

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

GUILHERME IRIA D'ABBADIA FONTES PEREIRA
THIAGO SANGIACOMO MASELLO

ESTUDO DE TÉCNICAS MULTICAMINHOS EM CAMADA
DE ENLACE PARA NOVAS DEMANDAS EM REDES DE
DATACENTERS MODERNOS

RIO DE JANEIRO

2019

GUILHERME IRIA D'ABBADIA FONTES PEREIRA
THIAGO SANGIACOMO MASELLO

ESTUDO DE TÉCNICAS MULTICAMINHOS EM CAMADA
DE ENLACE PARA NOVAS DEMANDAS EM REDES DE
DATACENTERS MODERNOS

Trabalho de conclusão de curso de graduação apresentado ao Departamento de Ciência da Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Profa. Silvana Rossetto,
D.Sc., UFRJ

RIO DE JANEIRO

2019

P436e

Pereira, Guilherme Iria D'Abbadia Fontes

Estudo de técnicas multicaminhos em camada de enlace para novas demandas em redes de datacenters modernos / Guilherme Iria D'Abbadia Fontes Pereira, Thiago Sangiacomo Masello. – 2019.

82 f.

Orientadora: Silvana Rossetto.

Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Federal do Rio de Janeiro, Instituto de Matemática, Bacharel em Ciência da Computação, 2019.

1. Redes de computadores. 2. Análise comparativa. 3. Fluxo de dados. I. Masello, Thiago Sangiacomo. II. Rossetto, Silvana (Orient.). III. Universidade Federal do Rio de Janeiro, Instituto de Matemática. IV. Título.

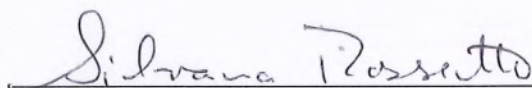
GUILHERME IRIA D'ABBADIA FONTES PEREIRA
THIAGO SANGIACOMO MASELLO

ESTUDO DE TÉCNICAS MULTICAMINHOS EM CAMADA
DE ENLACE PARA NOVAS DEMANDAS EM REDES DE
DATACENTERS MODERNOS

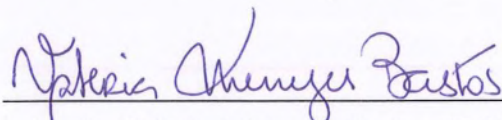
Trabalho de conclusão de curso de graduação apresentado ao Departamento de Ciência da Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 23 de maio de 2019.

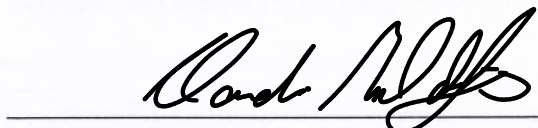
BANCA EXAMINADORA:



Profa. Silvana Rossetto, D.Sc., UFRJ



Profa. Valeria Menezes Bastos, D.Sc., UFRJ



Prof. Claudio Miceli de Farias, D.Sc., UFRJ

AGRADECIMENTOS

Guilherme Iria D'Abbadia Fontes Pereira

Agradeço aos meus pais e familiares próximos, principalmente minha vó Lucie (In memoriam).

À Professora Silvana por toda dedicação e atenção como orientadora neste trabalho, e aos professores e professoras pelo conhecimento transmitido ao longo do curso.

Thiago Sangiacomo Masello

Gostaria de agradecer aos meus pais e à minha família por toda a dedicação e educação proporcionada, em especial à minha mãe.

À minha orientadora por toda a atenção, paciência e dedicação no decorrer deste trabalho, e a todos os professores pelo conhecimento e pelos ensinamentos fornecidos.

Enfim, à todas as pessoas que, de alguma forma, contribuíram direta ou indiretamente para que a conclusão deste trabalho fosse possível.

RESUMO

Nos últimos anos, as redes de *datacenter* passaram a experimentar uma mudança significativa no padrão do tráfego de dados. Requisitos como alta disponibilidade de banda e tolerância a falhas se tornaram indispensáveis, de modo que se fez necessário o surgimento de novos modelos de organização topológica dessas redes e novos protocolos de operação, em substituição aos tradicionais. Este trabalho se propõe a apresentar um estudo sobre as mudanças observadas no padrão de tráfego das redes de *datacenter* e os problemas gerados pelo uso dos modelos e protocolos convencionais, assim como avaliar modelos e protocolos alternativos que visam contornar estes problemas e atender às novas demandas e requisitos desses ambientes. Realizou-se um estudo teórico e experimentos práticos em cenários específicos, de caráter comparativo, com o uso de protocolos propostos como solução e protocolos tradicionais, demonstrando a melhora na eficiência da operação da rede no contexto apresentado.

Palavras-chave: TRILL. Datacenter. Redes de Computadores. STP. IS-IS. ECMP. Fluxo de Dados. Análise Comparativa.

ABSTRACT

Over time, datacenter networks have experienced a significant change in the pattern of data traffic. Requirements such as high bandwidth availability and fault tolerance became indispensable, so that new models of topological organization of these networks and new operating protocols were necessary, replacing the traditional ones. This paper proposes to present a study on the observed changes in the traffic pattern of datacenter networks and the problems generated by the use of conventional models and protocols, as well as to evaluate alternative models and protocols that aim to overcome these problems and to meet the new demands and requirements of these environments. A theoretical study and practical experiments were carried out in specific scenarios of a comparative nature, using protocols proposed as solution and traditional protocols, demonstrating the improvement in the efficiency of the network operation in the presented context.

Keywords: TRILL. Datacenter. Computer Networks. STP. IS-IS. ECMP. Throughput. Comparative Analysis.

LISTA DE FIGURAS

Figura 1:	Pilha de Protocolos TCP/IP	20
Figura 2:	Modelo OSI vs. Modelo TCP/IP	21
Figura 3:	Plano de Controle e Plano de Dados	22
Figura 4:	Modelo Hierárquico Convencional de Três Camadas	28
Figura 5:	Requisições ARP	30
Figura 6:	<i>Broadcast Storm</i>	31
Figura 7:	Árvore de Dispersão	32
Figura 8:	Seleção de Portas com o <i>Spanning Tree Protocol</i>	33
Figura 9:	Uso do campo Idade da Mensagem no STP	35
Figura 10:	Quadro Comparativo entre Big Data e Dados Tradicionais	38
Figura 11:	Capacidade de Recursos no Modelo Tradicional	38
Figura 12:	Capacidade de Recursos no Modelo Spine Leaf	39
Figura 13:	Bloqueio de Caminhos Redundantes no STP	42
Figura 14:	Utilização de Caminhos não-ótimos com o STP	43
Figura 15:	Arquitetura <i>Spine-Leaf</i>	45
Figura 16:	Topologia com Caminhos de Custos Diferentes	47
Figura 17:	Nuvem TRILL	49
Figura 18:	Tabela de Encaminhamento do TRILL	50
Figura 19:	Cabeçalho TRILL	52
Figura 20:	Topologia Mista de Redes TRILL e STP	53
Figura 21:	Topologia Mista de Redes TRILL e STP vista pelas <i>RBridges</i>	54
Figura 22:	Designação de VLANs para <i>RBridges</i> em Topologias Multiacesso	57
Figura 23:	Troca de Mensagens ICMP	62
Figura 24:	Topologia Utilizada nos Cenários de Experimentação	65
Figura 25:	Topologia dos Testes Após Convergência do STP	67
Figura 26:	Mudança Topológica no Cenário do Experimento 1	68
Figura 27:	Mudança Topológica no Cenário do Experimento 2	69
Figura 28:	Mudança Topológica no Cenário do Experimento 3	70
Figura 29:	Comparação dos Cabeçalhos TRILL e FabricPath	71
Figura 30:	Quadro de Resultados dos Experimentos	73

Figura 31: Gráfico IPG x Experimento/Etapa	74
Figura 32: Gráfico <i>Throughput</i> x Experimento/Etapa	75

LISTA DE TABELAS

Tabela 1: Perda de Pacotes Durante a Transmissão	72
--	----

LISTA DE ABREVIATURAS E SIGLAS

ACL	Access Control List
AP	Access Point
API	Application Programming Interface
ARP	Address Resolution Protocol
BID	Bridge ID
BPDU	Bridge Protocol Data Unit
ECMP	Equal Cost Multipath
EIGRP	Enhanced Interior Gateway Routing Protocol
FCS	Frame Check Sequence
FP	FabricPath
FTP	File Transfer Protocol
HTTP	HyperText Transfer Protocol
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IP	Internet Protocol
IPG	Inter Packet Gap
IS-IS	Intermediate System to Intermediate System
ISO	International Standards Organization
L2MP	Layer 2 Multipath
LAN	Local Area Network
LSDB	Link State Database

LSP	Link State Packet
MAC	Media Access Control
MTU	Maximum Transmission Unit
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
OTV	Overlay Transport Virtualization
PDU	Protocol Data Unit
POE	Power Over Ethernet
PVST	Per-VLAN Spanning Tree
RB	Routing Bridge
RFC	Request for Comments
RSTP	Rapid Spanning Tree Protocol
RTT	Round Trip Time
SDN	Software Defined Networking
SPB	Shortest Path Bridge
SSH	Secure Shell
STP	Spanning Tree Protocol
TCP	Transmission Control Protocol
TLV	Type-Length-Value
TRILL	Transparent Interconnection of Lots of Links
TTL	Time to Live
UDP	User Datagram Protocol
VLAN	Virtual Local Area Network
VM	Virtual Machine (Máquina Virtual)

SUMÁRIO

1	INTRODUÇÃO	14
1.1	OBJETIVOS E METODOLOGIA	16
1.2	ORGANIZAÇÃO DO TEXTO	17
2	CONCEITOS BÁSICOS	18
2.1	PILHA DE PROTOCOLOS TCP/IP	18
2.2	PLANO DE CONTROLE E PLANO DE DADOS	21
2.3	ALGORITMOS DE ROTEAMENTO	23
2.4	MODELO TOPOLÓGICO DE TRÊS CAMADAS	27
2.5	REQUISIÇÕES ARP	29
2.6	SPANNING TREE PROTOCOL	31
3	NOVAS DEMANDAS PARA OS <i>DATACENTERS</i>	36
3.1	MUDANÇAS NO PADRÃO DE TRÁFEGO	36
3.2	BIG DATA	37
3.3	COMPUTAÇÃO EM NUVEM	40
3.4	PROBLEMAS COM O SPANNING TREE PROTOCOL	41
4	PROPOSTAS PARA ATENDER ÀS NOVAS DEMANDAS DOS <i>DATACENTERS</i>	44
4.1	RECURSOS DE SUPORTE ÀS PROPOSTAS	44
4.1.1	Arquitetura <i>Spine-Leaf</i>	45
4.1.2	Agregação de Links	45
4.1.3	ECMP	46
4.2	TRILL	47
4.2.1	Encaminhamento e Rotas	49
4.2.2	Cabeçalho TRILL e Encapsulamento	51
4.2.3	Topologias Mistas	52
4.3	IS-IS	54
4.3.1	Mudanças do IS-IS para o TRILL	55
4.3.2	Pacotes de Controle do TRILL	55

4.3.3	Descoberta dos Vizinhos no TRILL	56
4.3.4	Adquirindo Identificadores	58
5	EXPERIMENTOS E RESULTADOS	60
5.1	METODOLOGIA	60
5.2	FERRAMENTAS AUXILIARES E MÉTRICAS	63
5.3	CENÁRIOS DE EXPERIMENTAÇÃO	65
5.3.1	Experimento 1	68
5.3.2	Experimento 2	69
5.3.3	Experimento 3	69
5.4	IMPLEMENTAÇÃO BASEADA NO TRILL: FABRICPATH	70
5.5	RESULTADOS	72
6	CONCLUSÃO	77
	REFERÊNCIAS	79

1 INTRODUÇÃO

Datacenters são instalações que abrigam uma infraestrutura de sistemas computacionais (e diferentes sistemas de suporte) com a finalidade de fornecer recursos destes sistemas em larga escala. Geralmente possuem estruturas físicas montadas especialmente para suportá-los [35].

Com o passar do tempo, as demandas por esses recursos computacionais aumentaram, e conseqüentemente mudanças em diversos aspectos foram exigidas. Em consequência disso, as estruturas físicas que antes suportavam essas instalações precisaram de adequações para acompanhar essa evolução, assim como os modelos conceituais do funcionamento de um *datacenter*. Modelos que antes atendiam as necessidades desses *datacenters*, hoje já não o fazem e encontram-se defasados e obsoletos.

Até pouco tempo atrás, quando os modelos de serviço cliente-servidor apresentavam predominantemente usuários finais como clientes e servidores físicos (*bare-metal*), ao invés de virtualizados, *datacenters* se utilizavam do mesmo modelo topológico tradicional de redes corporativas. Quando esses modelos topológicos foram desenvolvidos, dispositivos finais da rede eram, por exemplo, computadores de usuários, impressoras, pontos de acesso, entre outros, cuja frequência de requisições e nível de exigência em relação à disponibilidade e uso de banda eram relativamente pequenos.

Agora, ao invés de computadores pessoais na ponta da rede, temos servidores realizando centenas ou milhares de requisições por segundo [4], ou então necessitando transferir grandes quantidades de dados entre si. Nesse contexto, a exigência por disponibilidade e largura de banda se torna bem maior. Falhas na infraestrutura e certos tempos de recuperação que antes eram aceitáveis, agora se tornam catastróficos. Um cenário onde a demanda por serviços e aplicações é tão grande que se torna necessário uma infraestrutura dedicada para provê-los, é no que consiste um *datacenter* moderno. Por esse motivo, modelos convencionais de infraestrutura de rede se tornam inviáveis para as novas necessidades dos *datacenters*, fazendo com

que seja necessário a criação de novos modelos que sejam, por sua vez, pensados e desenhados especificamente para esse novo e exigente cenário.

Cenários convencionais de redes corporativas utilizam um dos protocolos mais importantes de redes *Ethernet*, o STP [30] (*Spanning Tree Protocol*). O STP foi criado para mitigar problemas de degradação severa de redes causada por requisições *broadcast* e sua capacidade de expansão sem controle pelas interfaces dos dispositivos conectados, a partir da presença de caminhos redundantes na topologia da rede. O STP atua identificando e bloqueando caminhos redundantes entre comutadores de um mesmo domínio de camada de enlace.

A forma utilizada pelo STP para tratar esse problema cria efeitos colaterais, como o fato de forçar a utilização de um caminho não ótimo, uma vez que os *links* do caminho ótimo podem ser desabilitados pelo protocolo; ou então impedir a utilização da banda total disponível ao desativar *links* redundantes. Esses efeitos são amenizados a partir de versões mais atuais do protocolo (*Rapid STP*, por exemplo, que é o padrão vigente do STP), ou são de menor impacto em modelos convencionais de topologias de rede. No entanto, para infraestruturas que demandam maior eficiência na utilização da largura de banda e alta disponibilidade, estes efeitos colaterais são desastrosos. Como solução, surgem modelos e tecnologias que visam resolver os problemas que o STP se propôs a resolver, porém, de forma a anular os efeitos colaterais que este causa.

Uma dessas propostas é o padrão TRILL [42], proposto pela IETF, que possui, dentre outras funcionalidades, a capacidade de suprimir o uso do STP em partes da topologia onde se faz necessário, ao mesmo tempo que trata os problemas decorrentes da utilização de *links* duplicados e redundantes, porém, sem gerar os efeitos colaterais provenientes do uso do STP, como citado anteriormente. Uma característica importante do TRILL é a interoperabilidade que ele apresenta com outros componentes presentes nos modelos de comunicação, o que permite que ele seja implementado sem a necessidade de mudanças significativas na comunicação como um todo.

Entretanto, não são apenas os protocolos e o plano de controle da rede que pre-

cisam de modificações. Uma outra mudança que se mostra necessária no cenário de *datacenter* é quanto a questão de sua topologia, pois agora novos padrões de tráfego se tornam predominantes. Em um cenário de rede convencional, a maior parte dos dados trafega em um padrão norte-sul, ou seja, onde os dispositivos finais da rede acessam na maioria das vezes recursos externos. Dessa forma, o tráfego flui das camadas mais baixas (acesso) diretamente para as camadas mais altas (núcleo), com pouca utilização dos *links* de interconexões das camadas intermediárias. Já no cenário de *datacenter*, a infraestrutura computacional e de armazenamento faz com que o fluxo predominante seja muito diferente, onde a maior parte dos dados flui na direção leste-oeste. Dessa forma a maior parte do tráfego se concentra dentro da própria rede, pois agora, na mesma camada lógica, os servidores utilizam armazenamento externo compartilhado. Além disso, com o crescimento do uso de ambientes virtualizados, os dados computacionais estão em constante migração entre esses ambientes. Utilizando a topologia convencional para esse cenário, apenas cerca de metade da capacidade de banda agregada fica disponível [1]. Dessa forma torna-se clara a necessidade na alteração do desenho da rede de modo a favorecer um uso mais eficiente do que antes era pouco necessário.

1.1 OBJETIVOS E METODOLOGIA

O objetivo deste trabalho consiste em apresentar um estudo sobre os problemas e os empecilhos existentes nas arquiteturas e protocolos convencionais utilizados em redes de *datacenters*, uma vez que novas demandas são encontradas a partir do surgimento de aplicações que exigem cada vez mais da infraestrutura e da banda por ela provida; e avaliar alternativas propostas para atender a essas novas demandas.

Para realizar essa avaliação, exploramos o padrão TRILL, seu modo de operação e suas funcionalidades. Utilizamos para a realização dos experimentos, o protocolo proprietário *FabricPath* [19], baseado no TRILL e desenvolvido pela Cisco. Definimos alguns cenários básicos para experimentação e mostramos os ganhos alcançados por essas soluções.

1.2 ORGANIZAÇÃO DO TEXTO

O restante deste trabalho está dividido em três partes. A primeira parte, contida no Capítulo 2, diz respeito a apresentação dos conceitos básicos utilizados para o posterior desenvolvimento dos tópicos principais do trabalho, incluindo a organização topológica de uma rede de computadores e seus principais protocolos e modelos de funcionamento. A segunda parte, contida nos Capítulos 3 e 4, trata das novas demandas em um ambiente de *datacenter* resultantes das mudanças observadas no padrão de tráfego da rede a partir da expansão de aplicações de *Big Data* e Computação em Nuvem; e da busca por novas soluções para essas demandas e para problemas descritos previamente. Por fim, a terceira e última parte, contida nos Capítulos 5 e 6, apresenta experimentos que foram realizados para avaliar as soluções estudadas e as conclusões do trabalho.

2 CONCEITOS BÁSICOS

Alguns conceitos são fundamentais para o presente trabalho e são expostos neste capítulo de forma sucinta. Apresentamos inicialmente o modelo TCP/IP que agrega protocolos de comunicação utilizados na transmissão de pacotes em redes de computadores e é o modelo utilizado na Internet. Ao longo deste trabalho são referenciados protocolos que atuam em camadas específicas desse modelo, tornando fundamental o entendimento do mesmo. Em seguida, descrevemos o modelo convencional de três camadas e o padrão de tráfego que ele atende. Diferentemente do modelo TCP/IP, que é um modelo de comunicação, o modelo convencional de três camadas é um modelo topológico, representa a disposição dos comutadores de pacotes (*switches*) e a hierarquia entre eles durante os processos de comunicação na rede. O entendimento deste modelo é necessário para que posteriormente ele possa ser comparado ao modelo proposto para redes de *datacenter*. Explicita-se as diferenças entre o plano de controle e o plano de dados e os algoritmos de roteamento utilizados. Em seguida, apresentamos um detalhamento do protocolo ARP e suas requisições, acompanhado do problema que este pode ocasionar na rede se não tratado de forma adequada. O tratamento para esse problema é oferecido pelo protocolo STP, que é descrito ao final do capítulo.

2.1 PILHA DE PROTOCOLOS TCP/IP

O TCP/IP é um modelo de comunicação organizado em camadas com interfaces bem definidas, onde cada camada é responsável por um serviço de comunicação. O objetivo principal dessa estruturação em camadas é permitir que o protocolo (ou a implementação) de uma camada específica possa ser substituído por outro da mesma camada sem a necessidade de alterar todo o processo de comunicação [24]. Em suma, ao final, os mesmos dados serão entregues de maneira bem definida para a camada superior, mas o método de obtenção dos dados pode variar livremente sem comprometer o resultado final e sem a necessidade de ajustes nos protocolos de outras camadas.

O modelo TCP/IP apresenta quatro camadas: camada de acesso à rede, camada de internet, camada de transporte e camada de aplicação, definidas a seguir.

Camada de acesso à rede: Esta camada é atrelada ao escopo de redes locais. Ela é responsável pela parte da comunicação que define como o sinal que carrega os dados trafega pelo meio físico e como a informação é trocada entre dispositivos vizinhos, ou seja, direta e fisicamente conectados sem a intervenção de um roteador. Como exemplos de protocolos pertencentes a esta camada temos o protocolo *Ethernet* e o ARP que serão posteriormente apresentados.

Camada de internet: Esta camada é a responsável por estabelecer a comunicação e traçar rotas entre redes distintas, é ela que provê o endereçamento lógico de um dispositivo na rede e permite a comunicação com qualquer outro dispositivo que esteja fora da sua rede local. Assim sendo, ela serve de base estrutural para o funcionamento da rede Internet em si. Seu principal protocolo é o IP (*Internet Protocol*), além de possuir protocolos importantes como o ICMP (*Internet Control Message Protocol*) e o IGMP (*Internet Group Management Protocol*) que carregam mensagens de controle entre os dispositivos intermediários da rede.

