



13^a ESCOLA REGIONAL DE REDES DE COMPUTADORES
9–11 de setembro de 2015
Passo Fundo – RS

ANAIS

Editora
Sociedade Brasileira de Computação – SBC

Organizadores
Luis Augusto Dias Knob
Jéferson Campos Nobre
Juliano Araujo Wickboldt

Realização
Faculdade Meridional (IMED)
Universidade Federal do Rio Grande do Sul (UFRGS)

Editor
Lucas Bondan (UFRGS)

Promoção
Sociedade Brasileira de Computação – SBC

Copyright © 2015 Sociedade Brasileira de Computação
Capa: Lucas Bondan e Luis Augusto Dias Knob

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Escola Regional de Redes de Computadores (13.: 9–11 set 2015: Passo Fundo)

Anais / Organizadores: Luis Augusto Dias Knob, Jéferson Campos Nobre, Juliano Araujo Wickboldt. — : , 2015.

130 f.: il.

ISSN 2237-3748

Conhecido também como ERRC 2015.

- 1. Redes de Computadores. 2. Sistemas Distribuídos.
- I. ERRC (13.: 9–11 set 2015: Passo Fundo). II. **IMED**.
- III. Dias Knob, Luis Augusto. IV. Nobre, Jéferson Campos.
- V. Araujo Wickboldt, Juliano. VI. Título.

É proibida a reprodução total ou parcial desta obra sem o consentimento prévio dos autores

ERRC 2015

<http://errc2015.imed.edu.br>

Comitê de Programa

Adenauer Yamin (UFPEL)
Alberto Schaeffer-Filho (UFRGS)
Ana Cristina Benso da Silva (PUCRS)
André Peres (IFRS)
Andre Du Bois (UFPEL)
Andrea Charao (UFSM)
Andrea Krob (UNILASALLE)
Antonio Rodrigo Delepiane de Vit (UFSM)
Carlos Raniery Paula dos Santos (UFSM)
Claudio Schepke (Unipampa)
Clarissa Marquezan (Huawei Technologies, Germany)
Cristiano Bonato Both (UFCSPA)
Cristiano Costa (Unisinos)
Cristina Nunes (PUCRS)
Daniela Brauner (UFPEL)
Diego Kreutz (Unipampa)
Eduardo Back (IENH)
Eduardo Monks (FATEC SENAC - Pelotas)
Erico Amaral (Unipampa)
Erico Rocha (Unisinos)
Ewerton Salvador (UFMG)
Flávio Roberto Santos (Chaordic Systems)
Glederson Santos (IFSul - Charqueadas)
Guilherme Rodrigues (IFSul - Charqueadas)
Guilherme Sperb Machado (University of Zürich (UZH), Switzerland)
Iara Augustin (UFSM)
Jéferson Campos Nobre (UFRGS)
José Jair Cardoso de Santanna (University of Twente, The Netherlands)
Juliano Araujo Wickboldt (UFRGS)
Lisandro Zambenedetti Granville (UFRGS)
Lucas Bondan (UFRGS)
Luciano Ignaczak (Unisinos)
Luciano Paschoal Gaspary (UFRGS)
Lucio Prade (Unisinos)
Maicon Kist (UFRGS)
Marcelo Marotta (UFRGS)
Marcia Pasin (UFSM)
Marco Trentin (UPF)
Marinho Barcellos (UFRGS)
Mauricio Pilla (UFPEL)
Oscar Caicedo (Universidad del Cauca, Colombia)

Pedro Arthur Duarte (Hewlett Packard Enterprise)
Rafael Kunst (UNILASALLE)
Rafael Pereira Esteves (IFRS)
Raul Ceretta Nunes (UFSM)
Ricardo Becker (UCS/SENAI)
Ricardo Luis dos Santos (UFRGS)
Ricardo Neisse (IPSC - Joint Research Center (JRC))
Ricardo Schmidt (University of Twente, The Netherlands)
Roben Lunardi (IFRS)
Rodrigo Calheiros (The University of Melbourne, Australia)
Rodrigo Righi (Unisinos)
Rogério Turchetti (UFSM)
Taisy Weber (UFRGS)
Tiago Ferreto (PUCRS)
Vinícius Guimarães (IFSul - Charqueadas)
Vinicius Ribeiro (Ritter dos Reis)
Walter Priesnitz Filho (UFSM)
Weverton Luis da Costa Cordeiro (UFRGS)

Comitê Organizador

Coordenação Geral

Luis Augusto Dias Knob (IMED)

Organização Local

Marcos Roberto dos Santos (IMED)

Marco Antônio Sandini Trentin (UPF)

Organização do Comitê de Programa

Jéferson Campos Nobre (UFRGS)

Juliano Araujo Wickboldt (UFRGS)

Organização de Divulgação e Patrocínios

Eduardo Santos Back (IENH)

Roben Castagna Lunardi (IFSUL – Restinga)

Organização de Minicursos, Palestras e Oficinas

Cristiano Bonato Both (UFCSPA)

Weverton Luis da Costa Cordeiro (UFRGS)

Comissão de Organização

Lucas Bondan (UFRGS)

Lucas Pereira (IMED)

Mateus Scherner (IMED)

Revisores

Adenauer Yamin (UFPEL)
Alberto Kummer Neto (UFSM)
Alberto Schaeffer-Filho (UFRGS)
Ana Cristina Benso da Silva (PUCRS)
Anderson da Silva (UFRGS)
André Peres (IFRS)
Andre Du Bois (UFPEL)
Andrea Charao (UFSM) Andrea Krob (UNILASALLE)
Carlos Raniery Paula dos Santos (UFSM)
Clarissa Marquezan (Huawei European Research Center)
Claudio Schepke (Unipampa)
Cristian Machado (UFRGS)
Cristiano Bonato Both (UFCSPA)
Cristiano Costa (UNISINOS)
Cristina Nunes (PUCRS)
Daniela Brauner (UFPEL)
Diego Kreutz (Faculty of Science of University of Lisbon)
Eder John Scheid (UFRGS)
Eduardo Back (IENH)
Eduardo Germano da Silva (UFRGS)
Eduardo Monks (FATEC SENAC Pelotas)
Erico Amaral (Unipampa)
Ewerton Salvador (UFMG)
Flávio Santos (Chaordic Systems)
Giovani Rinaldi (UFRGS)
Glederson Santos (IFSul)
Guilherme Cassales (UFSM)
Guilherme da Cunha Rodrigues (IFSul)
Guilherme Sperb Machado (University of Zürich (UZH), Switzerland)
Gustavo Miotto (UFRGS)
Jéssica Lasch de Moura (UFSM)
José Jair Santanna (University of Twente, The Netherlands)
Juliano Wickboldt (UFRGS)
Leonardo Bays (UFRGS)
Lisandro Zambenedetti Granville (UFRGS)
Lucas Bondan (UFRGS)
Luciano Ignaczak (UNISINOS)
Lucio Prade (UNISINOS)
Maicon Kist UFRGS (UFRGS)
Marcelo Caggiani Luizelli (UFRGS)
Marcelo Marotta (UFRGS)
Marcia Pasin (UFSM)
Marco Trentin (UPF)
Marinho Barcellos (UFRGS)

Matias Schimuneck (UFRGS)
Mauricio Pilla (UFPEL)
Miguel Neves (UFRGS)
Muriel Franco (UFRGS)
Oscar Caicedo (UFRGS)
Pedro Arthur Duarte (UFMG)
Pedro Heleno Isolani (UFRGS)
Rafael Avila (IF Sul)
Rafael Esteves (UFRGS)
Rafael Kunst (UFRGS)
Raul Ceretta Nunes (UFSM)
Ricardo Becker (UCS)
Ricardo Neisse (IPSC - Joint Research Center (JRC))
Ricardo Pfitscher (UFRGS)
Ricardo Schmidt (University of Twente, The Netherlands)
Ricardo Luis dos Santos (UFRGS)
Roben Lunardi (IFRS)
Rodrigo Calheiros (The University of Melbourne, Australia)
Rodrigo Righi (UNISINOS)
Rodrigo Ruas Oliveira (UFRGS)
Rogério Turchetti (UFSM)
Taisy Weber (UFRGS)
Thiago Oliveira (UFSJ)
Vinícius Guimarães (IF Sul - Charqueadas)
Vinicius Ribeiro (Ritter dos Reis)
Vinicius Silva (UFMG)
Walter Priesnitz Filho (UFSM)
Weverton Cordeiro (UFRGS)

Apresentação

É com grande satisfação que apresentamos a 13^a Escola Regional de Redes de Computadores (ERRC 2015). Neste ano o evento é organizado em esforço conjunto entre a Faculdade Meridional (IMED) e a Universidade Federal do Rio Grande do Sul (UFRGS), entre os dias 9 e 11 de setembro, nas dependências da IMED em Passo Fundo. A ERRC é um evento tradicionalmente promovido pela Sociedade Brasileira de Computação (SBC), tendo como objetivo reunir pesquisadores, estudantes e membros da indústria, ligados à área de redes de computadores e afins no Rio Grande do Sul.

Nesta edição, o evento conta com palestras e minicursos abordando temas atuais e relevantes da área de redes de computadores. Um dos principais pontos da escola são as contribuições da comunidade acadêmica com submissões de artigos, que neste ano contou com duas trilhas de submissão: Artigos Completos e Resumos Estendidos. Todos os trabalhos foram inicialmente revisados em um processo onde pelo menos três revisores avaliaram cada artigo, a fim de garantir a qualidade e, ao mesmo tempo, apresentar aos autores sugestões relevantes para seus trabalhos. Desta forma, agradecemos ao Comitê de Programa e demais revisores pelo excelente trabalho na seleção dos textos que compõem este livro.

Em relação ao programa técnico, um total de 24 Artigos Completos e 7 Resumos Estendidos foram aceitos para publicação e apresentação, de um total de 54 trabalhos submetidos. Os Artigos Completos são apresentados por seus autores de forma oral e estão distribuídos em 6 sessões técnicas de acordo com sua temática. Os artigos aceitos tratam dos assuntos mais relevantes da pesquisa em redes da atualidade como, por exemplo, Redes Definidas por Software, Computação em Nuvem e Internet das Coisas. Para os Resumos Estendidos, os autores preparam pôsteres para serem apresentados em duas sessões de curta duração, com abertura para eventuais questionamentos dos participantes.

Para tornar a programação ainda mais interessante, convidamos um conjunto de palestrantes, da indústria e da academia, para discutir o estado-da-arte em redes de computadores e sistemas distribuídos. As palestras abordam tópicos bastante atuais e relevantes para área. Um exemplo é Virtualização de Funções de Rede, uma ideia que está gradualmente deixando de ser apenas foco de pesquisa e tornando-se uma potencial tecnologia em redes corporativas. Outro tema que será abordado é Computação em Nuvem, com um foco especial nos principais serviços existentes e suas principais características. As outras palestras abordam tópicos como gerenciamento de tecnologias emergentes em redes com base em políticas, monitoração da qualidade de experiência de usuários de redes sem fio, operações ágeis em tecnologia da informação, e padronização das tecnologias de redes de computadores e da Internet.

Por fim, temos três minicursos que serão oferecidos para os participantes da ERRC 2015. O primeiro trata sobre o Akka, um middleware e framework para desenvolvimento de aplicações baseado no modelo de atores, o qual tem sido usado com sucesso no desen-

volvimento de aplicações tolerantes a falhas. O segundo minicurso apresentará uma visão geral do Mininet, uma ferramenta que permite criar uma rede virtual inteira, em um computador de configurações modestas, eliminando a necessidade de desenvolver caríssimos e complexos ambientes de testes para a avaliação de novas soluções para redes. O terceiro minicurso abordará o IPv6, o protocolo de rede sucessor do IPv4, que representa uma peça fundamental para permitir o crescimento e o alcance cada vez maior da rede mundial de computadores.

Em nome de toda a equipe de organização da 13^a ERRC, damos as boas vindas à Passo Fundo, e esperamos que o programa técnico, palestras e minicursos oferecidos sejam de grande proveito para todos os participantes!

Luis Augusto Dias Knob
Jéferson Campos Nobre
Juliano Araujo Wickboldt
Cristiano Bonato Both
Weverton Luis da Costa Cordeiro

Passo Fundo, setembro de 2015.

Sumário

I Sessão 1 - Redes Definidas por Software	1
Avaliação de desempenho em Switches OpenFlow	
Gustavo de Araújo, Cristiano Bonato Both	3
Balanceamento de carga baseado em regras OpenFlow: análise de impacto durante a troca de regras	
Clébio Dossa, Rodrigo da Rosa Righi, Vinícius Meyer	11
Uma abordagem prática da tecnologia SDN / OpenFlow em redes voltadas para tráfego de voz e dados	
Carlos Alfredo Weisseimer Junior, Diego Menine	19
Towards Green SDN: An Approach Based on Graph Connectivity	
Eder J. Scheid, Muriel F. Franco, Matias A. K. Schimuneck, Cristiano B. Both, Juergen Rochol and Lisandro Z. Granville	27
II Sessão 2 - Computação em Nuvem	35
Um Modelo de Consumo de Energia para Ambientes de Nuvem com Elasticidade	
Gustavo Rostirolla, Vinicius Facco Rodrigues, Rodrigo da Rosa Righi	37
Elasticidade Assíncrona: Transferência não Bloqueante de VMs para Viabilizar a Reorganização de Aplicações HPC em Cloud Computing	
Vinicio F. Rodrigues, Gustavo Rostirolla, Rodrigo da R. Righi	45
Analizando a Camada de Gerenciamento das Ferramentas CloudStack e OpenStack para Nuvens Privadas	
Demetrius Roveda, Adriano Vogel, Carlos A. F. Maron, Dalvan Griebler, Claudio Schepke	53
Implementação de um sistema de emissão automatizada de certificados de atributos para autorização de acesso em ambientes de cloud computing	
Cassiano Rodolfo Jung, Luciano Ignaczak	61
III Sessão 3 - Segurança de Aplicações	69
Um sistema de pagamento eletrônico com garantia de privacidade baseado no algoritmo criptográfico RSA	
Gustavo Gattino, Marcelo Danesi, Luciano Ignaczak	71
Uma análise dos certificados digitais usados na assinatura de aplicação Android	
Jonata Fröhlich, Claudia Angelita Fagundes Raupp, Luciano Ignaczak	79

Uma análise dos certificados digitais utilizados nas conexões TLS dos aplicativos de Mobile Banking na plataforma Android	87
Diego Baierle Sebastiany, Mirelle Daíara Vieira Freitas, Luciano Ignaczak	87
URLBlackList Lite: Uma lista enxuta de catalogação baseada na URLBlackList	95
Nilson Mori Lazarin, Tielle da Silva Alexandre	95
IV Sessão 4 - Segurança de Redes	103
NS²A: consciência de situação aplicada a segurança de redes de computadores	
Ricardo Borges Almeida, Roger da Silva Machado, Diógenes Y. L. da Rosa, Henrique de Vasconcellos Rippel, Lucas Medeiros Donato, Adenauer Corrêa Yamin, Ana Marilza Pernas	105
DLNA-ML: Uma Abordagem de Análise Dinâmica de Log e Tráfego da Rede	
Roger da Silva Machado, Ricardo Borges Almeida, Diógenes Y. L. da Rosa, Henrique de Vasconcellos Rippel, Adenauer Corrêa Yamin, Ana Marilza Pernas	113
Mecanismos de Segurança aplicados a Interface para o Sistema de Roteamento (I2RS)	
Joel Molling, Jéferson Campos Nobre	121
Um Estudo para Identificação e Mitigação de IP Spoofing em Redes IPv6 utilizando SDN	
Manoel F. Ramos, Rafael B. Avila	129
V Sessão 5 - Computação Ubíqua e Internet das Coisas	137
Avaliação de Simuladores para Redes Veiculares Ad Hoc para Implementação de Aplicações P2P de Gerenciamento	
Lucas De Marchi Dreger, Jéferson Campos Nobre	139
Uma abordagem híbrida para armazenamento de dados de contexto no EXEHDA	
Diógenes Yuri Leal da Rosa, Ivan José Rambo, Roger da Silva Machado, Ricardo Borges Almeida, Henrique de Vasconcellos Rippel, Adenauer Corrêa Yamin, Ana Marilza Pernas	147
Controle Remoto de Dispositivos de Baixo Custo Utilizando Agentes SNMP	
Willian Tadeu Poloni, Ricardo Becker, Ricardo Balbinot	155
EXEHDA-IoT: Uma Abordagem Consciente de Contexto Direcionada à Internet das Coisas	
Patrícia Davet, Huberto Kaiser Filho, Leonardo João, Lucas Xavier, Tainá Carvalho, Rodrigo Souza, João Lopes, Adenauer Yamin	163
VI Sessão 6 - Modelagem e Análise de Desempenho	171
Análise Comparativa entre HTTP 1.1 e 2.0	
Simei Tabordes Goncalves, Eduardo Maroñas Monks	173

Análise do comportamento de servidores web sob ataques causados por Bot-Nets	
Leandro Ferreira Canhada, Eduardo Maroñas Monks	181
Proposição de um Método de Análise de Qualidade de Vídeo sem Referência Completa	
Alessandro Marchetto, Ricardo Becker, Ricardo Balbinot	189
Problemas e Soluções no Desenvolvimento de Componentes para o NS3: o Caso do DHCP	
Andrey Blazejuk, Alexsander Silva de Souza, Sérgio Luis Cechin	197
VII Sessão de Pôsteres	203
Uma Proposta Ubiservice para Tratamentos de Serviços Direcionados a UbiPri	
Adrian R. Lemes Caetano, Maycon Viana Bordin, Wagner Kolberg, Gustavo B.Brand, Guilherme Dal Bianco, Valderi R. Q. Leithardt	205
BlueTApp - Um Aplicativo Móvel para Registro Automático da Presença Acadêmica via Bluetooth	
Fernando Weber Albiero, Fábio Weber Albiero, João Carlos Damasceno de Lima, Iara Augustin	209
Redes Definidas por Software: Monitoramento Sensível ao Contexto	
Lucas Powaczuk, Leonardo da C. Marcuzzo, Luiz E. G. da Silva, Vania Freitas, Tassiana Kautzmann, Roseclea D. Medina	213
Da Elaboração a Implantação da Política de Segurança da Informação: uma proposta baseada no ciclo PDCA	
Peter Prevedello, Diogo Otto Kunde	217
Processamento Dinâmico de Regras Semânticas para Identificação de Situações	
Lidiane Costa da Silva, João Ladislau Lopes, Ana Marilza Pernas	221
Laboratório Remoto de Rede de Computadores	
Alexander M. Diaczenko, Vitor S. Brixius, Leandro J. Cassol, Luís C. M. Cáruso, Taciano A. Rodolfo, Vanderson da Silva	225
O problema da padronização de interfaces norte no paradigma SDN	
Kazuki Yokoyama, Alexsander de Souza, Sérgio Luis Cechin	229



Sessão 1 - Redes Definidas por Software

Avaliação de desempenho em *Switches OpenFlow*

Gustavo de Araújo¹, Cristiano Bonato Both²

¹Departamento de Informática – Curso de Engenharia de Computação
Universidade de Santa Cruz do Sul (UNISC)
Santa Cruz do Sul – RS – Brazil

²Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

inf.araujo@gmail.com, cbbboth@inf.ufrgs.br

Abstract. *Software-Defined Networking (SDN) is a paradigm of computer networks that allows flexibility and agility in developing new technologies. This is achieved by separating the control of network devices that route the user packages. This paradigm has gained prominence with the specification of the OpenFlow protocol, which allows a communication interface between the control plane and the data plane. Despite the rapid spread of this protocol, the literature does not go deep performance analysis of data routers devices (switches). This article proposes a methodology for evaluating performance OpenFlow switches focused on analyzing the data plane hardware. As a result it is obtained a comparison of four OpenFlow switches, and two virtual-two reais.*

Resumo. *Software-Defined Networking (SDN) é um paradigma de redes de computadores que permite a flexibilidade e agilidade no desenvolvimento de novas tecnologias. Isto é alcançado, separando o controle da rede dos dispositivos que encaminham os pacotes do usuário. Este paradigma ganhou destaque com a especificação do protocolo OpenFlow, que permite uma interface de comunicação entre o plano de controle e o plano de dados. Apesar da rápida disseminação deste protocolo, a literatura não aprofunda na análise de desempenho dos dispositivos encaminhadores de dados (switches). Este artigo, propõem uma metodologia para a avaliação de desempenho em switches Openflow focada em analisar o hardware do dispositivo de encaminhamento. Como resultados é obtido uma comparação entre quatro switches OpenFlow, sendo, dois virtuais e dois reais.*

1. Introdução

Software-Defined Networking é um paradigma emergente em redes de computadores que propõe uma plataforma flexível para o desenvolvimento de novas soluções para infraestruturas de redes. Uma de suas principais características está na separação da lógica de controle da rede (controlador) dos equipamentos que encaminham os pacotes através de um plano de dados (*switches*). Com essa separação, os *switches* se tornam apenas equipamentos de encaminhamento de pacotes [Kreutz et al. 2014]. Para que essa separação ocorra, é necessário um protocolo de comunicação entre o plano de controle e o plano de dados. Com este propósito, o protocolo Openflow foi desenvolvido, sendo

atualmente, a mais relevante implementação do paradigma SDN [McKeown et al. 2008]. Um *switch* Openflow possui uma ou mais tabelas com regras para manipulação dos pacotes de dados (tabela de fluxos). Cada regra corresponde a um subconjunto do tráfego e uma sequência de ações de como lidar com os pacotes, como por exemplo, descartar, encaminhar ou modificar cabeçalho [ONF 2013].

Com esse novo paradigma, as métricas tradicionais de avaliação de desempenho como latência, *jitter*, perda de pacotes e largura de banda se tornam insuficientes para analisar o desempenho de equipamentos de redes baseadas em SDN. O desempenho de um *switch* OpenFlow não está ligado somente à vazão do tráfego de dados, mas também em como é feito o processamento do tráfego de controle. Por exemplo, é importante avaliar a latência da comunicação entre *switch* e controlador. Pois, a latência deste canal impacta diretamente no tráfego de dados, uma vez que, o *switch* não toma decisões autônomas, sendo de responsabilidade do controlador informar o que deve ser feito com determinado fluxo. Deste modo, uma metodologia de avaliação de desempenho deve ser elaborada para esse novo paradigma de rede.

Atualmente, a literatura sobre SDN apresenta poucas soluções para a avaliação de desempenho. Cbench [Tootoonchian et al. 2012] avalia a latência de atualização da tabela de fluxos o atraso de pacotes que necessitam de modificação do cabeçalho para serem encaminhados. OFCBenchmark [Jarschel et al. 2012] também realiza uma análise de controladores, verificando métricas como os pacotes por segundos enviados do controlador para o *switch*. E Oflops [Rotsos et al. 2012] permite avaliar algumas características do *switch* como tempo de instalação de uma nova regra na tabela de fluxos, latência do canal de controle e impacto das mensagens de coleta de estatística no tráfego de controle e tráfego de dados.

O restante deste trabalho é organizado da seguinte forma: na seção 2 são apresentadas as bases para a realização deste trabalho. Na Seção 3 é apresentado a metodologia para a validação de desempenho. No Seção 4, são exibidos os resultados obtidos com a implementação do protótipo. E finalmente, na Seção 5 são exibidas as considerações finais e trabalhos futuros.

2. Fundamentação Teórica

Esta Seção aborda os conceitos básicos sobre SDN e Openflow, assim como, alguns dos trabalhos relacionados a avaliação de desempenho neste paradigma. Na Seção 2.1 são apresentados os principais conceitos do paradigma SDN e do protocolo Openflow. E na Seção 2.2 são apresentados os trabalhos relacionados a avaliação de desempenho em SDN.

2.1. SDN/Openflow

Em uma rede IP tradicional, os planos de dados e controle são acoplados e embarcados no mesmo equipamento de rede, com controle sendo descentralizado, fazendo com que os equipamentos tomem decisões sobre o encaminhamento de pacotes de maneira autônoma. Este tipo de abordagem torna a arquitetura de uma rede muito complexa. Como consequência, tem-se uma rede muito estática que dificulta o desenvolvimento de novos serviços. Visando permitir que soluções de rede sejam facilmente projetadas e desenvolvidas o paradigma SDN foi baseado nos seguintes pilares:

- Desacoplar plano de dados do plano de controle. Desta maneira, os equipamentos de rede se tornam apenas encaminhadores de pacotes, não tomando decisões autônomas.
- Lógica de controle é movida para uma entidade externa, centralizada, implementada em software. Esta entidade chama-se controlador e tem a funcionalidade de implementar toda a lógica de controle da rede, isto faz com que, o desenvolvimento de novos serviços de rede sejam facilmente implementáveis.
- O encaminhamento de dados torna-se baseado em fluxos ao invés de ser baseado em destino. Um fluxo pode ser definido como um conjunto de campos de cabeçalhos utilizados para identificação (*match*) e um conjunto de ações (*action*) [Kreutz et al. 2014].

Na arquitetura SDN, o plano de dados é representado pelos equipamentos de rede como *switches* e roteadores, porém, nenhuma decisão de controle é tomada de maneira embarcada nestes equipamentos. Sendo assim, o plano de dados apenas implementa as regras definidas pelo plano de controle [ONF 2013]. O plano de controle, no paradigma SDN, é representado pelo controlador que é responsável pela inteligência da rede. Através de uma interface de comunicação, o plano de controle “programa” os equipamentos do plano de dados, de acordo com a necessidade da rede. Desta maneira os controladores são responsáveis por decidir como serão encaminhados os pacotes do plano de dados.

O paradigma SDN foi amplamente difundido, após a especificação do protocolo OpenFlow. Este protocolo define uma interface entre o plano de dados e o plano de controle. O protocolo OpenFlow é implementado em ambos os planos, de forma que, o equipamento do plano de dados possui uma tabela que armazena regras de encaminhamento de dados, chamada tabela de fluxos. Este protocolo também prevê um canal seguro para comunicação entre *switch* e controlador, como por exemplo *Security Socket Layer*, assim como um conjunto de mensagens para manipulação da tabela de fluxos.

2.2. Trabalhos Relacionados

As pesquisas referentes a análise de desempenho em SDN, em sua maioria, focam-se em analisar controladores. Por exemplo, na metodologia apresentada pelo *benchmark* Cbench [Tootoonchian et al. 2012], avalia a latência de processamento do controlador para requisições do tipo *Packet-in*, bem como quantidade máxima de mensagens *Packet-in* que o controlador consegue suportar por segundo.

O *framework* Oflops possui uma grande quantidade de ferramentas para avaliação de desempenho em *switches* OpenFlow [Rotsos et al. 2012]. Com esta ferramenta é possível verificar a capacidade dos *switches* de encaminhar os pacotes de dados, o tráfego do canal de controle, latência para instalação de um novo fluxo, além de monitorar o tráfego. Essa ferramenta também possibilita a geração de tráfego de dados e tráfego de controle. Oflops possui uma arquitetura modular, permitindo que novos testes sejam adicionados ao *framework* de maneira rápida [Rotsos et al. 2012].

É possível perceber que apenas a metodologia Oflops fornece testes para a avaliação de desempenho de *switches* OpenFlow, enquanto, Cbench e OFBenchmark focam-se em testes para controladores. E mesmo assim, o *framework* Oflops oferece poucas ferramentas para avaliação do hardware dos *switches*.

3. Metodologia de Avaliação

Esta Seção descreve a metodologia utilizada para avaliação de desempenho, os tipos de testes realizados e o cenário onde os testes foram feitos. Na Seção 3.1 são descritos detalhes da implementação dos testes. Na Seção 3.2 são apresentados os tipos de testes utilizados. E na Seção 3.3 é apresentado o cenário de teste utilizado para obter os resultados.

3.1. Implementação

Para alcançar esses objetivos foi escolhido o controlador Floodlight [Network 2014]. É um controlador muito difundido dentro da comunidade SDN possuindo uma boa documentação sendo de fácil utilização. Além disso, este controlador possui uma API JSON permitindo uma fácil interação de maneira remota. Neste trabalho, esta API foi utilizada para a passagem de parâmetros, inicialização e coleta de resultados.

Para gerar novos fluxos é utilizado um algoritmo que incrementa o MAC destino e o MAC origem. Este mecanismo é utilizado, pois, o endereço MAC possui 48 bits de endereçamento, isto permite uma grande combinação de fluxos diferentes sejam instalados como por exemplo, um fluxo com MAC origem: 00:00:11:11:22:22 e vlan: 100. A *action* utilizada é o direcionamento do tráfego para uma determinada porta.

3.2. Tipos de Teste

Foram definidos seis tipos de *match* para a realização dos experimentos, Podem ser visualizados na Tabela 1.

Tabela 1. Tipos de Match utilizados para teste.

Tipo de Match	Campos de Match
1	MAC Destino
2	MAC Destino e MAC Origem
3	MAC Destino, MAC Origem, VLAN e VLAN Priority
4	L2, IP Origem e IP Destino (sem Type)
5	L2, IP Origem, IP Destino e Type
6	Todos os campos

3.3. Cenário de Testes

Para coleta e análise dos resultados, foram escolhidos quatro *switches*, onde os testes foram efetuados. Estes *switches* foram divididos em dois grupos: (*i*) *switches* virtuais e (*ii*) *switches* reais. O primeiro grupo, diz respeito a uma implementação em software que emula um *switch* real. Os *switches* utilizados para este grupo foram *OpenvSwitch* [OpenvSwitch 2015] e a implementação do *switch* de Stanford [Naous et al. 2008] utilizando NetFPGA [NetFPGA 2015]. A Tabela 2 apresenta a comparação entre os *switches* virtuais utilizados, assim como, algumas informações das máquinas hospedeiras de cada um deles.

Características	OpenvSwitch	Stanford
Capacidade da Tabela de Fluxos	100KB	32KB
CPU	Core Intel i5-3337U	Intel Core 2 Duo
Memória	6GB	3GB
OS	Ubuntu 12.04	CentOs 4
Interfaces	4 x 10/100BASE-T RJ45 (todas Virtuais)	4 x 1000BASE-T RJ45 (NetFPGA)
Capacidade de emcaminhamento	Não especificado	1000M

Tabela 2. Comparativo entre switches virtuais

O segundo grupo, diz respeito a *switches* reais que são comercializados. A Tabela 3 mostra um comparativo entre os *switches* escolhidos, as informações foram retiradas da especificação disponibilizada pelo fabricante de cada *switch*. Para o *Switch A* foi utilizado o *firmware* Indigo [Indigo 2015] e para o *Switch B* foi utilizado o *firmware* original de fábrica.

Feature	Switch A	Switch B
Capacidade da Tabela de Fluxos	Não especificado pelo fabricante	2.5KB
CPU	P2020	Cavium CN5010
Memória	2GB	256MB
OS	Debian	Indigo
Interfaces	48 x 10/100/1000BASE-T RJ45 4 x 1 GbE (SFP)	48 x 100/1000BASE-T RJ45 4x 1 GbE (SFP)
Capacidade de emcaminhamento	132GB	176GB

Tabela 3. Comparativo entre switches reais

4. Resultados

Esta Seção apresenta os resultados obtidos com os testes implementados. Na Seção 4.1 apresenta os resultados para a capacidade máxima da tabela de fluxos. Na Seção 4.2 apresenta os resultados para a utilização da memória. E finalmente, na Seção 4.3 são apresentados os resultados para a utilização de CPU.

4.1. Capacidade da Tabela de Fluxos

Com os testes realizados no *Switch A*, pode-se notar que a quantidade de campos utilizados para *match* influencia na capacidade total de armazenamento de regras da tabela de fluxos. Na Figura 1 são exibidos os resultados obtidos. Com a análise do gráfico percebe-se que existe uma relação entre a quantidade de fluxos que podem ser instalados e a quantidade de campos utilizados para *matches*. Quanto maior a quantidade de campos necessários para realizar um *match*, menor a quantidade máxima de fluxos que podem ser instalados. Percebe-se também que para *matches* utilizando cabeçalho L2 e IP origem e

destino (Tipo 4) há uma grande queda na quantidade de fluxos que podem ser instalados. Entretanto, ao utilizar o campo *type* do cabeçalho Ethernet a quantidade máxima de fluxos instalados aumenta. Isso acontece devido a algum mecanismo de otimização para o armazenamento de fluxos na tabela de acordo com os campos utilizados para *match*.

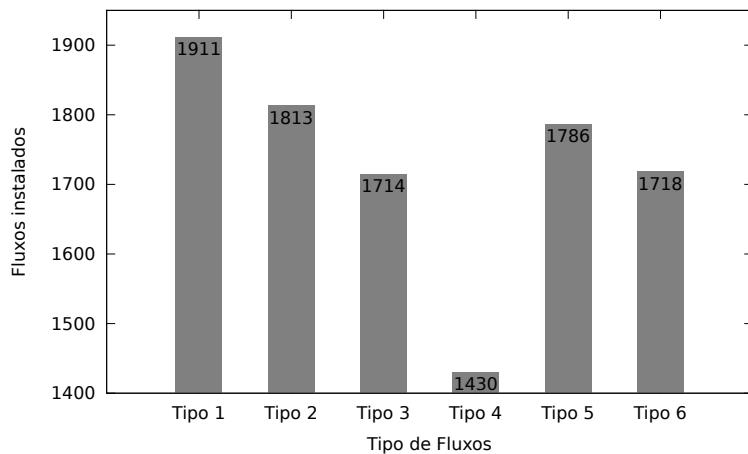


Figura 1. Resultados obtidos para o Switch A.

Os outros *switches* não apresentaram variações de acordo com o tipo de *match* realizado, além de, permitirem uma capacidade de armazenamento de regras igual ou maior do que a anunciada pelo fabricante. A tabela 4 são exibidos os resultados para o *Switch B*, *OpenvSwitch* e *Stanford*.

Switch	Capacidade Máxima da Tabela de Fluxos
Switch B	2560
Stanford	32792
OpenvSwitch	1000000

Tabela 4. Quantidade máxima de fluxos instalados em cada switch.

4.2. Utilização da Memória

Para cada tipo de *match* foi realizada a medição da quantidade de memória utilizada conforme os fluxos fossem instalados. A Figura 2 mostra a comparação da utilização da memória entre os *Switch A* e o *Switch B*. O *Switch A* variou a quantidade de memória utilizada de acordo com a quantidade de fluxos instalados. O *Switch B* mostrou uma diminuição da utilização de memória de acordo com a quantidade de campos utilizados para o *match*.

Os *Switches* virtuais mostraram um comportamento constante, a quantidade de campos utilizados para o *match* não variou a quantidade de memória utilizada. O *OpenvSwitch* utilizou 500MB de memória RAM para instalação de 100K regras, enquanto, o *Switch* de *Stanford* utilizou 16MB para a instalação de aproximadamente 32K regras.

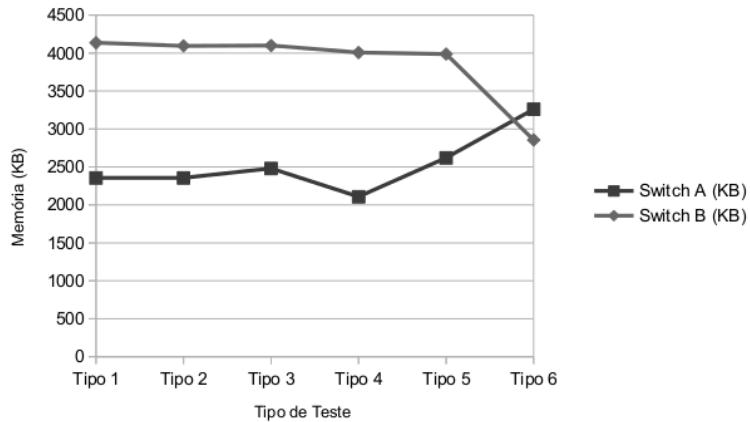


Figura 2. Resultados de utilização de memória entre os switches reais.

4.3. Utilização de CPU

Para realização deste teste foi enviado para cada *switch* a quantidade máxima de fluxos que podem ser instalados em suas tabelas de fluxos. Como resultado, foi obtido a sobrecarga que a instalação seguida de vários fluxos possui em cada *switch*. A Tabela 5 apresenta os resultados obtidos. É possível notar que a instalação de seguida de várias fluxos nos *switches* reais não apresentam um sobrecarga significativa na CPU. Porém, nos *switches* virtuais a utilização da CPU apresenta um acréscimo significativo. Isso ocorre, pois os *switches* virtuais não possuem processadores otimizados para este tipo de funcionalidade. Além de que, os *switches* reais possuem um hardware específico para o processamento de mensagens OpenFlow.

	Switch A	Switch B	OpenvSwitch Memória RAM	Stanford Memória RAM
Sobrecarga de Utilização da CPU	2%	3%	27%	14%

Tabela 5. Sobrecarga da utilização da CPU no momento da instalação de fluxos

5. Conclusão e Trabalhos Futuros

Nesse artigo foi proposto uma metodologia de testes de desempenho para avaliar o hardware de *switches* Openflow. Para a implementação dessa metodologia, foi estudado as métricas que influenciam no desempenho destes dispositivos. Além disso, foi estudado como as atuais metodologias de avaliação de desempenho são implementadas. Após a implementação, a metodologia de testes foi aplicada em quatro *switches* OpenFlow, sendo dois *switches* reais e dois *switches* virtuais. Os resultados obtidos foram apresentados comparando os *switches* utilizados, onde, pode-se verificar qual *switch* possui o melhor desempenho para uma determinada métrica.

Este trabalho, abre a possibilidade para a realização de projetos como otimizações na metodologia desenvolvida, aumentar a quantidade de tipos testes de testes, criação de uma base de dados com os resultados obtidos por cada *switch*. Dessa forma, pode-se aumentar o escopo da metodologia de avaliação, possibilitando a fácil comparação dos resultado entre diferentes *switches*.

Referências

- Indigo (2015). Indigo. Disponível em: <<http://www.projectfloodlight.org/indigo/>>. Acesso em: Set. 2015.
- Jarschel, M., Lehrieder, F., Magyari, Z., and Pries, R. (2012). A flexible openflow-controller benchmark. In *Proceedings of the 2012 European Workshop on Software Defined Networking*, EWSN '12, pages 48–53, Washington, DC, USA. IEEE Computer Society.
- Kreutz, D., Ramos, F. M. V., Veríssimo, P., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. (2014). Software-defined networking: A comprehensive survey. *Computing Research Repository*, abs/1406.0440.
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, pages 69–74.
- Naous, J., Erickson, D., Covington, G. A., Appenzeller, G., and McKeown, N. (2008). Implementing an openflow switch on the netfpga platform. In *Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, ANCS '08, pages 1–9, New York, NY, USA. ACM.
- NetFPGA (2015). Netfpga. Disponível em: <<http://netfpga.org/site/>>. Acesso em: Ago. 2015.
- Network, B. S. (2014). Floodlight. Disponível em: <<https://projectfloodlight.org/floodlight/>>. Acesso em: Set. 2014.
- ONF (2013). Sdn architecture. Disponível em: <<https://www.opennetworking.org/>>. Acesso em: Mar. 2014.
- OpenvSwitch (2015). Openvswitch. Disponível em: <<http://openvswitch.org/>>. Acesso em: Ago. 2015.
- Rotsos, C., Sarrar, N., Uhlig, S., Sherwood, R., and Moore, A. W. (2012). Oflops: An open framework for openflow switch evaluation. In *Proceedings of the 13th International Conference on Passive and Active Measurement*, PAM'12, pages 85–95, Berlin, Heidelberg. Springer-Verlag.
- Tootoonchian, A., Gorbovov, S., Ganjali, Y., Casado, M., and Sherwood, R. (2012). On controller performance in software-defined networks. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services*, Hot-ICE'12, pages 10–10, Berkeley, CA, USA. USENIX Association.

Balanceamento de carga baseado em regras OpenFlow: análise de impacto durante a troca de regras

Clébio Dossa¹, Rodrigo da Rosa Righi¹, Vinicius Meyer¹

¹Programa de Pós-Graduação em Computação Aplicada – Universidade Do Vale do Rio dos Sinos (UNISINOS)

clebiiodossa@gmail.com, rrrighi@unisinos.br, vinimeyer@hotmail.com

Abstract. *Actually, most services balance the load between distinct hosts forwarding connections with a load balance strategy in front. Usually, a dedicated server that may be a fault point and become expensive makes this load balance. The OpenFlow protocol definition allow us to use a new solution to address this issue. This work shows a load balance solutions between distinct hosts with the destination change of connections made by the network core.*

Resumo. *Atualmente muitos serviços distribuem a carga entre diversos hosts direcionando as conexões com alguma estratégia de balanceamento em frente. Este balanceamento geralmente é realizado por algum servidor dedicado que pode se tornar ponto único de falhas e ter um custo caro. A definição do protocolo OpenFlow permite uma solução alternativa e eficiente para este problema. Neste trabalho é apresentado uma solução para balanceamento de carga entre distintos hosts com a troca do destino do tráfego realizada pelo núcleo da rede.*

1. Introdução

Desempenho e alta disponibilidade são itens críticos para *data centers* atuais. A infraestrutura de rede necessita maximizar a vazão, diminuir a latência e permitir uma elasticidade natural para atender as atuais demandas de aplicações. A comunicação entre a origem e o destino percorre múltiplos caminhos passando por distintos serviços e servidores. Cada um destes itens pode ser um ponto de falha, afetar a capacidade, a vazão ou simplesmente aumentar a latência. Desta forma, a simplificação e remoção destes pontos adicionais deve ser considerada na construção de um ambiente robusto e performático. Contudo, processos e equipamentos não podem ser rígidos e engessados.

Para atender a demanda necessária ocorre geralmente a aquisição de novos equipamentos, realização upgrade de hardware ou mesmo aumento da vazão. Neste cenário, mecanismos de balanceamento de carga realizam um importante trabalho distribuindo acessos concorrentes ou adequando aos limites da capacidade. Muitos métodos e fabricantes propõem soluções complexas, difíceis de serem implementadas [1] e testadas, rígidas e caras que necessitam de hardware potente para armazenar milhares de fluxos de comunicação. Além disso, o uso de equipamento dedicado para

balanceamento de carga adiciona latência ao processamento, podem ser um gargalo ou mesmo um ponto adicional de falha. Mesmo com tantas implementações disponíveis, a técnica mais eficaz de balanceamento de carga é um item incógnito para cada solução e depende de variáveis mutáveis para ser atingido em sua excelência.

Delegando a tarefa de balanceamento de carga ao núcleo da rede, onde equipamentos específicos já realizam a administração dos fluxos de conexão, se reduz a necessidade de adição de equipamentos dedicados à tarefa, também diminuindo a latência incrementada por estes equipamentos. Redes programáveis permitem a manutenção de regras de fluxos e fornecem estatísticas para tomada de decisão. A tecnologia do protocolo OpenFlow proporciona o acesso à um modelo que pode ser explorado para realização da tarefa de balanceamento de carga. Neste sentido, o OpenFlow permite a criação de técnicas distintas para o balanceamento de carga com grande flexibilidade, administrando cada fluxo da comunicação e proporcionando a habilidade de controlar cada host a qualquer momento de maneira centralizada[6].

Este trabalho realiza uma pesquisa no impacto do uso do protocolo OpenFlow para realização de balanceamento de carga. Esta é uma etapa de uma pesquisa mais ampla que tende à criação de uma solução que, com o uso do OpenFlow em conjunto do protocolo SNMP, proporciona a execução de balanceamento de carga adaptável, equilibrando a carga de uma forma eficaz. Com a visão de distintos pontos: fluxos de comunicação de todos os hosts do平衡amento através do OpenFlow e da capacidade de recursos de cada host através do SNMP, é possível ter uma visão estratégica da situação do ambiente. Existe uma grande diversidade de algoritmos de balanceamento de carga, entretanto, a análise de decisão com informações baseadas na rede não é comum de ser utilizada para este propósito [8]. Com isso, este trabalho e uma pesquisa necessária para o desenvolvimento de uma solução que, posteriormente, irá unir informações de tráfego de rede com consumo de recursos dos servidores para tomada de decisão no balanceamento de carga.

A seguir uma breve introdução ao OpenFlow e a apresentação de trabalhos relacionados a esta pesquisa onde o balanceamento de carga com o uso do OpenFlow já foi explorado. Seguindo esta estrutura, a solução proposta está dividida na idealização do processo, a implementação, a metodologia de avaliação e os resultados desta etapa inicial, que tem como objetivo mapear o impacto da troca de tráfego realizada pelo OpenFlow.

2. Tecnologia OpenFlow

A arquitetura OpenFlow pode ser dividida em alguns switches OpenFlow, um ou mais controladores, um canal dedicado e seguro de comunicação entre os switches e controladores e o protocolo OpenFlow para sinalizar e administrar os switches. O objetivo deste trabalho não é detalhar o funcionamento do OpenFlow, detalhes podem ser encontrados em [2].

Clássicos roteadores ou switches executam a tarefa de *data plane* e *control plane*. O plano de dados é onde as transmissões dos pacotes são realizadas e o plano de controle é onde as decisões de como este pacote será administrado são realizadas. O OpenFlow

separa estes dois planos e um protocolo define as trocas de mensagens entre eles criando um padrão de como o plano de dados solicita informações ao plano de controle e como o plano de controle informa as regras. O plano de controle define as decisões adicionando as regras de transmissão de pacotes no plano de dados, conforme figura 1.

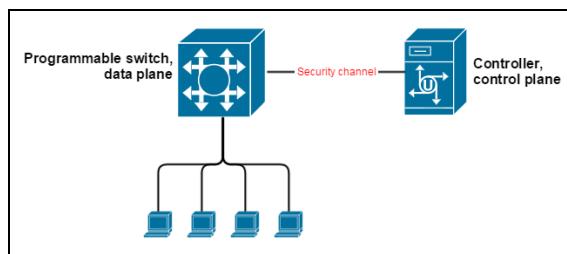


Figura 1. Topologia OpenFlow

3. Trabalhos relacionados

O uso do OpenFlow tem sido explorado em diversos aspectos. Muitos modelos e técnicas clássicas de administração e implementação de redes estão agora sendo implementadas com o uso do OpenFlow. O uso de balanceamento de carga diretamente no núcleo da rede não é uma atividade comum no modelo clássico de redes, entretanto o advento da tecnologia OpenFlow possibilita a absorção desta tarefa.

O trabalho desenvolvido em [5] demonstra a técnica de balanceamento de carga com OpenFlow onde o processo de decisão é definido por um algoritmo incorporado no plano de controle. Usando a estratégia de *Server-based load balancing* (SBLB) o controlador define o host com melhor capacidade para receber a requisição e o fluxo é alterado dinamicamente com troca de regras no plano de dados do switch OpenFlow. As avaliações neste trabalho demonstram uma grande flexibilidade e baixo custo de implementação e o uso desta técnica de balanceamento de carga aumenta o tempo de resposta de *Web Servers* e permite uma distribuição mais racional de recursos.

Em [6] uma avaliação é realizada usando diferentes técnicas de balanceamento de carga implementadas com o uso de OpenFlow: *Random choice*, *Time slice based choice* e *Weighted balancing*. A avaliação é baseada na melhor distribuição da carga em cada host usando ICMP requests. A técnica mais eficiente foi *Weighted balancing*, no entanto, a complexidade da implementação desta técnica é linear ao número de hosts que atendem o serviço pois, para cada fluxo recebido é realizada uma análise de performance em cada um dos hosts.

O uso do Openload, um software para gerar carga web e realizar medições de tempo de resposta e transações por segundo, foi tema da pesquisa em [7] onde a estratégia *Round robin* foi comparada com *Random choice* implementados em OpenFlow. Esta avaliação apontou *Round robin* com melhor desempenho, no entanto salientou a necessidade de tests em hardware real e não apenas testes em ambientes simulados.

4. Solução proposta

As técnicas clássicas para balanceamento de carga podem ser reproduzidas da mesma forma com o modelo proposto pelo protocolo OpenFlow inclusive, com melhor performance de acordo com a pesquisa demonstradas em [4] que indicam um melhor desempenho no modelo OpenFlow de switch comparado com modelos convencionais, provando ser uma melhor alternativa para software *Ethernet Switching* ou mesmo IP *Routing*.

Entretanto, a melhor solução para o problema, como implementar e o que controlar para atender o balanceamento são itens distintos. Os fluxos de conexões previamente estabelecidas não podem ser direcionados para distintos hosts, da mesma forma, a proporção de processamento por parte dos hosts precisa ser equilibrada sem sobrecarregar determinado host enquanto outros estão com boa capacidade de processamento.

O protocolo OpenFlow viabiliza a adição de regras no plano de dados que é alimentado com informações do destino que os fluxos entrantes devem tomar. Estas regras podem ser alteradas de acordo com a necessidade de processamento, vasão, latência ou mesmo o histórico dos fluxos. Basicamente, regras predefinidas de fluxos podem ser utilizadas para balanceamento de carga. Com o uso das distintas técnicas de balanceamento, este processo de alteração de fluxo pode ser dinâmico e automatizado através de algoritmos específicos.

A solução apresentada nesta pesquisa atende balanceamento de conexões para aplicações que não necessitam o uso de contexto (stateless). Tem-se em consideração que todo o host de destino tem a mesma capacidade de atender a solicitação entrante pois realiza o mesmo serviço. O processo abaixo visa analisar o impacto das trocas de regras em intervalos de tempo com o intuito de clarificar o impacto desta ação. Esta análise precisa ser considerada para desenvolvimento de um processo eficiente de

4.1 O Processo Idealizado

O balanceamento no contexto do OpenFlow é realizado alterando o destino do pacote para direcionar a conexão entrante para um host com o maior poder de processamento no momento da solicitação. Utilizando as técnicas de *layer 3* do OpenFlow, o pacote destinado ao serviço tem seu endereço IP de destino reescrito para o endereço de IP do host com maior capacidade de processamento e com menor quantidade de pacotes já recebidos. No momento em que o host retorna a solicitação, o IP de origem é alterado para o IP que foi inicialmente solicitado para não danificar a comunicação. Sendo duas alterações realizadas no pacote recebido: no momento em que ele entra e no momento em que sai.

A análise de desempenho realizada por [3] comprova a eficiência e vasão das operações em *layer 3* do OpenFlow superando entre 30% e 40% a performance de *switching layer 2* em implementações baseadas em Linux. Nestes testes também se observa uma ótima performance do OpenFlow para lidar com múltiplos fluxos e capacidade de gerencia-los combinado com a excelente vasão e baixa latência, itens que tem encorajado muito o uso desta tecnologia.

4.2 Desenvolvimento e Implementação

O experimento do funcionamento desta metodologia foi realizado com o uso de Openvswitch, que é um comutador virtual que suporta diversas camadas e um impressionante conjunto de recursos. Na figura 2 um exemplo de regra genérica que este comutador pode receber para alterar o destino de um pacote.

```
ovs-ofctl add-flow <comutador> \
ip, tcp_flags=+syn, nw_dst=<IP origem>, \
actions=mod_nw_dst=<IP destino>, mod_dl_dst:<MAC destino>, output:<porta destino>
```

Figura 2. Regra para alteração do destino

O controle da *flag* de entrada é importante para não modificar o destino de conexões já estabelecidas. Opções adicionais podem ser utilizadas para controle de portas, protocolos e demais itens definidos pelo protocolo OpenFlow.

Da mesma forma que o destino foi alterado, uma regra de retorno se faz necessário para que o cliente que solicitou a conexão receba a informação de forma correta. A figura 3 contempla uma regra genérica para este objetivo.

```
ovs-ofctl add-flow <comutador> \
ip, nw_src=<IP destino>, actions=mod_nw_src=<IP origem>, output:<porta origem>
```

Figura 3. Regra para alterar o pacote de resposta

A regra para alteração do pacote de resposta pode ser sempre estática, existindo uma para cada host onde o serviço é atendido.

Neste exemplo a conexão é sempre originada para o mesmo endereço IP. O cliente utiliza um ponto único de acesso e o comutador decide para quem será enviado a conexão.

4.3 Metodologia de Avaliação e Análise dos Resultados

Para realizar esta análise e provar o funcionamento desta técnica, uma rede virtual foi inicializada com o uso do emulador Mininet. Neste ambiente 5 switches e 16 hosts virtuais foram inicializados. A topologia do ambiente é apresentada na Figura 4.

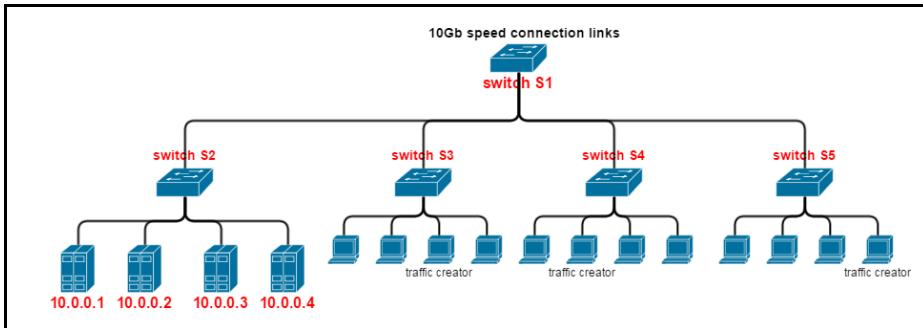


Figura 4. Topologia da rede criada

O switch S1 é responsável pelas conexões entre os outros 4. Os hosts existentes no switch S2 irão receber as conexões e o tráfego será alternado entre todos os hosts em intervalos variados de 100 segundos. O script exemplificado na figura 5 é responsável pela alteração de tráfego entre os equipamentos. A figura 6 demonstra o fluxograma do script desenvolvido para esta análise.

```

1 #clean all rules
2 ovs-ofctl del-flows s2
3 #hosts that belongs to the load balance group
4 ovs-ofctl --strict add-flow s2 ip,priority=65535,nw_src=10.0.0.2,actions=mod_nw_src=10.0.0.1,output:5
5 ovs-ofctl --strict add-flow s2 ip,priority=65535,nw_src=10.0.0.3,actions=mod_nw_src=10.0.0.1,output:5
6 ovs-ofctl --strict add-flow s2 ip,priority=65535,nw_src=10.0.0.4,actions=mod_nw_src=10.0.0.1,output:5
7
8 while true
9 do
10     #random choice to the destination host
11     HOST=$((RANDOM%4+1))
12     echo "Next destination $HOST \n";
13     #delete old rule
14     ovs-ofctl --strict del-flows s2 \
15         ip,priority=65535,tcp_flags=+syn,nw_dst=10.0.0.1
16     #add new rule
17     ovs-ofctl --strict add-flow s2 \
18         ip,priority=65535,tcp_flags=+syn,nw_dst=10.0.0.1, \
19         actions=mod_nw_dst=10.0.0.$HOST,mod_dl_dst:00:00:00:00:00:0$HOST,output:$HOST
20
21     #wait a random time between 0 and 10 seconds
22     WAIT=$((RANDOM%100))
23     echo "Waiting $WAIT \n\n"
24     sleep $WAIT
25 done

```

Figura 5. Script para automação da troca de tráfego

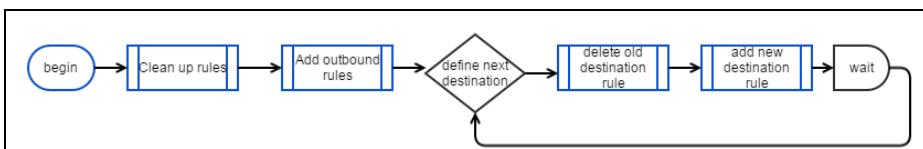


Figura 6. Fluxograma do script desenvolvido

O objetivo deste teste não é equilibrar a carga dos hosts de destino no switch S2, porém, validar se a troca de tráfego, com a técnica explicada no item anterior, pode causar alguma perda de pacote ou mesmo lentidão no processo durante o chaveamento entre

um e outro equipamento. Para esta validação, três equipamentos em pontos distintos desta topologia enviam requisições *ICMP request* para o equipamento de IP 10.0.0.1. Estas conexões foram balanceadas entre os equipamentos 10.0.0.1, 10.0.0.2, 10.0.0.3 e 10.0.0.4 de forma aleatória.

Foram realizados 133 trocas de destino e um total de 20400 pacotes enviados para estes 4 hosts. Nenhuma perda de pacote foi observada, apenas alguns desvios de tempo de resposta acontecem em cada atualização do comutador.

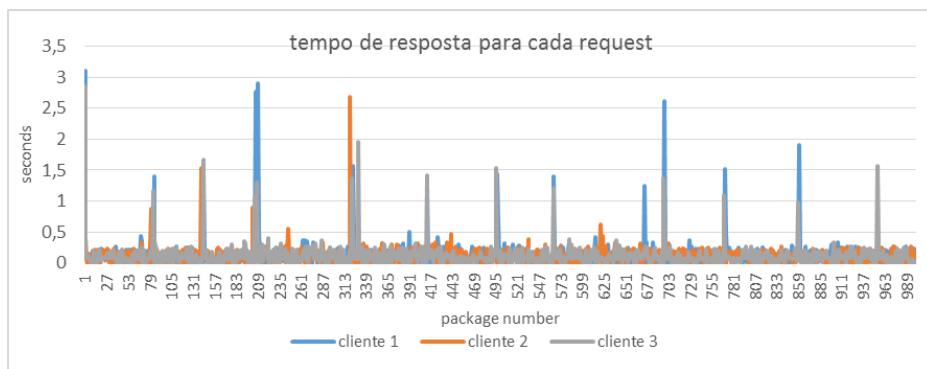


Figura 7. Incremento no tempo de resposta com a alteração de destino

Conforme observado na figura 7, há um incremento de tempo de resposta quando a troca de regras de fluxo acontece. Este tempo é devido à remoção da regra para que o fluxo seja direcionado há um host com melhor performance. Desta forma, a ação de troca de regras pode afetar as filas internas a serem processadas pelo equipamento.

5. Conclusão

Com os atrativos do auto desempenho da tecnologia OpenFlow demonstrados por trabalhos relacionados e a capacidade de realizar平衡amento de carga de forma nativa no núcleo da rede, removendo a necessidade de inclusão de servidores ou serviços adicionais para a realização desta tarefa, o OpenFlow se mostra uma excelente proposta para execução de平衡amento de carga simples onde a utilização de controle de contexto não se faz necessária. O processo de implementação desta tarefa é simples, com a inclusão e alteração de regras para modificação dos pacotes. Esta tarefa pode ser automatizada e atualizada de forma dinâmica de acordo com a necessidade do serviço prestado. Este processo permite elasticidade no quesito de adição, remoção e alteração de hosts para atender um determinado serviço.

Esta pesquisa demonstra que a atualização do plano de fluxos do OpenFlow para troca de regras não gera perda de pacotes ou cria um grande impacto no tempo de resposta, porém, há uma espera maior durante a troca de regras que deve ser considerada pois, os pacotes analisados tiveram uma pequena latência acrescida durante o tempo em que estiveram na fila do comutador aguardando a definição da regra de fluxo deste pacote. Desta forma, esta tarefa precisa ser veloz para minimizar o tempo em que os fluxos estão enfileirados aguardando a regra de decisão.

O uso do protocolo de monitoramento SNMP auxilia o processo de análise de carga de cada host. O modelo do protocolo OpenFlow proporciona uma visão clara e bem definida do processamento de fluxos onde estatísticas são controladas e acessadas de forma centralizada. Neste sentido, o host com melhor qualidade para processamento deve ser o responsável para processar a requisição. A melhor qualidade pode ser determinada considerando ambos itens monitorados onde a performance do hardware é um indicador de carga e o número de fluxos direcionados para cada host é um item *delta* de qualidade. Para não sobrecarregar o host com requisições de informação SNMP, este *delta* item pode ser utilizado para reduzir a capacidade de processamento do host em conjunto com a última informação de consumo de hardware recebida. A cada intervalo de análise um novo host é determinado e as regras de fluxo alteradas se necessário.

Com o uso desta medida de decisão de qualidade, mesmo hosts com diferentes capacidades de hardware podem ter o balanceamento equilibrado de uma forma eficaz. Como futuro desenvolvimento desta pesquisa um algoritmo que faz o uso de ambas as tecnologias será desenvolvido e a eficácia deste algoritmo analisada.

Referências

- [1] Wang Peng, Lan Julong, Chen Shuqiao (2014) "OpenFlow based Flow Slice Load Balancing", p. 72 -82, Communications, China (Volume:11 , Issue: 12)
- [2] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S. e Turnet, J. (2008) "OpenFlow: enabling innovation in campus network", p. 69-74, ACM SIGCOMM Computer Commnuication Review
- [3] Bianco, A., Birke, R., Giraudo, L. e Palacin, M. (2010) "OpenFlow Switching: Data Plane Performance", p. 1-5, Communications (ICC), 2010 IEEE International Conference
- [4] C.C. Okezie, Okafor K.C, Udeze C.C (2013) "Open Flow Virtualization: a Declarative Infrastructure Optimization Scheme for High Performance Computing", p. 232-244, Academic Research International, Vol.4
- [5] Shang, Z., Chen, W., Ma, Q., Wu B (2013) "Design and implementation of server cluster dynamic load balancing based on OpenFlow", p. 691-697, Awareness Science and Technology and Ubi-Media Computing (iCAST-UMEDIA), 2013 International Joint Conference
- [6] Marcon, D., Bays, L. (2012) "Flow Based Load Balancing: Optimizing Web Servers Resource Utilization", p. 76-83, Journal of Applied Computing Research, vol. 1, n. 2
- [7] Kaur, S., Singh, J., Kumar, K., Ghuman, N.S. (2015) "Round-robin based load balancing in Software Defined Networking", p. 2136-2139, Computing for Sustainable Global Development (INDIACom)
- [8] Belyaev, M., Gaivoronski, S. (2014) "Towards load balancing in SDN-networks during DDoS-attacks". P. 1-6, Science and Technology Conference (Modern Networking Technologies) (MoNeTeC)

Uma abordagem prática da tecnologia SDN / OpenFlow em redes voltadas para tráfego de voz e dados

Carlos Alfredo Weissheimer Junior¹, Diego Menine²

¹Docente do curso de Redes de Computadores – Faculdade IENH
Rua Frederico Mentz, 526 – Hamburgo Velho – Novo Hamburgo RS

²Dicente do curso de Redes de Computadores – Faculdade IENH
Rua Frederico Mentz, 526 – Hamburgo Velho – Novo Hamburgo RS
`{carlosawe, diego.sapi}@gmail.com`

Abstract. This article discusses the implementation of a software-defined network (SDN) to manage voice and data traffic using a switch that supports the OpenFlow protocol. The purpose of this document is to provide a fast alternative to applying QoS -shaped devices that can encourage the reader to think of other alternatives to application service quality using the SDN technology. For collecting and analyzing the results, was used the Model E proposed by the ITU- T G703 where searched tried to get an average for each indicator at the end of the test sections. Concluded that in a chaotic environment without application of QoS, use of voice service is not feasible in need of special treatment in this type of service, in form simply can be obtained by using devices that support SDN technology.

Resumo. Este artigo aborda a implementação de uma rede definida por software (SDN) para gerenciar o tráfego de voz e dados utilizando um switch com suporte ao protocolo OpenFlow. O objetivo deste documento é apresentar uma alternativa ágil para aplicação de QoS em dispositivos de forma que, possa incentivar o leitor a pensar em outras alternativas para aplicação de qualidade de serviço utilizando a tecnologia SDN. Para coleta e análise dos resultados, foi utilizado o Modelo E proposto pela ITU-T G703 onde procurou-se obter uma média de cada indicador ao final das seções de teste. Concluiu-se que em um ambiente caótico sem aplicação de QoS, a utilização do serviço de voz se torna inviável necessitando de um tratamento diferenciado neste tipo de serviço, a qual, pode ser obtido de forma simples utilizando dispositivos com suporte à tecnologia SDN.

1. Introdução

Uma nova abordagem de redes de computadores tem ganhado força nos últimos anos como alternativa para a forma de gerenciar, e também de baratear a implementação e expansão de redes de computadores, trata-se da tecnologia SDN (*Software Defined Network* ou Redes Definidas por Software), com o objetivo de simplificar a rede, segundo [Lara 2013]. A tecnologia de redes definidas por software é definida como “[...] uma forma de se implementar políticas de controle de acesso de forma distribuída, a partir de

um mecanismo de supervisão centralizado” [Guedes 2012 apud Casado et al. 2009]. Neste artigo, espera-se contribuir com a comunidade acadêmica, bem como auxiliar administradores, gerentes de projeto e profissionais da área de infraestrutura em T.I. através da quebra de paradigma em relação à utilização de novas tecnologias na área de redes de computadores. E assim, apresentar outra opção de implementação com *software* aberto sem vínculo com fabricantes de *hardware* justifica a pesquisa sobre o assunto.

Como objetivo o presente trabalho implementou e analisou o desempenho de tráfego VoIP utilizando a arquitetura de redes definidas por *software* descritas neste artigo. Neste contexto, este artigo aborda a implementação da tecnologia de redes definidas por *software* e a utilização do protocolo OpenFlow para configuração de um dispositivo físico, buscando comparar o desempenho de uma aplicação de voz sobre IP em um ambiente com grande volume de tráfego sofrendo saturação para análise dos dados obtidos de acordo com os dois cenários propostos neste artigo.

2. Componentes utilizados

Foi utilizado um *switch* modelo TL-TW1043ND versão 1.8 onde o *software* do fabricante foi substituído por um *firmware* da OpenWrt backfire versão 10.03.1 [**OpenWRT 2013**], com suporte ao protocolo OpenFlow versão 1.0, que é uma distribuição Linux com kernel 2.6.32.27 personalizada para ser embarcada nestes dispositivos, [**Moraes et al. 2014**]. As 4 portas LAN do *switch* foram configuradas para aplicar as regras propostas pelo controlador RYU. Um ambiente virtualizado foi criado através da ferramenta Virtual Box da Oracle versão 4.3.30 disponibilizada em uma máquina customizada utilizando um processador Intel Core i3 com 32GB de memória RAM e um HD de 1Tb rodando o sistema operacional Windows 7 para suportar três servidores virtuais: O controlador RYU versão 3.23, o Sip Server Elastix versão 2.4.0 e o CentOS versão 7.

Quanto aos *softwares*, foi utilizado para tráfego de voz, a distribuição StartTrynity Sip Tester versão 2015-03-04 15:01-UTC utilizando o protocolo SIP no modelo cliente/servidor em conjunto com o *Software* Elastix. Também foi utilizado o *software* Iperf versão 2.0.5 para gerar tráfego TCP também no modelo cliente/servidor com o servidor CentOS 7, compondo o ambiente virtualizado melhor visualizado na figura 1.

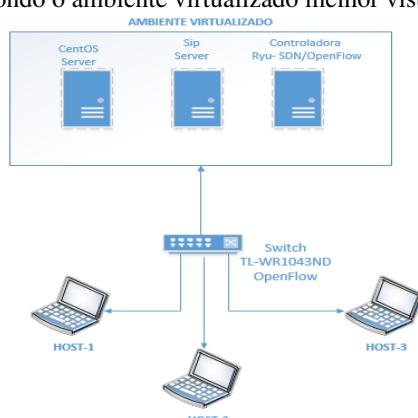


Figura 1 – Arquitetura de rede proposta

Para completar o cenário e esgotar as 4 portas LAN disponibilizadas no *switch*, foram utilizados três máquinas físicas, um *ultrabook* Samsung serie 5 com processador Intel Core i5 e 4 GB de RAM e sistema operacional Windows 8.1 nomeado de *host-1* para gerar tráfego TCP via Iperf com o servidor CentOS, outro *ultrabook* Samsung serie 5 com processador Intel Core i7 com 4 GB de RAM e sistema operacional Windows 8.1 nomeado de *host-2*, utilizado para gerar as chamadas VoIP utilizando o *software* Star Trinity, e um *notebook* Asus com processador Celeron com 2GB de RAM e sistema operacional Windows 7 nomeado de *host-3* para também gerar tráfego TCP executando o *software* Iperf com o servidor CentOS.

3. Referencial bibliográfico

Para que o conceito de SDN (*Software Defined Networks*) ou Redes Definidas por *Software* possa ser compreendido, será introduzida a base teórica quanto aplicação desta tecnologia e os componentes necessários para sua utilização.

Além disso, é apresentado o embasamento quanto ao conceito de aplicação do protocolo OpenFlow, utilizado neste projeto de forma que se torna essencial para o funcionamento da Rede Definida por *Software*, que é o objetivo deste artigo.

3.1. Conceito de SDN (*Software Defined Network*)

O termo SDN surgiu a partir da necessidade de se obter uma melhor forma de gerenciar as redes de computadores conforme [Lópes 2014].

A SDN permite que o administrador de rede gerencie serviços de redes através da abstração das funcionalidades de nível inferior. Isto é conseguido através da dissociação do plano de controle (onde são construídas as decisões de roteamento) e o plano de dados (o nível mais baixo composto pelos dispositivos físicos responsáveis de rotear o tráfego).

Pode ser citado, além da melhora no gerenciamento da rede, o amplo controle de fluxo de dados conforme [Santos et al. 2014]. “*Software Defined Networking (SDN) é um novo paradigma que promete fornecer a capacidade de amplo controle sobre fluxos de tráfego da rede, com o objetivo principal de simplificá-la e de torná-la mais barata e flexível [...]*”.

Desta forma, o objetivo de uma rede SDN é centralizar e separar o controle do encaminhamento de dados, de forma que proporcione uma visão geral da rede e que facilite o seu gerenciamento como um todo. Conforme [Guedes 2012], os elementos comutadores exportam uma interface de comunicação que permite ao controlador analisar, alterar e definir as entradas da tabela de roteamento do comutador. Corroborando a tudo isso [Lara 2013] apresenta uma motivação que estabelece um paralelo importante com o que este projeto aborda.

Uma das motivações da SDN é executar tarefas de rede que não poderiam ser realizadas sem um *software* adicional para cada um dos elementos de comutação. Aplicativos desenvolvidos podem controlar os *switches* rodando em cima de um sistema operacional de rede, que funciona em uma camada intermediária entre o *switch* e a aplicação.

A Figura 2, ilustra a disposição dos elementos de rede em uma arquitetura SDN.

