

ANO
2018

ANDERSON HENRIQUE DA SILVA MARCONDES | CONTROLE DA QUALIDADE DE
SERVIÇO PARA O SISTEMA MOODLE EM REDES DEFINIDAS POR SOFTWARE

O congestionamento em uma topologia de comunicação, que é um recurso ubíquo e compartilhado, pode impactar negativamente nas aplicações distribuídas e comunicantes.

Aplicações acadêmicas podem ter o seu desempenho comprometido em virtude de ruídos ocasionados por aplicações secundárias. Nesse contexto, este trabalho tem por objetivo reconfigurar os *switches* da rede local para priorizar o tráfego do Moodle. A solução implementada, denominada SDN4Moodle, identifica as atividades efetuadas pelos usuários do Moodle, reconfigurando o encaminhamento de fluxos de acordo com a qualidade de serviço necessária para acessar os recursos.

Orientador: Guilherme Piêgas Koslovski

Joinville, 2018



UDESC

UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

DISSERTAÇÃO DE MESTRADO

**CONTROLE DA QUALIDADE DE
SERVIÇO PARA O SISTEMA
MOODLE EM REDES DEFINIDAS
POR SOFTWARE**

ANDERSON HENRIQUE DA SILVA MARCONDES

JOINVILLE, 2018

UNIVERSIDADE DO ESTADO DE SANTA CATARINA - UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT
MESTRADO EM COMPUTAÇÃO APLICADA

ANDERSON HENRIQUE DA SILVA MARCONDES

CONTROLE DA QUALIDADE DE SERVIÇO PARA O SISTEMA
MOODLE EM REDES DEFINIDAS POR SOFTWARE

JOINVILLE

2018

ANDERSON HENRIQUE DA SILVA MARCONDES

**CONTROLE DA QUALIDADE DE SERVIÇO PARA O SISTEMA
MOODLE EM REDES DEFINIDAS POR SOFTWARE**

Dissertação submetida ao Programa de Pós-Graduação em Computação Aplicada do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina, para a obtenção do grau de Mestre em Computação Aplicada.

Orientador: Prof. Dr. Guilherme Piêgas Koslovski

JOINVILLE

2018

Marcondes, Anderson

Controle da Qualidade de Serviço para o Sistema Moodle em Redes Definidas por Software / Anderson Marcondes. - Joinville , 2018.

104 p.

Orientador: Guilherme Koslovski

Dissertação (Mestrado) - Universidade do Estado de Santa Catarina, Centro de Ciências Tecnológicas, Programa de Pós-Graduação em Computação Aplicada, Joinville, 2018.

1. Moodle. 2. SDN. 3. OpenFlow. 4. SDN4Moodle.
I. Koslovski, Guilherme. II. Universidade do Estado de Santa Catarina. Programa de Pós-Graduação. III. Título.

**Controle da Qualidade de Serviço para o Sistema Moodle em Redes Definidas
por Software**

por

Anderson Henrique da Silva Marcondes


Esta dissertação foi julgada adequada para obtenção do título de

Mestre em Computação Aplicada


Área de concentração em “Ciência da Computação”,
e aprovada em sua forma final pelo

CURSO DE MESTRADO ACADÊMICO EM COMPUTAÇÃO APLICADA
DO CENTRO DE CIÊNCIAS TECNOLÓGICAS DA
UNIVERSIDADE DO ESTADO DE SANTA CATARINA.


Banca Examinadora:



Prof. Dr. Guilherme Piegas Koslovski
CCT/UDESC (Orientador/Presidente)



Prof. Dr. Giovani Gracioli
University of Waterloo/Canadá



Prof. Dr. Charles Christian Miers
CCT/UDESC

Joinville, SC, 27 de fevereiro de 2018.

Dedico este trabalho ao meu filho Gabriel,
meu parceiro e incentivador que me acompanhou durante todo o mestrado.

AGRADECIMENTOS

Agradeço ao professor Guilherme Koslovski, por ter dado a oportunidade de me orientar no Programa de Mestrado da UDESC. Também agradeço aos colegas da Coordenação de Tecnologia da Informação do Instituto Federal Catarinense, por suprirem as demandas do setor durante a minha ausência para cursar a Pós-Graduação. Por fim, agradeço também à minha namorada Karolina Vitorino.

“Agradeça às adversidades e aos obstáculos, pois eles impedem que você se acomode e o estimulam permanentemente a superar seus limites.”

Roberto Shinyashiki

RESUMO

O congestionamento em uma topologia de comunicação, que é um recurso ubíquo e compartilhado, pode impactar negativamente nas aplicações distribuídas e comunicantes. Usualmente, a ocorrência de conflitos no compartilhamento da rede afeta a qualidade do serviço em execução, bem como a experiência observada pelos usuários finais dos sistemas. Aplicações acadêmicas podem ter o seu desempenho comprometido em virtude de ruídos ocasionados por aplicações secundárias, ou seja, o tráfego necessário para utilização de sistemas de gestão de aprendizagem pode ser prejudicado pelas demais aplicações em execução, tais como mídias sociais, compartilhamento de dados, clientes de sincronização de armazenamento com a nuvem, entre outros. O paradigma das Redes Definidas por *Software* (SDN) permite que o administrador programe a infraestrutura de rede para aumentar o desempenho de uma aplicação específica, definindo políticas de alocação de recursos guiadas pela própria aplicação, permitindo assim que a rede aumente o seu desempenho, superior ao observado em uma arquitetura de rede tradicional. Nesse contexto, o presente trabalho tem por objetivo reconfigurar os *switches* da rede local para priorizar o tráfego do sistema Moodle. A solução implementada, denominada SDN4Moodle, identifica as atividades efetuadas pelos usuários do Moodle, reconfigurando o encaminhamento de fluxos de acordo com a qualidade de serviço necessária para acessar os recursos (áudios, vídeos, textos interativos, provas, dentre outros). Para tal finalidade, o controle de métricas padronizado pelo protocolo OpenFlow é utilizado. A análise experimental indicou a diminuição na interferência do tráfego primário do Moodle, bem como uma potencial utilização em cenários acadêmicos.

Palavras-chaves: Moodle, SDN, OpenFlow, SDN4Moodle.

ABSTRACT

The network infrastructure is an ubiquitous and shared resource, and consequently congestion in communication can negatively impact distributed and communicating applications. Usually, the occurrence of network sharing conflicts affects the quality of the service, as well as the experience observed by end users of systems. Academic applications may have their performance compromised due to the noise induced by secondary applications. In other words, the traffic required to use learning management systems can be hampered by other applications running such as social medias, data sharing, clients storage synchronization with cloud, among others. The Software Defined Networks (SDN) paradigm allows the administrator to program the network infrastructure to increase the performance of a specific application by defining resource allocation policies guided by the application itself, thus enabling the network to help applications to perform satisfactorily, superior to observed in a traditional network architecture. In this context, the present work aims to reconfigure the local network switches to prioritize the traffic of the Moodle system. The implemented solution, called SDN4Moodle, identifies the activities carried out by Moodle users, reconfiguring the routing of flows according to the quality of service required to access the resources (audios, videos, interactive texts, tests, among others). For this purpose, the metric control standardized by the OpenFlow protocol is used. The experimental analysis indicated a decrease in Moodle primary traffic interference, as well as potential use in academic scenarios.

Key-words: Moodle, SDN, OpenFlow, SDN4Moodle.

LISTA DE ILUSTRAÇÕES

Figura 1 – Visão geral da plataforma de aprendizagem Moodle.	26
Figura 2 – Arquitetura típica do Moodle em redes acadêmicas. Adaptado de Brown e Wilson (2012).	28
Figura 3 – Comunicação entre os componentes do Moodle. Adaptado de Moodle (2017d).	30
Figura 4 – Topologia para investigação sobre o desempenho do LMS Moodle na ocorrência de congestionamento.	32
Figura 5 – Tempo médio por requisição HTTP submetido ao LMS Moodle em uma rede congestionada.	33
Figura 6 – Visão simplificada de uma arquitetura SDN. Adaptado de Kreutz et al. (2015).	38
Figura 7 – Switch OpenFlow. Adaptado de McKeown et al. (2008).	40
Figura 8 – Visão geral dos módulos que compõem SDN4Moodle e seu cenário de aplicação.	50
Figura 9 – Modelagem dos fluxos dos usuários do Moodle.	51
Figura 10 – Diagrama de sequência dos eventos dos usuários no Moodle e na ferramenta SDN4Moodle.	54
Figura 11 – Componentes do cabeçalho de um pacote OpenFlow analisados por SDN4Moodle.	58
Figura 12 – Exemplo de fluxo enviado pela SDN4Moodle para o controlador SDN	60
Figura 13 – Exemplo de métrica enviada pela SDN4Moodle para o controlador SDN	61
Figura 14 – Estrutura de arquivos do Módulo 1, <i>plugin</i> do Moodle.	66
Figura 15 – Primeiro cenário do plano de testes.	71
Figura 16 – Segundo cenário do plano de testes.	72
Figura 17 – Terceiro cenário do plano de testes.	73
Figura 18 – Quarto cenário do plano de testes.	73
Figura 19 – Cenário 1: gráfico da largura de banda do tráfego secundário do <i>iperf</i> com um usuário, sem o controle da ferramenta SDN4Moodle (vazão em Mbps).	79
Figura 20 – Cenário 1: dados do Moodle trafegados na placa de rede do dispositivo do usuário (em bytes).	80
Figura 21 – Cenário 2: gráfico da largura de banda do tráfego secundário do <i>iperf</i> com dois usuários, sem o controle da ferramenta SDN4Moodle (vazão em Mbps).	81

Figura 22 – Cenário 2: dados do Moodle trafegados na placa de rede do dispositivo do primeiro usuário (em bytes).	82
Figura 23 – Cenário 2: dados do Moodle trafegados na placa de rede do dispositivo do segundo usuário (em bytes).	83
Figura 24 – Cenário 3: gráfico com o comparativo da largura de banda do tráfego secundário do <i>iperf</i> com um usuário, com e sem o controle da SDN4Moodle (vazão em Mbps).	84
Figura 25 – Cenário 3: comparativo dos dados do Moodle trafegados na placa de rede do dispositivo do usuário (em bytes).	85
Figura 26 – Cenário 4: gráfico com o comparativo da largura de banda do tráfego secundário do <i>iperf</i> com dois usuários, com e sem o controle da SDN4Moodle (vazão em Mbps).	86
Figura 27 – Cenário 4: comparativo dos dados do Moodle trafegados na placa de rede do dispositivo do usuário 1 (em bytes).	87
Figura 28 – Cenário 4: comparativo dos dados do Moodle trafegados na placa de rede do dispositivo do usuário 2 (em bytes).	88
Figura 29 – Diagrama entidade relacionamento da ferramenta SDN4Moodle. . .	103
Figura 30 – Diagrama de Classes da ferramenta SDN4Moodle. Arquivo digital disponível em Marcondes (2018).	104

LISTA DE TABELAS

Tabela 1 – Resumo dos trabalhos relacionados com a presente proposta. . . .	42
Tabela 2 – Requisitos funcionais.	47
Tabela 3 – Requisitos não funcionais.	49
Tabela 4 – Exemplos de melhoria da aplicação alvo.	52
Tabela 5 – Atendimento aos requisitos funcionais e não funcionais na imple- mentação de SDN4Moodle.	57
Tabela 6 – Atributos de um fluxo de usuário na ferramenta SDN4Moodle. . . .	59
Tabela 7 – Prioridades dos tipos de fluxos.	61
Tabela 8 – Eventos monitorados na ferramenta Moodle.	62
Tabela 9 – Organização da fila de eventos.	63
Tabela 10 – Configuração geral da ferramenta SDN4Moodle.	63
Tabela 11 – Configuração de fluxos em SDN4Moodle.	64
Tabela 12 – Configuração das métricas para reserva de largura de banda. . . .	64
Tabela 13 – Definição de prioridades de páginas.	65
Tabela 14 – Configuração da prioridades para recursos.	65
Tabela 15 – Atributos dos eventos do Moodle capturados pela ferramenta SDN4Moodle. Adaptado de Moodle (2017f).	67
Tabela 16 – Descrição dos eventos monitorados pela ferramenta SDN4Moodle.	69
Tabela 17 – Classes do Módulo 2 da ferramenta SDN4Moodle	70
Tabela 18 – Arquitetura dos computadores utilizados para compor o cenário ex- perimental	76
Tabela 19 – Dispositivos de encaminhamento na topologia.	76
Tabela 20 – Atividades executadas pelos usuários do Moodle durante a análise experimental.	77
Tabela 21 – Sequência dos passos para coleta dos dados.	78

LISTA DE SIGLAS E ABREVIATURAS

API	<i>Application Programming Interface</i>
CAS	<i>Central Authentication Service</i>
CMSP	<i>Course Management Software Package</i>
CMS	<i>Course Management System</i>
HPC	<i>High-performance Computing</i>
IETF	<i>Internet Engineering Task Force</i>
KCD	<i>Knowledge Content Distributor</i>
KMS	<i>Knowledge Management System</i>
IMAP	<i>Internet Message Access Protocol</i>
JSON	<i>JavaScript Object Notation</i>
LAN	<i>Local Area Network</i>
LTI	<i>Learning Tools Interoperability</i>
LDAP	<i>Lightweight Directory Access Protocol</i>
LMS	<i>Learning Management System</i>
MLE	<i>Managed Learning Environment</i>
MPLS	<i>Multi-Protocol Label Switching</i>
NNTP	<i>Network News Transfer Protocol</i>
ODL	<i>OpenDayLight</i>
OLE	<i>Online Learning Environment</i>
PHB	<i>Per-hop Behavior</i>
PLE	<i>Personal Learning Environment</i>
POP3	<i>Post Office Protocol</i>
QoS	<i>Quality of Service</i>
REST	<i>Representational State Transfer</i>

RSVP *Resource Reservation Setup Protocol*

SDN *Software-Defined Networking*

SGBD Sistema de Gerenciamento de Banco de Dados

SLA *Service Level Agreement*

SDN4Moodle SDN for Moodle

TI Tecnologia da Informação

VLAN *Virtual LAN*

VLE *Virtual Learning Enviroment*

XML *Extensible Markup Language*

WAN *Wide Area Network*

SUMÁRIO

1	INTRODUÇÃO	16
1.1	PROBLEMA E MOTIVAÇÃO	17
1.2	OBJETIVO DO TRABALHO	20
1.3	METODOLOGIA	21
1.4	CONTRIBUIÇÕES DO TRABALHO	22
1.5	ORGANIZAÇÃO DO TEXTO	22
2	FUNDAMENTAÇÃO TEÓRICA	24
2.1	SISTEMA DE GESTÃO DE APRENDIZAGEM MOODLE	24
2.1.1	Ambiente de Execução e Configuração	25
2.1.2	Arquitetura e Componentes	28
2.1.3	IMPACTO DO CONGESTIONAMENTO DE REDES NO MOODLE	31
2.2	CONTROLE DE QUALIDADE DE SERVIÇO EM REDES TRADICIONAIS	33
2.3	REDES DEFINIDAS POR SOFTWARE	36
2.3.1	Planos de Controle, Dados e Gerenciamento	36
2.3.2	Interfaces de Gerenciamento	37
2.3.3	Fluxo de Dados	39
2.3.4	Tabelas de Fluxo	39
2.4	PROTOCOLO OPENFLOW	40
2.5	TRABALHOS RELACIONADOS	41
2.6	CONSIDERAÇÕES PARCIAIS	45
3	SDN4MOODLE: FERRAMENTA PARA CONTROLE DO TRÁFEGO	
	LMS	46
3.1	REQUISITOS PARA EXECUÇÃO	46
3.1.1	Pré-Requisitos	46
3.1.2	Requisitos Funcionais	47
3.1.3	Requisitos Não Funcionais	48
3.2	COMPONENTES DA ARQUITETURA E CENÁRIO DE APLICAÇÃO	49
3.3	PLUGIN NO MOODLE - MÓDULO 1	51
3.4	APLICAÇÃO DE CONTROLE - MÓDULO 2	53
3.5	COMUNICAÇÃO ENTRE OS MÓDULOS DA SDN4MOODLE	53
3.6	CONSIDERAÇÕES PARCIAIS	54

4	IMPLEMENTAÇÃO DA FERRAMENTA SDN4MOODLE	56
4.1	CONTROLADOR SDN	56
4.2	ADERÊNCIA AOS REQUISITOS	57
4.3	IDENTIFICAÇÃO DE UM FLUXO	57
4.4	ARMAZENAMENTO DE DADOS	61
4.5	IMPLEMENTAÇÃO DO MÓDULO 1	65
4.6	IMPLEMENTAÇÃO DO MÓDULO 2	68
4.7	PLANO DE TESTES	70
4.7.1	Cenário 1	71
4.7.2	Cenário 2	72
4.7.3	Cenário 3	72
4.7.4	Cenário 4	73
4.8	CONSIDERAÇÕES PARCIAIS	73
5	ANÁLISE EXPERIMENTAL	75
5.1	MÉTRICAS PARA ANÁLISE	75
5.2	AMBIENTE DE TESTES	75
5.3	CENÁRIOS DE TESTE	76
5.3.1	Cenário 1	79
5.3.2	Cenário 2	81
5.3.3	Cenário 3	84
5.3.4	Cenário 4	86
5.4	LIMITAÇÕES TÉCNICAS ENCONTRADAS	88
5.5	CONSIDERAÇÕES PARCIAIS	90
6	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	93
6.1	TRABALHOS FUTUROS	94
6.2	CONTRIBUIÇÃO	95
	REFERÊNCIAS	97
	ANEXO A – DIAGRAMA ENTIDADE RELACIONAMENTO DA FER-	
	RAMENTA SDN4MOODLE	103
	ANEXO B – DIAGRAMA DE CLASSES DA SDN4MOODLE	104

1 INTRODUÇÃO

Em uma arquitetura de rede tradicional, baseada na pilha de protocolos TCP/IP, o desenvolvimento de políticas para gerenciamento de tráfego é limitado pela rigidez no gerenciamento dos recursos de comunicação (HANDLEY, 2006). Usualmente, o desempenho de uma aplicação distribuída e comunicante é relacionado com o volume de dados trafegados, sendo limitado pela justiça de compartilhamento buscada pelo protocolo TCP (CERF; ICAHN, 2005) (DUKKIPATI et al., 2011). Ou seja, na ocorrência de n fluxos de comunicação sobre um canal congestionado, o protocolo considera a entrega de $1/n$ como abordagem justa de compartilhamento (CHIU; JAIN, 1989).

O princípio de compartilhamento justo é aplicado para redes locais, de longa distância e sobretudo para comunicação na Internet. Entretanto, em ambientes controlados (comerciais, acadêmicos, entre outros), a abordagem de compartilhamento orientada pela justiça não é preferível (POPA et al., 2011). Em determinados momentos, a execução de algumas aplicações constitui um gargalo operacional, como, por exemplo, a replicação de dados entre servidores, a sincronização de dispositivos de armazenamento ou, até mesmo, a aplicação de atividades de ensino.

No contexto das redes acadêmicas, o presente trabalho discute a problemática dos *Learning Management Systems* (LMSs) compartilhando o meio de transmissão com aplicações que podem criar gargalos no tráfego de dados na rede de computadores. Um LMS é um sistema informatizado que facilita o aprendizado *e-learning*, processando, armazenando e disseminando material educacional. Também suporta a administração e comunicação associada com ensino e aprendizagem (MCGILL; KLOBAS, 2009), podendo ser instalado na infraestrutura de Tecnologia da Informação (TI) da própria instituição (BAILEY, 1993).

A aplicação LMS tema deste trabalho é o Moodle, um *software* de código aberto executado no servidor, que foi desenvolvido inicialmente como um projeto de pesquisa de doutorado de Martin Dougiamas, com a ajuda de Peter Taylor na Curtin University of Technology (DOUGIAMAS; TAYLOR, 2003). Kumar e Dutta (2011) apresentam os principais motivadores para se utilizar o Moodle como plataforma de aprendizagem:

- É um *software* livre, o que significa que os usuários podem fazer *download*, utilizá-lo, modificá-lo e também distribuí-lo de acordo com os termos da licença GNU GPL.
- Permite que professores e tutores possam fornecer e compartilhar documentos,

atribuir tarefas, explorar discussões e também criar fóruns com os alunos.

- Pode ser utilizado em quase todos os servidores que têm o PHP instalado, sendo capaz de ser executado tanto em servidores GNU/Linux quanto MS-Windows. Ainda, o sistema pode armazenar as suas informações nos bancos de dados MySQL, PostgreSQL, Oracle, Microsoft SQL Server e MariaDB (MOODLE, 2017a).
- O Moodle é utilizado ao redor de todo o mundo por professores independentes, escolas, universidades e empresas. Há muitos recursos úteis, tais como fácil instalação, personalização de opções e configurações, boas ferramentas educacionais e suporte. Além disso, tem ótima documentação e forte suporte para segurança e administração.

O Moodle conta com mais de 87 mil sites registrados em 232 países, com 14 milhões de cursos e 118 milhões de usuários. O Brasil é o terceiro maior país em sites registrados, com um total de 5270 registros (MOODLE, 2017k). Com mais de 180 LMSs disponíveis no mercado (OPENEDUCATIONEUROPA, 2013), o Moodle se destaca como um dos mais utilizados nas instituições de ensino ao redor do mundo. Além do Moodle se destacam também o Blackboard Learn (BLACKBOARD, 2017), que é um software de código proprietário desenvolvido pela empresa Blackboard, lançado em 1997 e o Canvas (CANVAS, 2017), de código aberto, lançado em 2008 pela empresa Instructure.

O trabalho de Kuran, Pedersen e Elsner (2017) apresenta o *market share* na Europa e na região dos Estados Unidos/Canadá das principais ferramentas de LMS de acordo com o seu uso no ano de 2016. Na Europa, o Moodle domina o mercado de LMS com 65% de presença, seguido pelo Blackboard Learn com 12%. Já nos Estados Unidos e Canadá, o Moodle divide o segundo lugar junto com o Canvas (CANVAS, 2017), ambos com 20% da utilização, sendo que o LMS mais utilizado é o Blackboard Learn com 33%.

1.1 PROBLEMA E MOTIVAÇÃO

Tal como qualquer outra aplicação executada em uma rede de computadores, o Moodle pode apresentar problemas de desempenho, potencialmente perceptíveis pelos seus usuários, chegando a impactar na experiência de navegação e, consequentemente, de ensino. A problemática mais perceptível de ser notada pelo usuário do Moodle é a lentidão no carregamento das páginas no navegador Web durante a execução das atividades. Professores também podem perceber a degradação do desempenho do sistema quando há, por exemplo, a necessidade do *upload* de novas lições ou até mesmo a sua impossibilidade de envio, caso a conexão do TCP aberta

pelo navegador com o servidor do Moodle tenha que disputar com outra aplicação o compartilhamento do meio de transmissão. Por ser uma aplicação de rede executada com o protocolo TCP, o Moodle fica suscetível a qualquer gargalo que ocorra na rede, uma vez que essencialmente o TCP de cada conexão aberta pelos clientes do Moodle irá reduzir a sua janela para comportar a nova taxa disponível da largura de banda na rede, a fim de minimizar a perda de pacotes trafegados e também reduzir a retransmissão de dados entre o cliente e o servidor Moodle.

Com a eventual redução da largura de banda disponível para o Moodle e aumento da quantidade de aplicações disputando o canal de transmissão, podem ocorrer tanto o aumento da latência quanto do *jitter* nas conexões dos usuários (alunos e professores). A entrega irregular dos pacotes da aplicação até o seu destino pode acarretar o atraso do início das atividades depois que o usuário já tenha clicado no *hyperlink* da página (latência). Há também a possibilidade do aumento do *jitter*, acarretando a diminuição da qualidade da experiência do usuário caso ele esteja realizando atividades que envolvam a execução de *streaming* de áudios ou a exibição de vídeos.

A aplicação de regras para controle da qualidade do serviço (QoS, do inglês *Quality of Service*) é um processo rígido e complexo (STALLINGS, 2015), requerendo a intervenção do administrador da rede para a sua correta configuração. Sobretudo, não foi identificada uma padronização *de facto* para que uma aplicação como o Moodle informe seus requisitos de execução (em termos de largura de banda e latência) ao mecanismo de gerenciamento, para que este prepare a configuração da topologia para rajadas específicas que ocorrerão entre determinados nós da rede.

A ossificação da camada de transporte dificulta que uma aplicação expresse seus requisitos de QoS nesta camada. Extensões de *sockets* API propõem que uma aplicação interaja com a camada de transporte, informando as suas demandas de QoS para esta camada. Exemplos destas extensões são o Msocket (DAVOLI; GOLDWEBER, 2012), Socket Intents (ENGHARDT, 2013) e MultiSockets (HIGGINS et al., 2010). No entanto, não surgiu nenhuma proposta única que seja capaz de permitir a evolução da camada de transporte (PAPASTERGIOU et al., 2017). Sendo assim, informações disponíveis somente na camada de aplicação, como *login* de usuário no Moodle ou início e fim de atividades específicas não estão disponíveis aos dispositivos de encaminhamento da rede.

O uso de Redes Locais Virtuais (VLANs) (IEEE, 2014) é outra estratégia rígida da tentativa de priorização do tráfego dos dados de aplicações como o LMS Moodle em redes de computadores tradicionais. Apesar de haver a possibilidade de priorizar o tráfego de rede por portas de *switch* ou endereços MAC atribuídos a VLANs específicas, outras aplicações que estejam transmitindo ou recebendo dados nestas interfaces ou endereços físicos concorrem com o tráfego do Moodle, não sendo possível o

isolamento por aplicação ou fluxo de dados.

Além do mais, caso haja a alteração do *hardware* nas extremidades (local do ponto de rede na topologia ou a substituição da placa de rede do equipamento), é necessária a reconfiguração de todos os *switches* entre o cliente e o servidor da aplicação. Muitas vezes tratam-se de equipamentos de fabricantes distintos ou até mesmo com sistemas operacionais diferentes de um mesmo fabricante, perfazendo a necessidade da intervenção do administrador da rede e a consulta à documentação dos dispositivos. Nos piores casos, há a necessidade de se configurar equipamentos dos quais não há mais documentação disponível para consulta por não serem mais fabricados ou há a limitação física do número máximo de VLANs que podem ser criadas.

Variações do protocolo TCP com controle de congestionamento orientado pelos dispositivos de encaminhamento, como DCTCP (ALIZADEH et al., 2010a) e TIMELY (MITTAL et al., 2015), oferecem maiores vazões e menores latências ao garantir que as filas de encaminhamento dos *switches* não são totalmente preenchidas. Entretanto, a aplicação de políticas para controle de QoS individualizado não é facilitada, ou seja, o princípio de justiça no compartilhamento buscado pelo TCP é mantido, independente das otimizações aplicadas.

Quanto ao encaminhamento de dados, tradicionalmente, o equipamento responsável possui informações somente da camada na qual atua, seja ele um pacote para um roteador na camada de rede ou um quadro para um *switch* na camada de enlace. Oportunidades de roteamento baseadas em portas de origem e destino do servidor e clientes do Moodle, endereço físico na camada de rede ou de endereço lógico na camada de enlace, por exemplo, são impeditivos caso o dispositivo de encaminhamento não consiga interpretar as informações contidas na unidade de dados de protocolo diferente da sua camada correspondente. Essa limitação de informações no momento da tomada de decisão do encaminhamento ocasiona a rigidez, não permitindo que pesquisadores elaborem formas de roteamento distintas das que estão embarcadas no *hardware* em questão (HANDLEY, 2006).

O paradigma das Redes Definidas por *Software* (SDN, do inglês *Software Defined Network*) propõe a separação entre os planos de dados e de controle da rede (MCKEOWN et al., 2008). A motivação para essa abordagem é baseada na diferenciação e priorização de tráfego, ou seja, o administrador consegue disponibilizar fatias isoladas da rede para fins específicos, respeitando os requisitos das aplicações. Ainda, os fabricantes podem permitir acesso ao plano de controle de seus dispositivos, sem precisar abrir mão de segredos tecnológicos envolvidos na programação do plano de dados dos equipamentos. Nesse cenário, um dispositivo logicamente centralizado (denominado controlador) é responsável pelo gerenciamento do plano de controle.

Em SDN, o compartilhamento justo efetuado por TCP é contornado através da definição de prioridades no encaminhamento de fluxos. Essa tecnologia permitiu o desenvolvimento de novas soluções para engenharia de tráfego em *data centers* (MYSORE et al., 2009) (BEZERRA et al., 2016). Por meio da caracterização do tráfego em fluxos de dados na rede, os administradores podem desenvolver políticas específicas para gerenciamento de tráfego interno. Para tanto, cada fluxo pode ser individualmente analisado pelo controlador SDN, guiado por políticas de encaminhamento que atendam aos requisitos das aplicações em execução. Neste contexto, o uso de SDNs para melhorar o desempenho de aplicações executadas em rede torna-se uma solução contemporânea, elegante e padronizada para mapear possíveis gargalos no tráfego dos dados do Moodle, atuando de forma proativa para alocar largura de banda necessária conforme eventos da aplicação são encaminhados ao controlador da SDN.

É exatamente nessa oportunidade que a hipótese de pesquisa do presente trabalho está embasada. Em ambientes acadêmicos, a utilização e o desempenho de sistemas de gestão de aprendizagem é diretamente afetado por tráfegos concorrentes. Por atuar como centralizador de conteúdo, um LMS é um potencial gargalo na execução de atividades de ensino tais como aplicação de provas, desenvolvimento de exercícios, realização de reuniões com compartilhamento de dados, exibição de conteúdo multimídia, entre outros. Uma investigação preliminar (discutida em detalhes no Capítulo 2) indicou que a presença de 20% de tráfego concorrente no enlace de acesso ao LMS pode dobrar a latência percebida pelos usuários do sistema, portanto piorando a qualidade de experiência (QoE - *Quality of Experience*).

Este problema também está presente em outras áreas, tais como a de computação de alto desempenho (HPC, do inglês *High-performance computing*), em que aplicações sensíveis a gargalos na rede também podem ter o seu desempenho degradado caso não haja a priorização dos seus tráfegos.

Foram comparados diversos trabalhos relacionados ao escopo deste trabalho (apresentados na Seção 2.5), os quais propuseram melhorar o desempenho de métricas de aplicações que utilizam a rede de computadores. No entanto, estas propostas modificaram o protocolo OpenFlow para atender as demandas das aplicações ou foram desenvolvidos para situações específicas, sem haver a possibilidade da sua utilização para o Moodle.

1.2 OBJETIVO DO TRABALHO

O objetivo geral desta pesquisa é melhorar a qualidade de serviço do sistema em uma rede local definida por *software*. Assim, o objetivo principal é o desenvolvimento de uma aplicação SDN para encaminhamento prioritário do tráfego do Moodle

na rede.

Para atingir o propósito principal deste trabalho, os seguintes objetivos específicos são elencados:

- Alocação de largura de banda nos *switches* guiada por eventos do Moodle, como, por exemplo, o início ou fim de sessão de usuário ou de uma atividade.
- Realocação de caminhos para encaminhamento de fluxos em tempo de execução, orientada pelo Moodle.
- Prototipação do mecanismo proposto.
- Análise experimental quantificando a qualidade do serviço e a experiência dos usuários com e sem o uso da aplicação SDN proposta.

1.3 METODOLOGIA

Esta pesquisa foi classificada sob diversas características, as quais são especificadas em Gil (2007). Quanto à abordagem, a pesquisa deste trabalho é a quantitativa, na qual é possível comparar os resultados de forma numérica, avaliando o impacto das métricas de desempenho da aplicação Moodle, com e sem o uso da ferramenta da proposta da pesquisa. Com respeito à classificação da pesquisa, trata-se de um objeto, uma vez que o trabalho desenvolvido é um protótipo de *software*, podendo também ser classificada como pesquisa tecnológica (VALERIANO, 2004). Já em relação à natureza, é uma pesquisa aplicada, na qual foi possível aplicar uma solução elegante, contemporânea e padronizada a um problema específico no ambiente educacional.

De posse da fundamentação teórica necessária, foi elaborada a proposta da ferramenta através do uso do paradigma das SDNs. Esta foi então discutida e avaliada, através do desenvolvimento de um protótipo de *software* executado em uma topologia de rede controlada, a fim de apresentar a sua viabilidade em uma rede acadêmica que utilize o Moodle como aplicação LMS. Assim, a metodologia de trabalho utilizada na pesquisa desta dissertação é resumida em: (i) Estudo sobre os conceitos envolvidos no paradigma das SDNs; (ii) Compreender a arquitetura, o funcionamento e o comportamento dos eventos da aplicação LMS Moodle; (iii) Definição da arquitetura para controle do tráfego LMS; (iv) Implementação da ferramenta proposta; (v) Realização de testes comparativos; e (vi) Escrita da dissertação.