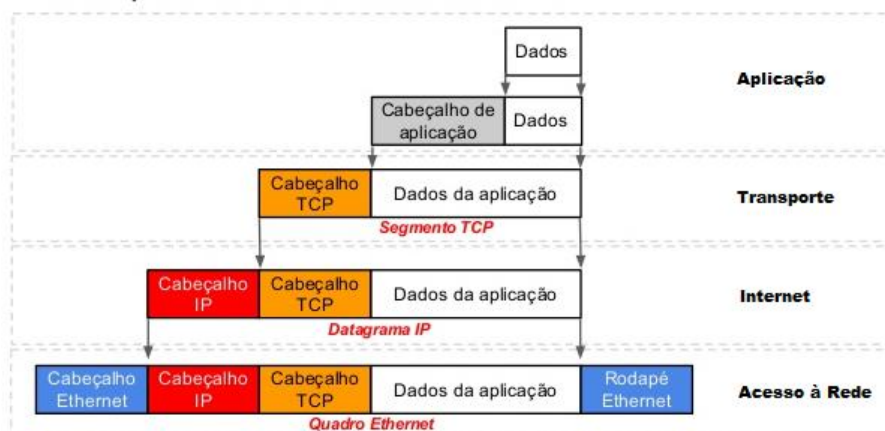
Camada de transporte: Esta camada fica responsável pela comunicação fim a fim, ou seja, não só ao nível de dispositivo mas também se expandindo ao nível do serviço que está sendo executado internamente em cada ponta. Um único dispositivo pode estar executando uma série de serviços onde cada um está recebendo ou enviando pacotes de forma independente. Essa independência de fluxos é obtida com a camada de transporte. Enquanto a camada anterior (internet) endereça o dispositivo na rede, o protocolo da camada de transporte vai além e endereça também o serviço ao qual o fluxo atual pertence, tanto na ponta de origem como na de destino. Assim sendo, é a combinação dos endereços de rede (tratados na camada de internet) e do identificador do serviço (tratados na camada de transporte) de ambas as pontas o que define uma identificação única para os fluxos que trafegam pela Internet. Além disso, mecanismos de controle de fluxo, de estabelecimento de conexão e de garantia de entrega na transmissão de dados também são funções dos protocolos da camada de transporte. Dos quais, dentre os principais, podemos citar

o TCP (*Transmission Control Protocol*) e o UDP (*User Datagram Protocol*).

Camada de aplicação: Por último, esta camada é a que possui o maior nível de abstração e também a mais próxima do usuário com relação aos seus dados. Ela carrega os dados úteis que são transmitidos de uma ponta a outra e são processados e interpretados de acordo com a arquitetura e a estrutura de cada aplicação. Os protocolos que atuam nessa camada são essencialmente serviços sendo executados em cada dispositivo. Serviços conhecidos que atuam na camada de aplicação são, dentre muitos, FTP (*File Transfer Protocol*), SSH (*Secure Shell*) e HTTP (*HyperText Transfer Protocol*).

Com as camadas bem definidas, é possível perceber que o processo de transmissão consiste em primeiro encapsular os dados junto com os cabeçalhos dos protocolos das camadas seguintes no dispositivo transmissor, depois enviá-los através do meio físico e, por fim, realizar o processo inverso no dispositivo receptor, desencapsulando os cabeçalhos a medida que o dado chega ao destinatário. O processo de encapsulamento é ilustrado na Figura 1. Dispositivos intermediários que operam em camadas específicas do modelo TCP/IP realizam o desencapsulamento e encapsulamento dos dados apenas até a camada a qual pertencem. É o caso de *switches* e roteadores que operam respectivamente nas camadas de acesso à rede e de internet.

Figura 1: Pilha de Protocolos TCP/IP

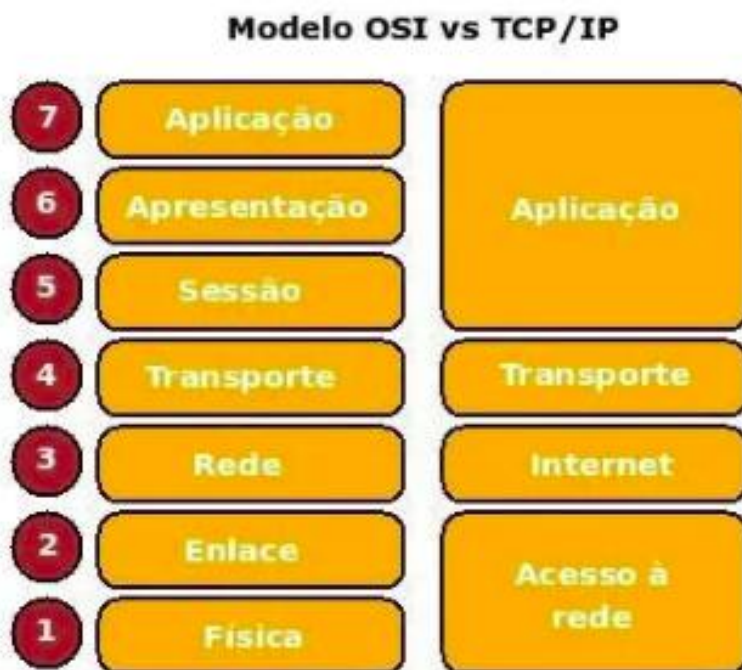


A pilha TCP/IP foi o modelo no qual a Internet se estruturou, posteriormente, a Organização Internacional para Padronização (ISO) lançou um modelo teórico para

fomentar a interoperabilidade entre diferentes tipos de rede. Este é o modelo de referência OSI [44].

Diferentemente do modelo TCP/IP, o modelo OSI apresenta sete camadas, são elas: Física, Enlace, Rede, Transporte, Sessão, Apresentação e Aplicação. Apesar da diferença numérica, é possível estabelecer uma relação entre os dois modelos. Essa relação é mostrada na Figura 2.

Figura 2: Modelo OSI vs. Modelo TCP/IP



Dessa forma, neste trabalho, quando referenciadas as camadas utilizando-se sua numeração, a referência a ser tomada é o modelo OSI. A fim de exemplificação, as camadas dois e três, ou seja, enlace e rede, referem-se respectivamente às camadas de acesso à rede e de internet do modelo TCP/IP.

2.2 PLANO DE CONTROLE E PLANO DE DADOS

Dispositivos de rede são classificados de acordo com sua atuação no tráfego e nos dados trafegados pela rede. Com relação às camadas de enlace e de rede, este tratamento se dá a nível de inspeção das informações contidas nos campos referentes à

unidade de dados de protocolo, PDU (*Protocol Data Unit*) e a decisão baseada nessas informações. Esta sessão pretende expor de forma simplificada como os dispositivos de roteamento atuam em uma rede de computadores.

Todo roteador possui basicamente duas funções principais chamadas de encaminhamento (ou repasse) e roteamento. Para realizar essas funções, o roteador emprega diferentes componentes de sua arquitetura, aqui representados na Figura 3. Um pacote de entrada pode ingressar em um roteador por alguma de suas portas (no exemplo pela porta de entrada Fa0/1) e sair por uma porta de saída (no exemplo pela porta Fa0/2). O endereço de destino do pacote é analisado e, baseado na correspondência desse endereço com as informações contidas na tabela de encaminhamento e outros elementos dependentes da arquitetura do roteador, uma porta de saída é definida e o pacote encaminhado. Até este ponto, os elementos envolvidos fazem parte do plano de dados. Este plano é frequentemente implementado em hardware pela necessidade de operação em ordem de nanossegundos [24].

Figura 3: Plano de Controle e Plano de Dados



Fonte: NetworkLessons.com

As informações contidas na tabela de encaminhamento são construídas a partir de componentes encontrados no plano de controle que, diferentemente do plano de dados, é geralmente implementado a nível de software e possui um processador tradicional como componente físico principal. Junto ao plano de controle atuam os protocolos de roteamento dinâmicos como os ilustrados na Figura 3: OSPF (*Open Shortest Path First*) e o EIGRP (*Enhanced Interior Gateway Routing Protocol*), e

seus respectivos componentes de suporte como tabelas de vizinhança, base de dados de estado de enlaces e tabelas de topologia; além das entradas de rotas estáticas (adicionadas diretamente). Estes recursos são utilizados para compor a tabela de roteamento que serve de base para o preenchimento da tabela de encaminhamento.

Na tabela de roteamento estão listadas, basicamente, as rotas contendo os prefixos de uma determinada rede, o custo e a interface de saída. Os protocolos de roteamento dos roteadores da rede interagem entre seus vizinhos, trocando informações sobre estado e custos dos enlaces. A partir dessas informações, o algoritmo de roteamento pode determinar o melhor caminho para uma determinada rede. Os algoritmos de roteamento serão discutidos na próxima sessão.

Esta disposição dos elementos dos planos mencionados acima, com o plano de controle trabalhando de forma descentralizada, os componentes organizados de forma vertical e em um mesmo equipamento, foi o objeto de estudo de muitos pesquisadores e muitos trabalhos foram feitos propondo alternativas a essa arquitetura, como por exemplo a de Redes Definidas por *Software* (SDN) [9].

2.3 ALGORITMOS DE ROTEAMENTO

A camada de rede (ou de internet, no modelo TCP/IP) é responsável pelo roteamento das mensagens até o seu destino final. No roteamento, a interação entre os equipamentos responsáveis pela interconexão de redes — os roteadores — permite determinar os caminhos que os pacotes de dados deverão percorrer para sair de seu nó de origem (neste caso o roteador de primeiro salto do nó de origem) e chegar ao seu nó de destino (roteador de primeiro salto do nó de destino). O cálculo das rotas é realizado pelo algoritmo de roteamento em atividade no plano de controle dos roteadores. Vale ressaltar que, pelo fato do enfoque ser na camada de roteamento, os cálculos e a topologia de rede são baseados nos roteadores, não nos dispositivos finais (computadores pessoais, servidores etc.), sendo estes representados pelos seus respectivos roteadores de primeiro salto, ou seja, o primeiro roteador que receberá um pacote de um dispositivo de origem, e último roteador do dispositivo de destino.

A tarefa, portanto, de um algoritmo de roteamento é a de calcular o melhor caminho entre o roteador de origem e o de destino. No geral, o melhor caminho, ou melhor rota, é o de menor custo. A definição de custo neste contexto não é fixa, mas geralmente envolve a largura de banda (ou capacidade) de um enlace de um segmento de rede. Grafos são utilizados para representar os problemas de roteamento. Por exemplo, um grafo $G = (N, E)$ é um conjunto N de nós e uma coleção E de arestas, onde cada aresta conecta um par de nós de N . Com esta representação, teremos os roteadores como os vértices do grafo e as arestas como os enlaces conectando dois roteadores.

Existem duas categorias principais de algoritmos de roteamento: vetor de distância e estado de enlace. Para o algoritmo de vetor de distância, o cálculo do caminho de menor custo é realizado iterativamente e de forma distribuída. Nesse caso, não há nó detentor da informação completa da rede, somente dos enlaces conectados a seus vizinhos diretos [24].

Diferentemente do algoritmo de roteamento do tipo vetor-distância, o algoritmo de roteamento de estado de enlace mantém informações do estado de toda a topologia, para posteriormente, a partir destas informações, realizar o cálculo que determina o custo a um determinado destino. Essa abordagem apresenta pontos fortes como: rápida convergência, robustez para comportar redes maiores (maior quantidade de nós e rotas) e menor suscetibilidade aos *loops* de roteamento. Isso se deve ao fato de que toda a topologia é conhecida pelo roteador ao calcular o melhor caminho para um determinado destino, utilizando o algoritmo do caminho mínimo de Dijkstra [11].

Nessas condições, uma rede pode ser modelada como um grafo orientado ponderado. Cada roteador sendo representado por um vértice e cada link por uma aresta. Diferentes tipos de peso podem ser atribuídos às arestas de grafos distintos, como por exemplo, propagação do atraso entre dois nós, ou o inverso da largura de banda multiplicado por uma constante (sendo este algum valor maior ou igual à largura de banda do link com a maior disponibilidade de banda na rede), como em 2.1. Desta forma, o algoritmo de caminho mínimo dá prioridade a caminhos com a maior banda

disponível de forma efetiva [14].

$$Custo = \frac{1}{\text{Maior Largura de banda na rede}} \quad (2.1)$$

O procedimento que os protocolos de estado de enlace seguem, consiste basicamente nos quatro passos a seguir [41]:

- Descobrir vizinhos e aprender seus respectivos endereços de rede;
- Construir um pacote para difusão das informações aprendidas;
- Enviar este pacote e receber o mesmo tipo de pacote de seus vizinhos;
- Calcular o caminho mínimo para todos os outros roteadores;

Uma das primeiras tarefas de um nó de uma rede formada por roteadores executando um protocolo de rede de estado de enlace, assim que inicializado, é a de descobrir seus vizinhos. Todos os nós executando o protocolo enviam, pelas interfaces configuradas para participar do protocolo, pacotes de mensagens HELLO. Estes pacotes são enviados aos vizinhos diretamente conectados, porém não são encaminhados após o recebimento. Esses tipos de pacotes também são utilizados para a detecção de falhas de nós e *links*, pois após um determinado tempo sem recebê-los, a vizinhança estabelecida é desfeita e o nó não será mais considerado ao se montar uma nova topologia. Após esta fase, cada nó enviou e recebeu pacotes de HELLO, conhecendo seus vizinhos diretamente conectados por suas identificações únicas na rede.

Depois que as informações sobre os vizinhos locais são coletadas e armazenadas, o nó está preparado para formatar esta informação para envio através da construção dos LSPs (*Link State Packet* - Pacotes de Estado de Enlace) e posterior distribuição.

As informações encontradas no pacote contém basicamente os seguintes dados:

- Identificação do nó remetente;
- Idade do LSP;

- Número de sequência do LSP;
- Uma lista contendo o identificador de um vizinho do nó remetente e sua métrica a partir do próprio remetente.

Como cada nó possui conhecimento de toda a topologia, a quantidade de informações trocadas pode ser grande. O número de nós participantes é um fator determinante para a quantidade de atualizações, assim como a estabilidade da rede, pois considerando cada mudança na topologia que pode ser causada por falha de links, saída ou inserção de nós, a frequência de atualizações de topologia tende a aumentar. Alguns mecanismos são empregados para minimizar a necessidade de envio de atualizações e redução da quantidade de informação nas atualizações que serão enviadas.

Depois da fase de descobrimento, os nós trocam informações sobre as atualizações da topologia que cada um possui através do número de sequência do LSP. No caso dos dois possuírem a mesma versão, não há envio pendente sobre a topologia de qualquer um dos lados, de outra forma, o nó que possuir a informação mais recente a envia para o vizinho desatualizado. Este procedimento evita também o encaminhamento de informações desatualizadas o que causa o tráfego desnecessário de pacotes durante a etapa de *flooding*.

Flooding é uma técnica de difusão de informação local simples onde cada pacote recebido por um nó é encaminhado para todos os vizinhos deste nó com exceção do vizinho remetente do mesmo pacote. Um efeito comum e esperado desta técnica é o recebimento de pacotes duplicados por um mesmo nó, porém algumas medidas são empregadas para minimizar as consequências inerentes ao processo. Uma das medidas mais simples é a implementação do contador de nós percorridos (*hop count*), onde o mesmo é decrementado a cada salto e descartado ao alcançar o valor de zero ao invés de ser encaminhado. Porém, mesmo com o *hop count*, um número exponencial de pacotes duplicados pode ser gerado a medida que a rede se expande, e por isso, uma alternativa a esta técnica é a de possibilitar ao nó verificar se o pacote que está chegando já foi reencaminhado, evitando assim enviá-lo novamente. Para alcançar isto, cada nó mantém na base de dados do estado de enlace (LSDB - *Link State*

Database) a relação entre o nó remetente e o número de sequência do último LSP recebido desta fonte. Caso seja menor ou igual ao número armazenado para aquele nó remetente, o pacote é descartado.

Desta forma, com a técnica de *flooding*, a informação coletada localmente por cada nó é difundida de forma robusta e confiável através de toda a topologia permitindo a cada nó executar o próximo passo do protocolo.

Após receber as informações de todos os nós participantes da mesma área, cada roteador calcula localmente o menor caminho para todos os outros utilizando o algoritmo de caminho mínimo. Ao final do cálculo, a tabela de roteamento é populada com a relação entre um dos nós da topologia e por qual link o pacote deve ser encaminhado, a partir do roteador atual, para que siga pelo caminho mínimo determinado pelo algoritmo.

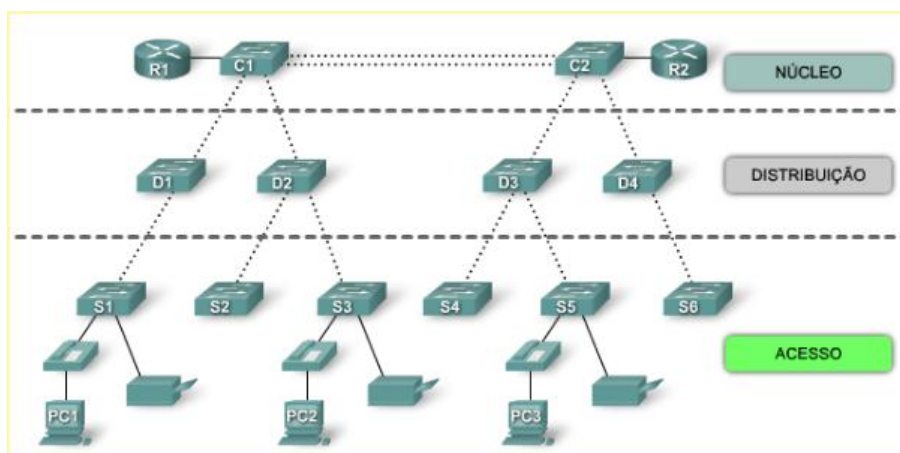
2.4 MODELO TOPOLÓGICO DE TRÊS CAMADAS

O modelo, ou seja, a topologia com a qual estruturamos os equipamentos que fazem parte de uma rede de computadores afeta diretamente a eficiência com que a mesma trata os dados que por ela trafegam. O conceito de redes convencionais ou tradicionais se refere a um modelo para um cenário específico, onde temos como *endpoints*: computadores pessoais, notebooks, impressoras, entre outros dispositivos, que irão gerar padrões de tráfego de dados característicos das suas aplicações. Com base nesse cenário, desenvolveu-se um modelo topológico que visa otimizar a operação da rede para um tráfego predominante nesses padrões, denominado modelo hierárquico de três camadas.

Esse modelo, conforme ilustrado na Figura 4, consiste em segmentar a rede em três camadas diferentes de modo a facilitar a gerência da mesma e otimizar a forma com que o tráfego de dados é tratado, tornando-a mais eficiente. As três camadas são núcleo, distribuição/agregação, e acesso.

A ideia da divisão em camadas cria um modelo hierárquico e visa trazer benefícios em termos de: (i) escalabilidade (quando há a necessidade de adicionar *endpoints* à

Figura 4: Modelo Hierárquico Convencional de Três Camadas



Fonte: Cisco CCNA Exploration V4.0

rede, exigindo assim a adição de *switches* de acesso); (ii) redundância (fazendo com que as camadas inferiores possam sempre acessar as camadas superiores e vice-versa, mesmo na falha de *links* ou equipamentos); (iii) segurança e gerenciabilidade (onde políticas de acesso e controle de portas são aplicados a nível de camada facilitando o controle e a gerência). As três camadas são descritas a seguir.

Camada de Núcleo: Essa camada é a “espinha dorsal” de toda a rede. Por ela passa praticamente todo o tráfego de uma rede de infraestrutura convencional. Na Figura 4, os *switches* C1 e C2 recebem todo o tráfego gerado pelos dispositivos finais e encaminham para os roteadores R1 ou R2. Dessa forma, é imprescindível que essa camada seja altamente disponível e redundante. É em geral a camada que também conecta a rede ao mundo externo. Logo, ela deve possuir links de grande capacidade para suportar todo o tráfego da rede.

Camada de Distribuição: Esta camada tem por função interconectar a camada de núcleo e a camada de acesso. Com isso ela passa a agregar múltiplos dispositivos da camada de acesso sendo necessária uma alta largura de banda para o manuseio de pacotes. Políticas de acesso e filtros de pacotes também são aplicados. Além de ser utilizada para conectar redes geograficamente segregadas.

Camada de Acesso: Esta é a camada que por sua vez conecta os dispositivos finais da rede, logo, apresenta, como uma de suas características principais, a alta

densidade de portas. São implementados nessa camada funções de VLAN e filtros de protocolos. Muitas das vezes apresenta dispositivos com capacidade de POE para alimentar APs, Câmeras, Telefones IP, dentre outros.