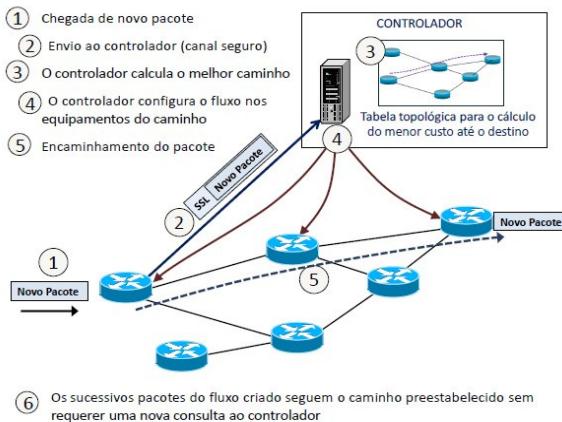


Figura 2 - Elementos de uma rede com arquitetura SDN Fonte: [Lópes 2014]

Conforme [Lópes 2014], quando o primeiro pacote de um fluxo chega ao dispositivo de rede e este não tem uma orientação quanto ao seu tratamento na tabela de fluxos que indique por onde encaminhar este pacote, o dispositivo de rede envia uma consulta ao controlador solicitando como proceder. Desta forma o controlador irá calcular a melhor rota para aquele fluxo específico e através de um protocolo de comunicação irá configurar o fluxo específico no plano de dados dos dispositivos participantes tornando possível o roteamento do pacote.

Na ótica de [Avelar 2013], o OpenFlow é um protocolo de padrão aberto, que implementa o conceito SDN e permite a criação de redes virtuais usando somente recursos L2 (switches Ethernet) com tabelas de fluxo internas e uma interface padrão para adicionar e remover entradas de fluxos. [Astuto et al. 2014], também ressalta a utilização para a indústria que desenvolve dispositivos com suporte a redes definidas por *software*.

4. Metodologia de Pesquisa

A metodologia *Absolute Category Rating* (ACR) padronizada pela ITU-T P.800, foi aplicada para avaliar a qualidade de voz no *Called* a qual recebe uma nota de 1 a 5 de acordo com o índice de *MOS* obtido ao final da seção de teste.

4.1. Técnica de coleta de dados

Para a coleta de dados, foi utilizado o Modelo E normatizado pela ITU-T na recomendação G.107, o qual indica a escala de pontuação definida como *Mean Opinion Score* (MOS) para avaliação da qualidade de voz em uma rede IP. O índice de MOS considera indicadores de perda durante a transmissão de pacotes na rede [Lustosa 2004].

4.2. Simulação e resultados

Conforme orientação da ITU-T P.800, foi definido a quantidade de 10 repetições de teste afim de coletar uma média aproximada entre cada teste, conforme [Gil 2002], “O

procedimento básico adotado na análise estatística nas pesquisas experimentais consiste no teste da diferença entre as médias”.

Neste contexto foram realizadas duas etapas de teste, a primeira utilizando uma configuração padrão para conexão e troca de pacotes pela rede utilizando uma programação disponível no controlador Ryu e a segunda repetindo os mesmos testes, mas utilizando uma programação agregada de QoS para tráfego de voz afim de analisar os índices de MOS, Jitter e Packet Loss (pacotes perdidos) de acordo com os dados obtidos pelo tráfego do *Called*.

4.3. Metodologia aplicada

Para viabilizar os dois cenários, foram editados os arquivos network.conf e o openflow.conf listando as interfaces Eth1 a Eth4 gerenciadas pelo controlador Ryu setado com o IP 192.168.200.254/24. A interface Eth4 recebeu o IP estático 192.168.200.254/24. A configuração simple_switch.conf foi iniciada e o *switch* foi identificado e as portas OpenFlow reconhecidas. A captura de pacotes foi realizada utilizando o *software* Wireshark.

Após a certificação de comunicação entre o *switch* e o controlador, os demais servidores virtuais e *hosts* 1, 2 e 3 foram configurados com IPs estáticos da rede 192.168.100.0/24. O servidor CentOS com IP 192.168.100.254 disponibiliza a conexão TCP na porta 445 via Iperf para aguardar a conexão dos clientes. Além disso, o *host* 1 foi configurado com IP 192.168.100.2/24 e o *host* 3 com IP 192.168.100.4/24.

Enquanto os *hosts* 1 e 3 mantinham o tráfego de dados afim de gerar gargalo no *switch*, o *host* 2 no IP 192.168.100.3/24 iniciava a transmissão de tráfego de voz com o servidor Sip Elastix no IP 192.168.100.100 utilizando o *software* Star Trinity Sip Tester. Para viabilizar o tráfego de voz, foram criados dez usuários no *software* Elastix simulando dez ramais e também uma sala de conferências, a qual atendia as chamadas originadas pelo Star Trinity Sip Tester. Cada chamada executada pelos ramais partindo do *host* 2 continha uma gravação de 30 segundos utilizando o Codec G.711 que era executada igualmente pelos dez ramais registrados no Sip Trinity de forma que os pacotes RTP (voz) pudessem partir do *Caller* para o *Called*.

SIP call quality indicators	Ncalls	Min	Average	Max	Percentile: 90%	95%	98%	99%	99.5%	99.8%	99.9%	99.95%	99.98%	99.99%
Quality indicator name														
Caller lost packets (%) ⁷	10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Caller G.107 MOS ⁷	10	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41
Caller G.107 R-factor ⁷	10	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20
Caller max delta (ms) ⁷	10	32.34	34.47	46.98	37.08	46.98	46.98	46.98	46.98	46.98	46.98	46.98	46.98	46.98
Caller max RFC3550 jitter (ms) ⁷	10	6.94	8.48	9.13	9.11	9.13	9.13	9.13	9.13	9.13	9.13	9.13	9.13	9.13
Caller mean RFC3550 jitter (ms) ⁷	10	6.24	6.26	6.27	6.27	6.27	6.27	6.27	6.27	6.27	6.27	6.27	6.27	6.27
Called lost packets (%) ⁷	10	7.00	11.59	16.06	15.52	16.06	16.06	16.06	16.06	16.06	16.06	16.06	16.06	16.06
Called G.107 MOS ⁷	10	2.63	3.12	3.56	2.67	2.63	2.63	2.63	2.63	2.63	2.63	2.63	2.63	2.63
Called G.107 R-factor ⁷	10	50.97	60.52	69.30	51.79	50.97	50.97	50.97	50.97	50.97	50.97	50.97	50.97	50.97
Called max delta (ms) ⁷	10	63.96	83.89	140.61	97.56	140.61	140.61	140.61	140.61	140.61	140.61	140.61	140.61	140.61
Called max RFC3550 jitter (ms) ⁷	10	7.47	8.26	8.86	8.85	8.86	8.86	8.86	8.86	8.86	8.86	8.86	8.86	8.86
Called mean RFC3550 jitter (ms) ⁷	10	4.28	4.38	4.45	4.44	4.45	4.45	4.45	4.45	4.45	4.45	4.45	4.45	4.45
100 response delay (ms) ⁷	10	90.00	280.40	442.00	416.00	442.00	442.00	442.00	442.00	442.00	442.00	442.00	442.00	442.00
Answer delay (ms) ⁷	10	260.00	408.59	635.00	601.93	635.00	635.00	635.00	635.00	635.00	635.00	635.00	635.00	635.00
-24dB delay (ms) ⁷	10	1722.65	1830.81	1926.88	1924.79	1926.88	1926.88	1926.88	1926.88	1926.88	1926.88	1926.88	1926.88	1926.88
RTCP RTT (ms) ⁷	10	24.38	34.09	42.18	37.86	42.18	42.18	42.18	42.18	42.18	42.18	42.18	42.18	42.18

Figura 3 - Indicadores de qualidade de voz sem QoS.

A Figura 3 demonstra os indicadores de desempenho de voz degradados devido ao grande fluxo de dados na rede. O indicador de *Called Lost Packets* está elevado, o qual afeta diretamente o índice de MOS que fica abaixo do nível mínimo aceitável para o bom

entendimento da voz. Observa-se na Figura 4, que o servidor Sip Elastix, por padrão, já utiliza o DSCP 46 como DiffServ para priorizar o tráfego RTP, mas sem sucesso em um ambiente com alto fluxo de dados e sem QoS.

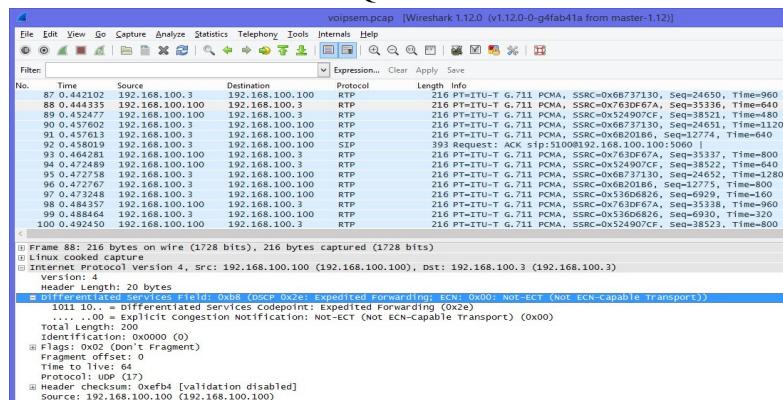


Figura 4 – DSCP 46 (0x2e)

Ao final de dez execuções foram obtidos os seguintes resultados quanto aos indicadores de MOS, atraso entre as entregas de pacotes (Jitter) e pacotes perdidos (pck loss) conforme a Figura 5.



Figura 5 – Gráfico dos indicadores coletados sem QoS

No segundo cenário de testes, mantiveram-se todas as configurações anteriores, com a exceção de uma mudança no controlador, que juntamente com o módulo simple_switch.py, também carregou o módulo rest_qos.py com a possibilidade de se executar priorização de tráfego de acordo com a programação enviada para o switch. Após a mensagem informando a identificação do switch e a confirmação de que o módulo de QoS estava ativo e disponível para receber as programações, foram executados dois comandos no controlador para priorizar o tráfego do DSCP 46 já enviado pelo Elastix e para priorizar o range de portas RTP padrão do servidor SIP que vai da porta 10000 a 20000 UDP.

Repetiu-se o mesmo modelo de teste executado no cenário anterior com as mesmas configurações de IPs e padrões de geração de tráfego com Iperf e com o Sip Trinity. A Figura 6 retorna o ganho de qualidade obtido no tráfego de voz com a priorização aplicada.

SIP call quality indicators	Ncalls	Min	Average	Max	Percentile: 90%	95%	98%	99%	99.5%	99.8%	99.9%	99.95%	99.98%	99.99%
Quality indicator name														
Caller lost packets (%) ⁷	10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Caller G.107 MOS ⁷	10	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41
Caller G.107 R-factor ⁷	10	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20
Caller max delta (ms) ⁷	10	32.83	34.27	37.34	37.31	37.34	37.34	37.34	37.34	37.34	37.34	37.34	37.34	37.34
Caller max RFC3550 jitter (ms) ⁷	10	7.05	8.99	10.78	10.53	10.78	10.78	10.78	10.78	10.78	10.78	10.78	10.78	10.78
Caller mean RFC3550 jitter (ms) ⁷	10	6.23	6.28	6.30	6.30	6.30	6.30	6.30	6.30	6.30	6.30	6.30	6.30	6.30
Called lost packets (%) ⁷	10	0.52	1.09	2.02	1.30	2.02	2.02	2.02	2.02	2.02	2.02	2.02	2.02	2.02
Called G.107 MOS ⁷	10	4.23	4.32	4.37	4.28	4.23	4.23	4.23	4.23	4.23	4.23	4.23	4.23	4.23
Called G.107 R-factor ⁷	10	86.01	89.24	91.28	87.80	86.01	86.01	86.01	86.01	86.01	86.01	86.01	86.01	86.01
Called max delta (ms) ⁷	10	42.45	58.71	153.39	60.12	153.39	153.39	153.39	153.39	153.39	153.39	153.39	153.39	153.39
Called max RFC3550 jitter (ms) ⁷	10	4.54	4.89	5.63	5.26	5.63	5.63	5.63	5.63	5.63	5.63	5.63	5.63	5.63
Called mean RFC3550 jitter (ms) ⁷	10	2.95	3.02	3.06	3.06	3.06	3.06	3.06	3.06	3.06	3.06	3.06	3.06	3.06
100 response delay (ms) ⁷	10	29.00	198.50	409.00	295.00	409.00	409.00	409.00	409.00	409.00	409.00	409.00	409.00	409.00
Answer delay (ms) ⁷	10	154.00	269.84	422.00	406.45	422.00	422.00	422.00	422.00	422.00	422.00	422.00	422.00	422.00
-24dB delay (ms) ⁷	10	1494.56	1595.44	1700.73	1670.22	1700.73	1700.73	1700.73	1700.73	1700.73	1700.73	1700.73	1700.73	1700.73
RTCP RTT (ms) ⁷	10	17.24	19.30	23.01	20.49	23.01	23.01	23.01	23.01	23.01	23.01	23.01	23.01	23.01

Figura 6 – Indicadores de qualidade de voz com QoS

Os resultados quanto aos indicadores de qualidade de voz (MOS), atraso entre as entregas de pacotes (Jitter) e pacotes perdidos (pck loss) apresentaram um excelente desempenho, podendo-se destacar o indicador MOS ficando em 4 pontos conforme a Figura 7.

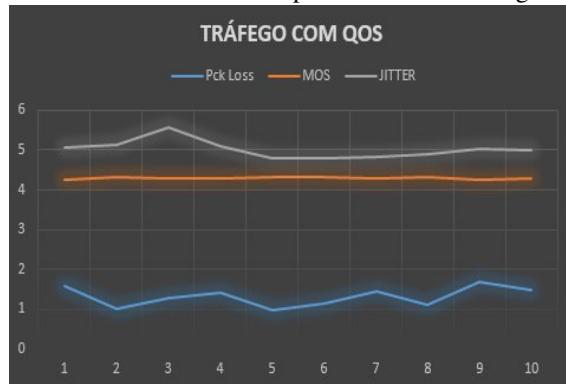


Figura 7 – Gráfico dos indicadores coletados com QoS

Conclusão

Em um ambiente com grande fluxo de dados sem priorização do tráfego de voz, os índices coletados são inaceitáveis inviabilizando a utilização desta modalidade de serviço, neste contexto, foi possível aferir a exigência de regras de controle específicas visando otimizar o desempenho de aplicações com maior sensibilidade a gargalos na rede, as quais podem ser manipuladas de forma centralizada com a implementação de um controlador e do protocolo OpenFlow.

A conclusão apresentada no parágrafo anterior é justificada ao se realizar a análise dos resultados obtidos após a aplicação de QoS na rede de forma rápida através do controlador e do protocolo OpenFlow sem necessitar de acesso ao *switch*, onde o indicador de Packet Loss ficou abaixo dos 2% e o indicador de MOS acima de 4 pontos em uma escala que vai de 1 a 5 onde 5 é excelente.

Referências

- Astuto, Bruno Nunes. Mendon, Marc. Ca, Xuan Nam Nguyen, Katia Obraczka, Thierry Turletti. A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks. Communications Surveys and Tutorials, IEEE

Communications Society, Institute of Electrical and Electronics Engineers (IEEE), 2014, p.1617 – 1634

Avelar, Edson Adriano Maravalho. Pmipflow: uma proposta para gerenciamento de mobilidade em redes definidas por software. Recife 2013. 167 p. Dissertação (Mestrado em Ciências da computação) – Universidade federal de Pernambuco, Recife, 2013. [Orientador: Prof. Kelvin Lopes Dias]

Casado, M., Freedman, M. J., Pettit, J., Luo, L., Gude, n., mckeown, n., and shenker, s. Rethinking enterprise network control. IEEE/ACM Transactions on Networking. 2009 p.1270–1283

Gil, A. C. Como elaborar projetos de pesquisa. 5. ed. São Paulo: Atlas, 2002.

Guedes, d.; vieira, l. F. M.; vieira, m. M.; rodrigues, h.; nunes, r. V. Redes Definidas por Software: uma abordagem sistêmica para o desenvolvimento de pesquisas em Redes de Computadores. Minicursos - Livro Texto do XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. 1ed. Porto Alegre, p. 161-212, 2012.

ITU-T 2003a. ITU-T Recommendation G.107. The E-Model, a computational model for use in transmission planning. Genève, mar. 2003.

ITU-T P.800. ITU-T Series P: Telephone Transmission Quality Methods for objective and subjective assessment of quality

Lara, a.; Kolasani, a.; Ramamurthy, B. Network Innovation Using OpenFlow: A Survey. IEEE Commun. Surv. Tutor. 2013, 16, 1–20.

López Rodríguez, Fernando. Arquitetura e Protótipo de uma Rede SDN-OpenFlow para Provedor de Serviço. Brasília 2014. 59 p. Dissertação (Mestrado em engenharia elétrica) – Universidade de Brasília, Distrito Federal, 2014. [Orientador: Prof. Divanilson Rodrigo de Souza Campelo].

Lustosa, L.C.G., Carvalho, Rodrigues, P.H.A., Mota, S. E., Utilização do Modelo E para avaliação da qualidade da fala em sistemas de comunicação baseados em voz sobre IP, em: Anais do XXII SBRC. Gramado, maio 2004.

Santos, Marcel. Endo, Patricia. Bezerra, Moisés. Gonçalves, Glauco. Sadok, Djamel. Fernandes, Stênio. Revisitando uma Infraestrutura Autonômica: Uma Perspectiva Baseada em uma Rede Definida por Software. Anais do 4º Workshop de Sistemas Distribuídos Autonômicos – WoSiDA. Universidade Federal de Pernambuco (UFPE). 2014

Towards Green SDN: An Approach Based on Graph Connectivity

¹**Eder J. Scheid, ¹Muriel F. Franco, ¹Matias A. K. Schimuneck, ²Cristiano B. Both,
¹Juergen Rochol and ¹Lisandro Z. Granville**

¹Institute of Informatics – Federal University of Rio Grande do Sul (UFRGS)
Porto Alegre – RS – Brazil

²Federal Health Science University of Porto Alegre (UFCSPA)
Porto Alegre – RS – Brazil

¹{ejscheid, mffranco, makschimuneck, juergen, granville}@inf.ufrgs.br
²cbboth@ufcspa.edu.br

***Abstract.** The reduction of energy consumption is a key research topic area in computer networks. Green networking consists of selecting energy-efficient networking technologies and minimizing resource use whenever is possible. Software-Defined Networking (SDN) is a new networking paradigm that allows innovation and simplifies network management, which leads to the development of new approaches to reduce energy consumption. In this paper, we present a solution for energy efficiency in business networks, based on the shutting down switches identified as idle. Our proposed approach have been prototyped using a component-based SDN framework denominated Ryu, which has been used to create the control application. The evaluation results discloses the influence of switch usage to save energy. We can eliminate inactive switches and redundancy, and so, reduce energy consumption in a simulated environment.*

1. Introduction

Reduction of energy consumption is becoming an interesting research topic in computer networks [Bianzino et al. 2012]. In this context, Green networking arise as a new approach to rethink network design and implementation. Traditionally, networking systems are designed and dimensioned with a set of classical principles, but in the recent years, a new energy concern emerged together with technology evolution. The primary goal of Green networking is minimizing resource usage, by shutting down of inactive routers, and consequently, leading to energy saving.

Nowadays, many challenges related to power management emerge. Both Network Functions Virtualization (NFV) and Software-Defined Networking (SDN) have high potential and in many ways are mutually supportive [Feamster et al. 2014]. SDN and Virtualization solve many problems that Network and Service Providers have to face for optimizing their infrastructures, such as reducing costs. Besides that, these paradigms emerge as an interesting tool to help the reduction of the network's energy consumption.

SDN is a new paradigm that allows innovations and simplifies network management. It is dynamic, manageable, cost-effective, and adaptable, making it ideal for dynamic systems. This architecture decouples the network into two planes, the control plane and the forwarding plane. This paradigm improves the capacity to design, deploy, and manage energy-efficient systems [Jammal et al. 2014].

In large networks, switches and forwarding devices can be turned off when inactive or dispensable, in order to save energy. We consider that a switch is inactive when there is no traffic and is dispensable when others switches that can sustain the traffic and support the connectivity of the network. It is essential that energy saving actions should not interrupt or produce a limitation in the network traffic.

In this paper, we present a prototype that discovers the minimum green routing path, where the traffic pattern is used to support its decisions. When network traffic is considered low, we use a customized topology to avoid the usage of some switches. Our algorithm is based on using a technique that eliminates redundant switches in the topology. An SDN controller can then find which switches can be removed and route the packets based on traffic observation patterns.

The remaining of this paper is organized as follows. In Section 2, we present a brief topic overview and related work. In Section 3, we introduce our controller prototype and describe our algorithm in details. In Section 4, we present a measurement environment, results, and discussions. Finally, in Section 5, we conclude the paper and list future work.

2. Background and Related Work

A significant challenge in green networking is finding a better way to guarantee full network operation while saving energy. Albers shows in his work [Albers 2010] that the two states advanced algorithm is a positive and simple way to reduce energy consumption. Grupta and Singh, in turn, was the pioneer on verifying the impact of network protocols on energy saving by putting network interfaces to a sleep state. These authors identify the problem of excessive energy consumption on the Internet and propose sleeping as an approach to save energy.

Bolla et al. proposed an energy-saving mechanisms based on topology optimization for the extension of traffic engineering. The basic approach is to reduce the network capacity, in terms of links and nodes, to match the actual traffic volumes. In other words, they try to route all traffic flows among network nodes using the minimum number of network resources [Bolla et al. 2011].

SDN can present an opportunity for the allocation of traffic flows and, at the same time, facilitating the management of power state of network nodes. As a consequence, recent Future Green Internet research efforts have been focusing on the SDN paradigm to ease energy management. Bruschi et al. propose a Green Abstraction Layer (GAL) to integrate the power management data and then it applies control strategies inside the SDN [Bruschi et al. 2014]. The results support the feasibility of allocating resources according to the incoming traffic characteristics while reducing the overall network energy consumption.

Other related work explores the SDN paradigm based on energy-aware flow scheduling. Li et al., for example, use exclusive routing for each flow in a data center to guarantee that it does not compete for link bandwidths with others flows [Li et al. 2014]. Thanh et al. propose a new testbed architecture that combines hardware network devices with virtual emulation test environment to improve scalability, flexibility, and accuracy [Thanh et al. 2013]. This testbed enables the design and the experiment of new solutions

for an energy-efficient data center. It combines smart sleeping and power scaling mechanisms. This new algorithm proposed can save up to 35% of energy consumption in case of a 16-server data center. Almost all the work in the proposal was developed, concluding that the energy saving level can be improved as the size of the data center increases.

3. An Approach to Green SDN

In our approach, we seek to reach an optimized topology in terms of energy efficiency. Therefore, we expect to shutdown the largest number of switches as possible. The choice of turning on or off a switch is performed according to the existence of alternative routes. If there are other ways to route the traffic, the switch can be turned off. Thus, in a low traffic situation, this switch remains off, being activated only in the event of the increase of demand. Another concern is related to the energy waste for restarting network equipment. Considering that this process can be very costly, we define a waiting window of three hours before changing a switch state.

Algorithm 1 optimized_graph_discovery

```
Input: G-graph
Output: S-graph
for i in switches do
    edges_copy  $\leftarrow$  edges(s_graph, i)
    s_graph.remove_node(i)
    if is_connected(s_graph) is False then
        s_graph.add_node(i)
        s_graph.add_edge(edges_copy)
    end if
end for
```

In the first step of our implementation, we map the network topology in a G -graph. After, we find a G -subgraph that contains all connected hosts. This subgraph, will be our optimized graph S . Aiming to exclude unnecessary interfaces, we focus on a graph property called connectivity, which is one of the fundamental concepts of graph theory: it asks for the minimum number of elements (nodes or edges) that need to be removed to disconnect the remaining nodes from each other. Algorithm 1, is used to find a S -graph.

Algorithm 1 creates a copy of G -graph, which is called S -graph, and applies a greedy algorithm in A -graph to find nodes that can be removed. First, it removes $switch_n$ from S -graph and verifies the graph connectivity. If the connectivity is still true, $switch_n$ can be removed; otherwise, $switch_n$ is inserted in the S -graph again.

Algorithm 2 find_routing_path

```
Input: source, destination
Output: forward-path
if currently_traffic is "Low" then
    path  $\leftarrow$  shortest_path(s_graph, src, dst)
else
    path  $\leftarrow$  shortest_path(g_graph, src, dst)
end if
```

After calculating S -graph, we implement the controller algorithm. This implementation, consists in defining the best topology to use for a determined network state. For example, when the network is experiencing a high demand, we need to use G -graph to forward packets. Otherwise, when the network is experiencing low traffic, we set G -graph and remove switches that are not in use. Algorithm 2 is used to define current path of the incoming packets.

As Algorithm 2 denotes, we check the current period flag. If the flag has the "low" label, we find the shortest-path between packet source and destination in S -graph, and with this path we set new rules in the switches to forward the packets. On other hand, when the flag has the "high" label, we use G -graph to calculate the route, since it is an indicative that the network traffic requires a large number of network resources.

Our proposed approach have been prototyped using component-based software-defined networking framework called Ryu [Team 2014], which has been used to develop the control application. This framework offers a platform for building SDN applications and provides useful libraries and a well-defined API to use all of OpenFlow (OF) versions. The controller waits for an *OFPacketIn* event to make a decision to forward the packet. If there is no rule for the destination, we add a flow rule based on the currently topology graph.

This approach has a good performance in networks with high-redundancy and multiple routes. Moreover, it is easy to maintain and add new functions. Also, it can be promptly activated in any network, with just a few configurations. For example, network administrators need to set traffic periods patterns, and the algorithm is responsible for removing switches and reporting to the administrator details about the current network status and switches.

3.1. Policy Monitor and Enforcer

To determine when to use an approach that reduces the energy consumption in the network, policies have been written. These policies are based on a traffic pattern measured on a link maintained by the RNP (the Brazilian research and education network) [Rede Nacional de Ensino e Pesquisa 2015].

The traffic pattern analyzed was the traffic over a week (Figure 1). In this graphic Axis Y represents the input (grey area) and output (black line) traffic on a link between two Brazilian states, while Axis X represents the day of the week. We can see that around midnight the traffic decreases from 7 Gbps to 1 Gbps and continues at this rate until the morning. The assumption inferred repeats itself, creating spikes between the 9th and the 12th. On the other hand, on the 13th and the 14th, which is Saturday and Sunday, there is minor traffic on the network, but on the 15th the traffic returns at its normal rate. Therefore, between these hours (midnight and 6-am) and on the weekends, the network can stay at a power conservative state.

Having those policies determined, a monitor was implemented within the controller. This monitor is responsible for enforcing the energy conservation policy if there is a match during the monitoring. Every fifteen minutes, it compares the current date and time with the hard-coded policy, and if the network (graph) is not optimized yet, it will optimize it. Also, when the current date and time do not match the policy, meaning that there is a need for the full network, the graph is restored to its full capacity.

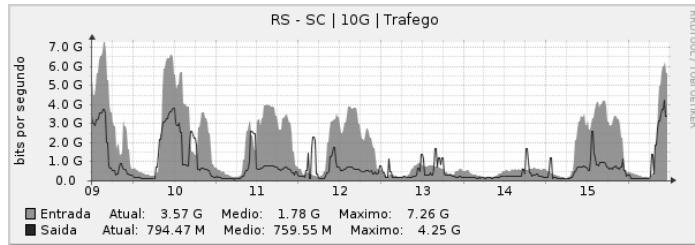


Figure 1. Traffic pattern over a 7-day period. [RNP 2015]

4. Experiments and Discussion

In order to demonstrate our controller actions in different scenarios, we used two different topologies simulated in Mininet [Lantz et al. 2010]. Besides, we performed tests to measure our approach in terms of energy efficiency. In this section, we present and discuss our results and the tools that were necessary to achieve them. First, we present the techniques and tools that were used to create the simulated environment and the controller evaluation. Afterwards, we present our experiments and provide a discussion about the controller behavior.

4.1. Measurement environment and tools

The measurement environment was created and simulated in Mininet, which is a system for rapidly prototyping large networks on the constrained resources of a single desktop computer. It is able to create a realistic virtual network, running real kernel, switch and application code. We used the Mininet Python API to create our topology and carry out interactive performance tests. We proposed a scenario, which is presented along this subsection, to evaluate our approach.

To generate and analyse the network traffic, we use Iperf [Hsu and Kremer 1998] and Wireshark [Chappell and Combs 2010]. Iperf is applied to measure the maximum TCP bandwidth, allowing the tune of various parameters on UDP packets. It was used to generate UDP traffic in our simulated network. Also, we created a performance test that generates traffic between virtual hosts. During the experiment, the virtual hosts sent UDP packets across the network considering the network bandwidth limit. Wireshark was used to monitor the traffic between switches.

All the experiments were performed on two different machines. One of the machines running Debian 7.8, an Intel Core 2 Duo CPU @ 2.33GHz with 2GB of RAM, was responsible for executing the controller. The other machine, a Raspberry Pi Model B [Halfacree and Upton 2012], performed all the virtualization of the network with Mininet. The Raspberry Pi, also running Debian 7.8, contains a 700MHz ARM processor and 512MB of RAM, was capable of running the test scenario. The workload was generated through an ssh connection using an external terminal with the Iperf tool.

4.2. Scenario

We depict, in Figure 2 (a), a network topology modeled in Mininet, which was used to apply our controller tests and traffic measurements. This topology consists of twelve

switches and nine hosts. In order to perform the traffic measurement, we monitored UDP packets in a single switch, s_7 . It is important to notice that all switches connections have a 1Gbps virtual bandwidth capacity.

For evaluation purposes, we considered three servers that receive requests and send responses: h_8 , h_3 and h_7 . All other hosts are clients. This topology was chosen because it provides redundancy and sufficient complexity to demonstrate our algorithm efficiency. Figure 2 (b) shows the optimized version of the topology computed by our algorithm.

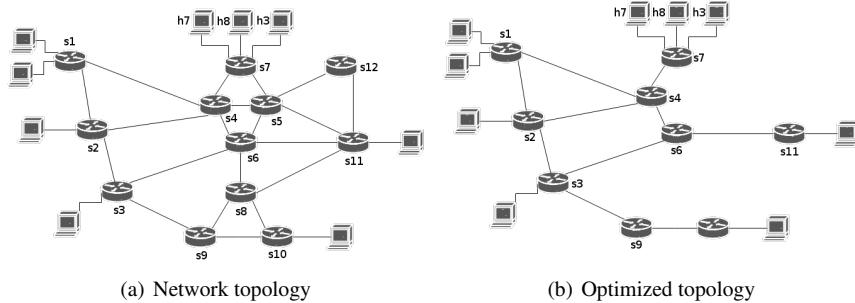


Figure 2. Scenario

4.3. Results

Considering that one single switch consumes 50Wh and that one port of the switch consumes 5Wh (10Wh per link), the topology consumes a total of 800Wh (12 switches + 20 links). When the optimization algorithm is applied to this topology, switches s_5 , s_8 , and s_{12} are removed along with their respective links (Figure 2 (b)). This removal represents a drop of 25% in the number of switches, and the energy consumption decreases from 800Wh to 550Wh (9 switches + 10 links), representing 31% of savings in one hour.

To demonstrate that the policies were being enforced correctly, a 7-day period was simulated in our environment. Every second was interpreted as a real-world hour. This representation allows quick tests over the topology and the controller. The controller, at every simulated hour, reports the number of switches that are present in the network and how much energy it is consuming.

In Figure 3, the number of switches that are being currently used in the network is presented in Axis Y. This plot follows the same pattern as the traffic flow presented in the Figure 1, which corresponds to the RNP link. We noticed that the number of switches corresponds to the traffic flow in the network; when the network is experiencing a low traffic demand, *e.g.*, on Saturday, Sunday, and Monday at dawn, the controller optimizes the graph and uses only nine switches.

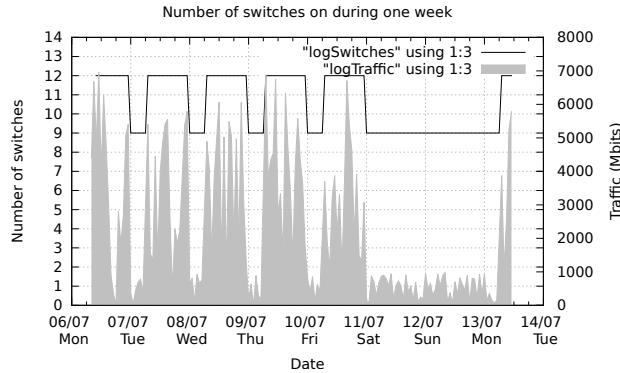


Figure 3. Network traffic analysis and daily active switches

With the policies being hard-coded within the controller, the number of switches being used will follow the same pattern weekly. It is visible that when the policies are enforced, the network will stop routing traffic through three switches. Therefore, these switches will be idle thus consuming less energy.

5. Conclusions and Future Work

The reduction of energy consumption is an important research topic in computer networks. In this context, Green Networking arises as a novel approach to rethinking the network design and deployment. The principal goal of green networking is to minimize resource usage, by turning off inactive routers, whenever possible, and with that, saving energy. To create dynamic and manageable systems, we leverage the SDN paradigm, which improves the capacity to design, deploy, and control energy-efficient networking systems.

In this work, we presented an SDN controller based on OpenFlow 1.0 for the management of packet forwarding and switch state. When network traffic is considered low, we use a customized topology to avoid using idle switches. Our algorithm consists of updating routing tables during idle periods while eliminating redundancy in the network. We reported on a prototype to find switches that can be removed based on the traffic observation pattern.

The evaluation results disclose the influence of switches usage to saving energy. We can eliminate inactive interfaces and redundancy, and so, reduce energy consumption. In the simulated solution, we decreased 31% of power consumption during the usage of the optimized solution and demonstrated the efficiency of our algorithm in terms of energy saving in networks with high redundancy. Our controller thus identifies low and high traffic periods based on policies, and can calculate new routing path, and with that avoid idle switches in low demand period.

As future work, we intend to: i) bring concepts from the Steiner-Tree problem to improve our controller solution, ii) implement a policy database with a more robust-enforcer and monitor (this database should allow the operator to create, remove and update

policies), iii) analyse other metrics (*e.g.*, delay) to evaluate our solution, and iv) compare our work to others Green SDN algorithms in heterogeneous environments.

References

- Albers, S. (2010). Energy-Efficient Algorithms. *Communications of the ACM*, 53:86–96.
- Bianzino, A. P., Chaudet, C., Rossi, D., and Rougier, J.-L. (2012). A Survey of Green Networking Research. *IEEE Communications Surveys Tutorials*, 14(1).
- Bolla, R., Divoli, F., and Cucchietti, F. (2011). Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures. *IEEE Communications Surveys Tutorials*, 13(2):223–244.
- Bruschi, R., Lombardo, A., Bolla, R., and Morabito, G. (2014). Green Extension of OpenFlow. *Teletraffic Congress (ITC)*, pages 1–6.
- Chappell, L. and Combs, G. (2010). *Wireshark Network Analysis: The Official Wireshark Certified Network Analyst Study Guide*. Laura Chappell University, 2 edition.
- Feamster, N., Rexford, J., and Zegura, E. (2014). The Road to SDN: An Intellectual History of Programmable Networks. *ACM SIGCOMM Computer Communication*, pages 87–98.
- Halfacree, G. and Upton, E. (2012). *Raspberry Pi User Guide*. Wiley Publishing, 1st edition.
- Hsu, C.-H. and Kremer, U. (1998). IPERF: A Framework for Automatic Construction of Performance Prediction Models. *Workshop on Profile and Feedback-Directed Compilation*.
- Jammal, M., Singh, T., Shami, A., Asal, R., and Li, Y. (2014). Software-Defined Networking: State of the Art and Research Challenges. *Computer Networks*, 72:201–213.
- Lantz, B., Heller, B., and McKeown, N. (2010). A Network in a Laptop: Rapid Prototyping for Software-Defined Networks. *ACM SIGCOMM Workshop on Hot Topics in Networks*.
- Li, D., Shang, Y., and Chen, C. (2014). Software Defined Green Data Center Network with Exclusive Routing. *IEEE INFOCOM*, pages 1743–1751.
- Rede Nacional de Ensino e Pesquisa (2015). Trafego Semanal RS-SC. <http://www.rnp.br/>. Accessed: 2015-06-16.
- Team, R. P. (2014). *RYU SDN Framework*. Ryu Project Team.
- Thanh, N. H., Cuong, B. D., and Thien, T. D. (2013). ECODANE: A Customizable Hybrid Testbed for Green Data Center Networks. *Advanced Technologies for Communications (ATC)*.

II

Sessão 2 - Computação em Nuvem

Um Modelo de Consumo de Energia para Ambientes de Nuvem com Elasticidade

Gustavo Rostirolla¹, Vinicius Facco Rodrigues¹, Rodrigo da Rosa Righi¹

¹Prog. Interdisciplinar de Pós-Graduação em Computação Aplicada, Unisinos – Brasil
grostirollal@gmail.com, viniciusfacco@live.com, rrrighi@unisinos.br

Resumo. *Uma das principais características da computação em nuvem é a elasticidade, que se refere à capacidade de alterar a quantidade de recursos em tempo real a fim de otimizar a execução de uma tarefa. Um dos principais desafios é como medir a sua eficácia, utilizando elasticidade em aplicações HPC é possível reduzir o tempo de aplicação, mas consumindo uma grande quantidade de recursos e/ou energia para concluir a tarefa. Particularmente, observa-se que o estado da arte não apresenta um modelo de consumo de energia que contempla um número maleável de recursos, mas apenas um número fixo e predefinido deles. Neste contexto, propõe-se um modelo de consumo de energia elástico. Os resultados revelaram uma acurácia média de 97,15%.*

Abstract. *One of the main characteristics of cloud computing is elasticity, which refers to the capacity of on-the-fly changing the number of resources to support the execution of a task. One of the main challenges in this scope is how to measure its effectiveness, because of elasticity enables high performance computing by reducing the application time, but an infeasible amount of resource and/or energy can be paid to accomplish this. Particularly, the state-of-the-art does not present an energy consumption model that fits a malleable number of resources, but only a fixed and predefined number of them. In this context, this article proposes an elastic energy consumption model. The results revealed a median accuracy of 97.15%.*

1. Introdução

Uma das principais características da computação em nuvem é a elasticidade, na qual os usuários podem escalar seus recursos computacionais a qualquer momento, de acordo com a demanda ou o tempo de resposta desejado [Lorido-Botran et al. 2014]. Considerando uma aplicação paralela de longa execução, um usuário pode querer aumentar o número de instâncias para tentar reduzir o tempo de conclusão da tarefa. Logicamente, o sucesso deste processo vai depender tanto do grão quanto da modelagem da aplicação. Por outro lado, se a tarefa não escala de forma linear ou perto de uma forma linear, e se o utilizador é flexível com respeito ao tempo de conclusão, o número de instâncias pode ser reduzida. Isso resulta em uma menor quantidade nós × horas, e portanto, em um custo mais baixo e melhor uso da energia. Graças aos avanços na área de virtualização [Petrides et al. 2012], a elasticidade em computação em nuvem pode ser uma alternativa viável para obter economia de custo significativa quando comparado com o método tradicional de manter uma infra-estrutura de TI baseada em *cluster*. Normalmente, neste último caso, há um dimensionamento para o uso de pico, sendo subutilizada quando observamos toda a execução do aplicativo ou ainda, ao analisar o uso real da infra-estrutura.

Elasticidade pode ser uma faca de dois gumes envolvendo desempenho e o consumo de energia. Ambos são diretamente relacionados ao consumo de recursos, o que também pode ajudar a medir a qualidade elasticidade. Embora elasticidade permita que os aplicativos aloquem e liberem recursos de forma dinâmica, ajustando às demandas da aplicação, estabelecer limites apropriados, medir o desempenho e consumo de energia com precisão neste ambiente não são tarefas fáceis [Lorido-Botran et al. 2014]. Desta forma, um utilizador pode conseguir um bom desempenho considerando o tempo para executar a sua aplicação, mas utilizando uma grande quantidade de recursos, resultando em um desperdício de energia. A ideia de apenas obter um melhor desempenho da aplicação com uma execução elástica, em alguns casos, não é suficiente para usuários e administradores da nuvem. Os usuários acabam pagando por um maior número de recursos, não efetivamente utilizados, de acordo com o paradigma *pay-as-you-go*. A medição do consumo de energia de tais sistemas elásticos não é uma tarefa fácil. Muitos trabalhos se concentram em medição e como estimar o consumo de energia em *data centers*, no entanto, essas tarefas são desafios ao considerar sistemas elásticos.

Desta forma, este artigo apresenta um modelo de consumo de energia pra ambientes elásticos que fornece dados sobre a energia consumida durante a execução de aplicações HPC *High Performance Computing* em ambientes de nuvem elásticos. Particularmente, o modelo proposto extrai dados de consumo de energia em uma infra-estrutura maleável (que permite variação do número de nós em tempo de execução), permitindo estabelecer relações entre o consumo de energia, consumo de recursos e desempenho. Com o objetivo de analisar o modelo de energia proposto, utilizou-se um trabalho anterior chamado AutoElastic [Righi et al. 2015], que consiste em um *middleware* que prove elasticidade reativa e gerencia os recursos da nuvem de acordo com a demanda de uma aplicação HPC. Assim, o modelo de energia atua como um complemento para o AutoElastic, salvando dados de energia durante o tempo de execução do aplicativo. Os resultados com uma aplicação de uso intensivo da CPU foram realizados em diferentes cenários: variando valores dos *thresholds* inferior e superior e variando as cargas de trabalho de entrada (Crescente, Descendente, Constante e Onda). A contribuição científica do artigo consiste no modelo de energia, incluindo equações e procedimentos de captura de dados, para infraestruturas de nuvem elásticas. Este modelo pode ser utilizado para medir a qualidade (*i.e.* eficácia), do *middleware* que prove elasticidade principalmente quando utilizados em conjunto com funções de custo.

O restante deste artigo irá apresentar primeiramente modelo de consumo energético proposto na Seção 2. A metodologia de avaliação e discussão dos resultados estão descritos na Seção 3. A Seção 4 apresentar os trabalhos relacionados. Por fim, a Seção 5 apresenta as considerações finais, destacando as contribuições com dados quantitativos e a direção dos trabalhos futuros.

2. Modelo de consumo de energia para ambientes elásticos

Esta seção apresenta um modelo de consumo de energia para extrair dados sobre o consumo, explorando as relações entre o consumo de energia, consumo de recursos e desempenho. O modelo apresentado leva em consideração uma das principais características da computação em nuvem, a elasticidade, onde a quantidade de recursos muda durante o tempo de execução, assim como o consumo de energia.

A implantação de sensores de corrente ou Wattímetros pode ser caro se não for feito no momento em que toda a infraestrutura (*i.e.*, *cluster* ou *data center*) é instalada, além de ser custosa tanto em questões financeiras como em tempo conforme a infraestrutura cresce. Uma solução alternativa e menos dispendiosa é a utilização de modelos de energia para estimar o consumo de componentes ou de um *data center* inteiro [Orgerie et al. 2014]. Bons modelos devem ser leves (em relação ao consumo de recursos computacionais) e não interferir no consumo de energia que eles tentam estimar. Tendo em vista estes requisitos, o modelo proposto explora dados de energia capturados em um pequeno conjunto de nós, a fim de formular uma equação que estende os resultados para um conjunto arbitrário de nós homogêneos. Mais precisamente, a metodologia utilizada é similar a de Luo et al. [Luo et al. 2013] que consiste em três etapas:

- (i) Coletar amostras de uso de recursos, bem como o consumo de energia da máquina utilizando um medidor de consumo. Neste caso, utilizou-se um medidor Minipa ET-4090 que coletou mais de 8000 amostras usando uma carga composta que pode consumir diversos tipos de recursos dos nós, a fim de representar aplicações reais em ambiente de nuvem [Chen et al. 2014];
- (ii) Executar métodos de regressão para gerar o modelo de energia a ser utilizada posteriormente;
- (iii) Testar o modelo em um conjunto diferente de dados, coletados com o medidor de diferentes máquinas homogêneas, a fim de validar se o modelo representa corretamente o consumo de energia das demais máquinas.

A fim de analisar a precisão do modelo gerado foram coletados dados de CPU, memória principal e consumo de energia instantâneos, aplicando posteriormente PCR (Regressão de Componentes Principais) em mais de 8.000 amostras obtidas a partir de um único nó. Os dados recolhidos estão alinhados com estudos anteriores [Orgerie et al. 2014], que apresentam a CPU como o principal vilão do consumo de energia. Após a geração deste modelo foi realizada a predição da mesma quantidade de amostras de energia baseada em amostras coletadas de CPU e memória de outro nó com mesma configuração de hardware. Comparando estas amostras geradas pela predição de consumo, com as amostras coletadas com o medidor, obteve-se uma precisão média e mediana de 97,15% e 97,72% respectivamente, como pode ser visto na Figura 1.

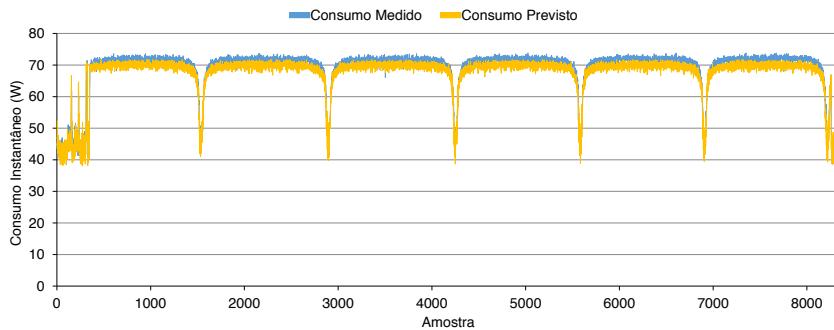


Figura 1. Comparativo do consumo instantâneo entre o consumo previsto e o consumo medido.

Após a execução da aplicação, os dados de CPU e memória principal são utilizados como entrada no modelo gerado, a fim de se obter o consumo de energia instantânea,

medido em Watts (W). A grande vantagem deste modelo é o fato de considerar a elasticidade da nuvem, em outras palavras, o modelo leva em conta apenas o consumo de energia dos recursos que foram efetivamente utilizados, e não o consumo total do *data center*, ou um nó específico. O uso de recursos é coletados de todos os nós durante o tempo de execução da aplicação, e através de um arquivo de log que informa o intervalo de tempo que cada máquina é utilizada, apenas as amostras relativas a execução da aplicação são consideradas para o cálculo do consumo de energia. Este processamento de registro é executado *post-mortem* e permite uma análise mais precisa do consumo de energia da aplicação, e não apenas o consumo de energia de toda a infraestrutura. Este granularidade mais fina permite a utilização de funções de custo, por exemplo, a fim de determinar a viabilidade da utilização da elasticidade em nuvem para executar uma determinada aplicação.

O modelo de consumo também poderia ser empregado para avaliar os ambientes de computação em nuvem heterogêneos, uma vez que é baseado em um modelo já consolidado [Orgerie et al. 2014] apresentado na Equação 1 onde α representa um consumo de energia quando o nó está ocioso e β e δ representam o consumo de energia variável determinado pela quantidade de recursos utilizados (neste caso de CPU e de memória) e retornando o consumo de energia instantâneo em Watts. A única adaptação necessária para contemplar ambientes heterogêneos seria a criação de modelos de consumo de energia distintos para cada tipo de máquina presente no *data center*.

Para complementar esta análise, apresenta-se um conjunto de equações que permitem o cálculo do consumo de energia em ambientes elásticos e também a quantidade de energia gasta por um determinado número de nós. A Equação 2 resulta no consumo de energia de uma máquina m de acordo com o valor de CPU e memória registrados em um instante i , utilizando a Equação 1 como base. A Equação 3 é utilizada para calcular o consumo total de energia de todas as máquinas alocadas em um instante t , ou seja, levando em conta a elasticidade, retornando o consumo em Watts. A Equação 4 calcula o consumo de energia total de um instante 0 a um instante t onde intervalos de tempo são calculados em segundos e utilizando a Equação 3 mencionada anteriormente que já considera a questão elasticidade, este cálculo resulta no consumo de energia em Joules ($W \times \text{segundo}$). Finalmente, a Equação 5 apresenta o consumo de energia da aplicação quando utilizando uma quantidade específica de nós representados por z . Este cálculo resulta no consumo total de energia, também representada em Joules, gasto quando utilizando esta quantidade específica de nós.

$$f(\text{CPU}, \text{Memoria}) = \alpha + \beta \times \text{CPU} + \delta \times \text{Memoria} \quad (1)$$

$$MC(m, i) = f(\text{CPU}(m, i), \text{Memoria}(m, i)) \quad (2)$$

$$ETC(t) = \sum_{i=0}^{M\text{aquina}} MC(i, t) \times x \begin{cases} x = 0 & \text{se a máquina } i \text{ não está ativa no instante } t; \\ x = 1 & \text{se a máquina } i \text{ está ativa no instante } t. \end{cases} \quad (3)$$

$$TC(t) = \sum_{i=0}^t ETC(i) \{ 0 \leq t \leq \text{TempoTotalAplicacao} \} \quad (4)$$

$$NEC(z) = \sum_{i=0}^{T\text{empoApp}} ETC(i) \times y \begin{cases} y = 0 & \text{se no instante } i \text{ o total de máquinas ativas } \neq z; \\ y = 1 & \text{se no instante } i \text{ o total de máquinas ativas } = z. \end{cases} \quad (5)$$

3. Análise dos resultados

Os experimentos foram conduzidos utilizando a nuvem privada OpenNebula¹ com 6 nós(1 FrontEnd e 5 nós). As máquinas utilizadas possuem processadores dual-core de 2,9 GHz com 4 GB de memória RAM e uma rede de interconexão de 100 Mbps. Um total de quatro padrões de carga (Crescente, Descendente, Constante e Onda) foram utilizados com o middleware AutoElastic [Righi et al. 2015] com e sem o recurso de elasticidade. No caso em que a elasticidade é ativa, os *thresholds* utilizados foram 70% e 90% para o limite superior e 30% e 50% para o limite inferior. Como resultado de uma combinação simples, todas as cargas foram testadas 4 vezes utilizando elasticidade, onde 4 é o número de combinações de *thresholds* superior e inferior selecionadas. Todas as execuções iniciaram a partir do mesmo cenário que consiste em um único nó com duas máquinas virtuais (igual ao número de núcleos da máquina). Nas execuções elásticas, a nuvem pode dimensionar para um limite de cinco nós (10 máquinas virtuais) definida por uma SLA (*Service Level Agreement*). O middleware AutoElastic registra quais nós foram utilizados e intervalo de tempo a fim de analisar o consumo de energia de forma elástica posteriormente.

A Figura 2 ilustra o consumo de energia em Watts de acordo com o modelo apresentado quando as ações de elasticidade estão desativadas. Neste contexto, um único nó com duas VMs está sendo utilizado para hospedar os processos escravos. Aqui, podemos observar que o simples fato de ligar o nó computacional (executando o sistema operacional Ubuntu e o middleware AutoElastic) consome cerca de 40 Watts. Qualquer computação realizada provoca uma elevação desse índice para o intervalo entre 40 e 71 Watts. Embora a função Crescente cresça lentamente, o consumo de energia aumenta rapidamente para o limite superior do intervalo. O mesmo comportamento aparece nas funções Decrescente e Onda.

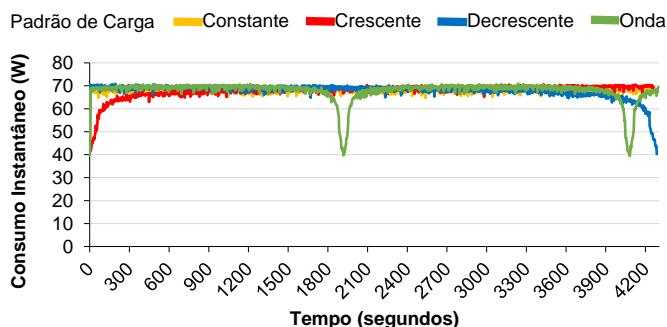


Figura 2. Consumo de energia instantâneo das diferentes cargas sem elasticidade.

Quando a elasticidade é ativada há uma variação elevada na quantidade de VMs utilizada durante a execução da aplicação. A Figura 3 apresenta o perfil de consumo de energia da aplicação de acordo com o número de VMs empregada para resolver o problema e sua contribuição no consumo de energia total (energia consumida até o término da aplicação) de acordo com a Equação 5. Na Figura 3, o resultado da Equação 5 foi traduzido para VMs considerando que os nós são homogêneos e possuem dois núcleos onde cada VM foi mapeada em um núcleo.

¹<http://opennebula.org/>

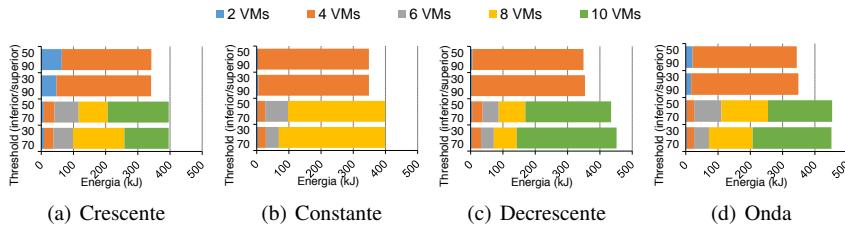


Figura 3. Consumo energético para diferentes quantidades de máquinas virtuais e cargas de trabalho variando os thresholds inferior e superior.

Considerando um modelo de consumo de energia que desconsidera a elasticidade o limite inferior do consumo instantâneo seria de 200 W uma vez que o consumo de energia de cada máquina ociosa é de 40 W conforme apresentado na Figura 2 e destacado pelo α na Equação 1. A Figura 4 apresenta o gráfico de execução destacando picos e quedas bruscas de consumo de energia quando se analisa o consumo de energia de forma elástica, utilizando a Equação 3 durante o tempo total de execução da aplicação. Neste gráfico podemos observar alocação e desalocação de hosts, além de oscilações durante a inicialização das VMs. Estes gráficos apresentam as vantagens em analisar a aplicação utilizando um modelo elástico, pois considera apenas o consumo de energia das máquinas que executam computação, e representa de forma mais fiel o consumo energético de uma aplicação que faz uso da elasticidade.

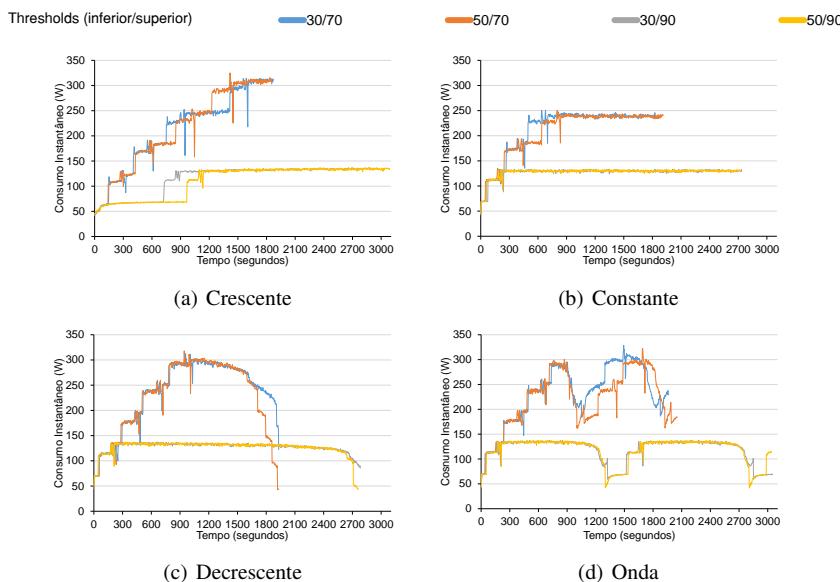


Figura 4. Comportamento do consumo energético das diferentes cargas de trabalho variando os thresholds inferior e superior.

4. Trabalhos relacionados

Alguns trabalhos concentram-se em modelos para estimar o consumo de energia em ambientes de nuvem, no entanto, estas obras não levam em conta a elasticidade de tais sistemas. Luo et al. [Luo et al. 2013] apresenta um algoritmo de gestão de recursos que considera tanto requisitos de consumo de energia como QoS (Qualidade de Serviço). O artigo apresenta um modelo para prever o consumo de energia dentro de uma única máquina, além de uma estrutura simulada para avaliar algoritmos de escalonamento de recursos que leva em consideração o consumo de energia. Os autores afirmam que na maioria dos estudos de energia de computação em nuvem existentes são utilizados modelos lineares para estimar o consumo de energia, descrevendo a relação entre consumo de energia e utilização de recursos. Garg et al. [Garg et al. 2011] apresenta um modelo de energia do *data center* com base nos dados de CPU. O modelo apresentado considera todas as CPUs no *data center* sem considerar a variação dos recursos disponíveis para a aplicação. Com relação a métricas específicas para estimar o consumo de energia, em [Zikos and Karatza 2011], os autores utilizam a seguinte equação para medir a energia: $E = P \times T$. A quantidade de energia utilizada depende da potência e o tempo no qual é utilizada. Assim, E , P e T , denotam energia, potência e tempo, respectivamente. A unidade padrão para a energia é o joule (J), assumindo que a energia é medida em watts (W) e o tempo em segundos (s).

Considerando a análise do consumo de energia, algumas obras focam em definir perfis de energia [Chen et al. 2014], avaliação de custo e desempenho energético [Tesfatsion et al. 2014]. Feifei et al. [Chen et al. 2014] propõe a StressCloud: uma ferramenta de análise de desempenho e consumo de energia e análise de sistemas em nuvem. Os resultados experimentais demonstram a relação entre o desempenho e o consumo de energia dos sistemas de nuvem com diferentes estratégias de alocação de recursos e cargas de trabalho. No entanto, os autores não abordam nem aplicações paralelas nem elasticidade em nuvem.

Finalmente, Tesfatsion et al. [Tesfatsion et al. 2014] realizar uma análise conjunta de custo e desempenho energético utilizando técnicas como DVFS (*Dynamic Voltage and Frequency Scaling*), a elasticidade horizontal e vertical. Esta abordagem combinada resultou em 34% de economia de energia em comparação com cenários onde cada política é aplicada sozinha.

Em relação ao consumo de energia, o método tradicional que leva em conta o consumo instantâneo e o tempo é normalmente utilizado. Desta forma, destaca-se o seguinte a respeito das métricas de avaliação: (i) a avaliação do consumo de energia, considerando um número maleável de recursos; (ii) em ambientes elásticos, há uma falta de análise conjunta do consumo de energia e a utilização de recursos para definir os valores para os limites de *thresholds* inferiores e superiores.

5. Conclusão

Este artigo apresentou e avaliou um modelo elástico de consumo de energia para *data centers* de computação em nuvem. O modelo proposto estima o consumo de energia com base em amostras de CPU e memória com precisão média e mediana 97,15% e 97,72%, respectivamente. Este modelo foi utilizado em conjunto com o *middleware* AutoElastic, que executa aplicativos HPC, alocando e desalocando recursos de acordo com as demandas dos processos. Os resultados mostraram que os melhores valores para economia de

energia foram obtidos quando se utiliza um limite superior (*threshold*) de cerca de 90%, e os piores valores para essa métrica quando se utiliza 70%. Entretanto, neste último caso obteve-se o melhor desempenho. Focando na reproduzibilidade dos resultados, introduzimos um conjunto de equações que permite que outros pesquisadores possam empregar o modelo energético proposto para medir o consumo de energia em suas aplicações elásticas. Por fim, esta pesquisa deve seguir com a extensão do modelo proposto para incluir máquinas heterogêneas, uma vez que a versão atual assume apenas os nós computacionais e máquinas virtuais com a mesma configuração, e avaliação de consumo energético de *middlewares* para Internet das Coisas.

Referências

- Chen, F., Grundy, J., Schneider, J.-G., Yang, Y., and He, Q. (2014). Automated analysis of performance and energy consumption for cloud applications. In *Proceedings of the 5th ACM/SPEC International Conference on Performance Engineering*, ICPE '14, pages 39–50, New York, NY, USA. ACM.
- Garg, S. K., Yeo, C. S., Anandasivam, A., and Buyya, R. (2011). Environment-conscious scheduling of hpc applications on distributed cloud-oriented data centers. *J. Parallel Distrib. Comput.*, 71(6):732–749.
- Lorido-Botran, T., Miguel-Alonso, J., and Lozano, J. (2014). A review of auto-scaling techniques for elastic applications in cloud environments. *Journal of Grid Computing*, 12(4):559–592.
- Luo, L., Wu, W., Tsai, W., Di, D., and Zhang, F. (2013). Simulation of power consumption of cloud data centers. *Simulation Modelling Practice and Theory*, 39(0):152 – 171. S.I.Energy efficiency in grids and clouds.
- Orgerie, A.-C., Assuncao, M. D. D., and Lefevre, L. (2014). A survey on techniques for improving the energy efficiency of large-scale distributed systems. *ACM Computing Surveys*, 46(4):1–31.
- Petrides, P., Nicolaides, G., and Trancoso, P. (2012). Hpc performance domains on multi-core processors with virtualization. In *Proceedings of the 25th International Conference on Architecture of Computing Systems*, ARCS'12, pages 123–134, Berlin, Heidelberg. Springer-Verlag.
- Righi, R., Rodrigues, V., Andre daCosta, C., Galante, G., Bona, L., and Ferreto, T. (2015). Autoelastic: Automatic resource elasticity for high performance applications in the cloud. *Cloud Computing, IEEE Transactions on*, PP(99):1–1.
- Tesfatsion, S., Wadbro, E., and Tordsson, J. (2014). A combined frequency scaling and application elasticity approach for energy-efficient cloud computing. *Sustainable Computing: Informatics and Systems*, 4(4):205 – 214. Special Issue on Energy Aware Resource Management and Scheduling (EARMS).
- Zikos, S. and Karatza, H. D. (2011). Performance and energy aware cluster-level scheduling of compute-intensive jobs with unknown service times. *Simulation Modelling Practice and Theory*, 19(1):239 – 250. Modeling and Performance Analysis of Networking and Collaborative Systems.

Elasticidade Assíncrona: Transferência não Bloqueante de VMs para Viabilizar a Reorganização de Aplicações HPC em Cloud Computing

Vinicius F. Rodrigues¹, Gustavo Rostiolla¹, Rodrigo da R. Righi¹

¹Programa Interdisciplinar de Pós-Graduação em Computação Aplicada (PIPCA)
Universidade do Vale do Rio dos Sinos (UNISINOS)

viniciusfacco@live.com, grostiolla1@gmail.com, rrrighi@unisinos.br

Resumo. A elasticidade é uma das principais funcionalidades de ambientes de computação em nuvem. Considerando a área de computação de alto desempenho (HPC), aplicações possuem dificuldade para usufruir deste recurso sendo necessárias alterações no código fonte da aplicação para tratar a elasticidade, impondo sobrecarga na aplicação. Neste contexto, este artigo apresenta um modelo de elasticidade assíncrona em nuvem para aplicações HPC iterativas em que as operações de elasticidade são executadas simultaneamente à execução da aplicação não impondo bloqueios ou modificações de código fonte.

Abstract. Elasticity is one of the main characteristics of cloud computing. Regarding the high performance computing (HPC) area, it is difficult for applications deal with dynamic environments, where it is necessary to modify the application code to treat the elasticity which imposes overhead on application for these tasks. In this context, this paper presents an asynchronous elasticity cloud model for iterative HPC applications where elasticity operations are performed simultaneously with the application execution not imposing blocking or source code modifications.

1. Introdução

Uma das principais funcionalidades de ambientes de computação em nuvem é a elasticidade, em que usuários podem reorganizar a disponibilidade de recursos a qualquer momento de acordo com a demanda ou necessidade de desempenho [Lorido-Botran et al. 2014, Raveendran et al. 2011]. Considerando o escopo de aplicações de alto desempenho (HPC) e aplicações paralelas de longa duração, um usuário pode querer aumentar a quantidade de recursos disponíveis com o objetivo de reduzir o tempo total de execução da aplicação. Apesar de transparente para o usuário, esse tipo de mecanismo é apropriado para aplicações fracamente acopladas em que réplicas não estabelecem comunicação entre si e a distribuição de tarefas é realizada por um平衡ador de carga [Galante and Bona 2012, Jennings and Stadler 2014]. Devido a isso, a elasticidade em nuvem é mais explorada em arquiteturas WEB cliente-servidor, como vídeo sob demanda, lojas online, aplicações BOINC, e-governance e serviços WEB [Raveendran et al. 2011]. Embora pertinente para aplicações do estilo sacola-de-tarefas, técnicas de replicação e balanceadores de carga centralizado não são funcionais por padrão para implementar elasticidade em aplicações HPC fortemente acopladas, como aquelas modeladas como *Bulk-Synchronous Parallel* (BSP), divisão-e-conquista ou pipeline [Raveendran et al. 2011, Frincu et al. 2013].

Grande parte das aplicações paralelas são desenvolvidas utilizando *Message Passing Interface* (MPI) 1, em que não há suporte para alteração da quantidade de processos durante a execução [Wilkinson and Allen 2005]. Por outro lado, com a segunda versão de MPI, em que há suporte para a dinamicidade de processos, é necessário um esforço significante em nível de aplicação para manualmente alterar o grupo de processos e redistribuir os dados para efetivamente utilizar uma quantidade de processos diferente e tirar proveito da elasticidade. Abordagens que requerem a reescrita do código impõem sobrecarga na aplicação, a qual, além de realizar tarefas de monitoramento, deve coordenar a reorganização de recursos. Neste contexto, com o objetivo de oferecer elasticidade em nuvem para aplicações HPC de maneira eficiente e transparente, é proposto um modelo de elasticidade assíncrona viabilizado através do modelo de elasticidade AutoElastic [Righi et al. 2015]. A contribuição de AutoElastic conta com o conceito de elasticidade assíncrona: reorganização transparente de recursos e processos pela perspectiva do usuário, não bloqueando ou finalizando a execução da aplicação em qualquer ação de alocação ou desalocação de recursos. Para realizar isso, AutoElastic oferece um *framework* com um controlador que transparentemente gerencia ações de elasticidade horizontal, não sendo necessárias modificações na aplicação.

2. Trabalhos Relacionados

Iniciativas de pesquisa acadêmica procuram reduzir lacunas e/ou aprimorar as abordagens de elasticidade em nuvem. ElasticMPI propõe elasticidade em aplicações MPI através da abordagem *stop-reconfigure-and-go* [Raveendran et al. 2011]. Tal ação pode ter um impacto negativo, especialmente para aplicações HPC que não têm longa duração. Em adição, a abordagem de ElasticMPI necessita alteração no código fonte da aplicação a fim de inserir políticas de monitoramento. Mao, Li e Humphrey [Mao et al. 2010] tratam com auto-escalabilidade alterando o número de VMs baseando-se em informações da carga de trabalho. Considerando que o programa possui tempo determinado para concluir cada uma de suas fases, a proposta trabalha com recursos e VMs para cumprir esses tempos corretamente. Martin et al. [Martin et al. 2011] apresentam um cenário típico de requisições a um serviço de nuvem que trabalha com um平衡ador de carga. A elasticidade altera a quantidade de VMs trabalhadoras de acordo com a demanda do serviço. Na mesma abordagem, Elastack aparece como um sistema executando sobre OpenStack para suprir sua falta de elasticidade [Beernaert et al. 2012].

Uma análise do estado da arte em elasticidade permite apontar algumas fraquezas nas iniciativas acadêmicas, que podem ser sumarizadas em cinco aspectos: (i) nenhuma estratégia propõe a avaliar se é um pico, quando atinge um *threshold* [Martin et al. 2011, Beernaert et al. 2012]; (ii) necessidade de alteração no código fonte da aplicação [Raveendran et al. 2011, Rajan et al. 2011]; (iii) necessidade de conhecer previamente dados da aplicação antes de sua execução, como tempo de execução esperado para cada componente [Raveendran et al. 2011]; e (iv) necessidade de reconfigurar recursos com a parada da aplicação e subsequente recuperação [Raveendran et al. 2011]. Observando os trabalhos mencionados, é possível destacar três deles que focam elasticidade em nuvem para aplicações HPC [Raveendran et al. 2011, Martin et al. 2011, Rajan et al. 2011]. Eles têm em comum a abordagem do modelo de programação mestre-escravo. Particularmente, as iniciativas [Raveendran et al. 2011] e [Rajan et al. 2011] são baseadas em aplicações iterativas,

onde há uma redistribuição de tarefas pelo processo mestre a cada nova fase. Aplicações que não possuem um laço iterativo não podem ser adaptadas para essa abordagem, pois ele usa o identificador da iteração como um ponto de reinício da execução. Em adição, a elasticidade em [Rajan et al. 2011] é gerenciada manualmente pelo usuário, obtendo dados de monitoramento utilizando o *framework* proposto pelos autores. Por fim, a proposta de solução de Martin et al. [Martin et al. 2011] é o tratamento eficaz das requisições de um servidor Web através da criação e consolidação de instâncias baseando-se no fluxo de requisições e carga das VMs trabalhadoras.

3. Modelo de Elasticidade Assíncrona

AutoElastic é um modelo de elasticidade em nuvem que opera no nível PaaS de uma plataforma de nuvem, agindo como um *middleware* que permite a transformação de uma aplicação paralela não elástica em uma elástica. O modelo funciona com elasticidade automática e reativa baseada em *thresholds* de forma horizontal, proporcionando alocação e consolidação de nós computacionais e máquinas virtuais. Como uma proposta PaaS, AutoElastic propõe um *middleware* para compilar uma aplicação mestre-escravo iterativa, além de um gerenciador de elasticidade. O Gerenciador esconde do usuário os detalhes da escrita de regras e ações de elasticidade. A Figura 1 apresenta a arquitetura do modelo, ilustrando a relação entre os processos, máquinas virtuais e nós computacionais. Levando em consideração que aplicações HPC são comumente intensivas quanto a CPU, optou-se pela criação de um único processo por VM e c VMs por nó de computação buscando explorar uma melhor eficiência em aplicações paralelas. Dessa maneira, uma nuvem AutoElastic é modelada com m nós homogêneos, os quais possuem c máquinas virtuais cada, em que c é a quantidade de núcleos de processamento de cada nó. Por fim, a qualquer momento, o número de VMs executando processos escravos é igual a $n = c \times m$.