1.4 CONTRIBUIÇÕES DO TRABALHO

Neste trabalho foi proposta e desenvolvida uma solução para alocação de largura de banda em tempo de execução para a aplicação Moodle em uma SDN, baseada nos eventos que ocorrem durante o uso da aplicação pelos seus usuários. A ferramenta proposta, o *SDN for Moodle (SDN4Moodle)*, permite que a largura de banda da rede seja priorizada para os usuários do Moodle que estejam efetivamente realizando atividades relacionadas ao ensino em uma rede de dados acadêmica, evitando que suas experiências de navegação sejam deterioradas caso tenham que concorrer com o tráfego de outras aplicações menos prioritárias. Essa alocação da largura de banda ocorre sem a necessidade da intervenção do administrador da rede para realizar a reconfiguração dos *switches* da rede para priorização do tráfego LMS.

Para analisar tal proposta, foi desenvolvido um protótipo, o SDN4Moodle, responsável pela alocação de largura de banda na SDN e também um *plugin* do Moodle responsável por enviar as informações sobre os eventos ocorridos durante a navegação dos usuários. A ferramenta desenvolvida foi confrontada com tráfego secundário (ruído) gerado na rede durante a navegação de usuários do Moodle, a fim de demonstrar quantitativamente que o desempenho de aplicações que utilizam a rede podem ter o seu desempenho melhorado através do paradigma das SDNs.

Os resultados alcançados mostram a viabilidade do uso da solução proposta para melhorar o desempenho da aplicação específica Moodle quando executada em uma SDN, bem como expandir o seu uso para mais aplicações e também potencializar a temática da necessidade das aplicações se comunicarem com os dispositivos da rede, pois somente as aplicações sabem de antemão quais serão as demandas de largura de banda para seus usuários (SALTZER; REED; CLARK, 1984).

1.5 ORGANIZAÇÃO DO TEXTO

O trabalho está dividido em seis capítulos, sendo o Capítulo 2 o que apresenta a fundamentação teórica necessária à compreensão do assunto, discutindo sobre o sistema de gestão de aprendizagem Moodle, os detalhes da sua arquitetura, além dos impactos ocasionados por ruídos na rede durante a sua execução. Também é apresentado como é realizado o controle de qualidade de serviço em redes tradicionais. O paradigma das Redes Definidas por *Software*, bem como o protocolo OpenFlow também são expostos neste capítulo. Por fim, são indicados os trabalhos relacionados.

O Capítulo 3 aborda a arquitetura proposta para priorização do tráfego da ferramenta Moodle, bem como o seu cenário de aplicação e requisitos funcionais e não-funcionais. Também é apresentado o funcionamento da comunicação entre os módulos da ferramenta. Por sua vez, o Capítulo 4 lista os detalhes técnicos da implemen-

tação, apontando como ocorre a identificação de um fluxo OpenFlow na ferramenta e o detalhamento da arquitetura do banco de dados. É explicado o funcionamento dos dois módulos da ferramenta proposta: o Módulo 1, que é um *plugin* da aplicação Moodle que captura os seus eventos e também o Módulo 2, que trata o eventos recebidos pelo Módulo 1. Por último, é exibido o plano de testes deste trabalho.

O Capítulo 5 apresenta a análise experimental de acordo com o plano de testes definido no Capítulo 4. Os experimentos apontam a eficácia da ferramenta para controle de tráfego de usuários do Moodle e demais clientes em uma rede compartilhada. Ainda, o capítulo discute as dificuldades técnicas encontradas durante a realização do presente trabalho. Por fim, o Capítulo 6 contém as considerações finais e perspectivas para continuidade do estudo.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo contém a fundamentação teórica para embasar o entendimento dos elementos-chave desta pesquisa: o LMS Moodle e sua arquitetura de funcionamento, bem como o paradigma das Redes Definidas por *Software*. Para analisar como é possível utilizar SDN para melhorar o desempenho de aplicações, torna-se necessário realizar um estudo para identificar quais conceitos estão envolvidos com este paradigma, identificando trabalhos prévios que conseguiram atingir objetivos semelhantes.

Sendo dividido em cinco seções, a Seção 2.1 apresenta o sistema de gestão de aprendizagem Moodle, sua arquitetura e funcionamento. Também traz uma investigação que verificou o impacto que ruídos na rede podem causar no seu desempenho. A Seção 2.2 discute os principais elementos que configuram a rigidez no gerenciamento de redes de comunicação relacionados com o desempenho do Moodle. O paradigma SDN é detalhado na Seção 2.3. Posteriormente, na Seção 2.4, é exibido o protocolo OpenFlow, padrão *de facto* adotado para a comunicação entre o plano de dados e o plano de controle em uma SDN. Por último, são apresentadas as considerações parciais do capítulo.

2.1 SISTEMA DE GESTÃO DE APRENDIZAGEM MOODLE

Variações do acrônimo LMS podem ser encontradas na literatura, tais como *Course Management System* (CMS), *Virtual Learning Environment* (VLE), *Knowledge Content Distributor* (KCD), *Knowledge Management System* (KMS), *Course Management Software Package* (CMSP), *Managed Learning Environment* (MLE) *Personal Learning Environment* (PLE) ou também *Online Learning Environment* (OLE), de acordo com a sua abrangência e aplicação (MOORE; DICKSON-DEANE; GALYEN, 2011). Dentre os sistemas de gestão de aprendizagem disponíveis, o presente trabalho selecionou como aplicação alvo o Moodle, um LMS que permite a criação de ambientes personalizados de aprendizagem.

Desenvolvido inicialmente em 1999 por Martin Dougiamas durante o seu projeto de doutorado na Curtin University of Technology, teve sua versão 1.0 lançada em 2002. Por ter sido implementado desde o início como um *software* de código aberto com licenciamento GNU GPL, é uma plataforma difundida no meio acadêmico em 232 países. Sobretudo, o Moodle é uma plataforma de aprendizagem amplamente utilizada nas instituições de ensino do país, tendo o Brasil o terceiro maior número de sítios registrados (MOODLE, 2017b). No Estado de Santa Catarina, a maioria das

instituições públicas de ensino superior utilizam o Moodle como ferramenta de aprendizagem, sendo elas a Universidade do Estado de Santa Catarina (UDESC, 2018), a Universidade Federal de Santa Catarina (UFSC, 2018), a Universidade Federal Fronteira Sul (UFFS, 2018), o Instituto Federal de Santa Catarina (IFSC, 2018) e o Instituto Federal Catarinense (IFC, 2018). Assim, a seleção do Moodle como aplicação alvo é primariamente justificada pela possibilidade de posterior aplicação em cenários reais de ensino em Santa Catarina.

Além da ampla aceitação no meio acadêmico (MOODLE, 2017k), a sua escolha como aplicação alvo também deve-se ao fato dele ser um *software* de código aberto, permitindo que sejam desenvolvidos módulos para agregar novas funcionalidades. Além disso, é uma aplicação cujo tráfego deve ser priorizado em ambiente escolar, no entanto esse tráfego acaba concorrendo com as demais aplicações na rede.

O problema se agrava em redes sem fio, uma vez que nesse meio de comunicação a velocidade máxima muitas vezes é menor que em uma infraestrutura de rede cabeada. Por ser na maioria das vezes *half-duplex* (JAIN et al., 2011), um ponto de acesso sem fio não pode enviar e receber dados ao mesmo tempo, tendo que executar uma ação por vez. Desta forma, usuários que estejam executando uma atividade no Moodle concorrem de forma injusta em um ambiente sem fio, pois além de disputarem entre si o canal de comunicação que só aceita uma ação de envio/recebimento por vez, terão que competir com as aplicações de outros usuários que não estejam executando atividades escolares no Moodle.

Para um melhor entendimento da problemática de como o funcionamento da rede pode impactar de forma negativa no desempenho do Moodle e a consequente depreciação na qualidade da experiência da navegação do usuário, esta seção abordará a arquitetura do LMS Moodle, apresentando os principais aspectos que envolvem o seu funcionamento e a interação dos seus usuários. As referências bibliográficas principais são a sua própria documentação oficial e também o livro de Brown e Wilson (2012).

2.1.1 Ambiente de Execução e Configuração

Uma visão geral da plataforma de aprendizagem Moodle é apresentada na Figura 1. A camada de acesso representa o *frontend*, enquanto que o sistema e o banco de dados são o *backend* da arquitetura. Apesar de haver a possibilidade do professor disponibilizar atividades a serem realizadas pelos alunos de forma *offline*, o Moodle é uma aplicação Web, na qual toda a interação de um usuário com o LMS ocorre através do protocolo HTTP na rede de computadores. O servidor do Moodle pode ser instalado tanto na rede da instituição na sua *Local Area Network* (LAN) ou pode ser alugado de

terceiros, tais como na modalidade SaaS de nuvens computacionais, hospedado fora da sua rede, acessível por *Wide Area Network* (WAN).

A escolha do local da instalação do Moodle depende da ponderação entre os custos de se manter uma infraestrutura interna de TI e a necessidade do sigilo dos dados sensíveis da aplicação armazenados em sistemas de terceiros. No escopo do presente trabalho, o foco principal corresponde à hospedagem local com acesso em LAN, cuja versão utilizada do Moodle foi a 3.3+ (*build* 20170629). Para hospedagens não locais (nuvem ou outro domínio administrativo), tem-se o pré-requisito dos *switches* terem o protocolo OpenFlow ativado e também o controle administrativo sobre estes equipamentos, que devem estar conectados ao controlador SDN que se comunica com a ferramenta proposta neste trabalho.

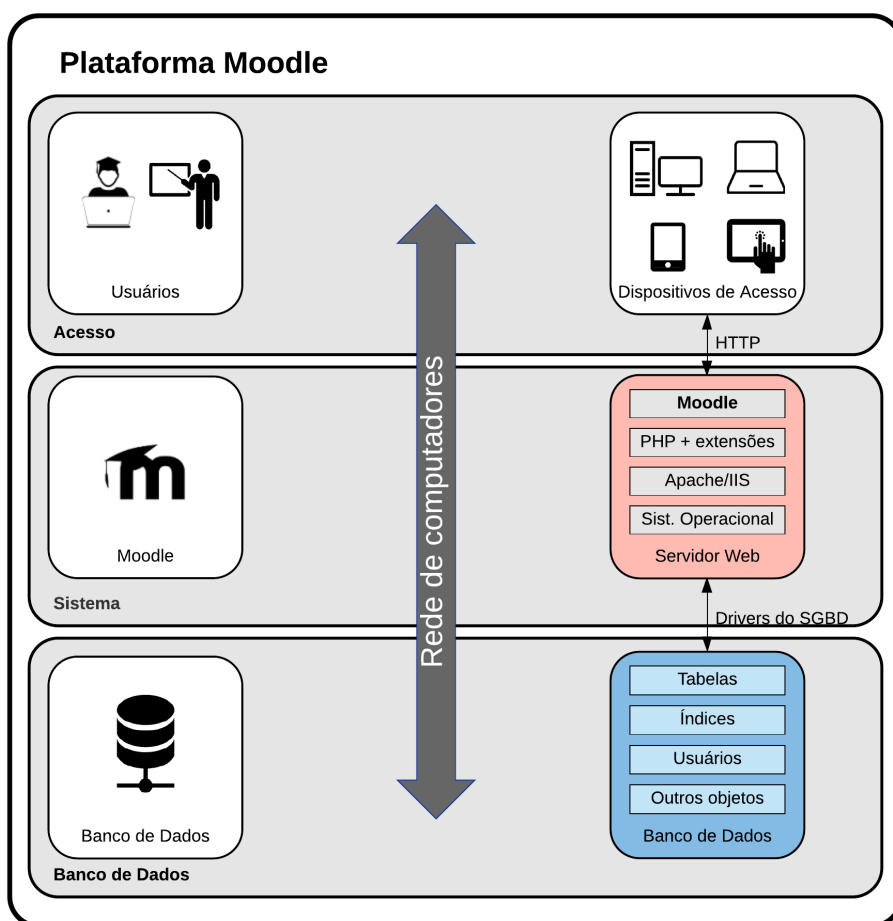


Figura 1 – Visão geral da plataforma de aprendizagem Moodle.

O acesso ao sistema Moodle é baseado em papéis e permissões. Os papéis básicos dos usuários são o de estudante, que pode acessar e participar de cursos; e o de professor, que pode gerenciar e adicionar conteúdo aos cursos, cada qual com as suas permissões de acesso no sistema conforme o contexto em que estão inseridos.

Um contexto é um espaço no Moodle, tais como cursos e atividades. Além disso, é possível criar novos papéis de usuários caso existam necessidades específicas. A lista completa de papéis padrão no Moodle (MOODLE, 2017j):

- Administrador: permissão para controlar o Moodle de forma irrestrita.
- Gerente: permissão de administração com menos privilégios.
- Criador de curso: permite a criação de cursos e atuar como professor nos cursos.
- Professor: pode gerenciar e adicionar conteúdo aos cursos, além de alterar e avaliar atividades.
- Moderador: pode interagir e avaliar atividades nos cursos, mas não pode modificá-las.
- Estudante: pode acessar e participar dos cursos aos quais tem autorização.
- Visitante: pode visualizar cursos, mas não pode participar.
- Usuário autenticado: todos os usuários que se autenticarem no sistema.
- Usuário autenticado na página inicial: papel de usuário autenticado apenas para a página inicial.

Além dos papéis dos usuários, há quatro tipos de permissões com capacidades específicas que podem ser aplicados aos usuários do Moodle. A primeira é a *Permitir*, que concede o acesso ao usuário que recebeu essa permissão. A permissão *Impedir* remove a autorização, mesmo que o usuário com este papel tivesse permissão de acesso. *Proibir* é utilizada quando se deseja negar para um papel específico qualquer acesso que antes pudesse haver em algum contexto específico. Por último, *Não definido/herdar* é a permissão padrão para novos papéis criados, definindo para o usuário a permissão que ele já possui (MOODLE, 2017g). No contexto do trabalho, todos os usuários estão suscetíveis aos problemas de comunicação, independente do papel desempenhado e permissões concedidas.

Como o Moodle foi desenvolvido na linguagem de programação PHP, é necessário que o seu servidor hospedeiro tenha um conjunto de extensões instalados. Em resumo, sua instalação compreende três partes: o diretório */moodle*, o diretório */moodledata* e o banco de dados, todos podendo estar localizados em um mesmo servidor caso seja uma instalação básica do sistema, ou distribuídos em uma instalação com diversos servidores balanceados, servidores de arquivos e bancos de dados redundantes (BROWN; WILSON, 2012):

- Diretório */moodle*: local em que fica armazenado todo o código-fonte do Moodle. Geralmente fica armazenado como um subdiretório dos documentos do servidor Web (e.g., */var/www/html/moodle*).
- Diretório */moodledata*: Localizado fora do diretório do servidor Web, é o local no qual o Moodle armazena os seus arquivos. Estes podem ser arquivos enviados, arquivos temporários e dados de sessão de usuários (e.g., */home/moodledata*).
- Banco de dados: com mais de 250 tabelas, o banco de dados do Moodle armazena as informações dos cursos e dos usuários do sistema. O Sistema de Gerenciamento de Banco de Dados (SGBD) pode ser o MySQL, PostgreSQL, Microsoft SQL Server, Oracle ou SQLite.

2.1.2 Arquitetura e Componentes

Adaptada de Brown e Wilson (2012), a Figura 2 apresenta uma arquitetura típica do Moodle em redes acadêmicas. Com foco na modularidade e interoperabilidade com outros sistemas, o Moodle permite a integração com diversas aplicações desenvolvidas por terceiros, além de permitir o desenvolvimento de *plugins* para a sua própria plataforma a fim de prover funcionalidades específicas.

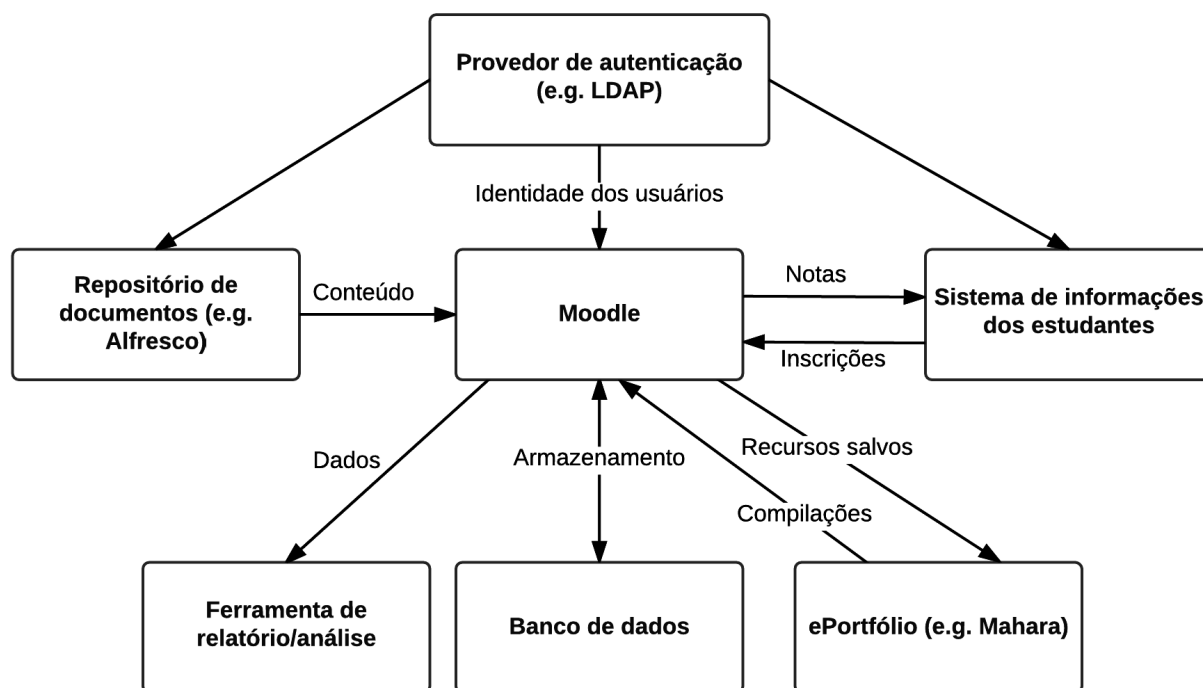


Figura 2 – Arquitetura típica do Moodle em redes acadêmicas. Adaptado de Brown e Wilson (2012).

A autenticação de usuários no Moodle pode ser realizada de diversas formas, sendo tanto interna através de usuários cadastrados manualmente no seu banco de dados ou externa, por meio de um provedor de autenticação. A forma mais comum

de autenticação se dá com o uso de um servidor LDAP (e.g., OpenLDAP e Microsoft Active Directory). Também são suportadas as autenticações por servidores IMAP, POP3 e Shibboleth, bem como outras menos usuais, como servidores NNTP, LTI e CAS (MOODLE, 2017c).

Um sistema de informações dos estudantes armazena todas as informações administrativas que os alunos têm com a instituição de ensino, tais como mensalidades, cursos que o aluno esteja realizando, disciplinas já cursadas e as restantes para conclusão dos cursos, além dos dados dos professores e suas respectivas disciplinas ministradas (BROWN; WILSON, 2012). A sua integração com o Moodle ocorre através de *scripts* que acessam os bancos de dados de ambos os sistemas, compartilhando as notas e inscrições dos alunos e professores.

Portfólios no Moodle permitem que os alunos exportem seus trabalhos para plataformas externas. Desta forma, os alunos podem comprovar que realizaram determinadas atividades práticas, além de servirem como repositório virtual de arquivos, permitindo o envio e recebimento de arquivos. Os sistemas Box.net, Flickr.com, Google Drive, Picasa e Mahara constituem exemplos de serviços de armazenamento remoto que podem ser integrados como portfólios ao sistema (MOODLE, 2017h).

Por sua vez, as ferramentas de relatório e análise recebem dados do Moodle que servem como entrada para relatórios gerenciais da instituição. Já os repositórios de documentos permitem que o usuários enviem arquivos para o Moodle através de repositórios externos. Exemplos de repositórios disponíveis para serem integrados ao Moodle são o Google Drive, YouTube, Flickr, OneDrive, Picasa, Wikimedia, Alfresco, Amazon S3, Box.net e DropBox (MOODLE, 2017i).

O código-fonte do Moodle é dividido em três componentes que se relacionam entre si de forma modular: o núcleo, subsistemas e *plugins*. Essa divisão permite que o sistema seja personalizável de acordo com as demandas das instituições de ensino, sem a necessidade de alterar o seu núcleo. A Figura 3 apresenta a comunicação entre esses componentes. Para que o sistema possa ser executado, são componentes obrigatórios o núcleo e os subsistemas. Já os *plugins*, que irão fornecer funcionalidades extras ao *software*, são opcionais.

O núcleo provê as funcionalidades para todas as outras partes do Moodle. Já subsistemas são grupos de funções e classes relacionadas que são parte do núcleo, mas são logicamente agrupadas. Como qualquer outro *software* desenvolvido em módulos, não é recomendado que se modifique diretamente o seu código-fonte, pois atualizações para novas versões do Moodle seriam impossibilitadas, uma vez que o código-fonte obrigatório (núcleo e subsistemas) foi modificado na versão anterior. Para tal, deve-se utilizar e desenvolver *plugins* para cada deficiência/limitação na

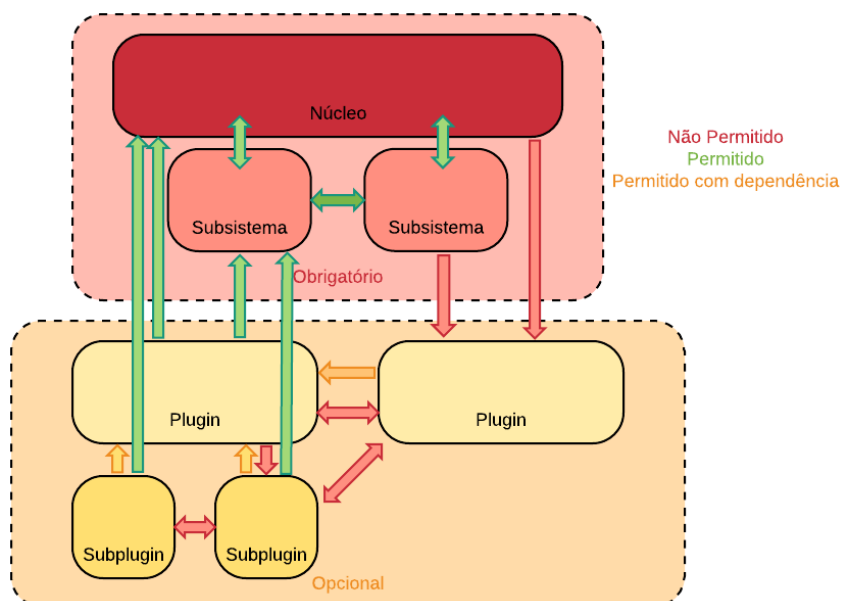


Figura 3 – Comunicação entre os componentes do Moodle. Adaptado de Moodle (2017d).

ferramenta. Os *plugins* são componentes opcionais no Moodle que estendem a sua funcionalidade e os *subplugins* estendem a funcionalidade de *plugins* já existentes (MOODLE, 2017d).

Por questões de segurança e estabilidade do sistema, a comunicação entre os componentes do Moodle é limitada de acordo com o nível de acesso que cada componente pode ter sobre os demais. As setas coloridas da Figura 3 representam os sentidos permitidos e negados da comunicação entre os componentes. As setas verdes e vermelhas correspondem aos sentidos permitidos e negados, respectivamente. Já as setas laranjas simbolizam a permissão caso haja a dependência de um *plugin* para outro ou também a dependência de um *subplugin* para um *plugin*.

Como pode ser observado, o Moodle é altamente dependente da infraestrutura de rede de computadores a qual está inserido, em que toda requisição de usuário ou resposta do servidor precisam atravessar o meio de transmissão e os seus dispositivos de rede. No entanto, não existe a efetiva comunicação entre a aplicação Moodle e a rede: o Moodle não consegue informar as suas demandas de largura de banda e qualidade de serviço necessárias para cada evento que ocorre nos seus componentes.

Eventos no Moodle são definidos como pedaços atômicos de informação que descrevem algo que aconteceu durante o seu uso, principalmente de ações de usuários, mas também de processos administrativos do LMS (MOODLE, 2017f). Para que estes eventos possam ser capturados e processados, é necessário que cada *plugin* defina quais eventos do Moodle deseja monitorar. O monitoramento é realizado através de implementação de uma classe abstrata para cada um dos eventos disponíveis.

Os detalhes de como foram capturados os eventos do Moodle são apresentados na Seção 4.5.

Todo o tráfego da aplicação é tratado como uma única massa de dados que trafega na rede, não permitindo que usuários sejam privilegiados de acordo com a importância das atividades que estejam realizando em determinados momentos. Por exemplo, não é possível que um usuário que esteja realizando uma prova tenha o seu tráfego de rede priorizado em detrimento dos demais usuários enquanto esteja realizando esta atividade. Encerrando-se a prova, essa priorização deveria ser desfeita em tempo de execução em todo o trajeto entre o dispositivo do usuário e o servidor do Moodle.

Além da necessidade de priorização de largura de banda para os usuários do Moodle, o seu desempenho pode ser impactado pelo congestionamento da rede. A Subseção 2.1.3 apresenta os resultados obtidos na análise do desempenho do serviço quando aplicados diversos níveis de tráfego secundário na rede.

2.1.3 IMPACTO DO CONGESTIONAMENTO DE REDES NO MOODLE

Por serem altamente dependentes da rede para a sua execução, aplicações executados em servidores Web têm o seu desempenho afetado por congestionamentos ocasionados em virtude da limitação de largura de banda. O Moodle, por também ser uma aplicação Web, está inserido neste contexto. Para verificar e quantificar o impacto que estes congestionamentos na rede podem causar no desempenho do Moodle, realizou-se uma investigação para constatar se pode haver perda quando ruídos são inseridos na rede ao mesmo tempo em que o Moodle é acessado por um cliente na topologia.

A topologia de rede do cenário experimental é apresentada na Figura 4: quatro *switches* interconectam cinco equipamentos na rede. O ambiente da topologia é composto de computadores distintos, com a configuração AMD Phenon II X4 com 4 GB de RAM e o sistema operacional GNU/Linux Ubuntu 14.04 instalado. sendo interconectados por um enlace com capacidade de 1 Gbps, tanto na interface de rede de computador quanto do *switch*. O *switch* raiz é um equipamento Dell, modelo N3048p, enquanto que os demais são TP-Links, modelo TL-WR1043ND com *firmware* Open-WRT (FAINELLI, 2008). Já o servidor Web utilizado foi o Apache versão 2.4.23, executando o Moodle versão 3.0.10. Estas versões dos *softwares* foram utilizadas em virtude de serem as mais atuais à época da realização dos testes.

Com duas aplicações sendo executadas na rede (o Moodle e um gerador de ruído na rede), a topologia foi elaborada de tal forma que ambas as aplicações compartilhassem o meio de transmissão, sendo o lado esquerdo composto pelos servidores

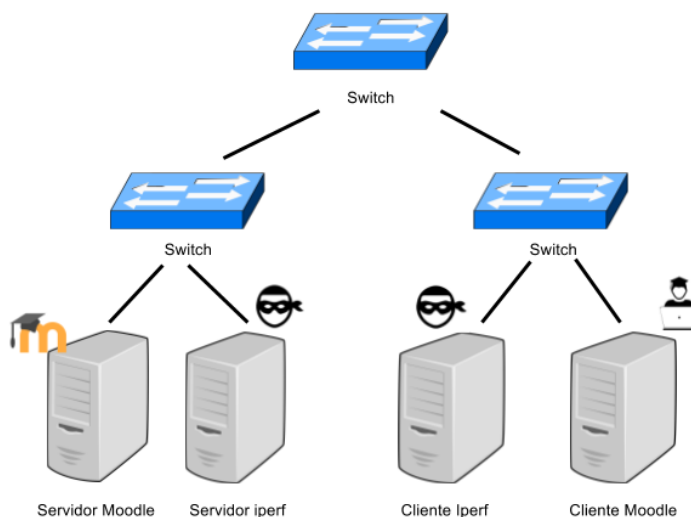


Figura 4 – Topologia para investigação sobre o desempenho do LMS Moodle na ocorrência de congestionamento.

e o lado direito pelos clientes. A geração de ruído na rede (tráfego bidirecional) foi realizada com a ferramenta *iperf* (TIRUMALA et al., 2005) nos hospedeiros identificados por Ruído 1 (servidor) e Ruído 2 (cliente). Também foi verificado que o limite físico agregado de tráfego era de 550 Mbps entre hospedeiros na rede, valor aquém do limite teórico devido à sobrecarga do *switches* de borda virtualizados (PFAFF et al., 2009).

Durante os testes foram gerados ruídos de 0 a 90% da capacidade identificada da rede. No dispositivo Servidor Web, executando o Moodle, foram disponibilizados arquivos com tamanhos de 1, 5 e 10 MB, os quais eram acessados pela ferramenta de *benchmark* ApacheBench versão 2.3 (FOUNDATION, 2018), no equipamento Cliente Web. Foram realizados testes com 1000 requisições por sessão e 50 sessões simultâneas, com a finalidade de representar uma carga de trabalho próxima à vivenciada em ambientes acadêmicos.

A Figura 5 apresenta os resultados das medições do tempo médio por requisição HTTP com um intervalo de confiança de 95%. À medida que se aumenta a concorrência pelo canal de comunicação com gargalo identificado entre o *switch* diretamente conectado ao Servidor Web (e Ruído 1) e o *switch* raiz, maior é a degradação do tempo médio do *download* dos arquivos. Para os arquivos de 10 MB a variação foi maior, sendo 240 ms com a rede exclusiva para a aplicação e 1220,5 ms com 90% de tráfego concorrente, um aumento de 481,19% no tempo da aplicação.

A variabilidade identificada no experimento é oriunda da sobrecarga computacional introduzida pelos *switches* virtualizados (PFAFF et al., 2009). A escolha dessa configuração é justificada para manter coerência com a análise experimental apresentada no Capítulo 5. Em resumo, é visível a degradação da qualidade de serviço

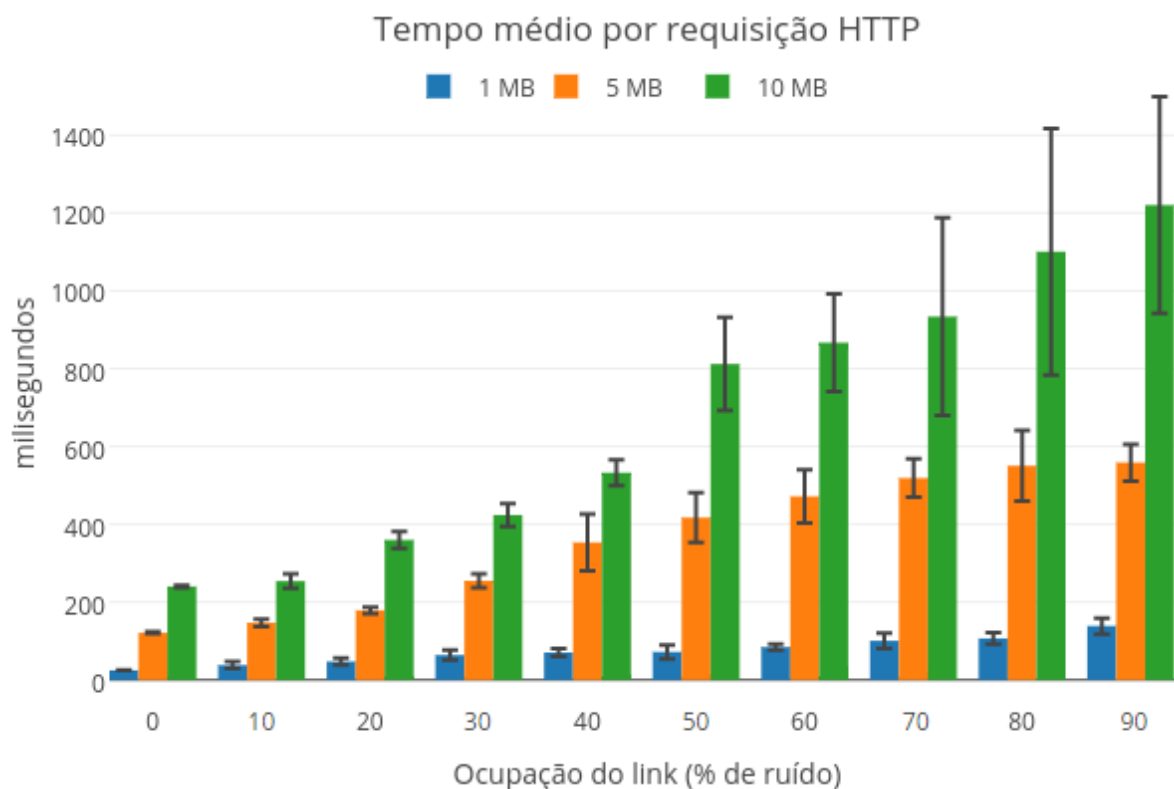


Figura 5 – Tempo médio por requisição HTTP submetido ao LMS Moodle em uma rede congestionada.

oferecida pelo Moodle na ocorrência de tráfego concorrente na rede de comunicação. Traduzindo para um cenário real, uma página do Moodle é composta por diversos arquivos com tamanhos distintos. Ou seja, um usuário perceberá, no pior caso, um atraso total correspondente ao somatório do tempo necessário para carregar todos os arquivos.

