack* de forma nativa, isso pode ser entendido como se um único *switch* funcionasse como uma pilha de um único membro, onde este membro atua como *stack master*. Porém, ao conectá-lo a outro *switch* através de sua porta

StackWise e o processo de eleição tem início. Uma vez eleito o *Stack Master*, suas informações como *Bridge* ID e MAC serão usadas para identificar a pilha. Essa eleição pode ser manipulada, podemos alterar a prioridade de um dos equipamentos para garantir que este se torne o *Stack Master*.[27][28] A prioridade pode variar de 1 a 15, sendo 15 a maior prioridade. Isso pode ser feito através do comando:

Stack(config)# ***switch*** *1* **priority** *15*

Este comando ira aumentar a prioridade do *switch* com ID igual a 1 para o valor 15.

Outra configuração que pode ser feita é a alteração do valor usado para identificar cada *switch* no *stack*. Inicialmente cada equipamento ganha um valor único para identificá-lo dentro do *stack* uma vez que se une a ela, este valor pode variar de 1 a 9, sendo que o menor valor disponível é sempre escolhido. Mas pode ser que por algum fator, como por exemplo, um conflito gerado por números iguais, o administrador deseje alterar este valor para um dos equipamentos. Tomando o cuidado para que não exista na pilha outro equipamento com o mesmo identificador, é possível alterá-lo valor de um dispositivo com o comando:

Switch(config)# ***switch*** *2* **renumber** *1*

Este comando irá alterar o ID do *switch* 2 para o valor 1.

O Quadro 11 mostra os comandos utilizados para verificação das configurações, bem como sua explicação.[1]

Quadro 11 - Comandos de verificação de *stacking* (Autoria própria)

Comando	Função
Switch# **show *switch***	Exibe o estado dos *switches* conectados à pilha.

3.12 Gerenciamento – SNMP, Syslog, Console e Acesso Remoto

Os métodos que serão utilizados para monitorar e gerenciar os equipamentos da rede são *Syslog* e SNMP. Ambos os métodos de monitoramento tem a função de registrar e alertar a respeito de eventos que ocorreram na rede e nos equipamentos, mas possuem enfoques distintos, enquanto o *Syslog* pode fornecer informações mais especificas sobre o equipamento e eventos ocorridos, o SNMP permite maior flexibilidade na escolha dos eventos monitorados entre outras funcionalidades.[1]

O *Syslog* é um recurso presente em *switches* Cisco que nos fornece uma forma de auditar eventos ocorridos no equipamento através de mensagens descrevendo-os. Estas mensagens possuem um formato definido:

- *TimeStamp*: horário em que o evento ocorreu, o valor padrão usado é o período decorrido desde que o *switch* foi ligado (*uptime*);
- *Facility Code*: código que identifica o recurso do sistema que gerou a mensagem;
- Severidade: valor que varia entre 0 e 7 indicando a severidade da mensagem, sendo que quanto menor o valor, mais severo é o evento;
- Mnemônico: uma mensagem que identifica o evento dentro de seu *facility code*;
- Mensagem de Texto: basicamente uma mensagem de texto que descreve de maneira mais clara o evento;

Abaixo segue alguns exemplos de mensagens.

00:00:46: %*LINK*-3-UPDOWN: Interface Port-channel1, changed state to up

00:00:47: %*LINK*-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up

00:00:47: %*LINK*-3-UPDOWN: Interface GigabitEthernet0/2, changed state to up

Para utilizar o *Syslog* devemos realizar algumas configurações nos equipamentos. A primeira diz respeito às *timestamp*, este deve ser alterado, pois o valor padrão pode tornar o diagnostico de problemas um pouco desafiador. O valor padrão utilizado para as mensagens geradas indicada o *uptime* do equipamento, esse valor não é recomendado, pois ao consultar as mensagens de *Syslog* geradas por diferentes equipamentos para determinar um problema ocorrido em toda rede, teremos que lidar com cálculos para correlacionar os horários dos equipamentos, processo que adiciona um nível de complexidade ao processo de diagnostico sem trazer beneficio algum. Sendo assim, a boa prática nos recomenda configurar o horário local para que todos os equipamentos possuam um valor unificado para seu *timestamp*, tornando assim mais fácil identificar fenômenos que desencadearam eventos em mais de um equipamento.