O fato de que o Gerenciador AutoElastic, e não a aplicação por si, realiza as operações de reconfiguração de recursos oferece o benefício da elasticidade assíncrona. Elasticidade assíncrona é uma maneira de assincronamente notificar uma aplicação que está executando em nuvem sobre mudanças na disponibilidade de recursos do ambiente, tais como o número de instâncias de máquinas virtuais. Por exemplo, a aplicação é notificada assim que uma nova instância de uma máquina virtual está disponível no ambiente, sem prejudicar seu fluxo de execução normal. No entanto, esta operação não-

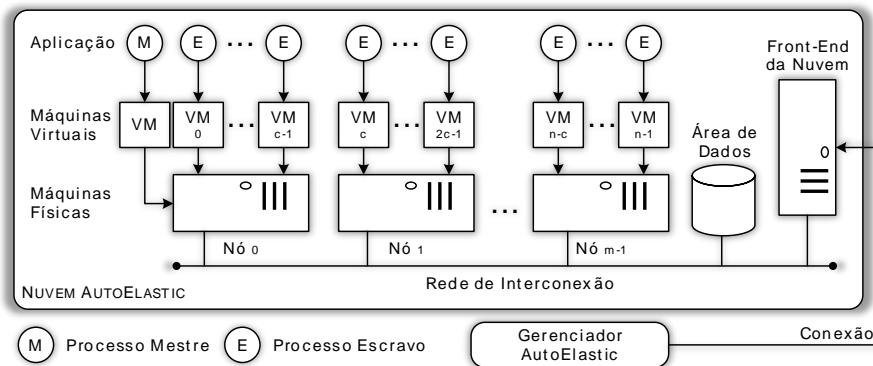


Figura 1. Distribuição de nós, VMs e processos em uma infraestrutura de nuvem AutoElastic, na qual cada VM engloba um único processo da aplicação e cada nó executa c VMs, em que c denota o número de núcleos de processamento do nó.

bloqueante implica na seguinte pergunta: *Como podemos notificar a aplicação sobre a reconfiguração de recursos?* Para isso, AutoElastic oferece a comunicação entre as máquinas virtuais e o Gerenciador AutoElastic utilizando uma área de memória compartilhada. Essa é uma área privada para as máquinas virtuais e as máquinas físicas dentro do ambiente da nuvem AutoElastic. A comunicação entre as máquinas virtuais e o Gerenciador AutoElastic através dessa área pode ser viabilizada, por exemplo, via NFS, *middleware* orientado a mensagens (tais como JMS ou AMQP) ou espaço de tuplas (como JavaSpaces). O uso de uma área compartilhada para interação de dados entre máquinas virtuais é uma abordagem comum em nuvens privadas [Cai et al. 2012].

A Tabela 1 apresenta as Notificações suportadas por AutoElastic. Baseado na Notificação 1, os processos correntes podem iniciar trabalhando com a nova configuração de recursos (um único nó com c VMs, cada uma com um novo processo). A Notificação 2 é relevante pelas seguintes razões: (i) não parando a execução do processo enquanto procedimentos ou de comunicação ou de computação estão ocorrendo; (ii) garantindo que a aplicação não será abortada com a súbita interrupção de um ou mais processos. A Notificação 3 é tomada pelo processo mestre, que garante que a aplicação está em um estado global consistente em que processos podem ser desconectados apropriadamente.

Tabela 1. Notificações fornecidas através da área de dados compartilhada.

Notificação	Direção	Descrição
Notificação 1	Gerenciador AutoElastic → Processo Mestre	Disponível c novos recursos.
Notificação 2	Gerenciador AutoElastic → Processo Mestre	Solicitação de permissão para consolidação de recursos.
Notificação 3	Processo Mestre → Gerenciador AutoElastic	Resposta à Notificação 2 permitindo a consolidação.

Como em Imai et al. [Imai et al. 2012], o monitoramento de dados é realizado de forma periódica. Assim, o Gerente AutoElastic obtém a métrica CPU, aplica séries temporais baseando-se em valores passados e compara a métrica final com os *thresholds* superior e inferior. Mais precisamente, é empregada a técnica de Média Móvel de acordo com a Equação 1. $PC(i)$ retorna uma previsão de carga de CPU quando considerando a execução de n VMs com processos escravos na observação número i do Gerente. Para realizar isso, $MM(i, j)$ informa a carga de CPU de uma máquina virtual j na observação i . $MM(i, j)$ calcula a média móvel considerando as z observações mais recentes da carga de CPU $Carga(k, j)$ da VM j , em que $i - z \leq k \leq i$. Por fim, a Notificação 1 é disparada se PC for maior que o *threshold* superior, enquanto a Notificação 2 é acionada quando PC for menor que o *threshold* inferior.

$$PC(i) = \frac{1}{n} \times \sum_{j=0}^{n-1} MM(i, j) \quad MM(i, j) = \frac{\sum_{k=i-z+1}^i Carga(k, j)}{z} \quad (1)$$

As aplicações paralelas de AutoElastic são projetadas segundo o modelo MPMD (*Multiple Program Multiple Data*), no qual o mestre tem um executável e os escravos outro. Baseado em MPI 2.0, AutoElastic trabalha com as seguintes diretivas: (i) publicar uma porta de conexão; (ii) procurar o servidor a partir de uma porta; (iii) aceitar uma conexão; (iv) requisitar uma conexão e; (v) realizar uma desconexão. O modelo proposto atua segundo a abordagem de MPI 2.0 para o gerenciamento dinâmico de processos: comunicação ponto-a-ponto com conexão e desconexão no estilo de Sockets. O lançamento de uma VM acarreta automaticamente na execução de um processo escravo, que requisita uma conexão com o mestre. Uma aplicação com AutoElastic não necessita seguir a interface MPI 2.0, mas a semântica de cada diretiva mencionada. A

transformação de uma aplicação não elástica em uma elástica pode ser oferecida através de diferentes caminhos, todos transparentes para os usuários: (i) implementação de um programa orientado a objetos utilizando polimorfismo para sobreescriver o método para gerir a elasticidade; (ii) utilizando um tradutor de fonte-para-fonte para inserir código após a diretiva `i` do código do mestre; (iii) desenvolvimento de um *wrapper* em linguagens procedurais para alterar a função da diretiva `i`.

4. Metodologia de Avaliação

Foi configurado uma nuvem privada, utilizando o ambiente OpenNebula 4.2 em um *cluster* com 10 nós homogêneos, para a execução de uma aplicação que calcula a aproximação para a integral do polinômio $f(x)$ num intervalo fechado $[a, b]$. Para tal, foi implementado o método de Newton-Cotes para intervalos fechados conhecido como Regra do Trapézio Repetida [Comanescu 2012]. Considere a partição do intervalo $[a, b]$ em s subintervalos iguais, cada qual de comprimento h ($[x_i, x_{i+1}]$, para $i = 0, 1, 2, \dots, s - 1$). Assim, $x_{i+1} - x_i = h = \frac{b-a}{s}$. Dessa forma, pode-se escrever a integral de $f(x)$ como sendo a soma das áreas dos s trapézios contidos dentro do intervalo $[a, b]$. Sendo assim, existem $s + 1$ equações $f(x)$ simples para se obter o resultado final da integral numérica. O processo mestre deve distribuir essas $s + 1$ equações entre os processos escravos. Como s define a quantidade de sub-intervalos, e consequentemente a quantidade de equações, seu valor define a carga computacional necessária para atingir o resultado final para uma equação em particular. Utilizando esse parâmetro, foram modelados quatro padrões de carga conforme demonstrado na Figura 2.

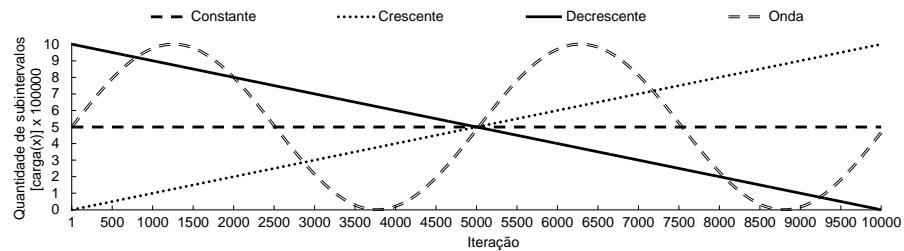


Figura 2. Modelos de padrões de carga. O eixo x expressa a iteração (cada iteração representa uma equação que será calculada), enquanto o eixo y representa a respectiva carga de processamento para aquela iteração (valor de s).

A aplicação foi executada com cada padrão de carga em diferentes cenários com a elasticidade habilitada, variando a configuração de *thresholds*. Assumindo as escolhas de diferentes trabalhos, nos quais podem ser encontrados *thresholds* superiores como 50% [Al-Haidari et al. 2013], 70% [Dawoud et al. 2011, Al-Haidari et al. 2013], 75% [Imai et al. 2012], 80% [Suleiman 2012, Al-Haidari et al. 2013] e 90% [Beernaert et al. 2012, Al-Haidari et al. 2013], foram adotados 70%, 75%, 80%, 85% e 90%, enquanto que para os *thresholds* inferiores foram adotados 30%, 35%, 40%, 45% e 50% baseando-se no trabalho de Haidari et al. [Al-Haidari et al. 2013].

5. Análise de Resultados

A aplicação foi executada com cada padrão de carga em diferentes configurações de *thresholds*, resultando em diferentes necessidades de reconfiguração de recursos. Durante a realização dos testes, foram realizadas 20 operações para a carga Constante, 50 para

a carga Crescente, 40 para a carga Decrescente e 83 para a carga Onda, totalizando 193 operações de alocação de recursos. Para cada operação, foi obtido o tempo total entre o momento em que os recursos são alocados no ambiente e o momento em que os novos recursos são disponibilizados para o uso da aplicação. Das 193 operações, o tempo médio obtido foi de 152,71 segundos, com uma mediana 148 segundos e um desvio padrão de 6,99 segundos. Durante todo o tempo de execução destas operações, a aplicação continuou executando simultaneamente sem ser afetada pelas tarefas relacionadas à alocação de recursos. A Tabela 2 apresenta o tempo total de execução da aplicação para cada uma das execuções, além de apresentar a quantidade total de operações de alocação de recursos que foram realizadas em cada execução e o tempo total dessas operações.

Tabela 2. Resultados obtidos em todos os cenários com todos os padrões de carga. Tempos estão em segundos e o valor entre parênteses do campo Tempo Alocação representa a quantidade de alocações realizadas.

Thresholds	Superior	Inferior	Constante		Crescente		Decrescente		Onda	
			Tempo Execução	Tempo Alocação						
70	30	1569	310 (2)	1578	602 (4)	1602	458 (3)	1730	759 (5)	
	35	1569	310 (2)	1578	602 (4)	1609	457 (3)	1733	753 (5)	
	40	1569	310 (2)	1578	602 (4)	1606	448 (3)	1742	740 (5)	
	45	1569	310 (2)	1578	602 (4)	1596	458 (3)	1829	886 (6)	
	50	1569	310 (2)	1578	602 (4)	1580	441 (3)	1835	904 (6)	
75	30	1799	162 (1)	1776	459 (3)	1689	327 (2)	1755	752 (5)	
	35	1799	162 (1)	1776	459 (3)	1684	295 (2)	1854	756 (5)	
	40	1799	162 (1)	1776	459 (3)	1680	311 (2)	1751	754 (5)	
	45	1799	162 (1)	1776	459 (3)	1675	312 (2)	1864	737 (5)	
	50	1799	162 (1)	1776	459 (3)	1659	310 (2)	1853	900 (6)	
80	30	1781	162 (1)	1933	309 (2)	1679	310 (2)	1907	591 (4)	
	35	1781	162 (1)	1933	309 (2)	1680	310 (2)	1877	604 (4)	
	40	1781	162 (1)	1933	309 (2)	1675	310 (2)	1892	606 (4)	
	45	1781	162 (1)	1933	309 (2)	1673	312 (2)	1921	622 (4)	
	50	1781	162 (1)	1933	309 (2)	1662	312 (2)	1891	594 (4)	
85	30	2305	0	2137	162 (1)	1872	164 (1)	2118	310 (2)	
	35	2305	0	2137	162 (1)	1860	162 (1)	2097	310 (2)	
	40	2305	0	2137	162 (1)	1857	163 (1)	2080	296 (2)	
	45	2305	0	2137	162 (1)	1859	162 (1)	2031	294 (2)	
	50	2305	0	2137	162 (1)	1850	146 (1)	2082	308 (2)	
90	TODOS		2297	0	2348	0	2334	0	2383	0

Através da Tabela 2 é possível identificar que quanto menor é o valor para o *threshold* superior, maior é a quantidade de operações de alocação necessárias. Isso se deve ao fato de que, com um valor baixo para esse *threshold*, a carga de processamento atinge o nível do *threshold* de maneira mais rápida, resultando em um número maior de operações necessárias. Além disso, a execução da aplicação com a carga Onda resultou em um maior número de operações devido ao seu comportamento variável em que são desalocados recursos em momentos de carga baixa e alocados novamente quando a carga de processamento volta a crescer. Uma maior quantidade de alocações resulta em maior tempo em que recursos estão sendo alocados em simultâneo com a aplicação. Graças a elasticidade assíncrona, esses tempos são escondidos da aplicação que não sofre nenhum bloqueio enquanto as operações estão ocorrendo. As execuções com o *threshold* superior 70 foram responsáveis pela maior parte das operações de alocação de recursos totalizando 73 das 193 operações. Considerando essas execuções com o *threshold* superior 70, para a carga Constante o tempo de alocação equivaleu em média a 19,76% do tempo total de execução da aplicação. Já para a carga Crescente, esta equivalência atingiu o valor médio de 38,15% enquanto que para a carga Decrescente esse valor médio foi de 28,30%. Os

maiores valores obtidos pela carga Onda em que o tempo de alocação de recursos equivaleu em média a 45,5% do tempo total de execução da aplicação.

Por fim, em todas as execuções com o *threshold* superior 90 não foram realizadas operações pois o valor de 90% não foi atingido em nenhum momento. Devido a isso, os tempos de execução com esse *threshold* equivalem ao tempo de execução da aplicação sem elasticidade. Com isso, é possível notar que o desempenho das execuções em que houveram alocações é melhor em comparação com a execução sem elasticidade.

6. Conclusão

Este artigo apresentou um modelo de elasticidade assíncrona horizontal em nuvem para aplicações HPC iterativas chamado AutoElastic. Apesar de que Spinner et al. [Spinner et al. 2014] afirmarem que apenas elasticidade vertical é adequada para cenários HPC devido a sobrecarga nas operações de reorganização de recursos, com a elasticidade assíncrona, torna-se viável a utilização da elasticidade horizontal sem impactar na execução da aplicação. Dessa maneira, a execução da aplicação e as ações de elasticidade ocorrem simultaneamente, não penalizando a aplicação com a sobrecarga da reconfiguração de recursos. AutoElastic é ciente quanto à sobrecarga para instanciar uma máquina virtual, escondendo da aplicação essa penalização. Ainda, AutoElastic conta com um *middleware* e um gerenciador não havendo necessidade de modificações no código fonte da aplicação para tratar o ambiente dinâmico. Resultados demonstraram que em alguns cenários em que muitas operações de elasticidade são executadas, o tempo total de alocação de recursos pode chegar a 49,26% do tempo total de execução da aplicação, dependendo do quanto longa é a execução da aplicação.

Futuramente, planeja-se estender o trabalho para diferentes modelos de elasticidade com o uso de *thresholds* dinâmicos, além de abordar diferentes modelos de programação como BSP e pipeline.

Referências

- Al-Haidari, F., Sqalli, M., and Salah, K. (2013). Impact of cpu utilization thresholds and scaling size on autoscaling cloud resources. In *Cloud Comp. Technology and Science (CloudCom), 2013 IEEE 5th Int. Conf. on*, volume 2, pages 256–261.
- Beernaert, L., Matos, M., Vilaça, R., and Oliveira, R. (2012). Automatic elasticity in openstack. In *Proc. of the Workshop on Secure and Dependable Middleware for Cloud Monitoring and Manag.*, SDMCM ’12, pages 2:1–2:6, New York, NY, USA. ACM.
- Cai, B., Xu, F., Ye, F., and Zhou, W. (2012). Research and application of migrating legacy systems to the private cloud platform with cloudstack. In *Automation and Logistics (ICAL), 2012 IEEE Int. Conf. on*, pages 400–404.
- Comanescu, M. (2012). Implementation of time-varying observers used in direct field orientation of motor drives by trapezoidal integration. In *Power Electronics, Machines and Drives (PEMD 2012), 6th IET Int. Conf. on*, pages 1–6.
- Dawoud, W., Takouna, I., and Meinel, C. (2011). Elastic vm for cloud resources provisioning optimization. In Abraham, A., Lloret Mauri, J., Buford, J., Suzuki, J., and Thampi, S., editors, *Advances in Comp. and Communications*, volume 190 of *Communications in Computer and Information Science*, pages 431–445. Springer Berlin Heidelberg.

- Frincu, M. E., Genaud, S., and Gossa, J. (2013). Comparing provisioning and scheduling strategies for workflows on clouds. In *Proc. of the 2013 IEEE 27th Int. Symposium on Parallel and Distributed Processing Workshops and PhD Forum, IPDPSW '13*, pages 2101–2110, Washington, DC, USA. IEEE Computer Society.
- Galante, G. and Bona, L. C. E. d. (2012). A survey on cloud computing elasticity. In *Proc. of the 2012 IEEE/ACM Fifth Int. Conf. on Utility and Cloud Comp., UCC '12*, pages 263–270, Washington, DC, USA. IEEE Computer Society.
- Imai, S., Chestna, T., and Varela, C. A. (2012). Elastic scalable cloud computing using application-level migration. In *Proc. of the 2012 IEEE/ACM Fifth Int. Conf. on Utility and Cloud Comp., UCC '12*, pages 91–98, Washington, DC, USA. IEEE Computer Society.
- Jennings, B. and Stadler, R. (2014). Resource management in clouds: Survey and research challenges. *Journal of Network and Sys. Manag.*, pages 1–53.
- Lorido-Botran, T., Miguel-Alonso, J., and Lozano, J. (2014). A review of auto-scaling techniques for elastic applications in cloud environments. *Journal of Grid Comp.*, 12(4):559–592.
- Mao, M., Li, J., and Humphrey, M. (2010). Cloud auto-scaling with deadline and budget constraints. In *Grid Comp. (GRID), 2010 11th IEEE/ACM Int. Conf. on*, pages 41 –48.
- Martin, P., Brown, A., Powley, W., and Vazquez-Poletti, J. L. (2011). Autonomic management of elastic services in the cloud. In *Proc. of the 2011 IEEE Symposium on Computers and Communications, ISCC '11*, pages 135–140, Washington, DC, USA. IEEE Computer Society.
- Rajan, D., Canino, A., Izaguirre, J. A., and Thain, D. (2011). Converting a high performance application to an elastic cloud application. In *Proc. of the 2011 IEEE Third Int. Conf. on Cloud Comp. Technology and Science, CLOUDCOM '11*, pages 383–390, Washington, DC, USA. IEEE Computer Society.
- Raveendran, A., Bicer, T., and Agrawal, G. (2011). A framework for elastic execution of existing mpi programs. In *Proc. of the 2011 IEEE Int. Symposium on Parallel and Distributed Processing Workshops and PhD Forum, IPDPSW '11*, pages 940–947, Washington, DC, USA. IEEE Computer Society.
- Righi, R., Rodrigues, V., Andre daCosta, C., Galante, G., Bona, L., and Ferreto, T. (2015). Autoelastic: Automatic resource elasticity for high performance applications in the cloud. *Cloud Comp., IEEE Transactions on*, PP(99):1–1.
- Spinner, S., Kounev, S., Zhu, X., Lu, L., Uysal, M., Holler, A., and Griffith, R. (2014). Runtime Vertical Scaling of Virtualized Applications via Online Model Estimation. In *Proc. of the 2014 IEEE 8th Int. Conf. on Self-Adaptive and Self-Organ. Sys. (SASO)*.
- Suleiman, B. (2012). Elasticity economics of cloud-based applications. In *Proc. of the 2012 IEEE Ninth Int. Conf. on Services Comp., SCC '12*, pages 694–695, Washington, DC, USA. IEEE Computer Society.
- Wilkinson, B. and Allen, C. (2005). *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers*. An Alan R. Apt book. Pearson/Prentice Hall.

Analizando a Camada de Gerenciamento das Ferramentas CloudStack e OpenStack para Nuvens Privadas

**Demétrius Roveda¹, Adriano Vogel¹, Carlos A. F. Maron²,
Dalvan Griebler^{1,2}, Claudio Schepke³**

¹Faculdade Três de Maio (SETREM), Laboratório de Pesquisas Avançadas para Computação em Nuvem (LARCC) – Três de Maio – RS – Brasil

²Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Faculdade de Informática, Programa de Pós-Graduação em Ciência da Computação, Porto Alegre – RS – Brasil

³Universidade Federal do Pampa (UNIPAMPA)
Laboratório de Estudos Avançados (LEA) – Alegrete – RS – Brasil

{roveda.demetrius,adrianovogel03}@gmail.com, {carlos.maron,
dalvan.griebler}@acad.pucrs.br, claudioschepke@unipampa.edu.br

Resumo. A camada de gerenciamento é um dos elementos mais importantes para o modelo de serviço IaaS nas ferramentas de administração de nuvem privada. Isso porque oferece aos usuários/clientes os recursos de infraestrutura sob-demanda e controla questões administrativas da nuvem. Nesse artigo, o objetivo é realizar uma análise da interface de gerenciamento das ferramentas CloudStack e OpenStack. Com o estudo realizado, constatou-se que as ferramentas tem gerenciamento distinto. No entanto, OpenStack se mostrou mais robusto e complexo, enquanto CloudStack é mais centralizado e possui uma interface gráfica mais completa e intuitiva.

1. Introdução

As ferramentas de computação em nuvem de infraestrutura como serviço (IaaS) oferecem abstração de recursos computacionais às camadas superiores: plataforma como serviço (PaaS) e software como serviço (SaaS)[Chandrasekaran 2014]. O modelo de serviço IaaS é extremamente importante para a nuvem e vem recebendo atenção de organizações e centros de pesquisa, motivados pelas vantagens oferecidas: redução de custos, baixo investimento inicial, flexibilidade e elasticidade [Buyya et al. 2013]. A camada de gerenciamento no modelo de serviço IaaS é vital para um ambiente de nuvem. Esta é responsável por gerenciar e fornecer recursos computacionais, tanto para as camadas superiores, como para usuários de infraestrutura como serviço em nuvem.

Sabendo-se das vantagens da utilização de ferramentas para gerenciar nuvens, torna-se atrativo para o meio científico e corporativo explorar tais soluções. Porém, não são encontrados trabalhos que analisam a interface de gerenciamento das ferramentas de IaaS. Isso ocorre porque o foco da maioria das pesquisas ainda se encontra em temas correlacionados, tais como arquitetura, desempenho, serviços oferecidos, entre outros. Diante disso, esse artigo explora as funcionalidades do gerenciamento nas soluções de nuvem com viés para o suporte nas interações de usuários e administradores. Sendo assim, as **principais contribuições** são listadas abaixo:

- Uma extensão da metodologia de estudo sobre a camada de gerenciamento proposta por [Dukaric and Juric 2013], detalhando o nível de controle do usuário e do administrador sobre itens da *dashboard* das ferramentas.

- Uma análise detalhada dos elementos de gerenciamento e flexibilidade das ferramentas CloudStack e OpenStack.

O artigo está organizado em 5 seções. Na Seção 2 (Ferramentas de gerenciamento de IaaS) é apresentada uma breve descrição das ferramentas CloudStack e OpenStack utilizadas na pesquisa. A seção 3 (Camada de Gerenciamento) são descritos itens que fazem parte da camada de gerenciamento. A Seção 4 (Trabalhos Relacionados) relaciona o presente estudo com demais trabalhos encontrados na literatura. A Seção 5 (Resultados) traz as comparações das ferramentas e contribuições deste trabalho. Por fim, na Seção 6 (Conclusões) são realizadas as conclusões do artigo e a indicação dos trabalhos futuros.

2. Ferramentas para Gerenciamento de IaaS

As ferramentas do tipo IaaS são a base para o gerenciamento da infraestrutura (rede, armazenamento, CPU e memória), oferecendo meios para otimizar de forma inteligente a provisão de recursos ao administrador da nuvem (alocação sob demanda). Sendo assim, as camadas superiores (PaaS e SaaS) são extremamente dependentes das ferramentas que atuam no modelo IaaS. Elas transcendem a virtualização, pois diversos fatores (segurança lógica, isolamento de recursos, suporte à usuários etc.) fazem parte de uma ferramenta de gerenciamento de IaaS, enquanto a virtualização é voltada para o *hardware*. Desta forma, a computação em nuvem é utilizada para disponibilizar algumas vantagens: elasticidade, flexibilidade, melhor utilização do *hardware*.

Ferramentas de código aberto são gratuitas. Por outro lado, em demandas específicas, o gerenciamento delas não possuem um alto nível de controle ou recursos avançados (suporte completo para venda de serviços - *pay per use*). Nesses casos, é necessário recorrer a ferramentas de terceiros, específicas, e que oferecem um gerenciamento completo da nuvem [Shroff 2010]. Entre diversas ferramentas de código aberto existentes para o gerenciamento IaaS foram escolhidas CloudStack, pela simplicidade e eficiência e OpenStack pela aceitação corporativa:

- **CloudStack:** é uma ferramenta de gerenciamento de IaaS de código aberto que pode ser utilizada em nuvens privadas, públicas e híbridas [CloudStack 2015]. CloudStack possui uma gama considerável de usuários. A comunidade de usuários é ativa e possui diversos recursos em desenvolvimento. Para instalá-la e configurá-la é simples (se comparada a do OpenStack). Oferece APIs que possibilitam ao administrador e usuários de IaaS gerenciar a infraestrutura computacional. A interface gráfica é organizada e de fácil utilização. Como opção, pode-se utilizar a CLI *cloudmonkey*, uma interface de linha de comando que possibilita à execução das mesmas funções e tarefas da interface gráfica.
- **OpenStack:** é uma ferramenta de gerenciamento de IaaS de código aberto, bastante difundida e utilizada por grandes empresas que investem no desenvolvimento de novos recursos [OpenStack 2015]. Esses empresas são chamados de “stacks”. Por ser uma ferramenta altamente fragmentada, pode alcançar altos níveis de flexibilidade e pode alcançar altos níveis de customização [OpenStack 2015]. Por outro lado, é uma ferramenta que geralmente possui uma implantação mais demorada, devido a sua arquitetura e toda comunicação dos serviços é feita através das APIs [Maron et al. 2014].

3. Camada de Gerenciamento

A proposta do estudo de [Dukaric and Juric 2013] é uma taxonomia dividida em camadas para uma ferramenta de IaaS. A taxonomia relaciona todos os recursos que uma ferra-

menta deve conter para ser considerada robusta. O controle computacional dos recursos (CPU, memória, armazenamento e rede) são feitos através de componentes das ferramentas e devem possuir uma interação que alcance as necessidades do controle dos recursos. Portanto, de acordo com a taxonomia, os principais itens que devem estar presentes em um ferramenta de gerenciamento são:

- *Interface de linha de comando (CLI)*: é uma interface que possibilita ao administrador de IaaS monitorar e gerenciar os componentes e recursos computacionais da nuvem, tais como: máquinas, usuários, grupos, redes e volumes [Petersen 2006].
- *Application Programming Interface (API)*: através dela é possível integrar diversos sistemas e serviços. Atuando como um *middleware*, todas essas interações são transparente ao usuário, e possibilita que aplicações trabalhem de forma unificada, integrando os componentes e ambientes distintos.
- *Dashboard*: É uma interface gráfica que centraliza as interações do administrador e usuário de IaaS com os componentes, APIs e serviços da nuvem. Ela controla e abstrai comandos e parâmetros necessários para gerenciar a infraestrutura através de menus gráficos.
- Orquestrador: a principal função é automatizar tarefas e procedimentos na utilização do ambiente com VMs (criar, atualizar, monitorar e excluir).
- Gerenciamento dos Recursos: Responsável pelo correto funcionamento da infraestrutura computacional, pois o administrador de IaaS deve disponibilizar recursos aos usuários ou aplicações, que podem ser totalmente adversas e com necessidades diferentes.
- Monitoramento: Permite que o administrador e usuários de IaaS consigam visualizar, de uma forma geral, a utilização dos recursos computacionais, sendo relevante para o controle da nuvem.
- Gerenciamento de Incidentes: Incidentes são falhas inesperadas na infraestrutura, e consequentemente nos serviços que ela provê. Esse gerenciamento é a base para que provedores de serviços consigam atender ao SLA estipulado no contrato com o consumidor de nuvem de modelo IaaS.
- Gerenciamento de Aluguel: Oferece aos clientes uma flexibilidade adicional em recursos e serviços de nuvem. Isso torna-se interessante para usuários que precisam de um determinado serviço por um intervalo descontínuo de tempo, tornando possível apenas o aluguel para um período específico e o pagamento apenas pelo período contratado.
- Gerenciamento de Energia: É a forma com que a ferramenta trata o gerenciamento dos recursos para alcançar uma maior eficiência energética. Um exemplo disso é agrupando as VMs em apenas um nodo, enquanto os outros ficam em *stand by*.
- Relatação: É inevitável ocorrer problemas na infraestrutura computacional. Dessa forma, é importante que o administrador de IaaS saiba o que aconteceu, para investigar determinada falha no sistema.
- Acordo de Nível de Serviço (SLA): São contratos de negócio, especificando quais recursos o fornecedor irá prover ao consumidor. Ainda há informações referentes ao desempenho do serviço. Caso o desempenho não seja cumprido pelo fornecedor, o consumidor deve ser resarcido (multa), que é estipulado no contrato.
- Gerenciamento de Elasticidade: Trata-se do gerenciamento da quantidade de recursos que determinado usuário está utilizando. Quando a carga é maior, mais recursos computacionais são disponibilizados dinamicamente. Quando o usuário

não necessita mais dos recursos, estes serão imediatamente distribuídos entre os outros usuários da nuvem.

- Gerenciamento de Federação: A federação de nuvem refere-se a união entre diversos ambientes de nuvem como se fossem uma. Esse paradigma é muito utilizado por clientes com grandes demandas computacionais que precisam gerenciar diversos provedores de nuvem.

A camada de gerenciamento é um dos principais fatores a serem considerados em uma ferramenta de IaaS, pois ela é responsável por gerenciar os recursos computacionais (eg., SLAs, monitoramento, agendamento) e oferecer serviços aos usuários. A próxima seção apresentará os trabalhos relacionados.

4. Trabalhos Relacionados

A seguir são apresentados alguns trabalhos relacionados sobre a análise de ferramentas de gerenciamento de nuvens IaaS. O estudo de [Cocozza et al. 2015] avaliou o gerenciamento que as ferramentas de nuvem exercem na infraestrutura, objetivando a construção de uma nuvem acadêmica. As ferramentas comparadas foram OpenStack, Eucalyptus, CloudStack e *Virtual Computing Lab* (VCL). Com a comparação, os autores optaram por instalar o VCL para oferecer recursos virtuais para os acadêmicos através de laboratórios virtuais, já o OpenStack para a instalação de uma nuvem administrativa.

No meio científico, o estudo de [Dukaric and Juric 2013] propôs uma taxonomia unificada para ferramentas de nuvem IaaS. Tal pesquisa elenca sete camadas conceituais e fundamentais que deveriam estar dentro destas ferramentas. Além disso, algumas soluções de nuvem pública e ferramentas gratuitas de IaaS de nuvem privada foram evidenciadas na taxonomia. Já o estudo de [Thome et al. 2013] realizou uma revisão da literatura sobre das ferramentas de código aberto, apontando as principais tecnologias e serviços. Finalmente, similar ao presente estudo, o trabalho de [Roveda et al. 2015] baseou-se na taxonomia de [Dukaric and Juric 2013] para analisar a camada de gerenciamento das ferramentas OpenStack e OpenNebula, discutindo novos aspectos que estenderam a taxonomia proposta.

Através de uma análise de estudos relacionados, percebe-se aspectos importantes a serem explorados. No estudo de [Thome et al. 2013], não foi considerado APIs, CLIs, interações de usuários bem como de administradores na nuvem. Os estudos de [Dukaric and Juric 2013] e [Cocozza et al. 2015] não avaliaram os recursos contidos nas ferramentas de IaaS com um viés para interações dos usuários e administradores de nuvem. Diferente também, os trabalhos de [Dukaric and Juric 2013] e [Roveda et al. 2015] não consideraram CloudStack em seus experimentos.

5. Resultados

Esta seção apresenta a discussão dos resultados nas comparações entre as ferramentas CloudStack e OpenStack, onde levou-se em conta a flexibilidade, robustez e suporte de cada uma, baseando-se na taxonomia proposta no estudo de [Dukaric and Juric 2013]. Para sumarizar o conhecimento, foram tabuladas as características de cada ferramenta e classificadas com detalhes voltados ao suporte e relevância do gerenciamento de acordo com a documentação oficial de cada uma das ferramentas.

Para a realização das comparações, as ferramentas foram implantadas e analisadas nos itens em que as tabelas se referem, analisando a camada de gerenciamento do ponto

de vista do usuário e administrador de IaaS. Os usuários de IaaS são geralmente administradores de redes e sistemas, pois eles que definem qual é a infraestrutura necessária para rodar suas aplicações. Já os administradores de IaaS, preparam a estrutura, monitoram os recursos computacionais e gerenciam as cotas dos usuários de IaaS. A comparação da interface do usuário entre as ferramentas é representada na Tabela 1.

Tabela 1. Comparação dos elementos da Dashboard.

Dashboard Web(UI)	CloudStack	OpenStack
Acesso Seguro	Encriptação Apache SSL	Secure HTTPS proxy certificate authority (CA)
Autenticação	Usuário/Senha	Usuário/Senha
Autorização	Permissões por Grupos de Segurança	Permissões por Grupos de Segurança
Painel de Notificações	Possui	/
Escolha do hospedeiro para instanciar a VM	/	/
Supor te a <i>Snapshots</i>	Supor ta	Supor ta
Supor te a migração de VMs	Supor ta	Nativo Horizon (utilizando armazenamento compartilhado)

Como evidenciado na Tabela 1, as ferramentas apresentam contrastes no suporte à interações na interface gráfica. Ambas oferecem acesso via web através do servidor que controla a nuvem. Os dois tipos tradicionais de usuários são os administradores e os clientes da nuvem. Na Tabela, semelhanças das ferramentas ficam evidentes nos tópicos de acesso seguro, autenticação, autorização, escolha de *hosts* e *snapshots*. Por outro lado, CloudStack possui um centralizador de notificações na interface gráfica e é mais flexível na migração de VMs. A comparação das ações possíveis como administrador e usuário de IaaS é representada na Tabela 2.

Tabela 2. Controle da interface gráfica. Ações possíveis como administrador e usuário na interface são: Criar (C), Acessar (A), Editar (E), Deletar (D).

Controle de Acesso	CloudStack		OpenStack	
	Administrador	Usuário	Administrador	Usuário
Usuários	C A E D	C A E D	C A E D	- A -
Grupos	C A E D	- A -	C A E D	- A -
ACLs	- A -	- A -	- A -	- A -
Nodos	C A E D	- A -	C A E D	- A -
<i>Clusters/Agregador de Nodos</i>	C A E D	- A -	C A E D	- A -
Discos/Volumes	C A E D	C A E D	C A E D	C A E D
Redes Virtuais	C A E D	- A -	C A E D	C A E D
Zonas de disponibilidade/Grupos de afinidade	C A E D	C A E D	C A E D	- A -
Imagens	C A E D	C A E D	C A E D	- A -
<i>templates/Flavors</i>	C A E D	C A E D	C A E D	- A -
Serviços de Infraestrutura	C A E D	- A -	C A E D	- A -
Cotas	C E	--	C E	--
Monitorar recursos (por nodo e VMS)	A	-	-	-

A ferramenta CloudStack possui uma interface do usuário (UI) organizada e intuitiva, possibilitando um gerenciamento efetivo dos recursos computacionais. A sua arquitetura é diferente da ferramenta OpenStack, sendo mais centralizada e oferece suporte à tarefas avançadas de IaaS pela UI (recuperar VMs, configurações específicas de rede, etc). Um recurso interessante do CloudStack é que o gerenciamento dos recursos da ferramenta é bem granular. Isso torna possível controlar a velocidade do barramento da memória (MHz) e do processador (GHz), onde na ferramenta OpenStack só é possível gerenciar os recursos por quantidade (método convencional em ferramentas de IaaS).

A Tabela 3 apresenta as APIs, CLIs e descrição delas. Nota-se que a ferramenta OpenStack possui diversas APIs, devido a sua arquitetura ter sido projetada assim, o que beneficia a robustez, flexibilidade e suporte de aplicações [Roveda et al. 2015]. Porém, necessita de um esforço maior no controle e comunicação (RabbitMQ) entre os componentes e sistemas.

Tabela 3. APIs e CLIs do OpenStack.

<i>API</i>	<i>Função</i>	<i>CLIs</i>
<i>Block Storage</i>	Gerencia volumes e <i>Snapshots</i> de armazenamento em bloco	Cinder
<i>Compute</i>	Controlar os recursos computacionais(CPU, memória) nas instâncias da nuvem	Nova
<i>Database Service</i>	Gerencia instâncias e serviços de banco de dados	Trove
<i>EC2 compatibility</i>	Oferece suporte para cargas de trabalho serem executadas na nuvem da Amazon	Euca2ools
<i>Identity</i>	Controla a autenticação e autorização	Keystone
<i>Image Service</i>	Controla as permissões dos usuários nas interações com imagens	Glance
<i>Networking</i>	Gerencia as redes do ambiente virtual de nuvem	Neutron
<i>Object Storage</i>	Gerencia o sistema de armazenamento de objetos	Swift
<i>Telemetry</i>	Gerencia o controle da utilização dos recursos	Ceilometer
<i>Orchestration</i>	Idealiza a orquestração na nuvem	Heat
<i>Data Processing</i>	Produz operações de dados (mineração, tratamento, análise, estatísticas)	/

Conclui-se assim que o OpenStack é bem flexível. No entanto, ele possui uma complexa implantação do sistema. Já o CloudStack é totalmente centralizado e o recurso *CloudMonkey* pode ser utilizado como uma CLI ou um *Shell* interativo com ele. Ainda, o *CloudMonkey* foi desenvolvido e é mantido por outra comunidade. CloudStack também possui APIs que são usadas para controlar a infraestrutura virtual. A API CloudStack Root Admin controla as funções do administrador web. O CloudStack *Domain Admin* API é usada para controlar o domínio de nuvem e CloudStack *User API* é integrada para suportar as interações com os usuários.

A Tabela 4 apresenta a resiliência para implantações nas ferramentas. Na comparação entre as ferramentas, é evidenciando que elas suportam resiliência para implantações, inclusive em alguns dos tópicos comparados elas possuem o mesmo suporte. A ferramenta OpenStack é mais flexível nos formatos de discos, suportando todos os tipos que o CloudStack suporta incluindo os formatos RAW e VDI. O suporte de redes das duas ferramentas são amplos e o suporte a sistemas operacionais é parecido, mas o OpenStack oferece suporte distribuições Fedora e Suse, o que não é visto no CloudStack.

Tabela 4. Comparação do suporte e resiliência para implantações.

<i>Suporte</i>	<i>CloudStack</i>	<i>OpenStack</i>
Virtualização	Hyper-V, Xen, KVM, VMware, VirtualBox	Hyper-V, VMware, Xen, KVM, VirtualBox
Tecnologias de armazenamento	NFS, SMB, SolidFire, NetApp, Ceph, LVM	LVM, Ceph, Gluster, NFS, ZFS, Sheepdog
Formatos de discos	LVM, VMDK, VHD, Qcow2	LVM, Qcow2, RAW, VHD, VMDK, VDI
Rede	Bridge, VLAN, DHCP, DNS, NVP, BigSwitch, OVS	Neutron, and B.Switch, Brocade, OVS, NSX, PLUMgrid
Sistema operacional	Debian, Ubuntu, RHEL, CentOS	Debian, Ubuntu, RHEL, CentOS, Fedora, Suse

Em geral, ambas as ferramentas oferecem o serviço de orquestrador para um gerenciamento melhorado dos recursos. O OpenStack utiliza o Heat¹, enquanto o CloudStack oferece o Cookbook². O mesmo ocorre com o gerenciamento de recursos, onde

¹<https://wiki.openstack.org/wiki/Heat>

²<https://github.com/OpenStackCookbook/OpenStackCookbook>

diferentes escalonadores e APIs são usados para de forma eficiente distribuir e balancear a utilização com as demandas.

Em alguns itens da camada de gerenciamento, as soluções *open source* de IaaS são muito pobres ou nem suportam determinados recursos. Exemplos são o gerenciamento de incidentes, de aluguel, SLAs, Federação e relatação. Além desses itens compartilham o limitado suporte, tratam-se de características necessárias para implantação de uma nuvem pública, enquanto que em uma nuvem privada esse itens não desempenhariam funções imprescindíveis. Em uma nuvem pública, a complexidade aumenta e as ferramentas OpenStack e CloudStack podem oferecer serviços adicionais de gerenciamento através da integração de outras aplicações com o *core* das ferramentas.

O gerenciamento de elasticidade ocorre de forma automática nas ferramentas. Na documentação oficial nada é apresentado, pois depende da infraestrutura do ambiente de nuvem. O mesmo acontece com a reportação, federação e gerenciamento de aluguel. O gerenciamento de energia é pobre e merece uma maior atenção devido a sua grande importância. Outro fator relevante é o gerenciamento de recursos, onde o escalonador de VMs irá sempre instanciá-las no servidor que possuir a menor carga. Já o monitoramento na ferramenta CloudStack é mais intuitivo. O OpenStack é mais completo e robusto, pois monitora a infraestrutura como um todo (ISOs, redes virtuais, volumes, etc).

6. Conclusões

A pesquisa realizada teve como objetivo apresentar uma análise do suporte e resiliência em duas ferramentas provedoras de nuvem IaaS (CloudStack e OpenStack), levando em conta a resiliência de integração com aplicações e relevância no gerenciamento. Sendo uma extensão da taxonomia de [Dukaric and Juric 2013], como evidenciado nas tabelas 2, 3 e 5 que analisam as interações com usuários, APIs, CLIs e tecnologias suportadas.

A interface gráfica é o *middleware* entre o usuário de IaaS e os recursos que a ferramenta possui. Usando CloudStack pode-se visualizar a utilização dos recursos do sistema de uma forma simples (primeira tela após o login do usuário). O usuário do CloudStack também consegue editar *templates* (modelo de configuração da VM), consegue visualizar as redes virtuais que tem permissão de acesso e selecionar qual será utilizada. No OpenStack, os usuários pertencem à projetos, e esse grupo de usuários possui acesso total aos volumes e redes virtuais, no CloudStack os usuários também possuem acesso aos volumes, *snapshots*, imagens e zonas de disponibilidade. Com isso, o CloudStack pode ser mais vantajoso na utilização de nuvens privadas ou híbridas. Já o OpenStack, devido a sua estrutura de organização pode ser mais vantajoso em nuvens públicas.

A ferramenta OpenStack é robusta e fragmentada, onde diversos “stacks” (investidores externos, empresas), desenvolvem novos projetos para ser utilizados com ele. Em implantações típicas, já é oferecido recursos avançados. Destaca-se, o componente de rede Neutron que oferece suporte completo em serviços de rede (roteadores virtuais, VLANs, segmentações, etc), as CLIs do OpenStack que possuem recursos mais avançados. Essa gama diversificada de recursos oferecidos, fazem com que os administradores de IaaS necessitem de um conhecimento avançado dos componentes da própria ferramenta de infraestrutura para sua implantação e gerenciamento.

Como trabalhos futuros, o objetivo é investigar com mais detalhes a camada de gerenciamento de nuvens IaaS, adentrando em quesitos como: integração de ferramentas através de APIs com outras tecnologias e o gerenciamento de energia pelas ferramentas.

Além disso, outro potencial estudo é a criação de uma taxonomia unificada para APIs das ferramentas de IaaS.

Agradecimentos

Esta pesquisa foi realizada com o apoio do projeto HiPerfCloud³. Os autores agradecem ao apoio financeiro da Abase Sistemas⁴ e Sociedade Educacional Três de Maio (SETREM)⁵.

Referências

- [Buyya et al. 2013] Buyya, R., Vecchiola, C., and Selvi, S. (2013). *Mastering Cloud Computing: Foundations and Applications Programming*. Mastering Cloud Computing: Foundations and Applications Programming. Elsevier Science.
- [Chandrasekaran 2014] Chandrasekaran, K. (2014). *Essentials of Cloud Computing*. Taylor & Francis.
- [CloudStack 2015] CloudStack (2015). CloudStack (Official Page) <<https://cloudstack.apache.org/>>. Last access in July, 2015.
- [Cocozza et al. 2015] Cocozza, F., López, G., Marín, G., Villalón, R., and Arroyo, F. (2015). Cloud Management Platform Selection: A Case Study in a University Setting. *CLOUD COMPUTING 2015*, page 92.
- [Dukaric and Juric 2013] Dukaric, R. and Juric, M. B. (2013). Towards a unified taxonomy and architecture of cloud frameworks. *Future Generation Computer Systems*, 29(5):1196–1210.
- [Maron et al. 2014] Maron, C. A. F., Griebler, D., Vogel, A., and Schepke, C. (2014). Avaliação e Comparação do Desempenho das Ferramentas OpenStack e OpenNebula. In *12th Escola Regional de Redes de Computadores (ERRC)*, pages 1–5, Canoas. Sociedade Brasileira de Computação.
- [OpenStack 2015] OpenStack (2015). OpenStack roadmap <<http://openstack.org/>>. Last access May, 2015.
- [Petersen 2006] Petersen, R. (2006). *Introductory Command Line Unix for Users*. Surfing Turtle Press.
- [Roveda et al. 2015] Roveda, D., Vogel, A., and Griebler, D. (2015). Understanding, Discussing and Analyzing the OpenNebula and OpenStack’s IaaS Management Layers. *Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação*, 3(1):15.
- [Shroff 2010] Shroff, G. (2010). *Enterprise Cloud Computing: Technology, Architecture, Applications*. Cambridge University Press.
- [Thome et al. 2013] Thome, B., Hentges, E., and Griebler, D. (2013). Computação em Nuvem: Análise Comparativa de Ferramentas Open Source para IaaS. In *11th Escola Regional de Redes de Computadores (ERRC)*, page 4, Porto Alegre, RS, Brazil. Sociedade Brasileira de Computação.

³<http://hiperfccloud.setrem.com.br>

⁴<http://www.abase.com.br>

⁵<http://www.setrem.com.br>

Implementação de um sistema de emissão automatizada de certificados de atributos para autorização de acesso em ambientes de cloud computing

Cassiano Rodolfo Jung¹, Luciano Ignaczak¹

¹Universidade do Vale do Rio dos Sinos (UNISINOS)
Av. Unisinos, 950 - 93022-000 - São Leopoldo - RS - Brazil

cassiano.jung@gmail.com, lignaczak@unisinos.br

Abstract. *Cloud computing it is emerging as a new paradigm in order to provide access to computing resources in a scalable and dynamic way. The purpose of the article is based on the use of digital certificates and attribute certificates automatically issued, with the purpose of being used as authentication and authorization elements of a resource in the cloud computing environment. The results of the experiment show that the use of authentication with digital certificates and authorization with attribute certificates have an acceptable time to the cloud and may be used as an alternative to the user and password model.*

Resumo. *A computação em nuvem surge como um novo paradigma de modo a prover o acesso a recursos computacionais de maneira escalável e dinâmica. A proposta do artigo baseia-se no uso de certificados digitais e certificados de atributos emitidos automaticamente, com o objetivo de serem utilizados como elementos de autenticação e autorização de um recurso no ambiente de computação em nuvem. Os resultados do experimento demonstram que a utilização da autenticação com certificados digitais e autorização com certificados de atributos possuem um tempo aceitável para a nuvem podendo ser utilizados como alternativa ao modelo de usuário e senha.*

1. Introdução

O termo computação em nuvem, atualmente, é um dos mais abordados na área de Tecnologia da Informação (TI). A computação em nuvem é um modelo que prove a ubiquidade, proporcionando conveniência, e acesso a rede sob demanda para um pool de recursos computacionais configuráveis que podem ser provisionados e disponibilizados rapidamente com esforço mínimo de gestão ou interação do provedor de serviços [Mell and Grance 2011]. Computação em nuvem não é mais uma tendência, mas uma realidade presente no cotidiano das empresas que oferece novos desafios de segurança [Junior 2011].

O método de autenticação largamente utilizado, tanto em sistemas computacionais hospedados internamente nas empresas como nos sistemas hospedados na nuvem, é baseado em usuário e senha. Este método traz uma série de desvantagens e limitações, podendo as senhas ser adivinhadas, esquecidas ou até mesmo roubadas [Fiorese 2000]. O método de autenticação com a utilização apenas de usuário e senha permite ainda a possibilidade de personificação, ou seja, um usuário poder se passar por outro tendo acesso

indevido ao sistema, sendo impossível garantir se aquele que está autenticando-se é realmente o usuário legítimo [Fiorese 2000]. O modelo atual baseado em usuário e senha não possui nenhum mecanismo específico de autorização, de modo que a autorização é normalmente feita em grande maioria dos sistemas computacionais por atribuição de permissões que são concedidas por usuários ou grupos de usuários [Fiorese 2000]. A autorização de acesso é fundamental no acesso do usuário para garantir que o mesmo possuirá o seu acesso controlado e restrito.

Este artigo tem como propósito a implementação de um sistema de emissão automatizada de certificados de atributos para autorização de acesso em ambientes de computação em nuvem. Para suportar a implementação, montou-se uma infraestrutura com dois servidores, um no datacenter de uma empresa localizada em São Leopoldo - RS, que hospeda o serviço de autenticação, e outro no serviço de nuvem da Microsoft Azure em três localizações geográficas distintas: Estados Unidos, Europa e Ásia, que hospeda a aplicação de autorização. Para a utilização de um ambiente de nuvem não é conhecido onde estão hospedados os serviços, portanto o objetivo é a verificação de viabilidade do sistema testando os serviços em diferentes locais geográficos.

A principal contribuição foi realizar a emissão dos certificados de atributos de maneira automatizada e também a autorização com os certificados de atributos permitindo maior segurança em relação ao processo de autorização baseado no usuário e senha como forma de validação de credenciais de acesso. Ao atender os requisitos de autenticação e autorização, o sistema garante a identidade do usuário, ou seja, garantindo a autenticidade. Essas garantias são atestadas em razão de o certificado digital e o certificado de atributo do usuário serem assinados digitalmente com a chave privada de uma autoridade de certificação.

O artigo está dividido na seguinte estrutura: a seção 2 apresenta os trabalhos relacionados selecionados utilizados como base para a implementação do modelo de emissão automatizado; a seção 3 descreve como foi realizada a implementação do sistema de autorização e de emissão automatizada de certificados de atributos; a seção 4 apresenta a definição dos testes e a maneira como os mesmos foram realizados utilizando o sistema implementado; a seção 5 demonstra os resultados dos testes realizados e a seção 6 aborda as considerações finais apontadas no trabalho.

2. Trabalhos Relacionados

[Kang et al. 2003] utilizam web services para realizar a computação distribuída baseado na tecnologia XML, sendo vista como uma tecnologia substituta para as tecnologias de computação distribuídas. O artigo propõe a implementação de uma solução de segurança, baseada em web services, fazendo-se necessário que sejam satisfeitos os requisitos de segurança como autenticação, confidencialidade, integridade, não repúdio, e autorização de usuário. É proposto um sistema de autorização, realizando a efetiva autorização do mesmo utilizando um certificado de atributo para a segurança de web services, na solicitação de um determinado recurso o web service provider verificará o certificado digital e o certificado de atributo do usuário. Após estas verificações, o web service provider fornecerá ou não o acesso do usuário ao recurso solicitado.

Em [Chapin et al. 2008] é examinado o estado da arte moderna em gerenciamento de autorização de confiança, focado em aspectos de política e direitos. O artigo caracte-

riza os sistemas em função de uma estrutura genérica, que leva em conta componentes de implementações práticas, sendo que os sistemas que possuem uma base formal são ressaltados. Nos sistemas de gestão de identidades o elemento principal é o processo de autorização, que determina se o acesso aos recursos deve ou não ser concedido, com base em uma série de condições. A semântica de autorização dá um sentido às funções suportadas por sistemas de gestão de confiança, tanto para a decisão da política quanto para o solicitante do recurso. Embora certo número de técnicas tenham sido propostas para caracterizar a autorização em sistemas de gerenciamento de confiança, o artigo defende que as mais promissoras são aquelas baseadas em rigorosas bases formais, uma vez que as propriedades de segurança passam a ser rigorosamente garantidas.

[Hwang et al. 2000] trabalham com uso de certificados de atributos aplicados ao controle de acesso baseado em função RBAC. Neste artigo, os autores definem um tipo particular de certificado de atributo, que é chamado de role attribute certificate, que traz informações sobre as funções atribuídas ao usuário do certificado, de acordo com suas responsabilidades e capacidades. A utilidade deste tipo de certificado é para restringir a um membro de uma organização certos privilégios em termos de acesso aos recursos de informação. A sua utilização é mais apropriada no comércio eletrônico porque a autorização efetiva baseia-se na informação transportada pelo certificado de atributo de papel.

[Thompson and Essiari 2003] propõe-se a desenvolver um sistema de autorização de sistemas, denominado Akenti, que utiliza um certificados X.509 para permitir o acesso de diversos usuários a um recurso. O objetivo do Akenti é prover uma maneira fácil para um serviço de autorização que atenda às necessidades dos colaboradores e das redes grid, semelhantes à computação em nuvem. Assume-se que os certificados digitais X.509 e os protocolos SSL e TLS foram utilizados para autenticar um usuário que solicitou o acesso a um recurso. Assim como, apresenta a política de autorização com um conjunto de certificados digitais. Os certificados são criados de forma independente pelos interessados. Quando uma decisão de autorização é solicitada o mecanismo de política do Akenti reúne todos os certificados relevantes para o usuário e o recurso, verifica-os e determina os direitos dos usuários no que diz respeito ao recurso.

Após a leitura e seleção dos trabalhos relacionados constatou-se que nenhum realiza a emissão automatizada de certificados de atributos. Tendo em vista que o processo automatizado é fundamental para o ambiente de nuvem, o presente trabalho implementa tal funcionalidade. Dessa forma, o sistema implementado viabiliza um sistema onde não há necessidade de intervenção humana no processo de emissão do certificado de atributo para autorização.

3. Implementação

Para oferecer um processo automatizado de autenticação e autorização no ambiente de computação em nuvem, este artigo propõe o uso de certificados digitais e certificados de atributos. O certificado digital é um documento eletrônico assinado digitalmente pela sua Autoridade Certificadora que permite que um indivíduo comprove sua identidade através de sua chave privada. [Housley et al. 2002]. O certificado de atributo é um documento eletrônico assinado digitalmente que apresenta qualidades associadas a uma determinada entidade. Além dos atributos, a temporalidade é outra característica importante, pois

permite que uma informação seja confiável por um período de tempo pré-determinado [Farrell and Housley 2002].

A implementação do sistema é composta por três módulos: cliente, provedor de serviço e provedor de autorização. O módulo cliente é responsável pelo envio do certificado digital do cliente (CDc) ao provedor de serviço, ou seja, o módulo cliente realiza o papel da aplicação que o usuário final irá utilizar para acessar um serviço hospedado em uma infraestrutura de computação em nuvem. O módulo provedor de serviço por sua vez recebe, valida e assina o CDc com o certificado digital do serviço (CDs) e envia o CDc e a assinatura digital ao provedor de autorização, ou seja, o módulo provedor de serviço é responsável pelo provimento de um serviço para o usuário final que fornece acesso a autenticação a um sistema hospedado em uma infraestrutura de computação em nuvem. O módulo provedor de autorização valida as informações enviadas pelo provedor de serviços, posteriormente realizando a emissão do certificado de atributo do cliente (CAC), ou seja, o módulo provedor de autorização realiza o papel da aplicação que fornece ao usuário final um serviço de autorização hospedado em uma infraestrutura de computação em nuvem. Para realização do experimento foi necessária a utilização das bibliotecas de criptografia BouncyCastle, que foi utilizada como base para a implementação do módulo de autorização e emissão automática dos certificados de atributos e a biblioteca FlexiCo-reProvider, para a realização do processo de assinatura digital dos certificados e validação das mesmas.

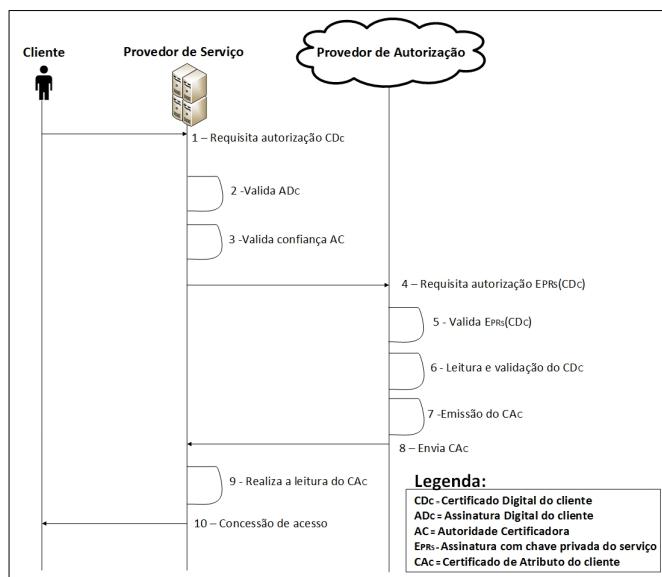


Figura 1. Processo de autenticação e autorização

Para a execução da implementação é necessário inicializar o provedor de autorização, que abre um socket em uma porta aguardando uma conexão do provedor de serviço. O provedor de serviço também necessita ser iniciado para abrir um socket em uma porta aguardando uma conexão do cliente. Após as inicializações dos módulos dos provedores, a próxima etapa é executar o módulo cliente que estabelece uma conexão

através de uma porta com o provedor de serviço enviando o CDc para iniciar o processo de autenticação (passo 1). O provedor de serviço por sua vez recebe o CDc e realiza a validação da assinatura digital (passo 2) e em seguida valida se a autoridade de certificação do CDc é confiável (passo 3).

Após as validações, o serviço assina o CDc digitalmente com a chave privada do certificado digital do serviço (CDs) e após envia-o juntamente com a Assinatura Digital do serviço (ADs) para o provedor de autorização (passo 4). O provedor de autorização, por sua vez, realiza a validação da ADs e realiza a validação do CDs (passo 5). É realizada a leitura e validação do CDc para que seja localizado em um banco de dados um cadastro de idêntico ao constante no CDc e verificado qual papel está atrelado ao mesmo para posterior emissão do CAC (passo 6). Posteriormente, é realizado o processo de emissão do CAC com base no CDc e no papel (passo 7). O provedor de autorização envia o CAC para o serviço através da mesma porta para continuar o processo de autorização (passo 8). O serviço realiza a leitura do CAC para verificar qual o papel que consta no certificado e qual será o acesso do cliente no sistema (passo 9). Por fim, é realizado o envio da mensagem ao cliente através da mesma porta de conexão concedendo o acesso a aplicação de acordo com o CAC emitido pelo provedor de autorização (passo 10).

4. Testes

A realização dos testes para acesso aos sistemas, com certificados digitais e de atributos, foram realizados a partir da implementação desenvolvida. Foi utilizado um ambiente de nuvem onde não é conhecido onde estão hospedados os serviços, portanto o objetivo é a verificação de viabilidade do sistema testando os serviços em diferentes locais geográficos. Foram definidos três cenários de testes, onde o provedor de serviço sempre permanece em um mesmo local geográfico. Já o provedor de autorização foi hospedado em três diferentes locais para permitir a comparação dos tempos de acesso entre cada provedor. Os locais definidos para hospedagem dos provedores de autorização foram América do Norte, Europa e Ásia, os quais são ilustrados na Figura 2. Em cada teste foram realizados 30 tentativas de acesso, que mediram o tempo necessário para a autenticação do cliente, a emissão do certificado de atributo e a concessão da autorização do acesso. Para a realização dos testes foram montados três cenários cada um com 120 testes de acessos, totalizando 360 testes. Para cada cenário foi definido que o provedor de autorização estaria em um local geográfico distinto dos demais cenários, para possibilitar um comparativo entre os tempos de cada um.

O primeiro cenário é composto pelo provedor de serviço hospedado em um datacenter no sul do Brasil e o provedor de autorização hospedado no datacenter da Azure nos Estados Unidos. Este cenário foi dividido em 4 estágios de testes, no primeiro estágio o teste foi realizado com o cliente na residência do autor em São Leopoldo-RS. O teste foi intitulado cliente casa + ADSL, pois foi utilizada a conexão de banda larga, onde foram realizados 30 testes de acessos ao sistema. O segundo estágio foi também realizado da residência do autor, intitulado cliente casa + 3G, pois foi utilizada uma conexão 3G, sendo realizados mais 30 testes de acesso ao sistema. O terceiro estágio foi realizado através de um servidor criado na nuvem no Datacenter da Azure na Indonésia, intitulado cliente Asia, pois foi testado a partir de um cliente na Asia, onde foram realizados também 30 testes de acesso ao sistema. O quarto estágio foi realizado através de um servidor criado na nuvem no Datacenter da Azure na Holanda, intitulado cliente Europa, pois foi testado

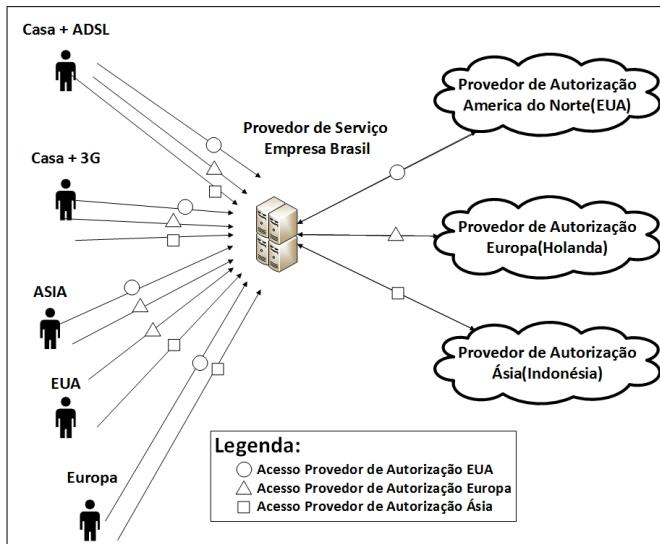


Figura 2. Cenário de testes

através de um cliente na Europa, sendo realizados 30 testes de acesso ao sistema.

O segundo cenário é composto pelos mesmos componentes do primeiro cenário porém o provedor de autorização está localizado provedor de autorização hospedado no Datacenter da Azure na Holanda. Já o terceiro cenário assim como os demais também tem seus componentes iguais aos demais, porém o provedor de autorização se localiza no continente Asiático no Datacenter da Azure da Indonésia.

5. Resultados

Conforme é possível verificar na Tabela 1, que apresenta os tempos dos testes para os três cenários, observou-se que o tempo médio varia de acordo com as formas de acesso e a localização geográfica. O cenário que o provedor de autorização está hospedado na América do Norte apresentou o tempo de acesso médio Casa + ADSL de 3733 milissegundos, já o tempo de acesso médio Casa + 3G apresentou uma média maior de 4180 milissegundos, seguida pelo cliente Ásia com média de 5453 milissegundos e o cliente Europa com a média de 4661 milissegundos.

Ao configurar o provedor de autorização na Europa, as formas de acesso permanecem as mesmas apenas alterando o cliente Europa para cliente Estados Unidos. Neste cenário, a média para o acesso realizado Casa + ADSL ficou em 4559 milissegundos, porém o acesso realizado através Casa + 3G apresenta uma média maior de 4899 milissegundos seguida pelo cliente Ásia com média de 5284 milissegundos e o cliente Europa com a média de 4797 milissegundos.

Com o provedor de autorização na Ásia, as formas de acesso permanecem as mesmas apenas alterando o cliente Ásia para cliente Europa. Para o acesso realizado Casa + ADSL a média ficou em 4724 milissegundos, para o acesso realizado através do cliente Casa + 3G a média apresentada é maior em torno de 5144 milissegundos seguida pelo

Tabela 1. Cenário de testes de acesso aos Provedores de autorização

Cliente	Cenário 1 de testes de acesso Provedor de autorização América do Norte.				
	Menor Tempo(ms)	Maior Tempo(ms)	Média(ms)	Desv. Padrão	Int. de conf. (Lim. Inferior/Superior)
Casa +ADSL	3525	4152	3733	162	3571 / 3794
Casa +3G	3797	4968	4180	251	3928 / 4182
Cliente ASIA	5309	5815	5453	149	5305 / 5455
Cliente Europa	4380	4902	4661	155	4505 / 4662
Cenário 2 de testes de acesso Provedor de autorização Europa.					
Casa +ADSL	4305	4913	4559	168	4391 / 4560
Casa +3G	4367	5501	4899	235	4665 / 4902
Cliente ASIA	4935	5781	5284	249	5035 / 5287
Cliente EUA	4691	5060	4797	94	4703 / 4798
Cenário 3 testes de acesso Provedor de autorização Ásia.					
Casa +ADSL	4683	4938	4724	79	4645 / 4725
Casa +3G	4931	6460	5144	285	4858 / 5147
Cliente EUA	5126	5589	5174	111	5062 / 5175
Cliente Europa	5270	5785	5311	132	5179 / 5312

Cliente EUA com média de 5174 milissegundos e o Cliente Europa com a média de 5311 milissegundos.

Com o provedor de autorização na Ásia observou-se que as médias apresentadas foram superiores aos demais cenários, isso deve-se ao fato de o servidor de autorização estar localizado no continente Asiático que possui maior latência de resposta em milissegundos em relação ao continente Americano e Europeu, nos demais cenários também observou-se sempre um maior tempo de resposta quando o teste partiu do continente Asiático. Outra importante observação também se deve aos testes realizados com a conexão 3G que apresentou um desvio padrão mais alto em relação aos demais clientes. Isso se deve em razão de ser um tipo de conexão bastante instável em relação as demais por ser uma conexão que se utiliza da rede móvel de celular para trafegar os dados.

6. Considerações Finais

O presente trabalho teve como objetivo elaborar um modelo de implementação de um sistema de emissão automatizada de certificados de atributos para autorização de acesso em ambientes de computação em nuvem, com a utilização de certificado digital para autenticação e certificados de atributos para autorização. O experimento demonstrou os mecanismos de autenticação e autorização do modelo juntamente ao intervalo de tempo para a realização do processo de autenticação e autorização do cliente. Através dos resultados obtidos com os testes de acesso ao sistema, constatou-se que o local geográfico onde está hospedado o provedor de autorização e também o local de acesso do cliente influenciam diretamente nos tempo do processo.

Porém os testes também mostram que independentemente do local do provedor de autorização, os tempos não variam tanto a ponto de inviabilizar a utilização do sistema na computação em nuvem, atestando que o modelo adotado para o presente experimento, tem total aderência a nuvem em razão de apresentar os tempos do processo aceitáveis por serem inferiores a 10 segundos em relação ao tempo [Nielsen 1994].

O presente trabalho buscou a implementação de um protocolo seguro mas também

considerando um tempo de execução aceitável do processo, além de utilizar componentes acessíveis como a linguagem java que é largamente utilizada para o desenvolvimento de sistemas. Utilizou-se também uma infraestrutura de nuvem bem consolidada e acessível, que é o Microsoft Azure, que permite a criação e configuração de máquinas virtuais em poucos minutos.

A principal preocupação do trabalho foi o processo de emissão automático do certificado de atributo, pois sem esse requisito atendido o processo de autenticação e autorização para a nuvem não seria atendido plenamente de forma dinâmica, em razão pela qual a cada tentativa de acesso por parte do cliente seria necessária uma ação humana de execução para a emissão do certificado de atributo. Com o sistema de autorização criado essa ação não é necessária pois o processo de emissão é automatizado no provedor de autorização, viabilizando o modelo apresentado para a computação em nuvem.

Após a realização de todos os testes foi possível identificar que um trabalho futuro é a adaptação do modelo para possibilitar a utilização da infraestrutura criada por mais de um provedor de serviço. Essa alteração permitiria ao cliente a utilização do mesmo certificado digital para mais de um serviço. Sendo assim a emissão dos certificados de atributos ocorreria conforme a necessidade dos sistemas que o usuário necessitasse utilizar. Com a alteração sugerida o modelo poderá ser ainda mais apropriado para o cenário de nuvem, possibilitando a utilização do provedor de autorização para diversos provedores de serviço.

Referências

- Chapin, P. C., Skalka, C., and Wang, S. X. (2008). Authorization in trust management: Features and foundations. *ACM Comput. Surv.* 40, 3, Article 9, page 48.
- Farrell, S. and Housley (2002). An internet attribute certificate profile for authorization. RFC 3281.
- Fioresi, M. (2000). Uma proposta de autenticação de usuários para ensino a distância. Master's thesis, Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul.
- Housley, R., Polk, W., Ford, W., and Solo, D. (2002). Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile. RFC 5280.
- Hwang, J.-J., Wu, K.-C., and Liu, D.-R. (2000). Access control with role attribute certificates. *Computer Standards and Interfaces*, 22:43–53.
- Junior, A. S. A. (2011). Gerenciamento de identidades como um serviço para ambiente de computação em nuvem. Master's thesis, Programa de Pós-Graduação em Ciência da Computação) – Universidade Federal de Santa Catarina, Florianópolis.
- Kang, M.-H., Kim, K.-N., and Ryou, H.-B. (2003). An authorization mechanism for web services using an attribute certificate. *Security Technology, 2003. Proceedings. IEEE 37th Annual 2003 International Carnahan Conference on*, pages 144–150.
- Mell, P. and Grance, T. (2011). The nist definition of cloud computing.
- Nielsen, J. (1994). *Usability Engineering*. Academic Press.
- Thompson, M. R. and Essiari, A. (2003). Certificate-based authorization policy in a pki environment. *ACM Transactions on Information and System Security*, 6:566–588.