2.2 CONTROLE DE QUALIDADE DE SERVIÇO EM REDES TRADICIONAIS

Uma vez que se leva em média até dez anos para se implantar um novo protocolo de rede, o desenvolvimento e adoção de novas tecnologias de comunicação passou a ser um processo lento. Iniciativas como a do Protocolo Internet versão 6, por exemplo, que foi especificado em 1998 (DEERING; HINDEN, 1998), ainda enfrentam dificuldades para serem empregadas na sua plenitude nos roteadores da Internet.

Este cenário piora nas redes acadêmicas nas quais um usuário com perfil de pesquisador, em que é necessário ter acesso à configuração dos dispositivos de rede, tem que enfrentar conflitos de interesse para desenvolver e testar novas tecnologias. Por um lado o administrador da rede tem dificuldade de separar, configurar e entregar fatias da topologia da rede e de seus dispositivos para que o pesquisador desenvolva e teste seus protocolos sem afetar os demais usuários. De outro lado, há os fabricantes

dos equipamentos de rede que impedem que seus dispositivos sejam programados diferentemente do *software* que já vem embarcado no *hardware*. Essa inflexibilidade prejudica o aprimoramento de protocolos já existentes na rede, bem como a adoção de novos protocolos.

Na abordagem tradicional de encaminhamento, cada um dos elementos da rede deve ser configurado para ser autônomo na tomada de decisão de encaminhamento, seja este roteamento de pacotes na camada de rede ou de quadros na camada de enlace. Uma das limitações do encaminhamento tradicional é que os dispositivos só conseguem interpretar informações contidas na camada para a qual foram projetados (STALLINGS, 2015). Por exemplo, os roteadores, por definição, são capazes de examinar não mais que dados da camada de rede (*e.g.*, endereços lógicos de origem e destino). Informações sobre portas de origem e destino, aplicação geradora do tráfego ou usuário remetente do pacote não estão disponíveis para que roteadores possam encaminhar de forma mais eficiente os pacotes pela rede. Sendo assim, a aplicação de estratégias baseadas em extensão de arquivos trafegados ou encaminhamento guiado por aplicação é praticamente inviável em redes tradicionais (HANDLEY, 2006).

Mesmo que informações importantes estejam disponíveis na camada nativa do dispositivo de encaminhamento, isso não garante que essa informação possa ser utilizada para tomada de decisão durante o caminho completo. Por exemplo, alterações durante o trajeto de um quadro na rede fazem com que se perca o endereço físico do dispositivo de origem e de destino contidos no cabeçalho, sendo que a cada troca de domínio de *broadcast*, os endereços físicos dos dispositivos de origem e de destino são substituídos pelos endereços físicos da interface do *gateway* de origem e do *gateway* de destino, respectivamente. Desta forma, por exemplo, um pesquisador é impossibilitado de implementar abordagens de encaminhamento baseadas em fabricante do dispositivo (informação contida nos 24 primeiros bits do endereço *MAC*), seja ele de origem ou de destino.

Considerando o desempenho final obtido pelas aplicações hospedadas, é comum a percepção do compartilhamento justo de largura de banda implementado pelo protocolo TCP (CERF; ICAHN, 2005) (CHIU; JAIN, 1989). Em suma, na ocorrência de congestionamento em um determinado enlace, o protocolo buscará estimar um compartilhamento justo entre os fluxos, visando sempre a utilização total do enlace. Ou seja, o tráfego é limitado pelo compartilhamento (do pior caso) de largura de banda entre os fluxos passantes, independentemente do tipo de aplicação em execução.

É factível que o desenvolvimento de soluções (mecanismos, protocolos, arquiteturas, entre outros) está limitado pela presença ubíqua das redes de comunicação, aliado às limitações arquiteturais e organizações discutidas (HANDLEY, 2006). Apesar de existirem modelos que pretendem garantir QoS em redes de computadores, suas

implementações são complexas. Por exemplo, a efetiva utilização de *IntServ* (serviços integrados) (BRADEN; CLARK; SHENKER, 1994), cujo foco é a garantia de QoS para aplicações específicas, e *DiffServ* (serviços diferenciados) (GROSSMAN, 2002), que é orientado a classes de serviços, ainda é foco de discussões devido a complexidade de operacionalização. Para o escopo deste trabalho não é utilizado nenhum deles, e sim o uso do paradigma SDN.

O *DiffServ* é uma arquitetura de rede de computadores que especifica um mecanismo escalável para classificar e gerenciar tráfego de rede e fornecer QoS em redes IP modernas. Com serviços diferenciados, a rede tenta entregar um tipo particular de serviço baseado no QoS especificado por cada pacote, sendo usado para aplicações de missão crítica e fornecer QoS fim-a-fim. Como o tráfego é classificado baseado em prioridade, ele é encaminhado utilizando um dos três mecanismos *per-hop behavior* (PHB) definidos pela Internet Engineering Task Force (IETF) (YILDIRIM; GIRICI, 2014). As suas principais características são (STALLINGS, 2005): (i) os pacotes IP são rotulados para um tratamento diferente de QoS, usando o campo TOS de 6 bits nos cabeçalhos do IPv4 e do IPv6; (ii) um acordo de nível de serviço, *Service Level Agreement* (SLA), é estabelecido entre o provedor de serviço e o cliente, antes do uso do campo TOS; (iii) oferece um mecanismo de agregação embutido. Todo o tráfego com o mesmo octeto é tratado da mesma forma pelo serviço de rede; e (iv) é implementado em roteadores individuais, enfileirando e encaminhando pacotes com base no octeto TOS.

Por sua vez, os serviços integrados (*IntServ*) constituem um modelo de QoS baseado em fluxo, significando que um usuário precisa criar um fluxo (circuito virtual) da origem até o destino e informar todos os roteadores sobre os recursos necessários. A especificação de um fluxo é definida pela Rspec (especificação de recursos), que define o recurso que o fluxo precisa reservar (*buffer* ou largura de banda, por exemplo) e também pela Tspec (especificação de tráfego), a qual define a caracterização de tráfego do fluxo (FOROUZAN, 2008). O protocolo *Resource Reservation Setup Protocol* (RSVP) (*Resource Reservation Setup Protocol*) (MANKIN et al., 1997) é um protocolo de sinalização que pode ser usado para efetuar a sinalização de chamada exigida pelo *IntServ* (KUROSE; ROSS; ZUCCHI, 2007).

A tecnologia *Multi-Protocol Label Switching* (MPLS) (*Multi-Protocol Label Switching*) combina a técnica de circuitos virtuais com a funcionalidade da pilha TCP/IP, uma vez que o mesmo dispositivo de rede desempenha conjuntamente o papel de um roteador IP tradicional na camada de rede e um *switch* de circuitos virtuais na camada de enlace (OLIFER; OLIFER, 2008). Essa abordagem utiliza informações de mais de uma camada para a tomada de decisão do encaminhamento dos dados na rede. No entanto, informações da camada de aplicação continuam indisponíveis para os dispo-

sitivos de encaminhamento nas camadas inferiores. Além disso, o plano de controle não é separado do plano de dados nos dispositivos de encaminhamento.

Como o protocolo IP foi projetado para ser um protocolo do tipo melhor esforço, aplicações multimídia têm o seu desempenho afetado quando ocorrem atrasos na rede. Kurose, Ross e Zucchi (2007) defendem que o projeto de aplicações de multimídia seria certamente facilitado se houvesse classificações específicas para serviços na Internet de primeira e segunda classe. Pacotes de primeira classe teriam número limitado e receberiam serviço prioritário em filas de roteadores. No entanto, na perspectiva atual dos protocolos de rede para multimídia, ainda é presente a igualdade em relação ao escalonamento de pacotes em filas de roteadores.

2.3 REDES DEFINIDAS POR SOFTWARE

O paradigma das Redes Definidas por *Software* surgiu como uma abordagem para flexibilizar a programação e o gerenciamento das redes de computadores, facilitando a sua configuração, operação e monitoramento (BARI et al., 2013), sem a necessidade da configuração individual de cada um dos dispositivos da rede. SDN é visto como uma proposta de arquitetura para condução do tráfego, sem a rigidez imposta por cada uma das camadas do protocolo TCP/IP no momento da tomada de decisão de encaminhamento.

Tal gerenciamento maleável é possível através da separação do plano de controle (que decide como encaminhar o tráfego da rede do plano de dados (que encaminha o tráfego de acordo com a decisão tomada pelo plano de controle) da rede. As SDNs introduziram diversas inovações, sobretudo: (i) a separação do plano de controle do plano de dados; (ii) a centralização lógica do plano de controle; (iii) a programabilidade do plano de controle; e (iv) a padronização das interfaces de programação de aplicações (API, do inglês *Application Programming Interface*) (JAIN; PAUL, 2013).

2.3.1 Planos de Controle, Dados e Gerenciamento

Considerando que o plano de dados é composto pelos pacotes que precisam ser encaminhados de acordo com as regras de encaminhamento definidas pelo plano de controle, o controle lógico da rede pode ser implementado em um equipamento separado, denominado controlador. Com isso, de forma logicamente centralizada, é possível programar o plano de controle para que regras de encaminhamento sejam aplicadas de forma mais flexível e dinâmica em todos os elementos da SDN. Com o uso de SDNs, os elementos de rede têm a capacidade de interpretar dados das camadas superiores para as quais foram desenvolvidos, tornando o encaminhamento mais flexibilizado quando comparado com os modelos tradicionais de encaminhamento.

Com a separação dos planos de dados e de controle, são dois os aspectos-chaves que envolvem uma SDN. O primeiro é a abstração do encaminhamento baseado em fluxos, que permite a programação do plano de dados com a utilização de uma API aberta. O segundo é a abstração centralizada do plano de controle, permitindo que aplicações de rede possam efetuar amplas operações na rede sem precisar configurar individualmente cada um dos elementos da rede (MEKKY et al., 2014). Surge o papel do controlador SDN, responsável por configurar as tabelas de encaminhamento de fluxos nos elementos da rede. A decomposição dos planos, combinada com um conhecimento completo da topologia de rede por parte do controlador, permite o desenvolvimento de algoritmos específicos para encaminhamento guiado pela aplicação.

A Figura 6 apresenta uma visão simplificada da arquitetura SDN. Em cada um dos elementos de encaminhamento de dados é implementado somente o plano de dados no *hardware*. Estes elementos podem ser equipamentos padrão da indústria, tais como roteadores, *switches* ou pontos de acesso sem fio. Como o plano de controle é gerenciado por outra entidade da rede (o controlador SDN) e não há mais a distinção do tipo de encaminhamento que é realizado, são comumente chamados de *switches* SDN, por conseguirem realizar tanto encaminhamento baseado nas informações da camada 2 quanto das camadas 3 e 4 da arquitetura TCP/IP.

Ainda, a Figura 6 introduz as aplicações de rede como parte da visão simplificada da arquitetura. Os requisitos das aplicações são expressos na arquitetura e constituem o plano de gerenciamento de dados. Nesse plano, os controladores são configurados de acordo com os requisitos de cada aplicação comunicante, informando quais os requisitos de comunicação (*e.g.*, encaminhamento por caminho mais curto, distribuição de carga, reserva de largura de banda).

No contexto do presente trabalho, a aplicação de rede alvo é o LMS Moodle. Informações extraídas da aplicação são utilizadas para configurar o comportamento do controlador SDN. Quando o controlador recebe requisições oriundas dos *switches* SDN sobre o procedimento a ser adotado com determinados pacotes, traduz os requisitos da aplicação em regras de encaminhamento que são armazenadas nas tabelas de encaminhamento dos *switches* intermediários.

2.3.2 Interfaces de Gerenciamento

Com a separação entre o plano de controle e o plano de dados, a comunicação dos *switches* SDN com o controlador ocorre através de uma API, pela qual o controlador SDN exerce controle direto sobre o estado dos *switches* (KREUTZ et al., 2015). Apesar do controlador SDN ser apresentado na Figura 6 como um único equipamento gerenciador da rede, este número pode variar de acordo com as necessidades de

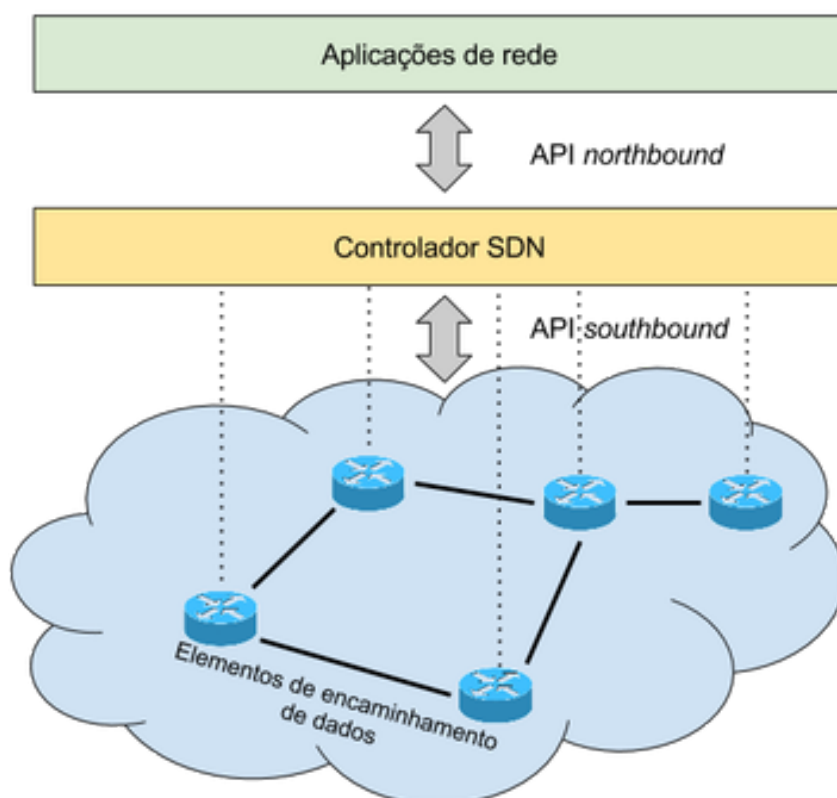


Figura 6 – Visão simplificada de uma arquitetura SDN. Adaptado de Kreutz et al. (2015).

desempenho e redundância exigidos pela arquitetura.

A *API southbound* se refere à interface e ao protocolo entre os *switches* SDN programáveis e o controlador. O protocolo OpenFlow é a interface *southbound* padrão *de facto* adotada para a comunicação entre o plano de dados e o plano de controle (YAO; BI; XIAO, 2011). Já a *API northbound* determina como expressar as tarefas operacionais e políticas de rede, além de traduzí-las de forma que o controlador SDN possa entender (KIM; FEAMSTER, 2013). O escopo principal do presente trabalho está na interface *northbound*, na qual se propõe converter ações que ocorrem em uma aplicação em parâmetros e regras de encaminhamento que possam melhorar métricas de desempenho da aplicação com a ajuda da arquitetura SDN.

A separação do plano de dados do plano de controle permite que novas formas de encaminhamento sejam implementadas nas redes SDN. Entretanto, nativamente na interface *southbound*, a definição de critérios de roteamento é limitada aos cabeçalhos das camadas 2 até 4. Para que uma aplicação na rede possa ter o roteamento dos seus dados baseado no seu comportamento de tráfego, é necessário explorar a interface *northbound*. Assim, uma aplicação pode avisar o controlador SDN sobre início e fim de tráfegos específicos (*e.g.*, rajadas, quadros jumbo), bem como acerca de eventos que possam ser utilizados como parâmetro para que o controlador

crie políticas de roteamento e também de QoS específicas para a aplicação Moodle.

2.3.3 Fluxo de Dados

Uma vez que no cabeçalho de um pacote SDN há mais informações disponíveis do que em um quadro Ethernet, há a possibilidade de mapear de forma mais flexível as necessidades de uma aplicação em políticas de roteamento na rede. Por exemplo, o fluxo do tráfego do Moodle na rede pode ser definido como uma tupla <endereço IP, porta>, enquanto que esta mesma abordagem pode ser utilizada para mapear os seus usuários.

Desta forma, a interface *northbound* torna-se responsável, através do controlador SDN, por entregar para a interface *southbound* as informações das aplicações que serão utilizadas como parâmetros do roteamento dos dados na rede. É na interface *northbound* que se concentrará o escopo deste trabalho, fazendo com que informações particulares da aplicação alvo possam ser utilizadas para melhorar métricas de desempenho do Moodle.

2.3.4 Tabelas de Fluxo

Em SDN, cada *switch* é equipado com uma ou mais tabelas de fluxo, utilizadas para o repasse de pacotes no plano de dados. As tabelas são usualmente implementadas com memórias que permitem acesso aleatório e concorrente. Enquanto em redes tradicionais o encaminhamento de pacotes para portas de saída é normalmente realizado considerando endereços MACs e IPs, em SDN as tabelas de fluxos podem ser indexadas por tuplas arbitrariamente definidas pelo controlador. Ou seja, ao receber um pacote, a porta de destino e ações aplicáveis são identificadas através de uma pesquisa combinado de informações das camadas 2, 3 e 4 do modelo TCP/IP.

É importante ressaltar que a maleabilidade introduzida por SDN aumenta o volume de entradas armazenadas em tabelas de fluxo. Assim, o controlador é responsável por gerenciar as entradas (cadastrar, atualizar e eventualmente excluir, ou indicar um tempo máximo de vida). Ainda, o gerenciamento de dados em tabelas de fluxo pode ocorrer reativamente ou proativamente (VIEIRA; FIORESE; KOSLOVSKI, 2016). Um controlador pode aguardar a solicitação individualmente de cada *switch*, respondendo com a configuração correspondente (reativo), ou cadastrar as entradas das tabelas de fluxo em todos os *switches* que compõem o caminho (pró-ativo).

A complexidade de implementação é reduzida na abordagem reativa, entretanto, o atraso percebido pelo utilizador do sistema aumenta. Em suma, na configuração reativa é necessária a interferência do controlador a cada salto. No contexto do presente trabalho essa situação é indesejada, ou seja, a abordagem pró-ativa é

utilizada.

2.4 PROTOCOLO OPENFLOW

Proposto por McKeown et al. (2008), o protocolo OpenFlow se tornou padrão *de facto* para API *southbound* SDN. Ele foi recentemente padronizado pela *Open Networking Foundation (ONF)*, com o objetivo de permitir que pesquisadores testem novos protocolos na própria rede de produção, sem ter a necessidade de utilizar equipamentos de uma rede separada.

O protocolo OpenFlow torna menos rígida a programação da rede, através da introdução de uma camada extra de *software* no dispositivo, simultaneamente removendo o seu plano de controle. Implementando as especificações do OpenFlow, o fabricante permite que pesquisadores utilizem seus equipamentos para embarcar novos protocolos de rede, conseguindo ainda manter a programação do plano de dados. Aliado a um controlador *Software-Defined Networking (SDN)*, o administrador da rede pode conceder partições da topologia para usuários específicos através de políticas próprias, baseadas em fluxos.

Os três principais componentes que envolvem o protocolo OpenFlow são o *switch* OpenFlow, o canal seguro entre o *switch* e o controlador OpenFlow e o próprio controlador OpenFlow. A Figura 7 apresenta a interação dos elementos de um *switch* OpenFlow em uma rede *Software-Defined Networking (SDN)*. Diferentemente de uma rede tradicional, na qual a unidade de dados de protocolo da camada de acesso é um quadro e este pode entrar ou sair por qualquer uma das portas de um *switch*, a comunicação entre *switches* OpenFlow ocorre através de pacotes, ou seja, informações de camadas superiores a camada 2 podem ser analisadas pelos *switches* para encaminhamento (FOUNDATION, 2015).

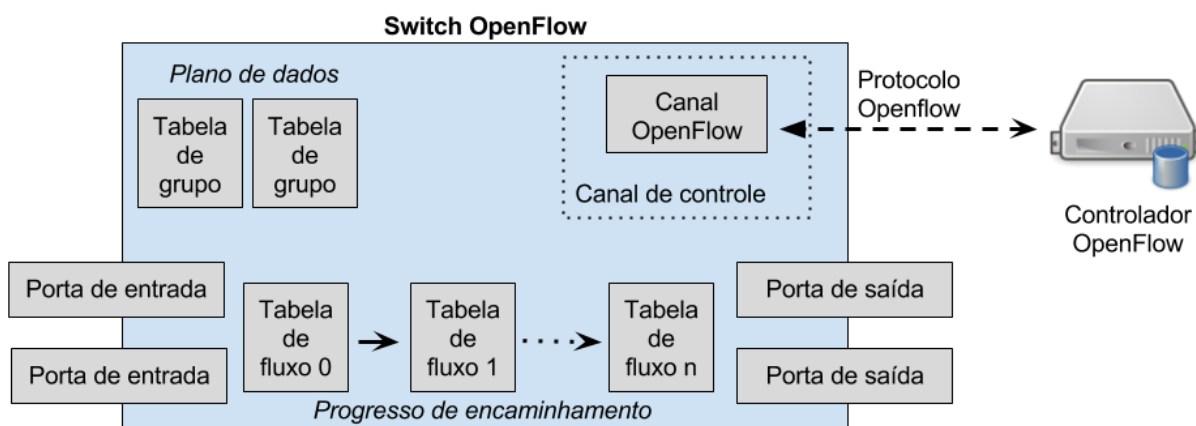


Figura 7 – Switch OpenFlow. Adaptado de McKeown et al. (2008).

Um *switch* OpenFlow possui três tipos de portas, podendo ter uma porta física, lógica ou reservada. Uma porta física geralmente é o mapeamento de uma interface do *hardware* do dispositivo de rede. Já portas lógicas são abstrações de alto nível que podem ser definidas no *switch*, que podem não corresponder a uma interface do *hardware*, como por exemplo um grupo de agregação de enlaces, túneis e interfaces de *loopback* (FOUNDATION, 2015).

Um fluxo OpenFlow é um pacote formado por uma tupla com 12 componentes: porta de entrada, endereço físico de origem, endereço físico de destino, tipo Ethernet, id da VLAN, prioridade da VLAN, IP de origem, IP de destino, protocolo IP, bits de ToS IP, porta TCP/UDP de origem e porta TCP/UDP de destino. Com esse atributos, um *switch* OpenFlow tem a capacidade de rotear pacotes desde a camada 2 até a camada 4. Assim é possível que fluxos caracterizados por informações de diferentes camadas sejam cadastrados em tabelas de fluxos específicas.

No progresso de encaminhamento da Figura 7, cada tabela de fluxo, numerada de 0 até n , é composta por 6 atributos (campos de combinação, prioridade, contadores, instruções, *timeouts* e *cookies*). Os campos de combinação consistem dos cabeçalhos do pacote, da porta de entrada e também de metadados especificados por uma tabela anterior. A prioridade é a precedência dos fluxos da tabela no encaminhamento. Os contadores atualizam os pacotes correspondentes. Já as instruções modificam o conjunto de ações ou o processo de encaminhamento. *Timeout* é o tempo máximo antes do fluxo expirar no *switch*. Por fim, *cookies* são utilizados pelo controlador para filtrar estatísticas dos fluxos. Por sua vez, a tabela de grupo consiste de entradas que permitem o encaminhamento agrupado de fluxos, representando políticas adicionais, como *multicast* por exemplo (FOUNDATION, 2015).

2.5 TRABALHOS RELACIONADOS

Foram encontrados na literatura trabalhos que investigaram o uso do paradigma de SDNs para melhorar o desempenho de aplicações em contextos variados. A Tabela 1 apresenta uma síntese do levantamento realizado. Para cada referência, são identificadas as informações sobre a aplicação e ambiente de execução, bem como a estratégia de encaminhamento adotada. Ainda as métricas de desempenho, utilizadas para identificar a aplicabilidade das soluções, são elencadas e analisadas.

Egilmez et al. (2012) implementaram um controlador para entrega de multimídia com suporte de QoS fim a fim. Para os autores, os requisitos de QoS foram traduzidos como atraso de encaminhamento, que foi minimizado através do monitoramento da utilização dos enlaces e reorganização das tabelas de encaminhamento.

Seguindo essa linha, Jeong, Kim e Kim (2012) propuseram um sistema ope-

Referência	Informações sobre a aplicação ou ambiente	Estratégia de encaminhamento	Métrica de desempenho
Egilmez et al. (EGILMEZ et al., 2012)	Monitoramento do uso dos enlaces	Definição dos caminhos mais curtos	Atraso de encaminhamento
Jeong et al. (JEONG; KIM; KIM, 2012)	Um SLA é estabelecido indicando as métricas	Um módulo de monitoramento avalia a disponibilidade dos recursos	Níveis de QoS definidos pelo usuário
Bari et al. (BARI et al., 2013)	Monitoramento ativa e passiva do uso da topologia	Quando ocorre violação de métricas de QoS, o controlador é acionado	Níveis de QoS definidos pelo usuário
Jarschel et al. (JARSCHEL et al., 2013)	Tamanho e influência dos <i>buffers</i> de vídeo	Inspeção de pacotes	Quantidade de paradas do vídeo
Mekky et al. (MEKKY et al., 2014)	Pacotes são interceptados entre o controlador e <i>switches</i> OVS	Regra de <i>table miss</i> indica que uma ação é necessária	Atraso na propagação dos pacotes
Watashiba et al. (WATASHIBA et al., 2014)	Largura de banda consumida e topologia de rede	Caminho mais curto em uma topologia <i>fat-tree</i>	Tempo de processamento
Marcondes et al. (MARCONDES et al., 2016)	Caracterização do tráfego de rede das aplicações	Comparação de estratégias de roteamento	Tempo de execução das aplicações
Wang et al. (WANG et al., 2016)	Proposta do paradigma de rede dirigida por aplicação (ADN)	A rede ADN é fatiada em várias sub-redes lógicas isoladas	Provê QoS sob demanda para diferentes aplicações
Zhao et al. (ZHAO; SYDNEY; MEDHI, 2016)	Estado da aplicação e requisitos de recursos dos nós de um cluster Hadoop	Abordagem de rede de aplicação para gerenciar o tráfego de rede do Hadoop distribuído	Tempo de execução
Avila et al. (AVILA; DIAS; SIMOES, 2018)	Utilização de CPU, memória e bytes de entrada/saída	Balanceamento de carga e elasticidade sob demanda	Taxa de perda de requisição

Tabela 1 – Resumo dos trabalhos relacionados com a presente proposta.

racional de rede para SDN guiado por requisitos de QoS informados durante o estabelecimento de um SLA. A implementação é baseada no controlador NOX (GUDE et al., 2008), que nativamente não oferece essas funcionalidades.

Um *framework* aberto de gerenciamento de políticas de QoS é apresentado em Bari et al. (2013). Este contempla um módulo de monitoramento (com métricas ativas e passivas), guiado por métricas definidas pelo administrador da aplicação. Ao ocorrer eventos que violem as métricas de QoS estipuladas, o controlador é acionado.

O uso do paradigma das redes definidas por *software* para melhorar o desempenho de aplicações que utilizam a rede de computadores já é uma realidade. Aplicações multimídia, como o YouTube (JARSCHEL et al., 2013), já conseguem ter o seu desempenho aprimorado com o uso de SDN e entregam uma experiência para usuário mais satisfatória do que somente quando utilizadas técnicas tradicionais de QoS. Os autores implementaram um módulo para monitorar o uso *buffer* de armazenamento de vídeos. Essa informação é enviada para o controlador SDN, responsável por priorizar o tráfego quando o *buffer* não contém dados suficientes para sustentar a execução do vídeo com qualidade, ou quando uma alta variação de fluxo é observada. Para os autores, o *buffer* representa a qualidade experimentada pelo usuário.

Uma arquitetura SDN estendida foi proposta por Mekky et al. (2014) para permitir a inspeção de informações além das que estão disponíveis entre as camadas 2 e 4 da pilha de protocolos TCP/IP. Pacotes classificados como *table miss* (ou seja, sem entrada correspondente nas tabelas de fluxos) são interceptados entre os *switches* OVS e o controlador OpenFlow. Após a interceptação, eles são analisados e a decisão de encaminhamento ocorre através da consulta de uma tabela auxiliar representando informações da aplicação (superiores à camada 4). Desta forma, a arquitetura permite que a maior parte do processamento de rede ocorra no plano de dados, reduzindo a necessidade de enviar pacotes para o controlador da rede.

Aplicações para processamento de alto desempenho podem ser beneficiadas por controladores especializados. Watashiba et al. (2014) propuseram um sistema para inicialização de tarefas em aglomerados. Os autores exploraram o conhecimento sobre a topologia do aglomerado (*fat-tree* (AL-FARES; LOUKISSAS; VAHDAT, 2008)) para otimizar o tráfego de dados entre os processos. Cada conjunto de aplicações da suíte de *benchmark* NAS representou um usuário do aglomerado, entretanto nenhuma informação foi extraída das aplicações.

Marcondes et al. (2016) caracterizaram o tráfego de rede das aplicações do *benchmark* NAS (BAILEY et al., 1991) em uma SDN. Também foram comparados diferentes algoritmos de roteamento de fluxos, mostrando que aplicações podem ter o seu tempo de execução reduzido quando aplicado roteamento de caminhos múltiplos.

Wang et al. (2016) propuseram um novo paradigma de rede, o *Application Drive Network* (ADN), que provê serviços diferenciados sob demanda para aplicações. A rede física é fatiada em diversas sub-redes lógicas isoladas, onde cada fatia tem a sua própria arquitetura e protocolo para servir a uma aplicação de forma exclusiva. A proposta é melhorar a experiência do usuário enquanto mantém a eficiência dos recursos, através de multiplexação entre as sub-redes isoladas.

Zhao, Sydney e Medhi (2016) desenvolveram aplicações SDN para aglomerados Hadoop (SHVACHKO et al., 2010) que consideram a descoberta de topologias de rede, monitoramento de tráfego e roteamento de fluxos para mecanismos de anulação de *loops*. Foi proposta uma abordagem de rede de aplicação ciente para gerenciar o tráfego de rede do Hadoop distribuído, fornecendo a capacidade da rede manter as informações sobre as aplicações que se conectam a ele. Foi utilizado SDN para fornecer uma implementação baseada em *software* para permitir que aplicações individuais e dispositivos de rede fossem controlados.

Por fim, Avila, Dias e Simoes (2018) propõem uma arquitetura adaptada para o Moodle integrada à computação em nuvem, com o reaproveitamento do hardware. Utilizando as informações do uso de CPU, memória e *bytes* de entrada e saída do servidor virtualizado, provisiona recursos sob demanda na rede para a criação de novos servidores. Ao alcançar determinado índice do uso de CPU do servidor virtual do Moodle, instancia-se nova máquina virtual para atender esta nova necessidade de tráfego (elasticidade positiva), bem como também exclui estas instâncias quando a carga é reduzida (elasticidade positiva).

Os trabalhos relacionados mostram que aplicações podem ter suas métricas de desempenho melhoradas com o uso do paradigma das Redes Definidas por *Software*. No entanto, para melhorar estas métricas, estes trabalhos ou modificaram o protocolo OpenFlow ou atendem apenas a uma única aplicação, dificultando o seu uso para outras aplicações em redes com o protocolo OpenFlow padrão habilitado.

Sendo assim, se pretende com este trabalho aperfeiçoar e priorizar o encaminhamento do tráfego de rede de um sistema de gestão de aprendizagem, de modo proativo, baseado em eventos da aplicação. A contribuição deste trabalho é a possibilidade de se realizar roteamento de rede baseado em fluxos de dados dos usuários da aplicação alvo, onde serão levados em consideração eventos para tomada de decisão de roteamento, evitando assim que tráfegos de rede alheios ao que deve ser priorizado interfiram no seu desempenho.

2.6 CONSIDERAÇÕES PARCIAIS

Com o avanço das tecnologias de rede, a configuração e gerenciamento de uma rede de computadores torna-se cada vez mais complexa, uma vez que tráfegos de diferentes protocolos precisam ser transportados por um mesmo dispositivo de rede. Desta forma, um pesquisador que precise utilizar a rede para desenvolver e testar novos protocolos, pode enfrentar dificuldades em ter acesso a fatias desta rede sem interferir no tráfego de outros usuários. Além disso, os fabricantes de dispositivos de rede impedem que o seu *hardware* seja programado com *software* diferente do que já vem embarcado nas suas soluções.