O primeiro passo nessa configuração é corrigir o horário local do equipamento, isso pode ser feito manualmente indicando a data e horário, mas como estamos tratando de vários equipamentos faz mais sentido utilizar um servidor de NTP:

Switch(config)# **ntp server** *a.ntp.br*

O próximo passo é configurar o local onde estas mensagens serão exibidas ou armazenadas. As opções que temos são exibí-las na console administrativa do equipamento, caso esta esteja disponível, armazenar em *buffer*, local que garante um volume padrão de 50 mensagens, embora possa ser configurado para armazenar um número maior de mensagens, e por fim o uso de um servidor externo para armazená-las. No caso iremos utilizar um servidor externo, pelo fato de não possuir as limitações dos demais métodos.

Switch(config)# **logging** *endereço_do_servidor*

O nível de severidade também deve ser definido, levando em conta o fato que se somente monitorarmos as mensagens críticas podemos ignorar sintomas que podem vir a ser tornar problemas, mas por outro lado se utilizarmos um nível de severidade muito baixo o volume de mensagens geradas será extremamente difícil de gerenciar e um tanto quanto inútil.

Switch(config)# **logging trap warnings**

O *Simple Network Management Protocol* permite que equipamentos de rede compartilhem informações a respeito de si mesmo e de suas atividade. Um sistema SNMP completo consiste de um Gerente SNMP, um sistema centralizador das informações responsável por coletar e armazenar informações, e de agentes SNMP, sistemas que executam nos dispositivos monitorados responsável por coletar informações e responder solicitações do gerente.

Os *switches* Cisco têm como procedimento padrão coletar informações sobre si mesmo e suas operações e armazená-las em uma base de dados estruturada hierarquicamente em forma de árvore chamada de *Management Information Base* (MIB). Cada informação coletada pode ser encontrada nessa base através de um identificador único denominado *Object Identifier* (OID), que pode ser considerado o nó da arvore que define esta informação. A maneira que o Gerente SNMP obtém informações de seus agentes pode ser basicamente de duas formas: através de requisições ou através de *traps*. A primeira forma geralmente é feita sob demanda quando um valor precisa ser verificado ou também pode ser feita de periodicamente pelo Gerente. Na segunda forma, uma regra é configurada no agente e caso um evento em particular ocorra o uma mensagem é enviada ao servidor.

A configuração que deverá ser feita nos equipamentos para habilitar os agentes deverá conter basicamente duas informações: a comunidade que irá ter acesso às informações SNMP e o endereço do servidor que irá receber as mensagens.

Switch(config)# **snmp-server community** *endereço_da_comunidade*

Switch(config)# **snmp-server host** *endereço_do_servidor*

É extremamente importante limitar o acesso aos equipamentos somente às pessoas que realmente tenham permissão. Para isto, é necessário adicionar senhas de

acesso para cada usuário remoto e para o acesso via console. Não somente adicionar senhas, também é necessário atentar para a segurança da mesma. Senhas são escritas na configuração como texto comum, onde qualquer pessoa que tenha acesso poderá vê-la. O exemplo a seguir mostra como adicionar um usuário e uma senha criptografada:[25]

Switch# **configure terminal**

Switch(config)# **username** *usuário* **secret** *senha*

Após criar os usuários, é necessário definir que o acesso somente será permitido via *login*, ao acesso remoto (vty), siga o exemplo a seguir:

Switch (config)#**line vty** *0 4*

Switch (config-line)# **login local**

Os números após a palavra vty definem quantas conexões serão permitidas, no exemplo, 5 (de 0 a 4).