III

Sessão 3 - Segurança de Aplicações

Um sistema de pagamento eletrônico com garantia de privacidade baseado no algoritmo criptográfico RSA

Gustavo Gattino¹, Marcelo Danesi¹, Luciano Ignaczak¹

¹Instituto de Informática – Universidade do Vale do Rio dos Sinos (UNISINOS)
Caixa Postal 275 – 93.022-000 – São Leopoldo – RS – Brasil

gustavo.gattino@gmail.com, {mdanesi, lignaczak}@unisinos.br

Abstract. *The exposure of customer's data can lead to profile mapping and unpleasant situations depending on the nature of the goods purchased. To solve this problem, a system to execute e-commerce transactions assuring the privacy of the customer is implemented. Through it, the customer is assured that the privacy of your information is secured toward the other entities involved in the process. The system uses a model with three entities — customer, shop and payment processor — which none of them can associate the purchase to the buyer.*

Resumo. *A exposição dos dados do cliente a diferentes entidades durante o processo de compra pode levar a mapeamento de perfil e situações constrangedoras, dependendo da natureza dos produtos adquiridos. Para solucionar esse problema, um sistema para realização de transações de comércio eletrônico garantindo a privacidade da parte compradora é implementado. Através dele, o usuário tem a garantia de que a privacidade de suas informações estão seguras para com as demais entidades envolvidas no processo. Esse sistema utiliza um modelo com três entidades — cliente, loja virtual e processador de pagamento — as quais não conseguem associar a compra realizada ao cliente comprador.*

1. Introdução

Recentemente as pessoas tornaram-se mais sensíveis em relação a sua privacidade *on-line* — o direito de controlar ou influenciar quais informações sobre elas podem ser coletadas, armazenadas e divulgadas [ISO 7498-2 1989] — percebendo que deixam todos os tipos de rastros ao navegar [Souza]. Casos como o monitoramento de pagamentos internacionais, bancos e transações de cartão de crédito tornaram usuários de serviços de pagamento *on-line* mais preocupados com quem acessa seus dados [Der Spiegel 2013].

A criptografia fornece meios de proteção à privacidade com relação a terceiros, de modo a dificultar a espionagem. No entanto, existem situações em que a proteção da privacidade deve ir ainda mais longe e é neste momento em que a identidade do usuário passa a ser o ponto mais importante nas trocas de informações. A demanda pela omisão da identidade *on-line* é completamente justificável, uma vez que muitas situações da vida *off-line* são anônimas: lojas físicas oferecem um certo grau de anonimato para seus clientes se esses pagarem com dinheiro.

O principal problema com pagamentos digitais é que o cliente é obrigado a passar as suas informações de identificação e pagamento ao vendedor, sendo essa falta de privacidade uma das principais razões que impedem o crescimento do comércio eletrônico, uma vez que limita a confiança de potenciais clientes [Grudzinski 2013] [Thomasson 2013].

Diante do problema levantado, o objetivo deste trabalho é elaborar um sistema de pagamento que garanta a privacidade das informações de pagamento do comprador e a não vinculabilidade entre as compras realizadas. Através desse sistema, o usuário terá a garantia total de que a privacidade de suas informações de transação estarão seguras para com as demais entidades envolvidas no processo de pagamento. O sistema define um modelo com três entidades: cliente, loja virtual e processador de pagamento. Entre as três entidades, apenas o processador de pagamento conhece a identidade do cliente, no entanto nenhuma entidade consegue associar a compra realizada ao cliente comprador.

A principal contribuição gerada por este trabalho é a construção do sistema de pagamento digital que faz uso da função criptográfica RSA, amplamente utilizada para garantir autenticidade e confidencialidade, como forma de garantir a não vinculabilidade entre uma compra realizada e o seu comprador. Essa garantia é possível através da utilização das propriedades de injeção e não sobrejeção que a função possui.

A segunda seção do artigo trata de trabalhos relacionados ao assunto de pagamentos digitais e privacidade de usuários na Internet. A terceira aborda o funcionamento do modelo elaborado. A quarta seção relata sobre a implementação do sistema criado, enquanto a quinta e a sexta seção apresentam os experimentos realizados e seus resultados. Por fim, a seção oito apresenta as considerações finais.

2. Trabalhos relacionados

Trabalhos anteriores mostram a dificuldade de manter a privacidade de um usuário no contexto de dados em rede e serviços *on-line* que expõem informações parciais do seu comportamento. [Backstrom et al. 2007] consideraram ataques à privacidade dos usuários, identificando-os através da estrutura de rede que os cercam e discutiram a dificuldade de garantir o anonimato do usuário quando há presença de dados na rede que os identificam. [Crandall et al. 2010] correlacionam laços sociais entre usuários, em que nenhuma correlação foi explicitamente declarada, apenas ao identificar padrões *off-line*: tendo em conta que duas pessoas estão aproximadamente na mesma localidade geográfica, aproximadamente ao mesmo tempo e em diversas ocasiões, estas, provavelmente, estão relacionadas. [Narayanan and Shmatikov 2008] quebraram o anonimato do *dataset* do algoritmo *Netflix Prize* usando informações do IMDB2, a qual tinha conteúdos de usuário similar, mostrando que a correlação estatística entre diferentes, mas relacionados, conjuntos de dados podem ser usados para atacar a privacidade.

As dificuldades na manutenção da privacidade motivaram a elaboração de novos modelos e sistemas. [Rennhard et al. 2004] apresentam novos componentes que permitem um comércio eletrônico seguro baseado em pseudônimos. De um lado, estes componentes permitem que clientes possam navegar através de um loja virtual, selecionar seus bens e pagá-los com seu cartão de crédito, de tal forma que nem a loja, nem o emissor do cartão de crédito, nem um intruso serão capazes de obter qualquer informações sobre a identidade do cliente. Por outro lado, é garantido que nenhuma das partes envolvidas é capaz de atuar desonestamente durante o pagamento. [Konidala et al. 2012] propõe um modelo de pagamento *mobile* pré-pago baseado em HTTPS, no qual o cliente obtém informações sobre a conta bancária do comerciante e instrui seu banco a transferir dinheiro a conta no momento do pagamento. O modelo faz uso do esquema de assinatura parcialmente cega para esconder a identidade do cliente do banco e co-

mercante. Como resultado, o modelo provê ao cliente um maior controle sobre seus pagamentos e proteção de sua privacidade, tanto para com o banco quanto ao comerciante. [Nakanishi and Sugiyama 2005] propõe um sistema de moeda eletrônica *on-line* anônima. No modelo, a moeda eletrônica é uma assinatura digital do banco, assinada de forma cega através do algoritmo Camenisch-Lysyanskaya e oculta através de criptografia e técnicas *zero-knowledge* durante o pagamento, impossibilitando a ligação entre transações. O modelo pode revogar o anonimato impedindo sua utilização para fins ilícitos.

3. Sistema de pagamento eletrônico

Este artigo define um sistema para realização de transações de comércio eletrônico garantindo a privacidade da parte compradora. Através dele, o usuário tem a sua privacidade assegurada durante o processo de compra. Essa segurança se dá através do uso de pseudônimos na compra com a loja virtual e da ofuscação do *ticket* escolhido pelo usuário com o processador de pagamento. Através desses dois elementos, o anonimato da compra é alcançado, garantindo a não vinculabilidade entre uma ou mais compras realizadas através do sistema e as identidades utilizadas para a compra. O sistema utiliza um modelo com três entidades — cliente, loja virtual e processador de pagamento.

3.1. Funcionamento do sistema

O sistema utiliza oito etapas para a realização da compra, conforme mostrado na Figura 1, as quais são descritas a seguir.

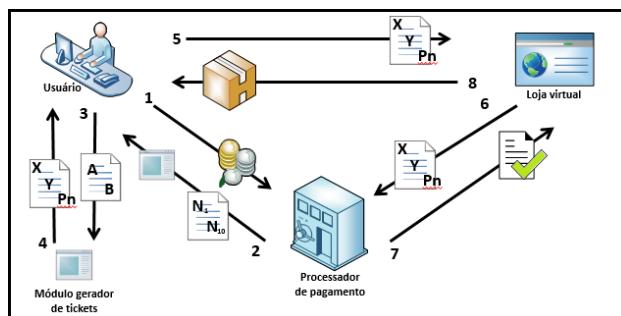


Figura 1. Processo de compra do sistema de pagamento proposto

No início do processo de compra, o cliente entra em contato com o processador de pagamento solicitando a aquisição de um *ticket* de pagamento. Nesse momento, para adquirir o *ticket*, o cliente realiza um pagamento digital amplamente adotado (1) — este pagamento pode ocorrer através de cartão de crédito, boleto bancário, Paypal, etc.

Após concluída a transação, o processador de pagamento envia ao cliente o módulo de geração de *tickets* e um conjunto de n valores (2). O módulo e os valores enviados são utilizados para a geração do *ticket* do usuário. A partir do momento em que esses dados são recebidos, o usuário tem um espaço de tempo t para geração e gasto do seu *ticket*.

Ao receber os dois componentes, o usuário escolhe um par de valores dentre os n recebidos — os dois valores escolhidos serão representados por A e B — e submete-os ao módulo de geração (3). O módulo de geração deriva matematicamente os valores A

e B em dois novos valores distintos, representados por X e Y , usando o algoritmo criptográfico RSA. Esses dois novos valores não podem ser vinculados aos valores originais, porém o processador de pagamento consegue verificar a sua autenticidade. A relação matemática entre A e B e X e Y é usada como autenticação.

Após a derivação, o módulo de geração utiliza os valores resultantes em uma função *Proof-of-Work*. Sua execução exige $(t/2) + 1$ minutos para conclusão (visando impedir a utilização de mais do que dois dos valores enviados — *double spending*). Após concluído o PoW, o módulo de geração retorna ao usuário o *ticket*: uma *string* que concatena os valores X e Y e o valor utilizado pelo PoW como *nounce*(4). Após a execução do módulo gerador, o usuário tem em torno $(t/2) - 1$ minutos para realizar a compra.

De posse do *ticket*, o usuário já pode realizar a sua compra virtual. Para isso, o usuário envia-o à loja virtual (5) que, por sua vez, valida sua autenticidade junto ao processador de pagamento (6). O processador de pagamento comprova a validade do ticket através de um verificador de PoW e confere se o valor recebido já não consta em seu banco de dados de tickets recebidos (7).

Validado o *ticket*, a loja virtual libera o produto ao comprador (8). A compra realizada não pode ser vinculada ao usuário, pois o processador de pagamento não sabe qual A e B o usuário escolheu para gerar X e Y e o mesmo não enviou seus dados reais.

O funcionamento correto do sistema depende do atendimento de alguns requisitos obrigatórios, os quais definem limites no escopo do seu uso.

1. Limite de instâncias do módulo gerador de *tickets*: O usuário deve ser incapaz de executar duas ou mais instâncias do módulo de geração de *tickets* de forma paralela.
2. Definição do processador de pagamento como entidade de confiança: O processador de pagamento deve ser uma entidade confiável e não vincula os números enviados a clientes com seus valores derivados finais através da submissão desses ao módulo gerador de *tickets*.
3. Definição do usuário como entidade anônima na rede: O usuário faz uso de tecnologias que o qualificam como anônimo na rede ao comunicar-se com as outras entidades envolvidas no processo de compra.

4. Implementação

Para a realização de experimentos foi necessário o desenvolvimento de um protótipo para realizar testes de segurança das funcionalidades do sistema proposto. A implementação contemplou o módulo de geração de *tickets*, utilizado pelo comprador, e o módulo de validação dos *tickets*, presente no sistema de processamento de pagamentos. Os dois módulos foram desenvolvidos usando a linguagem de programação Python 3.3.

O módulo de geração de *tickets* recebe o par de valores selecionados pelo usuário e aplica-os na função $m^e \bmod N$ do algoritmo RSA. Os valores de e e N são fixados pelo processador de pagamento de forma que sejam coprimos para que o resultado da função seja sempre único. Os valores resultantes da função são utilizados no PoW baseado em Hashcash onde, concatenados a um nounce, são submetidos a uma função *hash SHA-1*. O resultado desse hash é convertido para a base hexadecimal e tem seus dígitos iniciais analisados a procura de uma quantidade de zeros. Para que o *ticket* seja informado ao usuário,

a quantidade de zeros presentes nos dígitos iniciais do resultado do *hash* deverá ser maior ou igual a z . Caso a quantidade não seja suficiente, o processo é reiniciado informando um novo *nounce* até que esse requisito seja atendido. O número de zeros hexadecimais a serem gerados pelo PoW são escolhidos com base na velocidade de criação de *tickets* que se deseja alcançar. Ao fim do processamento, o módulo apresenta ao usuário o *ticket* de pagamento, formado pela concatenação do último *nounce* utilizado e os dois números resultantes da função $m^e \text{ mod } N$.

O módulo validador recebe o *ticket* enviado pela loja virtual, identificando se ele foi gerado através do módulo gerador e se o par de valores usados são válidos para a compra. O *ticket* recebido é aplicado a uma função *hash* SHA-1 e tem seu valor resultante convertido para a base hexadecimal. O valor convertido é analisado verificando se a quantidade de zeros aceitos configurados no PoW está correta. Se a quantidade mínima de zeros iniciais for identificada, então o módulo verifica se o par de valores passados está presente no banco de dados de *tickets* válidos. Caso os critérios sejam atendidos, o módulo classifica o *ticket* recebido como válido.

5. Experimentos

Para analisar a segurança do sistema a respeito de colisões na escolha dos valores enviados aos compradores e a resistência do sistema a ataques de força bruta de fabricação de *tickets* foi necessário implementar novas aplicações.

Para a realização dos testes de colisão e ataques de fabricação através do método de força bruta, foram utilizados três conjuntos de anonimato contendo uma quantidade de 100, 250 e 500 participantes. Além disso, os testes consideraram três quantidades de valores para a geração do *ticket*, sendo eles 1.000, 5.000 e 10.000. Esses valores foram escolhidos pelo autor com o objetivo de validar diferentes proporções na relação usuários e valores, uma vez que não foram encontradas referências abordando o assunto.

Para a análise das colisões na seleção de valores foi implementado um aplicativo capaz de gerar o par de valores pseudoaleatórios para cada usuário do teste, comparando-os à procura de colisões. O experimento fez uso de nove cenários de teste — 2 fatores com 3 níveis cada — no qual 100, 250 e 500 usuários escolhem um par de valores contidos em uma listagem de 1.000, 5.000 e 10.000 valores disponíveis. O número total de colisões geradas para cada uma das replicações foi utilizado como variável de resposta.

Para a avaliação da resistência a um ataque de força bruta foi implementado um aplicativo capaz de gerar aleatoriamente um par de valores pseudoaleatórios, submetê-los no módulo gerador de *tickets* e comparar os valores gerados com uma listagem de valores aceitos pelo processador de pagamento. A avaliação fez uso de três cenários de teste, nos quais um par de valores é escolhido de forma pseudoaleatória, submetido ao módulo gerador de *tickets* e é comparado a uma listagem de 1.000, 5.000 e 10.000 valores válidos. O número total de *tickets* válidos aceitos pelo processador de pagamento após a ocorrência de 100.000 tentativas foi utilizado como variável de resposta. Um total de 30 replicações foi utilizado para cada nível do fator primário.

6. Resultados

Nesta seção são apresentados e analisados os resultados provenientes dos experimentos identificados na Seção 5, os quais computaram 62 milhões de tentativas de colisão na

escolha de valores para variação e 9 milhões de tentativas de fabricação de *tickets* através do modulo gerador de *tickets*. Visando uma melhor compreensão, a análise dos resultados foi dividida em subseções distintas para cada teste realizado.

6.1. Colisão na seleção dos valores a serem variados

Inicialmente foram coletados o número total de colisões geradas para cada um dos três grupos de usuários e suas diferentes quantidades de valores disponíveis. Na Tabela 1, são identificados os resultados do teste para cada fator e seus níveis. A coluna “colisões” representa o total de ocorrências em que um par de valores foi escolhido duas ou mais vezes em uma mesma execução do teste.

Usuários	Valores	Repetições	Colisões	% de colisão
100	1.000	499.500	106	0,021 %
250	1.000	499.500	30.245	6,055 %
500	1.000	499.500	110.948	22,212 %
100	5.000	12.497.500	2.092	0,017 %
250	5.000	12.497.500	31.163	0,249 %
500	5.000	12.497.500	124.099	0,993 %
100	10.000	49.995.000	4.918	0,010 %
250	10.000	49.995.000	30.941	0,062 %
500	10.000	49.995.000	124.346	0,249 %

Tabela 1. Resultados do teste de colisão na escolha dos pares

Através da observação da Tabela 1 é possível perceber o aumento no número de colisões à medida que a quantidade de usuários cresce. Essa característica fica evidente, por exemplo, nos resultados dos níveis de 10 mil valores, os quais a duplicação da quantidade de usuários praticamente quadruplica a taxa de colisões — à medida que 250 usuários geram 30 mil colisões, 500 usuários aumentam a ocorrência para 124 mil.

Para identificar a relação entre o número de valores disponíveis para a escolha e taxa de colisões, um cálculo de probabilidade foi realizado, conforme apresentado na coluna "% de colisão". O resultado desse cálculo representa a chance de ocorrência de colisão para cada vez que os usuários devem escolher um par de números.

É possível observar que a probabilidade de ocorrência de colisão diminui a medida que mais valores para escolha são adicionados. Essa característica fica clara ao comparar as chances de colisão para 500 usuários, ao qual dispondo de mil valores para a escolha resulta em uma chance de 22% de colisão por rodada, porém o valor é reduzido assim que 5 mil valores são disponibilizados, diminuindo as chances para menos de 1%.

Após a análise dos dados, fica claro que nos dois diferentes fatores testados há um aumento na ocorrência de colisões, conforme a quantidade de usuários cresce. Essa condição independe da quantidade de valores disponíveis para escolha. A análise permite concluir que a proporção mínima adequada deve ser de 1 usuário para 100 valores, para que as chances de colisão fiquem abaixo de 0,01%.

6.2. Fabricação de *tickets* através de força bruta

Inicialmente, para a realização do experimento de fabricação através de força bruta, foi coletado o número total de *tickets* válidos gerados em cada um dos três níveis de valo-

res. Na Tabela 2, são apresentados os resultados dos testes para cada um dos níveis ao fator primário. A coluna “sucessos” identifica a quantidade total de *tickets* de pagamento válidos gerados para cada um dos níveis de teste.

Valores válidos	Repetições	Replicações	Sucessos
1.000	100.000	30	76
5.000	100.000	30	1.859
10.000	100.000	30	7.422

Tabela 2. Resultados do teste de fabricação de *tickets* através de força bruta

Após a coleta dos dados, os resultados do experimento foram analisados relacionando a quantidade de sucessos na geração dos *tickets* com a quantidade de valores aceitos pelo processador de pagamento. É possível perceber um crescimento da taxa de sucesso de criação de *tickets* válidos a medida que mais valores aceitos são disponibilizados para validação. Esse ponto apresenta uma relação inversa com a quantidade de valores disponíveis comparado aos dados apresentados pelo teste de colisão, no qual há uma maior dificuldade do usuário legítimo gerar um *ticket* válido quando a quantidade de valores disponíveis para escolha é menor.

No pior cenário, em que 10 mil valores estão disponíveis para validação, a taxa de fabricação de *tickets* é de 404 tentativas por sucesso. Ao considerar uma implementação do sistema configurado para que a aceitação do PoW exija 6 zeros de validação, um atacante gastará um tempo de 40 horas para obter sucesso na criação de um *ticket*. Todavia, apesar do aumento na taxa de sucesso na fabricação dos *tickets*, em virtude da funcionalidade de expiração dos valores enviados aos usuários, esse indicador não apresenta risco para o sistema caso uma grande quantidade de valores para escolha seja utilizada.

7. Considerações finais

Neste trabalho, foi elaborado um sistema para realização de transações de comércio eletrônico que assegura a privacidade do comprador durante o processo de pagamento da compra. Ela é garantida através do uso de pseudônimos no momento da compra com a loja virtual e da quebra de relação do *ticket* final utilizado para compra com os itens fornecidos pelo processador de pagamento para a sua geração. Através desses dois elementos, o anonimato da compra é alcançado, garantindo a não vinculabilidade entre uma ou mais compras realizadas através do sistema e as identidades utilizadas para a compra.

Dentre os itens enviados pelo processador de pagamento, o mais sensível é o conjunto de valores para a escolha do comprador, pois ele deve ser encaminhado a diferentes compradores para que o conjunto de anonimato do pagamento não permita ao processador de pagamento identificar qual usuário escolheu qual par de valores. Contudo, essa definição pode acarretar em colisão na escolha dos números — dois usuários diferentes podem escolher o mesmo par de números. Para tratar essa possibilidade, o sistema faz uso da mesma técnica utilizada pelos algoritmos criptográficos de chave pública: o conjunto de valores disponíveis para escolha é grande o suficiente para que a probabilidade de colisão seja mínima. No sistema, testes foram realizados identificando a necessidade de proporção de, pelo menos, 1 usuário para cada 100 valores disponíveis, fazendo com que as chances de colisão fiquem abaixo de 0,01%.

Além disso, o sistema é resistente a cenários nos quais um possível usuário mal-intencionado realize tentativas de fabricação de *tickets* válidos através de força bruta. Sua resistência a este tipo de ação ocorre através do uso do *proof-of-work*, o qual, além de exigir um *nounce* válido para o processamento do *ticket*, torna o processo de fabricação desinteressante a medida que seu processamento fica muito custoso. Mesmo assim, testes foram realizados simulando um cenário de ataque, os quais identificaram que, mesmo no pior cenário, um atacante não poderia fabricar um *ticket* rápido o suficiente para vencer a expiração dos valores disponibilizados para escolha.

As contribuições apresentadas neste trabalho representam um primeiro esforço para o uso de funções injetoras não-sobrejetoras como forma de alcançar a não vinculação entre transações de comércio eletrônico. Apesar da segurança apresentada por meio dos estudos experimentais, o modelo pode ser aprimorado de forma que um limite de instâncias do módulo gerador de *tickets* seja assegurada, fazendo com que o usuário não possa gerar mais de um *ticket* de forma paralela. Em um segundo momento, também é possível que o processador de pagamento não seja mais considerado uma entidade de confiança, identificando uma forma mais segura para envio dos valores a serem derivados aos usuários.

Referências

- Backstrom, L., Dwork, C., and Kleinberg, J. (2007). Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th international conference on World Wide Web*, pages 181–190. ACM.
- Crandall, D. J., Backstrom, L., Cosley, D., Suri, S., Huttenlocher, D., and Kleinberg, J. (2010). Inferring social ties from geographic coincidences. *Proceedings of the National Academy of Sciences*, 107(52):22436–22441.
- Der Spiegel (2013). Follow the money: Nsa spies on international payments. <http://is.gd/SpiegelNSA>.
- Grudzinski, G. (2013). Do online shoppers care about privacy? <http://is.gd/IRPriv>.
- ISO 7498-2 (1989). *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture*. IEC.
- Konidala, D. M., Dwijaksara, M. H., Kim, K., Lee, D., Lee, B., Kim, D., and Kim, S. (2012). Resuscitating privacy-preserving mobile payment with customer in complete control. *Personal and Ubiquitous Computing*, 16(6):643–654.
- Nakanishi, T. and Sugiyama, Y. (2005). An efficient on-line electronic cash with unlinkable exact payments. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 88(10):2769–2777.
- Narayanan, A. and Shmatikov, V. (2008). Robust de-anonymization of large sparse datasets. In *Security and Privacy, 2008. IEEE Symposium*, pages 111–125. IEEE.
- Rennhard, M., Rafaeli, S., Mathy, L., Plattner, B., and Hutchison, D. (2004). Towards pseudonymous e-commerce. *Electronic Commerce Research*, 4(1-2):83–111.
- Souza, E. A privacidade como diferencial. goo.gl/IbtzLr.
- Thomasson, E. (2013). Big retailer is watching you: stores seek to match online savvy. <http://is.gd/ITWatchYou>.

Uma análise dos certificados digitais usados na assinatura de aplicação Android

Jonata Fröhlich¹, Claudia Angelita Fagundes Raupp¹, Luciano Ignaczak¹

¹Universidade do Vale do Rio dos Sinos(UNISINOS)

Av. Unisinos, 950 – 93.022-000 – São Leopoldo – RS – Brasil

jonatafrohlich@gmail.com, {rauppc, lignaczak}@unisinos.br

Abstract. *The number of Android platform users is growing every day, and the number of available applications grows in similar proportion. Currently, each application installed on the Android platform must be digitally signed, but there is no control of the safety criteria used in used certificates, leaving to the developer himself the task of selecting information and characteristics of the certificate. This paper analyzed security characteristics of 397 digital certificates used for application signing on Android platform and demonstrated that the validity periods are very long, reaching like thousands of years. The results also shows that are differences between data inserted in the digital certificates and registered in the searched virtual store.*

Resumo. *O número de usuários da plataforma Android cresce a cada dia e o número de aplicativos disponíveis cresce em proporção semelhante. Atualmente, cada aplicativo instalado na plataforma Android deve ser assinado digitalmente, porém não existe um controle dos critérios de segurança nos certificados digitais utilizados, deixando para o próprio desenvolvedor a tarefa de selecionar as informações e características do certificado. Este artigo analisou características de segurança de 397 certificados digitais utilizados na assinatura de aplicativos da plataforma Android e demonstrou que os prazos de validade são muitos longos, chegando a alcançar milhares de anos. Os resultados também apresentam que existem diferenças entre os dados inseridos nos certificados digitais e o cadastrados na loja virtual pesquisada.*

1. Introdução

Os smartphones deixaram de ser um dispositivo de comunicação e tornaram-se um componente essencial na vida de muitas pessoas em todo mundo. É isso que a pesquisa do IDC (International Data Corporation) publicada em fevereiro desse ano apresenta. Segundo o IDC, no ano de 2013 foram comercializados mais de 1 bilhão de dispositivos em todo mundo, o que equivale a um aumento de 39,2% quando comparado com o ano de 2012. Segundo a mesma pesquisa do IDC o sistema operacional mais utilizado em smartphones é o sistema Android com 78,6%[IDC 2014]. Com o expressivo aumento dos dispositivos móveis, o número de aplicativos para o sistema operacional Android cresce na mesma ordem. No mês de maio de 2013 eram 782.038 aplicativos disponíveis para download na loja oficial do Google Play. Já no mês de maio de 2014, esse número passou para 1.229.672 aplicativos, um aumento de 57,24% de aplicativos[APPBrain 2014].

A fim de garantir a segurança, durante o processo de instalação de um aplicativo algumas verificações são efetuadas. Entre elas podemos destacar a *Update Integrity*,

que efetua uma verificação quanto a integridade do aplicativo, ou seja, é verificado se não houve nenhuma modificação desde que o código foi assinado pelo seu desenvolvedor. Em seguida o Android decidirá se a instalação se trata de um novo aplicativo ou somente uma atualização. Caso seja uma atualização é feita uma comparação da assinatura digital do aplicativo já instalado com a nova versão. Esse processo garante que somente o desenvolvedor legítimo do aplicativo seja capaz de lançar novas versões do aplicativo[Barrera et al. 2012].

A assinatura digital deve possibilitar que os usuários consigam verificar a autoria de um aplicativo, através da sua associação com o certificado digital. Nesse sentido, a plataforma Android obriga os desenvolvedores a assinar digitalmente os aplicativos antes de disponibilizá-los aos usuários. No entanto, a plataforma Android possibilita que o desenvolvedor utilize certificados auto-assinados na criação da assinatura digital[Gunasekera 2012]. Ao atribuir a tarefa de emissão de um certificado digital ao desenvolvedor, o Android possibilita que ele selecione as informações e características do certificado digital emitido, o que pode resultar no comprometimento da segurança provida pela assinatura digital.

Este artigo tem como objetivo realizar uma análise dos certificados digitais usados na assinatura de aplicações para plataforma Android. Para alcançar o objetivo foi efetuado o *download* de 397 aplicativos da loja Google Play e foi realizado a extração do certificado digital de cada aplicativo. A análise envolveu cinco características dos certificados digitais que podem comprometer a sua segurança ou a credibilidade da assinatura de código.

O restante do artigo está dividido em quatro seções. Na segunda seção são apresentados trabalhos publicados recentemente que discutem a utilização do uso de certificados digitais em aplicativos Android, problemas no modelo atual e outras pesquisas estatísticas relacionadas. A terceira seção apresenta a metodologia. Na quarta seção os autores discorrem sobre os resultados obtidos com a análise dos certificados digitais. Por fim, os autores apresentam suas considerações finais.

2. Trabalhos Relacionados

A segurança proporcionada pelo uso de assinaturas digitais baseadas em certificados digitais como busca de credibilidade vem sendo discutida em diversos trabalhos. No artigo [Barrera and van Oorschot 2011] é apresentado que o desenvolvedor é responsável em efetuar a assinatura de código, sem nenhum envolvimento da loja virtual. Na mesma publicação é abordado que a assinatura digital é utilizada somente em atualizações, uma vez que não existe processo de verificação do desenvolvedor na primeira instalação. No trabalho de [Vargas et al. 2012] os autores apresentam os controles de segurança adotados pelo Android, entre eles é citado a utilização de certificados digitais para a assinatura de código, porém é apresentado que o certificado digital não precisa ser emitido por um autoridade de certificação confiável, ou seja, o próprio desenvolvedor emite o certificado, o que é chamado de certificado auto assinado.

Mesmo utilizando certificados digitais para assinar digitalmente os aplicativos Android, os trabalhos atuais mostram que o modelo atual possui algumas fragilidades. No trabalho [Barrera et al. 2012] os autores citam que o Android tem uma abordagem de confiança no primeiro uso, ou seja, o desenvolvedor não é autenticado na primeira instalação, autenticando somente o desenvolvedor caso ocorra uma atualização, com isso

caso algum aplicativo seja modificado por um usuário mal intencionado, o sistema operacional identificará e não será possível realizar a instalação, porém caso o usuário mal intencionado assine a aplicação com outro certificado será possível realizar a instalação novamente, pois será considerada uma nova instalação. No trabalho publicado por [Zheng et al. 2014] os autores apresentam uma análise dos vários *firmware* que são baseados em Android e alerta para a possibilidade dessas versões possuírem vulnerabilidades. Os autores citam uma vulnerabilidade, *Master Key*, que o usuário mal intencionado explora a falta de verificação de nomes duplicados nas entradas dos arquivos .apk e então são criados dois arquivos com o mesmo nome, contornando a verificação de assinatura realizada durante o processo de instalação e atualização de um aplicativo.

Atualmente estão sendo desenvolvidas diversas análises estatísticas quanto ao uso de aplicativos na plataforma Android. Em [Fahl et al. 2012] é realizada uma análise de 13.500 aplicativos gratuitos mais populares do Google Play procurando identificar vulnerabilidades que permitam ataques *Man-in-the-Middle* e foi identificado que 8% dos aplicativos examinados estavam potencialmente vulnerável. No trabalho de [Enck et al. 2011] foi analisado o código fonte de 1.100 aplicativos gratuitos mais populares da loja virtual do Google. A análise descobriu o uso generalizado e indevido de identificadores pessoais do telefone, porém não foram encontradas vulnerabilidades exploráveis de *malware* nas aplicações estudadas.

Embora os trabalhos apresentados realizem análises de vulnerabilidades associadas aos aplicativos da plataforma Android, os trabalhos não analisam as características e os dados inseridos no certificado digital utilizado na assinatura do aplicativo. A inclusão de dados de forma indiscriminada e o uso de certificados auto assinados podem comprometer a credibilidade do aplicativo. A contribuição deste trabalho é avaliar características de cinco campos dos certificados digitais usados na assinatura de aplicativo e analisar seu impacto na credibilidade do aplicativo.

3. Metodologia

A primeira etapa do trabalho consistiu em definir os critérios de análise dos certificados digitais, para isso foi selecionada uma amostra inicial de certificados digitais usados na assinatura de aplicações Android. A partir disso foi realizada uma análise dos valores atribuídos aos campos da estrutura x.509 que poderiam resultar na perda de credibilidade da aplicação. Com base nesta amostra foram selecionados os seguintes critérios para análise: 1) Prazo de validade: Análise do tempo, em anos, que o certificado permanecerá válido; 2) Tamanho da chave: Analisado qual o tamanho da chave pública; 3) Possibilidade de revogação do certificado: Analisado se o certificado possuía alguma informação que possibilita a consulta do seu *status* de revogação; 4) Auto assinado: Verificado se o certificado é auto assinado ou emitido por um autoridade certificadora; e 5) Dados do Titular: Nesse campo foram verificados se o Common Name (CN) e/ou o campo Organization (O), presentes no campo Nome do titular, são condizentes com o nome do desenvolvedor publicado na loja virtual do Google Play.

Na segunda etapa, foi definido o tamanho da amostra de certificados digitais analisados. Para isso foi definido um nível de confiança de 95% e com margem de erro de 5 pontos percentuais. Para atingir os critérios acima foi necessário uma amostra de 385 certificados digitais. Para seleção dos aplicativos da amostra, os autores usaram como

critério o *ranking* dos aplicativos gratuitos mais populares na Google Play. A partir disso foi extraída uma listagem com 410 aplicativos gratuitos mais populares.

A etapa seguinte, consistiu em coletar os certificados digitais das aplicações que fizeram parte da amostra. Para isso, inicialmente foi realizado o *download* dos aplicativos listados na etapa anterior. Os aplicativos listados, foram baixados em um dispositivo Android que foram instalados. Após a instalação foi efetuado o *backup* do aplicativo para que fosse possível ter o aplicativo em extensão .apk. De posse dos aplicativos na extensão .apk, eles foram transferidos para um computador e descompactado. Uma vez possuindo os arquivos descompactados, foi necessário o uso do OpenSSL para extrair o certificado digital. Nessa etapa foram descartados 13 aplicativos devido a impossibilidade de efetuar o *download* por não estarem mais disponíveis na loja virtual.

Na quarta etapa foi realizada a análise e tabulação dos dados coletados. Para isso foi gerado uma planilha com os 397 aplicativos preenchida com os dados conforme realizada a análise individual dos certificados digitais. Por fim a última etapa consistiu em realizar uma análise dos resultados obtidos. Para isso foi realizado uma análise de modo quantitativo onde foram efetuados cálculos de porcentagem, cruzamento de informações e médias afim de identificar o impacto na credibilidade dos aplicativos.

4. Resultados

Nesta seção é realizada a discussão dos resultados obtidos a partir da análise dos 397 certificados digitais. A discussão foi dividida em subseções que abordam as diferentes análises realizadas. Em cada subseção, após a análise dos dados, os autores apresentam algumas consequências das características presentes nos certificados digitais.

4.1. Análise do Prazo de Validade e Comprimento da Chave

A primeira análise levou em consideração duas variáveis: o período de validade e o comprimento da chave. Inicialmente, foi realizada uma análise isolada do período de validade, com o objetivo de conhecer o tempo de duração dos certificados digitais usados em assinatura aplicativos na plataforma Android. Para essa análise os autores definiram três intervalos de tempo de validade: até 50 anos; entre 51 e 100 anos; e acima de 100 anos. Posteriormente, foram utilizados testes t para diferença entre duas médias, com amostras independentes, para verificar se há diferença significativa no comprimento das chaves públicas dos certificados digitais nos três intervalos de tempo definidos. A Tabela 1 apresenta os resultados desta análise.

Tabela 1. Prazo de validade x Tamanho de chave pública

Prazo de Validade	Certificados	% do total	Intervalo de confiança	Comprimento médio da chave	σ
1 -50	235	59,2	54,4 - 64,0	1.436,26	624,55
51- 100	105	26,4	22,1 - 30,8	1.149,59	337,52
Mais de 100	57	14,4	10,9 - 17,8	1.611,85	785,72

A análise constatou que 235 (59,2% do total) certificados digitais dos aplicativos possuem prazo de validade de 50 anos ou menos. Ao considerar um período maior de validade dos certificados digitais, entre 51 e 100 anos, a análise identificou que 105 (26,4%

do total) certificados digitais continham essa característica. Uma terceira verificação identificou certificados digitais com um período de validade superior a 100 anos, onde foram encontrados 57 certificados digitais (14,4% do total).

Com base nesses dados é possível concluir, com 95% de confiança, que entre 54,4% e 64% dos certificados digitais utilizados na assinatura de código de aplicativos possuem um prazo de validade igual ou inferior à 50 anos; entre 22,1% e 30,8% dos certificados digitais têm prazo de validade entre 51 e 100 anos; e, por fim, entre 10,9% e 17,8% dos certificados digitais têm prazo de validade acima de 100 anos.

Ainda sobre os período de validade dos certificados digitais, a análise constatou que apenas um aplicativo foi assinado digitalmente com um certificado digital com tempo de validade inferior a 10 anos. Por outro lado, 28 aplicativos foram assinados com certificados digitais que possuem um período de validade de 999 anos e 16 aplicativos foram assinados com certificados com período de validade de 1000 anos ou superior. O maior tempo de validade encontrado em um certificado digital da amostra foi 2738 anos.

Embora no site do Android, ao apresentar uma estratégia para assinatura de aplicações, recomende o uso de certificados digitais com um período de validade de 25 anos ou mais, garantindo assim a assinatura das atualizações da aplicação durante o seu ciclo de vida [Android 2014], essa característica pode resultar em uma vulnerabilidade no médio ou longo prazo, pois o aumento da capacidade de processamento poderá reduzir a quantidade de tempo necessário para comprometer as chaves do certificado digital.

Ao comparar o comprimento médio da chave pública, os certificados digitais com prazo de validade superior a 100 anos possuem o tamanho médio da chave pública foi de 1611,85, com desvio-padrão de 785,72. Já nos certificados com prazo de validade inferior a 100 anos, o tamanho médio da chave pública foi de 1349,41, com desvio-padrão de 565,67. Ao nível de 5% de significância, pode-se afirmar que essa diferença é significativa, ou seja, aplicativos com validade maior tem, em média, tamanho de chave menor.

O teste realizado no tamanho das chaves demonstra novos elementos que podem resultar em vulnerabilidades na utilização dos certificados digitais usados na assinatura de aplicações Android no médio ou longo prazo. A partir dela é possível conhecer que os certificados digitais emitidos com uma validade maior estão associados a chaves com tamanhos menores. Ou seja, além dos responsáveis pela emissão estenderem significativamente o prazo de validades dos certificados digitais, eles estão usando como base o comprimento mínimo das chaves consideradas seguras na atualidade.

4.2. Comparação dos dados dos Certificados Digitais com informações da loja

Uma segunda análise comparou os dados contidos no certificado digital do desenvolvedor usado na assinatura de aplicativos com os dados cadastrados por ele na Google Play. Esta análise levou em consideração a identificação do desenvolvedor do aplicativo, apresentada pela Google Play no momento do *download*, em relação a duas informações do certificado digital que deveriam estar associadas a ela: o nome e a organização requerente. Nesta análise foram considerados apenas os certificados digitais que continham nome ou organização do requerente, pois muitos não possuíam tais dados. Por isso, o número de certificados digitais das análises distingue do tamanho da amostra

Na amostra de aplicativos analisados, um total de 94 (23,7%) possuem o nome do

requerente do certificado digital relacionado ao nome da empresa divulgada na loja virtual e 249 (62,7%) certificados digitais, o nome do requerente não está relacionado. Por fim, em 54 (13,6%) certificados digitais não consta a informação do nome do requerente. Com base nesses dados é possível concluir, com 95% de confiança, que entre 57,9% e 67,5% do aplicativos são assinados com certificados digitais que não possuem o nome do requerente relacionado ao nome da empresa divulgada na loja virtual.

Na amostra de aplicativos, em 211 (53,1%) o nome da organização do requerente do certificado digital está relacionado ao nome da empresa divulgada na loja virtual e em 145 (36,5%), o nome não está relacionado ao nome da empresa divulgada na loja virtual. Além disso, 41 (10,3%) certificados digitais não possuem a informação da organização do requerente. Com base nesses dados pode-se concluir, com 95% de confiança, que entre 31,8% e 41,2% dos aplicativos são assinados com certificados digitais cujo o nome da organização requerente contida nele não está relacionado ao nome da empresa divulgada na loja virtual.

Em um segundo momento, a análise associou os três intervalos de período de validade dos certificados digitais com a relação existente entre os dados dos requerentes dos certificados digitais e os apresentados na loja virtual Google Play. Para definir se há ou não a existência de associação entre o período de validade do certificado e a relação entre as informações do certificado e da loja virtual foi utilizado o teste qui-quadrado, onde foi considerada uma associação significativa probabilidades superiores a 5%. A Tabela 2 apresenta o resultado desse cruzamento.

Tabela 2. Prazo de validade x Nome do requerente

Prazo de Validade (Anos)	Associação entre dados do certificado e loja virtual			
	Possui relação entre nome do requerente e site		Possui relação entre organização do requerente e site	
	Sim	Não	Sim	Não
1-50	59 (29,8%)	139 (70,2%)	132 (62,9%)	78 (37,1%)
50-100	20 (21,5%)	73 (78,5%)	51 (52,6%)	46 (47,4%)
>100	15 (28,8%)	37 (71,2%)	28 (57,1%)	21 (42,9%)

Ao analisar os certificados digitais que não possuem o nome do requerente relacionado ao nome da empresa divulgada na loja virtual, foram identificados 136 (67,7%) certificados com validade entre 1 e 50 anos, 74 (78,7%) certificados digitais entre 50 e 100 anos e 35 (72,9%) certificados digitais com mais de 100 anos. A partir desses dados não foi possível perceber uma associação significativa entre prazo de validade e o nome do requerente contido nele estar relacionado ao nome da empresa divulgada na loja virtual.

Ao analisar os certificados digitais que não possuem o nome da organização requerente relacionado ao nome da empresa divulgada na loja virtual, foram identificados 76 (36,7%) certificados com validade entre 1 e 50 anos, 47 (48%) certificados digitais entre 50 e 100 anos e 19 (41,3%) certificados digitais com mais de 100 anos. A partir desses dados não foi possível perceber uma associação significativa entre o prazo de validade e o nome da organização requerente contido nele estar relacionado ao nome da empresa divulgada na loja virtual.

A partir dessas análises é possível concluir que não há um cuidado em relacionar

as informações do certificado digital usado na assinatura das aplicações Android com os dados cadastrados na loja virtual. A partir disso é possível concluir que a assinatura de código na plataforma Android não está cumprindo o seu papel de atribuir credibilidade às aplicações, pois o usuário não consegue relacionar os dados da identidade usada na assinatura com a empresa ou desenvolvedor responsável por ela.

4.3. Dados de Revogação e modelo de confiança

A análise do item Dados de Revogação levou em consideração se o certificado utilizado na assinatura do aplicativo continha alguma informação de revogação do mesmo. A análise verificou se o certificado possuía um link que possibilitasse a revogação, independentemente do protocolo utilizado. Para realizar a análise do modelo de confiança foi verificado se o certificado utilizado na assinatura digital era auto assinado. A análise constatou que nenhum dos certificados analisados possuía algum dado que possibilitasse a revogação de um certificado utilizado para a assinatura de aplicações da plataforma Android. Essa característica pode ser considerada uma vulnerabilidade de segurança, pois caso algum desses certificados seja comprometido, não será possível verificar a sua revogação.

Para efetuar a análise do modelo de confiança foi verificado o nome do requerente do certificado e o nome do emissor do certificado. Se ambos os nomes fossem iguais o certificado era auto assinado. A análise constatou que todos os certificados eram auto assinados, ou seja todos os certificados utilizados para a assinatura de código foram emitidos pelos próprios desenvolvedores comprometendo a segurança proporcionada pela assinatura digital.

5. Considerações Finais

O presente artigo realizou uma análise de 397 certificados digitais utilizados na assinatura digital dos aplicativos gratuitos mais populares do Google Play. Após a análise foi possível constatar que os certificados digitais possuem prazos de validade muito longos, como, por exemplo o fato de 44 certificados possuírem o período de validade de 999 anos ou mais. Quando comparado a validade dos certificados digitais com o tamanho da chave constatou-se que quanto maior o prazo de validade menor o tamanho da chave, ou seja, além dos responsáveis pela emissão estenderem significativamente a validade dos certificados digitais, eles estão usando como base o comprimento mínimo das chaves consideradas seguras na atualidade.

Quando comparado os dados dos certificados digitais com informações da loja, constatou-se que não existe um cuidado por parte dos desenvolvedores em relacionar as informações contidas no certificado digital com as informações cadastradas na loja virtual, pois verificou-se que somente 23,7 % dos certificados analisados possuem o nome do requerente do certificado digital relacionado ao nome divulgado na loja virtual e que 53,1% possuem o nome da organização requerente do certificado digital relacionado ao nome divulgado na loja virtual. Portanto pode-se concluir que a assinatura de código na plataforma Android não está cumprindo o seu papel, que é atribuir credibilidade às aplicações, pois o usuário não consegue relacionar os dados entre o certificado digital e as informações divulgadas na loja virtual.

Por fim foi identificado que nenhum dos certificados utilizados para assinatura de aplicativos da plataforma Android possuía dados de revogação. É possível afirmar

que essa característica pode ser considerada uma vulnerabilidade no modelo de confiança adotado pela plataforma Android, pois caso um desenvolvedor tenha a segurança do seu certificado digital comprometida, ele não possui meios de tornar o certificado inválido.

Novas análises poderão ser iniciadas a partir dos resultados desta pesquisa, pois ainda é escasso o número de trabalhos que analisam o uso de certificados digitais nas plataformas em uso em dispositivos móveis. Como trabalho futuro é sugerida a análise dos certificados digitais usados em assinatura de aplicações em outras plataformas ou é possível realizar a análise de aplicativos de segmentos específicos na plataforma Android, como, por exemplo, aplicativos corporativos. Além disso pode-se sugerir um novo modelo de confiança para o uso de assinatura para aplicativos de plataforma Android.

Referências

- Android, D. (2014). Signing Your Applications. <http://developer.android.com/tools/publishing/app-signing.html>. Acessado em 2014-07-15.
- APPBrain (2014). APPBrain,Number of Android applications. <http://www.appbrain.com/stats/number-of-android-apps>. Acessado em 2014-05-29.
- Barrera, D., Clark, J., McCarney, D., and van Oorschot, P. C. (2012). Understanding and improving app installation security mechanisms through empirical analysis of android. In *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, SPSM '12, pages 81–92, New York, NY, USA. ACM.
- Barrera, D. and van Oorschot, P. (2011). Secure Software Installation on Smartphones. *Security & Privacy Magazine*, 9(3):42–48.
- Enck, W., Octeau, D., McDaniel, P., and Chaudhuri, S. (2011). A study of android application security. In *Proceedings of the 20th USENIX Conference on Security*, SEC'11, pages 21–21, Berkeley, CA, USA. USENIX Association.
- Fahl, S., Harbach, M., Muders, T., Baumgärtner, L., Freisleben, B., and Smith, M. (2012). Why eve and mallory love android: An analysis of android ssl (in)security. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 50–61, New York, NY, USA. ACM.
- Gunasekera, S. (2012). *Android Apps Security*. Apress, Berkely, CA, USA, 1st edition.
- IDC (2014). Android and iOS Continue to Dominate the Worldwide Smartphone Market with Android Shipments Just Shy of 800 Million in 2013, According to IDC. <http://www.idc.com/getdoc.jsp?containerId=prUS24676414>. Acessado em 2014-05-29.
- Vargas, R., Huerta, R., Anaya, E., and Hernandez, A. (2012). Security controls for android. In *Computational Aspects of Social Networks (CASoN), 2012 Fourth International Conference on*, pages 212–216.
- Zheng, M., Sun, M., and Lui, J. C. (2014). Droidray: A security evaluation system for customized android firmwares. In *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security*, ASIA CCS '14, pages 471–482, New York, NY, USA. ACM.

Uma análise dos certificados digitais utilizados nas conexões TLS dos aplicativos de Mobile Banking na plataforma Android

**Diego Baierle Sebastiany¹, Mirelle Daiara Vieira Freitas¹,
Luciano Ignaczak¹**

¹ Universidade do Vale do Rio dos Sinos (UNISINOS)
CEP 93.022-000 – São Leopoldo – RS – Brasil

diego.sebastiany@hotmail.com, mdfreitass@outlook.com, lignaczak@unisinos.br

Abstract. *Currently, it is increasingly common to use mobile banking applications on smartphones. Such applications must implement TLS to use encryption to secure communication between the client and the bank. However, often the applications have developmental problems that compromise their safety. This article analyzes whether the digital certificates used by banks from three countries in TLS connections on m-banking applications are recognized as trusted by Android. Furthermore, the size of keys and the validity of these digital certificates is discussed.*

Resumo. *Atualmente, é cada vez mais comum a utilização de aplicativos de mobile banking em smartphones. Tais aplicativos devem implementar o protocolo TLS para empregar criptografia para proteger a comunicação entre o cliente e o seu banco. No entanto, muitas vezes os aplicativos apresentam problemas de desenvolvimento que comprometem a sua segurança. Este artigo analisa se os certificados digitais usados por bancos de três países nas conexões TLS com as aplicações de m-banking são reconhecidos como confiáveis pelo Android. Além disso, é discutido o tamanho das chaves e o período de validade desses certificados digitais.*

1. Introdução

A crescente popularização da *internet* tem levado a um aumento expressivo da quantidade de dispositivos conectados. Com isso, cresceu também a quantidade de usuários que utilizam aplicativos para gerenciamento e movimentação financeira de suas contas bancárias. Uma pesquisa mostra que 52% das transações bancárias feitas no Brasil em 2014 foram realizadas via *internet* e *mobile banking* (m-banking). Entre as contas ativas no país em 2014, 24% dos clientes (25 milhões) realizaram transações utilizando m-banking em seus *smartphones* [Febraban 2015].

Os aplicativos de m-banking precisam implementar mecanismos de segurança para garantir que os dados do usuário não fiquem vulneráveis a roubo e interceptação na Internet. O protocolo TLS (*Transport Layer Security*) é utilizado como padrão para fornecer a segurança nesse ambiente [Elkhodr et al. 2012]. Além de garantir o sigilo e a integridade na comunicação, o protocolo TLS autentica o servidor do banco no qual o aplicativo está se conectando. A autenticação é importante e necessária

para confirmar que o computador que está respondendo é realmente a entidade que afirma ser [Stallings 2008] [Adams and Lloyd 2003].

O estabelecimento de uma comunicação segura exige que a autenticação do banco seja feita de forma correta. Ao fazer o *handshake* do protocolo TLS, o banco envia o seu certificado digital para que o cliente (o aplicativo) verifique a sua identidade. Para isso, o aplicativo deve utilizar um dos certificados raiz instalados no sistema Android para fazer a validação da confiança do certificado do banco. A tentativa de conexão deveria falhar se o certificado raiz usado pelo banco não for considerado confiável pelo Android. No entanto, muitas vezes os aplicativos falham ou simplesmente não executam a validação do certificado digital [Six 2012].

Além da validação da confiança, o tamanho da chave criptográfica e o período de validade de um certificado digital são muito importantes e devem ser considerados no momento de sua emissão para conferir-lhe resistência contra ataques de força bruta. As recomendações mais recentes sugerem um tamanho de chave mínimo de 2.048 bits [Barker and Roginsky 2011]. Além disso, a Microsoft recomenda que certificados digitais com tamanho da chave de 1.024 bits devem possuir até 1 ano de validade; certificados digitais com tamanho de chave de 2.048 bits devem possuir no máximo 2 anos de validade; e certificados digitais com tamanho de chave de 4.096 bits podem possuir validade de até 16 anos [OMeally 2009].

O objetivo deste trabalho é analisar a confiança dos certificados digitais utilizados por bancos nas conexões TLS com os aplicativos de m-banking na plataforma Android, além do período de validade e o tamanho da chave criptográfica desses certificados. A análise foi realizada a partir de uma amostra de 60 aplicativos de m-banking disponibilizados por bancos de três países: Brasil, Estados Unidos e Reino Unido. Para cada aplicativo foi realizada uma simulação de acesso à conta bancária e, a partir do tráfego capturado, foi verificada a utilização do TLS e obtidos os certificados digitais utilizados na conexão.

O restante deste trabalho segue com a seção 2 que apresenta alguns trabalhos relacionados. A seção 3 descreve a metodologia utilizada na realização desta análise, a seção 4 mostra os resultados obtidos da análise dos aplicativos de m-banking e a seção 5 expõe as considerações finais deste trabalho.

2. Trabalhos Relacionados

Muitos aplicativos vêm apresentando problemas de implementação, os quais têm motivado muitos trabalhos que discutem suas causas e possíveis soluções. O trabalho de [Georgiev et al. 2012] mostra que a segurança oferecida pelo TLS depende da correta validação do certificado digital fornecido quando a conexão é estabelecida. Esse trabalho analisa como alguns *softwares* e aplicativos implementam as funções do TLS para validação dos certificados digitais e mostra que mesmo os aplicativos desenvolvidos por grandes empresas possuem falhas graves. Muitas vezes, segundo os autores, as falhas na validação de um certificado é causada pela falta de entendimento e interpretação das APIs (*Application Programming Interface*) utilizadas pelos desenvolvedores. A falta de conhecimento e informação sobre essas APIs conduz o desenvolvedor ao erro e deixa o aplicativo vulnerável a ataques do homem do meio. O trabalho de [Hubbard et al. 2014] realizou uma pesquisa com o objetivo de

identificar falhas na validação dos certificados digitais. Com uma pequena amostra de 41 aplicativos para a plataforma Android, 11 falharam em estabelecer a relação de confiança necessária, pois aceitaram um certificado digital falsificado que, portanto, não pertencia à base de confiança do Android. O artigo também destaca que a falha dos aplicativos pode estar relacionada às APIs utilizadas. Por serem pouco restritivas, permitem que os desenvolvedores cometam erros de implementação do código, permitindo que qualquer certificado digital seja aceito pelo aplicativo ou, até mesmo, que nenhuma validação seja realizada.

A inconsistência da base de certificados raiz da plataforma Android também já foi alvo de estudo. [Vallina-Rodriguez et al. 2014] examinou os certificados raiz instalados nas diversas versões do Android em vários dispositivos. O trabalho analisou a composição dessas bases de confiança e como elas variam de acordo com a versão e marca do dispositivo. Como resultado, foi verificado que a base oficial de certificados digitais confiados pelo Android é modificada ou ampliada. Em alguns casos, o próprio fabricante do dispositivo e/ou a operadora de telefonia adicionam certificados digitais aos dispositivos para estabelecer a relação de confiança para aplicativos embarcados ou prestação de serviços. O trabalho também alerta para o fato que, em dispositivos que rodam com usuário *root*, aplicativos maliciosos podem instalar certificados digitais no Android sem o conhecimento do usuário, quebrando o modelo de confiança de certificados digitais supervisionados e auditados como confiáveis.

[Fahl et al. 2012] investigou o uso inadequado do TLS em 13.500 aplicativos de diversas categorias, obtidos da Google Play Market. Dos aplicativos analisados, 1.074 (17,28% dos que utilizam TLS) continham erros de código do TLS que permitiam a validação de qualquer certificado digital ou confiavam em qualquer certificado raiz. O autor mostra também que as falhas na implementação do TLS ocorrem porque o Android permite que os desenvolvedores criem códigos personalizados para seus aplicativos. Ele destaca que esse recurso devia ser desativado e que as APIs para Android deviam forçar a utilização das implementações padrão do TLS. Em outro trabalho, [Fahl et al. 2013] continua investigando as possíveis causas da má implementação do TLS em aplicativos. Os resultados da pesquisa mostram que as causas não são simplesmente a falta de cuidado por parte dos desenvolvedores, mas também questões e limitações envolvendo o atual paradigma de desenvolvimento do TLS. O trabalho sugere mudanças no atual paradigma em direção a uma maior abstração do código fornecido pelas APIs, permitindo que desenvolvedores utilizem corretamente o TLS com menos esforço e prevenindo falhas na validação dos certificados digitais.

Os trabalhos relacionados reforçam a necessidade de um maior cuidado na implementação do TLS em aplicativos que transmitem dados confidenciais. As falhas de implementação em aplicativos de m-banking podem acarretar muitos prejuízos para o cliente e para o banco. Os artigos citados nesta seção realizaram análises dos certificados digitais de diversos aplicativos, sem abordar um segmento específico. Já este artigo, analisou especificamente como aplicativos de m-banking estão validando os certificados digitais dos bancos.

3. Metodologia

Para a realização desta análise foi selecionada uma amostra com 60 aplicativos de m-banking divididos igualmente em três países: Brasil, Estados Unidos (EUA) e o Reino Unido (UK). A seleção dos aplicativos foi realizada utilizando *rankings* do Banco Central do Brasil¹, do *Federal Reserve System*² para os EUA, e do Relbanks³ para o UK, que classificam os bancos com maiores ativos em cada país. Baseados nestes *rankings*, os autores selecionaram os 20 primeiros bancos de varejo que possuem aplicativos de m-banking. A lista da Relbanks possui apenas 11 bancos e foi utilizada porque não foi encontrado um *ranking* oficial do Banco Central do Reino Unido. A amostra de aplicativos do país foi incrementada com mais 10 bancos conhecidos do Reino Unido, retirados do site do seu Banco Central⁴. A análise consistiu na avaliação das seguintes características de cada aplicativo:

- se o aplicativo utiliza o protocolo TLS para comunicação segura;
- a verificação da confiança no certificado raiz do aplicativo;
- o período de validade do certificado digital do aplicativo;
- o tamanho da chave do certificado digital do aplicativo;

O *software* Genymotion⁵ foi usado para emular um dispositivo rodando a versão 4.4 do sistema Android, que está instalada atualmente em 39,3% dos dispositivos dessa plataforma [Android 2015]. Desse dispositivo foram extraídos todos os certificados digitais armazenados em `/system/etc/security/cacerts/`. Esses são os certificados digitais das autoridades de certificação confiadas por esta versão do Android. Os certificados digitais extraídos foram armazenados para, posteriormente, analisar a confiança dos certificados raiz utilizados nas conexões TLS pelos aplicativos de m-banking. Para possibilitar a análise, os aplicativos de m-banking selecionados foram instalados no dispositivo virtual. Além disso, para que fosse possível a captura do tráfego TLS gerado pelo aplicativo de m-banking foi utilizado o `tcpdump`, disponível no emulador.

Após a instalação de cada aplicativo de m-banking, foram realizadas tentativas de acesso à conta bancária. O acesso foi simulado pela inserção dos dados necessários (como número da conta e senha) aceitos pelo aplicativo, para que ele iniciasse a comunicação com o banco, e assim, estabelecesse a conexão segura (TLS). Com o tráfego gerado pela simulação do acesso foi avaliado o primeiro critério desta análise: se o aplicativo utiliza o TLS.

No caso dos aplicativos de m-banking que possibilitaram a verificação da implementação do protocolo TLS com a captura do tráfego foram extraídos os certificados digitais utilizados pelo *handshake*: o certificado do banco e o certificado raiz, que é utilizado para avaliar a relação de confiança entre o banco e o sistema Android. A partir do certificado digital do banco foi avaliado o tamanho da chave criptográfica bem como o seu período de validade. Para auxiliar na consolidação dos resultados dessa análise foi utilizado um *software* desenvolvido pelos autores,

¹Disponível em: <http://www4.bcb.gov.br/top50/port/top50.asp>

²Disponível em: <http://www.federalreserve.gov/Releases/Lbr/current/default.htm>

³Disponível em: <http://www.relbanks.com/europe/uk>

⁴Disponível em: <http://www.bankofengland.co.uk>

⁵Disponível em: <https://www.genymotion.com>

na linguagem C#, que coleta os dados dos certificados e exporta os resultados no formato XML. O arquivo exportado foi utilizado como fonte para a construção de uma planilha.

Um segundo *software* na linguagem C# também necessitou ser desenvolvido pelos autores para analisar a confiança dos certificados raiz capturados. Esse *software* realizou o cruzamento entre os certificados raiz capturados e a base de certificados raiz considerada confiável pela versão avaliada do Android. O cruzamento desses certificados digitais consistiu em comparar os campos *Subject Key Identifier*, ou na ausência deste, a própria chave pública contida no campo *Subject Public Key Info*. A saída desse programa foi salva e adicionada à planilha anterior, usada como base para a avaliação dos resultados.

Não foi possível avaliar alguns aplicativos de m-banking pois a captura do tráfego desses aplicativos no momento da autenticação não apresenta a utilização do TLS, tampouco revela os dados do usuário em texto claro. Isso pode acontecer quando o aplicativo implementa os requisitos de segurança na camada de aplicação. Por isso, não é possível afirmar que o aplicativo falha em oferecer segurança para o usuário. Os aplicativos com essas características foram classificados como indefinidos.

A última etapa desse trabalho consistiu na realização da análise dos resultados obtidos. Nesta etapa foram efetuados cálculos de porcentagem, cruzamento de informações e médias, a fim de comparar as definições dos bancos dos três países em relação aos dados dos certificados digitais que são alvo deste artigo.

4. Resultados

A análise dos 60 aplicativos selecionados resultou em 2 aplicativos, ambos do Brasil, classificados como indefinidos, e 58 aplicativos que utilizaram o TLS para estabelecer a conexão segura.

Não foi possível verificar a utilização do TLS ao analisar a captura do tráfego gerado pelos 2 aplicativos que foram classificados como indefinidos. Embora não seja possível afirmar, o mecanismo de segurança utilizado por esses aplicativos pode ser o próprio TLS, mas implementado de forma personalizada pelos desenvolvedores. Isso é possível porque as APIs utilizadas para o Android permitem esse nível de personalização do código.

O resultado mostrou que os outros 58 aplicativos analisados utilizam o TLS, realizando o *handshake* e apresentando o certificado digital do banco como é padrão do protocolo. Porém, 18 (31%) desses aplicativos não poderiam ser considerados confiáveis, pois esses utilizam certificados digitais emitidos por autoridades de certificação que não são confiadas pelo Android. O resultado dessa verificação é apresentado na Tabela 1.

A análise da relação de confiança mostrou que o cenário mais preocupante é o brasileiro, onde 44% dos aplicativos analisados não podem ser considerados confiáveis pela versão da plataforma Android analisada. O Reino Unido apresentou o menor número de certificados digitais não confiáveis (15%), porém, ainda é preocupante considerando que o segmento analisado é o bancário, que deveria possuir um cuidado adicional no uso de certificados digitais.

Tabela 1. Certificados raiz sem relação de confiança com o Android.

Origem	Total de certificados raiz não confiáveis	Percentual de certificados raiz não confiáveis
Brasil	8	44,44%
Estados Unidos	7	35,00%
Reino Unido	3	15,00%
TOTAL	18	31,03%

Como foi mostrado pelos trabalhos relacionados, as falhas de validação da confiança expõem o cliente a diversos riscos, e são resultado da forma de implementação do código do aplicativo. Semelhante às análises nesses trabalhos, esta análise dos aplicativos de m-banking revelou um cenário inquietante, pois nenhum dos aplicativos que utilizam certificados não confiados pela plataforma Android apresentou qualquer mensagem de alerta durante o *handshake* do TLS.

A segunda parte desta análise, avaliou o período de validade e o tamanho da chave criptográfica do certificado digital do banco. Todos os 58 certificados digitais possuem o tamanho da chave igual a 2.048 bits com períodos de validade distintos. Os períodos de validade dos certificados digitais usados pelos aplicativos de m-banking são apresentados na Tabela 2.

Tabela 2. Período de validade dos certificados digitais dos bancos.

Origem	Total de certificados	Período de validade			
		1 ano	2 anos	3 anos	4 anos
Brasil	18	10	8	0	0
Estados Unidos	20	14	3	1	2
Reino Unido	20	12	8	0	0

Embora todos os certificados digitais dos bancos analisados atendam à recomendação do NIST no que diz respeito ao tamanho da chave [Barker and Roginsky 2011], 3 deles, todos dos EUA, possuem o período de validade superior a 2 anos. Conforme a recomendação da Microsoft, o período máximo de validade deve ser de 2 anos para certificados com tamanhos de chave de 2.048 bits. Um período de validade muito grande diminui a resistência da chave associada ao certificado digital, pois os avanços da tecnologia de computação podem comprometer um certificado digital que, para os padrões atuais, é considerado forte.

5. Considerações Finais

Quando o usuário instala e utiliza um aplicativo em seu *smartphone*, ele o faz confiando que a comunicação e seus dados estarão seguros. Quando se trata do segmento de m-banking, espera-se que todos os aplicativos implementem o TLS para atender os requisitos de segurança e proteger o usuário. Ao usuário resta apenas confiar no aplicativo, pois o sistema Android não oferece nenhuma indicação de que a comunicação é estabelecida de forma segura.

Existem normas e recomendações que os desenvolvedores de aplicativos devem seguir para atender requisitos no desenvolvimento de seus aplicativos e evitar erros comuns ao utilizar códigos personalizados. O segmento de m-banking deve observar especialmente as recomendações de segurança como a do NIST [Barker and Roginsky 2011] que especifica o tamanho mínimo da chave do certificado digital em 2.048 bits. Como foi mostrado nos resultados deste trabalho, todos os aplicativos seguiram essa recomendação pois todos possuem tamanho da chave igual a 2.048 bits. No entanto, 3 desses certificados digitais possuem o período de validade maior que 2 anos, em desacordo com a recomendação da Microsoft [OMeally 2009]. Isso pode resultar em uma falha, pois os avanços da tecnologia de computação poderão permitir a quebra de chaves com tamanho de 2.048 bits durante o período de validade do certificado digital.

Ademais, uma parcela significativa dos aplicativos, considerando-se sistemas de m-banking, falham na implementação da validação do certificado digital porque a relação de confiança que deveria existir entre o certificado raiz do banco e o sistema Android não é estabelecida. Sem a validação da confiança um certificado digital é aceito sem qualquer restrição, quebrando completamente o sistema de certificação digital, supervisionado e auditado como confiável. Esse problema mostra-se ainda mais grave quando considerado que isso ocorre de forma transparente para o usuário. Embora o protocolo TLS forneça o recurso para avisar o usuário que a relação de confiança não foi estabelecida, muitas vezes esse recurso é desativado ou mau implementado pelo desenvolvedor. Neste trabalho, dos 18 aplicativos que falharam ao estabelecer a relação de confiança, nenhum mostrou qualquer mensagem de aviso sobre essa falha, e todos prosseguiram funcionando como se nenhum erro tivesse ocorrido.

Este trabalho analisou uma amostra reduzida de certificados digitais utilizados por aplicativos de m-banking durante a conexão TLS. Como trabalho futuro é sugerido a análise de uma amostra mais ampla que reflita com mais precisão a realidade no segmento dos aplicativos de m-banking. Além disso, trabalhos futuros podem comparar as características de certificados digitais usados no TLS por aplicativos de outros segmentos.

Referências

- [Adams and Lloyd 2003] Adams, C. and Lloyd, S. (2003). *Understanding PKI: Concepts, Standards, and Deployment Considerations*. Addison-Wesley Pearson Education, Boston, MA, second edition.
- [Android 2015] Android, D. (2015). Dashboards, platform versions. Disponível em: <https://developer.android.com/about/dashboards/index.html>.
- [Barker and Roginsky 2011] Barker, E. and Roginsky, A. (2011). Transitions: Recommendation for transitioning the use of cryptographic algorithms and key lengths. NIST Special Publication 800-131A. Disponível em: <http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf>.
- [Elkhodr et al. 2012] Elkhodr, M., Shahrestani, S., and Kourouche, K. (2012). A proposal to improve the security of mobile banking applications. In *ICT and*

- Knowledge Engineering (ICT Knowledge Engineering), 2012 10th International Conference on*, pages 260–265.
- [Fahl et al. 2012] Fahl, S., Harbach, M., Muders, T., Baumgärtner, L., Freisleben, B., and Smith, M. (2012). Why eve and mallory love android: An analysis of android ssl (in)security. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 50–61, New York, NY, USA. ACM.
- [Fahl et al. 2013] Fahl, S., Harbach, M., Perl, H., Koetter, M., and Smith, M. (2013). Rethinking ssl development in an appified world. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, pages 49–60, New York, NY, USA. ACM.
- [Febraban 2015] Febraban (2015). Pesquisa febraban de tecnologia bancária 2014. Disponível em: https://www.febraban.org.br/Noticias1.asp?id_texto=2626.
- [Georgiev et al. 2012] Georgiev, M., Iyengar, S., Jana, S., Anubhai, R., Boneh, D., and Shmatikov, V. (2012). The most dangerous code in the world: Validating ssl certificates in non-browser software. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 38–49, New York, NY, USA. ACM.
- [Hubbard et al. 2014] Hubbard, J., Weimer, K., and Chen, Y. (2014). A study of ssl proxy attacks on android and ios mobile applications. In *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th*, pages 86–91.
- [OMeally 2009] OMeally, Y. (2009). Recommendations for pki key lengths and validity periods with configuration manager. Disponível em: <http://blogs.technet.com/b/configmgrteam/archive/2009/06/12/recommendations-for-pki-key-lengths-and-validity-periods-with-configuration-manager.aspx>.
- [Six 2012] Six, J. (2012). *Segurança de aplicativos Android*. Novatec Editora Ltda., São Paulo, SP.
- [Stallings 2008] Stallings, W. (2008). *Criptografia e segurança de redes*. Pearson Education do Brasil Ltda., São Paulo, SP, fourth edition.
- [Vallina-Rodriguez et al. 2014] Vallina-Rodriguez, N., Amann, J., Kreibich, C., Weaver, N., and Paxson, V. (2014). A tangled mass: The android root certificate stores. In *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*, CoNEXT '14, pages 141–148, New York, NY, USA. ACM.

URLBlackList Lite: Uma lista enxuta de catalogação baseada na URLBlackList

Nilson Mori Lazarin, Tielle da Silva Alexandre

CEFET/RJ - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca
Av. Roberto da Silveira, 1900 – 28.635-000 – Nova Friburgo – RJ – Brazil

Abstract. *The URLBlackList is one of the cataloging lists available for content controlling; however, it has domains unsolved and anomalies, which need to be repaired to optimize the content controlling process. Thus, the objective of this work is to present an automatized process of refinement of the URLBlackList. As solution, it is presented a web system capable of segmenting domains from the URLBlackList in two groups: the solved domains and the unsolved domains, besides it purges any anomaly found. After the refinement process, the system will allow the generation of a lean list composed only by solved domains. A comparative and qualitative analysis between the list generated by the URLBlackList Lite and the URLBlackList is also shown.*

Resumo. *As ferramentas de controle de conteúdo, tais como Proxy, são altamente dependentes de uma boa lista de catalogação de sites. Uma das listas disponíveis para download é a URLBlackList, entretanto, ela possui domínios cadastrados que não estão registrados (não resolvíveis via DNS), além de outras anomalias que afetam o custo computacional no processo de controle de conteúdo. Este trabalho apresenta uma ferramenta que através de um processo automatizado de refinamento, da URLBlackList, e segmentação dos domínios pertencentes à mesma em dois grupos: os domínios resolvíveis e os não resolvíveis, além de expurgar qualquer anomalia encontrada. Após o processo de refinamento, a ferramenta possibilita a geração de uma lista enxuta composta somente por domínios resolvíveis. Uma análise comparativa e qualitativa entre a lista gerada pela URLBlackList Lite e a lista da URLBlackList também será apresentada.*

1. Introdução

Um servidor *Proxy* atua como um intermediador entre os computadores de uma rede local e a Internet, analisando todas as requisições recebidas [MORIMOTO 2009]. O processo de filtragem de conteúdo por meio de um servidor *Proxy* ocorre através da comparação das requisições do cliente com uma lista de *Uniform Resource Locator* (URL) ou domínios. Para isso, regras de acesso são configuradas de forma a autorizar ou não o acesso à determinada página solicitada pelo usuário. Quando um servidor *Proxy* recebe uma requisição de uma página, inicia-se um processo de comparação entre a requisição e todas as linhas de domínio presentes na *URLBlackList* [FOROUZAN 2008].