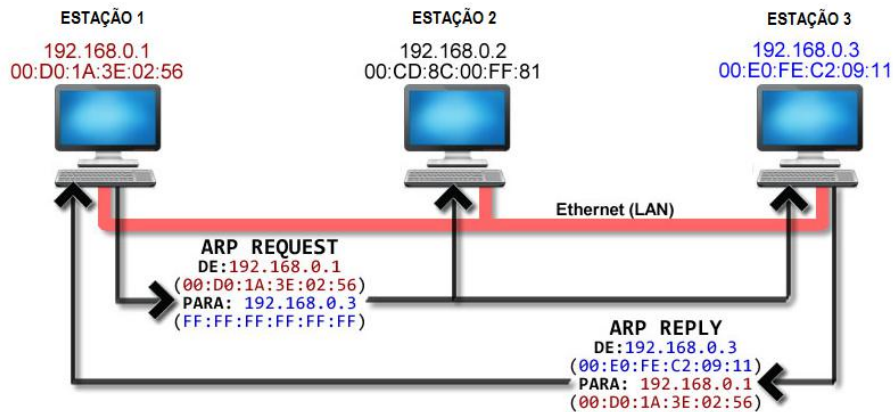
2.5 REQUISIÇÕES ARP

Quando um dispositivo deseja enviar um pacote pela rede, ele deve preencher o campo de endereço de destino do protocolo de camada dois com o endereço MAC do próximo dispositivo que irá receber o pacote. Porém, quando se tem apenas o endereço IP do destino, é preciso que o remetente descubra a qual endereço MAC ele está associado. Isso é feito enviando-se uma requisição ARP [33] (*ARP Request*), que consiste em um *broadcast* de camada dois para a rede, perguntando a todos os dispositivos alcançáveis no domínio qual deles possui o IP do destino para o qual se deseja enviar o pacote. Em seguida, o nó alvo que possui o endereço IP de destino, emite uma resposta ARP (*ARP Reply*), ou seja, ele devolve a informação necessária respondendo qual o seu endereço MAC. Essa é a finalidade do protocolo ARP (*Address Resolution Protocol*) que é usado para mediar a comunicação entre a camada de enlace (*Ethernet*) e a camada de rede (IP), em outras palavras, o protocolo relaciona endereços IP com endereços MAC.

Todo esse processo é ilustrado na Figura 5. No exemplo, uma estação de trabalho com o endereço de rede 192.168.0.1 e endereço físico 00:D0:1A:3E:02:56 (Estação 1) deseja estabelecer comunicação com outra estação de trabalho de endereço IP 192.168.0.3 e endereço físico 00:E0:FE:C2:09:11 (Estação 3). Primeiramente, a própria estação consulta sua tabela ARP em busca de alguma entrada referenciando o endereço IP alvo (192.168.0.3) com um endereço físico. Na ausência desta entrada, a estação envia a requisição ARP com sua informação de envio (endereço de IP e endereço físico do remetente) e com a informação de destino: o endereço IP alvo (da estação 3) e o endereço de *broadcast* do domínio (FF:FF:FF:FF:FF:FF). O *switch* realizará o *flood* dessa requisição, isto é, encaminhará o pacote para todas as suas portas excluindo aquela em que o mesmo foi recebido, chegando assim às estações 2 e 3. Porém, somente a estação 3 responderá a esta requisição com seu endereço

físico. Por fim, a estação 1 adiciona uma entrada na sua tabela ARP relacionando os dois endereços para serem utilizados nas comunicações futuras.

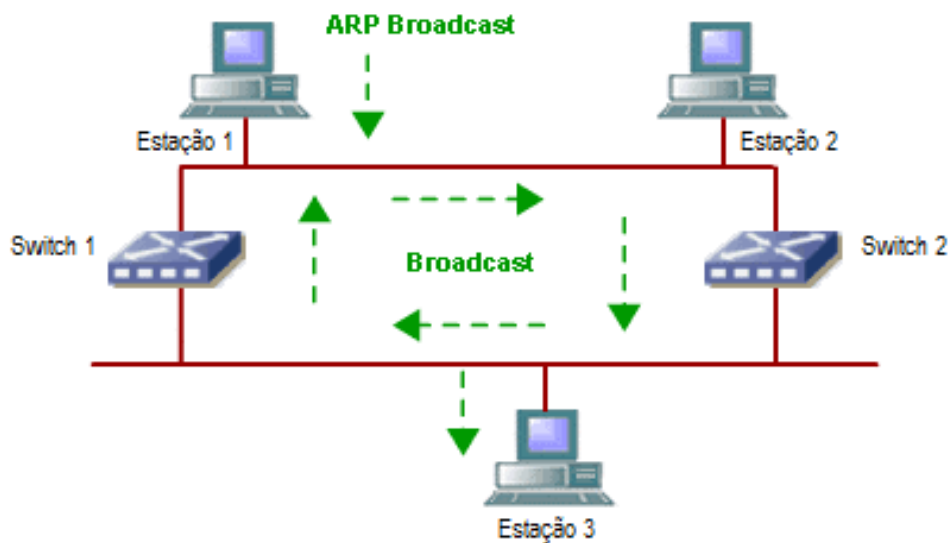
Figura 5: Requisições ARP



Requisições ARP é um tema importante a ser tratado, pois dependendo de como a topologia da rede foi estruturada, este tipo de requisição pode gerar um ciclo infinito de pacotes na rede causando uma ampla degradação do desempenho.

A Figura 6 exemplifica esta situação com dois *switches* interconectados. Quando o primeiro *switch* recebe uma requisição ARP de um dispositivo diretamente conectado a ele, imediatamente ele encaminha a requisição para todas as suas portas, excluindo a porta de origem, e incluindo as que o conecta ao *switch* dois. Este, por sua vez, ao receber o pacote irá encaminhá-lo para todas as suas portas com exceção daquela pela qual o pacote foi recebido. Dessa forma, os dois *switches* receberam um do outro, o mesmo pacote, e ainda o encaminharam por conseguintes vezes, gerando assim, uma sucessão de *broadcasts* na rede conhecida como *broadcast storm*.

Em pacotes da camada de rede, para evitar que um pacote circule na rede indefinidamente, existe o campo TTL (*Time to Live*), no qual um valor padrão é decrementado a cada dispositivo de camada três e descartado ao alcançar o valor zero. Entretanto, não há campo de TTL em pacotes de camada dois. Isso faz com que pacotes que por ventura entrem em ciclo de encaminhamento, passem a trafegar indefinidamente pela rede. Para evitar isso, se faz necessário um mecanismo de prevenção de ciclos. Este é o principal objetivo do protocolo de árvore de dispersão,

Figura 6: *Broadcast Storm*

Fonte: Cisco BTEC Internetwork Basics (Cap. 8)

o *Spanning Tree Protocol* ou simplesmente STP.

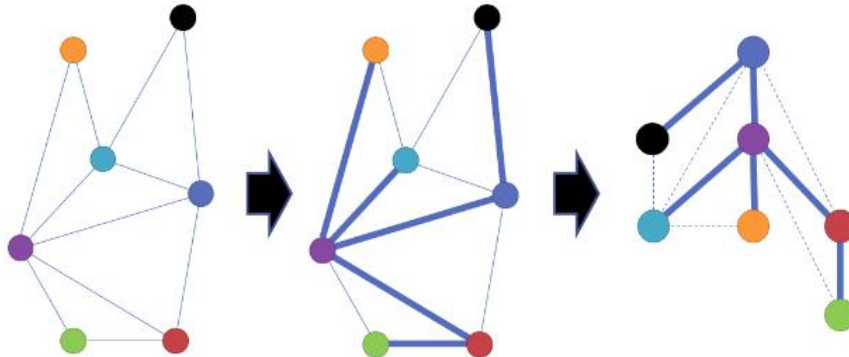
2.6 SPANNING TREE PROTOCOL

O STP [30] tem como objetivo primordial evitar *loops* na rede, anulando assim os efeitos de uma *broadcast storm*, ao mesmo tempo que permite a comunicação entre um nó e qualquer outro nó. O método usado para garantir uma topologia sem ciclos é o de montar a rede em uma estrutura de árvore, que por definição, é uma estrutura de dados livre de *loops*.

O entendimento de como o *Spanning Tree* monta essa estrutura é fundamental no escopo deste trabalho pois, como detalharemos mais adiante, a estrutura gerada pode se tornar um problema para redes de *datacenter*.

Como mostrado na Figura 7, o STP transforma a topologia da rede em uma árvore, mantendo desativado os caminhos secundários (pontilhados), ou seja, caminhos que se ativados podem gerar ciclos. Deixando assim, apenas um *link* ativo entre quaisquer dois dispositivos.

Figura 7: Árvore de Dispersão

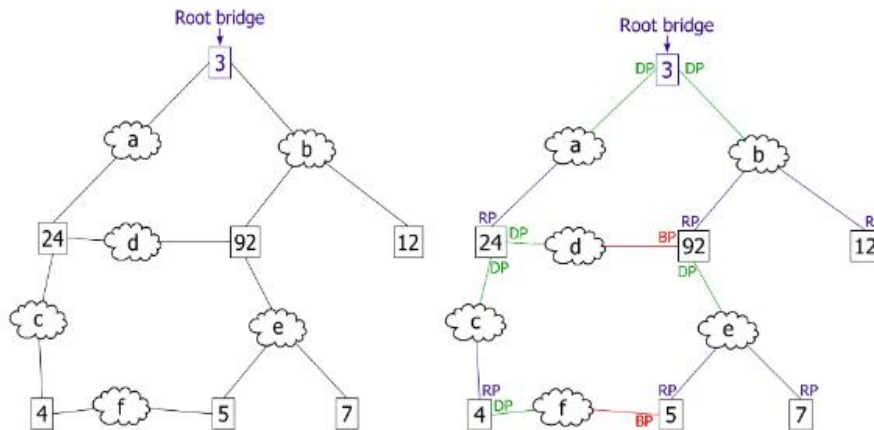


Para tal, é necessário primeiramente que seja eleito o dispositivo que será a raiz da árvore. Esse dispositivo é escolhido a partir do seu BID (*Bridge ID*), que é composto pela junção de dois parâmetros. O primeiro é o *Root Priority*, composto por dois *bytes* e que pode ser definido manualmente. O segundo é o endereço MAC do dispositivo, composto por seis *bytes*. O *switch* que possui o menor BID dentre todos do segmento de rede é eleito como raiz da árvore. Depois disso cada dispositivo da rede determina o custo de todos os caminhos possíveis até a raiz. Esse custo, por padrão, é implementado com base na largura de banda e segue uma fórmula específica para definição dos valores recomendados. Essa fórmula variou ao longo do tempo para acompanhar o crescimento da oferta de banda. Individualmente, os *switches* definem como *root port* (RP), a porta que os conecta ao caminho de menor custo até a raiz.

Em seguida, os *switches* de cada segmento de rede determinam coletivamente qual entre eles possui o menor caminho até a raiz. Esse *switch* então marca como *designated port* (DP) a porta que o conecta ao segmento. As outras portas do segmento são marcadas como *blocked port* (BP). Entende-se por segmento de rede portas que estão em um mesmo domínio de colisão.

A Figura 8 mostra, à direita, o resultado final da seleção de portas após concluído o processo descrito acima. Os segmentos de rede são representados pelas letras e os *switches* pelos números (onde o menor é eleito como raiz).

De forma a manter a rede livre de ciclos, os *switches* trocam certos pacotes de

Figura 8: Seleção de Portas com o *Spanning Tree Protocol*

dados que informam uns aos outros sobre as mudanças que ocorrem na topologia, de modo que todos possam atualizá-la em suas memórias para que não haja divergência de informações. A esses pacotes dá-se o nome de BPDUs (*Bridge Protocol Data Units*). E ao tempo que leva para todos os *switches* atualizarem suas tabelas quando ocorre uma mudança de topologia, dá-se o nome de tempo de convergência do STP. Durante a convergência do STP, as portas podem estar em cinco estados diferentes:

Blocking (Bloqueada): Este é o estado inicial quando uma porta é habilitada. A porta processa BPDUs, mas não encaminha nenhum dado, evitando assim a criação de ciclos após a sua inicialização. Caso a porta não seja uma *Blocked Port* (BP), após um determinado *timeout*, que por padrão é de 20 segundos, a porta avança para o próximo estado, *Listening*.

Listening (Ouvindo): Nesse estado a porta processa os BPDUs que recebe e também é autorizada a enviar BPDUs para os vizinhos. Ela não popula a tabela MAC nem encaminha nenhum dado.

Learning (Aprendendo): Ela ainda não encaminha dados, mas passa a popular a tabela MAC.

Forwarding (Encaminhando): A porta está completamente ativa. Ela encaminha pacotes e popula a tabela MAC. A porta recebe BPDUs e é monitorada para caso seja necessária a troca para o estado bloqueada de modo a evitar ciclos.

Disabled (Desabilitada): A porta foi manualmente desabilitada pelo administrador.

O STP possui temporizadores que interferem diretamente no tempo de convergência da rede após uma falha, como a queda de um link ativo, ou alguma outra alteração, como a adição de um *switch* na rede.

De forma global, o *Spanning Tree Protocol* apresenta 3 temporizadores:

Hello: Define o tempo de envio entre cada BPDU. Padrão: 2 seg. Variação possível: 1-10 seg.

Forward Delay: Define o tempo em que as portas ficam no estado *learning* e *listening*. Padrão: 15 seg. Variação possível: 4-30 seg.

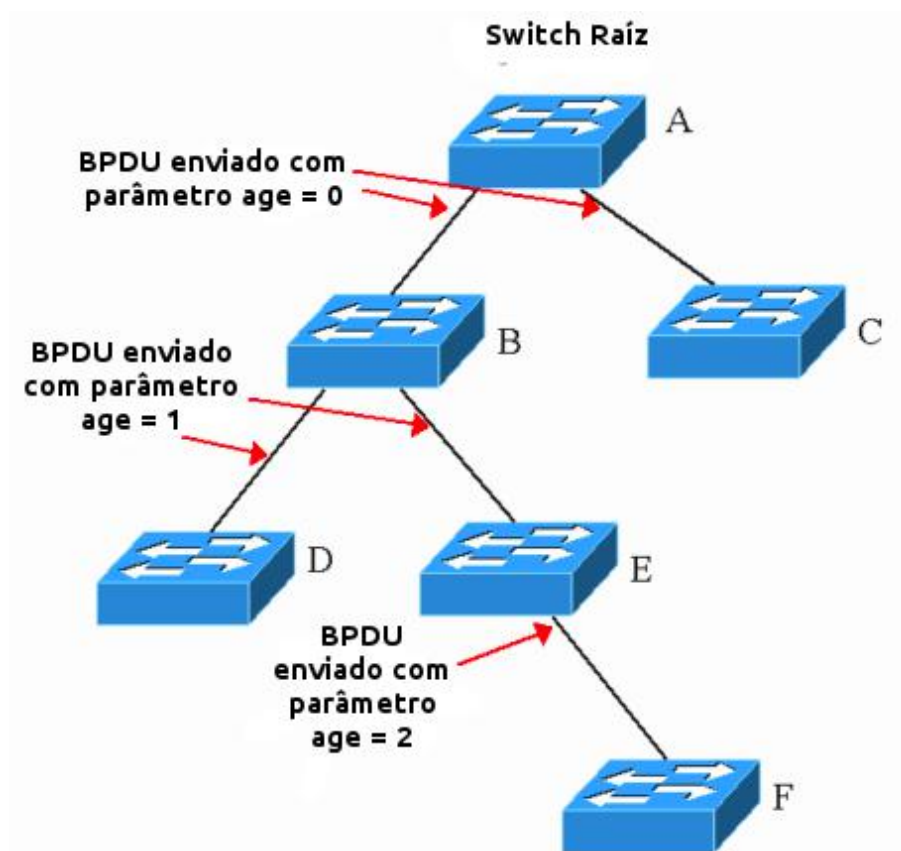
Max Age: Define a quantidade de tempo que uma porta leva para salvar a informação recebida por uma BPDU de configuração. Padrão: 20 seg. Variação possível: 6-40 seg.

Cada BPDU de configuração contém como parâmetro o campo *message age* (idade da mensagem). Esse campo é incremental, e contém a quantidade de tempo (medida em saltos) que passou desde que o *switch* raiz originou a mensagem. Cada *switch* pelo qual a mensagem passa, incrementa seu valor, como mostrado na Figura 9.

Dessa forma, quanto mais longe o *switch* está da raiz, mais rápida a informação do BPDU recebido expira. Quando um novo BPDU de configuração é recebido, caso ele indique um custo menor até a raiz ou uma raiz com um BID menor do que a atual, todas as informações contidas no novo BPDU são armazenadas, enquanto as informações anteriores são descartadas.

Como se pode observar, o STP apresenta um considerável tempo de convergência quando há alguma alteração na rede, e também inutiliza caminhos redundantes, tornando nulo o balanceamento de carga, que poderia dividir o tráfego entre os *links* redundantes aproveitando toda a disponibilidade de banda oferecida por ambos. Para minimizar este impacto, foram elaboradas algumas alterações no protocolo.

Figura 9: Uso do campo Idade da Mensagem no STP



Fonte: Cisco TechNotes

Exemplos de protocolos derivados do STP são o RSTP [21] (*Rapid Spanning Tree Protocol*) e o PVST+ [39] (*Per-VLAN Spanning Tree Plus*).

O RSTP visa encurtar o tempo de convergência da rede após uma falha ou alteração. Diferentemente do STP, o RSTP apresenta apenas 3 estados de porta, *Discarding*, *Learning* e *Forwarding* e classifica as portas que no STP seriam portas bloqueadas, como alternativas (*alternate ports*) ou reserva (*backup ports*). Dessa forma ele mantém informações de quais portas seriam melhores substitutas em caso de mudanças, além de passar menos tempo nos estágios de aprendizado. Por outro lado, o PVST+ trabalha melhor a questão do balanceamento de carga e mal aproveitamento de *links* redundantes. Ele gera uma instância de RSTP para cada VLAN, de modo que certas VLANs passam por caminhos que para outras VLANs estariam bloqueados.

3 NOVAS DEMANDAS PARA OS *DATACENTERS*

Este capítulo discorre sobre as mudanças ocorridas no *datacenter* causadas pelas novas demandas computacionais exemplificadas no texto pelos conceitos de *Big Data* e computação em nuvem, além de expor os problemas de limitação de desempenho ocasionados pelo STP.

3.1 MUDANÇAS NO PADRÃO DE TRÁFEGO

As topologias tradicionais de três camadas, predominantes em ambientes corporativos, apresentam tráfego com o comportamento esperado de percorrer as três camadas, camada de acesso à camada de núcleo e vice-versa, por isso recebe o nome de tráfego norte-sul. Essa denominação norte-sul, indicando a direção do tráfego, ilustra bem o que ocorre com um usuário que faz requisições para fora de sua rede, e por utilizar o STP, sem ciclos ou ramificações, este se torna um caminho único para os mesmos dispositivos finais da camada de acesso.

Numa rede de *datacenter*, o tráfego nem sempre percorre o caminho até a borda para atingir o seu destino. A começar com uma possível comparação com a topologia tradicional, na camada de acesso encontramos servidores ao invés de usuários, além de uma infraestrutura de *storage* anexada que pode estar ligada à camada de agregação ou mesmo à camada de acesso. Neste caso, a orientação do tráfego segue um padrão denominado leste-oeste, onde a origem e o destino das requisições se encontram na mesma camada lógica.

Essa mudança está relacionada com a utilização cada vez maior de aplicações computacionais que armazenam e analisam quantidades massivas de dados e a oferta de serviços de infraestrutura virtualizados que, por sua natureza, exigem disponibilidade de recursos de grande porte [23]. Por apresentarem um grande uso e crescimento ainda em curso, dois conceitos serão abordados neste trabalho para ilustrar as novas demandas para a arquitetura de *datacenters* modernos: Big Data e Computação em Nuvem.

3.2 BIG DATA

À coleta e a subsequente análise de qualquer amostra significativamente grande de dados estruturados ou não, que podem conter informações relevantes ou inteligência proveniente de dados de sensores, usuários e máquinas, dá-se o nome de Big Data. O conceito de Big Data leva em consideração a composição de basicamente quatro atributos principais: Variedade, Volume, Velocidade e Veracidade [43, 8].

Variedade: Refere-se a forma com que os dados são apresentados. Incluindo dados semi-estruturados ou ainda completamente não-estruturados, como textos, arquivos de áudio e vídeo, dados apresentados em XML, arquivos de logs, entre outros.

Volume: Bilhões de dispositivos gerando dados a todo momento, somando um volume imenso de dados a serem armazenados e analisados.

Velocidade: Em alguns casos é necessária a análise dos dados em tempo real de modo a maximizar a eficiência da mesma.

Veracidade: Trata-se da incerteza inerente ao dado. O quanto um dado é confiável para ser analisado e computado.

Nesse cenário, os dados e suas estruturas são fundamentalmente diferentes dos tipos tradicionais, como apontado no quadro na Figura 10. Isso evidencia e corrobora com a ideia de que a infraestrutura e a arquitetura utilizada para manter essas aplicações de análise e coleta também devem ser diferentes [43].

Nesse cenário, Big Data figura entre os principais propulsores para esta mudança de arquitetura. A medida que o tráfego se expande horizontalmente (com origem e destino na mesma camada) e se torna cada vez mais distribuído, o tráfego entre servidores e *storage* se torna significativamente maior do que o tráfego entre servidores e usuários finais. Implementar *datacenters* orientados a prover um ambiente com alta oferta de banda otimizada para tráfego leste-oeste é crucial para desenvolver cenários escaláveis e de alto desempenho para Big Data.