O paradigma das SDNs permitiu a separação do plano de dados do plano de controle da rede. Com essa separação, o administrador da rede consegue permitir que o pesquisador utilize as frações dos dispositivos de rede, sem afetar o tráfego de outros usuários. O escopo deste trabalho é utilizar o paradigma das redes definidas por *software* para que o roteamento dos fluxos na rede seja guiado pelos eventos que ocorrem na aplicação alvo Moodle.

Foi realizada uma investigação com o objetivo de verificar o impacto que a concorrência pelo canal de comunicação pode causar no desempenho das aplicações em uma rede de computadores, no qual foi utilizada uma topologia de rede e duas aplicações executadas simultaneamente. A primeira foi um servidor Web acessado por clientes simultaneamente para o *download* de arquivos com tamanhos diferentes (aplicação alvo) e a segunda foi o gerador de tráfego *iperf* (tráfego secundário). Os resultados mostraram que à medida que a ocupação da rede aumenta com tráfego secundário, maior é a degradação do desempenho da aplicação alvo.

O Capítulo 3 aborda a arquitetura da ferramenta proposta para controle do tráfego do Moodle, discutindo como SDN e o protocolo OpenFlow podem ser utilizados para melhorar a qualidade de serviço oferecida à aplicação.

3 SDN4MOODLE: FERRAMENTA PARA CONTROLE DO TRÁFEGO LMS

Este capítulo descreve a arquitetura proposta da SND4Moodle, uma ferramenta para controle e priorização do tráfego da aplicação Moodle em uma SDN. A Seção 3.1 discute os requisitos funcionais e não funcionais da ferramenta. Posteriormente, a Seção 3.2 apresenta uma visão geral do cenário da topologia de rede, enquanto os módulos são individualmente detalhados nas Seções 3.3 e 3.4. A comunicação entre os módulos é detalhada na Seção 3.5.

3.1 REQUISITOS PARA EXECUÇÃO

No processo do desenvolvimento deste trabalho, uma das etapas realizadas foi a do levantamento dos requisitos para a elaboração e execução da ferramenta SDN4Moodle em uma topologia de rede com o paradigma SDN. Tal levantamento é importante para que seja possível mapear e priorizar as necessidades do sistema como um todo, antes mesmo do início do seu desenvolvimento. Foram identificados tanto os requisitos funcionais quanto os não funcionais necessários para a sua construção. Os pré-requisitos são apresentados na Seção 3.1.1, enquanto os requisitos funcionais são apresentados na Seção 3.1.2 e os requisitos não funcionais na Seção 3.1.3.

3.1.1 Pré-Requisitos

Para aplicação da ferramenta proposta, alguns pré-requisitos devem ser observados no ambiente de execução, abaixo elencados:

- **Suporte ao protocolo OpenFlow.** A ferramenta proposta é baseada no protocolo OpenFlow e conseqüentemente exige total aderência por parte dos equipamentos. Em especial, é necessário suporte completo à versão 1.3 do protocolo (funcionalidade indicadas como obrigatórias).
- **Disponibilidade de *switches* SDN com capacidade de controle das tabelas de métricas.** Diversos equipamentos implementam parcialmente as funcionalidade indicadas como opcionais pelo protocolo OpenFlow. Para efetiva aplicação do controle de rede proposto, é essencial a existência de recursos nos *switches* para gerenciamento das tabelas de métricas.
- **Controlador SDN centralizado.** A arquitetura de comunicação da ferramenta proposta prevê a comunicação com apenas um controlador SDN centralizado na rede, não tendo a capacidade de trabalhar com múltiplos controladores.

3.1.2 Requisitos Funcionais

Os requisitos funcionais da ferramenta proposta foram levantados e priorizados de acordo com as exigências encontradas na necessidade de se desenvolver uma ferramenta que consiga priorizar tráfego específico dos usuários do Moodle. Essas ações devem ser realizadas de forma dinâmica, sem a necessidade da intervenção do administrador da rede para criar regras de QoS na rede. Os requisitos funcionais são resumidos na Tabela 2.

Tabela 2 – Requisitos funcionais.

	Descrição	Prioridade
RF01	Cadastrar, atualizar e remover fluxos dos usuários do Moodle nos <i>switches</i> da SDN	Alta
RF02	Cadastrar, atualizar e remover métricas nos <i>switches</i> da SDN	Alta
RF03	Capturar os eventos da navegação dos usuários no Moodle	Alta
RF04	Identificar a topologia de rede atualizada da SDN	Alta
RF05	Alocar largura de banda para os usuários conforme suas atividades	Alta
RF06	Monitorar a largura de banda disponível nos <i>switches</i>	Baixa
RF07	Criação de uma <i>dashboard</i>	Alta
RF08	Permitir a parametrização do sistema através do banco de dados	Alta

Fonte: produção do próprio autor.

O primeiro requisito funcional da aplicação SDN4Moodle, RF01, é a necessidade da ferramenta de cadastrar, atualizar e remover os fluxos dos usuários do Moodle nos *switches* da SDN. Essas atividades devem ser executadas através de ações enviadas ao controlador SDN da rede, que por sua vez as envia para os *switches*. Dessa forma é possível que tráfego dos usuários sejam categorizados e priorizados.

O segundo requisito funcional (RF02) diz respeito à necessidade da ferramenta permitir que métricas OpenFlow sejam cadastradas e removidas em tempo de execução nos *switches* da SDN. As métricas são fundamentais para que as operações de QoS sejam implementadas na topologia, controlando a largura de banda disponibilizada para os usuários do Moodle em cada *switch* entre o usuário e o servidor em tempo de execução.

Por sua vez, o terceiro requisito funcional, RF03, refere-se à captura e armazenamento dos eventos da navegação dos usuários do Moodle. Como não foi encontrado no diretório oficial do Moodle um *plugin* que faça essa tarefa, é necessário o desenvolvimento dessa funcionalidade na ferramenta. A SDN4Moodle utiliza os eventos para criar regras de QoS dinâmicas, de acordo com as atividades que os usuários estiverem realizando no Moodle.

Para que a ferramenta consiga aplicar regras de encaminhamento e priorização de tráfego nos *switches* da rede, é necessário que se conheça a topologia de rede atualizada da SDN (RF04). Quando um novo *switch* ou computador de usuário for inserido ou removido da rede, a ferramenta tem que identificar em tempo de execução essa mudança e já considerar para as suas próximas regras de encaminhamento a topologia de rede atualizada.

A largura de banda disponibilizada para cada usuário deve ser computada de acordo com as atividades que esteja realizando no Moodle (RF05). Caso um usuário se autentique e decida realizar uma prova ou assistir a um vídeo assíncrono, imediatamente uma nova reserva e configuração de largura de banda deve ser disponibilizada em todo o meio de transmissão entre o equipamento do usuário e o servidor do Moodle. O valor da largura de banda alocada é definida pelo administrador do sistema.

O sistema deverá monitorar a largura de banda utilizada e disponível. Essas informações podem servir como critério de entrada para a tomada de decisão do encaminhamento do tráfego entre as rotas possíveis (RF06). Ainda, para que o administrador possa monitorar as ações executadas pela ferramenta, é necessário que haja à sua disposição uma *dashboard* de gerenciamento (RNF07). As informações mínimas que devem estar disponíveis são:

- Histórico de eventos ocorridos no Moodle.
- *Status* da conexão dos dispositivos da topologia da rede.
- Histórico das ações enviadas ao controlador SDN.
- Largura de banda alocada para cada usuário na rede.

Para permitir uma maior integração com trabalhos futuros e a padronização no acesso aos seus dados, o sistema deve permitir que toda a sua parametrização seja armazenada em banco de dados (RF08). Na sequência, a Subseção 3.1.3 apresenta os requisitos não funcionais da ferramenta proposta neste trabalho.

3.1.3 Requisitos Não Funcionais

Além dos requisitos funcionais do sistema, também foram definidos os seus requisitos não funcionais. Estes estão ligados às características de desempenho e qualidade da ferramenta SDN4Moodle proposta e são elencados na Tabela 3.

Para que as regras de encaminhamento possam ser aplicadas nos dispositivos da rede de forma satisfatória, o sistema precisa manter em sua memória a representação da topologia da SDN. Para evitar que ações sejam aplicadas em dispositivos que

Tabela 3 – Requisitos não funcionais.

	Descrição	Prioridade
RNF01	Tempo de atualização da topologia	Alta
RNF02	O sistema deve poder ser executado na maior quantidade de plataformas possíveis	Alta

Fonte: produção do próprio autor.

já foram desconectados da rede ou deixar de aplicar em equipamentos recém adicionados, deve haver no seu núcleo de *software* uma *thread* que atualize a topologia a cada 1 segundo (RNF01).

Ainda, propondo evitar limitações na quantidade possível de plataformas aptas a executarem a ferramenta, o sistema deve ser implementado de tal forma que seja possível a sua execução na maior capacidade disponível de arquiteturas de computadores (RNF02). Sendo assim, deverá ser verificada qual é a linguagem de programação mais apta a atender esse requisito não funcional, de acordo com as limitações de prazo de desenvolvimento deste trabalho e curva de aprendizado da linguagem de programação.

3.2 COMPONENTES DA ARQUITETURA E CENÁRIO DE APLICAÇÃO

Desde o *login* de um usuário até o término das atividades no sistema, toda a interação no sistema Moodle ocorre através da rede de computadores. Esta seção apresenta o cenário ao qual esta pesquisa foi desenvolvida, bem como a topologia de rede utilizada no plano de testes da ferramenta. A Figura 8 ilustra uma visão geral dos módulos que compõem a SND4Moodle. Para que seja possível priorizar o tráfego de rede dos seus usuários em tempo de execução, foram desenvolvidos dois módulos neste trabalho, representados pelos Módulos 1 e 2.

Todo o tráfego de rede entre os navegadores dos usuários e o Moodle ocorre na rede OpenFlow, na qual são criados fluxos à medida que novos usuários se autenticam na ferramenta. A reserva de largura de banda de cada usuário é alocada de forma dinâmica, variando de acordo com qual tipo de atividade está sendo realizada. Atividades que exigem maior tráfego de rede, como a exibição de vídeos, têm maior largura de banda alocada do que outras que não têm tal necessidade, como a execução de um áudio. Já a comunicação entre os componentes da ferramenta SDN4Moodle ocorre fora da rede OpenFlow (ou seja, sem reserva de recursos), através de tráfego administrativo da SDN.

O Módulo 1 é um *plugin* do Moodle que foi desenvolvido para ser responsável por avisar o controlador SDN da rede sobre os eventos que ocorrem durante o uso da aplicação. O seu desenvolvimento foi necessário por não ter sido encontrado

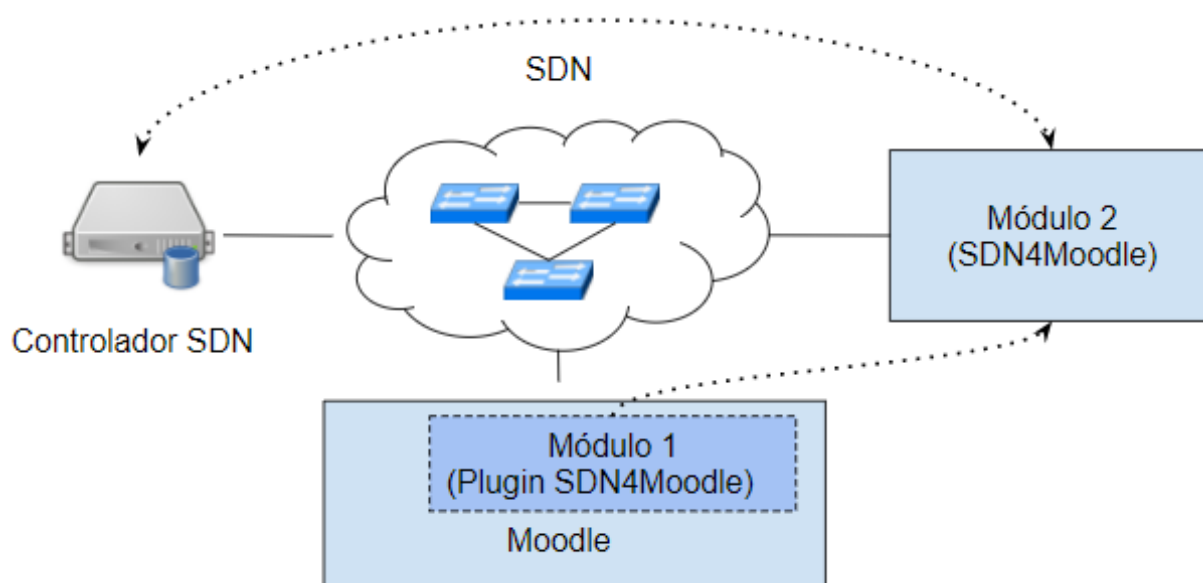


Figura 8 – Visão geral dos módulos que compõem SDN4Moodle e seu cenário de aplicação.

nenhum *plugin* disponível no diretório oficial do Moodle que fizesse a coleta dos eventos ocorridos na ferramenta e armazenasse em um banco de dados para posterior manipulação. Eventos que podem ser monitorados pelo Módulo 1 compreendem as seguintes ações:

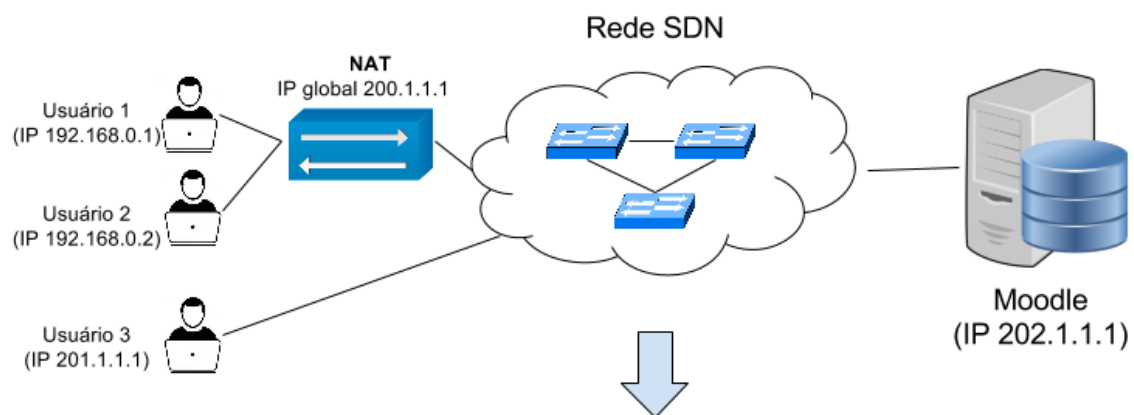
- Início e fim de sessão de usuário.
- Início e término de atividades (exercícios ou provas).
- Visualização de cursos pelos usuários.
- Visualização do *dashboard* do Moodle.
- *Download* de arquivos.
- Execução de fluxo de mídia de áudio e vídeos assíncronos.

Já o Módulo 2 é a aplicação de controle foco deste trabalho, a qual foi desenvolvida para ser executada em um hospedeiro separado da SDN. À medida que eventos no Moodle ocorrem, a aplicação remodela os fluxos OpenFlow para melhorar os valores das métricas de desempenho. Para tal, a aplicação consulta o estado da rede no controlador e envia configurações que devem ser implantadas na topologia. O funcionamento dos Módulos 1 e 2 são detalhados nas Seções 3.3 e 3.4, respectivamente.

3.3 PLUGIN NO MOODLE - MÓDULO 1

Para que dados de eventos do Moodle sejam enviados para a aplicação de controle SDN4Moodle, é necessário o desenvolvimento de um módulo de *software* no Moodle. De acordo com a revisão da Seção 2.1, o Moodle é um sistema modular que pode ser customizado para atividades específicas. Desta forma, através das informações fornecidas por este módulo, o controlador OpenFlow pode continuamente melhorar o encaminhamento do tráfego de usuários guiado por eventos oriundos da aplicação, através das ações de criação e remoção de fluxos enviadas pela SDN4Moodle.

Quando o usuário inicia uma sessão no Moodle, o *plugin* (Módulo 1) avisa a SDN4Moodle sobre os eventos. Esta, por sua vez, envia para o controlador ações para criar tabelas de fluxos para cada usuário em todos os *switches* OpenFlow entre o hospedeiro do cliente até o servidor do Moodle. Os critérios para um fluxo de dados ser interpretado como sendo da aplicação é a tupla <endereço IP origem, porta de origem, endereço IP destino, porta TCP destino>. Na modelagem dos fluxos, o endereço IP de destino e a porta são sempre os mesmos por se tratar de uma única aplicação, sendo necessário diferenciar os usuários através da sua porta e endereço IP de origem. Neste caso, a porta de origem é necessária para permitir que usuários oriundos de uma rede com tradução de endereços (NAT, do inglês *Network Address Translation*) possam ser vistos como tráfegos distintos.



Modelagem de fluxos de usuários				
Usuário	Endereço de origem	Porta de origem	Endereço de destino	Porta de destino
Usuário 1	200.1.1.1	50000	202.1.1.1	80
Usuário 2	200.1.1.1	50001	202.1.1.1	80
Usuário 3	201.1.1.1	50000	202.1.1.1	80

Figura 9 – Modelagem dos fluxos dos usuários do Moodle.

A Figura 9 apresenta um exemplo de modelagem dos fluxos do tráfego de rede do Moodle. Em virtude do NAT, os usuários 1 e 2 compartilham o mesmo endereço IP

global (200.1.1.1). Para que o controlador SDN consiga formar os fluxos dos usuários do Moodle, é necessário que tanto endereço quanto porta de origem sejam identificados. Como o endereço de origem é o mesmo, os fluxos são especificados pela porta de origem (portas 50000 e 50001, respectivamente). Sendo assim, o fluxo do usuário 1 do Moodle é identificado como a tupla <200.1.1.1, 50000, 202.1.1.1, 80>, enquanto que do usuário 2 é <200.1.1.1, 50001, 202.1.1.1, 80>. Já o usuário 3, que não compartilha o seu endereço IP, tem o seu fluxo especificado por <201.1.1.1, 50000, 202.1.1.1, 80>.

Tabela 4 – Exemplos de melhoria da aplicação alvo.

Atividade	Fluxo (Identificação)	Objetivo (QoS)	Forma
Prova	Endereço e porta de origem	Garantia de largura de banda	Cadastro de largura de banda mínima no fluxo do usuário na <i>meter table</i> e priorização do fluxo durante a realização da prova.
Streaming de áudio	Endereço e porta de origem	Garantia de largura de banda	Cadastro de largura de banda mínima no fluxo do usuário na <i>meter table</i> e priorização do fluxo durante a execução do áudio.
Vídeos	Endereço e porta de origem	Garantia de largura de banda	Cadastro de largura de banda mínima no fluxo do usuário na <i>meter table</i> e priorização do fluxo durante a execução dos vídeos.

Modelados os tráfegos da aplicação Moodle, são definidas as métricas a serem melhoradas na aplicação. O foco principal é proporcionar ao usuário uma experiência de navegação mais satisfatória quando comparado com o encaminhamento tradicional, sem ser guiado pela aplicação. Em virtude da percepção que um usuário tem de uma aplicação ser abstrata e complexa a sua medição, a intenção deste trabalho é a utilização de regras de QoS tradicionais combinadas com os eventos que ocorrerem na aplicação. As três principais atividades que são melhoradas são os eventos de provas, *streaming* de áudios e vídeos assíncronos.

A Tabela 4 apresenta um resumo das melhorias que podem ser observadas no desempenho das atividades da aplicação. Com o apoio do Módulo 1 (*plugin* do Moodle), os fluxos das atividades podem ser caracterizadas como pacotes com endereço e porta de origem específicos. Assim, a aplicação de controle configura proativamente a SDN, tomando decisões quando novos eventos ocorrerem no Moodle. Por exem-

plo, para uma prova, o objetivo de QoS é a garantia de largura de banda para esta atividade, no qual é realizado o cadastro de largura de banda mínima em cada um dos *switches* entre o usuário e o servidor do Moodle, assim que o Módulo 1 avisar a aplicação de controle que houve o início de uma prova.

Já para *streaming* de áudio pode-se levar em consideração a reserva de largura de banda para a sua execução, além da priorização do fluxo do usuário. Por fim, vídeos podem ser classificados de forma estática em três categorias pelo administrador da rede: baixa demanda de rede, média demanda e alta demanda. Eles podem ser categorizados de acordo com a regra tempo do vídeo vs tamanho do arquivo. Uma forma de melhorar o tráfego de vídeos assíncronos na rede é a garantia de largura mínima para os vídeos dos usuários.

3.4 APLICAÇÃO DE CONTROLE - MÓDULO 2

Em conjunto com o *plugin* do Moodle, a aplicação de controle SDN4Moodle tem como foco entregar informações dos eventos do Moodle para o controlador OpenFlow. Com estas informações, o controlador irá cadastrar regras de fluxo nos *switches* SDN. A primeira função da aplicação de controle é a de coletora de eventos do *plugin* do Moodle. São coletados os eventos de *login* e *logout* de usuários no Moodle, bem como início de atividades na aplicação. Atividades podem ser o início de uma prova, o *streaming* de um áudio ou o *download* de um arquivo de vídeo. Esses eventos são utilizados como parâmetros para roteamento guiado pela aplicação.

A principal atividade da aplicação de controle é a de informar ao controlador OpenFlow como devem ser caracterizados e roteados os fluxos dos usuários na rede. Ainda, realiza coletas dos contadores presentes em cada um dos *switches* OpenFlow da rede. As informações dos contadores são utilizadas para calcular de forma proativa o roteamento na rede. Além disso, o administrador da rede pode informar parâmetros adicionais tais como características dos vídeos (baixa, média e alta demanda de rede), autenticação do Moodle e do controlador OpenFlow.

3.5 COMUNICAÇÃO ENTRE OS MÓDULOS DA SDN4MOODLE

A ferramenta SDN4Moodle foi idealizada para ser executada de forma modularizada, permitindo, em trabalhos futuros, o desenvolvimento de novos módulos. A concepção em formato de *framework* (com escopo limitado à aplicação Moodle) possibilita até mesmo que outros controladores OpenFlow sejam utilizados para gerenciar o plano de controle da SDN.

A primeira interação que o usuário tem com o Moodle é a execução do *login* no navegador na página de autenticação. A partir desse momento a rede prioriza

largura de banda através da criação de fluxos nos *switches* OpenFlow. Para que essa priorização ocorra, é necessário que o Moodle (através do *plugin* desenvolvido neste trabalho) informe a SDN4Moodle sobre os eventos que ocorrem na aplicação. Esses eventos são armazenados em um banco de dados, que é compartilhado pelo *plugin* e pela ferramenta.

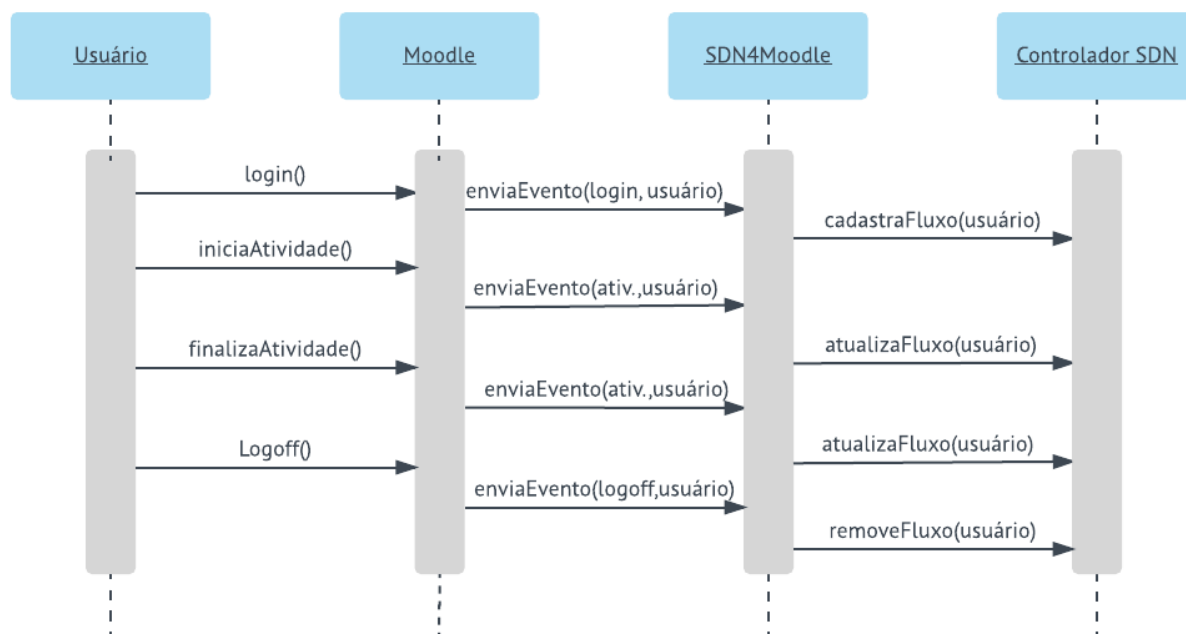


Figura 10 – Diagrama de sequência dos eventos dos usuários no Moodle e na ferramenta SDN4Moodle.

A Figura 10 apresenta o diagrama de sequência da interação do usuário com Moodle e deste com a ferramenta SND4Moodle. Os eventos recebidos pela SDN4Moodle são examinados por um monitor da ferramenta, que processa novos eventos a cada segundo. Quando a SDN4Moodle recebe um evento de autenticação de usuário, a ferramenta envia para o controlador uma ação para que este crie dois fluxos específico para este usuário. O primeiro fluxo se refere do tráfego com sentido para o servidor Moodle, enquanto que o segundo fluxo tem sentido para o cliente do LMS. Eventos que ocorrem durante a execução de atividades dos usuários, como a realização de provas, execução de áudios ou a exibição de vídeos são analisados pela ferramenta de forma a atualizar o fluxo deste usuários.

3.6 CONSIDERAÇÕES PARCIAIS

A disputa por recursos de comunicação entre aplicações ocorre em diversos cenários. Especificamente em ambientes acadêmicos, o desempenho do sistema Moodle é influenciado pelo tráfego oriundo de aplicações não acadêmicas, como por exemplo Facebook, YouTube, Spotify, entre outros.

Assim, este capítulo apresentou uma arquitetura, denominada SDN4Moodle, para gerenciar os fluxos Moodle em redes SDN. A arquitetura compreende as diversas atividades que podem ser realizadas pelos alunos, como realização de provas e estudos guiados com vídeos e áudios. Em resumo, dois módulos são essenciais para coletar os eventos do Moodle e gerenciar o cadastro dos fluxos de dados nos *switches*.

São apresentados alguns pré-requisitos para o funcionamento da ferramenta proposta, tais como a disponibilidade de *switches* SDN, controlador SDN centralizado e suporte ao protocolo OpenFlow versão 1.3. A necessidade da versão 1.3 se dá em virtude da mudança substancial que ocorreu com as novas versões do protocolo OpenFlow. No Capítulo 4, um protótipo experimental é apresentado, indicando como a arquitetura pode ser desenvolvida em um ambiente real.

4 IMPLEMENTAÇÃO DA FERRAMENTA SDN4MOODLE

Este capítulo apresenta os detalhes do desenvolvimento do protótipo da ferramenta SDN4Moodle. Inicialmente, a Seção 4.1 revisa o controlador SDN selecionado, enquanto a Seção 4.2 aponta como os requisitos foram atendidos. A Seção 4.3 discute como foi definida a identificação de um fluxo de usuário na rede, para permitir que os dispositivos de encaminhamento possam priorizá-los. A estrutura do banco de dados é exposto na Seção 4.4. As implementações dos Módulos 1 e 2 são exibidos nas Seções 4.5 e 4.6. Na sequência, o plano de testes e as considerações são apresentados nas Seções 4.7 e 4.8, respectivamente.

4.1 CONTROLADOR SDN

Inicialmente, foi necessária a escolha de um controlador SDN específico para o desenvolvimento da ferramenta SDN4Moodle. Para tal, realizou-se um mapeamento dos controladores compatíveis com os dispositivos disponíveis no LabP2D. Foi verificado que para um dos equipamentos disponíveis, um *switch* DELL N3048p, o único controlador SDN homologado pelo fabricante era o OpenDayLight (ODL) (DELL, 2017a). Como os demais equipamentos do laboratório também eram compatíveis com o este controlador, o ODL foi selecionado para o escopo deste trabalho.

OpenDayLight (ODL) é um controlador desenvolvido e mantido pela OpenDayLight Fundation, uma associação sem fins lucrativos que visa o desenvolvimento de um controlador robusto e padronizado. Quanto à plataforma interna, ODL possui um mecanismo modular (MEDVED et al., 2014). Especificamente, o desenvolvimento de SDN4Moodle foi baseado nos módulos disponíveis para descoberta de recursos e cadastro de regras de encaminhamento. Para interagir com ODL, SDN4Moodle utilizou as *Application Programming Interfaces* (APIs) padronizadas e chamadas *Representational State Transfer* (REST). A implementação da SDN4Moodle é dependente do ODL, dado que foram utilizados formatos de dados específicos deste controlador.

O cadastro de fluxos e métricas no ODL, para posterior envio para os *switches* da rede através das chamadas REST, é realizado através de entradas no formato *Extensible Markup Language* (XML) ou *JavaScript Object Notation* (JSON). Para o escopo deste trabalho foi escolhido o formato XML para ambos os cadastros. O formato dos fluxos e métricas é apresentado na Seção 4.3.

4.2 ADERÊNCIA AOS REQUISITOS

Os requisitos funcionais e não funcionais identificados nas Seções 3.1.2 e 3.1.3, respectivamente, foram atendidos conforme indicado pela Tabela 5. Em resumo, os requisitos funcionais foram atendidos pelos Módulos 1 e 2 da ferramenta, enquanto a padronização dos dados (e acesso), combinada com o protocolo OpenFlow, facilitou a concordância com os requisitos não funcionais. Os detalhes de implementação são discutidos nas seções subsequentes, indicando detalhes sobre a configuração dos módulos.

Tabela 5 – Atendimento aos requisitos funcionais e não funcionais na implementação de SDN4Moodle.

Requisito	Implementação (e.g., ferramenta utilizada ou protocolo)
RF01 (fluxos)	Módulo 2 de SDN4Moodle. Baseado no protocolo OpenFlow versão 1.3 (completo).
RF02 (métricas)	Módulo 2 de SDN4Moodle. Baseado no protocolo OpenFlow versão 1.3 (completo).
RF03 (eventos)	Módulo 1 de SDN4Moodle. Desenvolvido como um <i>plugin</i> para o Moodle.
RF04 (identificar topologia)	Módulo 2 de SDN4Moodle. Baseado na API do controlador ODL.
RF05 (largura de banda)	Módulos 1 e 2 de SDN4Moodle. Atendido combinando as chamadas OpenFlow com controle de dados internos em SDN4Moodle.
RF06 (monitorar largura de banda)	Não foi implementado (baixa prioridade). Indicado como trabalhos futuros.
RF07 (<i>dashboard</i>)	Disponibiliza visualização dos eventos e configurações cadastradas por SDN4Moodle.
RF08 (parametrização)	Pode ser cadastrado diretamente no banco de dados ou através da interface administrativa.
RNF01 (atualização da topologia)	Módulos 1 e 2 de SDN4Moodle. Cada ação de um usuário do Moodle é tratada como uma <i>thread</i> individual.
RNF02 (plataformas)	Atendido através do desenvolvimento através das linguagens de programação PHP (Módulo 1) e Java (Módulo 2), da padronização do acesso ao banco de dados e pelo uso do protocolo OpenFlow.

Fonte: produção do próprio autor.

4.3 IDENTIFICAÇÃO DE UM FLUXO

Para que seja possível identificar o tráfego dos usuários na SDN, é necessário que sejam utilizados os atributos disponíveis no cabeçalho do pacote. Dado que são lançadas periodicamente novas versões do protocolo OpenFlow, cada uma com as suas particularidades, foi realizado um levantamento para analisar quais as versões

disponíveis nos dispositivos do Laboratório de Processamento Paralelo e Distribuído (LabP2D) da UDESC Joinville. Verificou-se que todos os dispositivos implementavam a versão 1.3 do OpenFlow.

Assim, a modelagem dos fluxos da ferramenta foi elaborada de acordo com a especificação desta versão. Conforme discutido previamente (Seção 2.4), um pacote OpenFlow é uma tupla composta por 12 componentes, representados pela Figura 11. Deste total, são analisados sete itens (células sombreadas) como campos de combinação na tabela de fluxos para identificar o tráfego de um usuário do Moodle.

Porta de entrada	Endereço físico de origem	Endereço físico de destino	Tipo Ethernet	ID da VLAN	Prioridade da VLAN	IP de origem	IP de destino	Protocolo IP	Bits de ToS IP	Porta TCP/UDP de origem	Porta TCP/UDP de destino
------------------	---------------------------	----------------------------	---------------	------------	--------------------	--------------	---------------	--------------	----------------	-------------------------	--------------------------

Figura 11 – Componentes do cabeçalho de um pacote OpenFlow analisados por SDN4Moodle.