O projeto *URLBlackList* (2015) é uma lista de catalogação composta por um conjunto de listas de URL, domínios e expressões subdivididas em, aproximadamente,

100 categorias. A lista de catalogação da *URLBlackList* foi escolhida para o estudo e a análise desse projeto por ser uma lista de catalogação extensa, popular entre os administradores de redes e por disponibilizar, frequentemente, uma versão gratuita. Entretanto, a lista da *URLBlackList* apresenta domínios catalogados que estão inativos e que consequentemente consomem processamentos desnecessários por parte do servidor *Proxy*.

Outro fato da *URLBlackList* é a sua origem norte-americana, muitas vezes não atendendo o contexto em redes brasileiras e provocando certas distorções de entendimento. Por exemplo, a categoria *Guns* faz referência às armas de pequeno porte e de livre comércio, enquanto que a categoria *Weapons* contempla artefatos de guerra. Como no Brasil qualquer porte de arma deve passar por um processo de legalização, estas categorias poderiam ser unidas adequando-as ao contexto brasileiro e assim, diminuindo a duplicidade de registros na lista.

Na *URLBlackList*, um domínio pode ser encontrado em mais de uma categoria, ocasionando uma redundância de verificação no servidor *Proxy*. Caso seja necessário estabelecer um bloqueio de acesso a conteúdos pornográficos, provavelmente, deverá ser incluso as categorias *Adult*, *Sexuality* e *Porn*, entretanto, vários domínios existem em concomitância nestas categorias (e.g. *sexwork.com*). Além disso, é possível encontrar endereços IP onde deveriam existir apenas domínios. Destaca-se ainda a existência de domínios com anomalias de formato, como por exemplo, *googlex..com* e *relato-sexo..com* (possuem dois pontos consecutivos). Esses domínios ocasionam falhas de resolução por serem tratados como domínios inválidos.

Portanto, o objetivo desse trabalho é apresentar um processo capaz de refinar a lista da *URLBlackList* expurgando ou tratando as adversidades encontradas com intuito de fornecer uma lista de catalogação enxuta que proporcionará a otimização do processo de filtragem de conteúdo realizado por servidor *Proxy*. Para isso uma ferramenta, a *URLBlackList Lite*, foi desenvolvida com o intuito de implementar o processo de refinamento possibilitando gerar, após o refinamento, uma lista catalogada enxuta apenas com domínios resolvíveis. Além disso, a ferramenta desenvolvida possibilita uma reclassificação das categorias da *URLBlackList* a fim de proporcionar uma melhor adequação a diversos contextos.

Este trabalho está estruturado da seguinte forma: na seção 2 será apresentada a *URLBlackList Lite* e a sua metodologia; na seção 3, serão descritos os experimentos realizados com a *URLBlackList Lite* comparados com a *URLBlackList*; a conclusão é apresentada na seção 4.

2. Metodologia

Nesta seção serão apresentados os métodos e as etapas para solucionar os problemas identificados na *URLBlackList*, possibilitando: a eliminação de endereços IP existentes; a segmentação a *URLBlackList* em domínios resolvíveis e não resolvíveis; o tratamento dos domínios com formatos ilegais e as anomalias de resolução; a análise e o tratamento dos domínios redundantes; e a adequação da lista a vários contextos.

A metodologia utilizada no projeto *URLBlackList Lite* é dividida em: processo de refinamento; processo de análise de redundância; e processo de reclassificação das categorias. Os processos de refinamento e análise de redundância serão os responsáveis por refinar, tratar e gerar a *URLBlackList Lite*, enquanto que o processo de

reclassificação adequará a *URLBlackList* ao contexto brasileiro. A figura 1 ilustra a metodologia proposta evidenciando as etapas que um domínio percorre até o final do processo.

2.1. Processo de Refinamento

O processo de refinamento consiste na análise sequencial dos domínios pertencentes à *URLBlackList*, usando a ferramenta DIG do pacote BIND para a resolução de domínio, com o objetivo de segmentar a lista da *URLBlackList* em dois grupos de domínios distintos: os domínios resolvíveis e os não resolvíveis [BIND 2015]. Esse processo tem como *input* a *URLBlackList* descompactada e recebe para processamento um domínio por vez. O processo de refinamento é dividido em duas etapas: a identificação dos IP e a resolução de domínio.

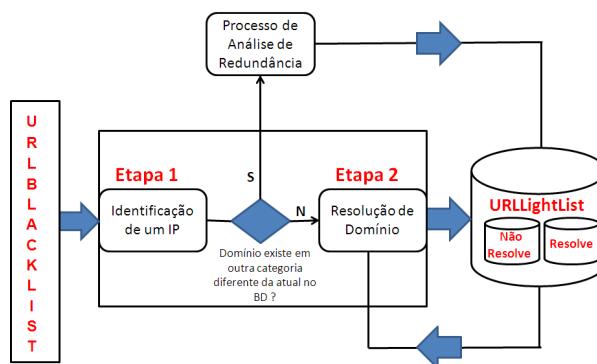


Figura 1. A metodologia da *URLBlackList Lite*.

A identificação de IP visa identificar e expurgar os endereços IP presentes nas listas de domínio da *URLBlackList*. Para isso, todos os registros da *URLBlackList* passam, inicialmente por esta etapa, para que um IP não seja resolvido e pertença a uma lista que deveria conter apenas domínios. Sendo assim, apenas domínios prosseguem para a próxima etapa, onde é identificado se um domínio já existe na base de dados. Se o domínio não existir no armazenamento, o mesmo é direcionado para a segunda etapa do processo de refinamento. Se for identificado que esse domínio já existe no armazenamento, ele é direcionado para o processo de análise de redundância.

A resolução de domínio pode receber como *input* os domínios oriundos de dois casos. No primeiro caso, recebe como *input* os domínios que passaram pela etapa de identificação de IP e não existem no armazenamento. Neste caso, estes domínios são submetidos à etapa de resolução de domínio e se houver uma resposta do comando de resolução (não ocorrer erros), então estes domínios serão direcionados para o armazenamento, sendo relacionados a um dos segmentos (resolvíveis ou não resolvíveis) de uma categoria. Se houver uma ocorrência de erro no processo resolução esses domínios serão expurgados. O resultado desse caso é uma lista enxuta, livre de anomalias e composta por dois segmentos, os domínios resolvíveis e os domínios não resolvíveis.

No segundo caso, o processo de resolução de domínios pode receber como *input* os domínios existentes no banco de dados da *URLBlackList Lite* (Figura 1). Neste caso, cada domínio armazenado passa pela etapa de resolução de domínio podendo mudar de um segmento resolvível para um não resolvível e vice e versa. É importante ressaltar que esse processamento apenas atualiza o *status* de resolução dos domínios a fim de se manter uma base de dados consistente.

2.2. Processo de Análise de Redundância

Após a primeira etapa do processo de refinamento, um mecanismo de identificação de domínios já existentes no armazenamento é realizado direcionando os domínios redundantes para o processo de análise de redundância. Dessa forma, esse processo tem por objetivo tratar as redundâncias de um domínio que estão associados a mais de uma categoria na *URLBlackList*, evitando assim, que um servidor *Proxy* verifique o mesmo domínio mais de uma vez.

O método de tratamento de redundância será embasado na atribuição de pesos para cada categoria existente na *URLBlackList*. O peso é um valor numérico associado à categoria e que representa o grau de perversidade dessa categoria. Entende-se por grau de perversidade: o quanto perverso representa o conteúdo das páginas de uma categoria, em relação às políticas de acesso definidas pelo administrador de redes. As categorias que possuem conteúdos cujos acessos são inadmissíveis pelas políticas de acesso podem receber como peso um valor entre 7 e 10 (perversos). As categorias que possuem conteúdos que são considerados admissíveis pelas políticas de acesso mesmo contendo certas restrições podem receber como peso um valor entre 5 a 6 (moderado). Já as categorias que possuem como peso um valor entre 1 e 4 (baixo), o administrador de redes não possui preocupação em relação ao acesso do usuário a esse conteúdo.

Supondo que o administrador de redes deseja bloquear o acesso a conteúdos pornográficos atribuindo os valores 9, 10 e 10 como peso para as categorias *Sexuality*, *Porn* e *Adult*, respectivamente; e que a análise de redundância ocorra nessa mesma ordem. O domínio *sexinfo101.com* se encontra na lista de domínio destas três categorias. Sendo assim, a primeira vez que esse domínio passar pelo processo de refinamento será resolvido e direcionado para o armazenamento pertencendo ao segmento de domínios resolvíveis da categoria *Sexuality*. Na segunda vez que o domínio *sexinfo101.com* passar pelo processo de refinamento (associado à categoria *Porn*), este domínio não será submetido à etapa de resolução de domínio, entretanto, este será direcionado para o processo de análise de redundância.

Neste processo, a categoria que possuir o maior grau de perversidade (peso) prevalecerá sobre a categoria com o menor grau. Logo, o domínio *sexinfo101.com* será associado à categoria *Porn*, por possuir maior grau de perversidade. A terceira vez que o domínio *sexinfo101.com* passar pelo processo de refinamento (associado à categoria *Adult*), este também será direcionado para a análise da redundância. Neste caso, o domínio não mudará de categoria, pois o grau de perversidade das categorias é o mesmo. Ao final do processo, o domínio *sexinfo101.com* estará associado somente à categoria *Porn*, eliminando as redundâncias existentes.

Caso seja estabelecida uma regra para bloquear o acesso somente aos domínios pertencentes à categoria *Sexuality* ou *Adult*, o domínio *sexinfo101.com* não será bloqueado já que o mesmo foi desassociado dessas categorias. Neste caso, o grau de

perversidade da categoria deve ser modificado, de forma que essa categoria possua o maior peso. Para o exemplo acima, se for desejado bloquear somente a categoria *Adult*, o grau de perversidade da categoria *Porn* deverá ser alterado para um valor inferior a 10 como peso. Quando o processo de redundância for acionado, os domínios redundantes serão associados à categoria *Adult* já que esta possui o maior peso.

2.2. Processo de Reclassificação de Categorias

O processo de reclassificação de categorias recebe como *input* as categorias da *URLBlackList* e tem como objetivo possibilitar a correção de certas distorções de entendimento relacionadas a interpretação do significado de cada categoria. Isso ocorre porque as categorias da *URLBlackList* estão atreladas a linguagem e a cultura americana. Esse processo possibilitará que as categorias da *URLBlackList Lite* estejam adequadas a especificidade cultural brasileira.

Esse processo possibilita a escolha das categorias, que devem ser unificadas em uma única categoria devido ao contexto a que se referem. Sendo assim, esse processo receberá as categorias selecionadas pelo administrador de redes e o resultado é uma única categoria, nomeada de forma a atender a interpretação do significado e que possui o conjunto dos domínios pertencentes a cada categoria unificada. Por exemplo, para as categorias *Guns* e *Weapons* o resultado obtido é a criação de uma categoria Armas.

3. Experimentos e Resultados

Nesta seção, serão apresentados os resultados dos experimentos realizados com o objetivo de proporcionar uma análise comparativa entre as listas da *URLBlackList* e da *URLBlackList Lite*; comprovar a compatibilidade de execução da ferramenta nos sistemas operacionais *Windows* e *Linux*; e para demonstrar o índice de redução dos registros de domínios da *URLBlackList*.

A *URLBlackList Lite* é uma ferramenta *web* que visa disponibilizar as listas de catalogação enxuta e a própria ferramenta para *download* a fim de possibilitar que o administrador de redes configure a ferramenta de acordo com as políticas de acesso vigentes em uma determinada organização. A ferramenta possibilita a geração de uma lista de catalogação enxuta para cada operador de acordo com as reclassificações de categorias estabelecidas por este. A figura 2 ilustra as telas da ferramenta *URLBlackList Lite*; a primeira tela mostra o menu da ferramenta exibindo as funcionalidades disponíveis, como analisar um arquivo da *URLBlackList*, gerenciar o grau de perversidade das categorias, verificar a versão e realizar o *download* de uma lista da *URLBlackList*. A segunda tela mostra a interface para reclassificar as categorias da *URLBlackList*.

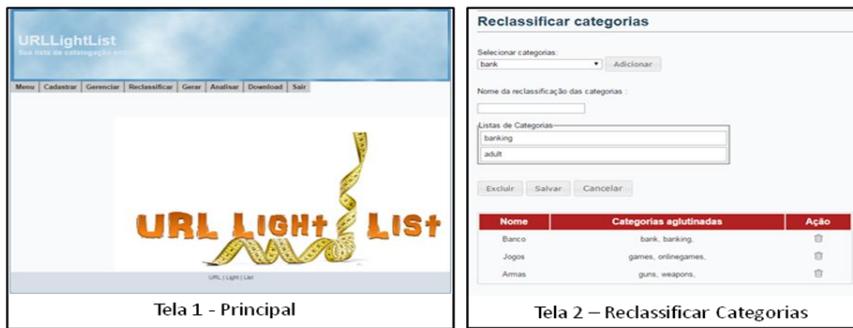


Figura 2. Telas da ferramenta *URLBlackList Lite*.

Os primeiros experimentos foram realizados no sistema operacional Linux devido ao fato do utilitário DIG ser nativo desse sistema operacional. Nessa experimentação inicial, a necessidade de que o sistema registrasse todo o seu funcionamento em um *log* foi evidenciada a fim de capturar qualquer comportamento desconhecido e o experimento foi executado via terminal do sistema operacional Linux. Em uma segunda fase de experimentação, a *URLBlackList Lite* teve o seu funcionamento verificado no ambiente Windows através da instalação do utilitário DIG neste sistema operacional [BIND 2015]. Depois que esses experimentos foram realizados, a *URLBlackList Lite* teve a compatibilidade de execução nos sistemas operacionais *Linux* e *Windows* devidamente comprovada.

A tabela 1 mostra os resultados obtidos com o processo de refinamento realizado pela ferramenta *URLBlackList Lite* tendo como *input* a versão da *URLBlackList* disponibilizada em 14 de maio de 2015. O processo analisou noventa e nove categorias, 2.901.374 domínios presentes nessa versão da *URLBlackList* e foi executado em, aproximadamente, cinco dias. Durante a execução do processo ocorreram algumas interrupções devido à falta de energia elétrica e conectividade, sendo assim, o processo foi reiniciado algumas vezes.

Recomenda-se que o processo de refinamento seja executado quando uma nova versão da lista da *URLBlackList* for disponibilizada. Como o tempo de execução do processo de refinamento é prolongado, uma nova versão da *URLBlackList* pode ser disponibilizada enquanto o processo da *URLBlackList Lite* ainda estiver executando uma versão anterior da *URLBlackList*. Quando julgar necessário, o administrador de redes poderá executar um processo de atualização do *status* de resolução dos domínios existentes no banco de dados.

Tabela 1. Resumo dos Resultados Obtidos

<i>URLBlackList</i>	2.901.374	100%
<i>URLBlackList Lite</i> Resolvível	1.202.452	41,44%
<i>URLBlackList Lite</i> Não Resolvível	756.096	26,06%
Domínios com formato ilegais/redundantes/IP	942.826	32,50
Índice redução (%)		58,56%

O índice *URLBlackList* representa a quantidade de domínios encontrados na lista da *URLBlackList*; já os índices *URLBlackList – Resolvível* e *URLBlackList*

Lite – Não Resolvível representam o quantitativo, respectivamente, de domínios resolvíveis e não resolvíveis identificados na lista da *URLBlackList*; o índice de domínios com formato ilegais/redundantes/anômalos/IP engloba a quantidade de domínios que foram expurgados devido as anomalias, IP e redundâncias encontradas na lista da *URLBlackList*; por fim, o índice de redução representa o percentual de registros retirados da *URLBlackList* revelando assim, o quanto a lista ou uma categoria ficou mais enxuta após o processo de refinamento da *URLBlackList Lite*.

Analizando a tabela 1, a ferramenta *URLBlackList Lite* identificou 756.096 domínios como não resolvíveis proporcionando uma redução de cerca de 26% em relação a lista da *URLBlackList*. Sendo assim, retirando os domínios não resolvíveis dos registros da *URLBlackList*, esta passou a ter 2.145.278 registros. O quarto índice, representa uma redução de 32,5 % nos registros da *URLBlackList* através do tratamento de redundância. Esse índice também abrange a quantidade de expurgos de domínios com formatos ilegais e com outras anomalias identificadas na etapa de resolução. Aplicando mais esse percentual de redução a *URLBlackList*, esta foi reduzida para 1.202.452 registros, totalizando um índice de redução de 58,56% proporcionado pelo processamento da *URLBlackList Lite*.

A tabela 2 mostra um *ranking (top10)* com as dez categorias analisadas que obtiveram os maiores índices de redução pelo processo de refinamento da *URLBlackList Lite*. Analisando a tabela 2 constata-se que: a categoria *Verisign* foi reduzida em 100%, pois os dois domínios existentes nessa categoria são não resolvíveis; as categorias *Malware*, *Phishing*, *Homerepair* possuem uma quantidade maior de domínios não resolvíveis do que domínios resolvíveis; as categorias *Malware*, *Arjel*, *Hacking*, *Aggressive*, *Spyware* e *Cleaning* apresentaram uma quantidade expressiva de domínios expurgados (índice de domínios com formato ilegais/redundantes/anômalos/IP) e a categoria *Adult* revelou um alto índice de redundância com a categoria *Porn* o que foi observado no arquivo de *log* da ferramenta.

Tabela 2. Resultado Obtido por Categoria

Categoria	<i>URLBlackList</i>	<i>URLBlackList Lite</i> Resolvível	<i>URLBlackList Lite</i> Não Resolvível	Domínios com formato ilegais/redundantes/IP	Índice redução (%)
<i>Verisign</i>	2	0	2	0	100%
<i>Malware</i>	340.030	25.633	53.176	261.221	92%
<i>Arjel</i>	69	7	4	58	90%
<i>Hacking</i>	581	90	25	466	85%
<i>Phishing</i>	121.388	18.711	82.098	20.579	85%
<i>Aggressive</i>	433	105	19	309	76%
<i>Spyware</i>	193	49	45	99	75%
<i>Adult</i>	997.238	272.664	195.824	528.750	73%
<i>Cleaning</i>	178	55	14	109	69%
<i>Homerepair</i>	21	7	9	5	67%

4. Conclusão

Esse trabalho apresentou uma ferramenta capaz de realizar um processo de refinamento na lista da *URLBlackList*, segmentando os domínios presentes nessa lista em domínios resolvíveis e não resolvíveis, expurgando IP e domínios com erros de resolução e tratando as redundâncias de domínios quando este pertencer a mais de uma categoria na lista da *URLBlackList*. Após o processo de refinamento, a ferramenta possibilita a geração da *URLBlackList Lite*, que é uma lista enxuta composta apenas por domínios resolvíveis, a qual será utilizada por um servidor *Proxy* para realizar um serviço de controle de conteúdo das páginas *web* acessadas por um usuário de uma determinada rede local.

Os resultados obtidos com os experimentos realizados se mostraram promissores já que a lista da *URLBlackList* foi reduzida em 58,56%, ou seja, os resultados comprovam o fato de que se a lista da *URLBlackList Lite* for usada por um servidor *Proxy* ao realizar o serviço de controle de conteúdo proporcionará uma otimização dos recursos computacionais utilizados por este serviço. Portanto, a ferramenta *URLBlackList Lite* conseguiu refinar a lista da *URLBlackList* e atingir o objetivo de gerar uma lista de catalogação efetivamente mais enxuta, a lista da *URLBlackList Lite*.

Como trabalhos futuros, o processo de refinamento poderá ser implementado empregando-se *Threads* para que mais de uma categoria seja analisada ao mesmo tempo, aumentando assim, a performance da ferramenta e diminuindo o tempo gasto em uma análise. Ao processo de analisar banco de dados e arquivo, um botão de parada deverá ser disponibilizado para que o usuário possa interromper o processo de análise quando desejar. Poderá ser implementado uma funcionalidade de agendamento para que a ferramenta dispare uma análise de banco de dados, automaticamente e um aperfeiçoamento do tratamento de erro do *time out*. Além disso, um experimento poderá ser realizado em um servidor *Proxy*, como o Squid, utilizando a lista de catalogação da *URLBlackList Lite* e a lista da *URLBlackList* a fim de mensurar e comparar o desempenho proporcionado pela processo de refinamento.

5. Referências

- BIND. BIND. Disponível em: <<https://www.isc.org/downloads/bind/>>. Acesso em: 12 de junho de 2015.
- FOROUZAN, Behrouz A. Comunicação de Dados e Redes de Computadores – Porto Alegre: Bookman, 2008.
- MORIMOTO, Carlos Eduardo. Servidores Linux, guia prático – Porto Alegre: Sul Editores, 2009.
- URLBLACKLIST. UrlBlackList. Disponível em: <<http://urlblacklist.com/?sec=search>>. Acesso em: 12 de junho de 2015.
- KRASNER, G. E., POPE, S. T. A description of the Model-View-Controller user interface paradigm in the Smalltalk-80 System. 1988. Disponível em: www.create.ucs.edu/~stp/PostScript/mvc.pdf. Acesso em: 05 de julho de 2015.

IV

Sessão 4 - Segurança de Redes

NS²A: consciência de situação aplicada a segurança de redes de computadores

**Ricardo Borges Almeida¹, Roger da Silva Machado¹,
Diógenes Y. L. da Rosa¹, Henrique de Vasconcellos Rippel¹,
Lucas Medeiros Donato², Adenauer Corrêa Yamin¹, Ana Marilza Pernas¹**

¹Universidade Federal de Pelotas (UFPel)
Pelotas – RS – Brasil

²De Montfort University – Cyber Security Centre
Leicester, Reino Unido

{rbalmeida, rdsmachado, adenauer, marilza}@inf.ufpel.edu.br,
diogenes.yuri@ufpel.edu.br, hvrippel@gmail.com
lucas.donato@myemail.dmu.ac.uk

Abstract. This paper presents a situation awareness approach to computational environments security, called NS²A (Network Security Situation Awareness). The architecture is a prominent part of the solution, designed to provide Situation-Awareness exploring different features since the gathering of events, passing by processing, contextual data storage and actuation. The approach was evaluated in an environment consisting of servers designed to provide Internet services (email, websites, etc.), proving to be stable and flexible concerning the visibility of security aspects in computational environments.

Resumo. Este artigo apresenta uma abordagem consciente de situação para segurança em redes de computadores, denominada NS²A. A solução destaca-se pela arquitetura concebida para o fornecimento da Consciência de Situação, explorando diferentes funcionalidades desde a coleta, passando por um processamento, armazenamento de dados contextuais e a decorrente atuação. A abordagem foi avaliada em ambiente formado por servidores destinados a prover serviços de Internet (email, websites, etc.), se mostrando estável e flexível quanto à visibilidade de aspectos de segurança em ambientes computacionais.

1. Introdução

Tim Bass (1999) propôs a aplicação dos conceitos de Consciência de Situação no campo da segurança em redes de computadores, com o intuito de fornecer uma visão mais apurada dos aspectos de segurança do ambiente computacional. Tim Bass é tido como o primeiro autor a empregar estes conceitos na obtenção de um melhor entendimento sobre o ambiente monitorado.

Embora a união destas duas áreas venha sendo estudada há aproximadamente quinze anos, ela ainda constitui um foco de estudo e pesquisa relevante na área de segurança da informação [Sharma and Kate 2014]. Por sua vez, é importante registrar que os riscos de segurança têm se potencializado devido à natureza volátil, espontânea,

heterogênea e transparente de como ocorre a comunicação nas atuais infraestruturas computacionais [Onwubiko 2012].

Um dos requisitos para se obter a Consciência de Situação é o monitoramento contínuo dos eventos de segurança. Estes eventos podem ser oriundos da utilização dos recursos computacionais (memória, processamento, disco rígido, rede, entre outros) e dos logs gerados pelos diferentes sistemas e ativos de rede. Uma vez coletados, os eventos podem ser correlacionados para detectar situações de interesse, aprimorando a visão geral sobre o ambiente [Chuvakin et al. 2012].

O objetivo central deste trabalho é apresentar a concepção de uma abordagem denominada NS²A (*Network Security Situation Awareness*) que fornece a Consciência de Situação sobre os aspectos de segurança das redes de computadores. A abordagem foi concebida com base em um *middleware* para computação ubíqua denominado EXEHDA (*Execution Environment for Highly Distributed Applications*), beneficiando-se da sua arquitetura e de seus mecanismos de consciência de contexto [Lopes et al. 2014].

Para obtenção dos conceitos de Consciência de Situação, a abordagem concebida e prototipada tem como principal premissa uma arquitetura que explora estes conceitos desde a coleta, passando por um processamento de contexto, armazenamento de dados contextuais e a decorrente atuação. Outro ponto a ser destacado refere-se à detecção de situações de interesse, empregando uma estratégia baseada em regras com sintaxe similar a SQL (*Structured Query Language*).

A abordagem foi avaliada em ambiente formado por servidores destinados a prover serviços de Internet (email, websites, etc.), se mostrando estável e flexível quanto à visibilidade de aspectos de segurança em ambientes computacionais.

Este artigo está organizado da seguinte forma: a Seção 2 introduz as características da Consciência de Situação e o *middleware* EXEHDA. Na Seção 3 é discutida a concepção da abordagem proposta. A Seção 4 apresenta os cenários de uso para avaliação do trabalho desenvolvido. Por sua vez, na Seção 5 os trabalhos relacionados são descritos e analisados. Finalmente, na Seção 6, são apresentadas as considerações finais.

2. NS²A: Base Conceitual

Esta seção introduz a base conceitual associada à concepção da abordagem NS²A. Estes conceitos também foram considerados nos esforços de avaliação e testes da mesma.

2.1. Consciência de Situação

A Consciência de Situação consiste da percepção e compreensão de uma ou mais informações contextuais e a projeção de seus efeitos em um futuro próximo. Percebe-se, então, a existência de três níveis para a obtenção da Consciência de Situação [Onwubiko 2012]:

- percepção: envolve os processos de monitoramento, detecção e reconhecimento, que levam a consciência de múltiplos elementos situacionais, tais como, alertas relatados por sistemas de detecção e prevenção de intrusão, eventos registrados em logs, relatórios de varredura, bem como os seus estados atuais (tempo em que ocorreram, locais, condições, formas e ações);

- compreensão: síntese e correlação dos elementos desconexos identificados no nível de percepção por intermédio de diferentes estratégias, tais como, baseada em conhecimento e baseada em anomalias. Este nível requer a integração dessas informações para entender como isso vai impactar a segurança do ambiente computacional;
- projeção: responsável pela capacidade de antecipação de ocorrências futuras, a partir da compreensão dos elementos no ambiente atual. Alcançado por meio do conhecimento da situação, da dinâmica dos elementos e da compreensão da situação, para depois projetar esta informação adiante no tempo e assim determinar se elas afetarão os futuros estados do ambiente operacional.

2.2. Middleware EXEHDA

O EXEHDA possui uma arquitetura distribuída e oferece suporte à aquisição, processamento e armazenamento de informações contextuais, características oportunas às funcionalidades da NS²A.

O EXEHDA objetiva a criação e o gerenciamento de um ambiente ubíquo formado por células de execução distribuídas, promovendo a computação sobre esse ambiente cuja composição é dinâmica e integralizada por equipamentos heterogêneos [Lopes et al. 2014].

Dentro de cada célula podem existir inúmeros SB's (Servidores de Borda) que são responsáveis pela comunicação com o ambiente por meio de sensores e atuadores. Além disso, cada célula possui um equipamento central (EXEHDAbase) no qual executa o SC (Servidor de Contexto), sendo este servidor responsável por armazenar as informações coletadas no RIC (Repositório de Informações Contextuais), bem como permitir a manipulação (processamento, visualização, etc.) destas informações.

3. NS²A: Concepção

A NS²A caracteriza-se principalmente pela Consciência de Situação sobre a segurança do ambiente computacional, tendo sido proposta considerando as premissas operacionais e as estratégias de concepção do *middleware* EXEHDA.

Os componentes prototipados denominados de NS²A-BS (NS²A Border Server) e NS²A-CS (NS²A Context Server) representam respectivamente os SB's e os SC's. A seguir, são descritos os dois componentes, detalhando como cada módulo interno oferece os conceitos para a formação da Consciência de Situação.

A Figura 1 apresenta uma abstração do componente de software proposto e desenvolvido para o NS²A-BS, destacando o fluxo de comunicação entre os módulos.

O módulo “Coletor de Logs (Internos)” realiza a leitura dos arquivos de log internos ao sistema onde o NS²A-BS está operacional. Já o “Coletor de Logs (Externos)”, foi concebido para receber eventos de diferentes dispositivos, neste último caso, funcionando como um servidor Syslog¹ permitindo o tratamento de eventos de dispositivos onde não é possível a instalação do NS²A-BS. O “Coletor de Status” por sua vez, foi projetado para coletar eventos sobre o uso dos recursos do sistema operacional, como por exemplo, erros

¹Syslog é um mecanismo padronizado para atividade de logging em sistemas de computador, <<http://www.syslog.org/>>

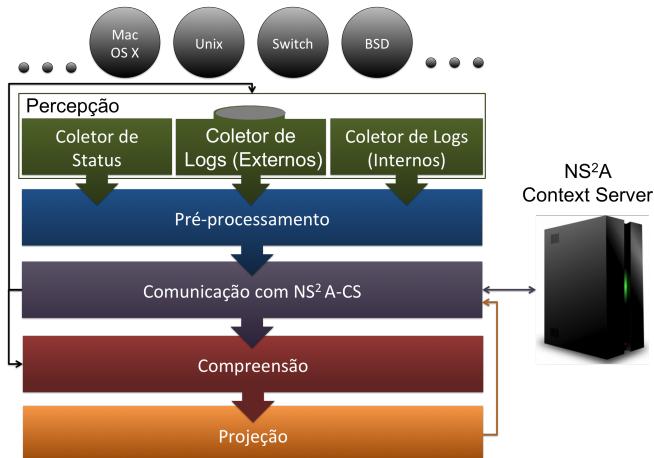


Figura 1. Componente de software concebido para o NS²A-BS

nas interfaces de rede, consumo de processamento, de memória, de disco e de rede, *hash* de arquivos como /etc/passwd, entre outros.

Para aprimorar a capacidade de Percepção da abordagem, reforçando sua flexibilidade, heterogeneidade e dinamicidade, foi desenvolvida a capacidade de descoberta automática de recursos existentes (interfaces de rede, partições, logs, entre outros) e também de situações a serem avaliadas com base na especificação de variáveis nas configurações dos itens e situações. No protótipo desenvolvido em Python, sempre que o NS²A-BS coletada os itens a serem monitorados, uma função é responsável por identificar a existência de variáveis na especificação de cada item. Caso alguma variável pré-definida seja identificada, a função executará rotinas para identificação dos dispositivos existentes no sistema. Esta verificação ocorrerá periodicamente de acordo com o intervalo de coleta (atraso) configurado no item [Almeida 2013].

Os três módulos, junto à descoberta de recursos, representam a Percepção no NS²A-BS, primeiro nível da Consciência de Situação.

Em particular, o módulo de “Pré-processamento” foi importante para este trabalho, pois realizou as tarefas de normalização e contextualização dos eventos coletados. Ele é utilizado para realizar a separação do evento em campos e posteriormente adicionar informações contextuais, auxiliando a etapa de compreensão [Machado 2013].

O módulo de “Comunicação com o NS²A-CS” foi previsto para ser responsável pela comunicação com o componente NS²A-CS, enviando os eventos coletados e situações identificadas no NS²A-BS para serem armazenados no RIC, que foi adaptado para estar de acordo com esta proposta. Este módulo também realiza a busca periódica no servidor, pelas informações necessárias para a execução do NS²A-BS, incluindo os logs e status que devem ser monitorados, as expressões para normalização e contextualização, e as situações a serem identificadas com as respectivas projeções.

Na concepção do módulo de “Compreensão” foi considerado o emprego da estratégia baseada em regras, com o apoio de uma solução de CEP (*Complex Event Processing*).

sing) denominada Esper², a qual realiza a correlação de eventos na busca por padrões descritos em uma EPL (*Event Processing Language*) com sintaxe similar à SQL. A utilização desta sintaxe é um diferencial no âmbito da solução concebida, pois apresenta uma alternativa ao tradicional uso de expressões regulares.

Adicionalmente, foi desenvolvido um sistema de priorização, no qual é possível especificar diferentes valores de severidade para cada regra e definir o grau de criticidade de cada sistema monitorado. Estas duas informações formam a prioridade da regra a ser confrontada com os eventos e das situações identificadas a serem exibidas ao administrador, auxiliando a compreensão das situações no ambiente.

O módulo de “Projeção” possui a finalidade prevista de evitar ocorrências futuras de situações indesejadas, envolvendo desde o envio de alertas até a efetiva atuação sobre o sistema. Após a projeção, a situação identificada, junto aos possíveis retornos referentes à atuação, são enviados ao NS²A-CS para serem armazenados no RIC, disponibilizando assim sua visualização na interface Web.

Continuando a descrição da arquitetura de software proposta, a Figura 2 apresenta uma abstração do componente de software projetado e desenvolvido para o NS²A-CS.

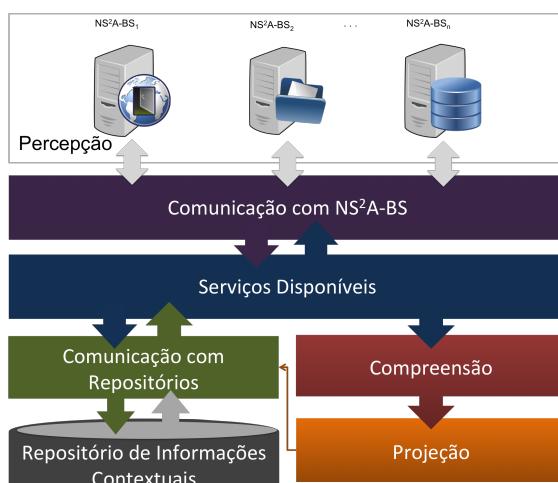


Figura 2. Componente de software concebido para o NS²A-CS

Na concepção do módulo de “Comunicação com o NS²A-BS” foi empregado o protocolo XML-RPC (*eXtensible Markup Language - Remote Procedure Call*) visto que ele já era empregado nos diferentes servidores da arquitetura do middleware EXEHDA.

Os NS²A-BS’s, ao coletarem e disponibilizarem as informações ao NS²A-CS, proporcionam a Percepção para Consciência de Situação de forma aprimorada.

O módulo de “Serviços Disponíveis” foi concebido para ser o responsável pelo provimento das funções que serão utilizadas pelo protocolo XML-RPC, já que a sua comunicação é realizada por meio de chamadas das funções previamente registradas.

²<http://esper.codehaus.org>

Dentre elas, estão o repasse de eventos e/ou situações ao módulo de compreensão e a comunicação com o módulo que realiza o acesso ao RIC.

O módulo de “Comunicação com Repositório” realiza a coleta de informações solicitadas pelos NS²A-BS’s e a inserção de dados no RIC. Os módulos “Compreensão” e “Projeção” do NS²A-CS funcionam de forma análoga aos propostos no NS²A-BS.

4. Cenário de Uso e Testes

Para a validação da NS²A, inicialmente foram realizados testes em ambiente simulado, e posteriormente, a solução foi configurada para operar durante 5 dias nos servidores da universidade onde este trabalho foi desenvolvido. O NS²A-BS foi instalado em (a) três servidores de envio de e-mail, os quais utilizam um anti-spam; (b) três servidores de hospedagem de páginas; (c) e um servidor de WAF (*Web Application Firewall*). O NS²A-CS foi instalado em uma máquina virtualizada com 4 núcleos Intel Xeon CPU E5606 2.13GHz, 2GB de memória física e 50GB de disco rígido.

Durante o tempo em que a NS²A ficou em execução, foram monitorados aproximadamente 60 arquivos de log e 420 itens de status, resultando em quase 10GB de eventos armazenados. Foram identificadas 20463 situações (incluindo reincidências), sendo 327 situações únicas.

A avaliação da descoberta automática de recursos se deu por meio de regras envolvendo as interfaces de rede e partições de disco, onde as variáveis \$IFACE e \$PARTITION foram especificadas nas configurações dos itens a serem monitorados. Como por exemplo, quando o NS²A-BS identifica o item que realiza a coleta da porcentagem utilizada em uma partição possuindo a chave “filesystem.size[\$PARTITION, pused]”, ele realiza a descoberta das partições existentes no sistema e envia um alerta ao NS²A-CS para criação de novos itens com esta variável sendo substituída por “filesystem.size[/, pused]” e “filesystem.size[/tmp, pused]”, considerando hipoteticamente que estas sejam as partições existentes em um servidor monitorado.

Dentre as situações especificadas, destaca-se a identificação de dez ou mais tentativas de acesso à arquivos inexistentes registradas nos três servidores de hospedagem de páginas em um intervalo de um minuto a partir de uma única fonte. Esta situação foi configurada para operar no NS²A-CS, tendo a visibilidade dos três servidores mencionados e podendo detectar estes acessos quando ocorrerem de forma distribuída.

Para a identificação desta situação, a regra “SELECT * FROM ApacheErrorLog(ip != 'null' and message like '%File does not exist%').win:time(1 min) GROUP BY ip HAVING count(*) >= 10” foi estabelecida. Como método de ação, o bloqueio no *firewall* dos servidores mencionados foi especificado na configuração da situação. Esta situação validou a capacidade de percepção e o módulo de compreensão disponibilizados no NS²A-CS, assim como a possibilidade de atuação distribuída. Durante o período de execução foram bloqueados 17 endereços IP (*Internet Protocol*), diminuindo o risco e a sobrecarga dos servidores.

Outra situação considerada na avaliação foi a identificação de dez ou mais tentativas consideradas suspeitas pelo WAF. Visto que o WAF executa em um único

servidor, esta situação foi especificada para operar no NS²A-BS, validando seus módulos de Consciência de Situação. Para isto, a regra “SELECT * FROM ApacheErrorLog(ip!=’null’ and severity in (‘EMERGENCY’, ‘ALERT’, ‘CRITICAL’)).win:time(1 min) GROUP BY ip HAVING count(*) >= 10” foi aplicada nos testes, tendo como atuação o envio de e-mail para a primeira ocorrência de cada IP. Como resultados da execução, o sistema identificou 13476 ocorrências, sendo 23 destas, situações únicas.

5. Trabalhos Relacionados

Em [Preden et al. 2011] são explorados os conceitos de formação de hierarquias e de modelos de informação situacional com base em dados disponíveis a partir de um sistema de monitoramento distribuído de onde as propriedades temporais e espaciais de informação situacional são levadas em conta. Um estudo de caso é apresentado, que mostra a viabilidade dos conceitos em um cenário de monitoramento real.

O artigo [Zhang et al. 2013] introduz um *framework* multinível de análises para a Consciência de Situação em segurança de rede como uma adaptação do modelo de Endsley [Endsley 1995]. Não são apresentados detalhes sobre a proposta, sendo destacado o fato de ser um trabalho em desenvolvimento.

Em [Timonen et al. 2014], é apresentado um *framework* para a criação de um COP (*Common Operation Picture*) de infraestruturas críticas. O framework SACIN (*Situational Awareness of Critical Infrastructure and Networks*) demonstra as principais características do conceito. Como contribuições o trabalho destaca a combinação do modelo JDL (*Joint Directors of Laboratories*) e a arquitetura baseada em agentes, apoiados pela implementação. Neste artigo foram apresentados também os resultados dos testes realizados com os operadores do sistema.

Apesar dos trabalhos discutirem arquiteturas aplicadas ao fornecimento de Consciência de Situação em segurança de redes de computadores, diferentemente dos mesmos, o presente trabalho busca este conceito por meio da arquitetura de software com módulos distribuídos, cuja atuação acontece desde o momento da coleta dos eventos, até seu processamento, armazenamento e projeção. Além disso, sente-se falta na descrição dos trabalhos relacionados de aspectos pertinentes no emprego das soluções concebidas mapeadas sobre as infraestruturas computacionais, sendo que a NS²A discute tópicos relacionados a coleta, processamento e sintaxe das regras, assim como a atuação no ambiente computacional.

6. Considerações Finais

Com a concepção e prototipação da NS²A, baseada no *middleware* EXEHDA, visando aplicação dos conceitos de Consciência de Situação, foi possível fornecer flexibilidade e heterogeneidade nos aspectos referente à percepção por meio da possibilidade de recebimento de eventos pelo protocolo Syslog e pela descoberta automática de recursos.

A compreensão torna-se flexível e apta para as infraestruturas heterogêneas, visto a possibilidade de criação de novas regras que refletem as necessidades do ambiente, explorando a sintaxe similar a SQL. Por fim, a projeção possibilita a execução de ações distribuídas potencializando a abordagem visto a atual distribuição dos ambientes computacionais.

Por meio da avaliação da proposta colocada em execução nos servidores mencionados, além dos ataques discutidos, foi possível identificar ataques da rede interna e situações que poderiam impactar na disponibilidade do sistema (um dos elos da segurança da informação), tais como: erros na configuração do *firewall*; pouco espaço disponível em disco de alguns servidores; erros em interfaces de rede em um servidor; servidores sobre-carregados; e erros no código de aplicações web desenvolvidas por terceiros.

Como trabalho futuro pretende-se aprimorar os testes realizados na busca por uma melhor quantificação dos resultados, avaliando o consumo de memória, processamento e largura de banda da rede. Espera-se avaliar a integração com soluções de análise de vulnerabilidades, e empregar conceitos de big data, visto a quantidade e variedade dos eventos de segurança e a velocidade em que eles devem ser tratados.

Referências

- Almeida, R. B. (2013). Segurança da informação e gerenciamento de eventos: Uma abordagem explorando consciência de situação. Monografia de graduação em ciência da computação, Universidade Federal de Pelotas.
- Chuvakin, A., Schmidt, K., and Phillips, C. (2012). *Logging and Log Management: The Authoritative Guide to Dealing with Syslog, Audit Logs, Events, Alerts and other IT ‘Noise’*. Elsevier Science.
- Endsley, M. R. (1995). Measurement of situation awareness in dynamic systems. *Human Factors*, 37.
- Lopes, J., Souza, R., Geyer, C., Costa, C., Barbosa, J., Pernas, A., and Yamin, A. (2014). A middleware architecture for dynamic adaptation in ubiquitous computing. *j-jucs*, 20(9):1327–1351.
- Machado, R. S. (2013). Loga-dm: Uma abordagem de análise dinâmica de log com base em mineração de dados. Monografia de graduação em ciência da computação, Universidade Federal de Pelotas.
- Onwubiko, C. (2012). *Situational Awareness in Computer Network Defense: Principles, Methods and Applications: Principles, Methods and Applications*. Premier reference source. Information Science Reference.
- Preden, J., Motus, L., Meriste, M., and Riid, A. (2011). Situation awareness for networked systems. In *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2011 IEEE First International Multi-Disciplinary Conference on*, pages 123–130.
- Sharma, C. and Kate, V. (2014). Icarfad: A novel framework for improved network security situation awareness. *International Journal of Computer Applications*, 87(19).
- Timonen, J., Puuska, S., Lääperi, L., Vankka, J., and Rummukainen, L. (2014). Situational awareness and information collection from critical infrastructure. In *Cyber Conflict (CyCon 2014), 2014 6th International Conference On*, pages 157–173.
- Zhang, H., Shi, J., and Chen, X. (2013). A multi-level analysis framework in network security situation awareness. *Procedia Computer Science*, 17(0):530 – 536. First International Conference on Information Technology and Quantitative Management.

DLNA-ML: Uma Abordagem de Análise Dinâmica de Log e Tráfego da Rede

**Roger da Silva Machado¹, Ricardo Borges Almeida¹,
Diógenes Y. L. da Rosa¹, Henrique de Vasconcellos Rippel¹,
Adenauer Corrêa Yamin¹, Ana Marilza Pernas¹**

¹Universidade Federal de Pelotas (UFPel), Pelotas – RS – Brasil

{rdsmachado, rbalmeida, marilza, adenauer}@inf.ufpel.edu.br,

diogenes.yuri@ufpel.edu.br, hvrippel@gmail.com

Abstract. This paper proposes an approach to perform log analysis with intent to prevent attack situations. The proposed solution explores two fronts: (i) log of applications; and (ii) network traffic. The proposed approach was evaluated with the conception of a prototype that employs modules for the collection and normalization of data. The normalization module also adds contextual information in order to assist the analysis of critical security situations. The network traffic records are collected and evaluated from connections in progress, in order to preserve the autonomous operation of the system. The tests developed in the proposed solution show good results for typical categories of attack.

Resumo. Este trabalho propõe uma abordagem para realizar a análise de log com intuito de prevenir situações de ataque. A solução proposta explora duas frentes: (i) logs de aplicações; e (ii) tráfego da rede. A abordagem proposta foi avaliada com a concepção de um protótipo que emprega módulos para a coleta e normalização dos dados. O módulo de normalização também adiciona informação contextual, a fim de auxiliar a análise de situações críticas de segurança. Os registros do tráfego da rede são coletados e avaliados a partir de conexões em andamento, com o intuito de conservar a operação autônoma do sistema. Os testes desenvolvidos na solução proposta mostram bons resultados para categorias típicas de ataque.

1. Introdução

Como introduzido no clássico artigo de Weiser [Weiser 1991] o paradigma da UbiComp tem como premissa prover computação de forma transparente, estando o modelo computacional integrado às demandas do usuário. Nesta perspectiva, a mobilidade do usuário e as decorrentes trocas de infraestrutura de acesso, presentes na UbiComp, potencializam a preocupação com a segurança da informação.

Uma das tarefas relevantes para segurança da informação é a análise de log, sendo esta uma técnica utilizada com o intuito de melhorar a compreensão e o funcionamento do sistema, visando a detecção de tentativas de ataques e identificar ações realizadas por um invasor [Hoepers and Steding-Jessen 2003]. Os diferentes formatos e informações de cada tipo de log fazem com que a tarefa de análise dos mesmos deixe de ser trivial. Além disso, os arquivos de log tendem a possuir inúmeras entradas, pois são gerados

registros de praticamente todas as atividades referentes às aplicações em uso no sistema computacional, o que também contribui para aumentar significativamente o custo de uma análise manual destes registros.

Este trabalho propõe uma abordagem denominada DLNA-ML (*Dynamic Log and Network Analyzer - Machine Learning*), o qual tem como objetivo central explorar a análise de logs e do tráfego da rede, a fim de tratar as incidências de atividades suspeitas, garantindo maior segurança na infraestrutura computacional em um ambiente ubíquo. De modo mais específico, a proposta é empregar a análise de logs com o objetivo de melhorar a compreensão do funcionamento do sistema e explorar uma técnica de aprendizagem de máquina com o intuito de classificar o tráfego da rede em tempo de execução, visando detectar tentativas de ataques.

Este artigo está organizado da seguinte forma: na seção 2 os trabalhos relacionados são descritos e analisados; a seção 3 apresenta os principais aspectos relacionados à tarefa de análise de log, mostrando algumas particularidades e benefícios da sua utilização; na seção 4 é discutida a concepção da abordagem proposta, caracterizando o funcionamento dos módulos disponíveis no componente de software; a seção 5 apresenta o cenário de uso para avaliação do trabalho desenvolvido. Finalmente, na seção 6, são apresentadas as considerações finais.

2. Trabalhos Relacionados

Em [Campos and Lima 2012], é apresentado um IDS baseado em aprendizagem de máquina, com o objetivo de classificar os registros em normais e ataques, utilizando a base de dados do KDD Cup 99 tanto para treinamento como para teste. Foram utilizadas as técnicas Redes Neurais, Árvore de Decisão e Redes Bayesianas, sendo que a técnica de Árvore de Decisão foi a que alcançou a melhor taxa de acertos.

Em [Arjunwadkar and Parvat 2015], é apresentado uma proposta de IDS híbrido que combina diferentes técnicas de aprendizagem de máquina com o objetivo de classificar o tráfego da rede. Para avaliar a abordagem, foi utilizada uma versão da base de dados KDD Cup 99 tanto para treinamento quanto para teste. A técnica que alcançou a melhor taxa de acertos foi a técnica de árvores de decisão.

Analizando os trabalhos relacionados, diferentemente do presente trabalho, os mesmos só analisam registros históricos do tráfego da rede, não possuindo a possibilidade de classificação em tempo de execução, dificultando a tomada de ação imediata por parte do administrador do sistema. Além disso, destaca-se que o presente trabalho propõe a coleta e o pré-processamento de logs de aplicações com o intuito de facilitar o processo de análise destes registros.

3. Análise de Log

O termo log refere-se a um arquivo gerado por uma determinada aplicação, que possui inúmeros registros de eventos, os quais permitem que um analista visualize as atividades que ocorrem nos sistemas computacionais (serviços em geral, e/ou o comportamento da própria rede de computadores utilizada) [Grégio 2008]. Log é considerado uma das principais fontes de dados para execução de uma perícia bem sucedida em um sistema [Cansian 2001]. Um arquivo de log pode ser produzido em modo texto, ou em outro modo de interesse específico da aplicação em questão.

Diferentes componentes que integram o sistema computacional geram registros de log, tais como: sistema operacional, SGBD (Sistemas Gerenciadores de Banco de Dados), IDS (*Intrusion Detection System*), *firewall*, antivírus, dispositivos de rede, dentre outros. Os eventos inseridos nos arquivos de log podem ser referentes às atividades normais, alertas ou erros. Observa-se que cada tipo de log possui um formato particular, sem um padrão convencional, dificultando, assim, a interpretação dos registros gerados pelas aplicações.

Atualmente, as diferentes atividades dos dispositivos computacionais geram registros de log de tamanhos elevados, trazendo dificuldades à análise manual destes eventos. Devido a este fato, muitas vezes não é possível analisar os registros coletados em um espaço razoável de tempo, o que pode tornar a implementação de contramedidas ineficiente, pois é necessário que a ação por parte do administrador do sistema seja o mais imediata possível ao acontecimento de um determinado evento ou conjunto destes, com o intuito de reduzir o impacto de um possível incidente de segurança, ou até mesmo evitá-lo.

Devido às dificuldades encontradas na análise de log, verificou-se o aumento das pesquisas que buscam propostas para auxiliar na realização desta tarefa. A revisão de literatura indicou que as principais propostas de auxílio para a análise de log implementam as seguintes funcionalidades [CLEMENTE 2008]:

- análise léxica: processo relativo à análise dos registros de log e produção de uma saída formatada em um padrão mais adequado para futuro processamento. O sistema deve permitir que diferentes arquivos de log em diferentes padrões possam ser analisados, pois em um ambiente dinâmico muitos sistemas geram logs variados e de forma simultânea;
- análise sobre eventos de log: processo para extração de informações relevantes sobre as mensagens de log através de algoritmos, regras ou consultas. Nesta atividade, podem ser aplicadas técnicas com o intuito de realizar filtros nos registros coletados, identificando quais devem ser analisados pelo analista e como devem ser enquadrados, se registros normais (rotinas do sistema), ou atividades suspeitas;
- transmissão: processo de transmissão dos registros de log para um servidor remoto. Esta atividade é importante para realização da tarefa de análise de log, pois é necessário manter os registros coletados em outro sistema, minimizando a possibilidade dos registros serem modificados de forma maliciosa. Isso se dá devido ao fato de que normalmente quando um atacante consegue acesso a um sistema ele modifica/apaga os registros gerados com o intuito de esconder as atividades que realizar;
- armazenamento: processo que compreende a retenção dos registros de log para futuras consultas, as quais servirão para atender casos de auditoria, entendimento e construção de padrões;
- visualização: processo que permite a visualização dos registros de log, sendo estes, atuais ou históricos. Desta forma, permitindo que analistas acompanhem a execução do sistema através dos registros de log gerados.

4. DLNA-ML: Concepção

A abordagem DLNA-ML foi concebida para realizar a coleta dos registros de log e do tráfego da rede na busca por situações de interesse. A Figura 1 apresenta uma abstração

do componente de software proposto e desenvolvido para o DLNA-ML, destacando o fluxo de comunicação entre os módulos.

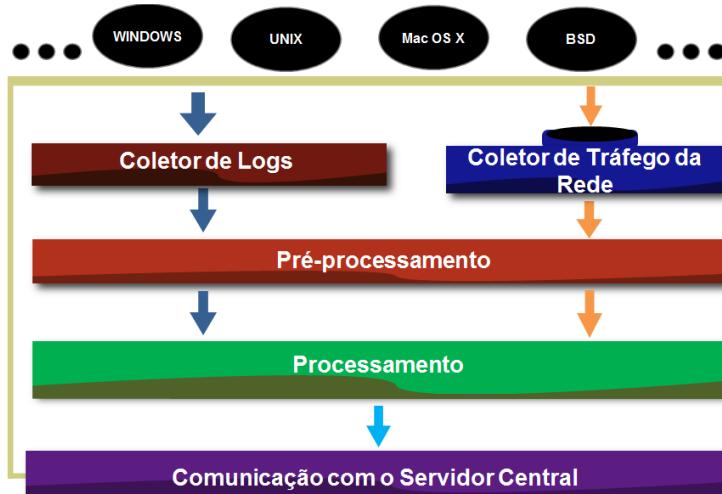


Figura 1. Componente de software concebido para o DLNA-ML

4.1. Módulos de Coleta

O módulo “Coletor de Logs” foi desenvolvido com a premissa de ler os arquivos de log internos ao sistema onde o DLNA-ML está operacional e receber eventos de diferentes dispositivos, neste último caso, funcionando como um servidor Syslog¹ permitindo o tratamento de eventos de dispositivos onde não é possível a instalação do DLNA-ML.

O “Coletor do Tráfego da Rede”, foi concebido para realizar a coleta de eventos na camada de rede, empregando a funcionalidade de *sniffer*. Cada pacote capturado é analisado para determinar se ele consiste de uma nova sessão ou se pertence a uma já existente. Quando ocorre o encerramento da sessão, o pacote é repassado para o módulo de pré-processamento.

4.2. Módulo de Pré-processamento

Considerando a necessidade de normalização e contextualização dos registros de log coletados, o módulo de pré-processamento da abordagem DLNA-ML foi concebido para realizar a separação dos registros em campos e posteriormente adicionar informações contextuais, auxiliando a etapa de processamento. Além disso, o módulo realiza a eliminação de campos que não sejam de interesse para análise.

Para concepção deste módulo foi explorado um *parser* denominado PyParsing², sendo um diferencial como alternativa ao tradicional uso de expressões regulares. Destaca-se que as expressões utilizando o pyparsing, embora sejam mais detalhadas são mais legíveis/intuitivas [McGuire 2007].

¹Syslog é um mecanismo padronizado para atividade de logging em sistemas de computador, <<http://www.syslog.org/>>

²pyparsing.wikispaces.com

4.3. Módulo de Comunicação com o Servidor Central

É o módulo previsto para ser responsável pela comunicação com o componente Servidor Central, enviando os eventos coletados para serem armazenados no repositório presente no servidor. Este módulo também realiza a busca periódica no servidor, pelas informações necessárias para execução do DLNA-ML, incluindo os logs que devem ser monitorados e as expressões para normalização e contextualização.

4.4. Módulo de Processamento

Na concepção do módulo de “Processamento” foi considerado o emprego da estratégia de aprendizagem de máquina por meio da técnica de árvores de decisão, onde o sistema aprende a partir de uma base de dados e passa a classificar os novos registros de acordo com as classes do conjunto de treinamento. Optou-se pela utilização da árvore de decisão por ela ser uma das principais técnicas utilizadas para a classificação de eventos e também pelas restrições de utilização em tempo de execução. Este último fato é devido à característica que após o processo de treinamento ser concluído a decisão calculada pela árvore é um processo rápido, uma vez que se baseia em um número limitado de instruções condicionais [Ammar 2015].

5. Cenário de Uso e Testes

A seguir, são apresentados dois cenários de uso desenvolvidos para a avaliação das funcionalidades da DLNA-ML, caracterizando a utilização dos módulos de pré-processamento e processamento.

5.1. Avaliação do Módulo de Pré-processamento

Para demonstrar o funcionamento do módulo de pré-processamento a Figura 2 apresenta um exemplo de utilização em um registro de log da aplicação Shorewall³. Primeiramente é apresentado um registro no seu respectivo formato, em seguida os campos em que devem ser separados o registro e por último, é mostrado o registro formatado.

Para realizar o pré-processamento dos registros foram desenvolvidas expressões com base no formato do log da aplicação, tendo como consequência que os eventos coletados são automaticamente separados em campos, os quais podem receber a adição de dados contextuais, como por exemplo, referentes à geolocalização do endereço IP (*Internet Protocol*).

Como pode ser observado na Figura 2, a visualização dos dados presentes no registro de log se torna facilitada após o pré-processamento, já que o registro é separado em campos. Pode-se observar, comparando o registro de log original e a saída do pré-processamento, que alguns campos foram eliminados, devido ao fato de não possuírem uma informação de interesse para a aplicação. Outro detalhe a ser notado é que foram adicionadas informações relacionadas à geolocalização do IP que acessou o serviço. Essa adição de informações contextuais pode ser útil para as análises que venham a ser realizadas.

³<http://www.shorewall.org/>

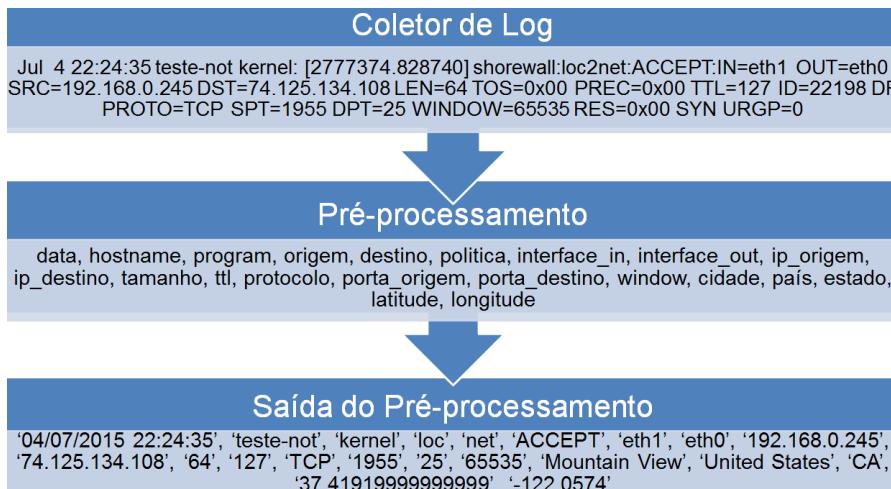


Figura 2. Exemplo de funcionamento do módulo de pré-processamento

5.2. Avaliação do Módulo de Processamento

Com o objetivo de avaliar o módulo de processamento com a estratégia de aprendizagem de máquina, foram utilizados os conjuntos de treinamento e teste *kddcup.data 10 percent*⁴ e *corrected*⁵ respectivamente, sendo a base de dados “KDD Cup 99 Data” considerada umas das principais bases utilizadas na avaliação de mecanismos para detecção de tentativas de ataques a servidores de rede [Elekar et al. 2015].

Os testes foram conduzidos de forma que a conexão pudesse ser classificada em uma das cinco categorias presentes no conjunto de treinamento: DoS (*Denial of Service*), U2R (*User to Root*), R2L (*Remote to Local*), Probe e Normal [Elekar et al. 2015]. Optou-se pelo desenvolvimento de dois classificadores utilizando a técnica de árvores de decisão para uso no módulo de processamento do DLNA-ML. O primeiro trabalha com todos os atributos presentes no conjunto de treinamento e o segundo trabalha somente com 5 atributos (*duration*, *protocol_type*, *service*, *src_bytes*, *dst_bytes*). Foram escolhidos estes 5 atributos pela facilidade de aquisição quando do monitoramento do tráfego da rede em tempo de execução, facilitando assim, a utilização do classificador sem a necessidade de um processamento extra para a inferência de outros campos.

Na Tabela 1 é apresentada uma comparação entre os resultados obtidos para os dois classificadores. Estes resultados representam a porcentagem de conexões corretamente detectadas entre cada uma das categorias analisadas, incluindo as taxas de falso positivo, a qual consiste da classificação de uma conexão normal como sendo de uma categoria de ataque, falso negativo, que ocorre quando uma conexão de uma categoria de ataque é classificado como normal e a taxa de acertos geral do classificador, que consiste na divisão do número de conexões classificadas corretamente pelo número de conexões analisadas.

⁴http://kdd.ics.uci.edu/databases/kddcup99/kddcup.data_10_percent.gz

⁵<http://kdd.ics.uci.edu/databases/kddcup99/corrected.gz>

Tabela 1. Resultados obtidos pelos classificadores

Categoria	Classificador com 41 atributos	Classificador com 5 atributos
Normal	98,18%	98,68%
DoS	99,99%	99,93%
U2R	17,95%	53,85%
R2L	25,71%	15,38%
Probe	99,20%	68,66%
Falso Positivo	1,82%	1,32%
Falso Negativo	1,77%	2,14%
Acertos Geral	98,07%	97,68%

De forma geral, ambos classificadores apresentaram bons resultados para as categorias de conexões analisadas, alcançando taxas aceitáveis de falso positivo e falso negativo. As taxas de acertos mais baixas das categorias R2L e U2R ocorrem devido ao número limitado de conexões destas categorias em comparação com as outras presentes no conjunto de treinamento, já que o classificador necessita de um número significativo de conexões para aprender a classificar de forma satisfatória as conexões.

Algumas diferenças foram percebidas com relação às categorias Probe e U2R. Na categoria Probe, o classificador com atributos reduzidos teve um desempenho relativamente inferior, o que se deve em grande parte à eliminação dos atributos calculados, os quais analisavam as demais conexões em uma janela de 2 segundos, já que esta categoria de ataque costuma gerar uma variedade de conexões em um intervalo pequeno de tempo.

No caso da categoria U2R, o classificador com atributos reduzidos alcançou um desempenho superior em relação ao outro classificador. Acredita-se que esta melhora se deve ao fato da eliminação de atributos, pois possivelmente alguns destes atributos estavam dificultando o aprendizado das classificações das conexões da categoria U2R.

Apesar do classificador com atributos reduzidos ter alcançado resultados inferiores em relação ao outro classificador, ele apresenta a vantagem de poder ser aplicado no momento da coleta das conexões, não sendo necessário outro tipo de processamento para calcular valores de outros atributos. Destaca-se que com a utilização do classificador em tempo de execução, este pode ser utilizado para apoiar à detecção de ataques à rede, fornecendo a categoria do ataque, e consequentemente, facilitando a tomada de decisão do administrador do sistema.

6. Considerações Finais

Com o intuito de automatizar a coleta de eventos e a detecção de situações que possam impactar na segurança do ambiente, este trabalho desenvolveu uma abordagem para realização automática da coleta dos registros de log de aplicações e do tráfego da rede. Para isso, a solução desenvolvida conta com módulos para a normalização destes registros e a contextualização dos dados presentes nos mesmos.

Com a concepção e prototipação da DLNA-ML, foi possível fornecer flexibilidade e heterogeneidade nos aspectos referentes à coleta de eventos, visto a possibilidade de recebimento de eventos pelo protocolo Syslog. Além disso, a solução oferece suporte ao

desenvolvimento de novas expressões, com uma sintaxe alternativa à expressões regulares para normalização e contextualização de logs com diferentes formatos.

Outra contribuição deste trabalho é a possibilidade de classificar as conexões capturadas pelo coletor de Tráfego da Rede no momento de sua captura, sendo um diferencial em relação a outros trabalhos que se propõem a realizar a classificação somente de registros históricos. Nos testes realizados, o desempenho do classificador referente à taxa de acertos foi satisfatório, demonstrando que o classificador utilizando a técnica de árvores de decisão pode ser utilizado para classificar as conexões capturadas, trazendo um novo mecanismo para facilitar a tomada de ações por parte do administrador dos sistemas.

Como trabalhos futuros, espera-se desenvolver expressões para tratamento dos registros de log de outras aplicações, estendendo a solução criada e aplicar técnicas de visualização de dados para facilitar a análise dos resultados. Além disso, avaliar outras técnicas de aprendizagem de máquina e outros conjuntos de dados para treinamento da técnica escolhida.

Referências

- Ammar, A. (2015). A decision tree classifier for intrusion detection priority tagging. *Journal of Computer and Communications*, 3.
- Arjunwadkar, N. M. and Parvat, T. J. (2015). An intrusion detection system, (ids) with machine learning (ml) model combining hybrid classifiers. *Journal of Multidisciplinary Engineering Science and Technology (JMEST)*.
- Campos, L. M. L. and Lima, A. S. (2012). Sistema para detecção de intrusão em redes de computadores com uso de técnica de mineração de dados. *V Congresso Tecnológico Infobrasil, Fortaleza. anais do V Congresso Tecnológico Infobrasil*.
- Cansian, A. M. (2001). Conceitos para perícia forense computacional. *Anais VI Escola Regional de Informática da SBC, Instituto de Ciências Matemáticas e Computação de São Carlos, USP (ICMC/USP), São Carlos, SP, São Carlos, SP, 30 de abril a 02 de maio de 2001.*, pages p.141–156.
- CLEMENTE, R. G. (2008). Uma arquitetura para processamento de eventos de log em tempo real. Dissertação de mestrado, Pontifícia Universidade Católica do Rio de Janeiro - PUC-RIO.
- Elekar, K., Waghmare, M., and Priyadarshi, A. (2015). Use of rule base data mining algorithm for intrusion detection. In *Pervasive Computing (ICPC), 2015 International Conference on*, pages 1–5.
- Grégio, A. R. A. (2008). Aplicação de técnicas de data mining para a análise de logs de tráfego tcp/ip. Dissertação de mestrado do curso de pós-graduação em computação aplicada, Instituto Nacional de Pesquisas Espaciais/INPE, São José dos Campos/SP.
- Hoepers, C. and Steding-Jessen, K. (2003). Análise e interpretação de logs. NIC BR Security Office(NBSO) Comitê Gestor da Internet no Brasil.
- McGuire, P. (2007). *Getting Started with Pyparsing*. O'Reilly, first edition.
- Weiser, M. (1991). The computer for the 21st century. *Scientific American*, 265(3):66–75.

Mecanismos de Segurança aplicados a Interface para o Sistema de Roteamento (I2RS)

Joel Molling¹, Jeferson Campos Nobre¹

¹Universidade do Vale do Rio dos Sinos (UNISINOS)
CEP 93.022-000 – São Leopoldo – RS – Brasil

Abstract. *The frequent security incidents related to infrastructure networks and the Internet, emphasize the importance of management mechanisms to apply security controls to the network elements. The proposed project presents the evaluation of a model based on I2RS, which aims to add security to the process management and configuration of existing routing infrastructure of a computer network.*

Resumo. *Os constantes incidentes de segurança relacionados a infraestrutura de redes e à Internet, ressaltam a importância de mecanismos de gerenciamento que permitam aplicar controles de segurança aos elementos de rede. O trabalho proposto apresenta a avaliação de um modelo baseado em I2RS, que visa agregar segurança no processo de gerenciamento e configuração das atuais infraestruturas de roteamento de uma rede de computadores.*

1. Introdução

O advento da Internet facilitou à expansão dos negócios e possibilitou às organizações proverem interoperabilidade às suas operações. A necessidade de manter um ambiente de interconexão de rede estável, demonstrou o quanto importantes e imprescindíveis são os mecanismos de administração em uma rede de computadores. As tecnologias de administração deveriam possibilitar a gestão, o controle e a manutenção dos dispositivos presentes no ambiente de rede, sem que houvessem impactos na operação e consequentemente as atividades de uma organização. Para [Shin et al. 2012], a administração de uma rede já se tornava uma tarefa complexa e crítica, uma vez que as tecnologias de rede já não suportavam mais a constante evolução da infraestrutura de redes de computadores.

As tecnologias existentes não proporcionavam uma gestão centralizada do ambiente, permitindo que ameaças pudessem comprometer a rede sem uma imediata identificação e correção dos problemas. A complexidade da análise necessária para solucionar os problemas, era proporcional ao tamanho da rede. Os problemas quando relacionados a infraestrutura de roteamento da rede, eram ainda mais difíceis de se identificar, uma vez que protocolos como RIP (*Routing Information Protocol*), EIGRP (*Enhanced Interior Gateway Routing Protocol*) e OSPF (*Open Shortest Path First*), não eram providos de grandes mecanismos de segurança, aumentando o vetor de ataque.

O novo paradigma de Redes Definidas por Software (*Software-Defined Networking* – SDN), permitiu através do conceito de abstração dos elementos de rede, a gestão e o controle do ambiente de forma centralizada [Drutskoy et al. 2013]. Segundo [McKeown et al. 2008], SDN ao contrário do modelo tradicional, proporciona uma maior flexibilidade às tarefas de administração de uma rede, possibilitando análises e correções mais eficazes.

SDN sendo flexível, possibilitou um ambiente para o desenvolvimento de novas tecnologias que pudessem aprimorar e também agregar novas funcionalidades ao conceito de redes programáveis. Uma dessas tecnologias, a Interface para o Sistema de Roteamento (*Interface to the Routing System - I2RS*), trouxe como premissa o desenvolvimento de uma camada que possibilitasse o devido controle e gerenciamento da infraestrutura de roteamento da rede. O I2RS incorporou ao conceito de SDN, agilidade e eficiência na manipulação dos protocolos e regras de roteamento que mantém uma rede [Hares and White 2013].