As Figuras 11 e 12 ilustram uma comparação entre a forma com que a capaci-

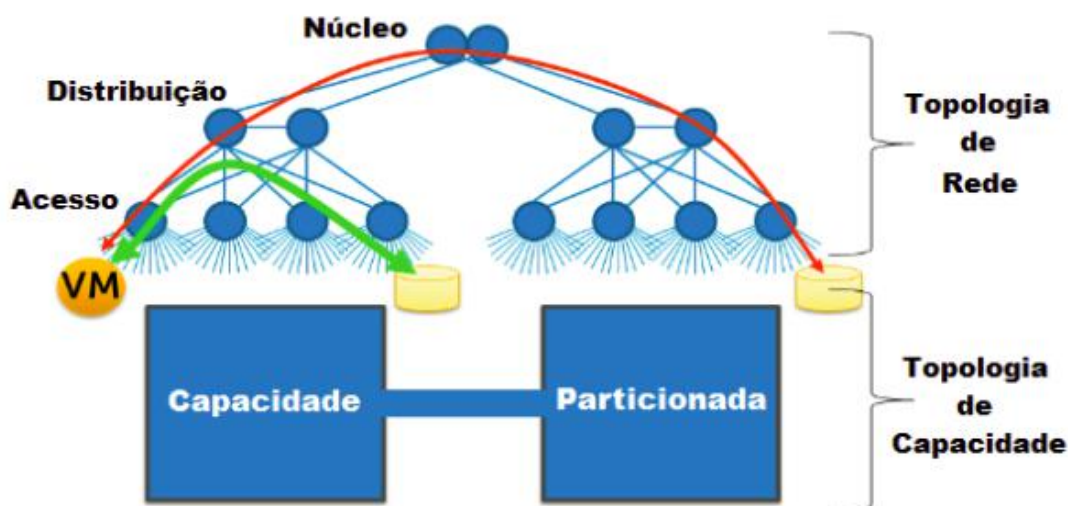
Figura 10: Quadro Comparativo entre Big Data e Dados Tradicionais

Componentes	Dados tradicionais	Big Data
Arquitetura	Centralizada	Distribuída
Volume de Dados	Terabytes	Petabytes / Hexabytes
Tipo de Dado	Estruturado / Transacional	Não-estruturado / Semiestruturado
Relacionamento entre Dados	Relações Conhecidas	Relações Complexas / Desconhecidas
Modelo de Dados	Esquema Fixo	Sem Esquema

Fonte: *International Journal of Engineering Research and Technology* (IJERT)

dade de recursos (em oferta de banda) é seccionada no modelo convencional de três camadas e no modelo topológico *spine-leaf*. Este último é um modelo utilizado para favorecer um padrão de tráfego predominantemente leste-oeste e será apresentado com mais detalhes no Capítulo 4.

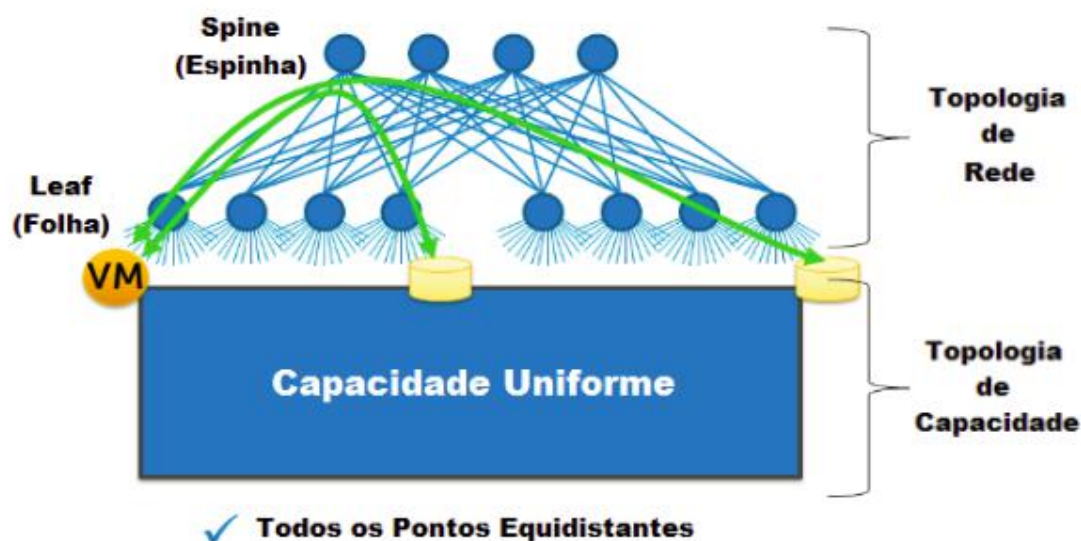
Figura 11: Capacidade de Recursos no Modelo Tradicional



Fonte: DELL - Brad Hedlund - Interop Maio 2012

Utilizado amplamente em aplicações de Big Data, temos o MapReduce [10], um modelo de programação desenvolvido para processar grandes quantidades de dados utilizando algoritmos de execução paralela. O sistema orquestra um processamento entre servidores, gerenciando a comunicação e a transferência de dados entre eles com

Figura 12: Capacidade de Recursos no Modelo Spine Leaf



Fonte: DELL - Brad Hedlund - Interop Maio 2012

redundância e tolerância a falhas. A comunicação entre os nós é predominantemente leste-oeste. Dentre suas principais implementações está o Hadoop [38] que embora não dependa exclusivamente de uma infraestrutura específica, apresenta os seguintes requisitos de rede:

Localidade de dados: As operações de agrupamento e ordenação entre os nós distribuídos, que executam tarefas paralelas, aumentam o tráfego leste-oeste e podem ser gravemente afetadas pela subutilização dos *links* de rede. A infraestrutura de rede deve prover alta disponibilidade de banda, baixa latência, e conexão de qualquer nó para qualquer outro.

Escalabilidade: Implementações iniciais podem começar com pequenos *clusters* e, então expandirem conforme a necessidade. Esse processo deve ser realizado com o menor impacto possível e prover desempenho satisfatório.

Otimização de tráfego leste-oeste: Os padrões de tráfego de comunicação entre os nós variam em questão de relacionamento, como um para um, um para muitos e muitos para muitos, onde esses nós estão topologicamente distribuídos de maneira horizontal. A eficiência dessa comunicação, depende de uma alta disponibilidade de banda e baixa latência entre os nós.

Complementar ao conceito de Big Data, o conceito de Internet das Coisas nos permite interpretar que qualquer objeto é uma fonte de dados em potencial, e que para a realização desta visão, recursos seguros, escaláveis e eficientes de computação e armazenamento de dados são essenciais [18].

3.3 COMPUTAÇÃO EM NUVEM

Computação em Nuvem é um paradigma computacional cuja função é prover acesso sob demanda a um conjunto de recursos computacionais incluindo: redes, servidores, *storage* e serviços [27]. A ideia é que esses recursos possam ser provisionados em quantidade específica utilizando o mínimo esforço em termos de gerência e em interação com o provedor de serviços. A computação em nuvem apresenta cinco características essenciais:

Auto-provisionamento de serviços sob demanda: o usuário pode, de maneira unilateral, provisionar recursos computacionais conforme sua necessidade.

Amplo acesso de rede: As funcionalidades e recursos estão disponíveis para o usuário através de múltiplas plataformas.

Agrupamento de recursos: Os recursos computacionais disponíveis podem ser segregados e oferecidos a múltiplos usuários de forma independente, de acordo com a demanda e a necessidade de cada usuário.

Elasticidade de recursos: Recursos podem ser provisionados e disponibilizados de forma elástica e automática. Dessa maneira o usuário pode rapidamente alocar recursos a medida que sua necessidade aumenta.

Medição do uso de recursos: Sistemas em nuvem automaticamente controlam e otimizam o uso de recursos através de monitoramento, medições e relatórios.

Basicamente, para uma arquitetura de rede prover serviços de computação em nuvem, ela deve atender aos seguintes objetivos:

Capacidade alta e uniforme: O fluxo de tráfego entre os servidores deve

ter sempre a banda máxima disponível, sendo limitado apenas pela capacidade da interface física de rede. Serviços podem ser associados e reassociados a qualquer servidor independentemente da topologia.

Livre migração de máquinas virtuais: Possibilita migrar o estado de uma máquina virtual para outro servidor físico na rede, permitindo que seja feito dinamicamente o balanceamento de carga de recursos, como processamento e memória, e distribuição eficiente de energia elétrica e dissipação de calor.

Resiliência: A medida que cresce o número de dispositivos, conexões e recursos, cresce também a probabilidade de falhas. Para que as comunicações e o funcionamento não sejam afetados, exige-se um ambiente tolerante à falhas de diversos tipos como indisponibilidade de servidores e quedas de links.

Escalabilidade: A infraestrutura de rede pode ser expandida em termos de recursos de rede e computacionais.

Compatibilidade com recursos anteriores: A arquitetura deve ser compatível com dispositivos e tecnologias legadas, permitindo a convergência com a arquitetura moderna.

Dessa forma, fica clara a necessidade de uma mudança infraestrutural de forma que seja possível suprir a demanda e atender os requisitos dessas novas tendências tecnológicas.

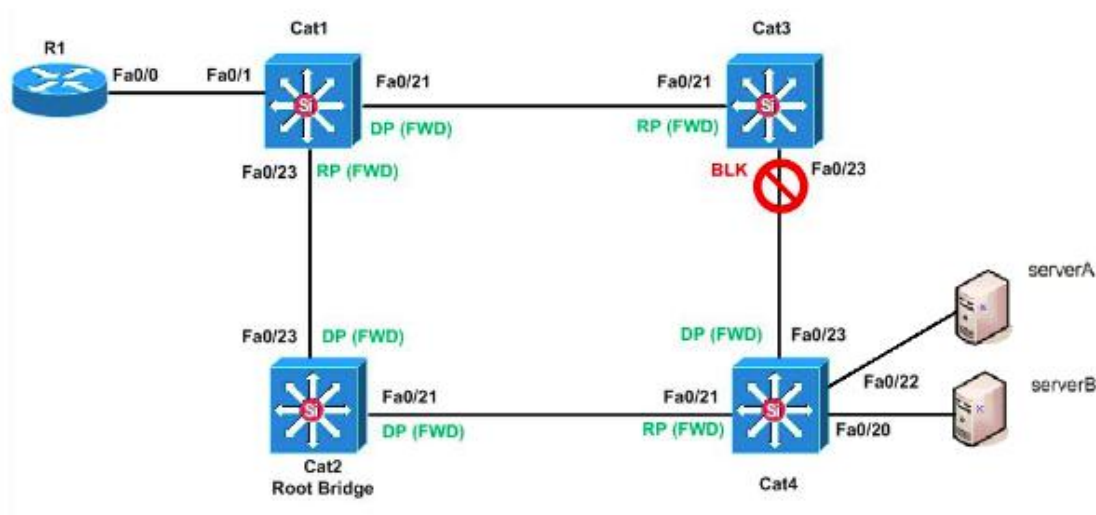
3.4 PROBLEMAS COM O SPANNING TREE PROTOCOL

Pelo exposto acima, são requisitos desejáveis para um *datacenter* moderno o total aproveitamento da banda disponível e a menor latência possível de comunicação. Sendo assim, o STP, que foi elaborado para solucionar problemas causados por ciclos em camada dois, nesse novo contexto, acaba gerando comportamentos antagônicos aos requisitos citados anteriormente.

O primeiro item a ser analisado diz respeito ao uso ineficiente de banda, justamente por impedir que seja utilizada a banda máxima através do bloqueio de

caminhos redundantes, como exemplificado na Figura 13.

Figura 13: Bloqueio de Caminhos Redundantes no STP



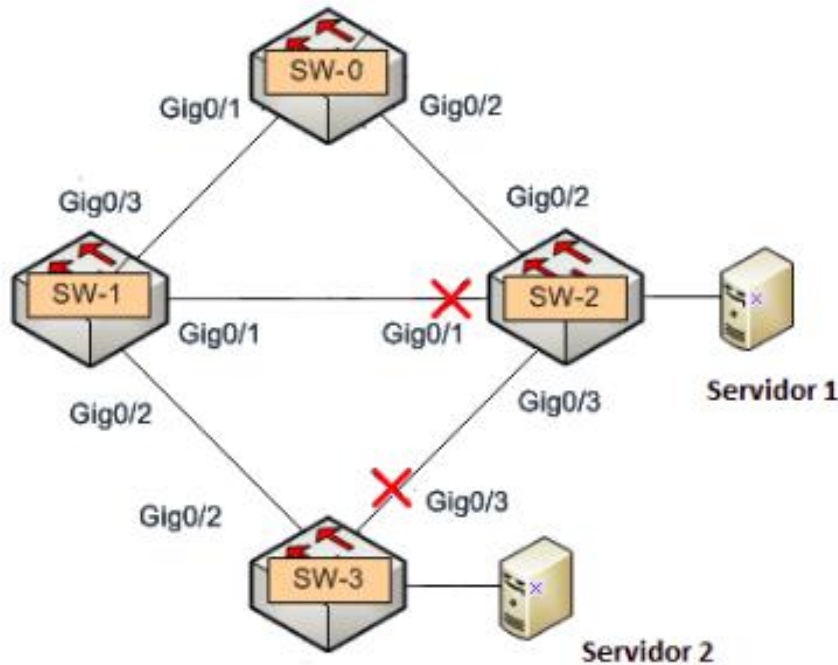
Fonte: Cisco CCNA *Study Material*

Tendo ambos os servidores enviando dados na taxa máxima de suas conexões com o *switch* Cat4, em uma situação onde todos os links estivessem disponíveis, o tráfego gerado seria balanceado pelas rotas Cat4-Cat2-Cat1 e Cat4-Cat3-Cat1 até o roteador R1. Entretanto, com o uso do STP a segunda rota acaba sendo bloqueada, limitando todo o tráfego à banda disponível pela primeira rota. Nessa situação caracteriza-se o uso ineficiente da banda por conta do STP. Eventualmente, com o uso do STP, o caminho percorrido na comunicação entre dois nós pode não ser o ótimo, como ilustrado na ocasião representada na Figura 14.

Tomando o *switch* SW-0 como raiz e tendo que todos os links tem custos iguais, foram bloqueadas as portas marcadas, inutilizando os caminhos entre os *switches* SW-1/SW-2 e entre SW-2/SW-3. Dessa forma, com um tráfego de migração de máquinas virtuais entre o Servidor 1 e o Servidor 2, por exemplo, os dados que poderiam trafegar de um servidor para o outro utilizando a conexão disponível entre os *switches* SW-2 e SW-3 agora precisam seguir pelo caminho SW-2 / SW-0 / SW-1 / SW-3 com custo três vezes maior, caracterizando assim, o uso de um caminho sub-ótimo para um tráfego entre nós da rede.

Em uma rede convencional, esse impacto pode ser considerado pequeno, pois a

Figura 14: Utilização de Caminhos não-ótimos com o STP



Fonte: Cisco CCNA *Study Material*

comunicação entre os nós é incomum. Porém como já explicado antes, numa rede de *datacenter* moderna a comunicação é predominantemente leste-oeste, tornando o impacto dessa utilização relativamente maior.

Outra característica do STP que afeta diretamente a eficiência de um *datacenter* é o tempo de convergência da rede nos casos de falhas e/ou alterações na topologia. Esse tempo é determinado pelas mudanças de estado das portas até estas se tornarem operacionais. Mesmo com a evolução para o RSTP que resulta em um tempo de convergência menor em comparação com o STP, a mudança apenas minimiza o impacto, podendo ser desprezível a nível de usuário, porém desastroso a nível de comunicação entre servidores.

4 PROPOSTAS PARA ATENDER ÀS NOVAS DEMANDAS DOS *DATACENTERS*

Neste capítulo, serão abordadas algumas soluções para problemas que surgem com as novas demandas para redes de *datacenter*. O foco será no padrão apresentado pelo IETF, chamado TRILL (*Transparent Interconnect of Lots of Links*), sua proposta e modo de funcionamento; e no protocolo de roteamento utilizado como base para o plano de controle de implementações de domínios com o TRILL, chamado IS-IS (*Intermediate System to Intermediate System*).

4.1 RECURSOS DE SUPORTE ÀS PROPOSTAS

Diferentes abordagens foram desenvolvidas para lidar com as necessidades de maior disponibilidade de banda, redundância e tolerância à falhas dos *datacenters* modernos. Alguns recursos foram empregados com o objetivo de contornar os problemas inerentes à utilização do STP como a limitação de banda através do bloqueio de caminhos redundantes. A agregação de *links* físicos em *links* lógicos transparentes ao STP é um desses recursos. Com múltiplos caminhos disponíveis para um mesmo destino, faz-se necessária uma técnica que permite rotear o tráfego de dados por entre eles. Neste contexto o ECMP [20] é apresentado como um recurso que atende à esse requisito.

Referente à arquitetura de redes, enquanto *datacenters* tradicionais utilizavam arquiteturas baseadas em topologias de 3 camadas, como visto em sessões anteriores, para lidar com um tráfego basicamente orientado ao fluxo norte-sul, a transição para o tráfego predominantemente leste-oeste revelou uma ineficiência em lidar com essas novas necessidades. Além das necessidades de escalabilidade, mobilidade (cargas de trabalho de virtualização que se deslocam entre servidores) e convergência para protocolos de armazenamento em redes, que necessitam estar no mesmo segmento de rede e se mostraram cada vez mais requeridas. Assim sendo, como uma proposta de mudança na organização topológica da rede, é apresentado o modelo *spine-leaf*.

4.1.1 Arquitetura *Spine-Leaf*

A arquitetura *spine-leaf* passou a oferecer uma alternativa, no âmbito de desenho e projeto de redes para os *datacenters* modernos. Nesta arquitetura os *switches* mais próximos dos servidores, se encontram na camada *leaf* e são chamados de *switches leaf*, enquanto a camada que conecta todos os *switches* da camada *leaf*, agindo como *backbone* desta camada, se chama *spine*. Todos os *switches leaf* se conectam a todos os *switches spine* e os dispositivos na mesma camada não se interconectam, como ilustrado na Figura 15. Desta forma, equipamentos conectados a essa malha estão equidistantes em quantidades de saltos, o que permite alcançar latências conhecidas e, em conjunto com protocolos implementando ECMP, melhor distribuição de cargas nas interconexões e oferta de banda [40].

Figura 15: Arquitetura *Spine-Leaf*



4.1.2 Agregação de Links

Como destacado anteriormente, cada vez mais a demanda por banda aumenta, e as vezes a banda necessária entre dois equipamentos é maior do que a fornecida por uma única conexão entre eles. Logo, surge a necessidade de se utilizar mais de uma conexão de maneira agregada entre os dois equipamentos. Para isso surgem padrões de agregação de *links* como o IEEE LAG [36] (*Link Aggregation*) e o Cisco EtherChannel / PortChannel [7].

A função desses protocolos é criar um *link* lógico através do agrupamento de *links* físicos. Com isso, é possível disponibilizar uma quantidade maior de banda entre

dois dispositivos diretamente conectados. Isso se deve ao fato do STP atuar em *links* lógicos. Se conectássemos os *links* físicos entre dois equipamentos sem nenhum tipo de abstração, de modo que cada *link* físico correspondesse a um *link* lógico, o STP manteria somente um *link* ativo (encaminhando dados) e bloquearia todos os outros, o que implicaria em limitação da banda disponível entre esses equipamentos.

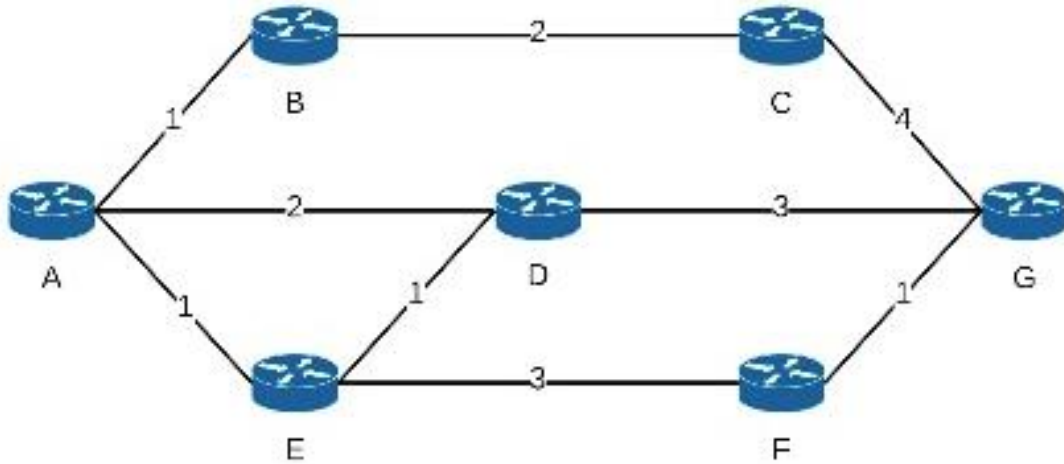
Nesse cenário, o STP continua ativo, o que não impede que *links* sejam bloqueados caso provoquem algum ciclo. Dessa forma, a agregação de *links* resolve apenas parte do problema, sendo uma medida paliativa. O que se busca é uma solução capaz de resolver o que o STP originalmente se propõe a resolver, porém, sem gerar os problemas descritos anteriormente.

4.1.3 ECMP

A necessidade de disponibilizar a maior quantidade de *links* possíveis para encaminhamento de tráfego é uma demanda crucial para uma rede de *datacenter* moderno. Porém, apesar da supressão do STP eliminar a necessidade de bloqueio de links, esta não é uma condição suficiente para atender a todos os requisitos do cenário presente. Um destes requisitos é o do encaminhamento do tráfego através dos *links* disponíveis. Para tanto, uma das estratégias conhecidas é a de múltiplos caminhos de custo equivalente, ou ECMP [20] (*Equal Cost Multipath*).