Pacotes na rede que preencham os requisitos das marcações de porta de entrada, tipo Ethernet, endereços IP de origem e destino, número do protocolo IP e portas TCP/UDP de origem e destino são elencados em cada uma das tabelas de fluxos dos *switches*. A porta de entrada é a interface no *switch* da origem do tráfego e varia para cada dispositivo. No escopo deste trabalho, o tipo Ethernet é o 2048, que representa o protocolo IP versão 4 nos números da IEEE 802. Como para cada usuário é necessário o cadastro de dois fluxos em cada *switch* da rede (representando os tráfegos de ida e volta), os endereços de origem e de destino consistem do endereço do dispositivo do usuário e o servidor do Moodle na ida (*upload* para o servidor do Moodle), respectivamente, e o contrário na volta (*download* do servidor Moodle).

Visto que a informação do número de porta pertence à camada de transporte no modelo TCP/IP e a proposta deste trabalho é a priorização de tráfego identificado a nível de usuário somente na camada de aplicação (*e.g.*, dados HTTP na porta TCP 80 do servidor Moodle), se faz necessário identificar de qual usuário é o tráfego a ser priorizado sem precisar realizar a leitura de cada cabeçalho HTTP nos *switches*. Esta sobrecarga adicional pode acabar inviabilizando o uso do paradigma das SDNs para melhorar o desempenho de aplicações que utilizam a rede para trafegar dados.

Para este trabalho, a proposta foi a utilização (junto com os componentes porta de entrada, tipo Ethernet e protocolo IP) da tupla <IP de origem, IP de destino, porta TCP de destino> para a ida ou <IP de origem, porta TCP de origem, IP de destino> para a volta. Destaca-se a impossibilidade do uso das portas de origem e de destino simultaneamente na tupla, uma vez que em uma mesma sessão de usuário seja possível que o seu dispositivo abra e feche diversas conexões TCP com o servidor do Moodle, em portas diferentes, durante a sua navegação.

A Tabela 6 apresenta a abstração do objeto fluxo de usuário na ferramenta

Tabela 6 – Atributos de um fluxo de usuário na ferramenta SDN4Moodle.

Atributo	Descrição
<i>Priority</i>	Prioridade do fluxo
<i>FlowName</i>	Nome do fluxo
<i>InstallHW</i>	Instalar o fluxo no <i>hardware</i>
<i>HardTimeOut</i>	Tempo de expiração do fluxo no <i>switch</i>
<i>IdleTimeOut</i>	Tempo de inatividade antes da expiração do fluxo
<i>Cookie</i>	Utilizado para filtros e modificações de fluxos
<i>EthernetType</i>	Protocolo de <i>payload</i> do quadro Ethernet
<i>InPort</i>	Interface de entrada do fluxo
<i>IpProtocol</i>	Número do protocolo IP
<i>OutputActionMaxLength</i>	Tamanho máximo do pacote OpenFlow
<i>OutputNodeConnector</i>	Interface de saída do fluxo
<i>Ipv4Source</i>	Endereço IP de origem
<i>TcpSourcePort</i>	Porta TCP de origem
<i>Ipv4Destination</i>	Endereço IP de destino
<i>TcpDestinationPort</i>	Porta TCP de destino
<i>FlowID</i>	ID do fluxo
<i>TableID</i>	ID da tabela
<i>NodeID</i>	ID do <i>switch</i> da rede
<i>Meter</i>	Métrica do fluxo
<i>Bandwidth</i>	Largura de banda reservada para o fluxo

SDN4Moodle. Além dos atributos presentes por padrão na especificação do protocolo OpenFlow versão 1.3 (e.g., prioridade do fluxo, *timeouts*, *cookies*), foram adicionados componentes extras para controles específicos na ferramenta, como por exemplo o identificador do *switch*, no qual o fluxo está instalado (*NodeID*) e também qual a largura de banda reservada (*bandwidth*).

A Figura 12 apresenta um exemplo de fluxo enviado pela SDN4Moodle para o controlador SDN. Trata-se de um fluxo específico para identificar tráfego de outras aplicações não classificadas anteriormente. A sequência do processamento dos fluxos no protocolo OpenFlow se dá através das suas prioridades (atributo *priority*), nos quais fluxos cadastrados com prioridades menores são processados primeiro.

Com uma prioridade 1001 (linha 3 da Figura 12), este fluxo é processado depois que os com menor prioridade. Como regras de combinação foram definidas o tipo Ethernet 2048 na linha 12 (protocolo IP versão 4) e a interface 1 como entrada do pacote no *switch*. Como instruções de saída, foi estabelecido o envio do pacote para a interface 2 (linha 32) e a aplicação da métrica 12 como regra de largura de banda para este fluxo (linha 23).

A prioridade dos fluxos (linha 3) foi definida de acordo com a preferência que os pacotes dos usuários do Moodle e do tráfego secundário devem ser processados nos *switches*, no qual menor o número, maior é a prioridade. Num total de 9 priorida-

```

1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <flow xmlns="urn:opendaylight:flow:inventory">
3   <priority>1001</priority>
4   <flow-name>flow-others1</flow-name>
5   <installHw>true</installHw>
6   <hard-timeout>0</hard-timeout>
7   <idle-timeout>0</idle-timeout>
8   <cookie>55</cookie>
9   <match>
10    <ethernet-match>
11      <ethernet-type>
12        <type>2048</type>
13      </ethernet-type>
14    </ethernet-match>
15    <in-port>1</in-port>
16  </match>
17  <id>10001</id>
18  <table-id>0</table-id>
19  <instructions>
20    <instruction>
21      <order>0</order>
22      <meter>
23        <meter-id>12</meter-id>
24      </meter>
25    </instruction>
26    <instruction>
27      <order>1</order>
28      <apply-actions>
29        <action>
30          <order>0</order>
31          <output-action>
32            <output-node-connector>2</output-node-connector>
33            <max-length>65535</max-length>
34          </output-action>
35        </action>
36      </apply-actions>
37    </instruction>
38  </instructions>
39 </flow>

```

Figura 12 – Exemplo de fluxo enviado pela SDN4Moodle para o controlador SDN

des, a Tabela 7 mostra os valores de cada um dos tipos de tráfego.

As métricas nos *switches* são elementos que medem e controlam a taxa de pacotes trafegadas (FOUNDATION, 2015). Elas são utilizadas para alterar a alocação de largura de banda dos usuários em tempo de execução, à medida que novos eventos são registrados no Moodle e enviados para a SDN4Moodle. Com o propósito de padronização e facilidade da interpretação do código-fonte para o desenvolvimento de trabalhos futuros, a nomenclatura utilizada para os atributos de todas as classes foi a mesma que a utilizada pelo controlador OpenDayLight.

Tabela 7 – Prioridades dos tipos de fluxos.

Ação	Prioridade
Login	0
Atividades de texto	1
Execução de <i>streaming</i> de áudio	2
Execução de vídeos de baixa resolução	3
Execução de vídeos de média resolução	4
Execução de vídeos de alta resolução	5
Demais ações realizadas no Moodle	6
Tráfego secundário servidor → cliente	1001
Tráfego secundário cliente → servidor	1002

```

1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <meter xmlns="urn:opendaylight:flow:inventory">
3   <meter-id>12</meter-id>
4   <meter-name>others</meter-name>
5   <container-name>container-others</container-name>
6   <flags>meter-kbps</flags>
7   <meter-band-headers>
8     <meter-band-header>
9       <band-burst-size>0</band-burst-size>
10      <band-id>0</band-id>
11      <band-rate>100</band-rate>
12      <drop-burst-size>0</drop-burst-size>
13      <drop-rate>100</drop-rate>
14      <meter-band-types>
15        <flags>ofpmbt-drop</flags>
16      </meter-band-types>
17    </meter-band-header>
18  </meter-band-headers>
19 </meter>

```

Figura 13 – Exemplo de métrica enviada pela SDN4Moodle para o controlador SDN

Um exemplo de métrica enviada para o controlador SDN é descrito na Figura 13. Com identificador único número 12 (linha 3), ela foi definida para que fluxos cadastrados com esta métrica permitam o tráfego máximo de 100 kbps (linhas 13 e 6, respectivamente). Caso esta taxa seja alcançada em algum fluxo, a ação padrão é a de descarte de pacotes (linha 15). Já os demais atributos das métricas não se aplicam ao escopo deste trabalho.

4.4 ARMAZENAMENTO DE DADOS

O projeto do banco de dados da ferramenta SDN4Moodle foi elaborado de acordo com o levantamento de requisitos apresentado na Seção 3.1. Um ponto considerado na sua implementação foi a possibilidade de expansão do banco de dados em trabalhos futuros, permitindo novas funcionalidades na ferramenta. Para tal, foi uti-

lizado o próprio mecanismo do banco de dados da aplicação Moodle, que instala e atualiza o banco de dados através de arquivos XML.

O sistema de gerenciamento de banco de dados escolhido foi o MySQL Community Server versão 5.7.20, instalado no servidor do Moodle, sem redundância. Para que os dados necessários para a SDN4Moodle, localizados no banco de dados, não atravessassem o meio de transmissão da SDN, foi empregada uma segunda interface de rede no servidor Moodle, conectada diretamente à rede de tráfego administrativo da SDN.

Totalizando 7 tabelas, o banco de dados armazena os eventos dos usuários do Moodle e também as configurações da ferramenta proposta neste trabalho. A apresentação destas tabelas foi dividida em três colunas, sendo a primeira o nome do atributo, a segunda o seu tipo (inteiro ou texto) e por último a descrição do atributo, sendo que o atributo precedido de asterisco representa a chave primária da tabela. O Diagrama Entidade Relacionamento da ferramenta está disponível no Anexo A.

Tabela 8 – Eventos monitorados na ferramenta Moodle.

Atributo	Tipo	Descrição
<i>*ID</i>	int	Identificador único (chave primária)
<i>EventName</i>	text	Nome do evento
<i>Component</i>	text	Componente gerador do evento
<i>Action</i>	text	Tipo de ação envolvida
<i>Target</i>	text	Alvo no qual a ação é tomada
<i>ObjecTable</i>	text	Tabela que representa o objeto do evento
<i>ObjectID</i>	int	ID do registro do objeto na <i>ObjectTable</i>
<i>Crud</i>	text	Operação de criação, leitura, atualização ou exclusão
<i>EduLevel</i>	int	Nível educacional no evento
<i>ContextID</i>	int	ID do contexto
<i>ContextLevel</i>	int	Nível do contexto (se é um curso, atividade, etc.)
<i>ContextInstanceID</i>	int	ID ou categoria de curso
<i>UserID</i>	int	ID do usuário
<i>CourseID</i>	int	ID do curso
<i>RelatedUserID</i>	int	Usuário ao qual a ação está relacionada
<i>Anonymous</i>	int	Verifica se a ação foi de usuário não autenticado
<i>TimeCreated</i>	int	Horário de ocorrência do evento
<i>IpAddress</i>	text	Endereço IP do usuário

A Tabela 8 apresenta a tabela Eventos no banco de dados, que armazena todos os 17 atributos gerados pela API de eventos do Moodle que são entregues para o Módulo 1. Dessa forma, caso haja alguma necessidade futura de informações sobre os eventos, estes já estariam sendo armazenados pelo banco de dados da ferramenta. São armazenados, por exemplo, o endereço IP do usuário, o horário do evento, o ID do usuário e o curso envolvido no evento. A lista completa dos atributos dos eventos é apresentada na Seção 4.5.

Tabela 9 – Organização da fila de eventos.

Atributo	Tipo	Descrição
<i>*ID</i>	int	Identificador único (chave primária)
<i>LastProcessedEventID</i>	int	Identificador do último evento processado
<i>LastProcessedEventTimeCreated</i>	int	Horário do último evento processado

A fila de eventos é necessária para que eventos ocorridos antes do início da ferramenta SDN4Moodle, ou durante uma eventual perda de conexão com o controlador SDN, possam ser devidamente processados. A Tabela 9 mostra os atributos armazenados, que é o último evento processado pela ferramenta e o seu respectivo horário de ocorrência.

Para este trabalho foi especificado que a parametrização da SDN4Moodle seria armazenada toda em banco de dados. Essa parametrização foi dividida em três tabelas, sendo elas a *GeneralConfiguration*, que armazena os ajustes globais da ferramenta, a *FlowConfiguration* para os parâmetros dos fluxos e a *MetricConfiguration* para as métricas.

Tabela 10 – Configuração geral da ferramenta SDN4Moodle.

Atributo	Tipo	Descrição
<i>*ID</i>	int	Identificador único (chave primária)
<i>Enabled</i>	int	Habilita o <i>plugin</i> Módulo 1
<i>MoodleIp</i>	text	Endereço IP do servidor Moodle
<i>MoodlePort</i>	int	Porta do servidor Moodle
<i>ControllerIp</i>	text	Endereço IP do controlador SDN
<i>ControllerPort</i>	int	Porta do controlador SDN
<i>ControllerUser</i>	text	Nome de usuário do controlador SDN
<i>ControllerPassword</i>	text	Senha do usuário do controlador SDN
<i>Bandwidth_login</i>	int	Largura de banda para <i>login</i>
<i>Bandwidth_text</i>	int	Largura de banda para texto
<i>Bandwidth_streaming</i>	int	Largura de banda para <i>streaming</i> de áudio
<i>Bandwidth_video_low</i>	int	Largura de banda para vídeos de baixa resolução
<i>Bandwidth_video_normal</i>	int	Largura de banda para vídeos de média resolução
<i>Bandwidth_video_high</i>	int	Largura de banda para vídeos de alta resolução
<i>Timeldle</i>	int	Tempo de inatividade para exclusão do fluxo

A tabela com os atributos para configuração geral da ferramenta é apresentada na Tabela 10. Esta é utilizada para armazenar os parâmetros definidos pelo administrador da rede, tais como endereço IP, porta, usuário e senha para acesso ao controlador SDN. Também é possível habilitar e desabilitar via banco de dados o *plugin* do Módulo 1 no Moodle.

Foram definidas 6 configurações de larguras de banda para as ações realizadas no Moodle pelos usuários, sendo que cada uma é alocada para o usuário em todos os *switches* entre o seu dispositivo e o servidor do Moodle. Caso ocorra um

evento com reserva diferente da já aplicada, é realizada uma nova alocação de largura de banda nos dispositivos.

A primeira reserva de largura de banda ocorre imediatamente após o *login* do usuário (*atributo Bandwidth_login*). A partir deste momento o usuário já têm o seu tráfego priorizado na rede em detrimento dos demais aplicativos. Caso o usuário decida realizar uma prova, uma nova largura de banda é alocada (*atributo Bandwidth_text*). Para a execução de arquivos multimídia também foram definidos campos específicos, sendo para arquivos de áudio o *Bandwidth_streaming* e para vídeos de baixa, média e alta resolução, *Bandwidth_video_low*, *Bandwidth_video_normal*, *Bandwidth_video_high*, respectivamente. *Timeldle* define o tempo de inatividade do usuário do Moodle para que os seus fluxos sejam removidos dos *switches* da SDN.

Tabela 11 – Configuração de fluxos em SDN4Moodle.

Atributo	Tipo	Descrição
*ID	int	Identificador único (chave primária)
FlowTable	int	Número da tabela de fluxos
HardTimeout	text	Tempo para exclusão do fluxo
IdleTimeout	int	Tempo de inatividade para exclusão do fluxo
Cookie	int	<i>Cookie</i>
EthernetType	int	Tipo Ethernet
IpProtocol	int	Número do protocolo IP
OutputActionNodeConnector	text	Interface de saída

A configuração padrão dos fluxos do tráfego dos usuários do Moodle é definida na tabela *FlowConfiguration*, apresentada na Tabela 11. São armazenados o número da tabela de fluxos (*FlowTable*), o tempo para expiração do fluxo (*HardTimeout*), o tempo de inatividade para exclusão do fluxo (*IdleTimeout*), o *cookie*, o tipo Ethernet (*EthernetType*), o número do protocolo IP (*IpProtocol*) e a interface de saída (*OutputActionNodeConnector*).

Tabela 12 – Configuração das métricas para reserva de largura de banda.

Atributo	Tipo	Descrição
*ID	int	Identificador único (chave primária)
<i>MeterID</i>	int	ID da métrica
<i>MeterName</i>	text	Nome da métrica
<i>ContainerName</i>	text	Nome do container
<i>DropRate</i>	int	Limite de largura de banda

As configurações das métricas da ferramenta são apresentadas na Tabela 12. Além do número de identificação (*MeterID*) e nome (*MeterName*), o principal atributo da tabela é o de largura de banda máxima para o fluxo (*DropRate*). Caso o fluxo cadastrado com esta métrica exceda esta velocidade, os pacotes são descartados. A utilização de métricas com velocidade máxima para os fluxos nos *switches* foi a

estratégia de controle de congestionamento utilizada para o escopo deste trabalho, uma vez que o emprego de filas para cadastro de largura mínima não está disponível nos dispositivos com o protocolo OpenFlow habilitado.

Tabela 13 – Definição de prioridades de páginas.

Atributo	Tipo	Descrição
<i>*ID</i>	int	Identificador único (chave primária)
<i>Page</i>	int	ID da página
<i>Priority</i>	int	Prioridade da página

Como cada tipo de recurso recebe um nível de priorização, as tabelas *Page-Priority* (Tabela 13) e *ResourcePriority* (Tabela 14) armazenam as prioridades de cada item. São necessárias duas tabelas em virtude de, por exemplo, uma atividade de texto ser tanto uma página quanto um recurso. Para cada uma das tabela é armazenado o atributo (*Page* ou *Resource*) e a sua respectiva prioridade (*Priority*).

Tabela 14 – Configuração da prioridades para recursos.

Atributo	Tipo	Descrição
<i>*ID</i>	int	Identificador único (chave primária)
<i>Resource</i>	int	ID do recurso
<i>Priority</i>	int	Prioridade do recurso

Apresentado o detalhamento do projeto do banco de dados da ferramenta SDN4Moodle, a próxima seção irá mostrar a descrição da implementação do Módulo 1, *plugin* do Moodle responsável por capturar os eventos dos seus usuários durante a execução de atividades no LMS.

4.5 IMPLEMENTAÇÃO DO MÓDULO 1

Como não foi encontrado no diretório oficial de *plugins* do Moodle um *plugin* que capturasse e armazenasse em banco de dados os eventos dos usuários, foi necessário o desenvolvimento de uma ferramenta que para realizar esta atividade. De modo que também não é possível que a aplicação Moodle avise diretamente a ferramenta desenvolvida SDN4Moodle, esta foi separada em dois módulos, conforme discutido no Capítulo 3: os Módulos 1 e 2. O Módulo 1 é um *plugin* com a função de armazenar os eventos ocorridos no Moodle.

Utilizando a linguagem de programação PHP versão 7.0, o desenvolvimento do Módulo 1 foi realizado conforme as diretrizes de elaboração de *plugins* do Moodle a partir da versão 3.0. Esta versão do Moodle foi escolhida em virtude de ter sido reescrita a API de eventos, denominada *Event 2*, mais robusta para o recebimento de todos os eventos em um único monitoramento (MOODLE, 2017f). Das 53 categorias de *plugins* disponíveis para desenvolvimento no Moodle, foi escolhido o tipo local, que

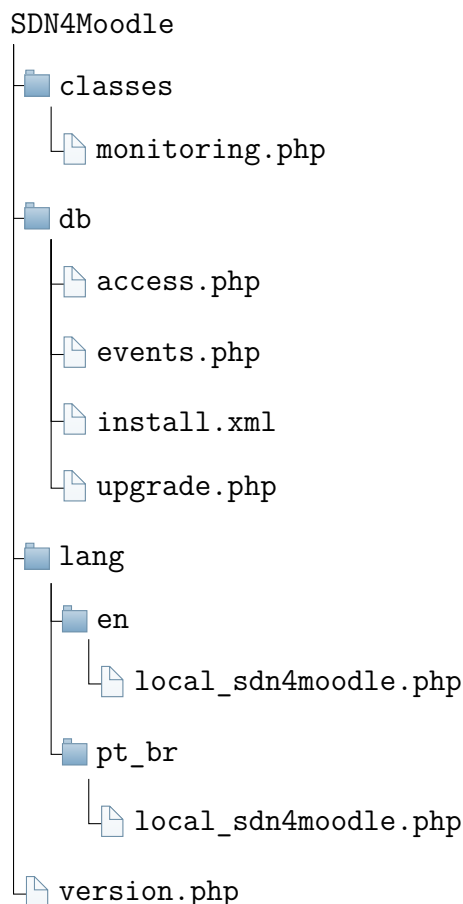


Figura 14 – Estrutura de arquivos do Módulo 1, *plugin* do Moodle.

são *plugins* genéricos para customizações locais específicas (MOODLE, 2017e). Para a instalação do Módulo 1 é necessário apenas a cópia da pasta do módulo para o diretório `moodle/local/` e posterior aceitação do *plugin* no painel administrativo do servidor do Moodle. Na Figura 3, o Módulo 1 corresponde a um *plugin*.

A Figura 14 apresenta a sua estrutura de arquivos, contendo tanto arquivo e diretórios obrigatórios quanto opcionais. O arquivo `classes/monitoring.php` é a classe principal do *plugin*, a qual contém a função que recebe os eventos do Moodle e os armazena no banco de dados. São conservadas informações como o nome de usuário e o seu endereço IP, o horário do evento, o curso e o tipo de evento. Esses dados são utilizados posteriormente pelo Módulo 2 para enviar ações para o controlador SDN.

Uma vez que há diversos tipos de *plugins*, cada um com as suas particularidades, o arquivo `db/access.php` define quais são as capacidades do *plugin* desenvolvido. As capacidades definidas para o Módulo 1 foram as mínimas possíveis, dado que não há a interação com os usuários e a única tarefa é a de guardar os eventos ocorridos no Moodle. Ainda, para que o Módulo 1 receba os eventos do núcleo do Moodle, é necessário que sejam cadastrados observadores para cada tipo de evento que se de-

seja monitorar. Esse cadastro é realizado através do arquivo `db/events.php`. Como um dos requisitos funcionais da ferramenta é a de capturar os eventos da navegação dos usuários no Moodle (Requisito Funcional 03), foi definido que o observador do *plugin* seja para monitorar todos os tipos de eventos do Moodle.

O arquivo `db/install.xml` é o responsável por realizar a primeira instalação do *plugin* no banco de dados do Moodle. Como o Moodle foi projetado de tal forma que cada *plugin* tem o seu próprio conjunto de tabelas no banco de dados, o módulo de instalação de *plugins* faz a leitura deste arquivo e cria as tabelas, campos e chaves de acordo com o que foi especificado. Já o arquivo `db/upgrade.php` trata as modificações necessárias na estrutura do banco de dados do *plugin* quando uma nova versão é instalada e já há uma anteriormente instalada.

As traduções dos textos do *plugin* são armazenadas no diretório `lang`. Para cada idioma que se deseja traduzir, são criados um diretório no padrão ISO 639-1 representando o idioma e um arquivo no formato `tipoPlugin_nomePlugin.php` com as traduções desejadas. Neste trabalho foram realizadas as traduções para os idiomas inglês e para o português brasileiro. Por fim, o arquivo `version.php` engloba as informações da versão do *plugin*. O arquivo contém a versão a ser instalada, a versão mínima do Moodle compatível com o *plugin*, a sua maturidade, a sua versão de liberação, nome e tipo.

Tabela 15 – Atributos dos eventos do Moodle capturados pela ferramenta SDN4Moodle. Adaptado de Moodle (2017f).

Atributo	Descrição
<i>EventName</i>	Classe do evento
<i>Component</i>	Componente gerador do evento
<i>Action</i>	Tipo de ação envolvida
<i>Target</i>	Alvo no qual a ação é tomada
<i>ObjectTable</i>	Tabela no banco de dados que representa o objeto do evento
<i>ObjectID</i>	ID do registro do objeto na <i>ObjectTable</i>
<i>Crud</i>	Operação de criação, leitura, atualização ou exclusão
<i>EduLevel</i>	Nível educacional no evento
<i>ContextID</i>	ID do contexto
<i>ContextLevel</i>	Nível do contexto (se é um curso, atividade, etc.)
<i>ContextInstanceID</i>	Baseado no nível do contexto (ID ou categoria de curso, etc.)
<i>UserID</i>	ID do usuário
<i>CourseID</i>	ID do curso
<i>RelatedUserID</i>	Usuário ao qual a ação está relacionada
<i>Anonymous</i>	Verifica se a ação foi de usuário não autenticado
<i>TimeCreated</i>	Horário de ocorrência do evento
<i>IPAddress</i>	Endereço IP do usuário

A Tabela 15 apresenta os atributos que são capturados de cada evento que ocorre no Moodle e armazenados no banco de dados da ferramenta SDN4Moodle.

Apesar da maioria dos atributos serem autoexplicativos, alguns são evidenciados: *EventName* define o nome da classe a qual ocorreu o evento, como por exemplo `\core\event\user_loggedin` que sinaliza a autenticação de um usuário e também `\core\event\course_viewed` que representa a visualização de um curso pelo usuário.

O atributo *Component* armazena o componente gerador do evento (e.g., *core*). *Crud* identifica se a operação foi de criação, leitura, atualização ou exclusão (do inglês *Create*, *Read*, *Update* e *Delete*). *EduLevel* representa o nível educacional no qual ocorreu o evento. Estes níveis podem ser o de ensinando, participando ou outro. *ContextLevel* é o nível do contexto, mostrando se o evento é de um curso, atividade, categoria de curso, etc. Baseado no nível do contexto, *ContextInstanceID* informa, por exemplo, o ID do curso, do módulo do curso ou da sua categoria.

Com a possibilidade da captura dos eventos do Moodle, passa a ser possível a criação de regras de encaminhamento nos dispositivos da rede no nível da camada de aplicação, podendo ser levado em consideração qual o nível de importância e prioridade de cada atividade que os usuários estejam fazendo. Sendo assim, a aplicação passa a informar a rede das suas demandas em tempo de execução, dispensando o administrador da rede da configuração de regras rígidas de controle de congestionamento.

A Seção 4.6 apresenta os detalhes da implementação do Módulo 2, que em conjunto com o *plugin* apresentado, processa os eventos recebidos do Moodle e cria regras de QoS de acordo com a navegação dos usuários.

4.6 IMPLEMENTAÇÃO DO MÓDULO 2

O Módulo 2 é o módulo da ferramenta SDN4Moodle responsável pelo gerenciamento e priorização do tráfego dos usuários da aplicação Moodle em uma SDN. Esta seção apresenta os detalhes da sua implementação, de acordo com o levantamento de requisitos realizado na Seção 3.1. A documentação atualizada da ferramenta é mantida em Marcondes (2018).

Inicialmente, a Tabela 16 resume os eventos monitorados pela SDN4Moodle. Estes eventos são armazenados em banco de dados assim que ocorrem, para posteriormente serem enfileirados e processados de forma cronológica. Assim, permite-se que eventos ocorridos antes do início da execução da ferramenta ou durante uma eventual interrupção, possam também ser processados para todos os usuários do Moodle quando a ferramenta for iniciada. Estes eventos são monitorados através de uma *thread* de *software* que faz uma varredura no banco de dados a cada 1 segundo para verificar a existência de novos eventos cadastrados.

Para que seja possível cadastrar e remover os fluxos dos usuários nos *swit-*

Tabela 16 – Descrição dos eventos monitorados pela ferramenta SDN4Moodle.

Evento	Descrição
<code>\core\event\user_loggedin</code>	<i>Login</i> de usuário
<code>\core\event\user_loggedout</code>	<i>Logoff</i> de usuário
<code>\core\event\course_viewed</code>	Visualização da <i>homepage</i> do curso
<code>\core\event\dashboard_viewed</code>	Visualização da <i>dashboard</i>
<code>\mod_page\event\course_module_viewed</code>	Visualização de página de curso
<code>\mod_quiz\event\course_module_viewed</code>	Realização de prova
<code>\mod_resource\event\course_module_viewed</code>	Visualização de arquivo multimídia

ches, é necessário o monitoramento dos momentos em que ocorrem o *login* e o *logoff* das sessões destes usuários. A API do Moodle fornece essas informações, respectivamente, através do monitoramento dos eventos `\core\event\user_loggedin` e `\core\event\user_loggedout`. Quando um usuário efetua o *login* no Moodle, o *plugin* SDN4Moodle armazena as informações deste evento, que contêm as informações essenciais para a criação do fluxo deste usuário nos *switches* (endereço IP origem, porta de origem, endereço IP destino, porta TCP destino, horário do *login*, etc.).

Cada fluxo que deve ser cadastro é formatado de acordo com as especificações da API do ODL. Os dados enviados para os *switches* representam a combinação das informações cadastradas no banco de dados. Por exemplo, para exibição de um vídeo, quando o evento é detectado pelo módulo, uma pesquisa é realizada na base de dados para verificar a configuração necessária (largura de banda) de acordo com a qualidade configurada pelo administrador. As informações combinadas são submetidas aos *switches* configurando as tabelas de encaminhamento de fluxos e de métricas. É evidente que existe um atraso de configuração, caracterizado pela detecção do evento, propagação e cadastro das regras. Essa informação é evidenciada e quantificada no Capítulo 5.

Devido ao protocolo OpenFlow instalado no *switch* não ter implementado as filas para alocação de largura mínima para os fluxos, esta limitação foi contornada utilizando métricas, que por sua vez permitem somente a determinação de largura de banda máxima para os fluxos. Sendo assim, optou-se pelas seguintes estratégias, atualizadas a cada novo evento:

- Usuários do Moodle: assim que realizam a autenticação no Moodle, é alocada toda a largura de banda disponível na rede (25 Mbps), variando a prioridade dos fluxos de acordo com os eventos.
- Tráfego secundário: a largura de banda alocada para os fluxos do *iperf* é equivalente ao total disponível na rede (25 Mbps), descontado o que foi alocado para os eventos do Moodle.

Tabela 17 – Classes do Módulo 2 da ferramenta SDN4Moodle

Classe	Descrição
<i>Configuration</i>	Configurações gerais da ferramenta
<i>ConfigurationFlow</i>	Atributos dos fluxos
<i>Controller</i>	Abstração do controlador SDN
<i>Database</i>	Comunicação com o banco de dados
<i>Event</i>	Abstração dos eventos do Moodle
<i>Events</i>	Processamento da fila de eventos do Moodle
<i>Flow</i>	Abstração de um fluxo OpenFlow
<i>Link</i>	Abstração de um link entre dois dispositivos OpenFlow
<i>Main</i>	Interface gráfica da ferramenta
<i>Meter</i>	Abstração de uma métrica OpenFlow
<i>Meters</i>	Controle das métricas cadastradas nos dispositivos OpenFlow
<i>Moodle</i>	Comunicação com o servidor Moodle
<i>Node</i>	Abstração de um dispositivo OpenFlow
<i>OpenFlow</i>	Comunicação com o controlador SDN
<i>Page</i>	Abstração de uma página no Moodle
<i>PanelConfiguration</i>	Interface gráfica das configurações gerais da ferramenta
<i>Resource</i>	Abstração de um recurso no Moodle
<i>SDN4Moodle</i>	Classe principal da ferramenta, com o método main
<i>TerminationPoint</i>	Abstração de um dispositivo final na rede
<i>Topology</i>	Abstração da topologia de rede
<i>User</i>	Abstração de um usuário do Moodle
<i>Users</i>	Controle de usuários do Moodle
<i>XML</i>	Conversão de dados XML

O Módulo 2 da ferramenta SDN4Moodle foi desenvolvido na linguagem de programação Java SE Development Kit 8, atualização 92 (JDK 8u92), com a utilização do Netbeans IDE versão 8.1. Para a instalação e execução deste módulo é necessário apenas que a Java Virtual Machine (JVM) versão 8 esteja instalada no dispositivo. O código-fonte do Módulo 2 tem um total de 23 classes, sendo que Tabela 17 apresenta a descrição resumida de cada uma. O Diagrama de Classes da ferramenta está disponível no Anexo B.

4.7 PLANO DE TESTES

O plano de testes deste trabalho, baseado na norma IEEE 829 (IEEE, 2008), foi especificado para analisar e apresentar a melhora no desempenho de aplicações que utilizam a rede de computadores através do paradigma das Redes Definidas por *Software*. Para os experimentos do escopo deste trabalho, o desempenho da aplicação Moodle em uma topologia de rede tradicional (grupo de controle) é comparado com os resultados da sua execução em uma topologia SDN. Foram definidos dois critérios de avaliação no plano de testes, cada qual com um objetivo distinto:

- Primeiro critério: avaliar o comportamento do tráfego secundário (outras aplicações, representado pela ferramenta *iperf*) durante a navegação do usuário do Moodle.
- Segundo critério: analisar o tempo de *download* dos dados das atividades da navegação do Moodle para o dispositivo cliente.