A configuração para acesso via console é exemplificada a seguir:

Switch (config)#**line con 0**

Switch (config-line)# **password** *senha*

Switch (config-line)#**login**

Para garantir que o equipamento criptografe todas as senhas, mesmo as que não foram introduzidas como secretas, utilize o comando de configuração global ***service password-encryption*.**

Os equipamentos Cisco também dispõem de 2 níveis de acesso, que permitem ou negam funções, dependendo do nível acessado. Para realizar alterações na configuração, é necessário entrar no modo privilegiado, através do comando ***enable***. Para proteger o acesso a esse modo, entre com o comando a seguir:[26]

Switch# **enable secret** *senha*

O Quadro 12 mostra os comandos utilizados para verificação das configurações, bem como sua explicação.[29]

Quadro 12 - Comandos de verificação da configuração, timestamp e SNMP (Autoria própria)

Comando	Função
Switch# **show running-config**	Exibe a configuração atual do equipamento.
Switch# **show clock**	Exibe a hora (timestamp) do equipamento
Switch# **show snmp**	Monitora o estado do SNMP
Switch# **show snmp groups**	Exibe informações sobre os grupos configurados

5 CONCLUSÃO

O objetivo deste trabalho foi elaborar um projeto reestruturação da rede do Cämpus Curitiba da UTFPR utilizando para isso os equipamentos Cisco adquiridos pela Instituição.

A maneira como a estrutura de rede do Cämpus Curitiba esta organizada atualmente não é a recomendada para uma rede das suas dimensões. A utilização de uma rede plana fornece condições ideais para que problemas proliferem causando indisponibilidade de serviços. Problemas que podem ser citados que são crônicos na instituição é o de tempestades de broadcasts, que surgem quando loops são fechados entre os equipamentos de redes, problema cuja gravidade pode variar desde perda de desempenho até indisponibilidade de serviços de rede. Outra enfermidade do ambiente são os chamados DHCPs clandestinos, geralmente surgem quando usuários desavisados introduzem equipamentos de rede (como *access points*) na estrutura atual, que funcionam como servidores DHCP e conflitam com o servidor original tornando a rede instável. A falta de escalabilidade do modelo atual também é preocupante, pois impossibilita a adição de novos equipamentos e recursos impedindo que a rede se adapte às mudanças impostas pela evolução tecnológica.

Uma revisão da estrutura atual foi realizada, e ao invés de manter o modelo plano que a instituição utiliza hoje, uma nova abordagem foi seguida fazendo uso do modelo hierárquico da Cisco que trabalha com 3 camadas para definir as atribuições dos equipamentos de camada 2 e 3. Graças aos novos equipamentos podemos sanar muitos dos problemas crônicos que a estrutura atual possui. A segmentação da rede através do uso de VLANs, onde cada departamento representa uma única fatia do todo, é importante por evitar que problemas surgidos em um segmento se espalhe para os demais isolando-o e outra vantagem ainda é que redes menores tornam o diagnostico de problemas mais dinâmico. A implantação do protocolo STP é importante por evitar o surgimento de tempestades de *broadcast* e, através da implementação da

Cisco, isso ainda pode ser feito com um desempenho superior ao normal, pois é utilizado o padrão 802.1w que ajuda a diminuir o tempo de convergência da rede drasticamente. O uso de *stack* e *Etherchannel* é também algo novo, uma forma de agregar recursos e simplificar o gerenciamento destes.

Um assunto que não fez parte do escopo deste estudo, mas é algo interessante e pode enriquecer ainda mais a estrutura da rede, é a questão da segurança, por possuir uma saída direta com a Internet alguns cuidados são fundamentais, como por exemplo restrição de acesso. Um dos equipamentos que também foi adquirido pela COGETI foi o Cisco ASA - *Adaptive Security Appliances*, uma ferramenta de segurança que operando como um *gateway* oferece uma série de recursos como análise de comportamento e detecção de intrusão, *firewall* e NAT e ainda serviços de VPN. A incorporação desta ferramenta atrelada à nova estrutura da rede pode tornar esta ainda mais robusta e segura.