Trata-se basicamente da utilização de duas ou mais rotas de mesmo custo disponíveis para encaminhamento a um mesmo destino. Isso permite que sejam usados todos os caminhos com o mesmo custo simultaneamente, de modo que se tenha todos os *links* ativos para tráfego e não apenas um, com os outros de reserva como em um sistema ativo-passivo. O cenário observado na Figura 16, apresenta uma série de caminhos com custos equivalentes entre os roteadores A e G, são eles: A-D-G, A-E-D-G e A-E-F-G, todos com custo 5. Aplicar uma estratégia de ECMP neste cenário consiste em utilizar um algoritmo capaz de balancear a carga, direcionando o tráfego pelos três caminhos de menor custo, escolhendo quando utilizar cada um deles.

Figura 16: Topologia com Caminhos de Custos Diferentes



4.2 TRILL

Um padrão que visa alcançar o objetivo de resolver o que o STP se propõe a resolver, podendo assim suprimi-lo, e, ao mesmo utilizar técnicas de ECMP para encaminhar o tráfego pelos múltiplos caminhos agora disponíveis (antes bloqueados pelo STP) é o TRILL [32]. Para entender melhor a necessidade do TRILL e seu funcionamento, é importante revisitar as características principais das camadas de enlace e de rede.

A camada de enlace, mais conhecida hoje pelo padrão *Ethernet*, foi desenvolvida com o objetivo de sinalizar o início e o fim dos pacotes durante a transmissão do fluxo de bits pela camada física, transportando dados de um dispositivo até outro diretamente conectado à ele. No início, essa comunicação era feita somente ponto-a-ponto, porém, com a evolução e expansão do conceito de redes de áreas locais, o protocolo *Ethernet* surge para possibilitar a conexão de múltiplos dispositivos em meios compartilhados [24, 31]. Embora seja destinado a conectar um número relativamente pequeno de dispositivos, seu endereçamento possui 6 *bytes*, o que permite aproximadamente 2^{48} endereços disponíveis. O motivo para este amplo espaço de endereçamento é evitar a configuração manual e individual de cada dispositivo co-

nectado na rede local, pois todo dispositivo já vem de fábrica com um endereço MAC único atrelado diretamente ao hardware. A primeira metade do endereço é destinado à identificação do fabricante, enquanto a segunda metade é destinada à identificação única do dispositivo, permitindo assim uma das maiores vantagens da camada 2: auto configurabilidade.

Na camada de rede, o protocolo IPv4 é o principal protocolo da suíte TCP/IP, além de ser o mais comum. A função primária do mesmo é a entrega de pacotes entre redes fornecendo um mecanismo de endereçamento hierárquico e único (dentro do domínio da rede) para os nós. Com isso, toda a estrutura do endereçamento fica atrelada à cada rede em si, e não ao dispositivo como na camada de enlace.

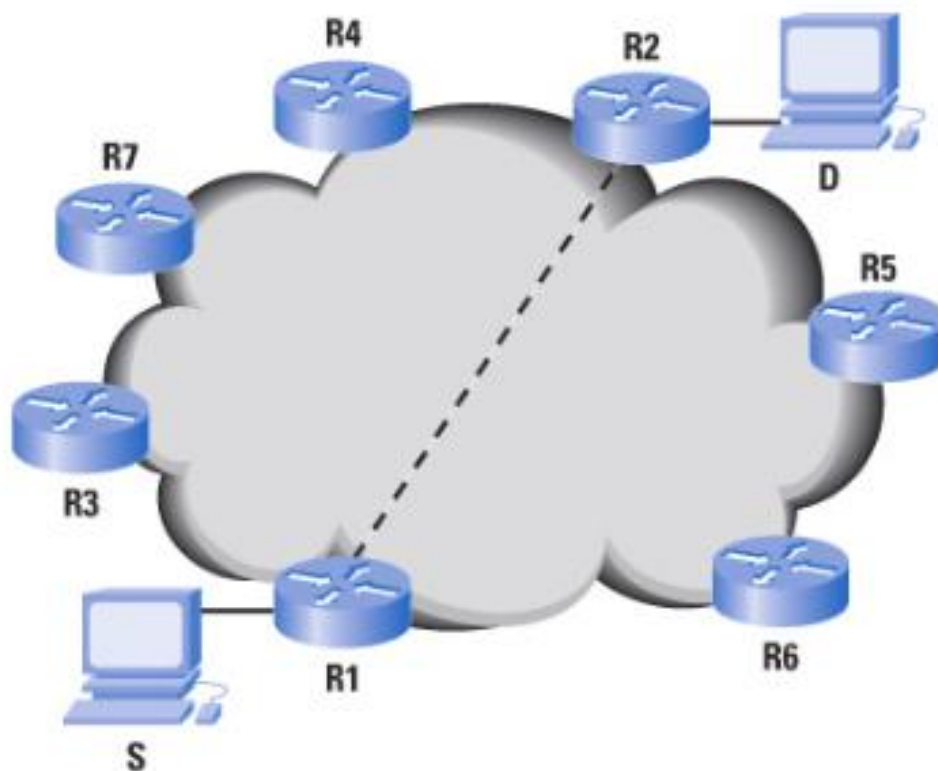
Essas características permitem que, em conjunto com protocolos de roteamento dinâmico, seja possível calcular a melhor rota até o destino mesmo com mudanças no caminho com o tempo, seja por falha de *links* ou por alteração de características como largura de banda e latência. Em contrapartida, isso adiciona uma complexidade de configuração, não só ao dispositivo final, mas também aos roteadores que os conectam à rede.

O padrão TRILL tem como característica fundamental o aproveitamento dos benefícios da camada dois, como a facilidade e a simplicidade de configuração, tanto quanto os benefícios da camada três, como a utilização de técnicas de roteamento. A premissa inicial é substituir, mesmo que gradualmente, os dispositivos que fazem as interconexões dos nós por dispositivos que implementam o padrão TRILL, formando uma nuvem com características de camada dois e três. Esses dispositivos são chamados de *RBridges* (*Routing Bridges*). A possibilidade de coexistência de redes TRILL com redes Ethernet clássicas permite a redução gradual da necessidade do uso do STP.

As *RBridges* operam em cima de um protocolo de roteamento do tipo de estado de enlace (*link state*), no caso do TRILL, o protocolo usado é o IS-IS [37]. Isso fornece o conhecimento completo da topologia que consiste em todas as *RBridges* e todas as conexões entre elas. Usando esse protocolo, cada uma calcula o melhor caminho a partir dela para todas as outras, assim como montam as árvores para

encaminhar tráfego a múltiplos destinos. A Figura 17 mostra o conceito básico do funcionamento do TRILL, onde um fluxo *unicast* é encaminhado de uma fonte a um destino conhecido. Os dispositivos numerados de R1 a R6 representam *RBridges* em uma nuvem TRILL.

Figura 17: Nuvem TRILL



Fonte: Internet Protocol Journal, Vol.14, No.3

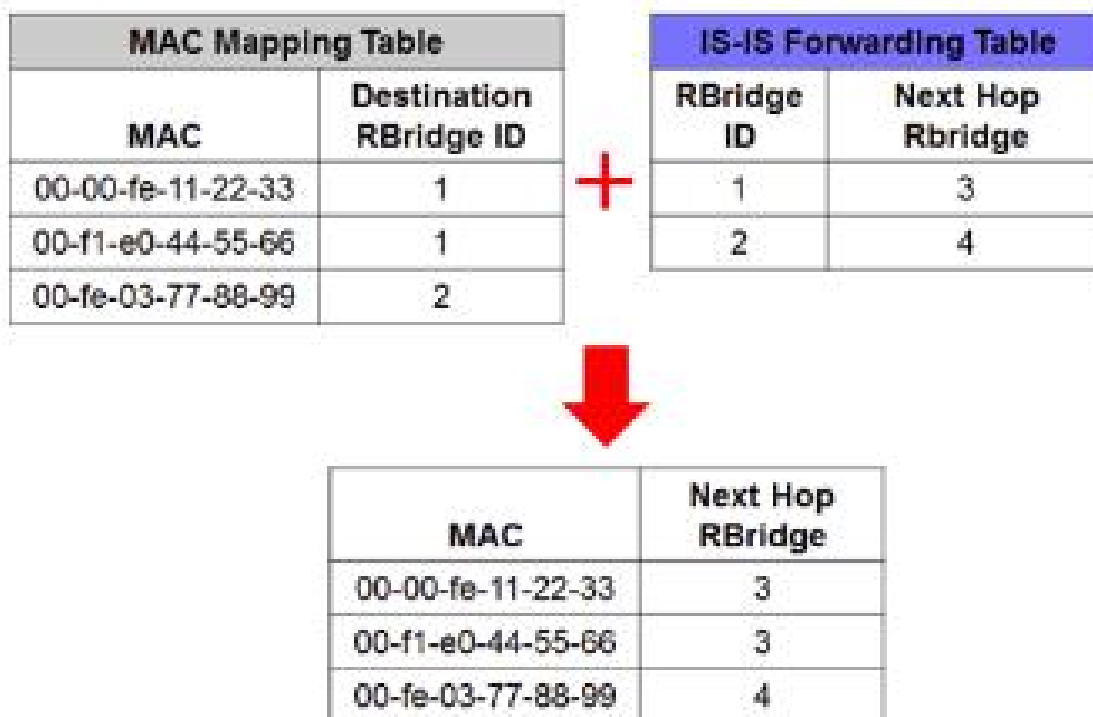
Quando R1 recebe um quadro Ethernet da fonte S, com endereçamento de camada dois para o destino D, ele encapsula o quadro em um cabeçalho TRILL onde o destino passa a ser R2. Quando R2 recebe o pacote encapsulado, remove o cabeçalho do TRILL e continua com o encaminhamento Ethernet clássico para o destino D.

4.2.1 Encaminhamento e Rotas

Quando uma *RBridge* armazena um mapeamento para o MAC de destino, ela o armazena junto com o ID da *RBridge* a qual ele está conectado, como é exemplificado na Figura 18. Esse nível de abstração permite que na ocorrência de uma mudança de

topologia na rede, basta que seja recalculada a árvore de *RBridge* e um novo caminho para as outras *RBridges* que estão referenciadas na tabela de encaminhamento. As entradas de MAC não precisam mudar, uma vez que elas apontam para um ID, além de não estarem diretamente conectadas à árvore.

Figura 18: Tabela de Encaminhamento do TRILL



Fonte: MovingPackets.net

Com o STP, algumas falhas na rede fazem com que a árvore seja recalculada e as tabelas de encaminhamento MAC sejam descartadas. Com o TRILL, quando ocorre uma falha em que a topologia necessita ser recalculada, o mesmo é realizado pelo protocolo de roteamento, mas assumindo que exista um caminho alternativo que leve à *RBridge* de saída, os mapeamentos de MAC não precisam ser descartados. Dessa forma, a inundação de pacotes após uma falha é reduzida e ainda devido a rapidez com que o IS-IS pode recalcular os caminhos na ocorrência de falhas em comparação ao STP, a interrupção do tráfego é minimizada [15].

4.2.2 Cabeçalho TRILL e Encapsulamento

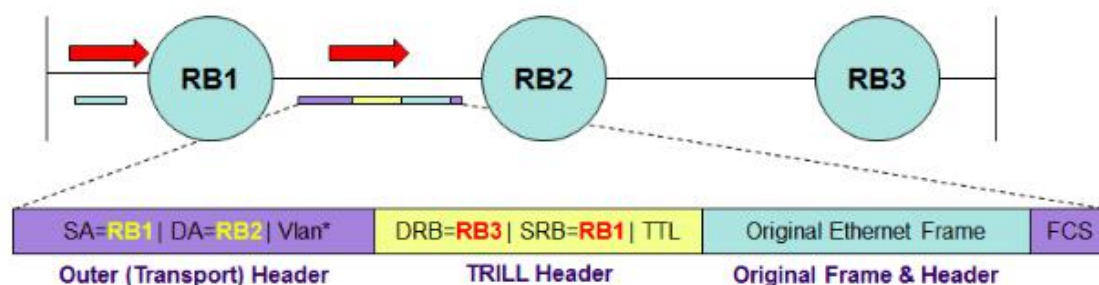
Em se tratando do roteamento dos pacotes, tomando uma situação onde uma *RBridge* precisa enviar tráfego para um determinado destino e, através do mapeamento e da tabela de encaminhamento do IS-IS, ela sabe para qual *RBridge* de egresso encaminhar o tráfego. Entretanto, não estando diretamente conectada a ela, é necessário determinar um caminho até a mesma através da nuvem TRILL. Dessa forma, há duas maneiras de garantir que o tráfego chegue ao destino. Uma forma seria garantir que todas as *RBridges* na rede tenham uma lista completa de mapeamento de MAC. Assim, cada uma poderia tomar sua própria decisão de encaminhamento. O problema dessa solução é que os dispositivos da camada de núcleo poderiam enfrentar problemas com relação ao tamanho das tabelas que precisariam manter, isso entra em conflito com a utilização normal do núcleo.

A alternativa para resolver este problema é encapsular o quadro original em um novo quadro, cujo destino passa a ser justamente a *RBridge* para o qual o tráfego deveria ser encaminhado originalmente. Dessa forma, as *RBridges* intermediárias não necessitam saber o MAC de destino, elas só precisam saber como direcionar o tráfego para a *RBridge* de destino. Isso se torna simples a medida que o IS-IS assegura que qualquer *RBridge* saiba encaminhar tráfego para todas as outras. Percebe-se então que a escolha de “roteamento” será uma decisão a ser tomada salto a salto, porém, a escolha da *RBridge* de egresso é feita pela *RBridge* de ingresso.

A medida que um pacote encapsulado do TRILL atravessa a rede, ele troca o endereço de destino a cada salto, logo, dentro do quadro deve haver uma referência de qual o real destino final do pacote. Isso é feito utilizando-se um cabeçalho externo, identificado como um cabeçalho de transporte, e um cabeçalho do TRILL, inseridos anteriormente ao quadro original, como melhor exemplificado na Figura 19. Nela é possível visualizar em destaque o pacote que vai de RB1 em direção à RB3, passando por RB2 e contendo o cabeçalho original (Original Ethernet Frame) encapsulado entre o cabeçalho TRILL e o FCS (*Frame Check Sequence*). O cabeçalho externo, é preenchido a cada salto (de maneira equivalente a um cabeçalho Ethernet) e contém a identificação de quem está transmitindo o pacote (SA=RB1) e quem está rece-

bendo (DA=RB2). Enquanto que o cabeçalho TRILL, que permanece inalterado durante toda a transmissão pela nuvem TRILL, contém a identificação de quem gerou o pacote inicial (SRB=RB1), ou seja, a origem do dado, e a *RBridge* que é o destino final do pacote (DRB=RB3). Dessa forma, o cabeçalho apresenta tanto características de camada dois, ao referenciar o enlace entre duas *RBridges* junto com um campo de VLAN, no cabeçalho externo; quanto de camada três, referenciando a conexão fim a fim junto com a contagem de saltos (TTL) no cabeçalho TRILL.

Figura 19: Cabeçalho TRILL



Fonte: MovingPackets.net

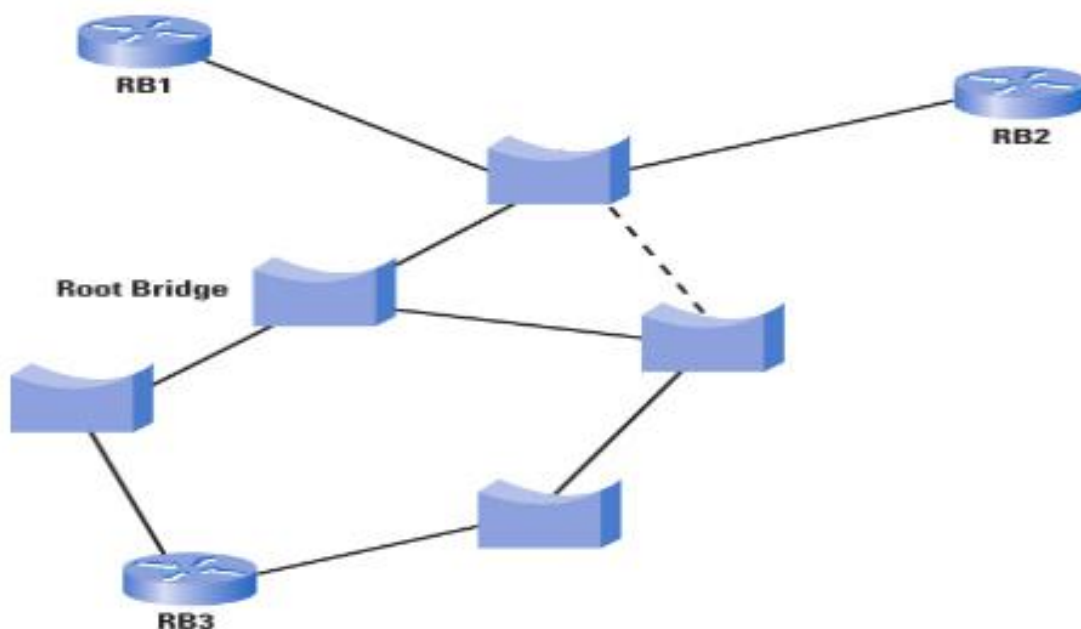
Uma consideração importante com referência ao TRILL, é a forma com que o protocolo popula as tabelas de mapeamento de endereços MAC. O primeiro mecanismo é aprender os endereços a partir de quadros nativos que são recebidos. Assim como *switches* comuns, as *RBridges* vão construir uma tabela MAC baseada nos dispositivos localmente conectados. Além disso, esse aprendizado pode vir também de pacotes TRILL. O cabeçalho do TRILL inclui a *RBridge* de ingresso e a *RBridge* de egresso, dessa forma, é possível extrair junto com a primeira, o endereço MAC de origem e assim adicionar um mapeamento para esse endereço apontando de volta para a *RBridge* de ingresso. Isso vai resultar tanto na capacidade de evitar que o tráfego se disperse pela rede inicialmente, quanto na garantia de tornar o tráfego simétrico com relação às *RBridges* de entrada e saída.

4.2.3 Topologias Mistas

Como discutido anteriormente, uma importante característica do TRILL é a possibilidade de *RBridges* coexistirem com *switches* em uma rede Ethernet clássica.

Dessa forma, o TRILL é desenhado para que qualquer conjunto de *switches* possa ser substituído por *RBridges*. Para isso, cada conjunto isolado de *switches* regulares passa a ser visto pelas *RBridges* como um único *link* que as conectam entre si. Os *switches* desses conjuntos, se comportam regularmente, formando uma árvore de dispersão (*spanning tree*) por onde encaminham o tráfego, como pode ser observado na Figura 20, onde a linha tracejada representa um *link* suspenso pela instância local do STP.

Figura 20: Topologia Mista de Redes TRILL e STP

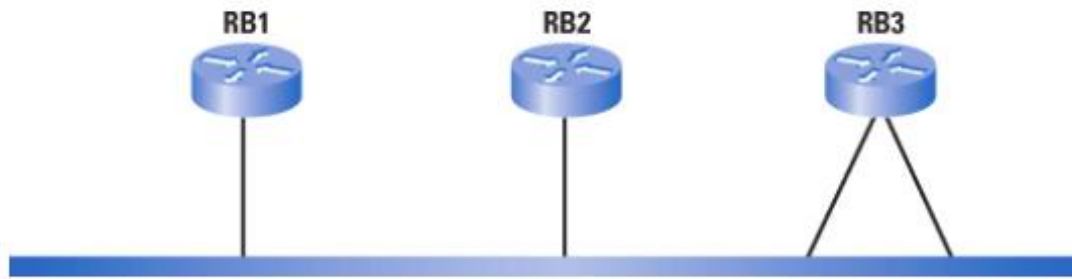


Fonte: Internet Protocol Journal, Vol.14, No.3

Em outra perspectiva, a Figura 21 representa a topologia que é enxergada pelas *RBridges*, onde somente elas estão presentes na rede, porém, conectadas através de um único *link* compartilhado.

Por fim, para complementar o entendimento do funcionamento do TRILL, é necessária uma análise mais minuciosa do protocolo utilizado para seu roteamento interno, o IS-IS.

Figura 21: Topologia Mista de Redes TRILL e STP vista pelas *RBridges*



Fonte: Internet Protocol Journal, Vol.14, No.3

4.3 IS-IS

O IS-IS (Intermediate System to Intermediate System) [37] é um protocolo de roteamento do tipo estado de enlace (*link state*) desenvolvido como um *interior gateway protocol*, ou seja, para ser implementado dentro de redes em um domínio administrativo, ou sistema autônomo (em contrapartida a um *exterior gateway protocol* que executa roteamento entre domínios administrativos diferentes). O IS-IS foi criado originalmente como um padrão ISO [29] para o *Connectionless Mode Network Protocol* (CLNP) em 1987 e depois estendido pelo IETF para suportar endereços IP em 1990.

Apesar de ter sido criado para a pilha de protocolos da ISO, todas as suas mensagens são extensíveis para outros padrões. Diferentemente do OSPF que foi originalmente criado sobre o IPv4, o IS-IS trabalha diretamente sobre a camada dois (enlace de dados) e utiliza parâmetros TLV (tipo-comprimento-valor ou *type-length-value*) na construção dos LSPs (*Link State Packets* - Pacotes de Estado de Enlace) utilizado em suas trocas de mensagens, o que garante grande flexibilidade em adaptações para outros protocolos, sem a necessidade de se alterar a sua base de funcionamento.

4.3.1 Mudanças do IS-IS para o TRILL

A RFC 1195 [6] define a utilização do IS-IS para roteamento em redes IP. Isto foi possível através do registro da TLV 128 que contém um conjunto de campos de 12 octetos compatível com o tráfego IP. A utilização de parâmetros TLV torna o protocolo extensível, pois permite que os LSPs transportem diferentes tipos de informação. Outras informações podem ser aninhadas dentro das TLVs a partir de diferentes SubTLVs. As TLVs e SubTLVs suportadas e implementadas no IS-IS para uso no TRILL podem ser encontradas em [12] e [3].