O primeiro critério irá comparar o consumo de largura de banda na rede entre o servidor e o cliente *iperf*. Como o meio de transmissão do *iperf* e do Moodle é compartilhado, à medida que o usuário executar as atividades no Moodle, a vazão de dados entre o servidor e cliente *iperf* é afetada. Já o segundo critério irá capturar a cada segundo o tráfego de *download* diretamente da placa de rede do cliente Moodle. Dessa forma é possível avaliar o tempo em que cada atividade é concluída em ambos os tipos de topologias. As execuções dos ensaios são divididos em 4 cenários, com topologia e quantidade de usuários distintas.

4.7.1 Cenário 1

O primeiro cenário é composto de 3 computadores interconectados através de 3 *switches* em uma topologia árvore, sem a execução da solução proposta. Conforme representado pela Figura 15), na folha do lado esquerdo são conectados os servidores Moodle e *iperf*, os quais compartilham o meio de transmissão até o *switch* raiz. Já no lado direito, o meio de transmissão também é compartilhado, dessa vez pelos clientes *iperf* e Moodle.

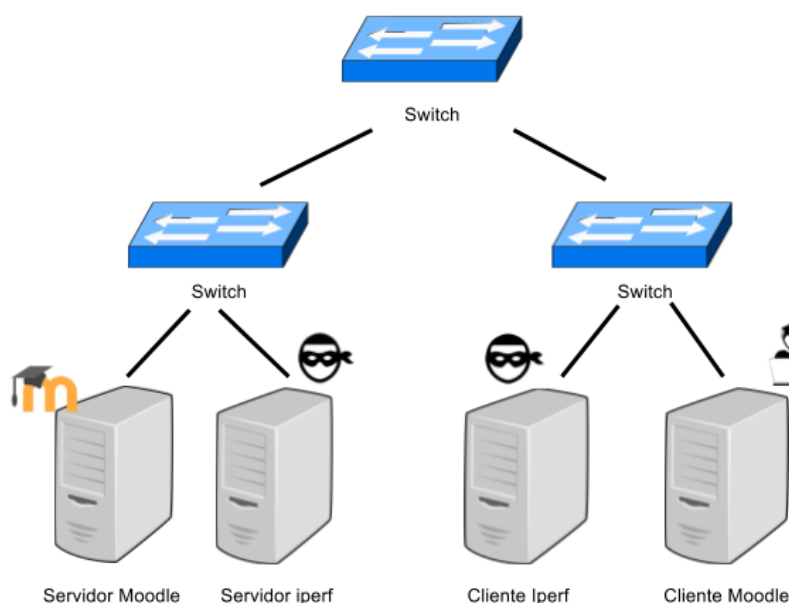


Figura 15 – Primeiro cenário do plano de testes.

Para este cenário, um usuário no dispositivo do cliente Moodle irá realizar atividades na aplicação enquanto todo o canal de comunicação é saturado pelo tráfego secundário do *iperf*. Os resultados da coleta servem como linha de base para comparação com o Cenário 3, que compreende a mesma quantidade de dispositivos, mas com a execução solução proposta em uma topologia SDN.

4.7.2 Cenário 2

O segundo cenário compreende uma topologia de rede tradicional com 3 *switches* e 5 dispositivos, sendo 2 servidores e 3 clientes (Figura 16). Na folha do lado esquerdo estão os servidores Moodle e *iperf*. Diferentemente do Cenário 1, neste cenário, 2 clientes Moodle compartilham o canal de comunicação até o *switch* raiz.

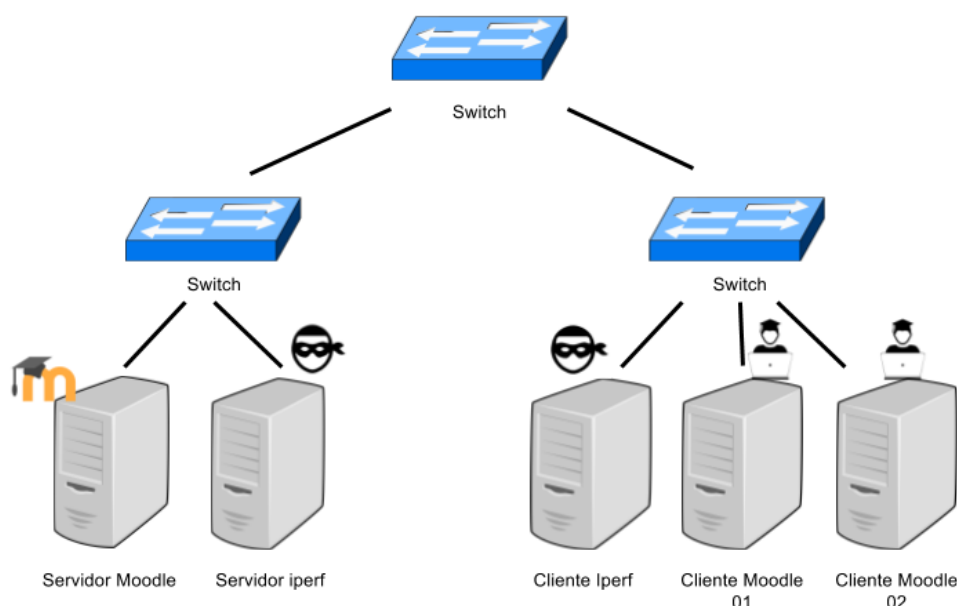


Figura 16 – Segundo cenário do plano de testes.

4.7.3 Cenário 3

O terceiro cenário compreende uma topologia com SDN. Como ilustrado pela Figura 17, o *switch* raiz possui suporte ao protocolo OpenFlow, reconhecendo as configurações enviados pelo SDN4Moodle. Os *switches* 1 e 2 não possuem suporte, representando a configuração de uma rede híbrida. Os servidores e clientes Moodle e *iperf* estão conectados nos *switches* 1 e 2.

Ainda, é importante ressaltar a existência de uma rede de gerenciamento, externa à rede que é analisada. No *switch* de gerenciamento, estão conectados o controlador OpenFlow, o SDN4Moodle e o servidor Moodle. O tráfego OpenFlow é representado por linhas tracejadas enquanto o tráfego de gerenciamento é denotado por linhas contínuas.

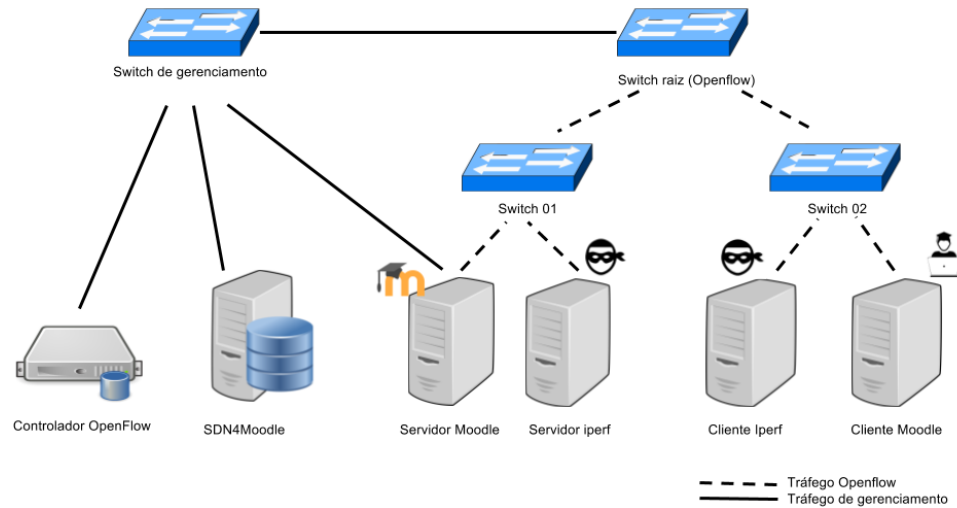


Figura 17 – Terceiro cenário do plano de testes.

4.7.4 Cenário 4

Por fim, o quarto cenário de testes possui uma topologia semelhante ao Cenário 3, aumentando apenas o número de clientes do Moodle, com um total de 2 usuários do Moodle. A Figura 18 resume os componentes do cenário. Novamente, o tráfego OpenFlow ocorre pelas arestas tracejadas enquanto o tráfego de gerenciamento é representado por arestas contínuas.

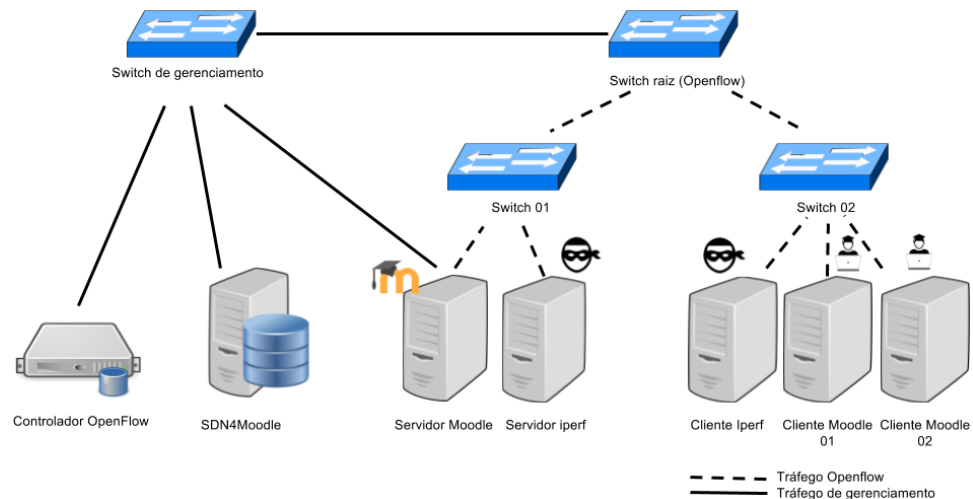


Figura 18 – Quarto cenário do plano de testes.

4.8 CONSIDERAÇÕES PARCIAIS

Este capítulo apresentou os detalhes da implementação da proposta de controle da qualidade de serviço para a aplicação Moodle em SDNs através da ferramenta SDN4Moodle, permitindo que o tráfego específico dos seus usuários seja priorizado

em tempo de execução de forma dinâmica e sem a necessidade da intervenção do administrador da rede. Este controle de qualidade é realizado através da capacidade da ferramenta proposta de identificar os pacotes da navegação dos usuários do Moodle na rede. Esta identificação ocorre através da definição de tuplas próprias disponíveis no protocolo OpenFlow versão 1.3.

O projeto do banco de dados da ferramenta foi elaborado para atender os requisitos funcionais e não funcionais previamente definidos. Além disso, foi levado em consideração a possibilidade deste projeto ser escalável e também permitir a sua expansão em eventuais trabalhos futuros. Como não há relação entre as tabelas do banco de dados, sem a ocorrência de chave estrangeira nas tabelas, estas foram apresentadas de forma isolada no texto do capítulo e não em formato de diagrama entidade relacionamento.

Para que fosse possível a coleta dos eventos de navegação dos usuários do Moodle e a posterior priorização deste tráfego, foi necessário a divisão da ferramenta em dois módulos, os Módulos 1 e 2. O primeiro é um *plugin* do Moodle específico para a captura e armazenamento em banco de dados dos eventos. Já o segundo é a própria SDN4Moodle, responsável por processar os eventos recebidos pelo Moodle e realizar o controle da qualidade de serviço dos fluxos dos seus usuários, priorizando o tráfego de acordo com a importância da atividade e a necessidade de recursos da rede. Por fim, foi apresentado o plano de testes deste trabalho.

5 ANÁLISE EXPERIMENTAL

Este capítulo apresenta a análise experimental da ferramenta SDN4Moodle em uma topologia de rede com o paradigma das SDNs. Inicialmente, na Seção 5.1 são definidas as métricas para análise, conforme proposto pelo plano de testes. Na sequência, a Seção 5.2 define o ambiente de testes com a descrição da topologia e dos dispositivos utilizados na coleta dos dados. Posteriormente, o plano de testes é executado, resumindo os resultados na Seção 5.3. As limitações técnicas encontradas são discutidas na Seção 5.4. Por fim, a Seção 5.5 apresenta as considerações parciais do capítulo.

5.1 MÉTRICAS PARA ANÁLISE

Duas métricas foram selecionadas para quantificar a aplicabilidade da ferramenta em um cenário experimental:

- Eficácia da reserva de largura de banda: É apresentada a diminuição do tráfego secundário em benefício da aplicação, quantificado em Mbit/s. O objetivo é evidenciar que a reserva realizada pela ferramenta diminui a vazão obtida pelo tráfego secundário, em benefício do utilizador do Moodle. Essa informação é coletada diretamente na aplicação *iperf*.
- Ocorrência de perda de pacotes na aplicação de regras de fluxos, com o objetivo de identificar o número de intervalos e pacotes perdidos durante a reconfiguração dos *switches* para aplicação das regras de encaminhamento. Ao analisar os dados referentes à métrica, espera-se quantificar a depreciação do serviço durante a reconfiguração. A coleta de dados sobre perda de pacotes ocorreu diretamente na interface de conexão do cliente Moodle.

5.2 AMBIENTE DE TESTES

O servidor da ferramenta SDN4Moodle comunica-se tanto com o controlador OpenFlow (enviando ações para criação e remoção de fluxos dos usuários) quanto com o servidor do Moodle (recebendo as informações dos eventos que ocorrem no LMS). Para aplicação do tráfego secundário, foi utilizado a ferramenta *iperf* com sessões TCP. Devido à limitação da capacidade de processamento do *switch* raiz virtualizado (PFAFF et al., 2009), a largura de banda máxima na rede foi definida em 25 Mbps. Valores acima deste ocasionam o travamento do dispositivo, acarretando no descarte de todos os pacotes recebidos.

Tabela 18 – Arquitetura dos computadores utilizados para compor o cenário experimental

Dispositivo	Sist. Operacional	Processador	Memória
Servidor Moodle	Ubuntu 16.04 64 bits	AMD Athlon TF-20 1,6 GHz	3,6 GB
Controlador SDN	Ubuntu 16.04 64 bits	AMD Athlon TF-20 1,6 GHz	3,6 GB
SDN4Moodle	Windows 10 64 bits	Intel Core I5 3340m 2,7 GHz	8,0 GB
Servidor Iperf	Ubuntu 14.04 32 bits	Intel Atom N270 1,6 GHz	1,5 GB
Cliente Iperf	Ubuntu 14.04 32 bits	Intel Atom N270 1,6 GHz	1,5 GB
Cliente Moodle 1	Ubuntu 16.04 64 bits	Intel Pentium T4300 2,1 GHz	3,0 GB
Cliente Moodle 2	Ubuntu 16.04 64 bits	Intel Pentium T4300 2,1 GHz	3,0 GB

Tabela 19 – Dispositivos de encaminhamento na topologia.

Switch	Equipamento	Interfaces	OpenFlow
<i>Switch</i> raiz	TP-Link TL-WR1043ND	1 Gbit/s	OpenFlow 1.3
<i>Switch</i> borda 1	Genérico 8 interfaces	100 Mbit/s	Não
<i>Switch</i> borda 2	Genérico 8 interfaces	100 Mbit/s	Não
<i>Switch</i> administrativo	Linksys SE 1500	1 Gbit/s	Não

A Tabela 18 resume a configuração dos servidores enquanto a Tabela 19 apresenta a configuração dos *switches*, sendo que a justificativa para composição do ambiente com os equipamentos listados (diante das dificuldades técnicas encontradas) é discutida na Seção 5.4.

5.3 CENÁRIOS DE TESTE

De acordo com o plano de teste apresentado na Seção 4.7, os valores para as métricas previamente elencadas foram coletados para cada cenário e são individualmente analisados na Seção 5.3. Em resumo, os 4 cenários objetivam:

- Cenário 1: Execução e coleta dos resultados sem a execução da SDN4Moodle, com os recursos de SDN e OpenFlow desabilitados. Utilizado como linha de base para comparações quando apenas um usuário utiliza o ambiente Moodle.
- Cenário 2: Execução e coleta dos resultados sem a execução da SDN4Moodle. Os recursos de SDN e OpenFlow são desabilitados. Os valores obtidos compõem linha de base para comparações com 2 clientes do Moodle em execução.
- Cenário 3: Execução de diversas atividades no Moodle, realizadas por um usuário. O gerenciamento é realizado por SDN4Moodle.
- Cenário 4: Execução de diversas atividades no Moodle com 2 usuários ativos, guiados por SDN4Moodle.

A Tabela 20 descreve as atividades executadas na coleta dos dados. Os testes foram realizados com duas pessoas no computador realizando as atividades de forma

Tabela 20 – Atividades executadas pelos usuários do Moodle durante a análise experimental.

Atividade	Descrição
Atividade de texto	Responder questionário com 10 questões
<i>Streaming</i> de áudio	Executar atividade com arquivo de áudio no formato mp3
Vídeo de baixa qualidade	Executar atividade com arquivo de vídeo de baixa qualidade (240p)
Vídeo de média qualidade	Executar atividade com arquivo de vídeo de média qualidade (480p)
Vídeo de alta qualidade	Executar atividade com arquivo de vídeo de alta qualidade (720p)

sequencial, sem a utilização de ferramentas automatizadas de testes. A atividade de texto elaborada para os testes é uma prova do tipo múltipla escolha, com um total de 10 questões com 4 alternativas cada, sendo uma pergunta apresentada por vez ao usuário. O papel da ferramenta SDN4Moodle é de identificar o início desta atividade e alocar largura de banda suficiente para que o tráfego secundário não inviabilize a sua execução, uma vez que a sessão do TCP do usuário pode ser prejudicada com as rajadas do tráfego do *iperf* (ALIZADEH et al., 2010b).

Tanto o *streaming* de áudio quanto os vídeos são exibidos diretamente no *Media Player* do Moodle, não ocorrendo o *download* prévio dos arquivos para o dispositivo do usuário. O arquivo de vídeo escolhido para a análise foi o "Prêmio Professores do Brasil é tema do Educação no Ar", do Ministério da Educação, disponível na plataforma Youtube (EDUCAÇÃO, 2017). O vídeo foi selecionado por ter dez minutos e estar acessível em uma plataforma pública com múltiplas taxas de resolução. Foram atribuídos três tipos de qualidade para o vídeo de acordo com a sua resolução: alta qualidade para 720p, média qualidade para 480p e baixa qualidade para a resolução 240p, todos no formato MP4.

Já o *streaming* de áudio selecionado para os testes foi a extração do som do próprio vídeo, em formato MP3 com qualidade de 192 kbps e tempo total de dez minutos. Tanto o *download* quando a conversão para os formatos especificados dos vídeos e do áudio foram efetuados através da plataforma *OnlineVideoConverter*¹. Definidos os arquivos, estes foram armazenados no servidor do Moodle para execução pelos usuários.

A Tabela 21 apresenta a sequência dos passos da navegação do usuário no Moodle e da coleta dos dados do tráfego da rede. Antes do começo da navegação do usuário é iniciada a coleta dos bytes recebidos pela placa de rede do dispositivo do usuário (A). A partir deste momento somente tráfego do Moodle é recebido e enviado pela placa de rede. Também são inicializados os processos do servidor e do cliente

¹ <https://www.onlinevideoconverter.com>

Tabela 21 – Sequência dos passos para coleta dos dados.

Sequência	Atividade
A	Início da coleta na placa de rede do usuário
B	Início do servidor <i>iperf</i>
C	<i>Login</i> no Moodle
D	Início da atividade de texto
E	Fim da atividade de texto
F	Início da atividade de áudio
G	Fim da atividade de áudio
H	Início do vídeo de baixa resolução
I	Fim do vídeo de baixa resolução
J	Início do vídeo de média resolução
K	Fim do vídeo de média resolução
L	Início do vídeo de alta resolução
M	Fim do vídeo de alta resolução
N	<i>Logoff</i> do usuário
O	Fim da coleta na placa de rede do usuário
P	Fim do servidor <i>iperf</i>

iperf nos equipamentos da topologia (B).

Realizada a atividade de texto, a próxima etapa é a execução do *streaming* de áudio (F). Como é realizado todo o *buffer* do arquivo já nos segundos iniciais da sua execução, o áudio é interrompido depois de trinta segundos do seu andamento. O retorno para a lista de tópicos do curso representa o fim da atividade de áudio (G). O mesmo roteiro é seguido para a execução dos vídeos (sequências H, J e L para o início e I, K e M para o fim), com a diferença que os vídeos são executados até o final.

Com o término das atividades, o usuário realiza o *logoff* (N), encerrando sua interação com o Moodle. Por fim, é encerrada a coleta dos bytes recebidos na placa de rede do usuário (O) e também finaliza-se o processo do servidor *iperf* (P). Para cada uma destas ações (coleta dos bytes e servidor) são gerados arquivos de *log* dos dados, que são utilizados na análise experimental.

As reservas de largura de banda utilizadas para o escopo deste trabalho foram as seguintes: 2 Mbps a partir do *login*, 4 Mbps para atividades de texto, *streaming* de áudio e vídeo de baixa resolução, 8 Mbps para vídeo de média resolução e 16 Mbps para vídeo de alta resolução. Os valores foram escolhidos em virtude da limitação imposta pelo *hardware* do *switch* OpenFlow, o qual descarta pacotes quando alcança vazão a partir dos 30 Mbps (limitações técnicas são discutidas na Seção 5.4). As próximas seções apresentam os resultados para 1 e 2 usuários utilizando o Moodle, com e sem o controle efetuado por SDN4Moodle.

Em todos os gráficos o eixo x representa o tempo na navegação dos usuários. Já o eixo y é a largura de banda alcançada pelo *iperf* ou a quantidade de bytes do

tráfego Moodle na placa de rede do usuário (informação obtida diretamente na placa de rede). Para a apresentação dos números nos gráficos foi realizada uma suavização, no qual cada valor no eixo y corresponde ao valor médio entre o valor atual e o do segundo anterior à coleta. Isto foi necessário em virtude do tempo mínimo de coleta dos *logs* do *software Iperf* ser de 1 segundo.

5.3.1 Cenário 1

O primeiro cenário da análise resume a execução sem o controle efetuado pela ferramenta SDN4Moodle, seguindo a topologia da Figura 15, sendo composta por três *switches* e quatro computadores interconectados em uma topologia em árvore. Tanto no Cenário 1, quanto nos demais cenários, é executada a mesma navegação do usuário, sendo composta pela sequência descrita na Tabela 21. A ocorrência dos eventos dos usuários foi marcada com as linhas verticais na parte superior do gráfico. Os eventos que ocorrem antes e depois da navegação do usuário foram desconsiderados nos gráficos (A, B, O e P), por não interferirem na análise dos resultados.

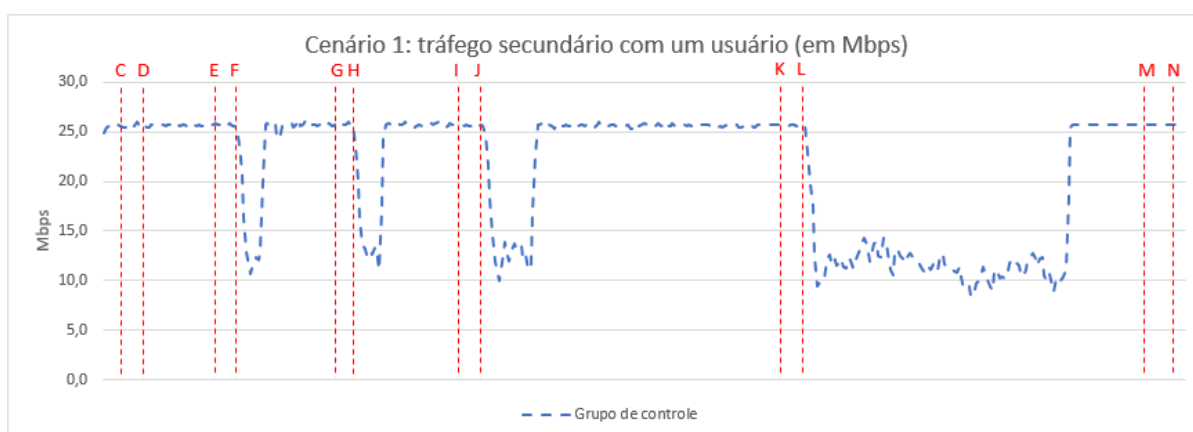


Figura 19 – Cenário 1: gráfico da largura de banda do tráfego secundário do *iperf* com um usuário, sem o controle da ferramenta SDN4Moodle (vazão em Mbps).

A Figura 19 apresenta o gráfico da largura de banda gerada pelo *iperf* durante a navegação do usuário do Moodle, sendo o eixo x o tempo e o eixo y a largura de banda. Como pode ser observado, o *iperf* ocupa todo o canal de comunicação nos eventos que não há transferência de dados considerável entre o servidor do Moodle e o dispositivo do usuário. Este comportamento é observado nos eventos de *login* (C) e atividade de texto (D), em virtude do protocolo TCP não ter tempo suficiente para detectar o congestionamento do meio de transmissão e aplicar de forma justa a divisão da largura de banda. Como consequência, o usuário tem a percepção de demora no início das suas ações no Moodle.

A partir dos eventos de execução do *streaming* de áudio (F) e vídeos (H, J e L), ocorre a diminuição da largura de banda que o servidor *iperf* consegue alocar com o seu cliente. Isto ocorre porque o protocolo escolhido para o tráfego secundário foi o TCP. Caso tivesse sido o protocolo UDP, o ponto mínimo da largura de banda durante estes eventos seria um valor maior. Testes realizados com dois usuários do Moodle e tráfego secundário UDP mostraram não ser possível a execução do vídeo de alta resolução sem o controle realizado pela SDN4Moodle no presente cenário experimental.

Como a largura de banda máxima considerada para a rede foi de 25 Mbps durante a execução dos testes (com o eixo x sendo o tempo), foi este o valor máximo que o *iperf* trafegou na rede (eixo y do gráfico). Já para os valores mínimos foram alcançados valores na faixa de 10 Mbps. Para a execução do vídeo de alta resolução, o valor mínimo do *iperf* foi de 8,4 Mbps.



Figura 20 – Cenário 1: dados do Moodle trafegados na placa de rede do dispositivo do usuário (em bytes).

Além da captura do tráfego secundário, também foi realizada a leitura, a cada segundo, da quantidade de bytes de *download* trafegados na placa de rede do dispositivo cliente durante a navegação no Moodle. A Figura 20 apresenta o gráfico destes valores. Como pode ser observado, foram realizados quatro blocos de dados transferidos do servidor do Moodle para o dispositivo cliente, sendo eles a execução do *streaming* de áudio (F, G), do vídeo de baixa resolução (H, I), de média resolução (J, K) e o de alta resolução (L, M). Os eventos de *login* (C), atividade de texto (D, E) e *logout* (N) não geraram tráfego considerável. Esses dados serão utilizados para comparação com os resultados do Cenário 3.

5.3.2 Cenário 2

O segundo cenário possui um usuário do Moodle a mais que no primeiro cenário, sem a execução da ferramenta SDN4Moodle. A topologia de rede é composta por três *switches* interconectados em árvore e cinco dispositivos (Figura 16). Os dois usuários do Moodle realizam a navegação simultânea em dispositivos distintos, cujos eventos correspondentes às suas ações (Tabela 21) estão sinalizados com as linhas tracejadas verticais vermelhas na parte superior dos gráficos. Os eventos que não fazem parte da navegação dos usuários foram desconsiderados nos gráficos (eventos A, B, O e P).

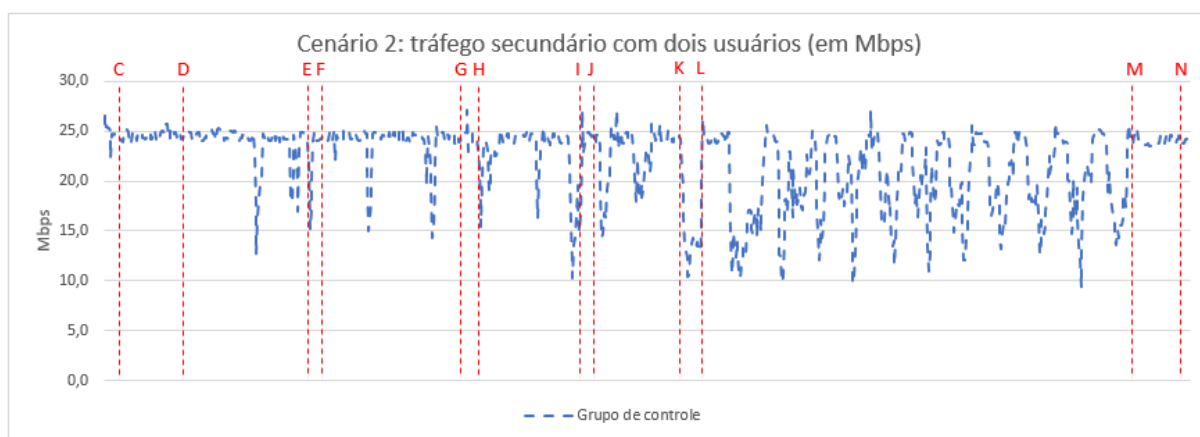


Figura 21 – Cenário 2: gráfico da largura de banda do tráfego secundário do *iperf* com dois usuários, sem o controle da ferramenta SDN4Moodle (vazão em Mbps).

A Figura 21 apresenta os resultados da largura de banda alcançada pelo *iperf* para dois usuários do Moodle, sendo o eixo x o tempo e o eixo y a largura de banda. Assim como no primeiro cenário, foi alcançada a largura de banda máxima da rede (25 Mbps) nos eventos em que há pouco tráfego entre o servidor do Moodle e os dispositivos dos usuários. Já a vazão mínima foi de 10 Mbps em diversos períodos durante a navegação dos usuários. Eventuais picos de velocidade acima dos 25 Mbps foram alcançado em virtude do tempo de leitura do *log* do *iperf*, que é de 1 segundo.

Para alguns eventos, tais como *login* (C), atividade de texto (D) e *logout* dos usuários, a vazão do tráfego secundário do *iperf* não permitiu uma divisão igualitária do canal de comunicação entre o dispositivos da aplicação Moodle e do *iperf*. Isto acontece, assim como no cenário 1, devido ao protocolo TCP não ter tempo suficiente para detectar o congestionamento e aplicar de forma justa a divisão da largura de banda. Como consequência, ambos os usuários do Moodle têm a percepção de demora no início das suas ações das atividades no Moodle. Também foi definido o protocolo TCP em detrimento do UDP, em razão da impossibilidade da execução dos vídeos sem o controle efetuado pela ferramenta SDN4Moodle.

Diferentemente do cenário 1, percebe-se a ocorrência de algumas quedas pontuais na vazão do tráfego *iperf*. Três delas podem ser vistas entre o início e fim da atividade de texto (intervalo D-E). Já para a execução do *streaming* de áudio (intervalo F-G), enquanto que no cenário 1, houve apenas no momento inicial da execução a queda da vazão, no cenário 2 isto ocorreu em diversos momentos durante a navegação dos usuários. Este comportamento pode ser constatado também na execução dos vídeos de baixa (intervalo H-I), média (intervalo J-K) e alta resolução (intervalo L-M).

Apesar de em ambos os cenários 1 e 2 a topologia de rede e as ações na navegação dos usuários serem idênticas, variando apenas em um computador cliente do Moodle, a diferença nos gráficos pode ser explicada por causa do dobro de tráfego Moodle transmitido pela rede, em que a taxa da transferência dos dados para os dispositivos dos clientes Moodle pode ter variado para estes usuários durante a navegação, em que o protocolo TCP procura ajustar as demandas de tráfego tanto para o tráfego *iperf* quanto para os 2 usuários do Moodle. Há também o controle do tamanho do *buffer* no navegador Web, em que este pode solicitar, de forma imprevisível, novas transferências de dados ao servidor Moodle.

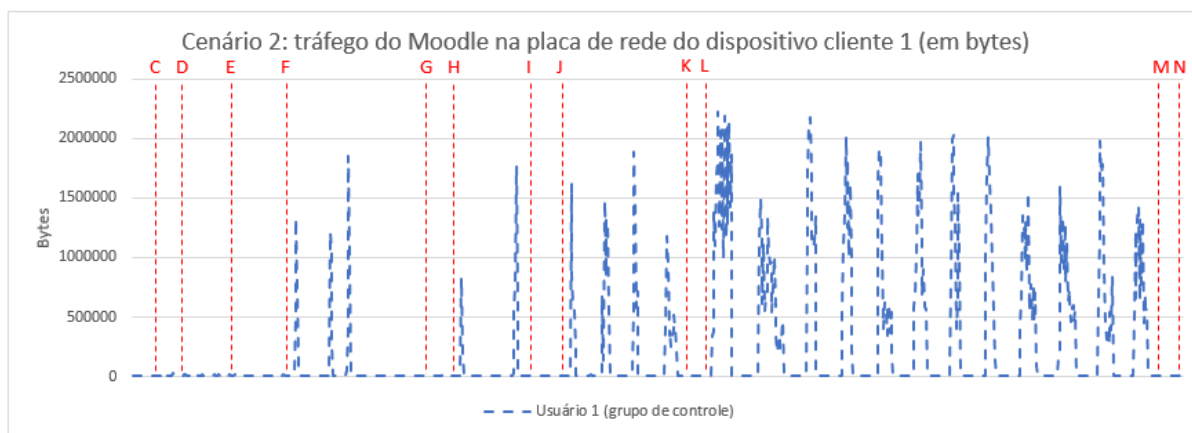


Figura 22 – Cenário 2: dados do Moodle trafegados na placa de rede do dispositivo do primeiro usuário (em bytes).