A estrutura proposta neste trabalho recomenda o uso de duas camadas funcionais hierárquicas, compostas de um núcleo colapsado, contendo as camadas de núcleo e distribuição, e a camada de acesso. Esta estrutura, futuramente, também pode ser otimizada separando as duas camadas que fazem parte deste núcleo colapsado, permitindo que cada camada seja responsável por suas funções, mediante a aquisição de equipamentos. Outro ponto que pode ser abordado é a reestruturação física, de camada 1, referente ao cabeamento estruturado.

Algo que deve ser citado é que apesar do projeto ter sido realizado visando a utilização dos equipamentos adquiridos pela COGETI, ele não pode ser implementado pelo fato dos equipamentos não estarem disponíveis até o momento da conclusão deste trabalho, o que nada impede o uso deste projeto futuramente.

REFERÊNCIAS

1. HUCABY, David. **CCNP *SWITCH* 642-813 Official Certification Guide**. Estados Unidos da América: Cisco Press, 2010.

2. CISCO SKILLS. **The Middle of Layer Two Redundancy.** 2011. Disponível em: <http://ciscoskills.net/2011/09/07/the-middle-of-layer-two-redundancy/>. Acesso em 20 abr. 2012.

3. CISCO. **Configuring EtherChannels.** Catalyst 3750-X and 3560-X *Switch* Software Configuration Guide, Release 12.2(55)SE. Disponível em: <http://www.cisco.com/en/US/docs/*switches*/lan/catalyst3750x_3560x/software/release/12.2_55_se/configuration/guide/swethchl.html#wpxref31717>. Acesso em 20 abr. 2012.

4. D., Rachel. **DHCP Snooping.** The Cisco Learning Network. 2011. Disponível em: <https://learningnetwork.cisco.com/docs/DOC-2314>. Acesso em 21 Abr. 2012.

5. KNOWLEDGE TRANSFER. **Fibre channel cascade topology.** 2009. Disponível em <http://www.knowledgetransfer.net/dictionary/Storage/en/Fibre_Channel_topology_cascade.htm>. Acesso em 21 Abr. 2012.

6. DICKENS, Christopher. **LAN Party How To - Part 2:** Building the LAN. Tom's Guide tech for real life. 2005. Disponível em: <http://www.tomsguide.com/us/lan-party-how-to,review-496-3.html>. Acesso em 25 Abr. 2012.

7. CISCO. **Cisco StackWise and StackWise Plus Technology.** White Paper. 2010. Disponível em <http://www.cisco.com/en/US/prod/collateral/*switches*/ps5718/ps5023/prod_white_paper09186a00801b096a.pdf>. Acesso em 29 Abr. 2012.

8. CISCO. **Connecting a Terminal to the Console Port on Catalyst *Switches*.** Cisco Catalyst 6000 Series *Switches*. 2007. Disponível em: <http://www.cisco.com/en/US/products/hw/*switches*/ps700/products_tech_note09186a008010ff7a.shtml>. Acesso em 29 Abr. 2012.

9. CISCO. **Using Cisco IOS Software for Release 12.3.** About Cisco IOS Software Documentation. Disponível em: <http://www.cisco.com/en/US/docs/ios/12_3/featlist/gusing.html>. Acesso em 4 Maio 2012

10. CISCO. **Configuring IP Access Lists.** Cisco IOS Firewall. 2007. Disponível em: <http://www.cisco.com/en/US/products/sw/secursw/ps1018/products_tech_note09186a00800a5b9a.shtml>. Acesso em 05 Maio 2012.

11. CISCO. **Cisco Catalyst 2960-S and 2960 Séries *Switches* with LAN Base Software.** Cisco Catalyst 2960 Séries *Switches*. Disponível em: <http://www.cisco.com/en/US/prod/collateral/*switches*/ps5718/ps6406/product_data_sheet0900aecd80322c0c.html>. Acesso em 11 Maio 2012.