A camada de gerenciamento criada pelo I2RS, por mais que forneça características como a análise de comportamento dos protocolos, até o presente momento não é capaz de proporcionar segurança às tarefas de administração da infraestrutura de roteamento, visto que muitas vulnerabilidades estão localizadas diretamente na estrutura ou no comportamento individual dos protocolos. Como alternativa para solucionar os problemas de segurança do I2RS, é proposta a utilização do protocolo de configuração de redes (*Network Configuration Protocol - NETCONF*), em conjunto com a linguagem de modelação de dados YANG. O NETCONF/YANG no contexto I2RS, tem a função de transportar as requisições I2RS na rede, fornecer segurança para a comunicação entre cliente e agentes I2RS, e prover segurança ao armazenamento das informações nos elementos da rede [Haas 2015].

O trabalho se propõe a avaliar o modelo NETCONF/YANG como sendo o mecanismo de segurança definido pelo I2RS e objetiva compará-lo aos mecanismos de segurança providos pelos atuais protocolos de roteamento.

2. Interface para o Sistema de Roteamento

[Hu et al. 2014], descrevem o conceito de arquitetura SDN como não sendo algo inovador por completo, fazendo referência a computação em nuvem que já possibilitava a abstração do sistema operacional das intensas instruções de hardware. A separação que SDN proporciona ao universo de redes de computadores, possibilita um ganho de desempenho no quesito de gerenciamento, controle e manipulação dos dados trafegados entre as redes [Hu et al. 2014]. Com SDN é possível controlar o fluxo de dados, gerenciar políticas de segurança e prioridades de tráfego através de um ponto único de controle, sem a necessidade de configuração individual de cada equipamento [ONF 2012].

Para [Hu et al. 2014], as características de SDN ficam evidenciadas quando é feito um comparativo com as tradicionais redes de computadores. Entre as principais características e vantagens em comparação ao modelo atual, é possível destacar a inteligência e velocidade, que torna mais eficiente o uso dos recursos, otimizando a distribuição de carga e dando mais agilidade as transmissões [Hu et al. 2014]. SDN proporciona facilidade no gerenciamento da rede, pois com a utilização de um ponto central para gerenciamento dos dispositivos de comutação, permite agilidade e eficiência às tarefas de configuração [Shin et al. 2012].

Entre as diversas arquiteturas existentes para a elaboração de um modelo de SDN, o *Internet Research Task Force* (IRTF), através de um de seus grupos de pesquisa denominado de *Software-Defined Networking Research Group* (SDNRG), definiu uma arquitetura de SDN. O modelo definido é devidamente abordado na RFC 7149, que descreve o conceito de separação entre os planos de controle e de dados.

A arquitetura de SDN possui as funções de gerenciamento de rede acopladas a camada de controle (*Control Layer*), que baseada em Software possibilita a sua programação conforme necessário, dando origem ao nome SDN. Essa arquitetura faz também com que a camada de aplicação (*Application Layer*) interprete a rede como sendo algo único [ONF 2012]. O conceito de redes programáveis delegou aos administradores da rede um poder de gerenciar, configurar e proteger a rede através de automatizações aplicadas à camada de controle. A arquitetura SDN permite ainda a implementação de *Application Program Interfaces* (APIs), que possibilitam a inclusão de serviços de rede como engenharia de tráfego, qualidade de serviço, controle de acesso e segurança [ONF 2012]. Entre as APIs de SDN, destacam-se duas: as interfaces sul (*SouthBound*) e norte (*NorthBound*).

Segundo [Kreutz et al. 2015], *NorthBounds* são as APIs que fornecem uma interface para o desenvolvimento de aplicações e as responsáveis por abstrair as instruções utilizadas pelas interfaces sul. Essas instruções consumidas pelas *SouthBounds*, são necessárias para programar e configurar os dispositivos presentes na camada de infraestrutura [Kreutz et al. 2015].

Southbounds são interfaces que buscam facilitar e tornar mais eficiente o controle da rede, permitindo que o controlador faça alterações dinâmicas de acordo com a necessidade e da forma mais ágil possível. Segundo [Kreutz et al. 2015], as interfaces sul são responsáveis por definir os protocolos de comunicação entre os dispositivos de encaminhamento de tráfego e o plano de controle. Soluções conhecidas como *southbounds* abertos, são o OpenFlow, o *Simple Network Management Protocol* (SNMP) e o NETCONF. O Openflow através de seu controlador, propõe uma administração centralizada dos dispositivos de rede, permitindo controle e manipulação sobre o fluxo dos dados trafegados entre a rede [Rothenberg et al. 2010]. A característica de utilizar o conceito de fluxo, apesar de altamente eficaz, não exerce funções de administração sobre a estrutura de roteamento da rede, permitindo uma distinção entre OpenFlow e I2RS, que oferece uma camada de gerenciamento de mais alto nível, operando em conjunto com os protocolos de roteamento [FUGITSU 2014].

O I2RS, projeto de um grupo de trabalho¹ do *Internet Engineering Task Force* (IETF), foi criado com o propósito de viabilizar soluções para o atual sistema de roteamento das redes de computadores sobre o conceito de SDN. Entre os objetivos, estão a elaboração de uma arquitetura de alto nível para o I2RS, incluindo gerenciamento de políticas de roteamento e segurança, e análise do funcionamento de protocolos de roteamento, aprimorando gerenciamento e desempenho.

A arquitetura do I2RS é composta por um cliente e um agente, sendo o primeiro, responsável pelo gerenciamento e dispersão das regras através do canal de comunicação estabelecido. Os agentes, localizados no elemento de rede, são responsáveis por aplicar as requisições encaminhadas pelos clientes. De acordo com o *Internet Draft* proposto por [Hares and White 2013], é possível estruturar uma arquitetura de um ou mais agentes I2RS. Em arquiteturas simples onde há somente um cliente, todo o gerenciamento é centralizado, ao contrário de arquiteturas mais robustas que incluem múltiplos clientes I2RS, cada qual com uma função específica e ambos operando em conjunto com os diversos agentes remotos.

¹<http://datatracker.ietf.org/wg/i2rs>

2.1. Gerenciamento do Sistema de Roteamento

Segundo [Haas 2015], para alcançar os objetivos propostos pelo projeto, era necessário que o I2RS pudesse estabelecer uma infraestrutura capaz de manipular o estado de configuração dos elementos da rede, provendo uma interface segura e capaz de controlar a estrutura de roteamento de uma rede. Para possibilitar o transporte das requisições I2RS e atender aos requisitos de segurança, como autenticação mútua, controle de acesso a estrutura de dados de cada dispositivo, confidencialidade e integridade das informações, o grupo de trabalho de I2RS do IETF definiu o NETCONF como mecanismos de suporte às funções de transporte, segurança e configuração necessárias para a implementação do I2RS. O NETCONF é um protocolo de gerenciamento de rede, mantido pelo IETF e publicado sobre a RFC 4741. Segundo [Wallin and Wikström 2011], o NETCONF provê mecanismos que permitem a instalação, exclusão e manipulação de configuração em dispositivos de rede enquanto estão em operação, através da utilização do conceito de armazenamento lógico.

[Choi et al. 2004] apresenta que o conceito de armazenamento lógico do NETCONF utiliza uma codificação de dados baseada em XML, onde todas ações são realizadas através de chamadas de procedimento remoto (*Remote Procedure Call - RPC*), que permitem a comunicação entre gerente e agente NETCONF. As chamadas RPC podem ser estabelecidas por protocolos como BEEP (*Blocks Extensible Exchange Protocol*), SSH (*Secure Shell Transport*), SSL (*Secure Sockets Layer*), TLS (*Transport Layer Security*) e SOAP (*Simple Object Access Protocol*) [Huang et al. 2009]. Segundo [Schonwalder et al. 2010], a camada RPC sobre a estrutura de protocolos de transporte são a base funcional do NETCONF, com o nível de manipulação e estruturação dos dados ocorrendo na camada de operação da arquitetura do NETCONF.

A camada mais inferior da arquitetura, denominada de camada de transporte seguro, nativamente utiliza o protocolo SSH para transporte de suas mensagens, que o torna muito semelhante as interfaces de linha de comando (Command Line Interfaces - CLI) proprietárias, que por sua vez são embarcadas aos dispositivos [Wallin and Wikström 2011]. O transporte através de SSH é o responsável pelo NETCONF atender a três importantes quesitos de segurança, como confidencialidade e integridade das informações, além da existência de um processo de autenticação.

A estrutura de armazenamento lógico, definida com o uso de NETCONF, é parte fundamental para implementação do I2RS, uma vez que é possível trabalhar com mecanismos de injeção de configuração em bases efêmeras ou não efêmeras, no caso do NETCONF, uma base *writable-running* ou *writable-running + startup*. O processo de injeção de configuração, além de permitir a utilização do conceito de bases de dados efêmeras, designou a linguagem de modelação de dados YANG como padrão de estrutura para as informações que são enviadas aos dispositivos com NETCONF e gerenciados pelo I2RS. A definição do YANG também é fundamental para garantir a confiabilidade das informações I2RS, principalmente quanto ao controle de acesso às informações armazenadas nos dispositivos.

O YANG, uma linguagem de modelação de dados padronizada e definida pelo IETF, revela uma abordagem distinta dos modelos XML atuais. [Schonwalder et al. 2010], retratam que o objetivo do YANG é ser uma linguagem legível e de alta compreensão para modelar os dados NETCONF. Segundo [Xu and Xiao 2008],

Tabela 1. Protocolos sem mecanismos de segurança x Ameaças

Protocolos / Ameaças	Eavesdropping	Personificação	Spoofing	Null Session	Session Hijack	Replay	Denial of Service
RIP	V	V	V	V	V	V	V
RIPv2	V	V	V		V	V	V
EIGRP	V	V	V		V	V	V
OSPF	V	V	C		V	V	V

V=Vulnerável; C=Combinação;

o formato de linguagem XML foi introduzido ao protocolo de configuração de redes para padronizar a linguagem responsável por modelar os dados. Embora já existam esquemas XML como XSD e RelaxNG em utilização junto ao NETCONF, ambos possuem características que dificultam a compreensão da linguagem, principalmente quando grande parte das extensões do protocolo são utilizadas, criando a necessidade de utilização do YANG [Schonwalder et al. 2010].

[Xu and Xiao 2008] em sua pesquisa, fazem um comparativo entre um Schema XML, YANG e SMI. Nesse estudo, [Xu and Xiao 2008] apresentam o quanto a linguagem YANG pode ser superior aos tradicionais métodos XML empregados com o NETCONF. Entre as questões abordadas, o YANG é amplamente superior no quesito segurança, prestando por conceitos como confidencialidade e integridade dos dados, além de possuir mecanismos de controle de acesso e bloqueio às informações.

3. Trabalho Proposto - Avaliação

O objetivo da avaliação apresentada nesta Seção é efetuar um comparativo entre os mecanismos de segurança atualmente aplicados aos protocolos de roteamento e as soluções de segurança providas pelo modelo de gerenciamento e controle proposto pelo I2RS, que atua em conjunto com o protocolo NETCONF e a linguagem de modelação de dados YANG. Apresenta-se uma análise qualitativa que demonstra a quais ameaças os protocolos de roteamento e o NETCONF estão vulneráveis.

3.1. Mecanismos de Segurança x Ataques

A avaliação consiste em uma análise qualitativa que apresenta os resultados de uma comparação entre os protocolos de roteamento e ataques aos quais estão vulneráveis. Primeiramente, a Tabela 1 demonstra a quais ataques estão suscetíveis os protocolos de roteamento quando não há ou não estão ativos os mecanismos de segurança.

O protocolo RIP em sua versão 1 não possui mecanismos de segurança agregados a sua estrutura, o que permite que ataques como *Spoofing* e *Replay* possam explorar vulnerabilidades do protocolo e obter vantagens sobre a sua estrutura e implementação. Apesar da versão 2 do RIP já possuir uma estrutura mais robusta, contendo mecanismos de segurança, ela continua sendo vulnerável a ataques maliciosos caso os mecanismos de segurança não sejam ativados pelos administradores do ambiente. O EIGRP com a característica de ser um protocolo híbrido e tendo o seu funcionamento baseado em um algoritmo muito semelhante ao utilizado pelo RIP, está sujeito as mesmas ameaças.

O protocolo OSPF é baseado no algoritmo de menor rota primeiro (*Shortest Path First* - SPF), que diferente dos protocolos anteriormente citados, foi projetado com uma infraestrutura mais robusta a qual nativamente já aplica alguns controles de segurança.

Tabela 2. Protocolos com mecanismos de segurança x Ameaças

Protocolos / Ameaças	Eavesdropping	Personificação	Spoofing	Null Session	Session Hijack	Replay	Denial of Service
RIP	V	V	V	V	V	V	V
RIPv2	V	V	V		V	V	V
S-RIP			C				V
EIGRP	V	V	V		V	V	V
OSPFv2	V	V	C		V	V	V
Digital Signature OSPF			C				
NETCONF over SSH							
NETCONF over TLS							

V=Vulnerável; C=Combinação;

Entre os mecanismos são aplicados controles aos números de sequência das mensagens de atualização e idade dos pacotes, o que inibe diretamente ataques de *Spoofing*. O mecanismo atribuído aos pacotes LSA não evita por completo tentativas de *Spoofing*, mas torna-as possíveis somente em casos de combinação, onde o atacante primeiro compromete algum roteador através de outro ataque para então utilizá-lo como recurso para uma nova ação.

A Tabela 2 ilustra que o cenário de ameaças é completamente diferente quando os mecanismos de segurança são devidamente aplicados ao ambiente, reduzindo o vetor de ataques e possibilitando mais segurança à infraestrutura de rede.

Os mecanismos de segurança incorporados a estrutura dos protocolos RIPv2, EIGRP e OSPFv2, se resumem a utilização da função de hash MD5 durante o processo de autenticação dos dispositivos adjacentes. Apesar dessa função criptográfica prover segurança na troca de informações, garantindo a verificação de autenticidade e integridade das mensagens, o MD5 só inibe ataques como *Spoofing*, *Session Hijack* e personificação, quando estes são executados diretamente à estrutura de roteamento, sem a combinação ou intervenção de outro ataque. O problema de não evitar ataques combinados, compromete a segurança de toda a rede caso um atacante consiga explorar uma vulnerabilidade de configuração de um dispositivo, permitindo acesso irrestrito ao mesmo e a rede por completo.

A função MD5, apesar de ainda ser utilizada em larga escala, não é mais recomendada para novas implementações, visto que a técnica de hash aplicada no MD5 já é criptograficamente insegura e passível de colisão, uma técnica que busca encontrar dois conjuntos de dados com o mesmo hash [Kuznetsov 2014]. [Wang et al. 2004] demonstra colisões de MD5 em seu primeiro trabalho e posteriormente apresenta dois certificados digitais com o mesmo hash MD5 [Lenstra et al. 2005].

O S-RIP, uma extensão do protocolo RIP, foi proposto por [Wan et al. 2004] para maximizar a segurança e reduzir o vetor de ataques sobre os protocolos que utilizam o algoritmo de vetor de distância. Os autores apresentam uma proposta de segurança que tem como objetivos a prevenção de ataques de personificação, espionagem e roubo de sessão. Em sua abordagem, [Wan et al. 2004] utilizam chaves compartilhadas distintas para cada par de roteadores adjacentes, o que aumenta a complexidade do ambiente e dificulta o sucesso do atacante, mesmo com o processo de autenticação sendo feito através de MD5. Para o controle de ataques como *Spoofing*, espionagem e roubos de sessão, o

S-RIP trabalha com um mecanismo que avalia a consistência dos pacotes através de testes de aproximação.

[Murphy and Badger 1996] apresentam uma proposta de melhorar o protocolo OSPF através da implementação de mecanismos de segurança baseados em criptografia assimétrica, especialmente com a utilização do conceito de assinatura digital (*Digital Signature*). O *Digital Signature OSPF* aumenta a força do processo de autenticação entre os dispositivos da infraestrutura de roteamento, provendo um ambiente seguro às trocas de mensagens do protocolo e inibindo ataques antes possíveis contra a função de hash MD5. Apesar de possuir um mecanismo forte de segurança, o *Digital Signature OSPF* não evita ataques de *Spoofing* pois geralmente são iniciados de dentro da rede e através de um dispositivo comprometido, burlando a segurança do processo assimétrico.

4. Considerações Parciais

A escolha do projeto I2RS pelo NETCONF, apesar de obter ganhos de gerenciamento, também é relacionada as técnicas de segurança que o mesmo agrega ao modelo proposto. O NETCONF trabalhando sobre SSH, proporciona segurança a níveis de autenticação, integridade e confidencialidade, superando em qualidade a segurança dos protocolos de roteamento durante o transporte das mensagens pela infraestrutura de rede. O ganho de segurança com NETCONF permite uma eficácia na proteção à ameaças dos mais variados aspectos de ataque, inclusive ataques de negação de serviço. A proteção contra DoS é possível devido a união das características do I2RS, que identifica alterações e comportamentos anômalos na topologia de rede, com os recursos fornecidos pelo NETCONF, que através de uma comunicação extremamente ágil, possibilita injeções de configuração do cliente I2RS nos agentes remotos até mesmo durante a execução de um ataque.

Nesse momento, objetiva-se avaliar de forma experimental e em laboratório os cenários existentes, a fim de medir de forma quantitativa os ganhos de desempenho e gerenciamento que o modelo NETCONF/YANG/I2RS possibilita.

Referências

- [Choi et al. 2004] Choi, M.-J., Choi, H.-M., Hong, J., and Ju, H.-T. (2004). Xml-based configuration management for ip network devices. *Communications Magazine, IEEE*, 42(7):84–91.
- [Drutskoy et al. 2013] Drutskoy, D., Keller, E., and Rexford, J. (2013). Scalable network virtualization in software-defined networks. *Internet Computing, IEEE*, 17(2):20–27.
- [FUGITSU 2014] FUGITSU (2014). Technical report - carrier software defined networking (sdn). Disponível em: <<http://stakeholders.ofcom.org.uk/binaries/research/telecoms-research/SDNReport.pdf>>.
- [Haas 2015] Haas, J. (2015). I2rs requirements for netmod/netconf. Internet-Draft draft-haas-i2rs-netmod-netconf-requirements-01, IETF Secretariat. <http://www.ietf.org/internet-drafts/draft-haas-i2rs-netmod-netconf-requirements-01.txt>.
- [Hares and White 2013] Hares, S. and White, R. (2013). Software-defined networks and the interface to the routing system (i2rs). *Internet Computing, IEEE*, 17(4):84–88.

- [Hu et al. 2014] Hu, F., Hao, Q., and Bao, K. (2014). A survey on software-defined network and openflow: From concept to implementation. *Communications Surveys Tutorials, IEEE*, 16(4):2181–2206.
- [Huang et al. 2009] Huang, J., Zhang, B., Li, G., Gao, X., and Li, Y. (2009). Challenges to the new network management protocol: Netconf. In *Education Technology and Computer Science, 2009. ETCS '09. First International Workshop on*, volume 1, pages 832–836.
- [Kreutz et al. 2015] Kreutz, D., Ramos, F., Esteves Verissimo, P., Esteve Rothenberg, C., Azodolmolky, S., and Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76.
- [Kuznetsov 2014] Kuznetsov, A. (2014). An algorithm for md5 single-block collision attack using high-performance computing cluster.
- [Lenstra et al. 2005] Lenstra, A., Wang, X., and de Weger, B. (2005). Colliding x.509 certificates. Cryptology ePrint Archive, Report 2005/067. <http://eprint.iacr.org/>.
- [McKeown et al. 2008] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74.
- [Murphy and Badger 1996] Murphy, S. and Badger, M. (1996). Digital signature protection of the ospf routing protocol. In *Network and Distributed System Security, 1996.. Proceedings of the Symposium on*, pages 93–102.
- [ONF 2012] ONF (2012). Software-defined networking - the new norm for networks. Disponível em: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf>.
- [Rothenberg et al. 2010] Rothenberg, C., Nascimento, M., Salvador, M., and Magalhães, M. (2010). Openflow e redes definidas por software: um novo paradigma de controle e inovação em redes de pacotes. *Cad. CPqD Tecnologia*, 7(1):65–76.
- [Schonwalder et al. 2010] Schonwalder, J., Bjorklund, M., and Shafer, P. (2010). Network configuration management using netconf and yang. *Communications Magazine, IEEE*, 48(9):166–173.
- [Shin et al. 2012] Shin, M.-K., Nam, K.-H., and Kim, H.-J. (2012). Software-defined networking (sdn): A reference architecture and open apis. In *ICT Convergence (ICTC), 2012 International Conference on*, pages 360–361.
- [Wallin and Wikström 2011] Wallin, S. and Wikström, C. (2011). Automating network and service configuration using netconf and yang. In *Proceedings of the 25th International Conference on Large Installation System Administration, LISA'11*, pages 22–22, Berkeley, CA, USA. USENIX Association.
- [Wan et al. 2004] Wan, T., Kranakis, E., and van Oorschot, P. (2004). S-rip: A secure distance vector routing protocol. In Jakobsson, M., Yung, M., and Zhou, J., editors, *Applied Cryptography and Network Security*, volume 3089 of *Lecture Notes in Computer Science*, pages 103–119. Springer Berlin Heidelberg.
- [Wang et al. 2004] Wang, X., Feng, D., Lai, X., and Yu, H. (2004). Collisions for hash functions md4, md5, haval-128 and ripemd. Cryptology ePrint Archive, Report 2004/199. <http://eprint.iacr.org/>.
- [Xu and Xiao 2008] Xu, H. and Xiao, D. (2008). Data modeling for netconf-based network management: Xml schema or yang. In *Communication Technology, 2008. ICCT 2008. 11th IEEE International Conference on*, pages 561–564.

Um Estudo para Identificação e Mitigação de *IP Spoofing* em Redes IPv6 utilizando SDN

Manoel F. Ramos¹, Rafael B. Ávila¹

¹Escola Politécnica – Universidade Vale do Rio dos Sinos (UNISINOS)
Av. Unisinos, 950 – Bairro Cristo Rei – 93.022-000 – São Leopoldo – RS – Brasil

manoel@dropreal.com, rafael.avila@gmail.com

Resumo. A técnica de *IP Spoofing* é empregada em diversos tipos de ataques cibernéticos para forjar o real endereço de rede do atacante. Com a expansão do uso do IPv6, projeta-se que a utilização dessa técnica será intensificada, principalmente porque o protocolo NDP — parte do IPv6 responsável pela descoberta de vizinhança — não possui mecanismos de validação dos endereços de rede e de enlace inseridos em seu cabeçalho. Este artigo apresenta uma análise sobre as possibilidades de aplicação de spoofing em IPv6 e os principais mecanismos e protocolos utilizados em sua exploração. Além disso, apresenta uma proposta para identificar e mitigar o uso da técnica de *IP Spoofing* na rede local de origem através do uso de SDN em redes IPv6.

Abstract. *IP Spoofing* is used in various types of cyber attacks in order to forge the real network address of an attacker. Its use is expected to be intensified with the expansion of IPv6, mainly because the NDP protocol—part of IPv6 responsible for neighborhood discovery—does not present validation mechanisms for network and link layer addresses employed in its header. This paper presents an analysis on the possibilities of applying IPv6 spoofing and the main mechanisms and protocols used in exploiting it. Furthermore, it proposes a method for identification and prevention of *IP spoofing* at the source local network by using SDN in IPv6 networks.

1. Introdução

A maioria dos ataques cibernéticos utilizam técnicas de falsificação (*Spoofing*) do endereçamento de rede de origem do atacante, tanto para ampliar ou redirecionar respostas de comunicação a um determinado alvo, quanto para forjar o real endereço de origem do computador do atacante. Esta técnica é conhecida como *IP Spoofing*. [Tanase 2003].

O *IP Spoofing* foi inicialmente discutido através de meios acadêmicos na década de 1980 onde foi descoberta uma falha de segurança no protocolo TCP (*Transmission Control Protocol*), conforme especificado por [Morris 1985] e aprofundado por [Bellovin 1989]. Essas falhas já foram solucionadas, porém a técnica de *IP Spoofing* ainda é muito utilizada nos dias de hoje. O protocolo IP é uma fraqueza presente nos sistemas utilizados na internet atualmente pois permite que o endereço IP de origem seja alterado ou seja, um atacante pode, além de utilizar o *IP Spoofing* para garantir o seu anonimato, enviar pacotes IPs com o endereço de origem falsificado para lançar ataques direcionados como *Non-Blind Spoofing*, *Blind Spoofing*, *Man in The Middle* (MiTM), DoS, *Decoy Scan*, entre outros. [Mukaddam et al. 2014].

O objetivo deste estudo é compreender o funcionamento da técnica de IP *Spoofing* em redes IPv6, destacando a preocupação da comunidade acadêmica sobre o tema, assim como apresentar um novo método delineado para mitigar e combater a sua usabilidade utilizando Redes Definidas por Softwares (SDN) na origem do tráfego.

2. Fundamentação Teórica

Nesta seção é apresentada uma breve descrição sobre o protocolo IPv6, o protocolo NDP e SDN.

2.1. IPv6

O IPv6 possui diversas melhorias perante o seu antecessor, entre estas destaca-se o cabeçalho no tamanho de 40 *bytes* sendo mais simples e versátil, aumento do tamanho do endereço de 32 *bits* (IPv4) para 128 *bits*, possibilidade de redução de fluxo e propriedade (rotulação de pacotes) e a substituição do protocolo ARP, utilizado pelo IPv4 para a resolução de endereços MAC, pelo protocolo NDP (*Neighbor Discovery Protocol*). [Kurose and Ross 2006].

Conforme [Narten et al. 2007], cada nó (dispositivos de rede que utilizam IPv6) utiliza o NDP para encontrar, identificar e registrar em *cache* os endereços MAC de nós vizinhos e se tornarem conhecidos, permitindo efetuar uma conectividade de forma rápida entre eles. Um nó utiliza o NDP para manter informações sobre os vizinhos que são ou não acessíveis. Quando o caminho para acessar um determinado nó já identificado anteriormente esteja indisponível para efetuar a conectividade, o NDP procura de forma ativa e rápida os vizinhos mais próximos para estabelecer a comunicação.

2.2. Neighbor Discovery Protocol

O protocolo NDP corresponde com a combinação do protocolo ARP [Plummer 1982], ICMP *Router Discovery Mensagens* [Deering 1991] e do ICMP *Redirect* [Postel 1981], possuindo diversas melhorias e novas funcionalidades. Além de efetuar a resolução de endereços da camada de rede com a camada de enlace, o NDP permite a descoberta de roteadores e nós, efetua a autoconfiguração de endereçamentos, permite que roteadores anunciem o MTU de seus vizinhos, detecta falhas de *links*, permite a utilização do *link local* para identificar exclusivamente roteadores, permite definir o limite até 255 saltos sem interromper a comunicação, além de utilizar o ICMPv6 para permitir a autenticação da camada de rede conforme o mecanismo definido pelas políticas de segurança.

Para a troca de mensagens, o NDP define cinco tipos de mensagens ICMPv6 para efetuar trocas de mensagens necessárias para a execução de determinadas funções, sendo elas as mensagens RS (*Router Solicitation*), que são originadas por um determinado *host* para solicitar que um roteador envie uma mensagem RA (*Router Advertisement*). As mensagens RA são enviadas por roteadores para anunciar seus parâmetros de presença e de um determinado *link*. As mensagens NS (*Neighbor Solicitation*) que são enviadas por um host a outro para solicitar o seu endereço MAC. Já as mensagens NA (*Neighbor Advertisement*) informam o endereço MAC a um determinado *host* ou responde a uma solicitação NS. Por fim, a mensagem *Redirect* informa os parâmetros necessários para redirecionar um determinado tráfego de um roteador a outros. [Narten et al. 2007].

Conforme [Barbhuiya et al. 2013], os seguintes ataques utilizam o uso da técnica de IP *Spoofing* através do envio de mensagens NDP em redes IPv6:

- *Neighbor solicitation/advertisement Spoofing;*
- *Man-in-the-Middle attack;*
- *Duplicate Address Detection attack* (Ataque DAD);
- *Neighbor Unreachability Detection attack* (Ataque NUD);
- *Spoofed Router Redirect Message attack;*
- *Replay attack.*

2.3. Redes Definidas por Software

SDN (*Software-Defined Networking*) está mudando a maneira de como as redes são concebidas. O conceito das Redes Definidas por *Software* vem atraindo a atenção de diversos pesquisadores e empresas. SDN possui duas características definidas, a primeira é a separação do plano de controle do plano de dados. O plano de controle decide como lidar com o tráfego da rede, já o plano de dados encaminha o tráfego conforme decisão do plano de controle. A segunda característica é que SDN consolida o plano de controle de modo que o *software* exerça o controle direto sobre o estado dos elementos contidos no plano de dados como, por exemplo, *switches* e roteadores. Este *software* de controle, também denominado “controlador SDN” é uma interface de programação de aplicativos (*Application Programming Interface - API*) definida como, por exemplo, o protocolo OpenFlow. [Fteamster et al. 2014].

Conforme ilustrado na Figura 1, a arquitetura SDN consiste em três camadas, sendo elas a camada superior, a camada inferior e a camada do meio (controlador). A camada superior é a camada de aplicação, que inclui os aplicativos que oferecem serviços como virtualização de rede, *firewall*,平衡adores, gerenciamento de fluxo, etc. A camada de aplicação é captada a partir da camada inferior no qual é a base da camada de rede física, também denominada como camada de infraestrutura. A camada do meio é o controlador SDN, também denominada de camada de controle, esta camada é o elemento crítico e primordial para o funcionamento de SDN pois o controlador remove o plano de controle da rede de *hardware* e passa a tratá-lo como *software*. É importante ressaltar que em SDN, todos os elementos físicos e virtuais são integrados. Desta forma o controlador facilita a gestão da rede automatizada e torna mais fácil de integrar e administrar a infraestrutura com o negócio das organizações. [Kreutz et al. 2015].

3. Trabalhos Relacionados

Existem diversos estudos relacionados ao entendimento e mitigação do uso da técnica de IP *Spoofing* em redes IPv4. Por outro lado, poucos mecanismos de segurança são propostos para a prevenção do uso da técnica em redes IPv6 e muito menos tratam o problema diretamente em sua origem.

[Barbhuiya et al. 2013] apresentam um IDS ativo para detecção e prevenção de ataques baseados nas fragilidades do protocolo NDP em redes IPv6, objetivando que uma determinada rede seja protegida caso sofra ataques que utilizam IP *Spoofing* através das mensagens do NDP. Os autores afirmam que a solução é eficaz para a validação de endereços MAC em redes IPv6. Os algoritmos desenvolvidos, chamados de NS_HANDLER() e NA_HANDLER() e os seus submódulos VERIFY_IP-MAC() e RESPONSE_ANALYSER() se destacam pela simplicidade do processo para a identificação e a validação de endereçamentos MAC e IPv6.

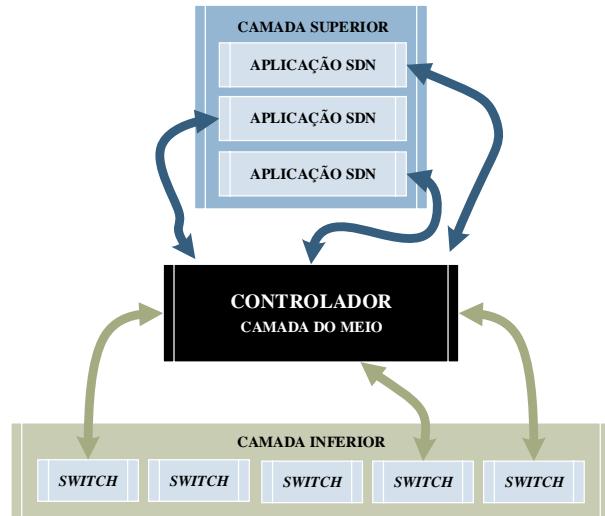


Figura 1. Arquitetura básica de SDN.

[Yao et al. 2011] contribuem com a implementação do mecanismo denominado VAVE (*Virtual source Address Validation Edge*), desenvolvido através do *framework* SAVI (*Source Address Validation Improvement*), o qual emprega o uso do protocolo OpenFlow [McKeown et al. 2008] para validar os endereços de origem do tráfego de entrada em uma rede local. SAVI é discutido através do grupo de trabalho da IETF descrito em [Wu et al. 2013].

[Mowla et al. 2015] propõem um mecanismo de defesa para combater os ataques relacionados ao uso da técnica de IP Spoofing no tráfego de dados recebidos, validando o tráfego legítimo e bloqueando o tráfego de *Spoofing*. A solução é composta por SDN com base na tecnologia CDNi (*Content Distribution Network Interconnection*), juntamente com a tecnologia ALTO (*Application Layer Traffic Optimization*). O objetivo da solução é utilizar SDN para detectar IP Spoofing, seguindo de um mecanismo para alimentar regras em *switches* com suporte à SDN através do controlador utilizando os mapas de marcação (*mark*) fornecidos pelo servidor ALTO. Um fator negativo desta solução é a complexidade de implementação pois, além da infraestrutura SDN baseada em CDNi, no qual poucos fabricantes de equipamentos de rede possuem suporte, a solução necessita de servidores e clientes ALTO em cada uma das redes gerenciadas, sendo um item indispensável para o processo de detecção de IP Spoofing.

[Yan et al. 2011], observando as fragilidades do protocolo IP, o qual não possui mecanismos de validação dos endereços de origem e a real expansão do uso do IPv6, desenvolveram um experimento de implementação do SAVI em uma rede local. É importante salientar que o SAVI utiliza funções de acompanhamento para filtrar pacotes não confiáveis, explorando as mensagens do NDP. Nesta contribuição, os autores consultam servidores DHCPv6 para efetuar o processo de validação através de mensagens NDP emitidas através do SAVI.

A partir de uma análise dos trabalhos apresentados, foi possível elaborar uma proposta alternativa para combater o IP *Spoofing* em redes IPv6. A proposta é fundamentada principalmente nas características dos algoritmos NS_HANDLER() e NA_HANDLER() e seus submódulos propostos por [Barbhuiya et al. 2013], da integração de servidores DHCP para auxiliar o processo de validação de endereços de rede em uma rede local proposto por [Yan et al. 2011] e do algoritmo de manipulação de IP *Spoofing* proposto por [Mowla et al. 2015] que, assim como o trabalho anterior, utiliza SDN para auxiliar no processo de manipulação de pacotes.

4. Mitigação do uso da técnica de IP *Spoofing* na origem

Segundo [Moura et al. 2014], os ISPs (*Internet Service Provider*) deveriam investir na mitigação de tráfegos maliciosos originados internamente em seu *Autonomous System* (AS), garantindo que seus clientes não originem ataques externos. Porém, os ISPs se abstêm de investir na mitigação do tráfego originado por seus clientes. Com a expansão crescente do IPv6, a não necessidade de utilização da técnica de NAT e as fragilidades do próprio protocolo NDP, pode-se supor que o uso de IP *Spoofing* continuará em constante ascensão.

Através deste estudo, foi possível compreender os principais componentes que permitem que a técnica de IP *Spoofing* seja aplicada em uma rede de computadores. Analisando os trabalhos relacionados, foi possível identificar métodos eficientes para identificar e mitigar o uso da técnica de IP *Spoofing*, assim como compreender como a comunidade acadêmica está discutindo o tema. O estudo sobre SDN foi realizado para compreender e verificar como as redes definidas por *software* podem colaborar com o combate de IP *Spoofing*.

4.1. Método identificado para o combate de IP *Spoofing*

Através dos trabalhos relacionados analisados, foi possível identificar que a maioria das soluções de combate ao uso da técnica de IP *Spoofing* trata a sua prevenção diretamente no destino, ou seja, no local onde um possível ataque possa ser realizado. Como já citado, é notável que se cada rede de computadores e/ou os IPSs mitigassem o tráfego de saída de seus perímetros, validando os endereços de origem de suas redes, o problema do uso da técnica de IP *Spoofing* possivelmente poderia ser solucionado.

4.2. Componentes da solução identificada

A solução identificada é composta pelo uso de SDN que, além permitir que se tenha uma visão global de todo o tráfego da rede através de uma análise de fluxos, permite que seja desenvolvida uma aplicação com base na API do controlador. Esta aplicação é a responsável por validar os endereços MAC e endereços IP de cada pacote que está saindo de uma rede local. O método de validação do endereçamento MAC e IP é desenvolvido com base na solução proposta por [Barbhuiya et al. 2013], destacando os algoritmos VERIFY_IP-MAC() e RESPONSE_ANALYSER(). Sua arquitetura global é baseada nas soluções propostas por [Yao et al. 2011] e por [Mowla et al. 2015], descartando a utilização do protocolo SAVI. Por fim, a implementação desta solução é baseada na solução proposta por [Yan et al. 2011], mas mitigará o tráfego de saída em vez do tráfego de entrada.

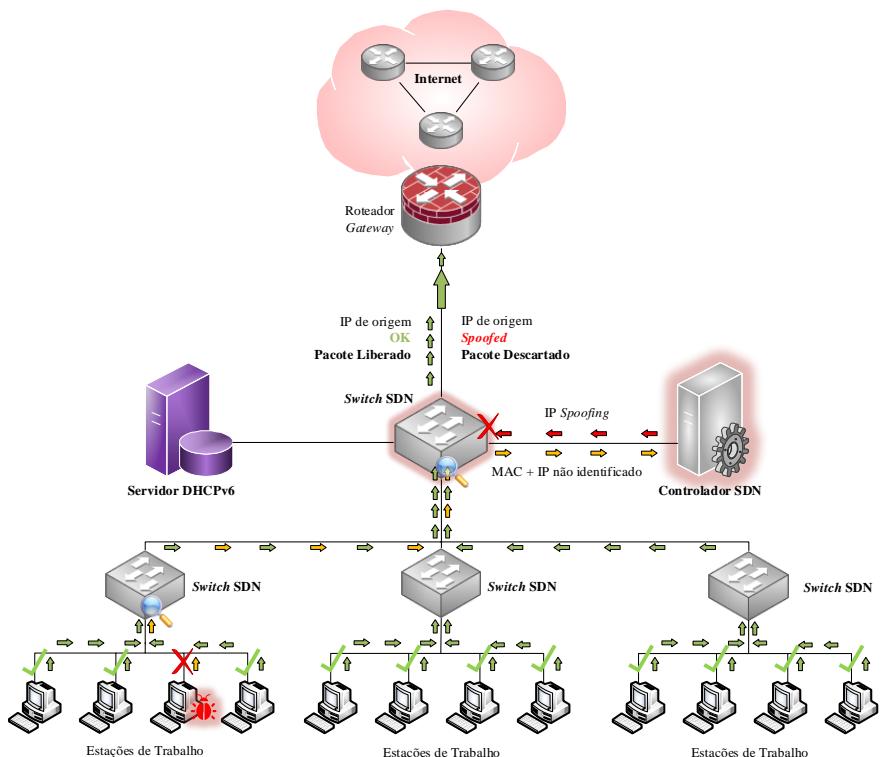


Figura 2. Infraestrutura e implementação da solução identificada.

A Figura 2 ilustra a implementação da solução em uma rede local. O switch com suporte à tecnologia SDN fica localizado entre o roteador (*gateway*) de borda e dos demais ativos da rede local. O controlador SDN encontra-se conectado neste switch para poder gerenciar o tráfego da rede. Todo o tráfego de saída será analisado pelo switch que analisará o endereço MAC e endereço IPv6 de origem de cada pacote que sair da rede. O switch compara o endereço MAC e o endereço IP através de sua tabela de fluxos. Caso estes endereços não estejam inseridos nesta tabela, o switch encaminha o pacote para o controlador SDN que, consequentemente, decide qual a ação a ser tomada sobre o respectivo pacote. O controlador, através do processo de validação de endereços, autoriza ou descarta o pacote através do switch, assim como atualiza a tabela de fluxos do equipamento.

4.3. Combatendo o IP Spoofing

A identificação do IP Spoofing é realizada através da comparação do endereço MAC e do endereço IP contido no campo “Endereço de Origem” (*Source Address*) do cabeçalho do protocolo IPv6 do respectivo pacote analisado. Ao receber o pacote, o controlador valida o endereço MAC, consultando a lista de endereços atribuídos na base de dados do servidor DHCPv6 e sua lista endereços IP atribuídos manualmente (*whitelist*). É importante

ressaltar que esta *whitelist* é uma forma alternativa que permite ao administrador inserir manualmente os endereços de rede e de MAC para liberar endereços e seus respectivos pacotes da rede.

Primeiramente a aplicação faz uma busca do endereço MAC em sua *whitelist*; caso seja encontrado e esteja atribuído o mesmo endereço IP do pacote analisado, o controlador atualiza a tabela de fluxos do *switch* e o autoriza a liberar o pacote.

Caso o endereço MAC não esteja inserido na *whitelist* ou a comparação com o endereço IP do pacote esteja errada, a aplicação consulta a base de dados do servidor DHCPv6 na tentativa de identificar a presença do endereço MAC. Caso o endereço seja localizado, ele é validado comparando o endereço IP do pacote com o endereço IP atribuído pelo servidor DHCPv6. Caso os dois endereços estejam em conformidade, o controlador atualiza a tabela de fluxos do *switch* e autoriza o repasse do pacote.

Caso o endereço MAC não esteja em conformidade com o endereço IP inserido na base do servidor DHCPv6 ou simplesmente os endereços MAC ou IP não sejam encontrados na base de dados do servidor DHCPv6, o uso da técnica de IP *Spoofing* é identificado. Com isto, o controlador descarta esse pacote, assim como registra um evento (*log*) contendo o endereço de origem e o endereço de destino do pacote, emitindo um alerta ao administrador da rede.

5. Conclusão e Trabalhos Futuros

Através do estudo realizado, foi possível compreender o funcionamento da técnica de IP *Spoofing*, entender os principais protocolos envolvidos, assim como identificar a preocupação e como a comunidade acadêmica está tratando o tema. Por fim, um novo e possível método para mitigar e combater o IP *Spoofing* foi delineado. Como trabalho futuro, este método deverá ser experimentado em uma rede real, validando sua teoria.

Referências

- Barbhuiya, F., Bansal, G., Kumar, N., Biswas, S., and Nandi, S. (2013). Detection of neighbor discovery protocol based attacks in IPv6 network. *Networking Science*, 2(4):91–113.
- Bellovin, S. M. (1989). Security Problems in the TCP/IP Protocol Suite. *SIGCOMM Comput. Commun. Rev.*, 19(2):32–48.
- Deering, S. (1991). ICMP Router Discovery Messages. RFC 1256, Internet Engineering Task Force.
- Feamster, N., Rexford, J., and Zegura, E. (2014). The road to SDN: An intellectual history of programmable networks. *SIGCOMM Comput. Commun. Rev.*, 44(2):87–98.
- Kreutz, D., Ramos, F., Esteves Verissimo, P., Esteve Rothenberg, C., Azodolmolky, S., and Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76.
- Kurose, J. F. and Ross, K. W. (2006). *Redes de Computadores e a Internet: uma abordagem top-down*. Addison Wesley, São Paulo, third edition.

- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74.
- Morris, R. T. (1985). *A Weakness in the 4.2BSD Unix TCP/IP Software*.
- Moura, G., Sadre, R., and Pras, A. (2014). Bad Neighborhoods on The Internet. *Communications Magazine, IEEE*, 52(7):132–139.
- Mowla, N., Doh, I., and Chae, K. (2015). An efficient defense mechanism for spoofed IP attack in SDN based CDN. In *Proc. of the International Conference on Information Networking (ICOIN)*, pages 92–97.
- Mukaddam, A., Elhajj, I., Kayssi, A., and Chehab, A. (2014). IP Spoofing Detection Using Modified Hop Count. In *Proc. of 28th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, pages 512–516.
- Narten, T., Nordmark, E., Simpson, W., and Soliman, H. (2007). Neighbor Discovery for IP version 6 (IPv6). RFC 4861, Internet Engineering Task Force.
- Plummer, D. (1982). Ethernet Address Resolution Protocol Or Converting Network Protocol Addresses. RFC 826, Internet Engineering Task Force.
- Postel, J. (1981). Internet Protocol. RFC 0791, Internet Engineering Task Force.
- Tanase, M. (2003). IP Spoofing: An Introduction. Disponível em: <<http://www.symantec.com/connect/articles/ip-spoofing-introduction/>>. Acesso em: abr. 2015.
- Wu, J., Bi, J., Bagnulo, M., Berker, F., and Vogt, C. (2013). Source Address Validation Improvement (SAVI) Framework. RFC 7039, Internet Engineering Task Force.
- Yan, Z., Deng, G., and Wu, J. (2011). Savi-based IPv6 source address validation implementation of the access network. In *Proc. of Computer Science and Service System (CSSS), 2011 International Conference on*, pages 2530–2533.
- Yao, G., Bi, J., and Xiao, P. (2011). Source address validation solution with openflow/nox architecture. In *Proc. of Network Protocols (ICNP), 2011 19th IEEE International Conference on*, pages 7–12.

V

Sessão 5 - Computação Ubíqua e Internet das Coisas

Avaliação de Simuladores para Redes Veiculares Ad Hoc para Implementação de Aplicações P2P de Gerenciamento

Lucas De Marchi Dreger¹, Jéferson Campos Nobre¹

¹Instituto de Informática – Universidade do Vale do Rio dos Sinos – UNISINOS
São Leopoldo – RS – Brazil

lucasdreger@gmail.com, jcnobre@unisinos.br

Abstract. Vehicular ad hoc networks (VANETs) were designed to bring benefits to the driver and passengers of a vehicle. These networks are characterized by the high degree of dynamism and constant mobility of its nodes. These characteristics imply, especially in its management, and may hinder the application of traditional management solutions. Management systems based on P2P technology (P2P-Based Network Management-P2PBNM) can be an alternative effective management. Uses tests can be performed through network simulators. However, the simulators need to be evaluated. Therefore, features and functionalities of the simulators were pointed. At the end, the results obtained from the simulation within a P2P management application were satisfied.

Resumo. Redes veiculares ad hoc (Vehicular Ad Hoc Networks - VANET) foram projetadas para trazer benefícios ao motorista e aos passageiros de um veículo. Estas redes são caracterizadas pelo alto grau de dinamismo e constante mobilidade de seus nós. Tais características implicam, principalmente, no seu gerenciamento e podem dificultar a aplicação de soluções tradicionais. Sistemas de gerenciamento baseados em tecnologia P2P (P2P-Based Network Management - P2PBNM) podem ser uma alternativa eficaz. Testes de aplicações podem ser realizados através de simuladores de rede. No entanto, convém que os simuladores sejam avaliados. Sendo assim, foram elencadas características e funcionalidades dos simuladores trabalhados. Os resultados obtidos das simulações com o uso de uma aplicação P2P de gerenciamento foram satisfatórios.

1. Introdução

Atualmente, a maior parte das pesquisas sobre Redes Veiculares Ad Hoc (Vehicular Ad Hoc Networks - VANET) é realizada com o auxílio de simuladores para este fim. Isto ocorre, entre outros motivos, porque o custo de implementação de VNs reais é considerado elevado [Alves et al. 2009]. A realização de simulações também traz vantagens, se comparada à implementação real. Além da redução do custo, podem ser citadas a comodidade na implementação e a redução do tempo investido.

VNs possuem a necessidade de um gerenciamento eficiente, tal como as redes tradicionais. No entanto, as características dessas redes (*e.g.*, alta mobilidade dos nós e topologia dinâmica) oferecem desafios para o gerenciamento efetivo das mesmas. A utilização de tecnologia Par-a-Par (*Peer-to-Peer* – P2P) no gerenciamento de redes pode ser uma alternativa viável para VNs. [Nobre et al. 2013]. Esta alternativa, no entanto, precisa ser avaliada.

Este trabalho está organizado da seguinte maneira. Na seção 2 é apresentada uma revisão teórica sobre VNs e seus diferentes tipos de simuladores. A seção 3 caracteriza as diferentes abordagens de gerenciamento de VANETs. Nas seções 4 e 5 será descrito como se deu a escolha dos simuladores, bem como as atividades práticas realizadas. Por fim, a avaliação dos resultados é apresentada na seção 6.

2. Fundamentação Teórica

Rede veicular (*Vehicular Network – VN*) é uma tecnologia emergente que visa integrar as capacidades de redes sem fio de nova geração com veículos inteligentes [Wang 2012]. Seu objetivo, portanto, é facilitar a troca de informação entre veículos ou entre veículo e infraestrutura (ponto de acesso).

VNs podem ser divididas em três tipos de arquitetura, de acordo com sua utilização: *ad hoc*, com infraestrutura e híbrida [Alves et al. 2009]. Em redes *ad hoc*, cada nó da rede é caracterizado por um veículo. Este utilizará mensagens *multicast* e *broadcast* para transmitir informações para veículos próximos [Zeadally et al. 2012]. As redes com infraestrutura contam com um terminal centralizador de mensagens, responsável pelo roteamento de informações. Por fim, redes híbridas são um meio termo entre as duas já citadas, possuindo alguns pontos com infraestrutura e alguns pontos com roteamento *ad hoc*.

A realização de testes de desempenho em VNs pode se tornar inviável por exigir um grande número de pessoas, condições climáticas favoráveis e possuir custos elevados. Como consequência, em muitos casos são utilizados simuladores de redes para a realização de testes e avaliações. Embora a utilização de simuladores traga mais facilidade e utilize menos recursos, seus resultados servem apenas como indicativos de solução, podendo divergir dos resultados reais de uma rede veicular [Alves et al. 2009].

3. Gerenciamento de VANETs

O gerenciamento de uma rede veicular passa pela sua arquitetura de comunicação. Em 2004, iniciou-se a padronização da comunicação em VNs pelo IEEE. O padrão criado ainda está em fase de desenvolvimento, e é conhecido como IEEE 802.11p WAVE. A arquitetura WAVE trabalha com uma nova pilha de protocolos de comunicação, baseada no protocolo *Wave Short Message Protocol* (WSMP). Mensagens WSMP são normalmente mais utilizadas em VANETs. Isto ocorre pelo fato de terem sido criadas especificamente para este tipo de rede, podendo trazer benefícios para esta comunicação. A arquitetura WAVE, no entanto, suporta também o envio de datagramas IP [Alves et al. 2009].

VANETs são caracterizadas pela grande mobilidade e o alto dinamismo dos nós. Dado o fato de que os veículos trafegam em alta velocidade em rodovias, é importante que eles possam se comunicar em tempo real. Caso ocorra uma situação de emergência, todos os veículos afetados poderão ser informados sobre o evento. Outro desafio encontrado em VANETs é o alto número de nós. Em áreas metropolitanas, o número de veículos que trafega em um dado horário pode chegar a ordem de milhões, o que pode significar um congestionamento na rede. O fato de grande parte das mensagens entre veículos ocorrer em forma de *broadcast* contribui para a ocorrência deste problema [Zhang 2012]. A importância destas redes, somada as suas características e particularidades, gera um aumento na busca de formas de um gerenciamento efetivo.

3.1. Gerenciamento de VANETs através de sistemas de gerenciamento P2P

O gerenciamento de rede baseado em P2P (*P2P-Based Network Management – P2PBNM*) surgiu a fim de integrar modelos de gerenciamento de redes tradicionais com os novos serviços introduzidos pelas redes P2P. Sistemas P2PBNM possuem um alto grau de descentralização em tarefas de gerenciamento. Desta forma, os próprios pares de gerenciamento são responsáveis por proverem os recursos necessários para tais tarefas. Também devido a esta descentralização, o uso de informações locais para a tomada de decisões aumenta, o que promove a autonomia dos pares de gerenciamento [Nobre et al. 2013]. Outra característica presente nos pares de gerenciamento é a de possuírem um papel duplo, operando não somente como pares de gerenciamento, mas também participando da comunicação entre os demais *peers* [Granville et al. 2005].

As características de gerenciamento encontradas em redes P2P podem ser muito úteis se aplicadas em redes onde a mobilidade dos nós é alta e constante, como é o caso de VANETs. A utilização de sistemas P2PBNM pode ser uma alternativa viável para seu gerenciamento efetivo. Este tipo de gerenciamento pode ser testado utilizando-se simuladores VANETs juntamente com aplicações P2P, a fim de se obter um ambiente funcional e de baixo custo.

4. Escolha dos simuladores VANET

O presente trabalho contou com a utilização de dois simuladores: O NS-3 e o CORE. A seguir serão citadas as principais características dos simuladores trabalhados.

4.1. NS-3

O Network Simulator 3 (NS-3¹) pode ser instalado em sistemas UNIX em geral e foi escrito em linguagem C++ e Python. O simulador não possui suporte nativo a sistemas Windows, porém pode ser instalado através do *cygwin*². O NS-3 não possui interface gráfica para criação de simulações. No entanto, após a criação de *scripts*, pode-se fazer uso do *NetAnim*³ para melhor visualizar as simulações criadas.

Alguns simuladores habilitam os nós da simulação a executar diferentes tipos de sistemas. Isto é feito para o caso de testes de aplicações entre os nós, e normalmente é realizado através de emulações de outros sistemas Linux. No NS-3, podem ser utilizadas algumas abordagens para este fim. Dois exemplos são o *Linux Container* (LXC) e *Direct Code Execution* (DCE).

LXC é utilizado para a criação de túneis de comunicação entre dois ou mais *hosts*. Portanto, o LXC pode ser útil se utilizado em simulações do NS-3, de forma que cada nó simulado represente um *container* diferente na ferramenta. O DCE é um módulo para o NS-3 que provê a implementação de protocolos de rede e aplicações, sem que seja necessário alterar seu código fonte [Camara et al. 2014].

4.2. CORE

O *Common Open Research Emulator* (CORE⁴) [Ahrenholz 2010] é também conhecido por ser um emulador. Ou seja, ele é capaz de realizar uma representação de redes de

¹<https://www.nsnam.org/>

²<http://www.cygwin.com/>

³<http://www.nsnam.org/wiki/NetAnim>

⁴<http://www.nrl.navy.mil/itd/ncs/products/core>

computadores como se cada computador fosse uma instância real. A virtualização do *kernel*, ou implementação LXC, possibilita que cada nó da rede seja uma instância virtual Linux, de forma que cada instância possua uma pilha de protocolos de rede independente (*i.e.*, diferente da máquina original).

O CORE não possui suporte a simulações específicas sobre VANET. Desta forma, foram utilizadas redes sem fio, adicionando-se padrões de mobilidade entre os nós. Para a simulação de redes sem fio é recomendado o uso do *Extendable Mobile Ad-hoc Network Emulator* (EMANE). O simulador também não possui modelos nativos de mobilidade dos nós, porém é possível importar padrões externos. Em função disso, utilizou-se a ferramenta BonnMotion⁵, capaz de implementar diversos padrões de mobilidade.

É possível fazer com que as topologias simuladas interajam com um ambiente real. Para tanto, é necessária a instalação de uma *Application Programming Interface* (API), que pode ser obtida da página principal do CORE. No caso de VANETs, isto pode ser útil à medida em que experimentos reais começam a ser criados, de forma a utilizar ambientes reais e virtuais para a criação de um experimento com grandes topologias. Esta função pode ainda ser utilizada a fim de se obterem recursos computacionais adicionais (*e.g.*, *clusters*).

5. Simulação realizada

Foram realizadas simulações em ambientes físicos e virtuais. Em ambos os casos, utilizou-se o sistema Ubuntu 12.04 de 64 bits.

A implementação original do simulador NS-3 possui exemplos de suporte a *Linux Containers* (LXC). Portanto, torna-se uma alternativa o uso de códigos funcionais de outros *scripts*, modificando-os para que se enquadrem na simulação em questão. Como resultado, obteve-se um *script* de dois nós em uma rede sem fio com um padrão de mobilidade nativo do NS-3. A medida em que a simulação cresce, torna-se necessária a criação de mais nós com suporte a LXC no *script* de simulação.

A opção pela utilização de poucos nós na simulação se deu por dois motivos: primeiramente para que a topologia seja melhor compreendida; o segundo motivo está relacionado à melhor utilização dos recursos computacionais envolvidos. Cada nova implementação LXC significa uma nova camada de virtualização no sistema Linux. Desta forma, é necessário cuidado com o número de implementações simultâneas. Para que a simulação aconteça, o ambiente Linux deverá poder suportar duas novas virtualizações.

A criação de uma topologia no CORE é feita rapidamente através de uma interface gráfica. Foram criadas simulações com dois e três nós. Dessa forma, gerou-se dois arquivos de mobilidade randômica no BonnMotion. Os roteadores foram configurados como unidade sem fio, a fim de melhor representarem veículos em movimento.

Por fim, ambos os simuladores foram configurados com padrões de mobilidade e funcionalidade LXC em seus nós. Neste momento, o ManP2P-ng⁶[Duarte et al. 2011], um sistema de monitoramento P2P, foi executado nos nós da rede, criando um *overlay* P2P. Uma vez que o módulo de monitoramento foi iniciado, observou-se que os nós criados iniciaram a troca de mensagens entre si. Não ocorreram incompatibilidades na

⁵<http://www.bonnmotion.net>

⁶<https://github.com/ComputerNetworks-UFRGS/ManP2P-ng>

execução do ManP2P-ng. De forma análoga, é possível que o monitoramento P2P obtenha resultados positivos, se implementado em VANETs reais. A Figura 1 ilustra o início da comunicação entre dois nós, chamados *n1* e *n2*, no *overlay* criado no CORE.

```

root@n1:/tmp/pycore.37607/n1.conf/bld
Reading file: 01-peerexchange.py
Reading file: 10-groupsExtension.py
Reading file: 20-httplib.py
Reading file: 30-sshRPC.py
Cannot load module 30-sshRPC.py
No module named conch
Reading file: 40-dtnHealingDataModel.py
Reading file: 40-dtnMonitoringDataModel.py
Reading file: 50-healService.py
Reading file: 50-monitor-service.py

++ Extending Protocol ++
BasicOverlay extension loaded
PeerExchange extension loaded
Groups extension loaded

... Data received ...
c:>n2:8001
Parsed data:
['c', 'n', 'n2', '8001']
Adding peer nicknamed n2

++ Data received ++
peerlist;
Parsed data:
('peerlist', 'r')

++ Data received ++
group:j:group
Parsed data:
['group', 'j', 'group']
Peer n2 requested to join group group

root@n2:/tmp/pycore.37607/n2.conf/bld
Reading file: 01-peerexchange.py
Reading file: 10-groupsExtension.py
Reading file: 20-httplib.py
Reading file: 30-sshRPC.py
Cannot load module 30-sshRPC.py
No module named conch
Reading file: 40-dtnHealingDataModel.py
Reading file: 40-dtnMonitoringDataModel.py
Reading file: 50-healService.py
Reading file: 50-monitor-service.py

++ Extending Protocol ++
BasicOverlay extension loaded
PeerExchange extension loaded
Groups extension loaded

... Data received ...
c:>n1:8001
Parsed data:
['c', 'a', 'n1', '8001']
Adding peer nicknamed n1

++ Data received ++
peerlist;
Parsed data:
('peerlist', 'r')

++ Data received ++
group:j:group
Parsed data:
['group', 'j', 'group']
Peer n1 requested to join group group

```

Figura 1. *overlay* criado pelo ManP2P em nós do CORE

6. Avaliação

Inicialmente, foram avaliadas questões relacionadas à instalação das ferramentas. Não existe um arquivo de construção (*e.g., script*) que realize a instalação completa das aplicações. No entanto, em ambos os casos, a documentação de instalação foi clara e precisa.

A próxima característica avaliada foram as implementações LXC apresentadas por ambos os simuladores. Os túneis LXC são uma importante opção para a execução de códigos diretamente nos nós. O CORE apresenta implementações nativas dos túneis, enquanto o NS-3 necessita uma série de configurações adicionais para poder utilizá-los. O NS-3, no entanto, possui outra alternativa para esta função, chamada *Direct Code Execution* (DCE).

Recomenda-se a utilização de sistemas Linux para a instalação dos simuladores. O sistema Mac OSX também suporta a instalação deste simulador. Simulações realizadas em ambientes Windows devem ser feitas através da ferramenta *Cygwin*⁷ ou similar. Neste trabalho foi utilizado o sistema operacional Ubuntu Linux 12.04 de 64 bits, sendo que, em nenhum momento foram encontradas incompatibilidades com as aplicações avaliadas.

O estudo anterior mostrou que os simuladores poderiam ser utilizados de forma conjunta com redes reais. Uma vez que foram trabalhados com o uso de máquinas virtuais, verificou-se também se poderiam trabalhar com equipamentos (*i.e., hosts*) reais. Como resultado, constatou-se que ambos os simuladores suportam este tipo de implementação. O CORE possui uma ferramenta própria para conexão com dispositivos externos. O NS-3 pode fazer esta conexão através das implementações LXC.

⁷<https://www.cygwin.com>

Foram utilizados protocolos de comunicação tradicionais nas duas simulações. Isto ocorreu pois os protocolos de comunicação VANET referenciados anteriormente (*e.g.*, WAVE, WSMP) ainda não foram implementados nos simuladores trabalhados. Sendo assim, utilizaram-se as abstrações já implementadas para o roteamento no NS-3. Para o roteamento entre roteadores no CORE, utilizou-se o protocolo OSPFv3.

A próxima característica avaliada refere-se aos arquivos *trace*. A utilização destes arquivos provê um padrão de mobilidade aos nós, simulando a movimentação de veículos. O NS-3 implementa nativamente uma série de padrões de mobilidade. Desta forma, utilizou-se o padrão *RandomWalk2dMobility*. O CORE, porém, não realiza a implementação de padrões de mobilidade nativos. Ambos os simuladores, no entanto, são capazes de importar arquivos *trace* de ferramentas auxiliares. Arquivos *trace* podem ser obtidos tanto de simuladores de tráfego, quanto de ferramentas de geração de modelos de mobilidade. A simulação do CORE foi realizada com padrões de mobilidades gerados pela ferramenta BonnMotion.

O NS-3 possui vantagem sobre o CORE, no que diz respeito a quantidade de nós suportados na simulação. O fato de não possuir uma interface gráfica pode ajudar neste sentido, aumentando o desempenho da simulação e fazendo com que mais nós sejam suportados na mesma simulação. Uma pesquisa diz que o NS-3 pode suportar algo em torno de 1000 nós na simulação [Stanica et al. 2011]. O número de nós suportados pelo CORE, de acordo com a sua documentação oficial, pode variar em função de diversos outros fatores (*e.g.*, *hardware* e sistema operacional utilizados, número de processos ativos).

Na Tabela 1, é possível identificar algumas das características avaliadas em cada simulador, bem como os resultados obtidos.

Tabela 1. Comparação de funcionalidades apresentadas pelos simuladores

	NS-3	CORE
Suporte a implementações LXC	x	x
Implementações LXC nativas		x
Suporte a <i>traces</i> externos	x	x
Suporite completo a protocolos de comunicação VANET (<i>e.g.</i> , WAVE)		
Integração a <i>hosts</i> reais	x	x
Integração a redes reais	x	x
Suporite a Sistema Operacional Linux	x	x
Suporite a Sistema Operacional Windows		
Suporite a Sistema Operacional Mac OS X	x	
Suporite a Sistema Operacional FreeBSD	x	x
Interface gráfica		x
Não possui custo para implementação	x	x
Código aberto	x	x
Implementa uma forma de execução de comandos alternativa ao LXC	x	

A aplicação P2P pôde ser implementada em ambos os simuladores sem qualquer tipo de incompatibilidade. Entretanto a utilização do CORE trouxe mais vantagens à simulação. Sua interface gráfica ajudou na solução de problemas e maximizou o tempo de criação de novas simulações. A implementação própria da ferramenta LXC também

foi útil na simulação. O NS-3 não possui esta característica nativamente, portanto foi necessária a criação e configurações de túneis manuais. *Scripts* de configuração LXC foram criados a fim de agilizar este processo, e podem ser visualizados no Anexo 2. O tempo de criação de novas simulações no NS-3 foi mais alto que o do CORE. Isto se deu pois os *scripts* de criação de novas simulações devem ser programados com linguagem C++.

O NS-3, por sua vez, também possui outras vantagens, se comparado ao CORE. Primeiramente, trata-se de uma das maiores ferramentas de simulação da atualidade. O NS-3 possui código aberto e uma documentação completa e organizada. Isto facilita e faz com que seja possível a atualização permanente, com o envio de novos projetos, criados por desenvolvedores. Os projetos podem ser dos mais diversos tipos, podendo variar desde a implementação de um novo protocolo de roteamento, até uma simples correção de falha.

O suporte encontrado pelo NS-3 também é superior ao do CORE. Talvez isto aconteça dada a dificuldade inicial apresentada pelo NS-3 aos novos usuários. Porém, este apresenta uma série de manuais, tutoriais, grupos de discussão e canais de comunicação. O CORE, por sua vez, possui um manual de instruções e um canal de comunicação, via e-mail.

A figura 2 foi criada para melhor compreender as avaliações realizadas. Diferentemente da tabela 1, cujo objetivo é apresentar funções implementadas nos simuladores, a figura 2 elenca qual simulador melhor implementa as características citadas, atribuindo notas de 0 a 5. A avaliação foi realizada com base na construções de simulações e implementação do ManP2P-ng na simulação.

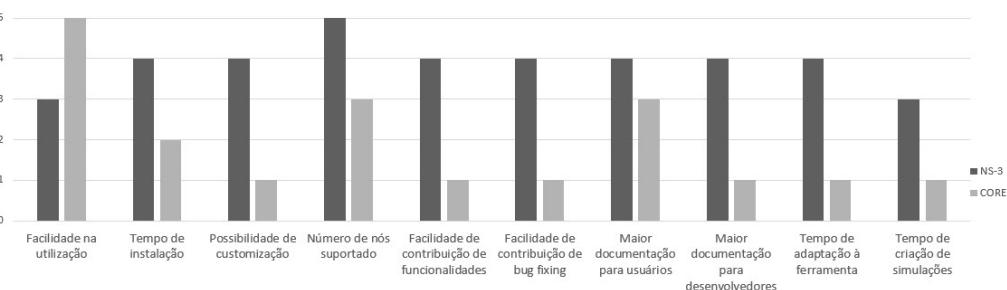


Figura 2. Avaliação de funcionalidades apresentadas por cada simulador

7. Conclusões

O presente trabalho abordou a realização de um estudo com base em simulações e avaliações dos resultados encontrados em simuladores de redes veiculares *ad hoc*. Este trabalho avaliou os simuladores quanto as suas características, considerando os benefícios que as aplicações P2PBNM podem trazer ao gerenciamento de VANETs. Pode-se dizer que, em ambos os casos, os resultados obtidos dos simuladores foram satisfatórios. A aplicação P2P não apresentou incompatibilidades em nenhum dos ambientes trabalhados. Da mesma forma, não houve extrapolação de dados com nenhum dos simuladores. De

forma análoga, portanto, é possível que o monitoramento P2P obtenha resultados positivos, se implementado em VANETs reais. Por outro lado, existem algumas características específicas de VANETs que ainda não estão disponíveis para implementação nos simuladores trabalhados. À medida que protocolos de roteamento utilizados em VANETs começarem a ser modelados também em simuladores, isto deverá trazer benefícios para as simulações.

Embora os resultados obtidos dos simuladores fossem positivos, constatou-se algumas diferenças na sua utilização. Algumas funções podem ser melhor realizadas pelo NS-3 ou pelo CORE. Para melhor visualizar dos resultados obtidos, criou-se uma tabela e um gráfico de comparação na seção 6.

Referências

- Ahrenholz, J. (2010). Comparison of core network emulation platforms. In *MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010*, pages 166–171.
- Alves, R., Campbell, I., Couto, R., Campista, M., Moraes, I., Rubinstein, M., Costa, L., Duarte, O., and Abdalla, M. (2009). Redes veiculares: princípios, aplicações e desafios. In *Minicurso do Simpósio Brasileiro de Redes de Computadores*, pages 199–254. SBRC.
- Camara, D., Tazaki, H., Mancini, E., Turletti, T., Dabbous, W., and Lacage, M. (2014). Dce: Test the real code of your protocols and applications over simulated networks. *Communications Magazine, IEEE*, 52(3):104–110.
- Duarte, P., Nobre, J., Granville, L., and Rockenbach Tarouco, L. (2011). A p2p-based self-healing service for network maintenance. In *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*, pages 313–320.
- Granville, L., da Rosa, D., Panisson, A., Melchior, C., Almeida, M., and Rockenbach Tarouco, L. (2005). Managing computer networks using peer-to-peer technologies. *Communications Magazine, IEEE*, 43(10):62–68.
- Nobre, J., Bertinatto, F., Duarte, P., Granville, L., and Tarouco, L. (2013). Gerenciamento oportunístico em redes tolerantes a atrasos/desconexões através da utilização de tecnologia par-a-par na previsão de encontros entre nós. In *XVIII Workshop de Gerência e Operação de Redes e Serviços*. SBRC.
- Stanica, R., Chaput, E., and Beylot, A.-L. (2011). Simulation of vehicular ad-hoc networks: Challenges, review of tools and recommendations. volume 55, pages 3179–3188. Elsevier North-Holland, Inc., New York, NY, USA.
- Wang, Y. (2012). Review of vehicular networks, from theory to practice. volume 43, pages 25–29. ACM, New York, NY, USA.
- Zeadally, S., Hunt, R., Chen, Y.-S., Irwin, A., and Hassan, A. (2012). Vehicular ad hoc networks (vanets): status, results, and challenges. *Telecommunication Systems*, 50(4):217–241.
- Zhang, J. (2012). Trust management for vanets: Challenges, desired properties and future directions. *Int. J. Distrib. Syst. Technol.*, 3(1):48–62.

Uma abordagem híbrida para armazenamento de dados de contexto no EXEHDA

**Diógenes Yuri Leal da Rosa¹, Ivan José Rambo¹, Roger da Silva Machado¹,
Ricardo Borges Almeida¹, Henrique de Vasconcellos Rippel¹,
Adenauer Corrêa Yamin¹, Ana Marilza Pernas¹**

¹Universidade Federal de Pelotas (UFPEL)
Pelotas – RS – Brasil

{diogenes, ijr, rds, rdsmachado, rbalmeida, adenauer, marilza}@inf.ufpel.edu.br
hvrippel@gmail.com

Abstract. This article presents a hybrid approach to storage contextual information present in EXEHDA middleware. The proposal consists of a layer of abstraction that coordinates access to context data and a hybrid repository of contextual data that uses various models of databases. The contribution was evaluated in order to demonstrate the benefits of its use, where it is emphasized that the results showed a performance gain considering the insertion time and also an improvement in disk storage space used.

Resumo. Este artigo apresenta uma abordagem híbrida para o armazenamento de informações contextuais presentes no middleware EXEHDA. A proposta consiste em uma camada de abstração que coordena o acesso aos dados de contexto e em um repositório híbrido de dados contextuais que conta com distintos modelos de bancos de dados. A contribuição foi avaliada de forma a demonstrar os benefícios de sua utilização, onde destaca-se que os resultados obtidos apresentaram um ganho de desempenho considerando o tempo de inserção e também uma melhora no espaço de armazenamento em disco utilizado.