Dos TLVs fornecidos pelo IS-IS para a implementação do TRILL, destacam-se dois principais: *Router Capability* TLV [25] e *MT-Aware Port Capability* TLV [3]. O primeiro permite que a *RBridge* anuncie sua capacidade em toda a área IS-IS. Das Sub-TLVs subjacentes a esta TLV, as de destaque para este trabalho são [16]: *TRILL version*, *Nickname*, *Trees*, grupo de VLAN e vizinho TRILL. Já o segundo, foi criado para tecnologias que utilizam endereçamento a nível de camada de enlace de dados para roteamento, como o próprio TRILL, o FabricPath (descrito brevemente na seção de experimentos), IEE802.1aq [13] e o OTV (*Overlay Transport Virtualization*) [17]. As Sub-TLVs de interesse para este trabalho são: *Special VLAN and Flags*, *Enabled VLANs* e *Appointed Forwarders*.

4.3.2 Pacotes de Controle do TRILL

O TRILL utiliza o IS-IS com extensões desenvolvidas com o objetivo de adaptar sua atuação como protocolo de plano de controle a nível de camada de enlace de dados, de forma separada e independente de sua versão para a camada de rede. Todavia, os pacotes de controle são processados de forma similar, eles não são encaminhados após o recebimento, sendo enviados aos vizinhos e processados localmente. Para manter as características de uma arquitetura de domínios grandes a nível de enlace de redes, todas as *RBridges* participantes de um mesmo domínio TRILL são do tipo de área nível 1 do IS-IS (mesma área). O formato do quadro TRILL que transporta pacote de controle é identificado por seu endereço *multicast* reservado de

destino (ALL-IS-IS-RBridges - 01-80-C2-00-00-41) [32] e por seu etherType 0x22F4 - L2-IS-IS. O pacote contido na carga útil deste quadro mantém a estrutura geral dos PDUs (*Protocol Data Unit*) do IS-IS para outras implementações. Para o TRILL, o IS-IS utiliza principalmente três PDUs: HELLO PDUs, LSPs e CSNPs (*Complete Sequence Numbers* PDU). Cada um desses PDUs transportam os objetos TLV em seu campo correspondente.

4.3.3 Descoberta dos Vizinhos no TRILL

Algumas adaptações foram necessárias de forma que o IS-IS execute a etapa de descoberta de vizinhos, comum aos protocolos de enlace de dados. Basicamente, as interfaces das *RBridges* se conectam de duas formas distintas, *point-to-point* (ponto-a-ponto, ou P2P) se estiverem diretamente conectadas ou como meio compartilhado, no caso de conexões LAN multiacesso. Nesta etapa, como visto anteriormente, pacotes HELLO são enviados por um nó aos seus vizinhos e dependendo do tipo de interface, um pacote correspondente será enviado sendo um P2P HELLO, para interfaces ponto-a-ponto ou TRILL LAN HELLO para interfaces em uma LAN.

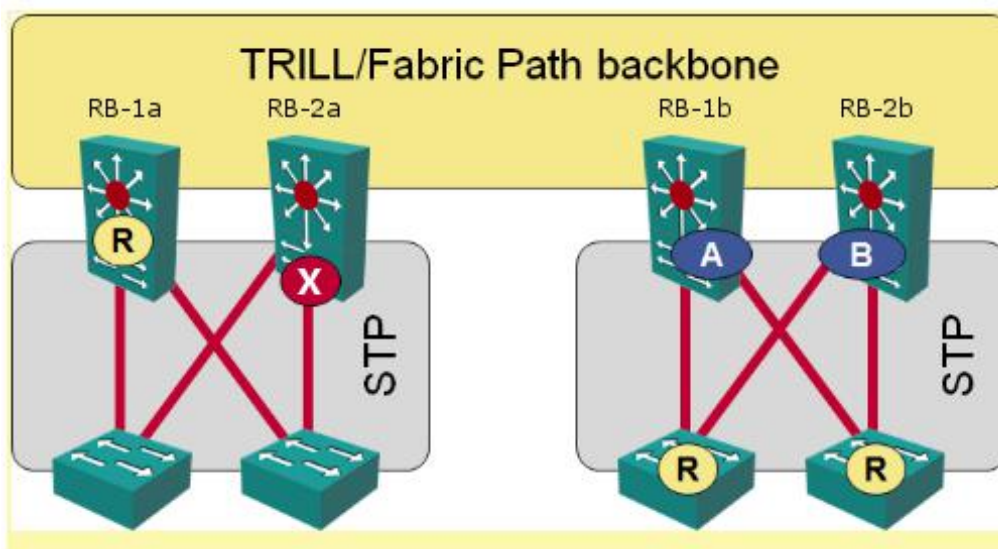
Em LANs, quando um nó é inicializado, um pacote TRILL HELLO é utilizado para descobrir os vizinhos. Este HELLO é enviado para o endereço MAC *multicast* de destino ALL-IS-IS-RBridge que, além de descobrir os vizinhos (respondem ao HELLO com o próprio endereço), permite também fazer a eleição do *Designated RBridge* na LAN e determinar o MTU da comunicação entre os vizinhos.

Uma DRB (*Designated RBridge*) é eleita entre todas as *RBridges* no segmento LAN. Este processo de eleição é baseado somente no endereço MAC e na prioridade do *switch* rodando o IS-IS em questão. Esta prioridade é um parâmetro configurável (na prática, por porta habilitada para o TRILL), por padrão com o valor igual a 64. O maior valor é eleito como DRB, no caso de prioridades iguais, o maior valor de endereço MAC é considerado como critério de desempate. O DRB atende a algumas atribuições importantes para o protocolo, sendo consideradas as mais pertinentes ao presente trabalho: a determinação de uma VLAN para o tráfego de quadros encapsulados TRILL e quadros IS-IS (campo 802.1q do cabeçalho externo

ao TRILL), com exceção de alguns pacotes HELLO; e eleger para cada VLAN ativa no domínio um *appointed forwarder*.

Supondo uma topologia multiacesso com mais de uma *RBridge* presente no domínio TRILL e mais de um equipamento na fronteira com o domínio de rede clássica Ethernet, com o STP em execução, torna-se necessário a eliminação de possíveis ciclos caso haja mais de um ponto de ingresso para as mesmas VLANs. Ou seja, o STP ainda pode bloquear portas de acesso ao domínio TRILL caso não exista um ponto único de acesso, como representado pelo cenário à esquerda ilustrado na Figura 22 onde as VLANs A e B possuem acesso ao domínio TRILL pelos equipamentos RB-1a e RB-2a. Pode-se notar que uma das portas de RB-2a está bloqueada pelo STP, enquanto RB-1a é a Root Bridge do domínio STP. Para evitar este bloqueio, o DRB atua através da designação de VLANs para cada *RBridge* em contato com o domínio STP em questão. Isso garante o único ponto de contato entre os dois domínios por VLAN ativa, como representado pelo cenário à direita ilustrado na Figura 22, onde RB-2a é o *appointed forwarder* designado para a VLAN A e RB-2b para a VLAN B. Os *switches* pertencentes à rede *Ethernet* clássica se tornam as raízes do domínio STP para cada VLAN. Essas *RBridges* designadas pelo DRB são chamadas de *Appointed Forwarders*.

Figura 22: Designação de VLANs para *RBridges* em Topologias Multiacesso



Após a etapa de descoberta dos vizinhos, as *RBridges* iniciam o procedimento de troca de informações sobre a topologia, que no caso do TRILL, não é diferente do padrão implementado pelo IS-IS em redes IP. De posse destas informações cada *RBridge* pode iniciar os cálculos para o caminho mais curto, porém, são necessárias etapas adicionais (em comparação a implementação padrão do IS-IS) que são a resolução dos identificadores das *RBridges* e o cálculo da árvore de distribuição.

4.3.4 Adquirindo Identificadores

Cada *RBridge* seleciona um identificador aleatoriamente, evitando os identificadores em uso no momento, averiguando a própria base de LSPs. Em caso de identificadores iguais o desempate é feito a partir da prioridade e do endereço físico de 6 *bytes*, o dispositivo que possuir o maior valor permanece com o identificador gerado e o outro seleciona um identificador que não consta em sua base de LSPs.

É possível configurar os identificadores das *RBridges* e estes são prioritários com relação aos aleatoriamente escolhidos. Se, por algum engano, nomes idênticos são definidos, o desempate é feito da mesma forma que no caso dos nomes aleatórios.

Estes identificadores servem também como entidades endereçáveis para os nós. Depois que ocorre a troca de informações sobre a topologia, e os nós possuem as informações atualizadas sobre a mesma, a rede pode ser considerada estável. Cada *RBridge* executará o algoritmo de caminho mais curto para todos os nós da topologia e a tabela de roteamento consistirá, em sua forma mais simples, em entradas para um nó de destino (representado por seu identificador) e um conjunto com possíveis próximos saltos e a interface de saída no qual o próximo salto poderá ser alcançado.

Os processos descritos acima são suficientes para cada nó no domínio TRILL calcular o melhor caminho até um outro nó conhecido na topologia, ou seja, para o tráfego *unicast* entre as *RBridges*. Para a entrega de quadros de tráfego multidespacho (*broadcast*, *multicast* e *unicast* desconhecido) o protocolo do plano de controle calcula as árvores de distribuição correspondentes que permitem o encaminhamento desse tráfego, de forma que ele produza um balanceamento na utilização dos *links* em

relação aos já utilizados, por exemplo, no tráfego *unicast*. Essas múltiplas árvores são calculadas usando caminhos mais curtos e critérios determinísticos de desempate, assim, cada nó calcula o mesmo caminho para cada árvore de distribuição. A troca de informação entre os nós permite determinar o nó raiz da topologia que por si, determina a quantidade de árvores a serem calculadas por cada nó e qual o nó raiz de determinada árvore. O tratamento para o tráfego multidestino foge do escopo do presente trabalho, porém, os detalhes podem ser encontrados em [32].

5 EXPERIMENTOS E RESULTADOS

O objetivo principal dos experimentos, a seguir apresentados, é o de obter um comparativo de desempenho e disponibilidade de recursos entre cenários que utilizam o STP e os que utilizam padrões baseados em TRILL. A implementação utilizada do STP foi o *Rapid Spanning Tree Protocol* IEEE 802.1D 2014, enquanto a implementação baseada em TRILL foi o protocolo proprietário FabricPath da Cisco, a ser detalhado posteriormente.

Dessa forma, utilizaremos equipamentos reais em um cenário prático para constatar as vantagens e desvantagens, a eficácia e o impacto, em teoria, pela aplicação de tecnologias baseadas em TRILL contraposta à um cenário tradicional, com STP.

5.1 METODOLOGIA

O ambiente criado para a realização dos experimentos utiliza quatro equipamentos de redes de *datacenter* dispostos de modo a representar um desenho comum e simples, tanto para redes tradicionais quanto em *datacenters* modernos. Essa disposição nos permite realizar experimentos específicos para alcançar os objetivos citados acima para ambos os cenários.

Como explicitado no decorrer do trabalho, as maiores diferenças entre os dois modelos topológicos (três camadas e *spine-leaf*) e a utilização do TRILL se dão em cenários redundantes, que são amplamente utilizados em redes modernas. Dessa forma, todos os objetos do ambiente são dispostos de forma minimamente redundante, ou seja, dois equipamentos na camada de núcleo/*spine*, dois equipamentos na camada de acesso/*leaf*, e dois *links* para cada conexão entre eles. Assim representamos com mais fidelidade modelos atualmente utilizados, além de nos permitir evidenciar com mais precisão as diferenças de comportamento entre os cenários propostos. Este cenário é simplista perto de cenários reais, contudo, considerando o objetivo principal desse experimento, a minimização da complexidade do ambiente é vantajosa para uma análise comparativa. Uma vez que ambientes maiores, acres-

centariam complexidade sem trazer ganhos significativos na obtenção dos resultados.

A escolha dos equipamentos se deu pelo fato de todos serem capazes de implementar ambos os cenários, para que dessa forma a troca de cenário não implique na troca de equipamento, o que dificultaria a análise comparativa dos resultados. Além disso, os equipamentos escolhidos possuem latências de encaminhamento muito baixas no que se refere ao tempo de processamento de pacotes, isto é, o tempo decorrido desde que o pacote é recebido pela porta de entrada, processado no sistema, e encaminhado à porta de saída.

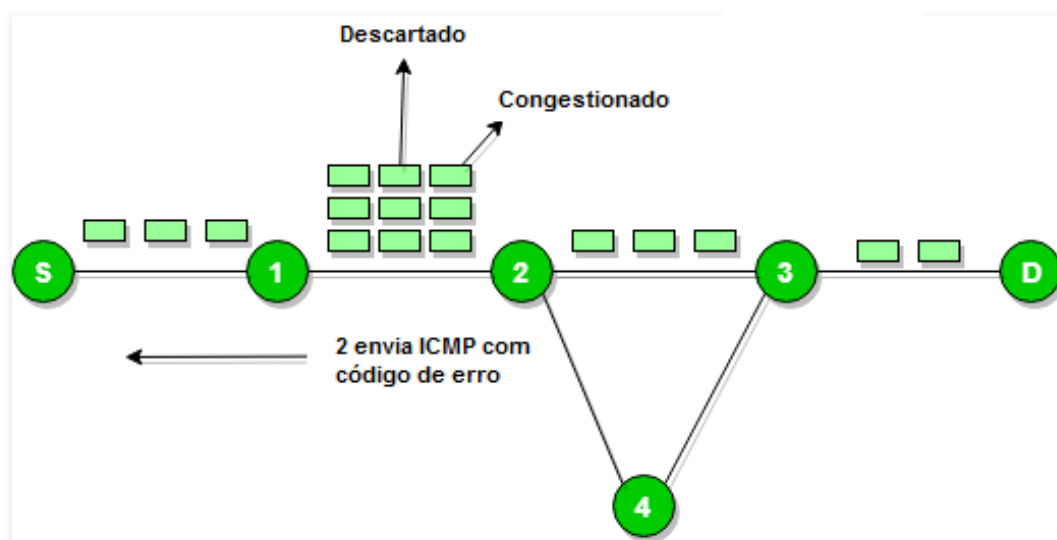
Seria possível replicar os cenários em um ambiente simulado, porém, devido às suas características intrínsecas, a análise em tal ambiente acarretaria em resultados menos próximos do real, uma vez que as instâncias virtuais estariam compartilhando recursos com outras aplicações presentes na plataforma de simulação e qualquer distúrbio externo poderia afetar significativamente os resultados. Logo, optou-se por utilizar equipamentos reais para a composição dos cenários.

Ao longo deste trabalho destacamos algumas características importantes de um *datacenter* moderno, como a ampla disponibilidade necessária dos *links* que compõem a rede, de modo a não impactar o desempenho na transmissão de dados. Foi também explicitado que um dos problemas no uso do *Spanning Tree Protocol* é justamente a necessidade de reconvergência da topologia que o mesmo apresenta quando há determinadas falhas. Este fato está diretamente ligado à questão da disponibilidade dos *links*, uma vez que durante a convergência a rede toda fica indisponível, interrompendo assim o tráfego de dados e impactando diretamente no desempenho da transmissão.

Assim sendo, o objetivo dos nossos experimentos será analisar o impacto tanto na disponibilidade da conexão quanto no desempenho do tráfego de dados. A disponibilidade será analisada através do uso do protocolo ICMP [34] (*Internet Control Message Protocol*). Este protocolo é usado para que os dispositivos da rede possam trocar mensagens com a fonte transmissora com o propósito de relatar erros no processamento de pacotes durante o trajeto dos mesmos até o destino. Dessa forma, quando um pacote não consegue chegar ao destino devido à algum motivo específico

(no caso dos nossos experimentos, em razão da falha provocada na rede), é possível obter essa informação através dos pacotes ICMP de resposta (ICMP *Echo Reply*) que serão enviados ao transmissor. Isso pode ser melhor observado na Figura 23, quando o equipamento 2 percebe um congestionamento e em decorrência disso descarta um pacote, ele envia à fonte transmissora (S) um pacote ICMP informando o descarte.

Figura 23: Troca de Mensagens ICMP



Fonte: GeeksForGeeks.org - ICMP

Como discutido anteriormente, toda a questão de perda de pacotes citada acima está diretamente relacionada com o impacto causado no desempenho do tráfego de dados. Esta é outra característica importante que será analisada em nossos cenários de experimentação, tanto para o ambiente com STP quanto para o baseado em TRILL. Essa análise será feita a partir do monitoramento constante da vazão do tráfego antes, durante e depois do evento de falha na rede ocorrer.

Para a obtenção dos resultados foram coletadas 30 amostras em cada etapa, de cada experimento executado. A partir da média e do desvio padrão obtidos foram calculados os limites superiores e inferiores com base em um intervalo de confiança de 95%. Tanto as métricas quanto as ferramentas utilizadas para a obtenção das mesmas em nossos experimentos são detalhadas na seção a seguir.

5.2 FERRAMENTAS AUXILIARES E MÉTRICAS

Foram utilizadas duas ferramentas para a realização dos experimentos. A primeira é o *ping* disponibilizado nativamente na distribuição Linux usada. A ferramenta consiste no envio e manipulação de pacotes ICMP *Echo Requests*, recebimento de pacotes ICMP *Echo Reply* e da coleta de métricas de interesse na comunicação entre as estações de trabalho, sendo elas o número de pacotes recebidos de volta e o IPG (*Inter-Packet Gap*) que apresenta a relação entre o tempo total de transmissão e número de pacotes enviados. Dessa forma é possível relatar a perda de pacotes experimentada pela rede nos momentos em que há alteração na topologia, sendo analisados ambos os cenários, com STP e com o FabricPath. Perdas de pacotes podem ocorrer por diversas razões como tráfego sendo descartado de forma silenciosa em algum ponto da rede (*blackholed*), congestionamento, hardware defeituoso, entre outras. Nos nossos experimentos, a queda do *link* ocorre a partir de falhas forçadas manualmente através da manipulação dos estados das portas dos equipamentos, isto é, enviando-se comandos para ligar e desligar as portas.

Com relação a execução do *ping*, o comando aplicado em todos os testes e etapas para iniciar a transmissão dos pacotes ICMP foi:

```
1 $ ping <IP> -i 0 -q -c 100000
```

Tanto para os testes com FabricPath quanto com o STP. O parâmetro “-i 0” configura a aplicação para que não espere nenhum tempo pré-determinado para enviar o próximo pacote *echo request*, diferentemente do seu valor padrão de um segundo de intervalo entre cada envio de pacote. Essa opção foi utilizada para que a proporção de pacotes perdidos em relação aos enviados reflita com maior precisão o tempo de indisponibilidade da rede. A opção “-q” é referente à supressão da saída apresentada pelo programa, limitando-se apenas à exibição dos resultados ao final da execução do comando. Por último, o argumento “-c 100000” configura o sistema para que sejam enviados cem mil pacotes seguido do encerramento do comando. Essa quantidade foi escolhida para garantir que o tempo total da transmissão seja maior do que o tempo de convergência da rede após a falha simulada. É importante ressaltar que

em todos os cenários de experimentação, o comando *ping* foi executado antes da coleta de dados para que a tabela ARP na fonte transmissora pudesse ser devidamente preenchida, não interferindo assim, nos resultados obtidos.

A segunda ferramenta é o Netperf [22]. Esta é uma ferramenta que pode ser utilizada como referência para medir diversos aspectos de desempenho de uma rede de computadores. O foco principal da ferramenta consiste na transferência de dados em massa e nos testes de desempenho com requisições e respostas utilizando tanto o TCP quanto o UDP como protocolos de transporte. O modelo cliente-servidor é utilizado, onde um executável `netserver` é invocado a partir do `inetd` da máquina alvo, quando o executável `netperf` for executado a partir da máquina cliente. Neste momento, a primeira ação do lado cliente é estabelecer uma conexão de controle com a máquina remota onde serão passadas informações referentes às configurações do experimento e os resultados obtidos. Independentemente do experimento a ser realizado, esses dados passam por uma conexão TCP entre cliente e servidor.

Depois de estabelecida a conexão de controle, uma outra conexão, separada, é criada para as medições utilizando as APIs e protocolos apropriados e definidos previamente. Durante os experimentos, nenhum tráfego de controle é trocado entre cliente e servidor.

Utilizamos, para os diferentes cenários propostos (explicados em seguida), os testes de medição com transferências de dados em massa, em um fluxo TCP com o envio de 1GB de dados do cliente para o servidor obtendo a média do *throughput* alcançado durante a realização da transferência. O *throughput* é medido a partir da equação 5.1 abaixo.