Além da coleta do tráfego secundário do *iperf*, também foi realizada a leitura, a cada segundo, da quantidade de bytes de *download* trafegados na placa de rede de ambos os dispositivos clientes durante as suas navegações no Moodle. As Figuras 22 e 23 apresentam os gráficos do tráfego do Moodle nas placas de rede dos usuários 1 e 2, respectivamente.

No gráfico da Figura 22, no qual é apresentada a transferência dos dados do Moodle para o dispositivo do cliente 1, o comportamento do tráfego é diferente do observado no cenário 1. Enquanto que no cenário 1 o modo predominante é o de

rajada única na transferência (com exceção do vídeo de alta resolução), no cenário 2 foram observadas mais de uma rajada para todas as execuções de mídia (*streaming* de áudio e vídeo). Este comportamento vem de encontro com o gráfico da Figura 21, em que foram observadas quedas na taxa de transferência do tráfego secundário. Mostra-se que esta queda do tráfego *iperf* foi causada pelas rajadas do tráfego do Moodle.

Além do número de rajadas ser em maior quantidade do que no cenário 1, as transferências do cenário 2 são mais estreitas e em maior quantidade. O mesmo tráfego é enviado de forma parcial, à medida que o *buffer* da execução do áudio ou dos vídeos é preenchido, e também por causa da gestão do congestionamento realizada pelo protocolo TCP. Acrescenta-se que para a análise deste trabalho considera-se tráfego Moodle todos os dados transmitidos pelo servidor Moodle, inclusive tráfego de controle e eventuais retransmissões causadas pelo congestionamento do canal de comunicação.

Como pode ser observado, foram realizados quatro blocos de dados transferidos do servidor do Moodle para o dispositivo cliente, sendo eles a execução do *streaming* de áudio (F, G), do vídeo de baixa resolução (H, I), de média resolução (J, K) e o de alta resolução (L, M). Os eventos de *login* (C), atividade de texto (D, E) e *logout* (N) não geraram tráfego considerável. Esses dados serão utilizados para comparação com os resultados do usuário 1 no Cenário 4.

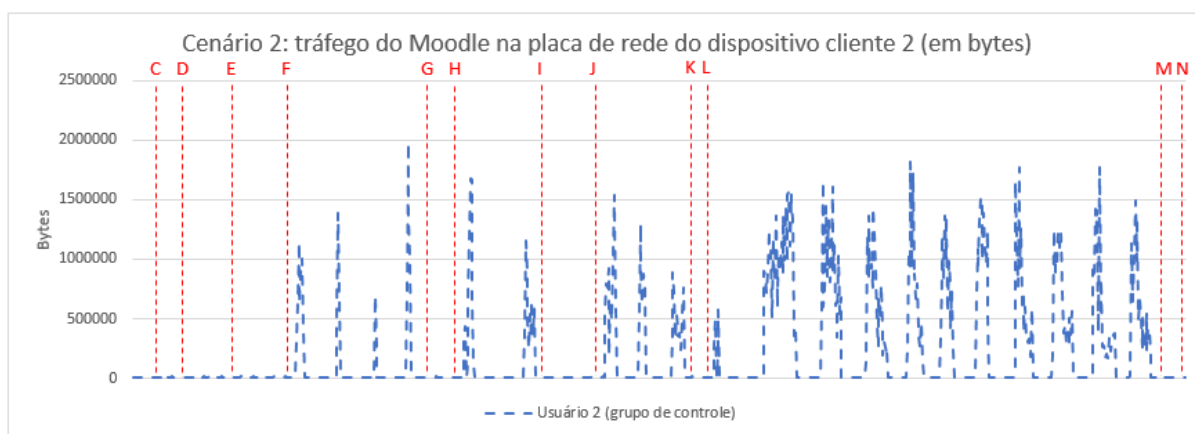


Figura 23 – Cenário 2: dados do Moodle trafegados na placa de rede do dispositivo do segundo usuário (em bytes).

O gráfico da Figura 23 apresenta a captura, no dispositivo cliente, dos dados do Moodle transferidos para a placa de rede do usuário 2. Os resultados apresentam similaridade com o que foi observado no gráfico da Figura 22 do usuário 1. As execuções do *streaming* de áudio (F, G), do vídeo de baixa resolução (H, I), de média resolução (J, K) e o de alta resolução (L, M) apresentaram mais de uma rajada na

transferência dos dados, em virtude do compartilhamento do canal de comunicação entre o tráfego secundário do *iperf* e os usuários 1 e 2. Esses dados serão utilizados para comparação com os resultados do usuário 2 no Cenário 4.

5.3.3 Cenário 3

O terceiro cenário corresponde à execução das atividades no Moodle por um usuário, confiando no gerenciamento efetuado pela ferramenta SDN4Moodle. Os resultados obtidos são comparados com os dados do primeiro cenário. A topologia de rede é apresentada na Figura 17, sendo composta de 4 *switches* e 6 computadores. O *switch* raiz da rede contém o protocolo OpenFlow ativado. O plano de controle é gerenciado pelo controlador OpenFlow, o qual recebe ações da SDN4Moodle.

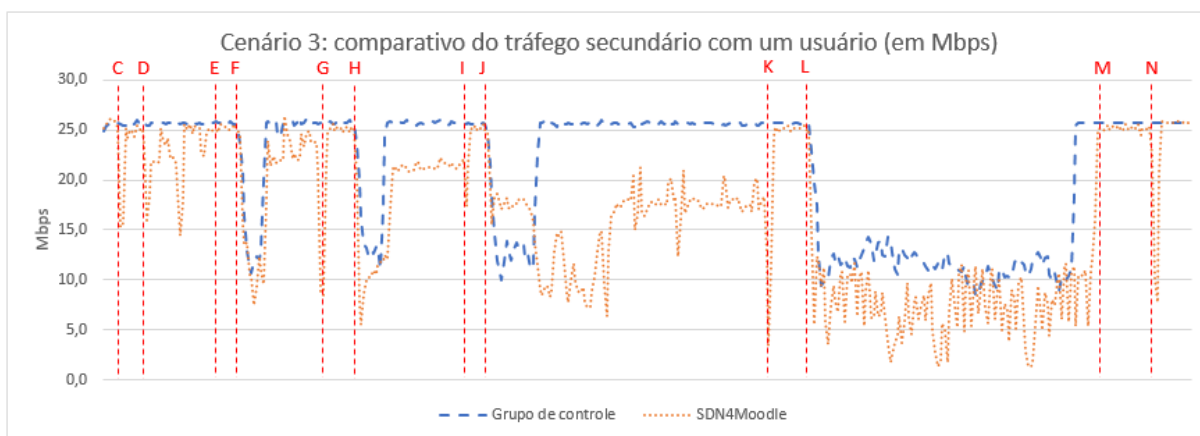


Figura 24 – Cenário 3: gráfico com o comparativo da largura de banda do tráfego secundário do *iperf* com um usuário, com e sem o controle da SDN4Moodle (vazão em Mbps).

A Figura 24 mostra o comparativo da largura de banda alcançada pelo *iperf* durante a navegação de um usuário do Moodle, sendo o eixo x o tempo e o eixo y a largura de banda. As linhas tracejadas (em azul) correspondem à navegação sem a SDN4Moodle e as linhas pontilhadas (em laranja) representam a navegação com o controle efetuado pela SDN4Moodle. Os eventos ocorridos durante a navegação foram sinalizados com as letras e as linhas tracejadas verticais na parte superior do gráfico. É importante ressaltar que a diferença entre as linhas caracteriza a eficácia da largura de banda reservada por SDN4Moodle.

À medida que o usuário realiza as atividades no Moodle, o Módulo 1 (*plugin* do Moodle) armazena estes eventos no banco de dados, para posterior processamento pela SDN4Moodle (Módulo 2). Quando um evento classificado como reserva de largura de banda é detectado, a métrica do fluxo do usuário é atualizada e, conseqüentemente, o tráfego secundário passa a ser menor entre o servidor *iperf* e o seu cliente.

A primeira observação a ser realizada é a queda acentuada na largura de banda durante os eventos em que é necessária a atualização do valor da largura de banda disponível para o usuário do Moodle (principalmente nos eventos C, D, E, F, G, H, I, K, L e N). No gráfico, estes valores chegam próximo dos 5 Mbps. Isto ocorre em virtude da limitação do protocolo OpenFlow instalado *switch* raiz da rede, que exclui a fila de pacotes da tabela de fluxos quando a métrica (largura de banda disponível para cada usuário do Moodle e para o *iperf*) é atualizada.

O comparativo entre os dois cenários, sem o controle da SDN4Moodle (grupo de controle - linhas tracejadas) e a efetiva reserva de largura de banda baseada em eventos da SDN4Moodle (SDN4Moodle - linhas pontilhadas) mostra que a ferramenta realiza de forma precisa a reserva de largura de banda, fazendo com que a taxa do tráfego secundário entre o servidor e o cliente *iperf* varie de acordo com as reservas efetuadas para o usuário do Moodle. A variação (subidas e descidas) nos resultados do gráfico pode ser o protocolo TCP ou também a configuração do *player* do navegador Web na camada de aplicação.

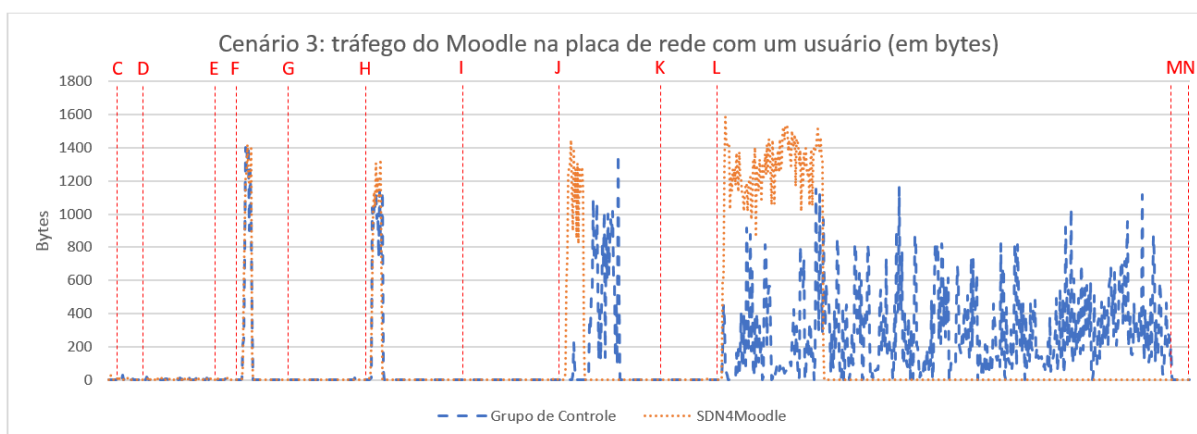


Figura 25 – Cenário 3: comparativo dos dados do Moodle trafegados na placa de rede do dispositivo do usuário (em bytes).

O comparativo do tráfego do Moodle na placa de rede do usuário é apresentado no gráfico da Figura 25. Assim como no primeiro cenário, os eventos de *login* (C) e atividade de texto (intervalo D-E) não geraram tráfego considerável durante a navegação do usuário.

De modo geral, percebe-se que à medida que se aumenta a quantidade de dados trafegados sob o controle da SDN4Moodle, mais rapidamente é realizada a transferência para o dispositivo cliente. Para o *streaming* de áudio, isto é pouco perceptível, pois a curva é a mesma para ambos os cenários. Mas a partir do vídeo de baixa resolução (intervalo H-I) o ganho com a SDN4Moodle já é nítido, pois para este a curva é mais alta e estreita, sinalizando que o *download* do arquivo foi efetuado de

forma mais rápida. Isto é explicado por duas premissas: i) a imediata reserva de largura de banda para a execução do arquivo e ii) previamente já ter reserva de banda para o usuário do Moodle, o que permitiu que não estivesse congestionado o canal de comunicação.

Para os vídeos de média resolução (intervalo J-K) e alta resolução (intervalo L-M), a diferença do tempo de *download* dos arquivos foi considerável, principalmente para o arquivo de alta resolução, em que as rajadas de transferência do cenário 1 não se repetiram no cenário 3, dado que a concorrência pelo canal de comunicação no cenário 1 fez com que o navegador Web do dispositivo do cliente levasse mais tempo para finalizar a transferência do arquivo no servidor do Moodle.

5.3.4 Cenário 4

O último cenário, composto por dois usuários do Moodle executando atividades simultaneamente, é composto por uma topologia de rede constituída de 4 *switches* e 6 computadores, apresentada na Figura 18. Assim como no cenário 3, o *switch* raiz da rede contém o protocolo OpenFlow ativado, com o plano de controle gerenciado pelo controlador OpenFlow. Este, por sua vez, recebe as ações a serem executadas enviadas pela SDN4Moodle.

As reservas de largura de banda para os eventos do Moodle foram os mesmos do cenário 3, sendo 2 Mbps a partir do *login*, 4 Mbps para atividades de texto, *streaming* de áudio e vídeo de baixa resolução, 8 Mbps para vídeo de média resolução e 16 Mbps para vídeo de alta resolução. Como a reserva de dois usuários executando o vídeo de alta resolução (32 Mbps) ultrapassa o total disponível na rede (25 Mbps), foi definido que nesta situação a reserva para o tráfego secundário seria de 0 Mbps.

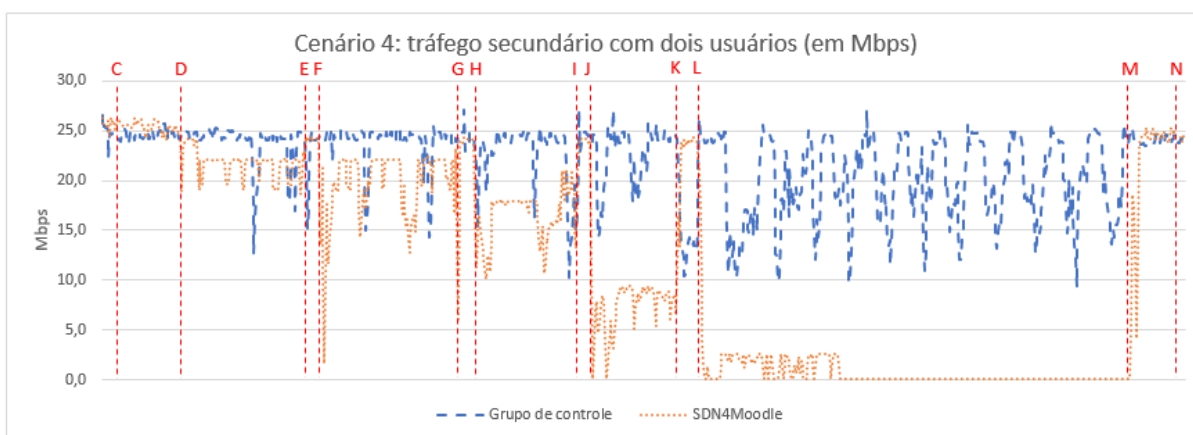


Figura 26 – Cenário 4: gráfico com o comparativo da largura de banda do tráfego secundário do *iperf* com dois usuários, com e sem o controle da SDN4Moodle (vazão em Mbps).

Apesar do cenário 3 mostrar que a largura de banda consumida pelo tráfego

secundário é reduzida quando ocorre os eventos do Moodle, ao utilizar o controle pela SDN4Moodle, é com a concorrência de mais de um usuário do Moodle que o resultado é realçado. Na Figura 26, a qual compara a largura de banda do *iperf* com e sem o uso do controle proposto neste trabalho, é possível perceber que à medida que os usuários executam as atividades, maior é a queda da vazão do tráfego secundário.

No vídeo de baixa resolução (intervalo H-I), o máximo alcançado pelo *iperf* foi de cerca de 17 Mbps, o que corresponde ao total da largura de banda de rede (25 Mbps) menos o total alocado para os usuários do Moodle (4 Mbps para cada). Já para o vídeo de média resolução (intervalo J-K), o tráfego secundário máximo foi de cerca de 9 Mbps, o que também condiz com as regras de reserva de largura de banda, sendo 25 Mbps menos os 8 Mbps para cada usuário do Moodle.

Na execução do vídeo de alta resolução (intervalo L-M), cuja reserva para o tráfego secundário seria menor que 0 Mbps (−7 Mbps), ocasionou a alocação de todo o canal de comunicação para os usuários Moodle, em detrimento do *software iperf*. Por fim, no *logoff* dos usuários (N), não foi percebido impactado na rede com o controle da SDN4Moodle.

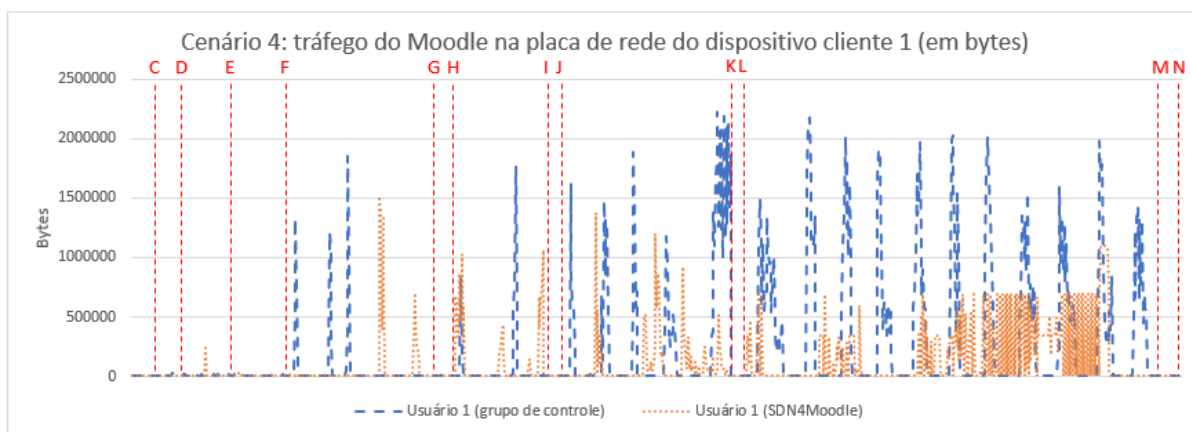


Figura 27 – Cenário 4: comparativo dos dados do Moodle trafegados na placa de rede do dispositivo do usuário 1 (em bytes).

As Figuras 27 e 28 apresentam o comparativo dos dados do Moodle trafegados na placa de rede dos usuários 1 e 2, respectivamente, do cenário 4. Diferentemente do cenário 3, em que apenas um usuário navega no Moodle, ocorreram diversas rajadas em tempos diferentes durante a execução dos eventos, como por exemplo foi na execução do vídeo de baixa resolução, em que a transferência ocorreu em tempos diferentes para o dispositivo do cliente 1. O mesmo comportamento foi percebido no segundo usuário (Figura 28).

Como principal diferença dos Cenários 2 e 4, percebe-se o tempo menor de transferência do vídeo de alta resolução. Tanto a transferência do usuário 1 quanto a

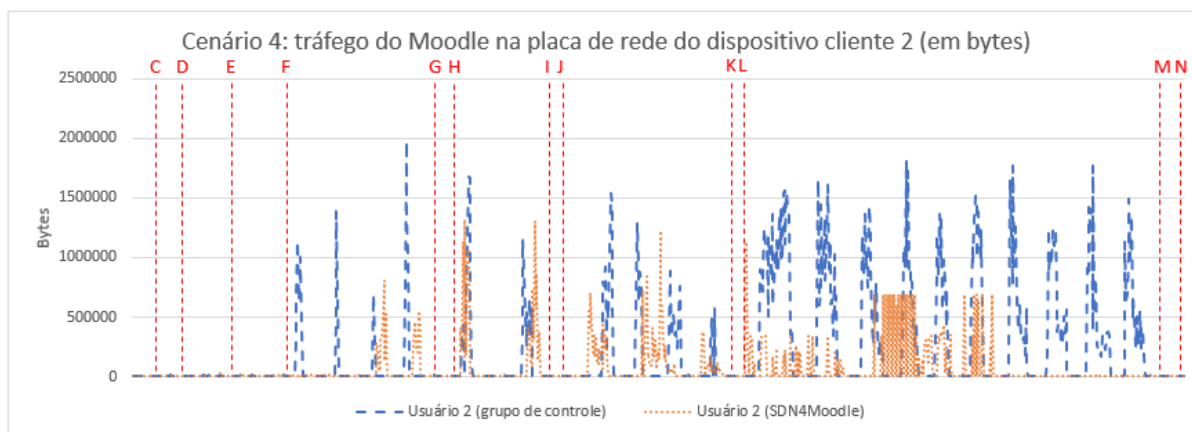


Figura 28 – Cenário 4: comparativo dos dados do Moodle trafegados na placa de rede do dispositivo do usuário 2 (em bytes).

do usuário 2 tiveram comportamentos semelhantes, chegando à taxa de transferência de cerca de 7 Mbps em ambos os gráficos. Percebe-se também que não foi utilizado toda a largura de banda disponível no canal de comunicação, uma vez que a soma das taxas de ambos os usuários foi somente cerca de 56% do total disponível.

5.4 LIMITAÇÕES TÉCNICAS ENCONTRADAS

Apresentados os resultados da análise experimental deste trabalho, nesta seção serão abordadas as limitações técnicas encontradas durante a realização do projeto de desenvolvimento da ferramenta SDN4Moodle. Durante a sua implementação, foram encontradas diversas restrições para a sua conclusão. Além das limitações descobertas, também são indicadas as soluções para contornar estes problemas durante o seu desenvolvimento.

O tráfego secundário da análise experimental deste capítulo foi gerado pelo *software iperf*, da mesma forma como realizado na Seção 2.1.3, com o servidor e o cliente em cada uma das folhas da topologia. Essa geração de ruído na rede ocasiona a ocupação de toda a largura de banda disponível no enlace. O papel da SDN4Moodle é priorizar os dados da aplicação Moodle, em detrimento do tráfego secundário do *iperf*. Como este *software* pode ser configurado para enviar tanto tráfego TCP quanto UDP, foi necessário definir qual cenário seria composto. Inicialmente a escolha mais apropriada foi o protocolo UDP, pois com a alocação adequada de largura de banda para os usuários do Moodle através da ferramenta SDN4Moodle, seria impossibilitado o sequestro do meio de transmissão pelo *iperf*.

Ao efetuar a coleta dos dados, foi possível a realização dos testes com a ferramenta proposta por este trabalho, contudo não foi viável a coleta dos dados da execução do vídeo de alta resolução no cenário tradicional (grupo de controle). Ao

iniciar a execução deste vídeo, o navegador Web do usuário encerrava a conexão TCP com o servidor do Moodle antes de concluir o *download* do arquivo, em virtude do meio de transmissão estar saturado com o tráfego secundário do *iperf* e não ser possível armazenar o *buffer* para a execução no *Media Player* do Moodle. Posto isto, não seria possível a comparação dos resultados entre os dois cenários, sendo então definido o protocolo TCP para ser executado pelo *iperf*.

Como apresentado na Tabela 19, os dispositivos utilizados foram do tipo domésticos, os quais não comportam tráfego de rede e processamento de dados consideráveis, quando virtualizados (PFAFF et al., 2009). Ao alcançar tráfego OpenFlow na faixa dos 30 Mbps, o processador do *switch* OpenFlow atinge 100% de uso e começa a descartar os pacotes que chegam nas suas interfaces. O processo responsável por esta sobrecarga de processamento é o próprio *daemon* do Openflow. Como não havia disponível um meio de reduzir a prioridade deste processo durante a sua execução, optou-se por definir como 25 Mbps a largura da banda global da SDN.

Além da limitação de processamento e memória dos dispositivos utilizados, outra restrição encontrada foi a da quantidade de interfaces disponíveis para encadear mais dispositivos de rede e computadores. Com um total de 5 interfaces disponíveis nos *switches*, a primeira é utilizada para a comunicação com o controlador SDN, conhecida como interface *out-of-band*, para tráfego administrativo da rede. Nos *switches* de borda a segunda interface é utilizada para se conectar ao *switch* raiz. Já a terceira interface conecta o servidor ou cliente Moodle. Por fim, a quarta interface conecta o servidor ou cliente *iperf*, restando apenas uma interface para expansão, que no caso foi utilizada para os cenários 2 e 4, com dois usuários do Moodle. Sendo assim, topologias com um maior número de dispositivos e mais complexas não poderiam ser implementadas, em virtude da restrição do número de dispositivos a serem encadeados.

Considerando que 25 Mbps de tráfego total é pouco mais que o alcançado em uma rede 10 Mbps *full duplex*, a solução proposta não é escalável para vazões maiores em razão da limitação do *hardware* utilizada para a análise apresentada neste capítulo. Apesar do LabP2D ter adquirido um equipamento DELL padrão da indústria, com capacidade de transferência de 256 Gbps e taxa de encaminhamento de 193 Mpps (modelo N3048p), este mostrou-se incapaz de atender as demandas do projeto, uma vez que o protocolo OpenFlow embarcado não implementou as métricas para os fluxos e as filas de QoS não podem ser acrescentadas nem modificadas via *software*. Por se tratar de um dispositivo robusto, com capacidade de encaminhamento de 260 Gbps (DELL, 2017b), o DELL N3048p poderia ter sido utilizado para a análise experimental e ter apresentado resultados diferentes dos alcançados neste trabalho.

Além do *switch* da DELL, foi verificada também a disponibilidade do uso de um equipamento da marca HP (série A5500EI), cujo fabricante atualizou o *firmware* para

ter suporte ao protocolo OpenFlow versão 1.3. Ao efetuar a atualização do *firmware*, foi possível que o seu plano de controle fosse gerenciado pelo OpenDayLight, definindo quais VLANs participariam da SDN. Apesar da comunicação satisfatória entre os computadores conectados na SDN, a criação e atualização de métricas não funciona com o controlador OpenDayLight, fazendo com que a intenção do uso do equipamento HP fosse abandonada.

Com um conjunto limitado de atributos a serem definidos, as métricas no protocolo OpenFlow versão 1.3 permitem apenas a definição de largura de banda máxima nos fluxos aos quais fazem parte. Estratégias de reserva de largura de banda mínima, essenciais em QoS, não foram possíveis serem aplicadas, uma vez que o *hardware* disponível não comportava tal configuração via protocolo OpenFlow. Com um *hardware* robusto, haveria a possibilidade da execução dos testes de forma plena, com mais usuários, até saturar a largura de banda disponível na interface dos *switches* (geralmente 1 Gbit/s ou até mesmo 10 Gbit/s).

Para contornar esta limitação, foi utilizado o método de definir a reserva de banda para os usuários como sendo a máxima disponível na topologia, para todos os usuários. Já para o tráfego secundário do *iperf*, define-se a sua largura de banda máxima disponível como sendo o total da topologia menos o já alocado para os usuários do Moodle. Esta decisão foi necessário em virtude de não haver como implementar filas nos *switches* nem atribuir largura de banda mínima nas métricas, além de não haver algum outro *hardware* disponível que tivesse estas especificações no seu protocolo OpenFlow embarcado.

Para minimizar eventuais conflitos no uso da largura de banda entre os usuários do Moodle, dado que teoricamente sempre é alocada n (sendo n a quantidade de usuários) vezes a largura de banda disponível na rede, foram definidas prioridades nos fluxos de acordo com o tipo de atividade que o usuário está trafegando na rede. As prioridades foram definidas como *login* (0), atividade de texto (1), execução de *streaming* de áudio (2), vídeo de baixa resolução (3), vídeo de média resolução (4), vídeo de alta resolução (6) e outros tipos de eventos (6), sendo o menor número a maior prioridade na rede. Esta técnica utilizada foi a que mais se aproximou das filas de *switches*.

5.5 CONSIDERAÇÕES PARCIAIS

Este capítulo apresentou a análise experimental da ferramenta SDN4Moodle em uma topologia SDN. Primeiramente foram definidas as métricas para análise, sendo escolhidas a eficácia da reserva de largura de banda e a ocorrência de perda de pacotes na aplicação de regras de fluxos. Estas foram selecionadas para quantificar a

aplicabilidade da ferramenta em um cenário experimental.

O ambiente de testes definido para a análise foi uma topologia de rede com dispositivos físicos que permitiram a execução dos testes com e sem o controle efetuado pela SDN4Moodle em uma SDN. Para esta mesma topologia foram definidos 4 cenários de testes, compostos por dois sendo de linha de base para as comparações e dois com o controle efetuado pela SDN4Moodle.

As atividades escolhidas para a coleta dos dados de navegação dos usuários do Moodle para a análise foram ações habituais para quem utiliza o LMS: realizar uma avaliação de texto, executar um *streaming* de áudio e assistir um vídeos com três resoluções diferentes, os quais foram arbitrados como baixa, média e alta resolução para a reserva de largura de banda pela SDN4Moodle.

Os resultados alcançados em ambos os cenários (com 1 e 2 usuários) com o controle de reserva de largura de banda em tempo de execução pela SDN4Moodle foram satisfatórios para demandas de maior largura de banda na rede. Para eventos com pouco consumo de rede, tais como atividades de texto, *login* e *logoff* de usuário, ao observar os dados obtidos na placa de rede do usuário, verifica-se que o tempo de transferência foi próximo do observado com o grupo de controle (com a SDN4Moodle desativada).

Isto acontece porque o tempo de transferência dos dados para o dispositivo do usuário precisa ser maior que o tempo de clique, o qual corresponde ao intervalo em que o mecanismo da SDN4Moodle recebe o evento do Moodle e cadastra as regras no *switch* OpenFlow. Apesar de já ter largura de banda reservada para o usuário, o tempo de reação da rede, para processar as regras e efetivamente aplicar as reservas de largura de banda, é maior que o tempo total do *download* dos dados. No melhor caso, a quantificação do tempo de reação do sistema é de 1 segundo, enquanto que no pior caso é de n eventos.

Para ações com maior consumo de rede, tais como a execução do *streaming* de áudio e do vídeo em diferentes resoluções, já é possível perceber claramente a vocação da SDN4Moodle: permitir que o Moodle exponha à rede as suas necessidades, informando quais são os requisitos quanto à largura de banda para os seus usuários e como estes dados devem ser priorizados quando ocorrer situações de congestionamentos na topologia de rede. Como limitação da análise, pode-se elencar que não houve a possibilidade de quantificar o número de interrupções dos vídeos durante a suas execuções, sendo listada esta tarefa com um trabalho futuro.

Além da capacidade de funcionar como um intermediário entre a aplicação alvo e a rede, a SDN4Moodle conseguiu, com as regras de reserva de largura de banda em tempo de execução, fazer com que a essência da justiça do protocolo TCP

fosse aplicada na rede, mudando o comportamento das rajadas do tráfego do Moodle no grupo de controle para um tráfego constante e igualitário entre os dois usuários do Moodle no cenário 4, em detrimento do tráfego secundário do *iperf* quando houver a real necessidade de priorização do tráfego Moodle. Esta priorização de tráfego também poderia ser realizada para outros LMSs, caso houvesse o envio dos eventos para a SDN4Moodle.

6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Este trabalho apresentou uma proposta de controle da qualidade de serviço para SDNs da aplicação Moodle, com a reserva de largura de banda em tempo de execução baseada nos eventos informados pela aplicação. Para tal, foi implementado um protótipo para analisar a viabilidade da proposta em um ambiente controlado de testes, denominado SDN4Moodle. Foram desenvolvidos dois módulos de *software*, responsáveis pela coleta dos dados dos eventos da navegação dos usuários da aplicação alvo (Módulo 1 - *plugin* do Moodle) e pela configuração do encaminhamento de fluxos dos usuários e dos dispositivos geradores de tráfego secundário (Módulo 2 - SDN4Moodle).

A implementação do protótipo foi avaliada em uma topologia SDN com dispositivos físicos reais, com cenários de 1 e 2 usuários que realizaram atividades típicas de um sistema acadêmico, para que suas navegações fossem analisadas com e sem o controle da qualidade de serviço efetuado pela ferramenta SDN4Moodle. Para cada tipo de ação (*login*, realização de atividades de texto e execução de arquivos multimídia), foram definidas diferentes larguras de banda para serem alocadas para os usuários do Moodle assim que ocorressem estas ações.