1. Introdução

Os avanços de diversas tecnologias que permeiam redes de computadores, especialmente aqueles obtidos pelas pesquisas envolvendo Computação Ubíqua (ubicomp), propiciam serviços, aplicações e informações aos usuários a qualquer hora e em qualquer lugar. Estes recursos distribuídos, por sua vez, acabam gerando volumes cada vez maiores de dados, de diferentes tipos e formatos, que precisam ser avaliados e tratados habilmente para garantir a natureza volátil, espontânea, heterogênea e distribuída das comunicações que são peculiares à Ubicomp [Langheinrich 2010].

Dessa forma, ao ambicionar as características computacionais da ubiquidade, toda a aplicação precisa estar preparada para atuar com: (a) um volume massivo de dados; (b) diferentes formatos de eventos e (c) eficácia no que diz respeito à velocidade no tratamento das informações contextuais. Para atender estas demandas, os conceitos de *Big Data* se mostram oportunos devido ao seu foco na utilização de funcionalidades analíticas capazes de lidar com variedades de formatos, velocidade e a volatilidade dos dados [Y Demchenko 2014].

Os bancos de dados não-relacionais vêm assumindo um papel de destaque no âmbito de *Big Data* justamente pelo seu desempenho no tratamento de grandes conjuntos de dados de formatos variados. Percebe-se assim o alinhamento entre os requisitos da Ubicomp e as potencialidades dos bancos de dados não-relacionais. [Sadlage and Fowler 2013]

O presente trabalho consiste em uma contribuição para com o *middleware* EXEHDA (*Execution Environment for Highly Distributed Applications*). Este middleware, proposto em [Yamin 2004] [Lopes et al. 2014], tem como objetivo definir a arquitetura para um ambiente de execução destinado às aplicações da UbiComp.

A proposta traz a concepção de uma abordagem híbrida de armazenamento para dados contextuais a qual foi incorporada no Servidor de Contexto do EXEHDA objetivando otimização no processo de interação com o repositório de dados. A contribuição consiste na concepção de um Repositório Híbrido de Dados Contextuais (RHDC) e de um Gerenciador de Dados Contextuais (GDC) atribuindo ao EXEHDA a possibilidade de um cenário de armazenamento misto.

Este artigo está organizado da seguinte forma: na Seção 2 são descritos os trabalhos relacionados; a Seção 3 introduz os conceitos referentes à *Big Data*, mais especificamente descreve as características dos bancos não-relacionais, juntamente com o *middleware* EXEHDA; na Seção 4 é discutida a concepção da abordagem proposta; a Seção 5 apresenta a avaliação do trabalho desenvolvido; e na Seção 6, são apresentadas as considerações finais.

2. Trabalhos Relacionados

[Carvalho 2014] visa promover a coexistência dos bancos de dados relacional e não-relacional oferecendo uma solução com base em uma abordagem híbrida. O trabalho destaca os desafios e tendências para o desenvolvimento de soluções de armazenamento híbridas.

Em [Marwa 2014] são realizados testes para avaliar o desempenho de três tecnologias de armazenamento de dados para detecção de APT (*Advanced Persistent Threats*): PostgreSQL; MongoDB; e Elasticsearch. A conclusão obtida foi que o MongoDB mostrou melhor desempenho ao monitorar grande volume de dados, e oferece diversos conceitos que podem otimizar ainda mais o processamento.

Em [Filho 2015] é feita uma análise comparativa de desempenho entre a abordagem de armazenamento relacional utilizando o PostgreSQL e não-relacional utilizando o MongoDB. Para analisar o comportamento dos testes e medir o desempenho, o autor utiliza a ferramenta JMeter. Para os testes foram utilizadas as quantidades de 2000, 4000 e 8000 usuários simultâneos realizando uma requisição na base de dados. Nessas três situações, apesar de ocupar maior espaço em disco, o MongoDB obteve um desempenho superior.

Analizando os trabalhos relacionados, pode-se notar que os bancos de dados não-relacionais têm proporcionado bons avanços aos sistemas. Outro fato interessante a ser observado é o trabalho de [Carvalho 2014], o qual propõe a coexistência de dois modelos e destaca que essa é a tendência para o desenvolvimento de novas soluções de armazenamento, de forma a se aproveitar dos pontos positivos de cada modelo.

Os resultados desses trabalhos motivaram a elaboração da proposta aqui estabelecida, contemplando o início dos esforços para atendimento das demandas de *Big Data* no middleware EXEHDA.

3. Referencial Teórico

Esta seção introduz a base conceitual associada à concepção da abordagem híbrida de armazenamento, sendo que estes conceitos também foram considerados para avaliação e testes da mesma. Na Sessão 3.1 será abordada a relação existente entre *Big Data* e bancos de dados não-relacionais bem como suas principais bases teóricas. Já na Sessão 3.2 serão tratados aspectos importantes sobre o middleware EXEHDA.

3.1. Big Data: bancos de dados não-relacionais

Big Data é o termo que engloba uma série de conceitos e tendências referentes ao grande volume de dados existentes hoje no contexto computacional, bem como a forma que interagimos com esses dados. Seu uso ocorre nas mais diversas áreas de negócio muitas vezes para proporcionar significado e estratégias por meio de informações coletadas dos mais diversos dispositivos e usuários. É possível afirmar que big data baseia-se, principalmente, nos aspectos Volume, Velocidade, Variedade (3 V's) [Y Demchenko 2014].

Impulsionada pelas demandas de *Big Data*, a utilização de bancos de dados não-relacionais vem ganhando espaço gradativamente. NoSQL, traduzido pela comunidade como 'Not only SQL', refere-se a um grupo cada vez mais familiar de sistemas de bancos de dados não-relacionais, nos quais a base de dados não é constituída de tabelas/esquemas e geralmente não são utilizadas funções em SQL para manipulação de dados. Estas soluções são utilizadas em aplicações que trabalham com enormes quantidades de dados e, também, quando não é possível representar a natureza dos dados no modelo relacional de banco de dados [Moniruzzaman and Hossain 2013]. NoSQL destaca-se também por apresentar capacidade de distribuição da solução de banco de dados como um todo, e trabalhar de forma eficiente em cluster's [Sadlage and Fowler 2013], [Han et al. 2011].

Pode-se identificar que NoSQL possui quatro diferentes categorias em seu ecossistema: chave-valor, documento, famílias de colunas e grafos. Os principais pontos a serem destacados em cada uma das categorias são:

- Chave-valor: apresenta a arquitetura mais simples em NoSQL que é composta apenas por uma chave seguida por um valor. O valor aceita qualquer tipo de dado ou objeto e não possibilita pesquisa através de sua estrutura. A recuperação de dados neste modelo pode ser feita apenas pela chave.
- Documentos: armazena estruturas de dados independentes na forma de árvores hierárquicas e autodescritivas, constituídas de mapas, coleções e valores escalares. Neste modelo são admitidas pesquisas realizadas a partir da estrutura do documento. Outro aspecto importante, consiste no fato de que os campos vazios são ignorados, o que otimiza o espaço em disco.
- Famílias de colunas: permite que o armazenamento de dados ocorra com chaves mapeadas para valores, e os valores agrupados em múltiplas famílias de colunas, cada uma dessas famílias de colunas funcionando como um mapa de dados. É um modelo interessante quando os grupos de informações são acessados de maneira conjunta.

- Grafos: trata o conjunto de dados como uma densa estrutura de redes, onde os nodos são conectados entre si estabelecendo relações.

Os três fatores apontados pelo trabalho [Couchbase 2012] como os principais problemas de bancos de dados relacionais são a inflexibilidade/rigidez dos esquemas, seguido por baixa escalabilidade e alta latência/baixo desempenho. Estes indicativos demonstram a relevância no estudo de novas alternativas de bancos de dados. Por outro lado, a presença de relacionamentos para demandas específicas traz ao sistema a facilidade do tradicional uso de consultas SQL, desonerando a aplicação da implementação lógica do tratamento de dados. Sendo assim, a adoção de estratégias híbridas tem buscado a união dos benefícios de ambas alternativas.

3.2. EXEHDA

O EXEHDA é um *middleware* adaptativo ao contexto e baseado em serviços que visa criar e gerenciar um ambiente ubíquo. Sua arquitetura é distribuída e oferece suporte à aquisição, processamento e armazenamento de informações contextuais [Lopes et al. 2014].

Os recursos da infraestrutura física que formam o ambiente ubíquo são mapeados para três abstrações básicas [Yamin 2004]:

- EXEHDAcels: indica a área de atuação de uma EXEHDAbase, sendo composta por esta e por EXEHDAanodos;
- EXEHDAbase: é o ponto de convergência para os EXEHDAanodos, sendo responsável por todos os serviços básicos do ambiente ubíquo;
- EXEHDAano: são os dispositivos de processamento disponíveis no ambiente ubíquo, sendo responsáveis pela execução das aplicações. Um subcaso deste tipo de recurso é o EXEHDAano móvel. São os nodos do sistema com elevada portabilidade, tipicamente dotados de interface de rede para operação sem fio.

Dentro de cada célula podem existir inúmeros SB's (Servidores de Borda) que são responsáveis pela comunicação com o ambiente por meio de sensores e atuadores. Cada célula possui um EXEHDAbase no qual executa o SC (Servidor de Contexto), sendo este servidor responsável por armazenar as informações coletadas no RIC (Repositório de Informações Contextuais), bem como permitir a manipulação (processamento, visualização, etc.) destas informações.

4. Modelo Proposto

A concepção da proposta híbrida de armazenamento teve por motivação as demandas de interação com os dados existentes no *middleware* EXEHDA, no intuito de contemplar de maneira mais efetiva os 3V's de *Big Data*. Ao unir os benefícios de modelos distintos de armazenamento o *middleware* consegue um cenário mais adaptado aos dados em questão.

A proposta desenvolvida nesse trabalho partiu da implementação do modelo não-relacional de documentos em conjunto com o banco relacional anteriormente empregado pelo repositório de informações. As adaptações necessárias foram realizadas de forma a propiciar o funcionamento das duas abordagens disponibilizadas pelo repositório de dados contextuais em paralelo. A contribuição foi estabelecida no SC do EXEHDA.

Na abordagem proposta fica a cargo da aplicação determinar onde prefere armazenar seus dados contextuais, sendo indicada a utilização do banco relacional para dados que possuem relação entre eles aproveitando-se das características do modelo. E recomenda-se o armazenamento dos dados contextuais no modelo não-relacional, quando é necessário o tratamento de grande volume de dados e/ou ainda quando os dados armazenados possuem variedade de formatos, o que resultaria em colunas vazias no modelo relacional.

A Figura 1 apresenta a proposta híbrida de armazenamento oferecido, onde o GDC é responsável por disponibilizar métodos de inserção e consultas para acesso aos dados presentes nos dois modelos de armazenamento, representando uma abstração para acesso ao RHDC. Destaca-se que as aplicações não precisam se envolver com a interoperabilidade entre as formas de armazenamento que estão sendo utilizadas no processamento dos vários contextos de seu interesse.

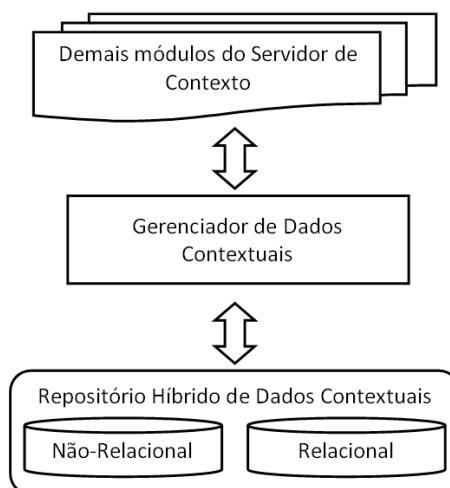


Figura 1. Gerenciador de Dados de Contexto

5. Estudo de Caso

As avaliações iniciais da proposta híbrida apresentada nesta sessão partiram dos trabalhos [Machado 2013] e [Almeida 2013] que foram baseados no EXEHDA. [Machado 2013] foi estruturado de forma a criar um analisador de registros de log. Já o trabalho [Almeida 2013] teve como foco o emprego da consciência de situação em soluções de SIEM (*Security Information and Event Management*), utilizando uma solução para processamento de eventos complexos.

Estes projetos caracterizam-se por possuírem em comum, a necessidade de um bom desempenho para o tratamento de um elevado volume de eventos que possuem diversidade nos formatos. Estes eventos são provenientes dos logs gerados pelo sistema operacional e por aplicações que operam em ambientes ubíquos.

Para implementação da abordagem foram utilizados o banco de dados PostgreSQL para o modelo relacional, e o MongoDB para o modelo não-relacional. Este último

consiste de um modelo não-relacional orientado a documentos, o qual admite pesquisas por intermédio de sua estrutura básica, sendo recomendado para o tratamento de logs [Sadlage and Fowler 2013]. Desta forma, os eventos registrados em logs passaram a ser armazenados no MongoDB, enquanto as configurações dos dispositivos monitorados e situações identificadas foram mantidas no PostgreSQL. A consistência entre as situações e os eventos que o representam fica a cargo do GDC que faz uso de funções python.

Como parte dos esforços de avaliação da abordagem híbrida proposta foram desenvolvidos dois cenários em [Rambo 2015] com o intuito de verificar o ganho na utilização de um modelo não-relacional comparado com o modelo relacional, sendo estes baseados respectivamente nos itens de coleta denominados access.log e kern.log.

O access.log é um log do servidor apache, que registra todas as solicitações processadas pelo mesmo. Já o kern.log fornece um registro detalhado das mensagens do kernel do Linux, que pode ser útil, por exemplo, para encontrar problemas no sistema operacional e analisar mensagens do *firewall*.

A escolha por estes logs é justificada por estes possuírem características distintas em relação ao formato de seus eventos e ainda por serem utilizados para identificação de situações normalmente encontradas em uma infraestrutura de redes de computadores [Swift 2010], sendo sua análise oportuna em ambientes ubíquos como os gerenciados pelo *middleware* EXEHDA.

Para realização dos testes com a abordagem híbrida desenvolvida, foram simulados localmente os ambientes necessários para a avaliação. A máquina responsável pelas simulações possui um processador Intel Core i5 com 2.27GHz de frequência, 4GB de memória e disco rígido de 500GB.

Para cada cenário foram coletados cinco diferentes quantidades de registros de logs (10000, 20000, 40000, 80000 e 100000) e para cada quantidade foi feita a análise de tempo de inserção e espaço em disco utilizado pelos modelos relacional e não-relacional. Para cada valor representado nas situações descritas nos cenários, a execução foi repetida quatro vezes e foi realizada a média dos valores coletados. Importante destacar que o desvio padrão máximo, considerando as diferentes medições dos testes de inserção foi de 0,02 segundos.

A Tabela 1 apresenta o tempo de inserção em ambas estratégias de armazenamento com os logs kern e access, onde o tempo é representado no formato de (horas:minutos:segundos).

Tabela 1. Tempos de inserção.

		Número de Eventos				
		1000	2000	4000	8000	10000
access.log	MongoDB	00:00:45	00:01:08	00:02:17	00:04:34	00:05:57
	PostgreSQL	00:01:41	00:03:16	00:06:41	00:13:20	00:16:55
kern.log	MongoDB	00:00:36	00:01:04	00:02:08	00:04:12	00:05:50
	PostgreSQL	00:01:40	00:03:17	00:06:54	00:13:16	00:16:31

Analizando a Tabela 1 fica evidente a superioridade em relação ao processamento realizado pelo modelo não-relacional, comprovando as características de desempenho do modelo. Dentre as características que proporcionam o ganho de desempenho, é possível citar a geração do identificador único (id) para cada documento de uma coleção, onde o id é gerado por um algoritmo que utiliza 12-bytes, fazendo com que os registros sejam inseridos simultaneamente. Já no modelo relacional, é necessário que os registros sejam salvos um após o outro.

A Tabela 2 apresenta os valores de espaço em disco para armazenar os diferentes valores de logs coletados. Pode-se notar que em ambos logs a estratégia não-relacional ocupa uma menor quantidade de espaço em disco, mostrando-se apta a ser utilizada para o tratamento do grande volume de dados gerado pelo monitoramento de logs.

Tabela 2. Espaço ocupado em disco.

		Número de Eventos				
		1000	2000	4000	8000	10000
access.log	MongoDB	3,84MB	7,7MB	15,39MB	30,77MB	38,5MB
	PostgreSQL	7,53MB	15MB	30MB	60MB	75MB
kern.log	MongoDB	1,78MB	3,5MB	6,96MB	14MB	17,2MB
	PostgreSQL	2,13MB	4,31MB	9,42MB	17,24MB	21MB

Analizando a Tabela 2 destaca-se que a diferença entre as estratégias de armazenamento deve-se ao formato dos eventos, onde os do access.log variam consideravelmente, o que gerava colunas em branco no modelo relacional. Em relação ao kern.log observa-se que não foi obtida tanta diferença, mas isso deve-se ao fato dos eventos deste log mantêm um padrão e assim não gerarem colunas em branco no modelo relacional.

6. Considerações Finais

Este trabalho apresentou a concepção de uma abordagem híbrida de armazenamento de informações contextuais, a qual foi empregada no *middleware* EXEHDA. A contribuição estabeleceu-se pela concepção do RHDC e pelo GDC. Desta forma, foi possível contribuir para o EXEHDA, fornecendo os benefícios das duas abordagens de banco de dados (relacional e não-relacional), em conjunto com uma camada para abstração na comunicação com os bancos.

A avaliação da proposta já pode quantificar as melhorias resultantes da adesão de um novo modelo de banco de dados por parte do *middleware* EXEHDA, obtendo resultados positivos.

Destaca-se que este é um trabalho inicial nos esforços para atender as demandas de *Big Data* do EXEHDA. Como trabalhos futuros pode-se considerar:

- ampliar a amostragem de testes e a avaliação do impacto da abordagem Híbrida nos diferentes tipos de aplicações que o *middleware* EXEHDA oferece suporte;
- fazer uso de estratégias envolvendo MapReduce.
- adequar os serviços da arquitetura de software do EXEHDA visando as características de *Big Data*;

Referências

- Almeida, R. B. (2013). Segurança da informação e gerenciamento de eventos: Uma abordagem explorando consciência de situação. Monografia de graduação em ciência da computação, Universidade Federal de Pelotas, Pelotas, RS, Brasil.
- Carvalho, A. G. (2014). Interface nosql integrada a banco relacional para gerenciamento de dados em nuvem privada. Monografia bacharelado em engenharia da computação, Centro Universitário de Brasília Faculdade de Tecnologia e Ciências Sociais Aplicadas.
- Couchbase (2012). Acesso em: 7 dez 2014. Couchbase Survey Shows Accelerated Adoption of NoSQL in 2012. Disponível em: <<http://www.couchbase.com/press-releases/couchbase-survey-shows-accelerated-adoption-nosql-2012>>.
- Filho, M. A. P. M. (2015). Sql x nosql: Análise de desempenho do uso do mongodb em relação ao uso do postgresql.
- Han, J., Haihong, E., Guan, L., and Jian, D. (2011). Survey on nosql database. *Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on*, pages 363 – 366.
- Langheinrich, M. (2010). *Privacy in Ubiquitous Computing*. J. Krumm, ed., CRC Press.
- Lopes, J., Souza, R., Geyer, C., Costa, C., Barbosa, J., Pernas, A., and Yamin, A. (2014). A middleware architecture for dynamic adaptation in ubiquitous computing. *j-jucs*, 20(9):1327–1351.
- Machado, R. d. S. (2013). Loga-dm: uma abordagem de análise dinâmica de log com base em mineração de dados. Monografia de graduação em ciência da computação, Universidade Federal de Pelotas, Pelotas, RS, Brasil.
- Marwa, A. (2014). Comparison of data base technologies for apt detection. Phd thesis, Royal Military Academy.
- Moniruzzaman, A. B. M. and Hossain, S. A. (2013). Nosql database: New era of databases for big data analytics - classification, characteristics and comparison. *CoRR*, abs/1307.0191.
- Rambo, I. J. (2015). Ricnr2: Uma proposta não-relacional para tratamento de dados de contexto no exehda. Monografia de graduação em ciência da computação, Universidade Federal de Pelotas, Pelotas, RS, Brasil.
- Sadalage, P. J. and Fowler, M. (2013). *NoSQL Essencial, Um Guia Conciso para o Mundo Emergente da Persistência Poliglota*. Novatec.
- Swift, D. (2010). Successful siem and log management strategies for audit and compliance. Technical report, SANS Institute - InfoSec Reading Room.
- Y Demchenko, C Laat Dee, P. M. (2014). Defining architecture components of the big data ecosystem. *Collaboration Technologies and Systems (CTS), 2014 International Conference on*, pages 104 – 112.
- Yamin, A. C. (2004). *Arquitetura para um Ambiente de Grade Computacional direcionado às Aplicações Distribuídas, Móveis e Conscientes do Contexto da Computação Pervasiva*. PhD thesis, Universidade Federal do Rio Grande do Sul.

Uma análise dos certificados digitais utilizados nas conexões TLS dos aplicativos de Mobile Banking na plataforma Android

**Diego Baierle Sebastiany¹, Mirelle Daiara Vieira Freitas¹,
Luciano Ignaczak¹**

¹ Universidade do Vale do Rio dos Sinos (UNISINOS)
CEP 93.022-000 – São Leopoldo – RS – Brasil

diego.sebastiany@hotmail.com, mdfreitass@outlook.com, lignaczak@unisinos.br

Abstract. *Currently, it is increasingly common to use mobile banking applications on smartphones. Such applications must implement TLS to use encryption to secure communication between the client and the bank. However, often the applications have developmental problems that compromise their safety. This article analyzes whether the digital certificates used by banks from three countries in TLS connections on m-banking applications are recognized as trusted by Android. Furthermore, the size of keys and the validity of these digital certificates is discussed.*

Resumo. *Atualmente, é cada vez mais comum a utilização de aplicativos de mobile banking em smartphones. Tais aplicativos devem implementar o protocolo TLS para empregar criptografia para proteger a comunicação entre o cliente e o seu banco. No entanto, muitas vezes os aplicativos apresentam problemas de desenvolvimento que comprometem a sua segurança. Este artigo analisa se os certificados digitais usados por bancos de três países nas conexões TLS com as aplicações de m-banking são reconhecidos como confiáveis pelo Android. Além disso, é discutido o tamanho das chaves e o período de validade desses certificados digitais.*

1. Introdução

A crescente popularização da *internet* tem levado a um aumento expressivo da quantidade de dispositivos conectados. Com isso, cresceu também a quantidade de usuários que utilizam aplicativos para gerenciamento e movimentação financeira de suas contas bancárias. Uma pesquisa mostra que 52% das transações bancárias feitas no Brasil em 2014 foram realizadas via *internet* e *mobile banking* (m-banking). Entre as contas ativas no país em 2014, 24% dos clientes (25 milhões) realizaram transações utilizando m-banking em seus *smartphones* [Febraban 2015].

Os aplicativos de m-banking precisam implementar mecanismos de segurança para garantir que os dados do usuário não fiquem vulneráveis a roubo e interceptação na Internet. O protocolo TLS (*Transport Layer Security*) é utilizado como padrão para fornecer a segurança nesse ambiente [Elkhodr et al. 2012]. Além de garantir o sigilo e a integridade na comunicação, o protocolo TLS autentica o servidor do banco no qual o aplicativo está se conectando. A autenticação é importante e necessária

para confirmar que o computador que está respondendo é realmente a entidade que afirma ser [Stallings 2008] [Adams and Lloyd 2003].

O estabelecimento de uma comunicação segura exige que a autenticação do banco seja feita de forma correta. Ao fazer o *handshake* do protocolo TLS, o banco envia o seu certificado digital para que o cliente (o aplicativo) verifique a sua identidade. Para isso, o aplicativo deve utilizar um dos certificados raiz instalados no sistema Android para fazer a validação da confiança do certificado do banco. A tentativa de conexão deveria falhar se o certificado raiz usado pelo banco não for considerado confiável pelo Android. No entanto, muitas vezes os aplicativos falham ou simplesmente não executam a validação do certificado digital [Six 2012].

Além da validação da confiança, o tamanho da chave criptográfica e o período de validade de um certificado digital são muito importantes e devem ser considerados no momento de sua emissão para conferir-lhe resistência contra ataques de força bruta. As recomendações mais recentes sugerem um tamanho de chave mínimo de 2.048 bits [Barker and Roginsky 2011]. Além disso, a Microsoft recomenda que certificados digitais com tamanho da chave de 1.024 bits devem possuir até 1 ano de validade; certificados digitais com tamanho de chave de 2.048 bits devem possuir no máximo 2 anos de validade; e certificados digitais com tamanho de chave de 4.096 bits podem possuir validade de até 16 anos [OMeally 2009].

O objetivo deste trabalho é analisar a confiança dos certificados digitais utilizados por bancos nas conexões TLS com os aplicativos de m-banking na plataforma Android, além do período de validade e o tamanho da chave criptográfica desses certificados. A análise foi realizada a partir de uma amostra de 60 aplicativos de m-banking disponibilizados por bancos de três países: Brasil, Estados Unidos e Reino Unido. Para cada aplicativo foi realizada uma simulação de acesso à conta bancária e, a partir do tráfego capturado, foi verificada a utilização do TLS e obtidos os certificados digitais utilizados na conexão.

O restante deste trabalho segue com a seção 2 que apresenta alguns trabalhos relacionados. A seção 3 descreve a metodologia utilizada na realização desta análise, a seção 4 mostra os resultados obtidos da análise dos aplicativos de m-banking e a seção 5 expõe as considerações finais deste trabalho.

2. Trabalhos Relacionados

Muitos aplicativos vêm apresentando problemas de implementação, os quais têm motivado muitos trabalhos que discutem suas causas e possíveis soluções. O trabalho de [Georgiev et al. 2012] mostra que a segurança oferecida pelo TLS depende da correta validação do certificado digital fornecido quando a conexão é estabelecida. Esse trabalho analisa como alguns *softwares* e aplicativos implementam as funções do TLS para validação dos certificados digitais e mostra que mesmo os aplicativos desenvolvidos por grandes empresas possuem falhas graves. Muitas vezes, segundo os autores, as falhas na validação de um certificado é causada pela falta de entendimento e interpretação das APIs (*Application Programming Interface*) utilizadas pelos desenvolvedores. A falta de conhecimento e informação sobre essas APIs conduz o desenvolvedor ao erro e deixa o aplicativo vulnerável a ataques do homem do meio. O trabalho de [Hubbard et al. 2014] realizou uma pesquisa com o objetivo de

identificar falhas na validação dos certificados digitais. Com uma pequena amostra de 41 aplicativos para a plataforma Android, 11 falharam em estabelecer a relação de confiança necessária, pois aceitaram um certificado digital falsificado que, portanto, não pertencia à base de confiança do Android. O artigo também destaca que a falha dos aplicativos pode estar relacionada às APIs utilizadas. Por serem pouco restritivas, permitem que os desenvolvedores cometam erros de implementação do código, permitindo que qualquer certificado digital seja aceito pelo aplicativo ou, até mesmo, que nenhuma validação seja realizada.

A inconsistência da base de certificados raiz da plataforma Android também já foi alvo de estudo. [Vallina-Rodriguez et al. 2014] examinou os certificados raiz instalados nas diversas versões do Android em vários dispositivos. O trabalho analisou a composição dessas bases de confiança e como elas variam de acordo com a versão e marca do dispositivo. Como resultado, foi verificado que a base oficial de certificados digitais confiados pelo Android é modificada ou ampliada. Em alguns casos, o próprio fabricante do dispositivo e/ou a operadora de telefonia adicionam certificados digitais aos dispositivos para estabelecer a relação de confiança para aplicativos embarcados ou prestação de serviços. O trabalho também alerta para o fato que, em dispositivos que rodam com usuário *root*, aplicativos maliciosos podem instalar certificados digitais no Android sem o conhecimento do usuário, quebrando o modelo de confiança de certificados digitais supervisionados e auditados como confiáveis.

[Fahl et al. 2012] investigou o uso inadequado do TLS em 13.500 aplicativos de diversas categorias, obtidos da Google Play Market. Dos aplicativos analisados, 1.074 (17,28% dos que utilizam TLS) continham erros de código do TLS que permitiam a validação de qualquer certificado digital ou confiavam em qualquer certificado raiz. O autor mostra também que as falhas na implementação do TLS ocorrem porque o Android permite que os desenvolvedores criem códigos personalizados para seus aplicativos. Ele destaca que esse recurso devia ser desativado e que as APIs para Android deviam forçar a utilização das implementações padrão do TLS. Em outro trabalho, [Fahl et al. 2013] continua investigando as possíveis causas da má implementação do TLS em aplicativos. Os resultados da pesquisa mostram que as causas não são simplesmente a falta de cuidado por parte dos desenvolvedores, mas também questões e limitações envolvendo o atual paradigma de desenvolvimento do TLS. O trabalho sugere mudanças no atual paradigma em direção a uma maior abstração do código fornecido pelas APIs, permitindo que desenvolvedores utilizem corretamente o TLS com menos esforço e prevenindo falhas na validação dos certificados digitais.

Os trabalhos relacionados reforçam a necessidade de um maior cuidado na implementação do TLS em aplicativos que transmitem dados confidenciais. As falhas de implementação em aplicativos de m-banking podem acarretar muitos prejuízos para o cliente e para o banco. Os artigos citados nesta seção realizaram análises dos certificados digitais de diversos aplicativos, sem abordar um segmento específico. Já este artigo, analisou especificamente como aplicativos de m-banking estão validando os certificados digitais dos bancos.

3. Metodologia

Para a realização desta análise foi selecionada uma amostra com 60 aplicativos de m-banking divididos igualmente em três países: Brasil, Estados Unidos (EUA) e o Reino Unido (UK). A seleção dos aplicativos foi realizada utilizando *rankings* do Banco Central do Brasil¹, do *Federal Reserve System*² para os EUA, e do Relbanks³ para o UK, que classificam os bancos com maiores ativos em cada país. Baseados nestes *rankings*, os autores selecionaram os 20 primeiros bancos de varejo que possuem aplicativos de m-banking. A lista da Relbanks possui apenas 11 bancos e foi utilizada porque não foi encontrado um *ranking* oficial do Banco Central do Reino Unido. A amostra de aplicativos do país foi incrementada com mais 10 bancos conhecidos do Reino Unido, retirados do site do seu Banco Central⁴. A análise consistiu na avaliação das seguintes características de cada aplicativo:

- se o aplicativo utiliza o protocolo TLS para comunicação segura;
- a verificação da confiança no certificado raiz do aplicativo;
- o período de validade do certificado digital do aplicativo;
- o tamanho da chave do certificado digital do aplicativo;

O *software* Genymotion⁵ foi usado para emular um dispositivo rodando a versão 4.4 do sistema Android, que está instalada atualmente em 39,3% dos dispositivos dessa plataforma [Android 2015]. Desse dispositivo foram extraídos todos os certificados digitais armazenados em `/system/etc/security/cacerts/`. Esses são os certificados digitais das autoridades de certificação confiadas por esta versão do Android. Os certificados digitais extraídos foram armazenados para, posteriormente, analisar a confiança dos certificados raiz utilizados nas conexões TLS pelos aplicativos de m-banking. Para possibilitar a análise, os aplicativos de m-banking selecionados foram instalados no dispositivo virtual. Além disso, para que fosse possível a captura do tráfego TLS gerado pelo aplicativo de m-banking foi utilizado o `tcpdump`, disponível no emulador.

Após a instalação de cada aplicativo de m-banking, foram realizadas tentativas de acesso à conta bancária. O acesso foi simulado pela inserção dos dados necessários (como número da conta e senha) aceitos pelo aplicativo, para que ele iniciasse a comunicação com o banco, e assim, estabelecesse a conexão segura (TLS). Com o tráfego gerado pela simulação do acesso foi avaliado o primeiro critério desta análise: se o aplicativo utiliza o TLS.

No caso dos aplicativos de m-banking que possibilitaram a verificação da implementação do protocolo TLS com a captura do tráfego foram extraídos os certificados digitais utilizados pelo *handshake*: o certificado do banco e o certificado raiz, que é utilizado para avaliar a relação de confiança entre o banco e o sistema Android. A partir do certificado digital do banco foi avaliado o tamanho da chave criptográfica bem como o seu período de validade. Para auxiliar na consolidação dos resultados dessa análise foi utilizado um *software* desenvolvido pelos autores,

¹Disponível em: <http://www4.bcb.gov.br/top50/port/top50.asp>

²Disponível em: <http://www.federalreserve.gov/Releases/Lbr/current/default.htm>

³Disponível em: <http://www.relbanks.com/europe/uk>

⁴Disponível em: <http://www.bankofengland.co.uk>

⁵Disponível em: <https://www.genymotion.com>

na linguagem C#, que coleta os dados dos certificados e exporta os resultados no formato XML. O arquivo exportado foi utilizado como fonte para a construção de uma planilha.

Um segundo *software* na linguagem C# também necessitou ser desenvolvido pelos autores para analisar a confiança dos certificados raiz capturados. Esse *software* realizou o cruzamento entre os certificados raiz capturados e a base de certificados raiz considerada confiável pela versão avaliada do Android. O cruzamento desses certificados digitais consistiu em comparar os campos *Subject Key Identifier*, ou na ausência deste, a própria chave pública contida no campo *Subject Public Key Info*. A saída desse programa foi salva e adicionada à planilha anterior, usada como base para a avaliação dos resultados.

Não foi possível avaliar alguns aplicativos de m-banking pois a captura do tráfego desses aplicativos no momento da autenticação não apresenta a utilização do TLS, tampouco revela os dados do usuário em texto claro. Isso pode acontecer quando o aplicativo implementa os requisitos de segurança na camada de aplicação. Por isso, não é possível afirmar que o aplicativo falha em oferecer segurança para o usuário. Os aplicativos com essas características foram classificados como indefinidos.

A última etapa desse trabalho consistiu na realização da análise dos resultados obtidos. Nesta etapa foram efetuados cálculos de porcentagem, cruzamento de informações e médias, a fim de comparar as definições dos bancos dos três países em relação aos dados dos certificados digitais que são alvo deste artigo.

4. Resultados

A análise dos 60 aplicativos selecionados resultou em 2 aplicativos, ambos do Brasil, classificados como indefinidos, e 58 aplicativos que utilizaram o TLS para estabelecer a conexão segura.

Não foi possível verificar a utilização do TLS ao analisar a captura do tráfego gerado pelos 2 aplicativos que foram classificados como indefinidos. Embora não seja possível afirmar, o mecanismo de segurança utilizado por esses aplicativos pode ser o próprio TLS, mas implementado de forma personalizada pelos desenvolvedores. Isso é possível porque as APIs utilizadas para o Android permitem esse nível de personalização do código.

O resultado mostrou que os outros 58 aplicativos analisados utilizam o TLS, realizando o *handshake* e apresentando o certificado digital do banco como é padrão do protocolo. Porém, 18 (31%) desses aplicativos não poderiam ser considerados confiáveis, pois esses utilizam certificados digitais emitidos por autoridades de certificação que não são confiadas pelo Android. O resultado dessa verificação é apresentado na Tabela 1.

A análise da relação de confiança mostrou que o cenário mais preocupante é o brasileiro, onde 44% dos aplicativos analisados não podem ser considerados confiáveis pela versão da plataforma Android analisada. O Reino Unido apresentou o menor número de certificados digitais não confiáveis (15%), porém, ainda é preocupante considerando que o segmento analisado é o bancário, que deveria possuir um cuidado adicional no uso de certificados digitais.

Tabela 1. Certificados raiz sem relação de confiança com o Android.

Origem	Total de certificados raiz não confiáveis	Percentual de certificados raiz não confiáveis
Brasil	8	44,44%
Estados Unidos	7	35,00%
Reino Unido	3	15,00%
TOTAL	18	31,03%

Como foi mostrado pelos trabalhos relacionados, as falhas de validação da confiança expõem o cliente a diversos riscos, e são resultado da forma de implementação do código do aplicativo. Semelhante às análises nesses trabalhos, esta análise dos aplicativos de m-banking revelou um cenário inquietante, pois nenhum dos aplicativos que utilizam certificados não confiados pela plataforma Android apresentou qualquer mensagem de alerta durante o *handshake* do TLS.

A segunda parte desta análise, avaliou o período de validade e o tamanho da chave criptográfica do certificado digital do banco. Todos os 58 certificados digitais possuem o tamanho da chave igual a 2.048 bits com períodos de validade distintos. Os períodos de validade dos certificados digitais usados pelos aplicativos de m-banking são apresentados na Tabela 2.

Tabela 2. Período de validade dos certificados digitais dos bancos.

Origem	Total de certificados	Período de validade			
		1 ano	2 anos	3 anos	4 anos
Brasil	18	10	8	0	0
Estados Unidos	20	14	3	1	2
Reino Unido	20	12	8	0	0

Embora todos os certificados digitais dos bancos analisados atendam à recomendação do NIST no que diz respeito ao tamanho da chave [Barker and Roginsky 2011], 3 deles, todos dos EUA, possuem o período de validade superior a 2 anos. Conforme a recomendação da Microsoft, o período máximo de validade deve ser de 2 anos para certificados com tamanhos de chave de 2.048 bits. Um período de validade muito grande diminui a resistência da chave associada ao certificado digital, pois os avanços da tecnologia de computação podem comprometer um certificado digital que, para os padrões atuais, é considerado forte.

5. Considerações Finais

Quando o usuário instala e utiliza um aplicativo em seu *smartphone*, ele o faz confiando que a comunicação e seus dados estarão seguros. Quando se trata do segmento de m-banking, espera-se que todos os aplicativos implementem o TLS para atender os requisitos de segurança e proteger o usuário. Ao usuário resta apenas confiar no aplicativo, pois o sistema Android não oferece nenhuma indicação de que a comunicação é estabelecida de forma segura.

Existem normas e recomendações que os desenvolvedores de aplicativos devem seguir para atender requisitos no desenvolvimento de seus aplicativos e evitar erros comuns ao utilizar códigos personalizados. O segmento de m-banking deve observar especialmente as recomendações de segurança como a do NIST [Barker and Roginsky 2011] que especifica o tamanho mínimo da chave do certificado digital em 2.048 bits. Como foi mostrado nos resultados deste trabalho, todos os aplicativos seguiram essa recomendação pois todos possuem tamanho da chave igual a 2.048 bits. No entanto, 3 desses certificados digitais possuem o período de validade maior que 2 anos, em desacordo com a recomendação da Microsoft [OMeally 2009]. Isso pode resultar em uma falha, pois os avanços da tecnologia de computação poderão permitir a quebra de chaves com tamanho de 2.048 bits durante o período de validade do certificado digital.

Ademais, uma parcela significativa dos aplicativos, considerando-se sistemas de m-banking, falham na implementação da validação do certificado digital porque a relação de confiança que deveria existir entre o certificado raiz do banco e o sistema Android não é estabelecida. Sem a validação da confiança um certificado digital é aceito sem qualquer restrição, quebrando completamente o sistema de certificação digital, supervisionado e auditado como confiável. Esse problema mostra-se ainda mais grave quando considerado que isso ocorre de forma transparente para o usuário. Embora o protocolo TLS forneça o recurso para avisar o usuário que a relação de confiança não foi estabelecida, muitas vezes esse recurso é desativado ou mau implementado pelo desenvolvedor. Neste trabalho, dos 18 aplicativos que falharam ao estabelecer a relação de confiança, nenhum mostrou qualquer mensagem de aviso sobre essa falha, e todos prosseguiram funcionando como se nenhum erro tivesse ocorrido.

Este trabalho analisou uma amostra reduzida de certificados digitais utilizados por aplicativos de m-banking durante a conexão TLS. Como trabalho futuro é sugerido a análise de uma amostra mais ampla que reflita com mais precisão a realidade no segmento dos aplicativos de m-banking. Além disso, trabalhos futuros podem comparar as características de certificados digitais usados no TLS por aplicativos de outros segmentos.

Referências

- [Adams and Lloyd 2003] Adams, C. and Lloyd, S. (2003). *Understanding PKI: Concepts, Standards, and Deployment Considerations*. Addison-Wesley Pearson Education, Boston, MA, second edition.
- [Android 2015] Android, D. (2015). Dashboards, platform versions. Disponível em: <https://developer.android.com/about/dashboards/index.html>.
- [Barker and Roginsky 2011] Barker, E. and Roginsky, A. (2011). Transitions: Recommendation for transitioning the use of cryptographic algorithms and key lengths. NIST Special Publication 800-131A. Disponível em: <http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf>.
- [Elkhodr et al. 2012] Elkhodr, M., Shahrestani, S., and Kourouche, K. (2012). A proposal to improve the security of mobile banking applications. In *ICT and*

Knowledge Engineering (ICT Knowledge Engineering), 2012 10th International Conference on, pages 260–265.

[Fahl et al. 2012] Fahl, S., Harbach, M., Muders, T., Baumgärtner, L., Freisleben, B., and Smith, M. (2012). Why eve and mallory love android: An analysis of android ssl (in)security. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS ’12, pages 50–61, New York, NY, USA. ACM.

[Fahl et al. 2013] Fahl, S., Harbach, M., Perl, H., Koetter, M., and Smith, M. (2013). Rethinking ssl development in an appified world. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS ’13, pages 49–60, New York, NY, USA. ACM.

[Febraban 2015] Febraban (2015). Pesquisa febraban de tecnologia bancária 2014. Disponível em: https://www.febraban.org.br/Noticias1.asp?id_texto=2626.

[Georgiev et al. 2012] Georgiev, M., Iyengar, S., Jana, S., Anubhai, R., Boneh, D., and Shmatikov, V. (2012). The most dangerous code in the world: Validating ssl certificates in non-browser software. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS ’12, pages 38–49, New York, NY, USA. ACM.

[Hubbard et al. 2014] Hubbard, J., Weimer, K., and Chen, Y. (2014). A study of ssl proxy attacks on android and ios mobile applications. In *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th*, pages 86–91.

[OMeally 2009] OMeally, Y. (2009). Recommendations for pki key lengths and validity periods with configuration manager. Disponível em: <http://blogs.technet.com/b/configmgrteam/archive/2009/06/12/recommendations-for-pki-key-lengths-and-validity-periods-with-configuration-manager.aspx>.

[Six 2012] Six, J. (2012). *Segurança de aplicativos Android*. Novatec Editora Ltda., São Paulo, SP.

[Stallings 2008] Stallings, W. (2008). *Criptografia e segurança de redes*. Pearson Education do Brasil Ltda., São Paulo, SP, fourth edition.

[Vallina-Rodriguez et al. 2014] Vallina-Rodriguez, N., Amann, J., Kreibich, C., Weaver, N., and Paxson, V. (2014). A tangled mass: The android root certificate stores. In *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*, CoNEXT ’14, pages 141–148, New York, NY, USA. ACM.

EXEHDA-IoT: Uma Abordagem Consciente de Contexto Direccionada à Internet das Coisas

Patrícia Davet¹, Huberto Kaiser Filho¹, Leonardo João¹, Lucas Xavier¹, Tainá Carvalho¹, Rodrigo Souza², João Lopes², Adenauer Yamin¹

¹Centro de Desenvolvimento Tecnológico (CDTec) – Universidade Federal de Pelotas (UFPel)
Caixa Postal 354 – 96010-900 – Pelotas – RS – Brazil

²Instituto de Informática
Universidade Federal do Rio Grande do Sul (UFRGS) – Porto Alegre, RS – Brazil

{ptdavet, hkaiser, ldrsjoao, lmdsxavier, trcarvalho, adenauer}@inf.ufpel.edu.br,
{rssouza, jlblopes}@inf.ufrgs.br

Abstract. One of the major research challenges of Ubiquitous Computing is related to the need of the applications being aware of their context of interest, and when appropriate, respond to it. This paper presents an architecture for context awareness, called EXEHDA-IoT. We consider that the main contribution of this work is an IoT based architecture that supports the managing of the acquisition, storage, and processing of context data, in a distributed way, independently of the application, in an autonomic perspective and rule-based. To assess the functionality of the EXEHDA-IoT, we present a case study, highlighting the prototyping and tests performed.

Resumo. Um dos principais desafios de pesquisa da Computação Ubíqua está relacionado à necessidade das aplicações terem consciência do seu contexto de interesse, e quando for o caso, reagir ao mesmo. Este artigo apresenta uma arquitetura para consciência do contexto, denominada EXEHDA-IoT. Considera-se como principal contribuição deste trabalho a concepção de uma arquitetura direcionada à IoT para suporte ao gerenciamento da aquisição, armazenamento e processamento dos dados de contexto, de forma distribuída, independente das aplicações, em uma perspectiva autonômica baseada em regras. Para avaliar as funcionalidades do EXEHDA-IoT é apresentado um estudo de caso, destacando a prototipação e os testes realizados.

1. Introdução

A Internet das Coisas, do inglês *Internet of Things* (IoT) vem se consolidando como o novo paradigma de evolução da Internet, na qual se preconiza a ideia do “tudo conectado”, ou seja, qualquer “coisa” (pessoa, animal ou objeto) pode se comunicar através da Internet, possuindo uma identificação única e sem a obrigatoriedade de uma intervenção humana. Esta visão da IoT promove a integração do mundo físico ao digital, de forma a criar uma rede de objetos inteligentes incorporados ao ambiente de forma ubíqua [Perera et al. 2013].

Esses objetos inteligentes possuem dispositivos embarcados com capacidade para armazenar e processar os dados sensorados (captados do mundo físico por sensores), que

interconectam-se com outros dispositivos e recursos (físicos ou virtuais), o que permite o surgimento de uma miríade de aplicações.

Aplicações de IoT, enquanto ubíquas e assim dotadas de comportamento autônomo, e com uma organização distribuída, necessitam ter consciência dos dados contextuais que lhe interessam e, quando for o caso, se adaptar aos mesmos. Esta classe de sistemas computacionais, reativos ao contexto, abre perspectivas ao desenvolvimento de aplicações mais ricas, elaboradas e complexas, e que exploram a natureza dinâmica das modernas infraestruturas computacionais, bem como a mobilidade do usuário [Caceres and Friday 2011].

No entanto, para a implementação de aplicações conscientes de contexto em cenários introduzidos pela IoT, uma série de desafios devem ser superados para a concepção de infraestruturas que atendam à demandas destes novos cenários. Um dos principais desafios está relacionado com a alta heterogeneidade decorrente da grande diversidade de tecnologias de *hardware* e *software* presentes neste ambiente, o que faz com que haja uma busca de soluções que permitam a interoperabilidade e integração destes diferentes componentes. Uma alternativa de solução promissora para tais desafios, como o da heterogeneidade, está na utilização de plataformas de *middleware* [Maia et al. 2015].

Este trabalho, denominado EXEHDA-IoT tem como objetivo principal contribuir com o Subsistema de Reconhecimento de Contexto e Adaptação do *middleware* EXEHDA (*Execution Environment for Highly Distributed Applications*) [Lopes et al. 2014] com uma abordagem que o capacite para atendimento das demandas IoT. A expectativa é obter uma arquitetura de *software* distribuída e extensível, comprometida com a premissa de fornecer mecanismos para o desenvolvimento de aplicações conscientes de contexto na IoT.

O artigo está organizado da seguinte forma: a Seção 2 caracteriza o *middleware* EXEHDA. A Seção 3 descreve a modelagem da arquitetura de software proposta. A Seção 4 apresenta um estudo de caso. A Seção 5 discute os trabalhos relacionados. Por fim, a Seção 6 apresenta as considerações finais.

2. Middleware EXEHDA

O EXEHDA é um *middleware* baseado em serviços que visa criar e gerenciar um ambiente ubíquo, bem como promover a execução de aplicações sobre esse ambiente. No EXEHDA, as condições de contexto são pró-ativamente monitoradas e o suporte à execução deve permitir que tanto a aplicação como ele próprio utilizem essas informações na gerência da adaptação de seus aspectos funcionais e não-funcionais. O mecanismo de adaptação proposto para o EXEHDA emprega uma estratégia colaborativa entre aplicação e ambiente de execução, através da qual é facultado ao programador individualizar políticas de adaptação para reger o comportamento de cada um dos componentes que constituem o software da aplicação [Lopes et al. 2014].

A estrutura de software do EXEHDA contempla um núcleo mínimo e serviços carregados sob demanda, os quais estão organizados nos seguintes subsistemas: (i) Reconhecimento de Contexto e Adaptação, (ii) Acesso Ubíquo, (iii) Execução Distribuída e (iv) Comunicação.

O EXEHDA-IoT contribui especificamente com o Subsistema de Reconheci-

mento de Contexto e Adaptação, ampliando as funcionalidades dos serviços que tratam a extração da informação direta dos sensores, passando pela identificação em alto nível dos elementos de contexto, até o disparo das ações de adaptação em reação a modificações no estado de tais elementos de contexto.

3. EXEHDA-IoT: Visão e Modelagem da Arquitetura Direcionada à IoT

Dentre os desafios inerentes a cenários IoT para aplicações conscientes de contexto destacam-se: (i) a aquisição do contexto a partir de um grande número de dispositivos heterogêneos e distribuídos; (ii) o processamento das informações de contexto adquiridas e a respectiva atuação sobre o meio físico; e (iii) a disponibilização dos dados contextuais processados aos usuários de forma distribuída e no momento oportuno.

No EXEHDA-IoT os dispositivos computacionais do ambiente ubíquo são distribuídos em células, sendo cada célula constituída dos seguintes componentes: (i) EXEHDAbase, o elemento central da célula, sendo responsável por todos serviços básicos e constituindo referência para os demais elementos; (ii) o EXEHDAano que corresponde aos dispositivos computacionais responsáveis pela execução das aplicações; (iii) o EXEHDAano móvel, um subcaso do anterior, que corresponde aos dispositivos tipicamente móveis que podem se deslocar entre as células do ambiente ubíquo, como *notebooks*, *tablets* ou *smartphones*; e (iv) o EXEHDAaborda que consiste no elemento de borda do ambiente ubíquo, responsável por fazer a interoperação entre os serviços do *middleware* e as redes de sensores e atuadores.

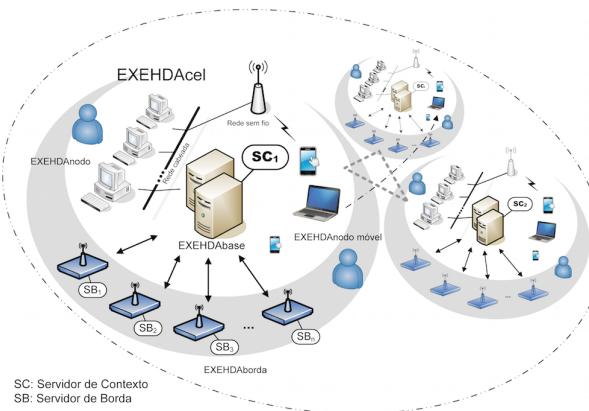


Figura 1. Organização Celular do Ambiente Ubíquo Gerenciado pelo EXEHDA-IoT

Para provimento de consciência de contexto no ambiente ubíquo, o EXEHDA-IoT (vide Figura 1) se vale de dois tipos principais de servidores: o Servidor de Borda, responsável pela interação com o meio através de sensores, atuadores ou *gateways*, e o Servidor de Contexto que atua no armazenamento e processamento das informações contextuais. A arquitetura proposta provê comunicação (i) entre os Servidores de Borda e o Servidor de Contexto; (ii) entre os Servidores de Contexto localizados em diferentes células do ambiente ubíquo; e (iii) com outros serviços do *middleware* ou aplicações. Os principais módulos dos Servidores de Contexto e Borda estão descritos a seguir.

3.1. Servidor de Contexto

Uma visão do Servidor de Contexto e sua relação com o Servidor de Borda pode ser vista na Figura 2. O Servidor de Contexto está organizado em seis módulos autônomos descritos a seguir, os quais interoperam no provimento das funcionalidades necessárias ao Serviço de Consciência de Contexto do EXEHDA.

Coletor: provê suporte à captura das informações contextuais, coletadas pelos Servidores de Borda, considerando sensores lógicos (software) e/ou hardware.

Atuador: controla os atuadores, após ser notificado pelos outros módulos do Servidor de Contexto. Dispara no ambiente ubíquo ações que mudam o estado do meio, viabilizando o uso de serviços de consciência de contexto em aplicações de controle e automação.

Interpretador: realiza tarefas de manipulação e dedução das informações contextuais, utilizando para isto informações especificadas nos **Contextos de Interesse** das aplicações. Este módulo mantém um **Repositório de Contexto**, onde são armazenadas as informações contextuais obtidas pelo Coletor, provendo a possibilidade de registro histórico dos contextos, o que permite a construção de regras que explorem aspectos temporais. Esses dados são utilizados pelas regras do componente **Tratador de Regras**, o qual dispara as ações pertinentes em função do estado contextual. A natureza das regras - tratamento lógico, numérico ou temporal - é uma decorrência do tipo de domínio da aplicação atendida pelo Serviço de Consciência de Contexto do EXEHDA.

Notificador: notifica o resultado do processamento contextual realizado pelo Interpretador. Recebe subscrições de todos os serviços e/ou aplicações que desejem notificações a respeito dos estados contextuais, interoperando através do módulo Gerenciador de Comunicação. Passa pelo Notificador todas as decisões de atuação provenientes do tratamento autônomico de regras de processamento contextual.

Gerenciador de Comunicação: empregado por Servidores de Contexto remotos e/ou aplicações quando da solicitação de dados contextuais e/ou o disparo de atuadores. Esse módulo provê a disseminação de informações para outros serviços do *middleware*, bem como o envio de mensagens aos usuários.

Gerenciador de Configuração: permite ao usuário um gerenciamento confortável das configurações do Servidor de Contexto. O mesmo provê facilidades para que sejam especificados os diferentes aspectos dos sensores e atuadores, bem como informações dos equipamentos cujo contexto está sendo aquisitado.

Gerenciador de Acesso Móvel: provê acesso móvel ao EXEHDA-IoT, possibilitando a exibição de informações contextuais e a disponibilização de alertas proativos. Particularmente, a disponibilização de alertas proativos em uma plataforma de hardware que possa acompanhar o usuário enquanto este desempenha suas atividades nos mais diferentes lugares, se mostrou um procedimento que potencializa a ubiquidade da solução de consciência de contexto disponibilizada.

3.2. Servidor de Borda

A arquitetura do Servidor de Borda contempla três módulos principais (vide Figura 2), os quais são destinados a: (i) tratar a rede de sensores; (ii) efetuar as publicações; e (iii) tratar a rede de atuadores. A seguir é realizada uma discussão sobre estes módulos.

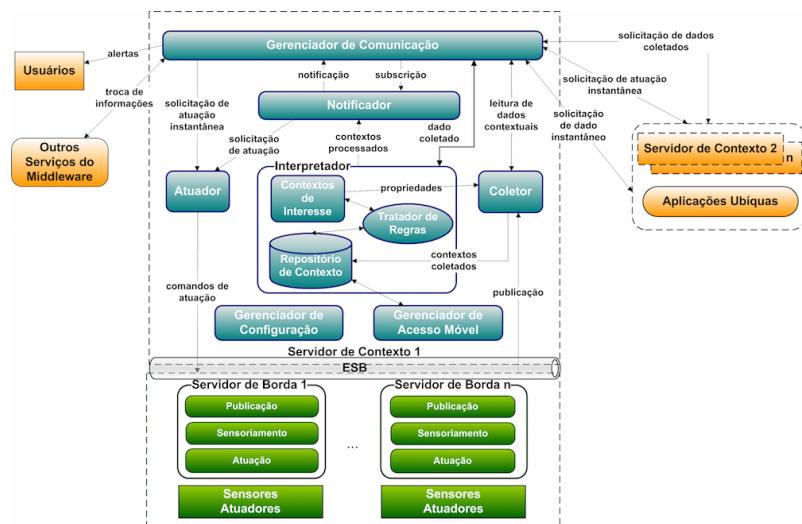


Figura 2. Arquitetura EXEHDA-IoT

O **Módulo de Sensoriamento** provê o tratamento de uma rede de sensores permitindo uma individualização de processamento por sensor. Este tratamento é responsável por aspectos desde gerência física (interfaces, frequência de leitura) até normalização computacional (validação, tradução) dos valores aquisitados. Também, este módulo provê funcionalidades para publicação das informações coletadas da rede de sensores no Servidor de Contexto.

O **Módulo de Publicação** é responsável por coordenar o principal fluxo de dados entre os Servidores de Borda e o Servidor de Contexto, promovendo a publicação de todos os dados coletados e garantindo uma **Persistência Local** dos mesmos nos períodos que a publicação ficar inviabilizada.

O **Módulo de Atuação** é responsável pelo gerenciamento dos atuadores. Pode receber comandos de atuação originários tanto do Servidor de Contexto como consequência da execução de uma regra, como de uma aplicação controlada pelo usuário.

4. Estudo de Caso

Nesta seção estão resumidos os principais aspectos do estudo de caso relacionado ao projeto AMPLUS¹, empregado na avaliação das funcionalidades do EXEHDA-IoT. O estudo de caso contemplou tarefas referentes ao sensoriamento e a coleta de dados contextuais, bem como o processamento e notificação das informações de contexto. Neste estudo de caso foi desenvolvida uma aplicação para interface Web, e outra para dispositivos móveis.

O Projeto AMPLUS foi concebido para prover serviços de consciência do contexto, permitindo um registro dos estados contextuais em que se encontram os equipamentos do Laboratório Didático de Análise de Sementes - LDAS², ao longo de todo o tempo de realização dos vários testes, e uma atuação proativa quando necessário.

¹AMPLUS: <http://amplus.ufpel.edu.br/>

²LDAS: <http://amplus.ufpel.edu.br/ldas>

Para prototipação dos mecanismos de coleta, armazenamento e processamento do contexto, bem como atuação junto ao meio foram priorizadas tecnologias escaláveis e robustas. Nesse sentido, o código dos servidores de contexto e de borda está escrito na linguagem Python, sendo empregado XML-RPC³ (*Extensible Markup Language - Remote Procedure Call*). O repositório de contextos emprega o gerenciador PostgreSQL para implementação das bases de dados. Os sensores e atuadores são acessados pelo protocolo 1-Wire⁴.

A interface Web possibilita a seleção do contexto de interesse a ser exibido, disponibilizando relatórios textuais e gráficos. O relatório gráfico (vide Figura 3) oferecido pelo sistema permite a visualização simultânea das informações de vários sensores. A seleção dos sensores é feita a partir de um menu com suporte a múltipla seleção. Também é disponibilizado um recurso de inspeção que permite a comparação dos valores em um determinado instante do tempo.



Figura 3. Projeto AMPLUS - Relatório gráfico

Ainda, com o intuito de promover a proatividade do Projeto AMPLUS junto a comunidade usuária, foram desenvolvidas interfaces para dois serviços públicos de comunicação: e-mail e SMS para rede celular. Estas mensagens são produzidas a partir do processamento das regras contextuais de forma autônoma pelo Servidor de Contexto.

Para atender o fato da rotina de trabalho dos laboratoristas do LDAS implicar em uma mobilidade nos diversos recintos do laboratório, foi desenvolvida uma interface de alerta visual, a qual é ativada sempre que um dispositivo estiver em um estado contextual que exija atenção. A partir deste alerta, detalhes podem ser inferidos através da interface computacional do Projeto AMPLUS.

A interface para acesso móvel foi concebida para a plataforma Android. Os relatórios gráficos e textuais oferecem a opção do usuário especificar intervalo de

³XML-RPC: <http://www.xmlrpc.com>

⁴1-Wire: <http://ubiq.inf.ufpel.edu.br/1-wire/>

visualização (hora, dia, semana), sendo o ajuste do eixo vertical feito de forma automática, minimizando o emprego de rolagem de tela. Para exibição dos alertas foi explorada a interface disponibilizada pela plataforma Android para esta finalidade, este aspecto potencializa a integração do mecanismo de alertas às funcionalidades do *smartphone* do usuário. As interfaces correspondentes a estas funcionalidades são exibidas na Figura 4.



Figura 4. Interfaces da aplicação móvel

5. Trabalhos Relacionados

O estudo dos trabalhos relacionados CARE [Agostini et al. 2009], CoCA [Ejigu et al. 2008], HiCon [Cho et al. 2008], Solar [Chen et al. 2008], WComp [Tigli et al. 2009] foi desenvolvido considerando as premissas de concepção do EXEHDA-IoT: (1) arquitetura (distribuída ou centralizada); (2) sensoriamento (suporte a redes de sensores); (3) aquisição dos dados de contexto; (4) suporte ao tratamento de regras e; (5) suporte à atuação sobre o meio.

As arquiteturas estudadas não mantém um caráter descentralizado para todas as etapas de tratamento dos dados de contexto, o que não é oportuno para o requisito de distribuição em larga escala dos ambientes ubíquos. Por sua vez, o modelo arquitetural do EXEHDA-IoT diferencia-se dos trabalhos relacionados por estar estruturado de forma distribuída, em todas as etapas de tratamento das informações de contexto, desde a aquisição até os procedimentos de atuação sobre o meio.

O EXEHDA-IoT pode gerenciar redes de sensores e atuadores, tal característica é encontrada em parte nos projetos CoCA e HiCon, que têm suporte a redes de sensores. O projeto WComp, por sua vez, permite atuação sobre o meio, entretanto, não suporta o gerenciamento de redes de atuadores.

Com exceção dos projetos CARE e Solar, os demais prevêem o emprego de mecanismos específicos para aquisição do contexto, que adotam uma estratégia de separação entre a obtenção e o uso do contexto. Além de contemplar esse aspecto, o EXEHDA-IoT apresenta um diferencial em relação aos projetos relacionados, que é o emprego de um caráter autônomo na aquisição dos dados de contexto, visto que estes continuam a ser obtidos pelo mecanismo, mesmo que as aplicações interessadas em seu uso estejam inoperantes.

A maioria dos projetos estudados possui suporte ao tratamento de regras, porém esta funcionalidade usualmente está restrita a algumas etapas do processamento do con-

texto. O EXEHDA-IoT, diferencia-se destes trabalhos, por sua arquitetura de software dar suporte ao tratamento distribuído de regras, operando tanto nos Servidores de Borda, como nos Servidores de Contexto.

6. Considerações Finais

O principal diferencial do EXEHDA-IoT em relação aos trabalhos relacionados diz respeito a possibilidade de gerenciar a aquisição, armazenamento e processamento dos dados de contexto, de forma distribuída, independente das aplicações, em uma perspectiva autonômica baseada em regras. No que tange à interoperabilidade entre os servidores que compõem o Serviço de Consciência de Contexto, o EXEHDA-IoT contempla uma abordagem compatível com a expectativa de operação, utilizando protocolos padrões da Internet para às comunicações.

Como trabalhos futuros, os seguintes aspectos foram priorizados: (i) explorar estudos de caso em que as regras de processamento contextual utilizem outros mecanismos de inferência de mais alto nível; e (ii) utilizar a arquitetura do EXEHDA-IoT para provimento de consciência de situação.

Referências

- Agostini, A., Bettini, C., and Riboni, D. (2009). Hybrid reasoning in the CARE middleware for context awareness. *International Journal of Web Engineering and Technology*, 5(1):3.
- Caceres, R. and Friday, A. (2011). Ubicomp systems at 20: Progress, opportunities, and challenges. *IEEE Pervasive Computing*, 11(1):14–21.
- Chen, G., Li, M., and Kotz, D. (2008). Data-centric middleware for context-aware pervasive computing. *Pervasive and Mobile Computing*, 4(2):216–253.
- Cho, K., Hwang, I., Kang, S., Kim, B., and Lee, J. (2008). HiCon: a hierarchical context monitoring and composition framework for next-generation context-aware services. *Network*, . . . , 22(4):34–42.
- Ejigu, D., Scuturici, M., and Brunie, L. (2008). Hybrid Approach to Collaborative Context-Aware Service Platform for Pervasive Computing. *Journal of Computers*, 3(1):40–50.
- Lopes, J., Souza, R., Geyer, C., Costa, C., Barbosa, J., Pernas, A., and Yamin, A. (2014). A middleware architecture for dynamic adaptation in ubiquitous computing. *J.UCS*, 20(9):1327–1351.
- Maia, P., Baffa, A., Cavalcante, E., Delicato, F. C., Batista, T., and Pires, P. F. (2015). Uma plataforma de middleware para integração de dispositivos e desenvolvimento de aplicações em e-health. *Anais do XXXIII SBRC*, pages 361–374.
- Perera, C., Zaslavsky, A., Christen, P., and Georgakopoulos, D. (2013). Context aware computing for the internet of things: A survey. *Communications Surveys & Tutorials, IEEE*, 16(1):414–454.
- Tigli, J.-Y., Lavirotte, S., Rey, G., Hourdin, V., Cheung-Foo-Wo, D., Callegari, E., and Riveill, M. (2009). WComp middleware for ubiquitous computing: Aspects and composite event-based Web services. *annals of telecommunications - annales des télécommunications*, 64(3-4):197–214.

VI

Sessão 6 - Modelagem e Análise de Desempenho

Análise Comparativa entre HTTP 1.1 e HTTP 2.0

Simei Tabordes Gonçalves¹, Eduardo Maroñas Monks¹

¹Curso de Tecnologia em Redes de Computadores
Faculdade de Tecnologia SENAC Pelotas (FATEC)
Rua Gonçalves Chaves 602 – 96015560 – Pelotas – RS – Brazil

{tabordes, emmonks}@gmail.com

Abstract. This article aims to show a comparative analysis in a test environment, the impact generated by the adoption of the HTTP 2.0 protocol in web servers. Will be used as web servers Apache, Nginx, G-WAN and IIS. The tests consists in multiple access to a server, where the results will be evidenced by the log analysis and packet capture.

Keywords: HTTP2, Apache, Nginx, G-WAN, IIS

Resumo. Este artigo se propõe a mostrar através de uma análise comparativa em ambiente de testes, o impacto gerado pela adoção da versão 2.0 do protocolo HTTP em servidores de páginas. Serão usados como servidores Apache, Nginx, G-WAN e IIS. Os testes consistem em múltiplos acessos em um servidor, onde os resultados serão evidenciados pela análise de logs e captura de pacotes.

Palavras-Chave: HTTP2, Apache, Nginx, G-WAN, IIS

1. Introdução

Em maio de 2015 foi publicada a RFC7540 [Belsh 2015] que padroniza o HTTP 2.0, que teve como foco segurança e desempenho. Esta nova versão do protocolo tem como características principais a utilização da compactação como padrão, obrigatoriedade do uso de SSL e multiplexação de requisições. O resultado final é um protocolo mais seguro que usa menos conexões e consequentemente menos recursos de rede. Com a expansão constante da Internet e um número cada vez maior de usuários, a busca por eficiência entre os protocolos de rede é a chave para um melhor aproveitamento dos recursos de clientes e servidores, resultando numa melhor experiência para o usuário final. As características do HTTP 2.0 como multiplexação e compactação de cabeçalhos tornam a comunicação mais robusta e aumentam a escalabilidade dos servidores. Neste artigo serão mostrados testes que irão evidenciar o impacto dessas novas características.

2. Protocolos

O protocolo HTTP é um dos mais utilizados na Internet. Segundo NETCRAFT [NETCRAFT 2015], existem hoje aproximadamente 176.788.328 páginas disponíveis.

2.1. Protocolo HTTP 1.1

O HTTP 1.1 [Fielding 1999] é um protocolo no nível de aplicação que vem sendo usado desde a sua versão 0.9 em 1990. Na versão 1.0 ainda não era implementada a técnica de *pipelining*, que acabou sendo implementada na versão 1.1. O HTTP *pipelining* foi um grande avanço, porque consistia no envio de múltiplas requisições dentro de uma

conexão TCP, enquanto no HTTP 1.0 a cada requisição resultava em mais uma conexão TCP. Embora esta melhoria tenha sido significativa em 1999, quando as conexões eram mais lentas que hoje em dia, havia uma limitação. As respostas do servidor às requisições tinham que ser retornadas na mesma sequência que foram enviadas. A Figura 1 ilustra a técnica de *pipelining*.

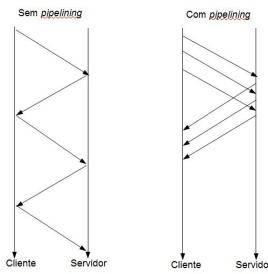


Figura 1. HTTP *pipelining*.

2.2. Protocolo SPDY

O SPDY [Google .inc 2012] é um protocolo que tem por objetivo reduzir o tempo de carregamento das páginas. Este protocolo permite compactar cabeçalho de respostas e requisições, o que reduz a banda utilizada quando os cabeçalhos se repetem, o que acontece frequentemente. Permite também a multiplexação de requisições, ou seja, substituiu o HTTP *pipelining*, já que agora as requisições são enviadas de forma assíncrona e, as respostas também retornam de forma assíncrona. No início de 2015 o time de desenvolvimento do SPDY achou por bem abandonar o projeto e se juntar ao time de desenvolvimento do HTTP 2.0, já que a maioria dos benefícios do SPDY já estavam presentes no HTTP 2.0 [Grigorik 2013]. Na Figura 2, um diagrama que compara o uso de pipelining do protocolo HTTP 1.1 com a multiplexação no SPDY.

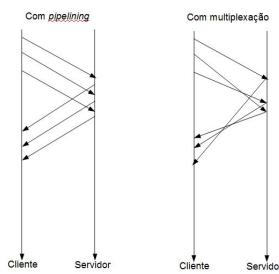


Figura 2. Comparaçāo do uso de multiplexaçāo do protocolo SPDY com o *pipelining* do protocolo HTTP 1.1.

2.3. Protocolo HTTP 2.0

No mês de maio de 2015, foi registrada a RFC 7540 [Belsh 2015] referente ao HTTP 2.0. Algumas características como o uso padrão de SSL e compressão, que já existiam no HTTP 1.1 apenas foram adotadas por padrão, e outras inteiramente novas tais como o uso de *streams*, multiplexação e compressão de cabeçalhos, foram adicionadas.

2.3.1. Streams e Multiplexação

A multiplexação se dá pelo uso de *streams*. Uma *stream* é uma sequência de *frames* trocados entre cliente e servidor de forma bidirecional. Uma conexão HTTP2 pode conter múltiplas *streams* concorrentes abertas. A *stream* pode ser estabelecida e usada unilateralmente ou compartilhada pelo cliente ou servidor, e podem ser fechadas por qualquer uma das pontas.