12. CISCO. **Cisco Catalyst 3750 Data Sheet.** Cisco Catalyst 3750 Séries *Switches*. Disponível em: <http://www.cisco.com/en/US/prod/collateral/*switches*/ps5718/ps5023/product_data_sheet0900aecd80371991.html>. Acesso em 12 Maio 2012.

13. CISCO. ***Switches* Cisco Catalyst 2960, 2960-C e 2960-S Séries.** Disponível em: <http://www.cisco.com/web/BR/solucoes/commercial/products/routers_*switches*/catalyst_2960_séries_*switches*/index.html#~models>. Acesso em 12 Maio 2012.

14. CISCO. **IP Addressing Commands.** Cisco IOS IP and IP Routing Command Reference, Release 12.1. Disponível em: <http://www.cisco.com/en/US/docs/ios/12_1/iproute/command/reference/1rdipadr.html#wp1020435 >. Acessado em 15 Maio 2012.

15. CISCO. **Configuring a Gateway of Last Resort Using IP Commands.** IP Routing. 2005. Disponível em: <http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080094374.shtml#route0.0>. Acesso em 16 Maio 2012.

16. CISCO. **Specifying a Next Hop IP Address for Static Routes.** IP Routing. 2006. Disponível em: <http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a00800ef7b2.shtml>. Acessado em 16 Maio 2012.

17. ADRIE; ANN. **IP Routing.** The Cisco Learning Network. 2011. Disponível em: <https://learningnetwork.cisco.com/docs/DOC-1318>. Acesso em 16 Maio 2012.

18. CISCO. **Access Control Lists: Overview and Guidelines**. Security Configuration Guide. Disponível em <http://www.cisco.com/en/US/docs/ios/11_3/security/configuration/guide/scacls.html> . Acesso em 16 Maio 2012.

19. STEWART, Paul. **Introduction to Extended IP Access Lists Application.** The Cisco Learning Network. 2010. Disponível em <https://learningnetwork.cisco.com/docs/DOC-7514>. Acesso em 16 Maio 2012.

20. DROMS, R. **Dynamic Host Configuration Protocol.** IETF Network Working Group. 1997. Disponível em: <http://www.ietf.org/rfc/rfc2131.txt>. Acesso em 16 Maio 2012.

21. UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ. **Organograma dos Campi.** 2011. Disponível em: <http://www.utfpr.edu.br/a-instituicao/documentos-institucionais/organograma-dos-campi-abril-2011/>. Acesso em 19 Maio 2012.

22. UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ. **Departamentos Acadêmicos.** 2012. Disponível em: <http://www.utfpr.edu.br/curitiba/estrutura-universitaria/diretorias/dirgrad/segea/departamentos-academicos/?searchterm=departamentos>. Acesso em 19 Maio 2012.

23. REKHTER, Y. et al. **Address Allocation for Private Internets.** IETF Network Working Group. 1996. Disponível em: <http://tools.ietf.org/html/rfc1918>. Acesso em 20 Maio 2012.

24. CISCO. **Configuring System Information on Catalyst *Switches***. Cisco Catalyst 6000 Séries *Switches*. 2005. Disponível em: <http://www.cisco.com/en/US/products/hw/*switches*/ps700/products_tech_note09186a00801aecbb.shtml>. Acesso em 22 Maio 2012.

25. CISCO. **Enhanced Password Security - Phase I.** Cisco IOS Software Releases 12.1 E. Disponível em: <http://www.cisco.com/en/US/docs/ios/12_1/12_1e8/feature/guide/8e_md5.html>. Acesso em 25 Maio 2012.

26. LIU, Stephen. **Cisco IOS Command Line Interface Tutorial.** Cisco Systems Small/Medium Business Solutions. 1997. Disponível em: <http://www.cisco.com/warp/cpropub/45/tutorial.htm>. Acesso em 27 Maio 2012.