Além de avaliar o comportamento do tráfego secundário durante a navegação dos usuários do Moodle na rede com o uso da ferramenta SDN4Moodle, também foi analisado o comportamento da transferência dos dados do Moodle na placa de rede dos dispositivos clientes. A proposta se mostrou proveitosa em ambos os casos, sendo capaz de aplicar regras de encaminhamento guiadas pela aplicação e também a redução do tempo de transferência de arquivos multimídia, respectivamente.

Desta forma, a proposta desta dissertação oferece contribuições significantes, demonstrando ser viável o uso das SDNs para melhorar o desempenho das aplicações que utilizam a rede de computadores. Esta melhoria se deve ao fato da capacidade da SDN4Moodle interpretar as demandas de rede informadas pelo Moodle e converter estas necessidades em regras de QoS, em tempo de execução, de uma forma elegante e inovadora, sem a necessidade da intervenção do administrador da rede.

Destaca-se que, com a elaboração deste trabalho, foi possível desenvolver a capacidade da aplicação Moodle de informar à rede as demandas de reserva de largura de banda para os seus usuários, em tempo de execução e sem a necessidade da intervenção do administrador da rede. Também não é necessário a reconfiguração dos dispositivos de rede para que as reservas sejam aplicadas. Assim, espera-se que a ferramenta seja expandida para suportar outras aplicações que utilizam a rede para

comunicação, bem como outras oportunidades de trabalhos futuros que são apresentados na próxima seção.

6.1 TRABALHOS FUTUROS

Durante o desenvolvimento desta dissertação foram identificadas diversas oportunidade de expansão da ferramenta SDN4Moodle e as suas eventuais implementações em trabalhos futuros. A primeira delas é expandir o suporte da ferramenta a outros controladores SDN utilizados pela comunidade científica, além do OpenDayLight, tais como o Floodlight, o Ryu, o NOX, entre outros.

Como a SDN4Moodle foi concebida para ser um *framework* para a reserva de largura de banda para qualquer aplicação que utilize a rede de computadores, o seu código-fonte foi escrito da forma mais genérica possível, com o processamento de apenas eventos do Moodle. Não foi tomada essa decisão em virtude da necessidade de se fechar o escopo do trabalho com o cronograma disponível para a conclusão do Mestrado. No entanto, a aplicação pode ser expandida em trabalhos futuros para reservar largura de banda de outras aplicações, com o eventual nome de SDN4App.

A ferramenta desenvolvida foi projetada para o protocolo OpenFlow versão 1.3. Uma outra oportunidade de trabalho futuro é o suporte para as mais recentes, tais como a versão 1.5. Além da atualização do protocolo OpenFlow, também pode-se mencionar a ampliação dos modelos de dispositivos de rede compatíveis com a ferramenta. Para isto seria necessária a identificação das particularidades de cada um dos dispositivos em que se deseja a comunicação com a SDN4Moodle. Também será submetido o código-fonte *plugin* ao repositório oficial do Moodle.

Como principal funcionalidade para a expansão da ferramenta, pode-se dizer que é a capacidade de identificar o usuário a partir da conexão na rede, e não na autenticação na aplicação alvo. Para tal, é necessário que a ferramenta consiga capturar outros tipos de *logs* na rede, uma vez que só há a captura de eventos disponibilizados pela API do Moodle. Como exemplos de eventos a serem processados futuramente, pode-se citar autenticações e desconexões armazenadas em servidores LDAP, radius, *captive portal*, syslog, bancos de dados, além dos dispositivos de rede (pontos de acesso sem fio e *switches* com o padrão 802.1x ativados).

Desta forma é possível reservar largura de banda para as aplicações que o usuário utiliza na rede, e não para o seu dispositivo, como é feito atualmente nas redes tradicionais. O administrador da rede poderia estabelecer até mesmo cotas globais de largura de banda para aplicações ou por usuário, sem a necessidade de configurar os dispositivos da rede.

Também como trabalho futuro foi percebida a oportunidade de se considerar

mais de uma atividade simultânea na reserva de largura de banda. No protótipo desenvolvido é considerado como critério apenas a última ação realizada pelo usuário. Nesta situação a reserva de largura de banda poderia ser a soma da reserva de todas as ações que usuário estivesse realizando. Já com relação às análises realizadas, uma oportunidade de trabalho futuro é a avaliação e medição do impacto da retransmissão dos dados do Moodle com e sem o controle da ferramenta SDN4Moodle em situações de congestionamento do canal de comunicação.

Há a possibilidade e trabalho futuro da execução dos testes com equipamentos homogêneos, tanto dos servidores quanto dos clientes Moodle e *iperf*. Dado que a análise experimental deste trabalho foi efetuada com hardwares de arquiteturas diferentes, esta variação pode influenciar nos resultados alcançados. Também pode ser elencado como trabalho futuro a criação de testes de limiares, para tratar impasses de usuários com a mesma prioridade quando não há mais recursos disponíveis. Além disso, há a oportunidade de trabalho futuro em estimar tempos individuais para cada etapa da análise experimental.

Por fim, destaca-se a possibilidade de trabalho futuro em quantificar o número de interrupções durante a execução vídeos, com e sem o controle da ferramenta SDN4Moodle. Isto pode ser realizado com a utilização de *player* específico no navegador Web.

6.2 CONTRIBUIÇÃO

Durante o desenvolvimento desta dissertação foram realizadas algumas publicações com o objetivo de apresentar à comunidade o escopo e os resultados alcançados da proposta desenvolvida. Foram publicados os seguintes artigos em eventos, sendo dois destes internacionais:

- Marcondes, Anderson; Koslovski, Guilherme. Caracterização de topologias virtuais em Redes Definidas por Software: um estudo preliminar com o benchmark NAS. In: XVI Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul - ERAD/RS 2016.
- Marcondes, Anderson; Diel, Gustavo; de Souza, Felipe; Vieira, Paulo; Koslovski, Guilherme; Fiorese, Adriano. Análise de Aplicações Distribuídas em SDN: Um Estudo com o Benchmark NAS. Proceedings of the VI Brazilian Symposium on Computing Systems Engineering - SBESC 2016.
- Marcondes, Anderson; Diel, Gustavo; de Souza, Felipe; Vieira, Paulo; Koslovski, Guilherme; Fiorese, Adriano. Executing Distributed Applications on SDN-based

Data Center: a Study with NAS Parallel Benchmark. 7th International Conference on Network of the Future - NoF 2016.

- Marcondes, Anderson; Koslovski, Guilherme. Encaminhamento de Fluxos em Redes Definidas por Software: Estudo com um Sistema de Gestão de Aprendizagem. XVII Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul - ERAD/RS 2017.
- Marcondes, Anderson; Koslovski, Guilherme. Controle da Qualidade de Serviço para o Sistema Moodle em Redes Definidas por Software. XVIII Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul - ERAD/RS 2018.
- Marcondes, Anderson; Koslovski, Guilherme; Miers, Charles; Pillon, Maurício. SDN4Moodle: An SDN-based Toolset to Enhance QoS of Moodle Platform. IEEE Symposium on Computers and Communications - IEEE ISCC 2018.

Também foram publicados trabalhos relacionados a este trabalho, com participação de coautoria:

- Vieira Junior, Paulo Roberto; Fiorese, Adriano; Marcondes, Anderson; Koslovski, Guilherme. Uma Abordagem para Redução de Consumo de Recursos de Serviços de Voz sobre IP baseada em Redes Definidas por Software. XVII Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul - ERAD/RS 2017.
- Diel, Gustavo; Marcondes, Anderson; Koslovski, Guilherme. Uma Ferramenta para Evitar Enlaces Congestionados em Redes Definidas por Software. XVII Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul - ERAD/RS 2017.
- Vieira Junior, Paulo Roberto; Fiorese, Adriano; Marcondes, Anderson; Koslovski, Guilherme. SDNVolP: gerenciamento de recursos de serviços de Voz sobre IP baseado em Redes Definidas por Software. XXII Workshop de Gerência e Operação de Redes e Serviços - Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC 2017 WGRS.

REFERÊNCIAS

- AL-FARES, M.; LOUKISSAS, A.; VAHDAT, A. A scalable, commodity data center network architecture. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 38, n. 4, p. 63–74, ago. 2008. ISSN 0146-4833. Disponível em: <<http://doi.acm.org/10.1145/1402946.1402967>>.
- ALIZADEH, M. et al. Data center tcp (dctcp). In: ACM. **ACM SIGCOMM computer communication review**. [S.l.], 2010. v. 40, p. 63–74.
- ALIZADEH, M. et al. Data center tcp (dctcp). **SIGCOMM Comput. Commun. Rev.**, ACM, v. 41, n. 4, p. —, ago. 2010. ISSN 0146-4833.
- AVILA, I.; DIAS, K.; SIMOES, R. Sdncloud: plataforma ead com suporte à computação em nuvem e openflow. In: **Computer on the Beach 2018 ()**. [S.l.: s.n.], 2018.
- BAILEY, D. H. et al. The NAS Parallel Benchmarks. **Int. Journal of High Performance Computing Applications**, 1991.
- BAILEY, G. D. Wanted: A road map for understanding integrated learning systems. **Educational Technology**, ERIC, v. 32, n. 9, p. 3–9, 1993.
- BARI, M. F. et al. Polycycop: an autonomic qos policy enforcement framework for software defined networks. In: IEEE. **Future Networks and Services (SDN4FNS), 2013 IEEE SDN for**. [S.l.], 2013. p. 1–7.
- BEZERRA, J. de M. et al. Engenharia de tráfego em redes definidas por software. **Minicursos do XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, SBC, v. 34, n. 1, p. 101–150, maio 2016. ISSN 2177-4978. Disponível em: <<http://www.sbrc2016.ufba.br/downloads/anais/MinicursosSBRC2016.pdf>>.
- BLACKBOARD. **Blackboard Learn**. 2017. Disponível em: <<https://www.blackboard.com>>.
- BRADEN, R.; CLARK, D.; SHENKER, S. **RFC 1633 Integrated Services in the Internet Architecture: An Overview**. [S.l.], 1994. 1-33 p. Disponível em: <<https://www.ietf.org/rfc/rfc1633.txt>>.
- BROWN, A.; WILSON, G. **The architecture of open source applications, volume ii**. [S.l.]: Kristian Hermansen, 2012. v. 2. ISBN 1105571815.
- CANVAS. **Canvas Learning Management System**. 2017. Disponível em: <<https://www.canvaslms.com/>>.
- CERF, V. G.; ICAHN, R. E. A protocol for packet network intercommunication. **ACM SIGCOMM Computer Communication Review**, ACM, v. 35, n. 2, p. 71–82, 2005.
- CHIU, D.-M.; JAIN, R. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. **Computer Networks and ISDN systems**, Elsevier, v. 17, n. 1, p. 1–14, 1989.

DAVOLI, R.; GOLDWEBER, M. Msocket: Multiple stack support for the berkeley socket api. In: ACM. **Proceedings of the 27th Annual ACM Symposium on Applied Computing**. [S.l.], 2012. p. 588–593.

DEERING, S.; HINDEN, R. **Internet Protocol, Version 6 (IPv6) Specification**. [S.l.], 1998. 1-38 p. Disponível em: <<https://www.ietf.org/rfc/rfc2460.txt>>.

DELL. **Dell EMC Networking N-Series N1100-ON, N1500, N2000, N2100-ON, N3000, N3100-ON, and N4000 Switches CLI Reference Guide Rev. A03**. [S.l.], 2017. Disponível em: <http://downloads.dell.com/manuals/common/networking_nxxcli_en-us.pdf>.

DELL. **Dell EMC Networking N3000 Series Switches**. [S.l.], 2017. Disponível em: <http://i.dell.com/sites/doccontent/shared-content/data-sheets/en/Documents/Dell_Networking_N3000_Series_SpecSheet.pdf>.

DOUGIAMAS, M.; TAYLOR, P. Moodle: Using learning communities to create an open source course management system. In: **Proceedings of the EdMedia: World Conference on Educational Media and Technology, Honolulu, Hawaii, 2003**. [S.l.: s.n.], 2003.

DUKKIPATI, N. et al. Proportional rate reduction for tcp. In: **Proceedings of the 11th ACM SIGCOMM Conference on Internet Measurement 2011, Berlin, Germany - November 2-4, 2011**. [s.n.], 2011. Disponível em: <<http://conferences.sigcomm.org/imc/2011/program.htm>>.

EDUCAÇÃO, M. da. **Prêmio Professores do Brasil é tema do Educação no Ar**. 2017. Disponível em: <<https://www.youtube.com/watch?v=2Wo3wacqfys>>.

EGILMEZ, H. E. et al. Openqos: An openflow controller design for multimedia delivery with end-to-end quality of service over software-defined networks. In: **Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific**. [S.l.: s.n.], 2012. p. 1–8.

ENGHARDT, T. **Socket intents: Extending the socket API to express application needs**. Tese (Doutorado) — MS thesis, Dept. Telecommun. Syst., Tech. Univ. Berlin, Berlin, Germany, 2013.

FAINELLI, F. The openwrt embedded development framework. In: **Proceedings of the Free and Open Source Software Developers European Meeting**. [S.l.: s.n.], 2008.

FOROUZAN, B. A. **Comunicação de dados e redes de computadores**. 4th. ed. [S.l.]: AMGH Editora, 2008.

FOUNDATION, A. S. **The Apache Software Foundation**. 2018. Disponível em: <<https://www.apache.org/>>.

FOUNDATION, O. N. **OpenFlow Specification Version 1.5.1 (Protocol version 0x06)**. 2015. Disponível em: <<https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf>>.

GIL, A. C. **Como elaborar projetos de pesquisa**. 4st. ed. [S.l.]: Atlas, 2007. ISBN 9788522431694.

GROSSMAN, D. **New Terminology and Clarifications for Diffserv**. [S.l.], 2002. 1-10 p. Disponível em: <<https://www.ietf.org/rfc/rfc3260.txt>>.

GUDE, N. et al. Nox: Towards an operating system for networks. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 38, n. 3, p. 105–110, jul. 2008. ISSN 0146-4833. Disponível em: <<http://doi.acm.org/10.1145/1384609.1384625>>.

HANDLEY, M. Why the Internet only just works. **BT Technology Journal**, Kluwer Academic Publishers, Hingham, MA, USA, v. 24, p. 119–129, July 2006. ISSN 1358-3948. Disponível em: <<http://portal.acm.org/citation.cfm?id=1188052.1188101>>.

HIGGINS, B. D. et al. Intentional networking: opportunistic exploitation of mobile network diversity. In: ACM. **Proceedings of the sixteenth annual international conference on Mobile computing and networking**. [S.l.], 2010. p. 73–84.

IEEE. **IEEE Std 829-2008 - IEEE Standard for Software and System Test Documentation**. [S.l.], 2008.

IEEE. **802.1Q-2014 - IEEE Standard for Local and metropolitan area networks—Bridges and Bridged Networks**. [S.l.], 2014. v. 2014.

IFC. **Moodle - IFC**. 2018. Disponível em: <<https://fic.ifc.edu.br/moodle/login/index.php>>.

IFSC. **Moodle - IFSC**. 2018. Disponível em: <<https://moodle.ifsc.edu.br/>>.

JAIN, M. et al. Practical, real-time, full duplex wireless. In: ACM. **Proceedings of the 17th annual international conference on Mobile computing and networking**. [S.l.], 2011. p. 301–312.

JAIN, R.; PAUL, S. Network virtualization and software defined networking for cloud computing: a survey. **IEEE Communications Magazine**, IEEE, v. 51, n. 11, p. 24–31, 2013.

JARSCHER, M. et al. Sdn-based application-aware networking on the example of youtube video streaming. In: **2013 Second European Workshop on Software Defined Networks**. [S.l.: s.n.], 2013. p. 87–92. ISSN 2379-0350.

JEONG, K.; KIM, J.; KIM, Y.-T. Qos-aware network operating system for software defined networking with generalized openflows. In: IEEE. **2012 IEEE Network Operations and Management Symposium**. [S.l.], 2012. p. 1167–1174.

KIM, H.; FEAMSTER, N. Improving network management with software defined networking. **IEEE Communications Magazine**, IEEE, v. 51, n. 2, p. 114–119, 2013.

KREUTZ, D. et al. Software-defined networking: A comprehensive survey. **Proceedings of the IEEE**, IEEE, v. 103, n. 1, p. 14–76, 2015.

KUMAR, S.; DUTTA, K. Investigation on security in lms moodle. **International Journal of Information Technology and Knowledge Management**, v. 4, n. 1, p. 233–238, 2011.

KURAN, M. Ş.; PEDERSEN, J. M.; ELSNER, R. Learning management systems on blended learning courses: An experience-based observation. In: SPRINGER. **International Conference on Image Processing and Communications**. [S.l.], 2017. p. 141–148.

KUROSE, J. F.; ROSS, K. W.; ZUCCHI, W. L. **Redes de Computadores e a Internet: uma abordagem top-down**. [S.l.]: Pearson Addison Wesley, 2007.

MANKIN, A. et al. **RFC 2208 Resource ReSerVation Protocol (RSVP) Version 1 Applicability Statement Some Guidelines on Deployment**. [S.l.], 1997. Disponível em: <<https://www.ietf.org/rfc/rfc2208.txt>>.

MARCONDES, A. H. d. S. et al. Executing distributed applications on sdn-based data center: a study with nas parallel benchmark. In: IEEE. **Proceedings of the 7th International Conference on Network of the Future - NoF 2016**. [S.l.], 2016.

MARCONDES, A. H. da S. **Documentação SDN4Moodle**. 2018. Disponível em: <www.marcondes.pro.br/sdn4moodle>.

MCGILL, T. J.; KLOBAS, J. E. A task–technology fit view of learning management system impact. **Computers & Education**, Elsevier, v. 52, n. 2, p. 496–508, 2009.

MCKEOWN, N. et al. Openflow: Enabling innovation in campus networks. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 38, n. 2, p. 69–74, mar. 2008. ISSN 0146-4833. Disponível em: <<http://doi.acm.org/10.1145/1355734.1355746>>.

MEDVED, J. et al. Opendaylight: Towards a model-driven sdn controller architecture. In: **Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014**. [S.l.: s.n.], 2014.

MEKKY, H. et al. Application-aware data plane processing in sdn. In: **Proceedings of the Third Workshop on Hot Topics in Software Defined Networking**. New York, NY, USA: ACM, 2014. (HotSDN '14), p. 13–18. ISBN 978-1-4503-2989-7. Disponível em: <<http://doi.acm.org/10.1145/2620728.2620735>>.

MITTAL, R. et al. Timely: Rtt-based congestion control for the datacenter. In: ACM. **ACM SIGCOMM Computer Communication Review**. [S.l.], 2015. v. 45, p. 537–550.

MOODLE. **Installing Moodle 3.3**. 2017. Disponível em: <https://docs.moodle.org/33/en/Installing_Moodle>.

MOODLE. **Moodle - Open-source Learning Platform**. 2017. Disponível em: <https://www.moodle.org/?lang=pt_br>.

MOODLE. **Moodle Authentication**. 2017. Disponível em: <<https://docs.moodle.org/34/en/Authentication>>.

MOODLE. **Moodle Communication Between Components**. 2017. Disponível em: <https://docs.moodle.org/dev/Communication_Between_Components>.

MOODLE. **Moodle Documentation**. 2017. Disponível em: <<https://docs.moodle.org>>.

MOODLE. **Moodle Event 2**. 2017. Disponível em: <https://docs.moodle.org/dev/Event_2>.

MOODLE. **Moodle Override permissions**. 2017. Disponível em: <https://docs.moodle.org/34/en/Override_permissions>.

MOODLE. **Moodle Portfolios**. 2017. Disponível em: <<https://docs.moodle.org/34/en/Portfolios>>.

MOODLE. **Moodle Repositories**. 2017. Disponível em: <<https://docs.moodle.org/34/en/Repositories>>.

MOODLE. **Moodle Standard roles**. 2017. Disponível em: <https://docs.moodle.org/34/en/Standard_roles>.

MOODLE. **Moodle Statistics**. 2017. Disponível em: <<https://moodle.net/stats/>>.

MOORE, J. L.; DICKSON-DEANE, C.; GALYEN, K. e-learning, online learning, and distance learning environments: Are they the same? **The Internet and Higher Education**, Elsevier, v. 14, n. 2, p. 129–135, 2011.

MYSORE, R. N. et al. Portland: A scalable fault-tolerant layer 2 data center network fabric. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 39, n. 4, p. 39–50, ago. 2009. ISSN 0146-4833. Disponível em: <<http://doi.acm.org/10.1145/1594977.1592575>>.

OLIFER, N.; OLIFER, V. **Redes de computadores: princípios, tecnologias e protocolos para o projeto de redes**. 1th. ed. [S.l.]: LTC, 2008.

OPENEDUCATIONEUROPA. **183 Learning Management Systems (plus a handy comparison guide)**. 2013. Disponível em: <<https://www.openeducationeuropa.eu/en/blogs/list-183-learning-management-systems-plus-lmss-comparison-checklist-99-features>>.

PAPASTERGIOU, G. et al. De-ossifying the internet transport layer: A survey and future perspectives. **IEEE Communications Surveys & Tutorials**, IEEE, v. 19, n. 1, p. 619–639, 2017.

PFAFF, B. et al. Extending networking into the virtualization layer. In: **Proceedings of the 8th ACM Workshop on Hot Topics in Networks (HotNets-VIII)**. New York, NY, USA: [s.n.], 2009. p. 1–6. Disponível em: <<http://www.icsi.berkeley.edu/pubs/networking/extendingnetworking09.pdf>>.

POPA, L. et al. Faircloud: Sharing the network in cloud computing. In: **Proceedings of the 10th ACM Workshop on Hot Topics in Networks**. New York, NY, USA: ACM, 2011. (HotNets-X), p. 22:1–22:6. ISBN 978-1-4503-1059-8.

SALTZER, J. H.; REED, D. P.; CLARK, D. D. End-to-end arguments in system design. **ACM Trans. Comput. Syst.**, ACM, New York, NY, USA, v. 2, n. 4, p. 277–288, nov. 1984. ISSN 0734-2071. Disponível em: <<http://doi.acm.org/10.1145/357401.357402>>.

SHVACHKO, K. et al. The hadoop distributed file system. In: IEEE. **2010 IEEE 26th symposium on mass storage systems and technologies (MSST)**. [S.l.], 2010. p. 1–10.

STALLINGS, W. **Redes e sistemas de comunicação de dados**. 5. ed. Rio de Janeiro: Elsevier, 2005. ISBN 8535217312.

STALLINGS, W. **Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud**. 1st. ed. [S.l.]: Addison-Wesley Professional, 2015. ISBN 0134175395.

TIRUMALA, A. et al. Iperf: The tcp/udp bandwidth measurement tool. <http://dast.nlanr.net/Projects>, 2005.

UDESC. **Centro de Educação a Distância - UDESC**. 2018. Disponível em: <<https://www.moodle.udesc.br/>>.

UFFS. **Implantações do Moodle da UFFS**. 2018. Disponível em: <<https://moodle.uffs.edu.br/>>.

UFSC. **Moodle UFSC - Apoio aos cursos presenciais**. 2018. Disponível em: <<https://moodle.ufsc.br/>>.

VALERIANO, D. L. **Gerência em projetos: pesquisa, desenvolvimento e engenharia**. [S.l.]: Pearson Education, 2004.

VIEIRA, P. R.; FIORESE, A.; KOSLOVSKI, G. Comparação de políticas para configuração de fluxos em redes definidas por software. In: **Computer on the Beach 2016**. [S.l.: s.n.], 2016.

WANG, Y. et al. Application driven network: Providing on-demand services for applications. In: **Proceedings of the 2016 Conference on ACM SIGCOMM 2016 Conference**. New York, NY, USA: ACM, 2016. (SIGCOMM '16), p. 617–618. ISBN 978-1-4503-4193-6. Disponível em: <<http://doi.acm.org/10.1145/2934872.2959075>>.

WATASHIBA, Y. et al. Efficacy analysis of a sdn-enhanced resource management system through nas parallel benchmarks. **The Review of Socionetwork Strategies**, v. 8, n. 2, p. 69–84, 2014. ISSN 1867-3236. Disponível em: <<http://dx.doi.org/10.1007/s12626-014-0045-9>>.

YAO, G.; BI, J.; XIAO, P. Source address validation solution with openflow/nox architecture. In: IEEE. **2011 19th IEEE International Conference on Network Protocols**. [S.l.], 2011. p. 7–12.

YILDIRIM, A. S.; GIRICI, T. Cloud technology and performance improvement with int-serv over diffserv for cloud computing. In: **Proceedings of the Intl. Conf. on Cloud Computing and eGovernance**. [S.l.: s.n.], 2014. v. 1, n. 2014, p. 92.

ZHAO, S.; SYDNEY, A.; MEDHI, D. Building application-aware network environments using sdn for optimizing hadoop applications. In: **Proceedings of the 2016 Conference on ACM SIGCOMM 2016 Conference**. New York, NY, USA: ACM, 2016. (SIGCOMM '16), p. 583–584. ISBN 978-1-4503-4193-6. Disponível em: <<http://doi.acm.org/10.1145/2934872.2959059>>.

ANEXO A – DIAGRAMA ENTIDADE RELACIONAMENTO DA FERRAMENTA SDN4MOODLE

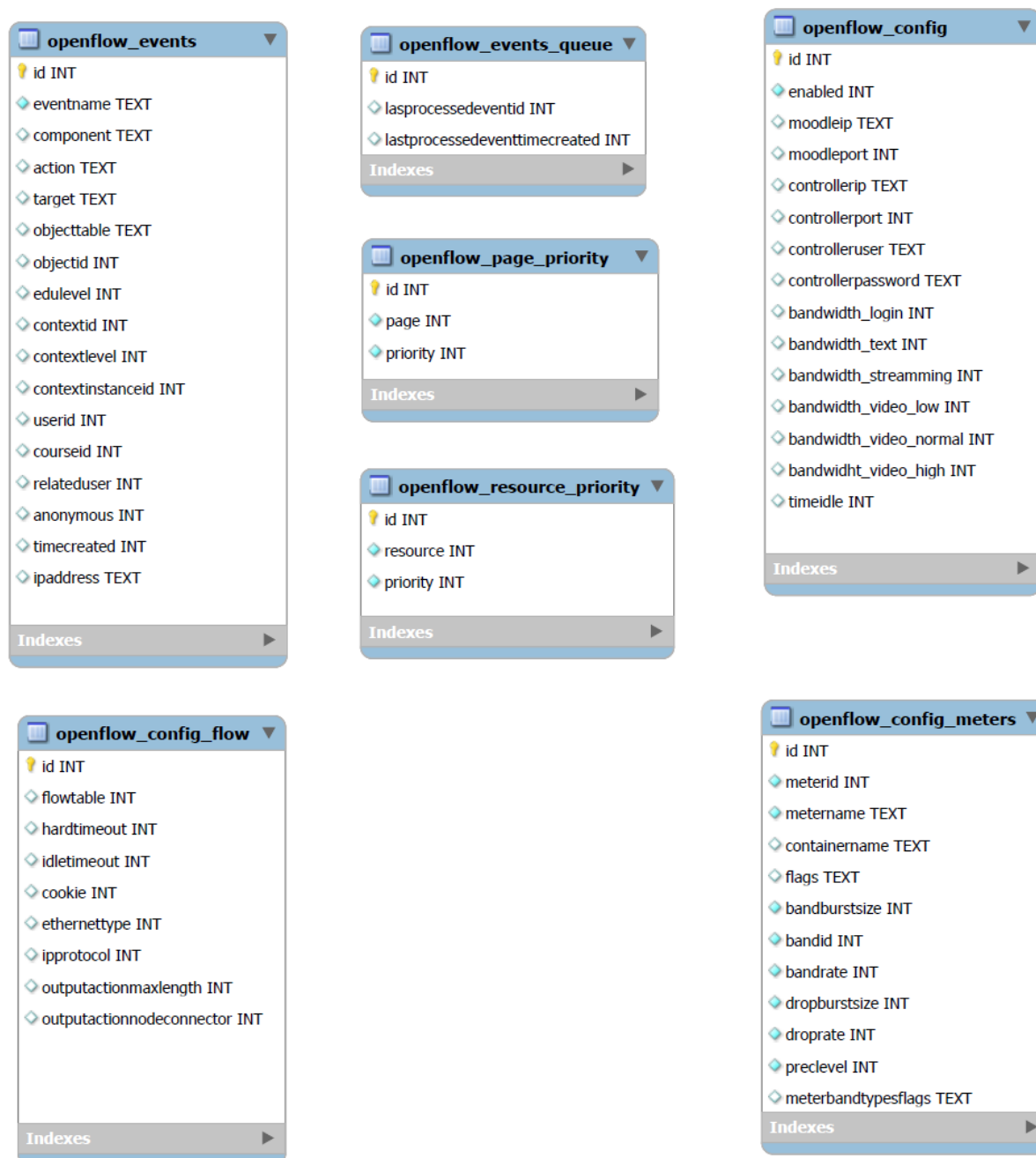


Figura 29 – Diagrama entidade relacionamento da ferramenta SDN4Moodle.

ANEXO B – DIAGRAMA DE CLASSES DA SDN4MOODLE

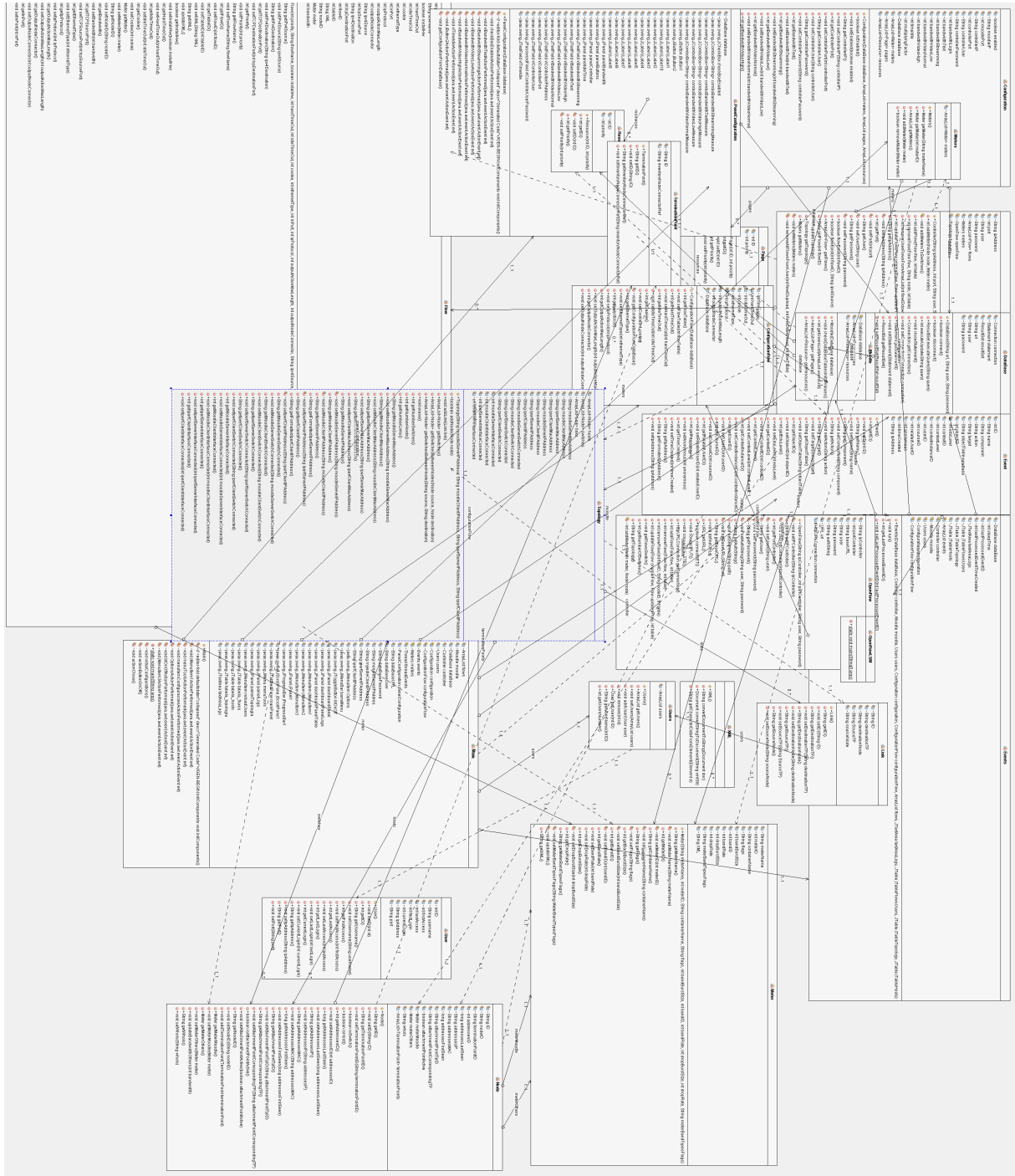


Figura 30 – Diagrama de Classes da ferramenta SDN4Moodle. Arquivo digital disponível em Marcondes (2018).