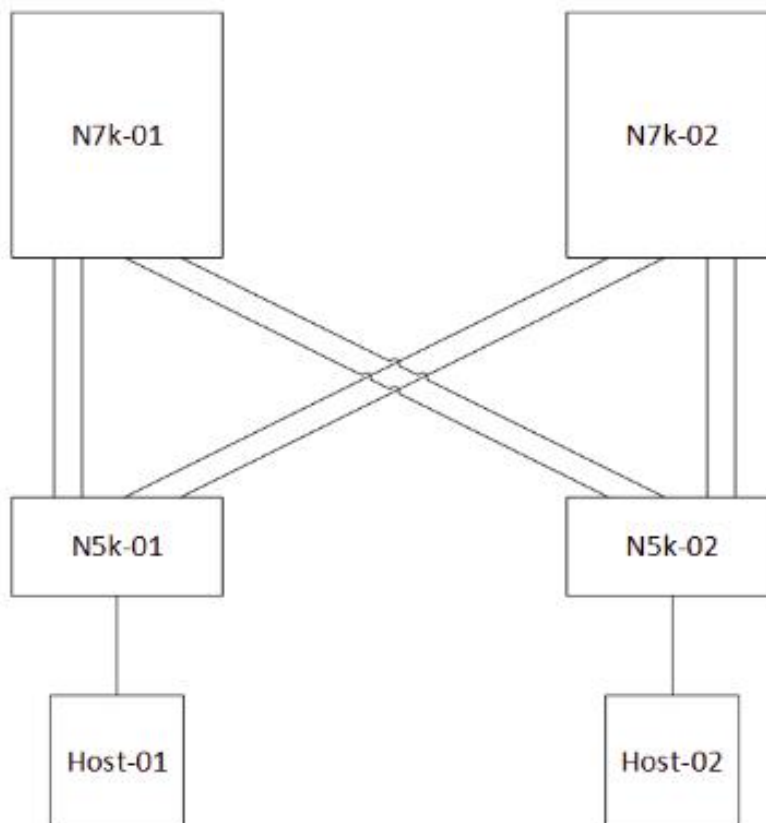
$$Throughput \leq \frac{RcvdWin}{RTT} \quad (5.1)$$

Onde RcvdWin (TCP Received Window) é o tamanho da janela TCP, ou seja, a quantidade de segmentos recebidos pelo destinatário, e RTT é o *Round Trip Time*.

5.3 CENÁRIOS DE EXPERIMENTAÇÃO

Os equipamentos utilizados foram: dois Nexus 7010 (N7k-01 e N7k-02), que foram designados para compor a camada *spine* quando a rede é baseada em TRILL e a camada de core quando a rede utiliza o STP; e dois Nexus 5548UP (N5k-01 e N5k-02) como *switches* de *leaf* e acesso, respectivamente, para TRILL/FP e STP. Estes últimos foram conectados a um único servidor, com duas interfaces físicas distintas, executando duas máquinas virtuais (Host-01 e Host-02) que trocam dados entre si gerando o tráfego necessário para os experimentos. Embora as máquinas virtuais estejam hospedadas em um único servidor, elas possuem suas interfaces virtuais associadas a uma interface física diferente, além de estarem separadas logicamente de modo a permitir que o tráfego entre elas passe pela infraestrutura desejada. A Figura 24 mostra a topologia utilizada para coletar os resultados dos experimentos.

Figura 24: Topologia Utilizada nos Cenários de Experimentação



Os *switches* de núcleo/*spine* possuem sistema operacional NX-OS v6.2(10) en-

quanto os *switches* de *leaf*/acesso estão com o mesmo sistema 7.0(7)N1(1). Ambas as VMs utilizadas possuem sistema operacional Linux Ubuntu 16.04 LTS 4.4.0-32 x86_64.

Em cada um dos cenários, foram realizados três experimentos. Todos consistem na execução de um comando que, por consequência, provoca uma alteração na topologia. Os eventos gerados são apresentados a seguir:

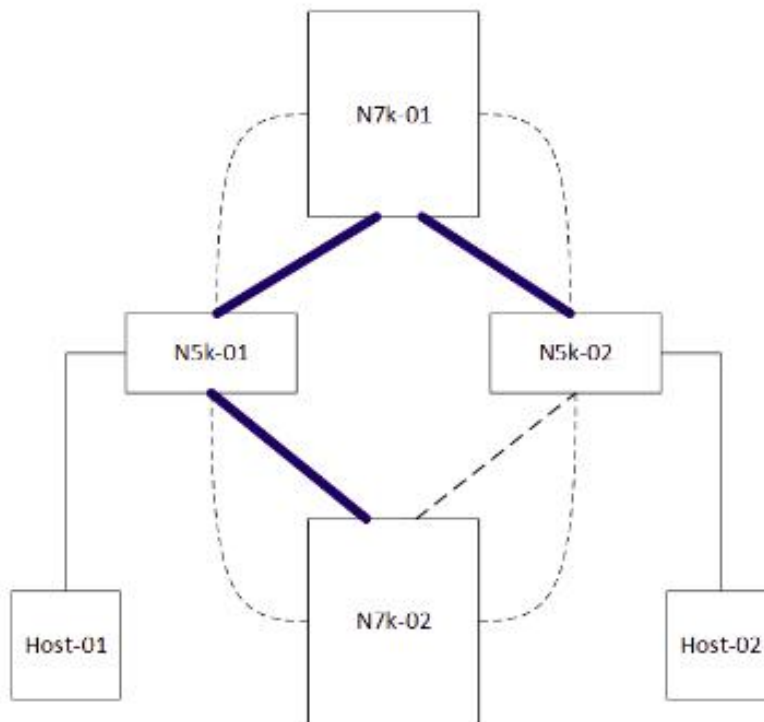
- Experimento 1: Queda, seguida de restauração, de um dos *links* entre o *switch* de core eleito como raiz da árvore de dispersão e o *switch* de acesso ligado ao host-1.
- Experimento 2: Queda, seguida de restauração, de todos os *links* entre o *switch* de core eleito como raiz da árvore de dispersão e o *switch* de acesso ligado ao host-1.
- Experimento 3: Queda, seguida de restauração, de todos os *links* entre o *switch* de core eleito como raiz da árvore de dispersão e todos os *switches* da camada de acesso.

No primeiro experimento temos como objetivo avaliar o comportamento da rede e dos protocolos envolvidos numa situação correspondente à perda e restauração de um único *link*, podendo esta ser gerada a partir de eventos como a falha de uma porta física de um dos equipamentos envolvidos, ou então, a falha de um dos cabos que os conectam fisicamente, entre outras situações. No segundo experimento, a situação já corresponde a perda total da comunicação entre os dois equipamentos, podendo ser originada, por exemplo, na falha de um *patch panel*. O terceiro experimento representa a falha total de um equipamento (o *switch* que é a raiz da árvore) por onde o tráfego passa obrigatoriamente. Este último cenário representa o maior impacto na rede, uma vez que a raiz da árvore fica indisponível e o protocolo é obrigado a eleger uma nova raiz, além de recalcular toda a topologia baseada na nova escolha.

No primeiro cenário, com o STP em execução, o BID *Priority* do primeiro *switch* de núcleo/*spine* (N7k-1) foi manualmente colocado no valor mínimo para que esse

switch fosse necessariamente eleito como a raiz da árvore. Ao eleger a raiz e montar a árvore, a topologia neste cenário passa a ser descrita como na Figura 25.

Figura 25: Topologia dos Testes Após Convergência do STP



As linhas destacadas representam os *links* ativos, onde ambas as portas presentes na conexão estão em estado *Forwarding*. Enquanto as tracejadas representam os *links* alternativos/*backup*, onde pelo menos uma de suas portas encontra-se no estado *Blocked*.

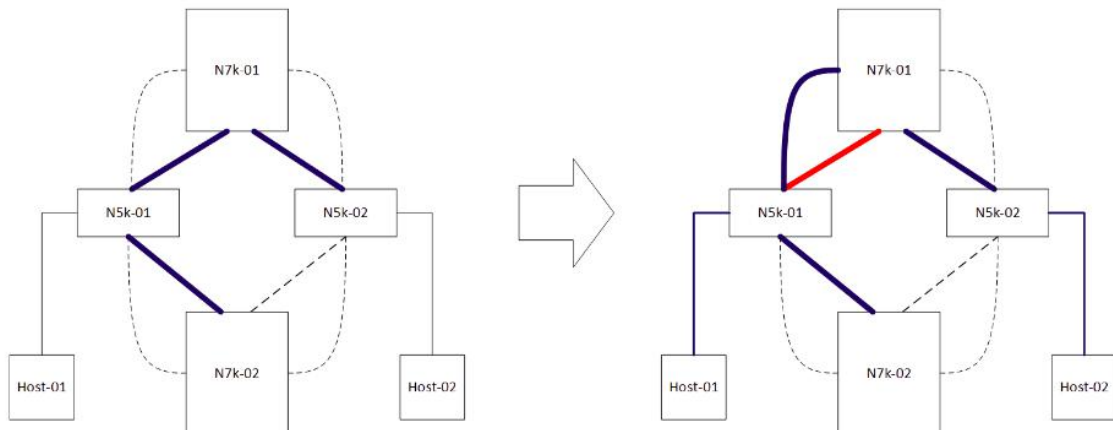
Todos os experimentos foram realizados em duas etapas. A primeira etapa consiste no desligamento manual dos *links* referentes a cada experimento, enquanto a segunda etapa consiste no religamento manual dos mesmos. Cada um dos três experimentos e suas duas etapas gera uma mudança diferente na árvore formada pelo STP. O objetivo é analisar o comportamento e o impacto na rede em cada um desses seis pontos.

5.3.1 Experimento 1

O primeiro experimento, como explicado anteriormente, consiste em realizar o desligamento de um dos *links* que liga o *switch* N5k-1 à raiz N7k-1. O desligamento é feito na porta do *switch* raiz. Nessa etapa o impacto é baixo, pois a instabilidade ocorrerá desde o momento do desligamento, até o momento em que o *switch* de acesso N5k-1 perceber que parou de receber os BPDUs da porta vindo da raiz. Nesse instante, ele muda imediatamente o estado da porta do *link* de *backup* para *Forwarding*. Essa mudança topológica é exibida na Figura 26.

Na segunda etapa deste primeiro experimento, o religamento da porta também irá impactar o tráfego, devido à prioridade da mesma. Assim que o *switch* de acesso voltar a receber os BPDUs da porta principal que o liga à raiz, ele mudará o estado da porta alternativa novamente para *Blocking* e alterará o estado da porta principal de *Blocking* para *Forwarding*, fazendo com que a rota do tráfego seja alterada novamente.

Figura 26: Mudança Topológica no Cenário do Experimento 1

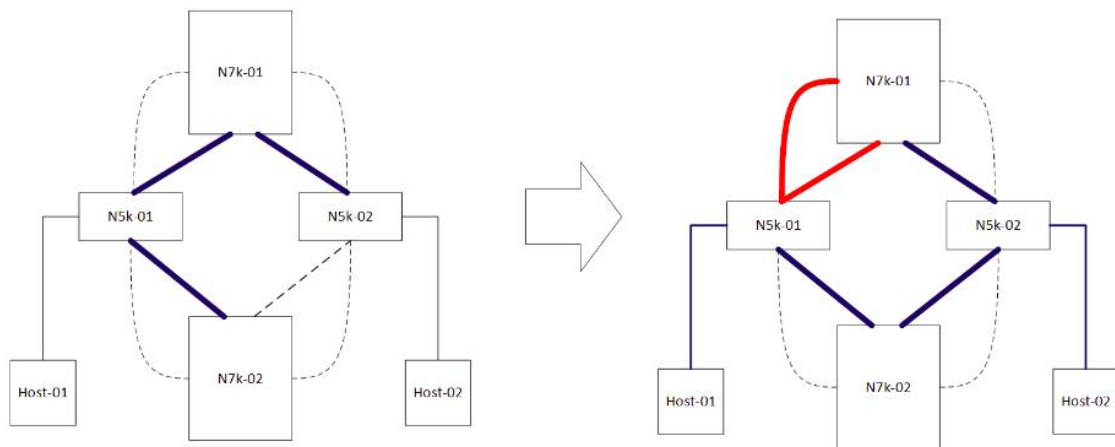


O objetivo desse experimento é analisar o impacto mesmo com uma mudança mínima na rede. A nível de nó, ou seja, de equipamento, a rota do tráfego se mantém a mesma, passando apenas por uma conexão diferente. O nó raiz se mantém o mesmo e a árvore praticamente permanece a mesma.

5.3.2 Experimento 2

No caso do segundo experimento, todas as conexões entre o *switch* de acesso N5k-01 e o nó raiz são desligadas. Dessa forma, o tráfego precisa ser desviado e uma conexão alternativa é habilitada para permitir que qualquer nó da árvore alcance todos os outros, como é mostrado na Figura 27.

Figura 27: Mudança Topológica no Cenário do Experimento 2



Nesse experimento observa-se um impacto maior, pois mais nós da rede participam da alteração. Ao religar os *links* na segunda etapa, um impacto também será gerado, pois o *switch* N7k-1 é a raiz da árvore, que ao recuperar suas conexões com a camada de acesso pelo N5k-1 fará com que outros nós bloqueiem suas portas alterando o caminho do tráfego corrente.

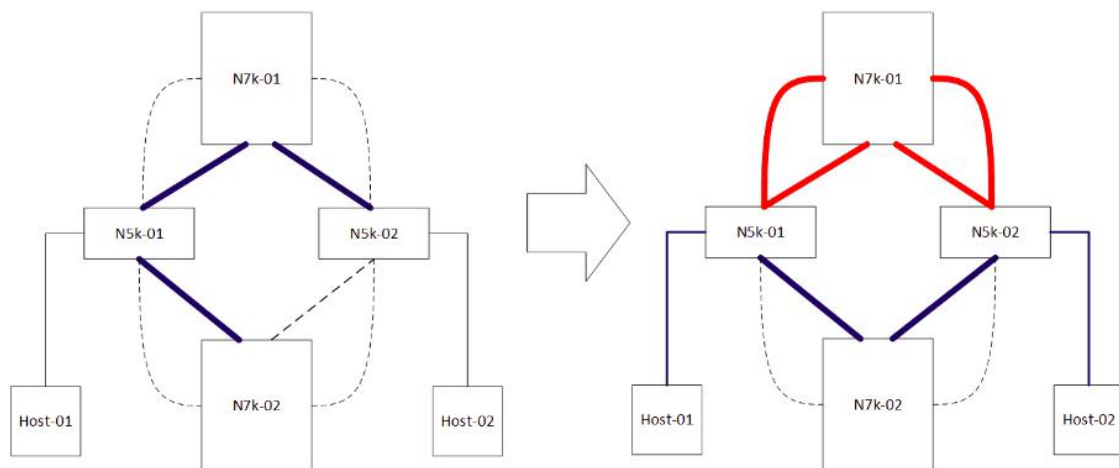
5.3.3 Experimento 3

O terceiro experimento é o que, em teoria, apresenta o maior impacto para a rede. Neste experimento todas as portas do nó raiz são desligadas durante a transmissão de dados entre o host-1 e o host-2. O impacto nesse caso é equivalente à falha geral do equipamento em questão, pois ao desligar todas as suas interfaces conectadas à rede, esse *switch* sai do cálculo da instância atual do STP, exigindo assim que se calcule uma nova árvore com um novo nó raiz.

De maneira análoga aos itens anteriores, o religamento das portas causa impacto semelhante na rede pois, devido à sua prioridade, a partir do momento que o *switch* N7k-1 retorna à árvore, ele assume novamente o papel de nó raiz fazendo com que a estrutura seja recalculada. Por consequência, o caminho utilizado pelo tráfego gerado entre o host-1 e o host-2 é alterado novamente.

A Figura 28 mostra a transformação que a árvore de dispersão sofre à medida que o nó raiz tem suas portas desligadas.

Figura 28: Mudança Topológica no Cenário do Experimento 3



5.4 IMPLEMENTAÇÃO BASEADA NO TRILL: FABRICPATH

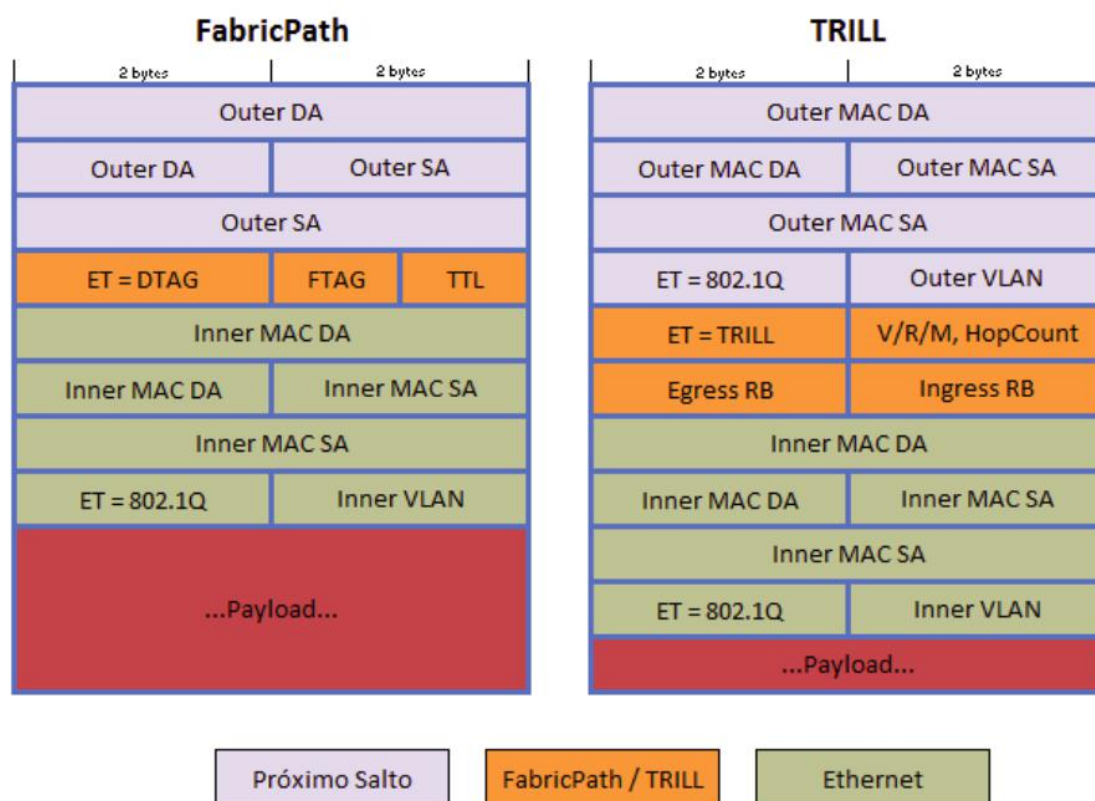
No segundo cenário, o STP não está presente, dando lugar a uma implementação baseada em TRILL. Como o TRILL é um padrão para redes de multicaminhos em camada 2, na prática, existem diferentes implementações com o mesmo propósito. Como foi salientado no início do capítulo, uma dessas implementações é o FabricPath, padrão proprietário da Cisco. Seu plano de controle apresenta algumas diferenças em relação ao padrão do IETF, mas para os fins deste trabalho, a sua utilização como forma de implementação do conceito de roteamento a nível de camada dois é o mesmo.

A diferença entre os protocolos começa na terminologia usada para o dispositivo

de camada dois, onde o TRILL os nomeia como RBridges enquanto o FabricPath os chama de *D-Bridge*. O formato do quadro e seus cabeçalhos também apresentam ligeiras diferenças. Na perspectiva do plano de controle, existem algumas diferenças adicionais em suas funcionalidades. O FabricPath usa um paradigma diferente no que diz respeito à aprendizagem de endereços MAC, onde os *switches* de núcleo não aprendem o MAC do dispositivo final mas sim o ID do *switch* (equivalente ao *RBridge ID*), com o objetivo de simplificar a topologia e acelerar a convergência. Uma outra diferença está na forma de lidar com tráfego de múltiplos destinos, porém, neste trabalho, não utilizamos tráfego de múltiplos destinos para os fins de medição dos parâmetros definidos na proposta dos experimentos.

A Figura 29 mostra a composição dos cabeçalhos do TRILL e do FabricPath, onde é possível visualizar algumas semelhanças e diferenças entre os dois.

Figura 29: Comparação dos Cabeçalhos TRILL e FabricPath



Fonte: Cisco - *Flexible Data Centre Fabric*

A parte inicial do cabeçalho identifica o próximo salto a ser tomado dentro da

rede TRILL/FP, indicando o endereço de destino externo (*Outer DA*) e o endereço de origem externo (*Outer SA*). Todavia, a forma de identificar esses endereços é diferente em cada protocolo. Interfaces configuradas habilitadas para o FabricPath sempre encapsulam quadros Ethernet em quadros com cabeçalhos FabricPath de 16 bytes e os encaminha com base em entradas nas tabelas de identificação de *switches*, não em tabelas de endereço MAC. *Switches* com todas as interfaces habilitadas para o FabricPath encaminham quadros exclusivamente baseados nos endereços de destino externos (*Outer DA*) do cabeçalho utilizado pelo protocolo.

5.5 RESULTADOS

Os resultados completos, apresentando média, desvio padrão e limites do intervalo de confiança calculados para cada experimento realizado são apresentados na Figura 30.

Primeiramente, serão analisados os resultados referentes aos experimentos utilizando os pacotes ICMP *echo reply/request* nos cenários 1 e 2. A Tabela 1 mostra os resultados, em ambos os cenários, referentes ao número de perdas de pacotes durante a transmissão.