27. CISCO. **Troubleshooting *Switch* Stacks.** Cisco Catalyst 3750 Series *Switches*. Disponível em: <http://www.cisco.com/en/US/docs/*switches*/lan/catalyst3750/software/troubleshooting/*switch*_stacks.html>. Acesso em 27 Maio 2012.

28. CISCO. **Managing *Switch* Stacks**. Catalyst 3750 *Switch* Software Configuration Guide, 12.1(14)EA1. Disponível em: <http://www.cisco.com/en/US/docs/*switches*/lan/catalyst3750/software/release/12.1_14_ea1/configuration/guide/swstack.html>. Acesso em 27 Maio 2012.

29. CISCO. **Configuring SNMP Support.** Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2. Disponível em: <http://www.cisco.com/en/US/docs/ios/12_2/configfun/configuration/guide/fcf014.html>. Acesso em 29 Maio 2012.

30. MURHAMMER, Martin W. et al. **IP Network Design Guide**. Disponível em: http://www.redbooks.ibm.com/redbooks/pdfs/sg242580.pdf Acesso em: 29 maio 2012.

31. LAMMLE, Todd. **Cisco Certified Network Associate Study Guide**. Estados Unidos da América: Sybex, 2000.

32. CISCO. **Interface Configuration Overview.** Cisco IOS Interface Configuration Guide, Release 12.1. Disponível em: <http://www.cisco.com/en/US/docs/ios/12_1/interface/configuration/guide/icdoverv.html#wp1012725>. Acesso em 30 maio 2012.

32. CISCO. **Connection and System Banner Commands**. Configuration Fundamentals Command Reference. Disponível em: <http://www.cisco.com/en/US/docs/ios/11_3/configfun/command/reference/frconban.html>. Acesso em 30 maio 2012.

33. VSTRABELLO. **Os modelos OSI e TCP/IP.** 2010. Disponível em: <http://vstrabello.blogspot.com.br/2010/10/os-modelos-osi-e-tcpip.html>. Acesso em 13 jun. 2012.

34. NASCIMENTO, M. B.; TAVARES, A. C. **Roteadores e *Switches:*** Guia para Certificação CCNA e CCENT Exames. Rio de Janeiro, RJ: Ciência Moderna, 2012.

4 APÊNDICE

4.1 *Hostnames*

Os equipamentos Cisco podem ter os nomes do sistema alterados para diferenciação dos mesmos.[24] Neste trabalho, será definido um padrão de *hostname* de acordo com sua localização física, para fácil reconhecimento do mesmo, da seguinte maneira (as cores são somente para demonstração neste trabalho):

departamentobloconúmero

O Quadro 13 mostra os blocos e departamentos identificados na topologia física enviada pela COGETI:

Quadro 13 - Blocos e departamentos para *hostname*(Autoria própria)

Bloco	Departamentos
H	AINFO
I	VIDEO / BLOCO01
E	DACEX
N	DAQBI / DAFIS / EAD
F	DACOC
C	CGR / DADIN
L	DIBIB / PPGEM
Q	HOTEL
A	DAMEC
B	B003

Assim, 3 hipotéticos *switches* localizados no AINFO teriam como *hostname*:

AINFOH01, AINFOH02 e AINFOH03.

Os *switches* do núcleo colapsado terão uma diferenciação no *hostname*, como são equipamentos especiais, será somente declarado como núcleo, seguido do número

do mesmo, no caso atual “01” como terá somente um. O *hostname,* portanto, ficará NUCLEO01.

Para configurar um *hostname*, siga o exemplo de configuração a seguir:

Switch(config)#**hostname** *nome_do_switch.*

4.2 Descrições de interface

O administrador pode adicionar uma descrição no formato de texto a cada interface do equipamento. Essa descrição tem a única função de identificar para o que a interface é utilizada e/ou onde ela está conectada. Para adicionar uma descrição à interface, veja o exemplo a seguir[31]:

Switch(config)# **interface** *tipo módulo/porta*

Switch(config-if)# **description** *string*

4.3 *Banners*

Outra ferramenta utilizada pelo administrador são mensagens informativas que serão exibidas aos usuários que se conectarem a interface de administração dos equipamentos, também chamadas de *banners*. Existem vários tipos de *banners* que são exibidos em momentos diferentes. Este trabalho aborda os 2 tipos mais comuns. São eles[31]:

- *Banner Message-of-the-Day* (MOTD): Este tipo de banner é exibido a todos os terminais conectados.

- *Banner* EXEC: É exibido sempre que um usuário conectado ao equipamento efetuar o loigin com sucesso.

A configuração dos banners é exemplificada a seguir:

Switch(config)# **banner motd** @

Mensagem @

Switch(config)# **banner exec** @

Mensagem @

As os caracteres especiais antes e depois da mensagem são utilizados pelo sistema como argumentos delimitadores de início e fim do banner. Nos exemplos anteriores foi utilizado o @.

O próximo exemplo sugere as mensagens de banners:

```
Switch(config)# banner exec @
     +---------------------------------------------------------------------------------------------------+
   |                                    ATENCAO                                                          |
   |                                    =========                                                        |
   |                Todo acesso nao autorizado e explicitamente PROIBIDO.                                |
   |Acessos nao autorizados serao passiveis de medidas administrativas e/ou legais                      |
    +---------------------------------------------------------------------------------------------------+
@
Switch(config)# banner motd @
      +---------------------------------------------------------------------------------------------------+
    |           UNIVERSIDADE TECNOLOGICA FEDERAL DO PARANA                                               |
    |                                  ===================                                              |
    |          Este equipamento e de utilizacao e acesso exclusivo da UTFPR                              |
    |          Caso nao tenha permissao, favor desconectar-se IMEDIATAMENTE                              |
    |         Em caso de duvidas, contacte a COGETI ou o administrador da rede                           |
    |                                    ####################                                             |
    |Acessos nao autorizados serao passiveis de medidas administrativas e/ou legais                      |
     +---------------------------------------------------------------------------------------------------+
@
```

4.4 Apêndice A

Ministério da Educação

Universidade Tecnológica Federal do Paraná

Pró-Reitoria de Graduação e Educação Profissional

Pró-Reitoria de Pesquisa e Pós-Graduação

Sistema de Bibliotecas

DECLARAÇÃO DE AUTORIA

Autores[1]: Pedro Henrique Modesto Deguchi / Francisco Bittencourt dos Santos

CPF[1]: 054112249-50 / 04373025983 Código de matrícula[1]: 887943 / 373796

Telefone[1]: (41) 30238612 / (41) 30191031 e-mail[1]: pdeguchi@gmail.com / franciscobitten@gmail.com

Curso/Programa de Pós-graduação: Curso Superior de Tecnologia em Desenvolvimento de Sistemas Distribuídos

Orientador: : Prof. Ms. Luiz Augusto Pelisson

Co-orientador: Prof. Ms. Wilson Horstmeyer Bogado

Data da defesa: 13 de Julho de 2012

Título/subtítulo: REESTRUTURAÇÃO DAS CAMADAS 2 E 3 (ENLACE E REDE) DA UTFPR CÂMPUS CURITIBA

Tipo de produção intelectual: (X) TCC[2] ( ) TCCE[3] ( ) Dissertação ( ) Tese

Declaro, para os devidos fins, que o presente trabalho é de minha autoria e que estou ciente:

- dos Artigos 297 a 299 do Código Penal, Decreto-Lei nº 2.848 de 7 de dezembro de 1940;
- da Lei nº 9.610, de 19 de fevereiro de 1998, sobre os Direitos Autorais,
- do Regulamento Disciplinar do Corpo Discente da UTFPR; e
- que plágio consiste na reprodução de obra alheia e submissão da mesma como trabalho próprio ou na inclusão, em trabalho próprio, de idéias, textos, tabelas ou ilustrações (quadros, figuras, gráficos, fotografias, retratos, lâminas, desenhos, organogramas, fluxogramas, plantas, mapas e outros) transcritos de obras de terceiros sem a devida e correta citação da referência.