	T1E1	T1E2	T2E1	T2E2	T3E1	T3E2
Spanning Tree	31	2495	2566	2503	2716	2580
FabricPath	10	0	10	0	35	71

Tabela 1: Perda de Pacotes Durante a Transmissão

Na Tabela 1, cada uma das colunas está indicada com o número do experimento seguido do número da etapa. Por exemplo, a segunda etapa do experimento três (restauração de todos os *links* do *switch* raiz) é descrita como T3E2. A partir dos resultados obtidos é possível perceber que o impacto na confiabilidade da rede foi consideravelmente maior com o cenário do STP. O impacto real disso no desempenho de uma aplicação vai depender do modo com que seus dados trafegam. Ao utilizar um fluxo TCP por exemplo, a perda de pacotes influencia diretamente na degradação

Figura 30: Quadro de Resultados dos Experimentos

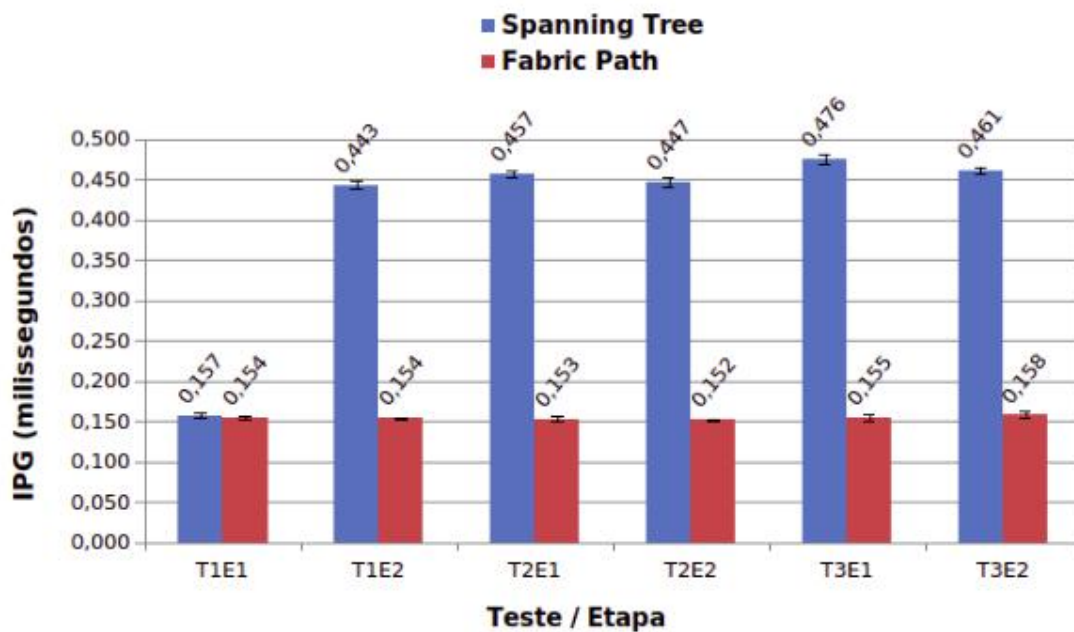
Teste / Etapa	Experimento	Protocolo	Média	Desvio Padrão	Limites
T1E1	Perda de Pacotes	STP	30,5	3,1	29,5 - 31,5
		FP	9,7	1,0	9,4 - 10,0
	IPG	STP	0,157	0,008	0,154 - 0,160
		FP	0,154	0,006	0,152 - 0,156
	Throughput	STP	140,1	0,1	140,1 - 140,1
		FP	918,9	0,4	918,9 - 918,9
T1E2	Perda de Pacotes	STP	2494,0	15,3	2488,6 - 2499,4
		FP	0,0	0,0	0,0 - 0,0
	IPG	STP	0,443	0,015	0,438 - 0,448
		FP	0,154	0,003	0,153 - 0,155
	Throughput	STP	140,1	0,2	140,1 - 140,1
		FP	938,9	0,1	938,9 - 938,9
T2E1	Perda de Pacotes	STP	2564,5	10,5	2560,5 - 2568,5
		FP	9,8	0,7	9,5 - 10,1
	IPG	STP	0,457	0,013	0,452 - 0,462
		FP	0,153	0,008	0,150 - 0,156
	Throughput	STP	140,1	0,1	140,1 - 140,1
		FP	918,5	0,4	918,5 - 918,5
T2E2	Perda de Pacotes	STP	2503,1	15,9	2499,0 - 2507,2
		FP	0,0	0,0	0,0 - 0,0
	IPG	STP	0,447	0,017	0,441 - 0,453
		FP	0,152	0,002	0,151 - 0,153
	Throughput	STP	140,2	0,1	140,2 - 140,2
		FP	938,9	0,0	938,9 - 938,9
T3E1	Perda de Pacotes	STP	2717,4	16,5	2711,8 - 2723,0
		FP	35,0	0,7	34,8 - 35,3
	IPG	STP	0,476	0,017	0,470 - 0,482
		FP	0,155	0,011	0,151 - 0,159
	Throughput	STP	140,1	0,1	140,1 - 140,1
		FP	880,3	0,5	880,3 - 880,3
T3E2	Perda de Pacotes	STP	2578,3	12,7	2573,0 - 2583,6
		FP	70,6	1,1	70,2 - 71,0
	IPG	STP	0,461	0,013	0,456 - 0,466
		FP	0,159	0,012	0,155 - 0,163
	Throughput	STP	140,0	0,1	140,0 - 140,0
		FP	938,9	0,1	938,9 - 938,9

da vazão, tanto com a sobrecarga gerada com a retransmissão dos pacotes, quanto com os cortes na taxa de transmissão devido aos ajustes de janela pelo controle de congestionamento. No caso de uma transmissão sem garantia de entrega como faz

o UDP, a perda de uma quantidade grande e consecutiva de pacotes pode impactar diretamente na estabilidade da aplicação. Com isso o cenário que deve ser ressaltado é o de uma rede de *datacenter*, onde o esperado é alta carga e diversidade de fluxos constantes, aliados a uma enorme utilização de alguns *links* [5].

Uma demonstração mais clara do impacto gerado pelas falhas se dá também no tempo total da transmissão de todos os pacotes. O índice utilizado como referência para essa análise foi o IPG (*Inter-Packet Gap*) que representa o valor médio do tempo gasto (em milissegundos) para transmitir cada pacote. Ele é calculado dividindo-se o tempo total de execução do comando pelo número de pacotes ICMP enviados, no nosso exemplo, cem mil. O resultado obtido é apresentado no gráfico da Figura 31.

Figura 31: Gráfico IPG x Experimento/Etapa



Na segunda etapa de cada experimento, o comando aplicado resulta na restauração dos *links* previamente desligados. No cenário com o uso do STP, a quantidade de pacotes perdidos e o IPG são altos nessa etapa, quase tanto quanto no desligamento dos *links*. Isso se deve ao fato de que ao restaurar os *links*, a rede toda precisa convergir novamente, alterando a árvore de dispersão. Isso faz com que o caminho do tráfego seja alterado, mesmo que os *links* atuais pelo qual o tráfego

está sendo encaminhado não tenham sido afetados. No cenário do FabricPath, essa convergência não afeta o tráfego corrente, exceto no último experimento onde pode ser observada uma perda mínima devido à inserção do novo nó na topologia com a restauração dos *links*.

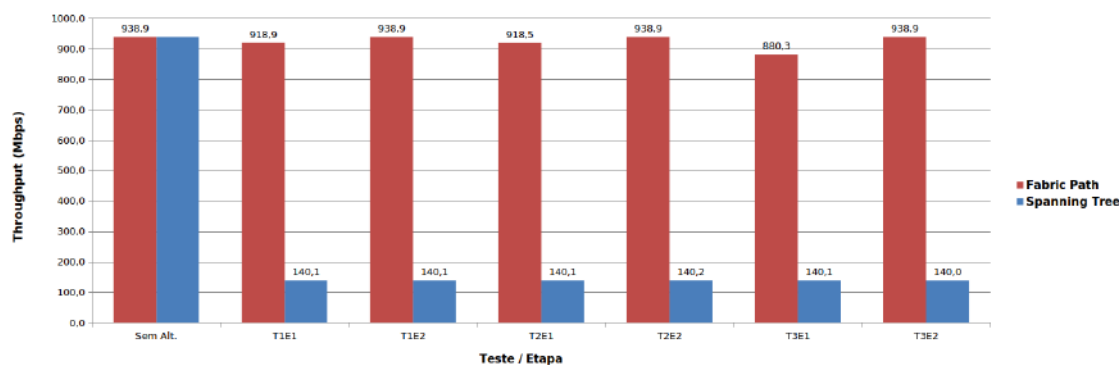
O que também é interessante ressaltar é a robustez da rede, uma vez que pode-se observar que as características como tolerância a falhas e a utilização de múltiplos caminhos preservam o *throughput* no cenário com o FabricPath, como ilustrado na Figura 32. Essas são características importantes que devem ser ressaltadas com relação ao uso do TRILL/FP em comparação ao STP, uma vez que as alterações que são feitas na rede privilegiam o *throughput* de modo a não degradar o desempenho das aplicações.

Na fase seguinte, foram realizados os mesmos experimentos, porém, a ferramenta utilizada foi o *netperf* e a métrica observada é o *throughput*. O comando utilizado foi:

```
1 $ netperf -H <IP> -t TCP_STREAM -l -1G
```

A opção “-H <IP>” é o parâmetro para indicar o endereço IP do servidor remoto. A opção “-t TCP_STREAM” sinaliza para que seja gerado e enviado um fluxo TCP cujo tamanho é especificado pela opção “-l -1G” que o estipula em um gigabyte de dados. O motivo da escolha do tamanho é análogo ao da escolha do número de pacotes a serem enviados, realizados nos experimentos com *echo reply/request*.

Figura 32: Gráfico *Throughput* x Experimento/Etapa



Como observado na Figura 32, diferentemente do STP, o cenário com o Fabric-Path mantém o *throughput* estabilizado mesmo quando a topologia é afetada.

Um ponto a se ressaltar, que experimentos realizados não puderam demonstrar, é o aumento da capacidade de banda disponível pelo fato da tecnologia se utilizar de técnicas de multicaminhos. Isto seria possível utilizando concorrentemente diversos fluxos através dos *links* disponíveis, o que aumentaria consideravelmente o *throughput*. O que não ocorre naturalmente no cenário com STP, pois apenas um *link* está encaminhando o tráfego.

6 CONCLUSÃO

Aumentar a disponibilidade de banda e diminuir os tempos de resposta em ocasião de falhas se tornaram condições indispensáveis para a organização de redes de *datacenter* que oferecem serviços de alta complexidade e demanda computacional. No presente trabalho nos propusemos a contextualizar as mudanças atravessadas pelas arquiteturas de redes de *datacenters* até o presente momento, ilustrando sua demanda por maior desempenho e expondo um protocolo que fornece uma alternativa viável, sem grandes impactos às infraestruturas existentes.

No capítulo inicial de conceitos básicos, explicamos como o *Spanning Tree Protocol* pode se tornar um problema e por que se torna necessária a busca por alternativas. É importante ressaltar que o modelo topológico tradicional em conjunto com o uso do STP mantém sua eficácia, resolvendo o problema que originalmente se propôs a resolver, porém, com os novos paradigmas, sua eficiência é posta em xeque, ao se utilizar de bloqueios de conexões redundantes, limitando a disponibilidade de banda e permitindo a utilização de caminhos sub-ótimos. Ainda assim, a utilização do STP continua viável em ambientes corporativos, uma vez que as novas demandas se aplicam especialmente aos ambientes de *datacenter*, e são impulsionadas pela presença crescente de tecnologias de Computação em Nuvem e *Big Data*, entre outras.

Mostramos que as formas de solucionar esse problema incluem mudanças na organização topológica da estrutura de *datacenter*, permitindo uma melhor adequação ao novo padrão de tráfego observado; e a utilização de novos protocolos que permitem atender aos requisitos de escalabilidade, alta disponibilidade e grande capacidade de transferência de dados, ao mesmo tempo que protegem a rede de ciclos infinitos e da sobrecarga de *links* causada por eles. Mostramos também que no caso do protocolo proposto como solução, isto é, o TRILL, isso pode ser obtido unindo características intrínsecas de camada dois (como auto-configurabilidade) e de camada três (como a utilização de técnicas de roteamento).

Como trabalhos futuros, ressaltamos que outras tecnologias se propõem a resol-

ver os mesmos problemas expostos nesse trabalho, como por exemplo a tecnologia de múltiplos caminhos em camada de enlace (L2MP). Um outro exemplo é o SPB [2] (*Shortest Path Bridging*), protocolo construído a partir do padrão 802.1 do IEEE e o FabricPath, proprietário da Cisco. Todos utilizam técnicas de ECMP e encapsulamento dos quadros, além de possuírem características das camadas dois e três (enlace de dados e rede) a partir do uso de extensões do IS-IS. Uma evolução dos protocolos de encapsulamento para tratar a demanda de mobilidade de máquinas virtuais em diversos segmentos de rede foi dada com estruturas VXLAN [26] (*Virtual Extensible Local Area Network*), que provê a sobreposição da camada dois em uma estrutura de camada três. Há também a possibilidade do uso de SDN [28] (*Software Defined Network*) que permite a separação entre o plano de controle e o plano de dados da rede, tornando programável o encaminhamento dos dados de modo a explorar com mais eficiência cada tipo de situação envolvendo diferentes aplicações. Tais tecnologias porém, envolvem outras abordagens, e se encontram fora do escopo deste trabalho.

Por fim, através dos experimentos executados foi possível demonstrar a ineficiência de se utilizar o modelo de infraestrutura tradicional em conjunto com o STP em redes de *datacenter*, onde as aplicações demandam mais capacidade lateralmente (padrão de tráfego leste-oeste) do que verticalmente (padrão de tráfego norte-sul). Pôde-se observar por meio dos resultados obtidos que existe um impacto maior na eficiência da operação da rede quando na ocorrência de falhas ao utilizar o cenário tradicional com STP em comparação à utilização do cenário proposto com as bases do TRILL. Vale ressaltar, porém, que para medir o impacto real se faz necessária a realização de experimentos em um ambiente onde estão presentes as aplicações de *Big Data* e Computação em Nuvem citadas no decorrer do trabalho. Serve assim, este trabalho, como base para essas futuras análises.

REFERÊNCIAS

- [1] AL-FARES, M., LOUKISSAS, A., E VAHDAT, A. A scalable, commodity data center network architecture. In *ACM SIGCOMM Computer Communication Review* (2008), vol. 38, ACM, pp. 63–74.
- [2] ALLAN, D., ASHWOOD-SMITH, P., BRAGG, N., FARKAS, J., FEDYK, D., OUELLETE, M., SEAMAN, M., E UNBEHAGEN, P. Shortest path bridging: Efficient control of larger ethernet networks. *IEEE Communications Magazine* 48, 10 (2010), 128–135.
- [3] BANERJEE, A., E WARD, D. Extensions to IS-IS for layer-2 systems. Relatório técnico, 2011.
- [4] BARROSO, L. A., CLIDARAS, J., E HÖLZLE, U. The datacenter as a computer: An introduction to the design of warehouse-scale machines. *Synthesis lectures on computer architecture* 8, 3 (2013), 1–154.
- [5] BENSON, T., AKELLA, A., E MALTZ, D. A. Network traffic characteristics of data centers in the wild. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement* (2010), ACM, pp. 267–280.
- [6] CALLON, R. RFC 1195, entitled Use of OSI ISIS for routing in TCP. *IP and Dual Environments* (1990), 1–80.
- [7] CARTHEN, C., WILSON, W., BEDWELL, R., E RIVERA, N. Intermediate lan switching. In *Cisco Networks*. Springer, 2015, pp. 69–91.
- [8] CHE, D., SAFRAN, M., E PENG, Z. From big data to big data mining: challenges, issues, and opportunities. In *International conference on database systems for advanced applications* (2013), Springer, pp. 1–15.
- [9] CORRÊA, C., LUCENA, S., ROTHENBERG, C., E SALVADOR, M. Uma plataforma de roteamento como serviço baseada em redes definidas por software. In *Workshop de Gerência e Operação de Redes e Serviços (WGRS). SBRC* (2012), vol. 12, sn.

- [10] DEAN, J., E GHEMAWAT, S. Mapreduce: simplified data processing on large clusters. *Communications of the ACM* 51, 1 (2008), 107–113.
- [11] DIJKSTRA, E. W. A note on two problems in connexion with graphs:(numerische mathematik, _1 (1959), p 269-271).
- [12] EASTLAKE 3RD, D., SENEVIRATHNE, T., GHANWANI, A., DUTT, D., E BANNERJEE, A. Transparent interconnection of lots of links (TRILL) use of IS-IS. Relatório técnico, 2014.
- [13] FEDYK, D., ASHWOOD-SMITH, P., ALLAN, D., BRAGG, A., E UNBEHAGEN, P. Is-is extensions supporting ieee 802.1 aq shortest path bridging. Relatório técnico, 2012.
- [14] FORTZ, B., REXFORD, J., E THORUP, M. Traffic engineering with traditional ip routing protocols. *IEEE communications Magazine* 40, 10 (2002), 118–124.
- [15] FRANCOIS, P., FILSFILS, C., EVANS, J., E BONAVENTURE, O. Achieving sub-second IGP convergence in large IP networks. *ACM SIGCOMM Computer Communication Review* 35, 3 (2005), 35–44.
- [16] GINSBERG, L., E PREVIDI, S. M. chen,"IS-IS extensions for advertising router information. Relatório técnico, RFC 7981, DOI 10.17487/RFC7981, October 2016,< <http://www.rfc-editor.org>
- [17] GROVER, H., RAO, D., FARINACCI, D., E MORENO, V. Overlay transport virtualization. *Internet Engineering Task Force, Internet Draft* (2011).
- [18] GUBBI, J., BUYYA, R., MARUSIC, S., E PALANISWAMI, M. Internet of things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems* 29, 7 (2013), 1645–1660.
- [19] HOODA, S. K., KAPADIA, S., E KRISHNAN, P. *Using Trill, FabricPath, and VXLAN: designing massively scalable data centers (MSDC) with overlays*. Cisco Press, 2014.
- [20] HOPPS, C. Analysis of an equal-cost multi-path algorithm. Relatório técnico, 2000.

- [21] IEEE. 802.1D-2004 - IEEE standard for local and metropolitan area networks.
- [22] JONES, R. Netperf. <https://hewlettpackard.github.io/netperf/>. Accessed: 2019-04-09.
- [23] KANT, K. Data center evolution: A tutorial on state of the art, issues, and challenges. *Computer Networks* 53, 17 (2009), 2939–2965.
- [24] KUROSE, J., E ROSS, K. Computer networks and the internet. *Computer networking: A Top-down approach. 7th ed. London: Pearson* (2016).
- [25] LINDEM, A., SHEN, N., VASSEUR, J., AGGARWAL, R., E SHAFFER, S. Extensions to OSPF for advertising optional router capabilities. Relatório técnico, 2016.
- [26] MAHALINGAM, M., DUTT, D., DUDA, K., AGARWAL, P., KREEGER, L., SRIDHAR, T., BURSELL, M., E WRIGHT, C. Virtual extensible local area network (vxlan): A framework for overlaying virtualized layer 2 networks over layer 3 networks. Relatório técnico, 2014.
- [27] MELL, P., GRANCE, T., E OTHERS. The NIST definition of cloud computing.
- [28] NADEAU, T. D., E GRAY, K. *SDN: Software Defined Networks: an authoritative review of network programmability technologies*. "O'Reilly Media, Inc.", 2013.
- [29] ORAN, D. OSI IS-IS intra-domain routing protocol. Relatório técnico, 1990.
- [30] PERLMAN, R. A protocol for distributed computation of a spanning tree in an extended LAN. In *Ninth Data Communications Symposium* (1985).
- [31] PERLMAN, R., E EASTLAKE, D. Introduction to TRILL. *The Internet Protocol Journal* 14, 3 (2011), 2–19.
- [32] PERLMAN, R., EASTLAKE 3RD, D., DUTT, D., GAI, S., E GHANWANI, A. Routing bridges (RBridges): Base protocol specification. Relatório técnico, 2011.

- [33] PLUMMER, D. C. Rfc 826: An ethernet address resolution protocol. *Network Working Group* (1982).
- [34] POSTEL, J. Internet control message protocol. Relatório técnico, 1981.
- [35] SANTANA, G. A. *Data center virtualization fundamentals: understanding techniques and designs for highly efficient data centers with Cisco Nexus, UCS, MDS, and beyond*. Cisco Press, 2013.
- [36] SEAMAN, M. Link aggregation control protocol. *IEEE http://grouper.ieee.org/groups/802/3/ad/public/mar99/seaman 1* (1999), 0399.
- [37] SHAND, M., E GINSBERG, L. Reclassification of rfc 1142 to historic. Relatório técnico, 2014.
- [38] SHVACHKO, K., KUANG, H., RADIA, S., CHANSLER, R., E OTHERS. The hadoop distributed file system. In *MSST* (2010), vol. 10, pp. 1–10.
- [39] SYSTEMS, C. Understanding multiple spanning tree protocol (802.1s). Relatório técnico, 2007.
- [40] SYSTEMS, C. Cisco data center spine-and-leaf architecture: design overview. Relatório técnico, 2016.
- [41] TANENBAUM, A. S., E OTHERS. Computer networks, 4-th edition. *ed: Prentice Hall* (2003).
- [42] TOUCH, J., E PERLMAN, R. Transparent interconnection of lots of links (TRILL): Problem and applicability statement. Relatório técnico, 2009.
- [43] WIXOM, B., ARIYACHANDRA, T., DOUGLAS, D. E., GOUL, M., GUPTA, B., IYER, L. S., KULKARNI, U. R., MOONEY, J. G., PHILLIPS-WREN, G. E., E TURETKEN, O. The current state of business intelligence in academia: The arrival of big data. *CAIS* 34, 1 (2014), 1–13.
- [44] ZIMMERMANN, H. Innovations in internetworking. Artech House, Inc., Norwood, MA, USA, 1988, ch. OSI Reference Model&Mdash;The ISO Model of Architecture for Open Systems Interconnection, pp. 2–9.