_____ Curitiba, 10 de Agosto de 2012

_____

Assinatura dos Autores[1] Local e Data

[1] Para os trabalhos realizados por mais de um aluno, devem ser apresentados os dados e as assinaturas de todos os alunos.

[2] TCC – monografia de Curso de Graduação.

[3] TCCE – monografia de Curso de Especialização.

4.5 Apêndice B

Ministério da Educação

Universidade Tecnológica Federal do Paraná

Pró-Reitoria de Graduação e Educação Profissional

Pró-Reitoria de Pesquisa e Pós-Graduação

Sistema de Bibliotecas

TERMO DE AUTORIZAÇÃO PARA PUBLICAÇÃO DE TRABALHOS DE CONCLUSÃO DE CURSO DE GRADUAÇÃO E ESPECIALIZAÇÃO, DISSERTAÇÕES E TESES NO PORTAL DE INFORMAÇÃO E NOS CATÁLOGOS ELETRÔNICOS DO SISTEMA DE BIBLIOTECAS DA UTFPR

Na qualidade de titular dos direitos de autor da publicação, autorizo a UTFPR a veicular, através do Portal de Informação (PIA) e dos Catálogos das Bibliotecas desta Instituição, sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9.610/98, o texto da obra abaixo citada, observando as condições de disponibilização no item 4, para fins de leitura, impressão e/ou *download*, visando a divulgação da produção científica brasileira.

1. Tipo de produção intelectual: (E) TCC[1] ( ) TCCE[2] ( ) Dissertação ( ) Tese

2. Identificação da obra:

Autores[3]: Pedro Henrique Modesto Deguchi / Francisco Bittencourt dos Santos

RG[3]: 8306062-8 / 85579576 CPF[3]:054112249-50 / 04373025983

Telefone[3]: (41) 30238612 / (41) 30191031

e-mail[3]: pdeguchi@gmail.com / franciscobitten@gmail.com

Curso/Programa de Pós-graduação: Curso Superior de Tecnologia em Desenvolvimento de Sistemas Distribuídos

Orientador: Prof. Ms. Luiz Augusto Pelisson

Co-orientador: Prof. Ms. Wilson Horstmeyer Bogado

Data da defesa:13 de Julho de 2012

Título/subtítulo (português): REESTRUTURAÇÃO DAS CAMADAS 2 E 3 (ENLACE E REDE) DA UTFPR CÂMPUS CURITIBA

Título/subtítulo em outro idioma: RESTRUCTURING LAYERS 2 AND 3 (DATA LINK AND NETWORK) OF UTFPR CURITIBA

Área de conhecimento do CNPq: _____

Palavras-chave: Reestruturação, camadas 2 e 3, enlace e rede_____

Palavras-chave em outro idioma: Reestructuring, layers 2 and 3, data link and network_____

3. Agência(s) de fomento (quando existir): _____

4. Informações de disponibilização do documento:

Restrição para publicação: ( ) Total[4] ( ) Parcial[4] (X) Não Restringir

Em caso de restrição total, especifique o por que da restrição: _____

_____

Em caso de restrição parcial, especifique capítulo(s) restrito(s): _____

_____

Curitiba 10/08/2012

Local e Data

_____ _____

Assinatura do Autor[3] Assinatura do Orientador

[1] TCC – monografia de Curso de Graduação.

[2] TCCE – monografia de Curso de Especialização.

[3] Para os trabalhos realizados por mais de um aluno, devem ser apresentados os dados e as assinaturas de todos os alunos.

[4] A restrição parcial ou total para publicação com informações de empresas será mantida pelo período especificado no Termo de Autorização para Divulgação de Informações de Empresas. A restrição total para publicação de trabalhos que forem base para a geração de patente ou registro será mantida até que seja feito o protocolo do registro ou depósito de PI junto ao INPI pela Agência de Inovação da UTFPR. A íntegra do resumo e os metadados ficarão sempre disponibilizados.