2.3.2. Server Push

O HTTP 2.0 permite ao servidor opcionalmente enviar *frames* antecipadamente para um cliente. Isto é útil quando o servidor sabe que o cliente irá necessitar de um objeto na página, antes mesmo dele fazer a requisição. Através da multiplexação, esses frames não requisitados pelo cliente chegam em paralelo com os que foram requisitados.

2.3.3. Gerenciamento de conexões

As conexões HTTP 2.0 são persistentes. É esperado que clientes não fechem suas conexões, até que nenhuma comunicação com o servidor seja necessária. Os clientes não devem abrir mais de uma conexão TCP.

2.3.4. Compressão de cabeçalhos

As listas de cabeçalho são uma coleção de zeros ou mais cabeçalhos. Quando transmitidos por uma conexão, uma lista de cabeçalhos é serializada em um bloco usando compressão de cabeçalho HTTP.

2.4. SSL obrigatório

Hoje ainda opcional nos servidores de páginas, devido a maioria ainda utilizar HTTP 1.1, no HTTP 2.0 tornou-se obrigatório o uso de SSL nas conexões.

3. Servidores de páginas

Foram testados os servidores G-WAN [Trueleap 2007], Nginx [Sysoev 2004], Apache [Behlendorf 1995] e o Microsoft IIS [Microsoft 2015]. Os três primeiros são de código aberto, enquanto que o IIS da Microsoft é de código fechado.

3.1. G-WAN

O servidor de páginas G-WAN, é focado em multiprocessamento. A idéia é que otimizar o aproveitamento de múltiplos cores dará os melhores resultados. Até o momento não há notícias de que o G-WAN passará a suportar HTTP 2.0, porém foi incluído nos testes para comparação com os outros servidores de páginas, devido ao desempenho otimizado com o protocolo HTTP 1.1.

3.2. Nginx

O Nginx é um servidor de páginas de código aberto e alta performance. Entre suas grandes vantagens estão a grande escalabilidade e baixo uso de memória RAM.

3.3. Apache

O Apache é um servidor de páginas de código aberto lançado em 1995. Tem um grande número de extensões e sempre se mantém em sincronia com os padrões HTTP atuais. É o servidor mais popular desde abril de 1996 NETCRAFT [NETCRAFT 2015].

3.4. Microsoft IIS

O servidor de páginas IIS, é um serviço disponível em sistemas operacionais Windows, desde o Windows NT 3.51. Segundo NETCRAFT [NETCRAFT 2015] o IIS é junto com Apache e Nginx, um dos servidores de páginas mais usados na internet.

4. Testes

Os testes foram realizados utilizando-se um notebook, como servidor de virtualização executando o VMware Workstation [VMware 2015], para os servidores de páginas, e cinco máquinas clientes usando o Linux Ubuntu 14.04.

4.1. Página de teste (Flags)

A página Flags [Liu 2015] é composta de duzentas e cinquenta e quatro bandeiras de países. As várias bandeiras geram múltiplas conexões com o servidor, e é usada para comparação de desempenho entre o HTTP 1.1 e HTTP 2.0. Cada bandeira tem aproximadamente 1,6KB. Durante os testes, as conexões usadas para fazer as requisições das bandeiras se comportaram de forma diferente entre o HTTP 1.1 e HTTP 2.0. No HTTP 1.1 o *download* das bandeiras, criou diversas conexões TCP. No HTTP 2.0, foi criada apenas uma conexão para o conteúdo da página, exemplificando o uso de *streams*. na Figura 3 a página *flags*.



Figura 3. Página de testes Flags

4.2. Ambiente de Testes 1

No primeiro ambiente de testes, foi usado apenas um computador, que rodava o servidor a ser testado no VMware, e também era usado como cliente que acessava o servidor a partir de um navegador com suporte a HTTP 2.0 e SPDY. Quando havia controle de banda, era feito via VMware.

4.3. Ambiente de Testes 2

Os testes foram feitos em laboratório na Faculdade de Tecnologia SENAC Pelotas. Foram usadas cinco máquinas virtuais em cinco computadores, que abriam simultaneamente várias abas com o endereço do servidor de páginas a ser testado. Foram feitos testes com

dez e cem abas sendo abertas simultaneamente em cada cliente. Os acessos eram disparados por comandos remotos via SSH. Foi utilizado o WANem para simular o ambiente de rede da Internet. Foram configurados 30ms de latência, 0.148% de perda de pacotes e 4,95Mb/s de banda entre servidores e clientes. Estes números foram baseados em estatística encontradas em um estudo feito por Chen [Chen 2012] sobre o desempenho de redes Wifi, 3G e 4G em universidades americanas.

4.4. Testes com o G-WAN

O G-WAN foi instalado em um servidor Debian Linux versão 7. Como o G-WAN não tem suporte a HTTP 2.0 ou SPDY, houveram apenas 2 testes, um de 50 conexões simultâneas e outro com 500 conexões simultâneas, ambos com HTTP 1.1. Na Tabela 1 são apresentados os resultados dos testes com o G-WAN.

Tabela 1. Resultados dos testes com o G-WAN.

Abas	Conexões(TCP)	Duração(s)	Dados(MB)
50	30	11,03	6,7
500	32	16,81	9,9

4.5. Testes com o Nginx

O servidor de páginas Nginx foi instalado em um servidor Linux CentOS 6.3 e foi ativado um módulo do SPDY 3.1. Foram feitos quatro testes de 50 e 500 abas, sendo dois com HTTP 1.1 e dois com SPDY 3.1. Na Tabela 2 estão listados os resultados deste teste.

Tabela 2. Resultados dos testes com o Nginx

Abas	Conexões(TCP)	Duração(s)	Protocolo	Dados(MB)
50	30	11,21	HTTP 1.1	6,8
500	61	13,81	HTTP 1.1	13,1
50	31	11,38	HTTPS 1.1	7
500	56	10,05	HTTPS 1.1	10,4
50	5	10,05	SPDY 3.1	7
500	5	16,81	SPDY 3.1	10,3

4.6. Testes com Apache

O servidor de páginas Apache foi instalado em um servidor Linux Debian 7.0 e ativado o módulo do SPDY 3.1. Foram feitos quatro testes de 50 e 500 conexões, sendo dois com HTTP 1.1 e dois com SPDY 3.1. Na Tabela 3 estão listados os resultados deste teste.

Tabela 3. Resultados dos testes com o Apache

Aba	Conexões(TCP)	Duração(s)	Protocolo	Dados(MB)
50	32	13,85	HTTP 1.1	6,8
500	60	11,62	HTTP 1.1	10,1
50	34	13,34	HTTPS 1.1	7,1
500	56	13,03	HTTPS 1.1	10,5
50	5	9,72	SPDY 3.1	6,4
500	5	14,47	SPDY 3.1	9,25

4.7. Testes com IIS

A versão do IIS utilizada foi a versão beta do Windows 10 Server. Através de uma modificação no registro, é possível habilitar o HTTP 2.0 [Nazim 2014].

Tabela 4. Resultados dos testes com o IIS

Aba	Conexões(TCP)	Duração(s)	Protocolo	Dados(MB)
50	31	9,43	HTTP 1.1	6,9
500	31	20,24	HTTP 1.1	10,3
50	45	11,53	HTTPS 1.1	6,9
500	30	11,41	HTTPS 1.1	7
50	5	8,6	HTTP 2.0	5,7
500	5	12,36	HTTP 2.0	8,4

5. Análise dos resultados

Na Tabela 5 fica evidente a melhoria de desempenho de todos os servidores que usaram SPDY ou Http 2.0 no teste de 50 conexões. Nos teste com 500 conexões nenhum obteve melhora.

Tabela 5. Resultado geral dos testes

	HTTP 50	HTTP 500	HTTPS 50	HTTPS 500	SPDY/H2 50	SPDY/H2 500
Apache	13,85	11,62	13,34	13,03	9,72	14,47
Nginx	11,21	13,81	11,38	10,05	10,55	16,81
IIS	9,43	20,24	11,53	11,41	8,63	12,36
G-WAN	11,03	16,81	-	-	-	-

5.1. G-WAN

O G-WAN foi o segundo melhor colocado nos testes com 50 conexões ficando atrás apenas do Microsoft IIS. Com 500 conexões ficou em na penúltima colocação, considerando a média de tempo das requisições.

5.2. Nginx

Na Tabela 2, o SPDY utilizado com o Nginx se mostrou com bom desempenho, já que no teste de 50 requisições houve uma diferença na média de tempo das requisições de 7.3% utilizando o SPDY comparado com HTTPS, enquanto que no teste de quinhentas conexões houve um acréscimo de 40.2%.

5.3. Apache

Nos testes com 50 conexões o SPDY teve um ganho de 27,13% na média de tempo das requisições, enquanto que com 500 conexões houve uma perda de 9.95% comparando com HTTPS.

5.4. IIS

No Microsoft IIS O ganho foi de 25.41% na média de tempo das requisições no teste com 50 conexões e perda 7.68% com 500 conexões comparando o HTTP 2.0 com SPDY.

6. Considerações finais

A única situação em que o SPDY 3.1 ou HTTP 2.0 ficaram sempre na frente foi com os testes de 50 conexões. Nos testes com 500 conexões os resultados foram aleatórios. Devido aos resultados encontrados pode-se concluir que os módulos de ativação do protocolo SPDY/HTTP2 disponibilizados nos servidores testados ainda não se encontram otimizados. Entretanto, o protocolo SPDY/HTTP2 gerou um volume menor de dados trafegados e um número menor de conexões TCP em todos os testes.

6.1. Habilitação de módulos

As maiores dificuldades se deram na habilitação dos módulos para Apache e Nginx. Embora haja pacotes já disponíveis do SPDY para estas distribuições Linux, há alguns problemas de compilação não devidamente documentados. Foi necessário buscar informações em diversos sites e blogs de como habilitar os módulos.

6.2. IIS

O Windows 10 Server, talvez por ser uma versão beta ainda, tem problemas de instabilidade, gerando tela azul, e obrigando a uma segunda instalação durante os testes. Entretanto o processo de habilitação do HTTP2 é o mais fácil de todos, necessitando apenas uma modificação no registro.

6.3. Testes

Os testes primeiramente, foram feitos com apenas um navegador, onde era medido apenas o tempo de carregamento da página no browser. Os resultados se mostraram favoráveis ao SPDY e HTTP 2.0, porém essa medição inclue junto o tempo de renderização do navegador. Então optou-se por criar um teste de acessos simultâneo, de 5 máquinas virtuais, em 5 computadores diferentes, o que só foi possível com o uso de scripts em shell.

6.4. Ferramentas de captura

O Wireshark ainda não possui suporte total de filtragem do SPDY e HTTP2. Ele detecta no *frame*, o tipo de pacote, mas não filtra corretamente, por que ainda está em fase de implementação. Nos testes foi feita a filtragem por IP e porta de acesso para serem analisados os pacotes.

6.5. Análise de logs

Uma das dificuldades encontradas, foram as análises de desempenho através do log dos servidores de páginas. Acontece que no SPDY ou HTTP 2.0, utilizando o IIS e o Nginx, as conexões de elementos da página flags.html, como as bandeiras, ficam com valores zerados ou muito próximos de zero. O mesmo não ocorre no Apache, onde todas as conexões, sejam elas feitas como um *stream* do HTTP 2.0 ou não, são logadas corretamente.

6.6. Trabalhos Futuros

Quando o Wireshark tiver suporte completo a filtragem de HTTP 2.0, será possível fazer um teste mais apurado. Será interessante também testar, com outros navegadores, e servidores de páginas que já tem implementado o HTTP 2.0.

Referências

- Behlendorf, B. (1995). Apache webserver. Disponível em: <http://httpd.apache.org/ABOUT_APACHE.html>. Acesso em 15/05/2015.
- Belshe, M. (2015). Hypertext transfer protocol version 2 (http/2). Disponível em: <<https://tools.ietf.org/html/rfc7540>>. Acesso em: 22/05/2015.
- Chen, Y.-C. (2012). Characterizing 4g and 3g networks: Supporting mobility with multi-path tcp. Disponível em: <http://people.cs.umass.edu/yungchih/publication/12_mtcp_4g_tech_report.pdf>. Acesso em 18/05/2015.
- Fielding, R. (1999). Hypertext transfer protocol – http/1.1. Disponível em: <<https://www.ietf.org/rfc/rfc2616.txt>>. Acesso em 15/06/2015.
- Google_inc (2012). Spdy protocol. Disponível em: <<https://tools.ietf.org/html/draft-mbelshe-httplibbis-spdy-00>>. Acesso em 18/04/2015.
- Grigorik, I. (2013). High performance browser networking. Disponível em: <http://chimera.labs.oreilly.com/books/1230000000545/ch12.html#_brief_history_of_spdy_and_http_2>. Acesso em 18/06/2015.
- Liu, G. (2015). World country flags demo. Disponível em: <<https://h2ohttp2.centminmod.com/flags.html>>. Acesso em 18/05/2015.
- Microsoft (2015). Iis webserver. Disponível em: <<https://technet.microsoft.com/pt-br/library/hh831725.aspx>>. Acesso em 15/05/2015.
- Nazim, S. B. (2014). Http/2 for iis in windows 10 technical preview. Disponível em: <<http://blogs.iis.net/nazim/http-2-for-iis-in-windows-10-technical-preview>>. Acesso em 18/04/2015.
- NETCRAFT (2015). May 2015 web server survey. Disponível em: <<http://news.netcraft.com/archives/2015/05/19/may-2015-web-server-survey.html>>. Acesso em 18/05/2015.
- Sysoev, I. (2004). Nginx webserver. Disponível em: <<http://wiki.nginx.org/Main>>. Acesso em 15/05/2015.
- Trueleap (2007). G-wan webserver. Disponível em: <<http://gwan.ch/>>. Acesso em 15/05/2015.
- VMware (2015). Vmware workstation. Disponível em: <<http://www.vmware.com.br/products/workstation>>. Acesso em 30/03/2015.

Análise do comportamento de servidores web sob ataques causados por BotNets

Leandro Ferreira Canhada¹, Eduardo Maroñas Monks¹

¹Faculdade de Tecnologia Senac Pelotas- Fatec Senac Pelotas
Rua Gonçalves Chaves, 602 - Centro - Pelotas - RS, 96015-560 (053) 3225-6918

lfcanhada, emmonks@gmail.com

Abstract. This article aims to analyze the performance of web servers under denial of service attacks using botnets simulation tools. These attacks have caused substantial losses to companies who need to spend large sums to develop solutions and protection tools against these attacks.

Resumo. Este artigo tem o objetivo de analisar o desempenho de servidores web sob ataques de negação de serviço, com o uso de ferramentas de simulação de BotNets. Estes ataques tem causado grandes prejuízos à empresas, que necessitam gastar altas somas para desenvolver soluções e ferramentas de proteção contra esses ataques.

1. Introdução

Atualmente, um servidor Web é indispensável para qualquer empresa de pequeno, médio ou grande porte, que tenha um sistema desenvolvido para ser executado por meio de navegadores. Todo servidor web está suscetível a ataques, causando lentidão, ou até mesmo tornando-o indisponível, por algum tempo, ou de forma permanente, enquanto esse ataque Dos (*Denial of Service*) ainda estiver em execução. Na Figura 1, mostra estatísticas do aumento do número de incidentes reportados ao CERT.br [Freire 2015].



Figura 1. Aumento dos ataques ao longo dos anos.

Segundo está análise realizada pelo CERT.br no ano 2014 foram recebidas 1.047.031 notificações de incidentes relacionados a Internet, sendo, 223.935 ataques de negação de serviço a servidores web.

Sendo assim esse artigo visa um estudo de como os servidores Web se comportam sob ataque de negação de serviço originados de BotNets, entendendo o funcionamento e analisando o desempenho dos servidores mais utilizados, criando um comparativo entre eles.

2. BotNets

A definição de *BotNet*, consite em uma rede de computadores com a finalidade de executar algum programa ou comando, normalmente ataques de negação de serviço distribuído, podendo ser utilizado de forma benéfica, mas também de forma ilícita. A utilização ilícita desse tipo de rede, está ligada diretamente a ataques a servidores web, pois essa é a forma que os criminosos utilizam para tornar uma aplicação web lenta ou até mesmo inoperante.

Segundo a empresa TrendNet, [Rocha 2015] o Brasil é o quarto colocado entre a lista de países com a maior quantidade de servidores de controle de BotNets, como mostra a Figura 2.

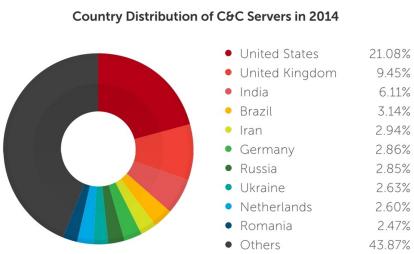


Figura 2. Ranking dos países que possuem servidores de controle de BotNets.

2.1. DoS (*Denial of Service*)

O DoS ou ataque de negação de serviço, tem como objetivo impedir que usuários utilizem algum serviço relacionado à Internet, tais como e-mails, sites de banco, comércio eletrônico, jogos e outros serviços, na medida que este ataque é bem sucedido, os usuários legítimos ficarão sem acesso e o serviço indisponível.

2.2. DDoS (*Distributed Denial of Service*)

O DDoS ou ataque de negação de serviço distribuído, consiste em atacar um servidor, utilizando vários computadores (*nodes*), comandados por um computador central, tornando um servidor, um ou mais serviços indisponíveis.

3. Tipos de Ataques e suas Ferramentas

Existem diversos tipos de ataques de negação de serviço e serão abordados os tipos mais utilizados.

3.1. HTTP Get Flood

Este ataque simula muitas solicitações para a mesma URL, testando a capacidade do servidor de processar as requisições HTTP GET. A ferramenta *BoNeSi* [Ulucan 2012] foi utilizada para realizar a simulação de um tráfego BotNet, desenvolvida para estudar os efeitos de ataques DDoS. Essa ferramenta gera pacotes ICMP, UDP e TCP (http), executando ataques de *flood* (inundação), sendo uma ferramenta de fácil configuração e ajustes de volume de tráfego.

3.2. HTTP Get

Este Teste envia requisições de conexão aos servidores web, simulando um alto tráfego de usuários acessando o servidor [W3.org 2015] A ferramenta para esse teste, foi o *httping* [Die.net 2015], para gerar conexões aos servidores web, utilizando o método GET.

3.3. TCP Syn Flood

O SYN flood ou ataque SYN é uma forma de ataque de negação de serviço (DoS) que consiste em enviar uma sequência de requisições SYN para um servidor [Incapsula.com 2015]. Foi usada a ferramenta *Hyenae* [Hyenae 2012] que permite reproduzir alguns cenários de ataque DoS e DDoS.

3.4. ICMP Flood

Este ataque consiste no envio massivo de pacotes ICMP. O *Hping3* [Tomicki 2015] possibilita a realização de ataques por ICMP flood.

4. Ferramentas de Análises

Algumas ferramentas foram utilizadas para o monitoramento e coletas dos dados necessários para realizar o comparativo entre servidores web. Os recursos analisados foram a largura de banda, o processamento e tráfego protocolo HTTP.

4.1. PRTG

A ferramenta PRTG [Paessler 2015], analisa o tráfego das interfaces de rede dos servidores web, mostrando o fluxo de entrada e saída do dados. Na Figura 3 é mostrado um exemplo de gráfico gerado pela ferramenta PRTG.



Figura 3. PRTG monitorando as três interfaces de redes dos servidores.

4.2. NtopNG

A ferramenta NtopNG[Ntop.org 2015] monitora a rede e os hosts e a quantidade de dados que foram trafegado nas interfaces de rede.

4.3. GANGLIA

A ferramenta Ganglia [Clusters 2008] é um sistema de monitoramento de clusters, e foi utilizada para monitorar o uso de recursos dos hosts e servidores.

5. Servidores Web

Segundo a Netcraft [Netcraft.com 2015] com dados de maio de 2015, os servidores mais utilizados na Internet foram Apache, Nginx e IIS, somando 80% do total de uso.

5.1. APACHE

O Apache Server [Apache.org 2015] é um software livre, é o mais conhecido e utilizado, possui vários módulos com recursos para uso conforme necessidade.

5.2. LIGHTTPD

O Lighttpd [Lighttpd.com 2015] é um servidor web seguro, rápido e muito flexível, projetado para ambientes de alto desempenho.

5.3. NGINX

O Nginx [Nginx.org 2015] é um servidor web rápido, leve, e com inúmeras possibilidades de configuração para melhor performance.

6. Cenário de simulação

Para fazer a simulação de ataques utilizando um BotNet, foram utilizadas algumas ferramentas em conjunto. Na Figura 4, o diagrama com os componentes da simulação dos ataques.

6.1. Cluster Beowulf

O cluster Beowulf [Neto 2008] foi utilizado para coordenar os ataques aos servidores web, utilizando o recurso MPI (*Message Passing Interface*) [Mpich.org 2015], para a troca de mensagens com os *nodes* do cluster.

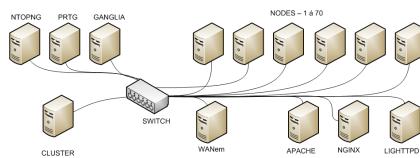


Figura 4. Organização dos servidores

6.2. VMware Player

O VMware Player 7 [Vmware.com 2015] foi utilizado para executar as máquinas virtuais, Na Tabela 1 são listadas as configurações das máquinas virtuais utilizadas.

Tabela 1. Configuração dos equipamentos utilizados

Máquina Virtual	Sistema Operacional	Cores	Mémoria Ram	Hard Disk
Cluster/Ganglia	Linux Debian 6	2	512MB	10G
Nodes	Linux Debian 6	1	256MB	10G
Servidores Web	Linux Debian 7	2	512MB	10G
NtopNG	Linux Debian 6	1	256MB	10G
PRTG	Windows 7	2	6GB	320G

Para realizar a simulação de uma rede BotNet, as máquinas virtuais utilizadas, possuem configuração baseada em serviços de hospedagem disponibilizado, na Internet, Na Tabela 2, é mostrada uma lista de configurações comuns de algumas empresas provedoras desses serviços de hospedagem.

Tabela 2. Configuração de máquinas virtuais comerciais

Empresa	Sistema Operacional	Cores	Mémoria Ram	Hard Disk
Donweb.com	SO à escolher	1 Core	1GB	10G
Hostweb.com	CentOS 6	1 Core	512MB	30G
Linode.com	SO à escolher	1 Core	256MB	24G

6.3. Conexão de Internet

Para simular uma conexão de Internet, foi utilizada uma ferramenta o WANem [WANem 2015], limitando a largura de banda entre os *nodes* e o servidor em 10MBit/s, de acordo com a pesquisa feita pela Anatel [G1.com 2015].

7. Comparativo entre os servidores WEB

Para realizar o comparativo entre os servidores, a instalação dos servidores web foi de forma padrão, sem alterações ou instalação de módulos para melhorias no desempenho.

Foi executado um teste nos servidores para definir um ambiente ideal, ou seja, carregar a URL, sem nenhum ataque ou restrição. Esse teste faz o carregamento da URL **relatorio.php**, na qual faz uma busca por todos usuários de um banco de dados Mysql. O código PHP está programado para atualizar a página de 5 em 5 segundos, calculando o tempo de carregamento. A coleta de dados para a formulação do comparativo é de 10 minutos, esse tempo foi definido, de modo que em 10 minutos o desempenho dos servidores já demonstrassem se houve alteração no desempenho, ou pararam de responder aos comandos.

Para fazer a simulação de um *BotNet* foram criados três cenários, um sem ataques, outro com 45 *nodes* atacantes e outro com 70 *nodes* atacantes.

7.1. Cenário sem Ataques

Nesse cenário, foi simulado um ambiente ideal, ou seja, os servidores não sofreram nenhuma interferência ou restrição, os resultados estão listados na Tabela 3.

Tabela 3. Sem ataques

Servidores	Tempo de carregamento (s)
Nginx	0,04
Apache	0,05
Lighttpd	0,09

7.2. Cenário de ataques com 45 nodes

Nas Tabelas 5, 6, 7, 9, 10, 11, mostram o tempo de carregamento com os ataques. Na Tabela 4, os tempos não são mostrados pois houve perda total de comunicação.

Tabela 4. Com Hping3

Servidores	Tempo de carregamento (s)
Nginx	-
Apache	-
Lighttpd	-

Tabela 5. Com ataques BoNeSi

Servidores	Tempo de carregamentos (s)
Nginx	20,3
Apache	26,4
Lighttpd	34,2

Tabela 6. Com Httping

Servidores	Tempo de carregamento (s)
Nginx	7,6
Apache	8,6
Lighttpd	10,3

Tabela 7. Com ataques Hyenae

Servidores	Tempo de carregamento (s)
Nginx	41,2
Apache	45,3
Lighttpd	69,4

7.3. Cenário de ataques com 70 nodes

Nesse cenário foram utilizados 70 nodes para simular os ataques. Nessa situação, o tempo de carregamento da URL dos servidores, aumentou proporcionalmente, como mostram as Tabelas 9, 10 e 11. Na Tabela 8, os tempos não são mostrados pois houve perda total de comunicação com os servidores.

Tabela 8. Com Hping3

Servidores	Média de carregamento (s)
Nginx	-
Apache	-
Lighttpd	-

Tabela 9. Com ataques Bonesi

Servidores	Média de carregamento (s)
Nginx	39,1
Apache	45,3
Lighttpd	63,2

Tabela 10. Com Httping

Servidores	Média de carregamento (s)
Nginx	17,4
Apache	19,1
Lighttpd	30,9

Tabela 11. Com ataques Hyenae

Servidores	Média de carregamento (s)
Nginx	52,6
Apache	63,8
Lighttpd	89,4

As Tabela 12 mostra um comparativo dos recursos dos utilizados em cada ataque, na coluna "PER", estão listadas as perdas de pacotes, na coluna "LAR", a largura de banda utilizada.

Tabela 12. Tabela Comparativa com utilização de recursos dos servidores.

Servidor	Bonesi - %			Hyenae - %			hping3 - %			httping - %		
	CPU	PER	LAR	CPU	PER	LAR	CPU	PER	LAR	CPU	PER	LAR
Nginx	20	40	100	2,79	5	2	1	100	100	6,1	0,01	100
Apache	25	46	100	3,35	8	2	1	100	100	4,2	0,01	100
Lighttpd	28	62	100	0,45	10	2	1	100	100	14,2	0,01	100

8. Análises dos Resultados

Com os ataques das ferramentas *Bonesi* e *Hyenae* houve várias perdas de comunicação com os servidores que foram detectadas com o ping e com a ferramenta PRTG, pois os três servidores deixaram de se comunicar por vários momentos, não enviando as informações das interface de rede necessárias para a coleta dos dados para análise.

Com a ferramenta *hping3* a comunicação com os servidores foi totalmente perdida, com a execução do comando, não sendo possível a coleta do tempo de carregamento da URL.

Já com o *httping*, foi definido a execução do cluster com o comando para todos nodes, utilizando configuração padrão, foi definido que fosse executado 10 minutos, dessa forma os servidores receberam requisições Http Get simultaneamente de todos os *nodes*. Com esse teste os três servidores conseguiram responder com sucesso todas as solicitações Http Get, mesmo utilizando 100% da largura de banda.

9. Conclusões

Foi comprovado que com o aumento de *nodes*, os danos aos servidores aumentam consideravelmente. Os três servidores web apresentaram muitas perdas de pacotes, com as ferramentas *Bonesi* e *Hyenae*, causando lentidão, serviços sem responder e páginas não encontradas. O Nginx, trabalhou melhor com várias conexões, se mostrando mais robusto, considerando, que em um ambiente real, os ataques podem ser bem mais maciços, pois o número de *nodes* pode ser elevado exponencialmente.

O Apache se mostrou um pouco abaixo do desempenho, em comparação ao Nginx, nos testes efetuados, houve resposta em todas solicitações, embora com um tempo de resposta maior.

O Lighttpd, com a execução das ferramentas BoNeSi e *Hyenae* parou de responder, antes do final dos 10 minutos somente voltando a responder, com o final dos testes.

De acordo com os teste realizados, o servidor Nginx apresentou os melhores resultados sob ataque simulados de BotNets, em comparação aos servidores Apache e Lighttpd.

Referências

- Apache.org (2015). Apache. <https://httpd.apache.org>. [Acesso em: 20-Junho-2015].
- Clusters, D. (2008). Ganglia: Installation. <http://www.debianclusters.net/index.php/Ganglia>. [Acesso em: Março-2015].
- Die.net (2015). httping - Linux man page. <http://linux.die.net/man/1/httping>. [Acesso em: Junho-2015].
- Freire, R. (2015). CERT.br registra aumento de ataques DoS, fraudes e phishing no Brasil. <http://www.techtudo.com.br/noticias/noticia/2015/04/certbr-registra-aumento-de-ataques-dos-fraudes-e-phishing-no-brasil.html>. [Acesso em: Março-2015].
- G1.com (2015). Banda larga no Brasil. <http://especiais.g1.globo.com/tecnologia/banda-larga-brasil/2015/>. [Acesso em: Junho-2015].

- Hyenae (2012). How to simulate a http get botnet ddos attack. <http://sourceforge.net/projects/hyenae/>. [Acesso em: Abril-2015].
- Incapsula.com (2015). Syn flood - ddos - incapsula. <https://www.incapsula.com/ddos/attack-glossary/syn-flood.html>. [Acesso em: Maio-2015].
- Lighttpd.com (2015). <http://redmine.lighttpd.net/projects/lighttpd/wiki>. [Acesso em: Junho-2015].
- Mpich.org (2015). Message passing interface. <http://www.mpich.org>. [Acesso em: Junho-2015].
- Netcraft.com (2015). Netcraft.com. <http://www.netcraft.com>. [Acesso em: 20-Junho-2015].
- Neto, J. L. C. (2008). Cluster Beowulf. <http://www.vivaolinux.com.br/artigo/Cluster-Beowulf>. [Acesso em: Fevereiro-2015].
- Nginx.org (2015). Nginx. <http://nginx.org>. [Acesso em: Junho-2015].
- Ntop.org (2015). NtopNG. <http://www.ntop.org/products/traffic-analysis/ntop/>. [Acesso em: Abril-2015].
- Paessler (2015). PRTG Network Monitor. <https://www.br.paessler.com/prtg>. [Acesso em: Abril-2015].
- Rocha, L. (2015). Brasil é o quarto país do mundo com mais servidores de controle de botnets. <http://www.tecmundo.com.br/seguranca/75570-brasil-quarto-pais-mundo-servidores-controle-botnets.htm>. [Acesso em: Junho-2015].
- Tomicki, L. (2015). Ping flood. <http://tomicki.net/ping.flooding.php>. [Acesso em: Junho-2015].
- Ulucan, C. (2012). How to simulate a http get botnet ddos attack. <http://cagdasulucan.blogspot.com.br/2012/12/how-to-simulate-http-get-botnet-ddos.html>. [Acesso em: Abril-2015].
- Vmware.com (2015). Vmware player. <http://www.vmware.com>. [Acesso em: Junho-2015].
- W3.org (2015). Http/1.1 method. <http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html#sec9.3>. [Acesso em: Abril-2015].
- WA Nem (2015). Wanem. <http://wanem.sourceforge.net>. [Acesso em: Maio-2015].

Proposição de um Método de Análise de Qualidade de Vídeo sem Referência Completa

Alessandro Marchetto¹, Ricardo Becker^{2,3}, Ricardo Balbinot⁴

^{1,2}Universidade de Caxias do Sul
Caixa Postal 32 – 95700000 – Bento Gonçalves – RS – Brasil

³Faculdade SENAI de Tecnologia
91140000 – Porto Alegre – RS – Brasil

⁴IFRS - Instituto Federal do Rio Grande do Sul
92412-240 - Canoas - RS - Brasil

{amarchetto90, ricardobecker.eng}@gmail.com, ricardo.balbinot@canoas.ifrs.edu.br

Abstract. This paper proposes a method of objective video quality assessment, to analyze the impact of packet loss on measured quality, without full reference. First, for comparison purposes, tests were conducted using full reference. The method consists of inserting an extra frame, every defined interval, which is subsequently used to assist in measuring the quality. The extra frames are transmitted along with the other frames of the video. After comparing the result with the proposed method and full reference, the error is calculated. From the error, a graphic adjustment constant was calculated, so that the results from the method were closer than the results using full reference. Subsequently, the error was recalculated, resulting in a significantly lower value than before the adjustment.

Resumo. Este trabalho propõe um método de análise objetiva de qualidade de vídeo, para avaliar qual o impacto do descarte de pacotes sobre a qualidade medida, sem a utilização de referência completa. Primeiramente, para fins de comparação, foram realizados testes utilizando referência completa. O método consiste na inserção de um frame extra, a cada intervalo definido, que é posteriormente utilizado para auxiliar na medição da qualidade. Os frames extras são transmitidos juntamente com os demais frames do vídeo. Após comparar o resultado do método proposto com o resultado em referência completa, o erro é calculado. A partir do erro, foi calculada uma constante de ajuste dos gráficos, para que o método proposto se aproxime mais dos resultados em referência completa. Posteriormente, o erro foi recalculado, resultando em um valor significativamente menor do que antes do ajuste.

1. Introdução

De acordo com [Tao et. al. 2007], estudos mostram a transmissão de vídeo como um elemento crescente no tráfego da Internet. Provedores de vídeo estão adotando redes IP como seu veículo para entrega, visto IPTV.

A qualidade do vídeo é afetada em conjunto por vários fatores dependentes da rede e de fatores específicos da aplicação. Por exemplo, perda de pacotes e jitter são os principais fatores que dependem da rede, enquanto o codec de vídeo, a perda técnica de recuperação, a taxa de bits de codificação, o esquema de empacotamento, e

características de conteúdo são os principais fatores específicos da aplicação que afetam a qualidade de vídeo e sua sensibilidade ao erro de rede [Tao et. al. 2007].

Dado o crescimento da transmissão de vídeos via rede IP, este trabalho visa analisar a influência da perda de pacotes, um fator que depende da rede, na qualidade percebida do vídeo, analisando a mesma de forma objetiva, através da métrica PSNR[Chikkerur et. al. 2011] [Seshadrinathan et. al. 2010].

A análise da qualidade medida é feita sem a utilização de referência completa, onde a ideia é inserir um *frame* extra, para que a qualidade possa ser aferida. O processo de descarte de pacotes é baseado em um modelo de processo de Poisson, onde cada *frame* é processado individualmente, para diferentes taxas de perda.

Após obter os resultados pelo método proposto e através de referência completa, foi calculado o erro entre os resultados, para ajustar os valores do método proposto, para que fiquem mais próximos dos medidos com referência completa.

Na sequência deste trabalho são apresentadas na seção dois as principais referências relacionadas. Na seção três é apresentada a metodologia aplicada e nas seções quatro e cinco os resultados e conclusões, respectivamente.

2. Referencial Teórico

Nesta seção serão apresentados conteúdos pertinentes ao desenvolvimento do trabalho. Na subseção 2.1 é apresentada a teoria de descarte de pacotes. Nas seções 2.2 e 2.3 são abordados os temas de qualidade de vídeo e de análise de qualidade de vídeo, respectivamente.

2.1. Modelo de descarte de pacotes

Segundo [Jain 1991], processos de Poisson são usados em modelos de filas, os quais, por sua vez, são a base para explicar descarte em redes de pacotes. Eles são apropriados se os dados chegam a partir de um grande número de fontes independentes. Esses processos de chegada são chamados processos de Poisson ou fluxos de Poisson. Os fluxos de Poisson são populares nas teorias de filas, pois as chegadas são sem memória, enquanto o tempo entre chegadas é distribuído exponencialmente.

O valor esperado de uma variável aleatória distribuída através de Poisson é igual a média da distribuição de Poisson, e deve ser um número real ($\lambda > 0$) [Jain 1991]. Segundo [Johnson et. al. 1993], o coeficiente de variação é dado por $\lambda^{-1/2}$, enquanto o índice de dispersão é igual a 1.

De acordo com [Yates et. al. 2004], uma variável aleatória discreta X é dita ter uma distribuição de Poisson, se a função de massa de probabilidade de X é dada pela equação (1), onde e é o número de Euler, λ é a média da distribuição de Poisson, k é gama de valores ocorridos e $k!$ é o fatorial de k.

$$f(k; \lambda) = \Pr(X = k) = \frac{\lambda^k * e^{-\lambda}}{k!} \quad (1)$$

De acordo com [Knuth 1997], distribuições exponenciais são números que representam situações de tempo de chegada, e a distribuição é dada pela equação (2), onde μ é a média. Pode-se usar a distribuição exponencial para estimar o instante de ocorrência de um evento em um processo de Poisson.

$$F(x) = 1 - e^{-\frac{x}{\mu}} \quad (2)$$

O método de [Knuth 1997] é uma forma de implementar a função cumulativa de probabilidade da distribuição exponencial. Pelo método, se a equação (3) é verdadeira, então o mesmo vale para a (4). Assim, conclui-se que a equação (5) tem distribuição exponencial. Como $1-U$ é uniformemente distribuído quando U também é, pode-se concluir que a equação (6) é distribuída exponencialmente com média μ . Esse método requer menor precisão para estimar o instante de ocorrência dos eventos no processo.

$$y = F(x) = 1 - e^{-\frac{x}{\mu}} \quad (3)$$

$$\xi = F^{-1}(y) = -\mu * \ln(1 - y) \quad (4)$$

$$-\mu * \ln(1 - U) \quad (5)$$

$$X = -\mu * \ln(U) \quad (6)$$

2.2. Qualidade de Vídeo

Segundo [Winkler 2007], qualidade de vídeo tem muitos aspectos, e a avaliação da qualidade de sistemas de vídeo digital se tornou algo complexo. Isso se deve a:

- *Sistemas de vídeos consistem de vários componentes que afetando a qualidade do vídeo de alguma maneira: hardware de captura e reprodução; codecs e rede.*
- *A percepção visual é ainda mais complexa. Para medir a qualidade de uma forma precisa, como as pessoas percebem o vídeo e sua qualidade.*

De acordo com [Winkler 2007], vários fatores contribuem para o que um observador percebe como qualidade de vídeo. Elas incluem interesses pessoais, expectativas de qualidade, tipo de *display* e suas propriedades, condições de visualização, entre outras. As métricas utilizadas focam na fidelidade visual do vídeo em termos de distorções causadas pelo processamento, ao invés de fatores subjetivos.

2.3. Técnicas de avaliação de Qualidade de Vídeo

2.3.1 PeakSignal-to-NoiseRatio - PSNR

Para [Huynh-thu et. al. 2008], a relação sinal-ruído de picoé usada em sistemas analógicos como uma métrica de qualidade. No entanto, a tecnologia de vídeo digital expôs algumas limitações desta, mesmo assim, devido à sua baixa complexidade, ainda é utilizada como métrica para avaliar algoritmos de processamento de imagem.

Para cada sequência de vídeo, o PSNR é medido pela equação (7), onde $\varepsilon(i)$ é o erro médio quadrático da luminância do pixel, correspondente ao quadro i no vídeo de referência e compactada, e N é o número de quadros no vídeo degradado.

$$PSNR = \frac{1}{N} \sum_{i=1}^N 10 * \log_{10} \frac{255^2}{\varepsilon(i)} \quad (7)$$

3. Metodologia

3.1. Dataset

A database utilizada, obtida do IVP Laboratory[IVP 2014], possui vídeos em alta definição (1920x1088), filmados *RAW*, a 25 *frames/s* sem compressão, que são utilizados como referência. Antes de gerar distorção nos vídeos, eles foram convertidos de *RAW* para YUV 4:2:0 e após para RGB. Cada vídeo tem 10 segundos, sem áudio. O vídeo utilizado como referência foi a sequência *laser*, escolhido entre os disponíveis.

3.2. Modelo de Descarte de Pacotes

O processo consiste em simular o processo de fila, encontrado em redes IP, para avaliar qual o impacto destas perdas na qualidade do vídeo. Para tanto, segundo [Mansfield et. al. 2010], perdas entre 5% e 10% do total de pacotes enviados tem um impacto significante na qualidade de serviço. Para [Zennaro et. al. 2015], menos de 1% de perda de pacotes é “bom” para áudio e vídeo *streaming*, e entre 1% e 2,5% são aceitáveis.

Neste trabalho as perdas foram entre 0% e 10%, considerando [Mansfield et. al. 2010] e [Zennaro et. al. 2015]. Os intervalos na faixa de simulação variam de 0,2%.

O modelo de descarte, baseado no processo de Poisson, foi implementado em *Matlab*. Foi utilizada a equação (8), que expressa quantos pacotes após o atual se dará o descarte. λ é a média esperada do número total de descartes para a população. Começando do primeiro pacote, uma variável auxiliar é utilizada para controlar quais pacotes serão perdidos. Os pacotes perdidos são multiplicados por zero, cor preta. As figuras 1 e 2 ilustram o primeiro *frame* da sequência, com perda de 1% e 10%, respectivamente. O processo é repetido para todos os *frames* do vídeo, para todos os níveis de perda estabelecidos.

$$Prox = \text{round} \left(\frac{-\ln(1-\text{rand}(1))}{\lambda} \right) \quad (8)$$



Figura 1. Primeiro *frame* da sequência com perda de 1%, em escalas de cinza.

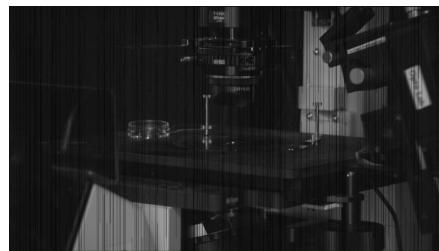


Figura 2. Primeiro *frame* da sequência com perda de 10%, em escalas de cinza.

3.3. Método Proposto

O método proposto consiste em inserir um *frame* extra, que será enviado para o receptor juntocom os demais *frames* do vídeo. Por exemplo, com o intervalo em 5, um *frame* extra, que é uma cópia do *frame* anterior, é enviado a cada 5 *frames* originais, a fim de

comparação do *frameextra* com o seu original. Como o método utiliza uma referência limitada, primeiramente foram realizados testes utilizando referência completa.

Os intervalos foram definidos como um *frame extra* a cada 2, 4, 6, 8 e 10 frames. O *frame extra* também passou pelo modelo de descarte de pacotes, para que seja feita uma medida de qualidade.

4. Resultados

4.1. Testes utilizando referência completa

A figura 3 representa os valores PSNR medidos para a sequência *laserem* em relação aos níveis de descarte de pacotes mencionados. Os valores foram obtidos através de referência completa.

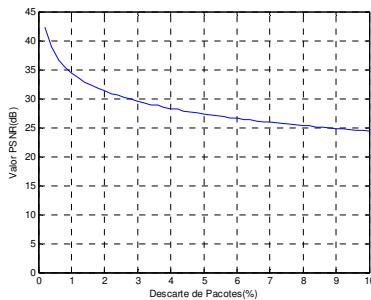


Figura 3. Valores PSNR utilizando referência completa.

4.2.2 Métrica PSNR

Os valores PSNR, figura 4, não apresentaram variação significativa entre si, determinando um intervalo de 10 frames. O intervalo escolhido gera menor volume de dados.

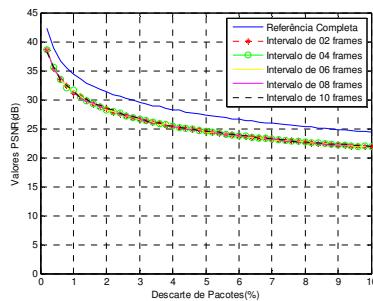


Figura 4. Comparação dos valores PSNR.

O erro RMS, equação (10), e a sua representatividade, equação (11), são mostrados na tabela 1. E_{rms} é o valor do erro RMS e max (PSNR) é o valor máximo medido pela métrica PSNR para determinado intervalo.

$$E_{RMS} = \sqrt{\frac{\sum_{i=1}^{i=n} (y(i) - y'(i))^2}{n}} \quad (10)$$

$$R\% = \frac{E_{rms}}{\max(PSNR)} \quad (11)$$

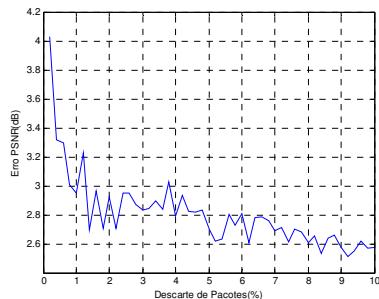
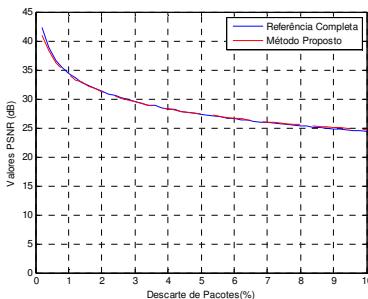
Tabela 1. Erro e representatividade para a métrica PSNR.

Intervalo (frames)	Erro RMS (dB)	Representatividade (%)
2	2.8482	6.7401
4	2.8521	6.7493
6	2.8713	6.7948
8	2.8835	6.8236
10	2.8190	6.6710

4.3. Ajuste dos Gráficos

Foi feito um ajuste dos gráficos para aproximar dos valores da referência completa. Para a PSNR, como as curvas se assemelham, ajustando a amplitude, o erro foi extraído através de uma subtração da curva dos resultados do método proposto, no intervalo de 10 frames, da curva em referência completa (figura 5). Para tanto, foi feita uma média pela equação (12), para se obter uma aproximação com menor valor de erro. Assim, aplicando a equação (12) nos dados da figura 5, foi obtido um erro médio de 2.8077 dB. Após aplicar a equação, o resultado foi somado aos valores do método proposto. A figura 6 representa o valor corrigido.

$$E = \frac{\sum_{i=1}^{i=n} e(i)}{n} \quad (12)$$

**Figura 5. Erro entre os valores PSNR.****Figura 6. Valores corrigidos para um intervalo de 10 frames.**

Após corrigir os valores dos gráficos, para que os mesmos se aproximem mais dos valores de referência completa, o erro RMS e a sua representatividade foram recalculados pelas equações (10) e (11). Os valores obtidos estão expostos na tabela 2.

Para validar o método proposto, o mesmo foi aplicado para as sequências de vídeo *tube* e *tractor*, da mesma *database*. As figuras 7 e 8 representam os resultados.

Ao analisar os gráficos da figura 7 e 8, se comprova o comportamento obtido para a sequência *laser*. Para ajustar ambos os gráficos, foi utilizado o mesmo valor de erro obtido para a sequência *laser*. As figuras 9 e 10 representam os gráficos ajustados.

A tabela 3 representa os valores de erro e a representatividade do erro em relação ao máximo valor medido, de acordo com as equações (10) e (11).

Tabela 2. Representatividade dos valores de erro ajustado para a métrica PSNR, para a sequência *laser*.

Intervalo	Erro RMS (dB)	Representatividade(%)
2	0.2157	0.5104
4	0.2212	0.5235
6	0.2257	0.5341
8	0.2222	0.5258
10	0.2521	0.5966

Tabela 3. Erro RMS e Representatividade para os valores corrigidos das sequências *tube* e *tractor*.

Intervalo	Erro RMS (dB)		Representatividade	
	<i>tube</i>	<i>tractor</i>	<i>tube</i>	<i>tractor</i>
2	0.1589	0.1629	0.4715	0.4628
4	0.1670	0.1931	0.4955	0.5486
6	0.1525	0.1995	0.4523	0.5667
8	0.2010	0.2449	0.5962	0.6959
10	0.1872	0.1751	0.5554	0.4974

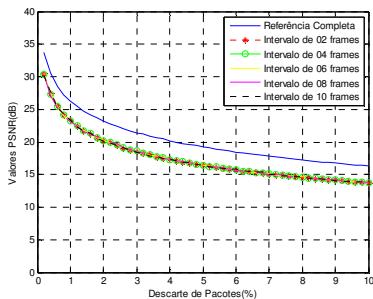


Figura 7. Valores PSNR para a sequência *tube*.

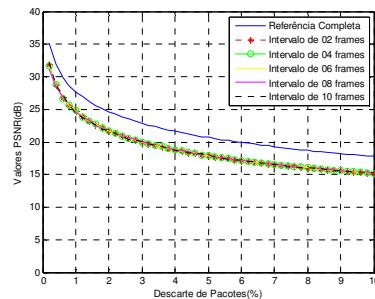


Figura 8. Valores PSNR para a sequência *tractor*.

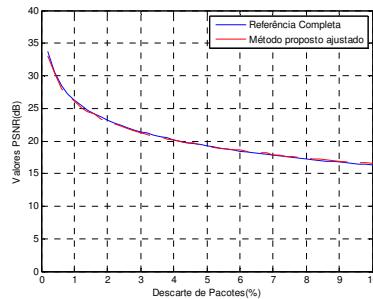


Figura 9. Valores corrigidos para um intervalo de 10 frames, para a sequência *tube*.

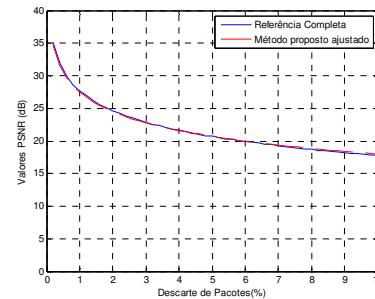


Figura 10. Valores corrigidos para um intervalo de 10 frames, para a sequência *tractor*.

5. Conclusão

Este trabalho propôs um método para avaliar a qualidade de vídeos enviados por uma rede IP, com a inserção de um *frame* extra, o qual é inserido em intervalos e utilizado para a comparação e medição da qualidade de maneira objetiva. Tal método insere

frames extras durante a transmissão do vídeo. Tais informações podem ser adequadas para gerar a menor sobrecarga possível da rede, porém, ao adequar os valores, também se diminui o número de amostras utilizadas para medir a qualidade do vídeo.

A métrica PSNR se mostrou resultados ajustáveis através de uma constante. Mesmo com a variação na amplitude dos resultados, pode-se considerar o erro “constante”, utilizando o mesmo valor de erro para ajustar as medições. Haja visto que, o valor do erro RMS apresentou-se abaixo de 1% entre os valores de referência completa e os valores do proposto.

A bibliografia utilizada não apresenta implementação como o método proposto, de maneira que são escassos os dados para comparação. Neste sentido, o trabalho mostra potencial para continuidade, vista a necessidade de massificação dos testes com variações diversas dos parâmetros apresentados.

Referências

- CHIKKERUR, S.; S., V.; R., M.; K., L. J. (2011) “Objective Video Quality Assessment Methods: A Classification, Review, and Performance Comparison.”*Broadcasting, IEEE Transactions on*: 165 – 182.
- INTERNATIONAL TELECOMMUNICATION UNION.(2012) ITU-R BT.500-13.“Methodology for the subjective assessment of the quality of television pictures.”
- IVP Laboratorydatabase (2014).Disponível em: <<http://ivp.ee.cuhk.edu.hk/research/database/subjective/index.shtml>> Acessado em: 02 outubro 2014.
- JAIN, R. (1991) “Art of Computer Systems Performance Analysis Techniques for Experimental Design Measurements Simulation and Modeling.”New YorkJ. W.
- JOHNSON, N. L.; K., S.; K., A. W. (1993) “Univariate Discrete Distributions.”2. Ed. New York: John Wiley & Sons.
- KNUTH, D. (1997) “The art of comptuer programming.”3. Ed. Addison-Wesley Professional.”
- MANSFIELD, K. C.; A., J. L. (2010) “Computer Networking from LANs to WANs: Hardware, Software, and Security.” Boston: Course Technology, C. L.
- SESHADRINATHAN, K.; S, R.; B., A. C.; C., L. K. (2010) “Study of Subjective and Objective Quality Assessment of Video.”*IEEE Transactions on Image Processing*.
- TAO, S.; A. J.;G., R. (2007)“Real-time monitoring of video quality in IP networks.”*IEEE/ACM Transactions on Networking*.
- WINKLER, S. (2007) “Video quality and beyond.”*15th European Signal Processing Conference, IEEE*: 150 - 153
- YATES, R. D.; G., D. (2004) “Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers.”2.Ed. New York: J. W.
- ZENNARO, M.; C., E. S., K. R.; R., A. A.; C., K. R. (2006) “Scientific Measure of Africa’s Connectivity”.*Information Technologies and International Dev.*: 55-64

Problemas e Soluções no Desenvolvimento de Componentes para o NS3: o Caso do DHCP

Andrey Blazejuk¹, Alexander Silva de Souza¹, Sérgio Luis Cechin¹

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{ablazejuk, asouza, cechin}@inf.ufrgs.br

Abstract. *Network simulations enable scientific research to create topologies that have similar behavior to real environments. Therefore, it allows researchers to develop new communication protocols, or to enhance protocols that are already consolidated in the area of computer networks. In this paper, strategies used by NS3 to model communication protocols and its applications in DHCP protocol implementation are shown. As a result, protocol implementation alternatives on NS3 will be discussed.*

Resumo. *A simulação de redes através de software possibilita a pesquisa científica com a reprodução de topologias que apresentem comportamentos semelhantes aos de um ambiente real. Dessa forma, permite que o pesquisador possa desenvolver novos protocolos de comunicação ou modelar protocolos já consolidados. Neste artigo são apresentadas as estratégias empregadas pelo NS3 para a modelagem de protocolos de comunicação e a aplicação das mesmas na implementação do protocolo DHCP. Como resultado serão discutidas as alternativas de implementação de protocolos no NS3.*

1. Introdução

O desenvolvimento de protocolos e sistemas para a área de redes requer especificações claras e implementações que possam ser verificadas em relação àquelas especificações. Essas implementações são complexas e de difícil depuração, considerando a sofisticação que as redes atingiram atualmente. Nesse cenário de alto custo de desenvolvimento e de verificação dessas implementações, tem-se utilizado com frequência a técnica de simulação. Essa técnica permite prever o comportamento de sistemas quando estes ainda estão indisponíveis, o que é comum durante o desenvolvimento. Mesmo quando o sistema está disponível, a simulação pode ser preferível por permitir a avaliação numa ampla gama de cargas de trabalho e ambientes [Jain 1991].

O projeto SDCN (*Software Defined Carrier Network*), uma parceria entre o Instituto de Informática da UFRGS e a Parks S/A Comunicações Digitais, utiliza o simulador NS3 (*Network Simulator 3*) [NS3 2015] para analisar o funcionamento de redes definidas em software. Apesar dos recursos oferecidos pela ferramenta, é comum não encontrar algum protocolo necessário. Isso acontece porque esses recursos são desenvolvidos segundo a demanda da comunidade de usuários, que podem não coincidir com os requisitos do projeto.

Nesse artigo serão apresentadas as dificuldade e discutidas as soluções adotadas para criar componentes no simulador NS3. Especificamente será discutido o desenvol-

vimento necessário para utilizar o protocolo DHCP (*Dynamic Host Configuration Protocol*) [Droms 1997]. Uma solução para a integração protocolo DHCP ao NS3 foi apresentada por Radu Lupu [Lupu 2011]. Essa implementação possui simplificações e limitações incompatíveis com os objetivos do projeto SDCN, o que motivou a criação de novos componentes. Entre os problemas existentes, destacam-se três limitações fundamentais para este artigo:

Limitação 1 todas as mensagens DHCP mal formadas, que não atendem ao formato especificado pelo padrão por serem serializadas incorretamente, tornando o componente incompatível com outros modelos;

Limitação 2 máquina de estados reduzida, gerando menor tráfego de mensagens;

Limitação 3 artifícios empregados para contornar limitações do NS3, também gerando problemas de compatibilidade com outros componentes.

O artigo está organizado da seguinte forma. A Seção 2 apresenta o simulador, justificando sua escolha para o projeto SDCN. Na Seção 3, os aspectos relevantes do protocolo DHCP para a simulação são revistos. O modelo desenvolvido com os recursos fornecidos pelo simulador é apresentado na Seção 4, onde são enumeradas as dificuldades enfrentadas e as soluções empregadas. A Seção 5 mostra a validação dos componentes, verificando a correta operação do protocolo. Finalmente, a Seção 6 conclui o artigo, sugerindo a abordagem proposta como forma de desenvolvimento de novos componentes.

2. Apresentação do simulador

O NS3 é um simulador de redes usado principalmente para pesquisa e uso educacional. Encoraja a contribuição da comunidade científica ao possuir código aberto, documentação gerada a partir do código, facilidade de depuração e de análise e extração de resultados. A equipe de desenvolvimento do NS3 gera melhorias e correções de defeitos a cada três meses, diretamente no sítio da ferramenta.

A ferramenta é um simulador de eventos discretos, cujas descrições são feitas em linguagem C++ ou Python. Caso seja desejado, também possui um agendador de tarefas em tempo contínuo que permite a integração dos modelos simulados com equipamentos de rede reais.

O simulador define classes que representam nodos, dispositivos de rede, canais, aplicações e assistentes de topologia (chamados de *helpers*). Nodos podem ser vistos como computadores, aos quais funcionalidades são adicionadas. Dispositivos de rede são instalados em nodos para permitir a comunicação através dos canais. Canais de comunicação conectam dispositivos de rede, permitindo a definição de características tais como a taxa de transferência de dados e o atraso. Aplicações modelam funcionalidades que executam nos nodos. Os *helpers* permitem a simplificação de tarefas rotineiras durante a definição de descrições de simulação.

3. Funcionamento do protocolo DHCP

O protocolo DHCP é baseado no protocolo BOOTP (*Bootstrap Protocol*) [Croft and Gilmore 1985], acrescentando a capacidade de alocação automática de endereços reusáveis da rede e outras opções de configuração. Segue o modelo cliente-servidor, onde os servidores alocam endereços disponíveis da rede e configuram dinamicamente os clientes. O protocolo de transporte utilizado é o UDP. A porta 67 do

servidor é reservada para receber as mensagens do cliente e os pacotes enviados pelo servidor são recebidos na porta 68 do cliente.

A concessão de um endereço IP para um cliente DHCP ocorre da seguinte maneira:

1. O cliente envia uma mensagem do tipo DHCPDISCOVER em *broadcast* para encontrar servidores disponíveis.
2. Ao receber o pacote, o servidor reserva um endereço IP que ainda não esteja alocado na rede e oferece o mesmo ao cliente através de uma mensagem DHCPOFFER, que pode ser em *broadcast* ou *unicast*, de acordo com o valor do campo *BOOTP flags* da mensagem recebida anteriormente.
3. A resposta do cliente vem com a mensagem de DHCPREQUEST, onde o endereço oferecido em DHCPOFFER é aceito e solicitado. O pacote é enviado em *broadcast*.
4. Para confirmar e concluir o processo, a mensagem DHCPACK é enviada ao cliente, informando o tempo de duração da concessão. Assim como DHCPOFFER, este pacote pode ser enviado em *broadcast* ou *unicast*.

Para renovar seu endereço atual o cliente envia um pacote DHCPOFFER em *unicast* para o servidor, que responde com um DHCPACK. A partir deste ponto ambas as partes consideram o tempo da concessão estendido.

4. Modelagem do protocolo no simulador

A Limitação 1 apresentada na Introdução é causada pelas rotinas de serialização do cabeçalho DHCP, que não respeitavam o formato definido no padrão do IETF. O projeto SDCN requer que esse componente interaja com outros módulos, que esperam mensagens bem formadas. Esta característica também era desejada para permitir que os pacotes gerados sejam validados, utilizando softwares como o Wireshark [Foundation 2015].

A solução adotada por este artigo preenche todos os campos obrigatórios e acrescenta a opção *DHCP Message Type*, utilizada para determinar em qual etapa o processo de concessão se encontra. As duas mensagens enviadas pelo cliente contém a opção *Requested IP Address*, usada para que o cliente solicite determinado endereço IP ao servidor. Adicionalmente, DHCPREQUEST utiliza a opção *DHCP Server Identifier* para especificar de qual servidor DHCP o endereço oferecido é solicitado. Para ambas mensagens enviadas pelo servidor foram escolhidas as opções: *Subnet Mask* (máscara da rede parametrizada através do *helper* do servidor), *Renewal Time Value* (tempo até que o cliente envie a mensagem de DHCPOFFER para renovar o empréstimo de seu endereço), *IP Address Lease Time* (tempo total, a partir da concessão, até que o IP se torne inválido) e *DHCP Server Identifier* (informando ao cliente qual o servidor que realizou o empréstimo). A opção *End* marca o fim da lista de opções DHCP em todas as mensagens.

O segundo problema, a máquina de estados muito simplificada (Limitação 2), provocava uma redução do número de mensagens trocadas durante a configuração de endereços IP. O processo de atribuição era feito apenas com o envio de uma mensagem DHCPDISCOVER e de sua resposta DHCPOFFER, sem utilizar DHCPREQUEST e DHCPACK. A renovação do empréstimo do IP também era falha, pois não utilizava uma mensagem de confirmação DHCPACK. Uma das métricas avaliadas no projeto SDCN é o tráfego de protocolos de controle, onde essas simplificações tem um impacto direto.

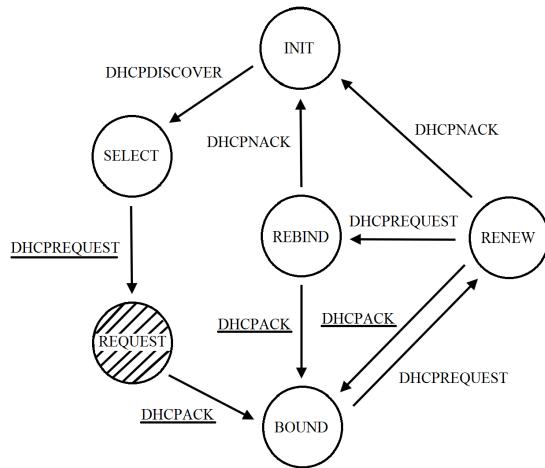


Figura 1. Máquina de estados do cliente DHCP. O estado hachurado e as mensagens sublinhadas foram adicionados pelo novo modelo.

O componente desenvolvido para o projeto implementa uma máquina de estados mais próxima daquela prevista no protocolo (figura 1), utilizando as quatro mensagens. Alguns aspectos descritos na especificação do DHCP foram explicitamente ignorados, pois não interferem com os objetivos do projeto. Entre as abstrações feitas estão:

- O envio de pacotes do servidor para o cliente é sempre em *unicast*, mesmo quando o mesmo solicita o envio em *broadcast*.
- Apenas as opções fundamentais para a comunicação entre as partes são processadas em cada aplicação.
- Os tempos de concessão (4 segundos), renovação (metade do tempo de concessão) e retransmissão de pacotes (5 segundos) foram definidos como constantes para ambas aplicações.
- As mensagens DHCPNAK, DHCPDECLINE, DHCPRELEASE e DHCPINFORM não são utilizadas.

Finalmente, a solução de Lpu necessita de artifícios não usuais para contornar uma premissa do NS3, como descrito pela Limitação 3. O simulador assume que toda interface de rede compatível com o protocolo IP terá um endereço definido antes da simulação iniciar. Essa limitação é incompatível com o objetivo do projeto de avaliar o funcionamento de protocolos de gerência de rede, que podem alterar a configuração da rede dinamicamente. Lpu requer que atribua-se endereços provisórios a todas as interfaces, o que polui a tabela de roteamento do simulador com prefixos de rede inexistentes. A nova solução tirou proveito da natureza de software livre do NS3, e modificou o próprio simulador para remover essa premissa, evitando o uso de artifícios não intuitivos na descrição da topologia.

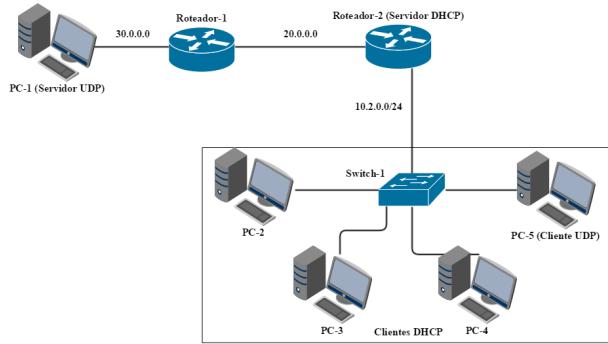


Figura 2. Topologia de teste.

5. Resultados obtidos

Para validar o módulo desenvolvido foi utilizada a topologia apresentada na figura 2. Nessa topologia é possível observar como esse módulo pode ser combinado com outros componentes disponíveis no NS3. Por exemplo, um método para gerar tabelas de roteamento atualizadas durante a simulação, uma aplicação que envia pacotes UDP e outra que os recebe (executadas no PC-5 e no PC-1, respectivamente).

O simulador permite a captura do tráfego de pacotes dos dispositivos de rede selecionados, gerando arquivos com o formato PCAP. As limitações do modelo de LUPU impediam uma validação externa do traço de execução devido ao formato incorreto das mensagens. O modelo criado com cabeçalhos adaptados para não só permitir a troca de informações entre as aplicações, mas também atender o formato previsto pelo IETF, gera um traço de execução que foi reconhecido com sucesso pelo software Wireshark, como mostrado na figura 3.

Na nova implementação a máquina de estados do protocolo utiliza as quatro principais mensagens do protocolo para configuração do cliente. Em destaque na figura 3, é possível observar as mensagens que estavam ausentes na implementação de LUPU, e que impactavam na análise do tráfego de gerenciamento no projeto SDCN.

Finalmente, antes das alterações realizadas no NS3 exigia-se que roteadores sempre tivessem endereço IP definido. O descumprimento dessa premissa gerava um colapso do software durante a execução de seu processo de resolução de rotas. Com a nova implementação é possível que uma interface com capacidade IP não possua nenhum endereço configurado, permitindo a modelagem de uma gama de protocolos de configuração dinâmica.

6. Conclusão

O modelo criado fornece um módulo para o simulador NS3 que pode ser usado em topologias que utilizem o protocolo DHCP. Suas características, em relação ao que existia anteriormente, permitem representar melhor o funcionamento de dispositivos reais que suportam o protocolo para que o comportamento das simulações realizadas seja mais semelhante ao de uma rede física.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	DHCP Discover - Transaction ID 0x67458b6b
2	0.000062	10.2.0.1	10.2.0.254	DHCP	DHCP Offer - Transaction ID 0x67458b6b
3	0.000062	0.0.0.0	255.255.255.255	DHCP	DHCP Request - Transaction ID 0xc6237b32
4	0.000127	10.2.0.1	10.2.0.254	DHCP	DHCP ACK - Transaction ID 0xc6237b32
7	1.004024	10.2.0.254	30.0.0.2	UDP	Source port: 49153 Destination port: 10
10	1.018980	30.0.0.2	10.2.0.254	UDP	Source port: 10 Destination port: 49153
11	1.500000	10.2.0.254	30.0.0.2	UDP	Source port: 49153 Destination port: 10
12	1.514929	30.0.0.2	10.2.0.254	UDP	Source port: 10 Destination port: 49153
13	2.000000	10.2.0.254	30.0.0.2	UDP	Source port: 49153 Destination port: 10
14	2.000127	10.2.0.254	10.2.0.1	DHCP	DHCP Request - Transaction ID 0x69983c64
15	2.000190	10.2.0.1	10.2.0.254	DHCP	DHCP ACK - Transaction ID 0x69983c64
16	2.014929	30.0.0.2	10.2.0.254	UDP	Source port: 10 Destination port: 49153
17	2.500000	10.2.0.254	30.0.0.2	UDP	Source port: 49153 Destination port: 10
18	2.514929	30.0.0.2	10.2.0.254	UDP	Source port: 10 Destination port: 49153
19	3.000000	10.2.0.254	30.0.0.2	UDP	Source port: 49153 Destination port: 10
20	3.014929	30.0.0.2	10.2.0.254	UDP	Source port: 10 Destination port: 49153

Figura 3. Traço de execução da simulação.

O trabalho apresentado serve como referência para a modelagem de outros protocolos, que podem ser implementados em qualquer simulador de redes, de acordo com a preferência ou necessidade do desenvolvedor. A partir do módulo criado e do resultado final, pode-se deduzir que o processo descrito foi efetivo e tem sua utilidade para a pesquisa, ilustrando o comportamento de redes reais através de simulações.

Referências

- Croft, W. and Gilmore, J. (1985). Bootstrap protocol. RFC 951, RFC Editor. <http://www.rfc-editor.org/rfc/rfc951.txt>.
- Droms, R. (1997). Dynamic host configuration protocol. RFC 2131, RFC Editor. <http://www.rfc-editor.org/rfc/rfc2131.txt>.
- Foundation, W. (2015). What is wireshark? <https://www.wireshark.org/faq.html#q1.1>. [Online; acesso em 10-Julho-2015].
- Jain, R. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley-Interscience, 1st edition.
- Lupu, R. C. (2011). Dhcp client-server. <http://www.elcom.pub.ro/~rlupu/>. [Online; acesso em 10-Julho-2015].
- NS3 (2015). What is ns3. <http://www.nsnam.org/overview/what-is-ns-3/>. [Online; acesso em 10-Julho-2015].

VII

Sessão de Pôsteres

Uma Proposta Ubiservice para Tratamentos de Serviços Direcionados a UbiPri

Adrian R. Lemes Caetano¹, Maycon Viana Bordin¹, Wagner Kolberg¹, Gustavo B. Brand², Guilherme Dal Bianco², Valderi R. Q. Leithardt[†]

FATEC¹ – Faculdade de Tecnologia SENAI - Grupo de Pesquisas em Processamento Paralelo e Distribuído Inteligente - GPPD-i CEP: 91140000
Porto Alegre – RS - Brazil

UFFS² – Universidade Federal da Fronteira Sul - Grupo de Banco de Dados - Chapecó - SC - Brazil

{adrianlemess, mayconbordin, wagnerkrs, gugabrand, dbguilherme
profvalderi}@gmail.com

Abstract. *Ubiquitous computing is rapidly becoming a reality in different environments. One of the challenges of it is to provide relevant and quality information for users. In this context, we propose a smart message notification that respects the user's and environment's privacy. To achieve this goal, the system uses the environment's context information and the user's preferences to decide when and how the messages should be delivered.*

Resumo. *A computação ubíqua está rapidamente se tornando uma realidade nos mais diferentes cenários. Um dos grandes desafios da computação ubíqua é fornecer informações relevantes e de qualidade para os usuários. Neste contexto, o artigo descreve uma proposta para o envio inteligente de mensagens que respeita a privacidade do usuário e do ambiente. Para tal, o sistema utiliza as informações de contexto do ambiente e as preferências do usuário para decidir como e quando as mensagens devem ser enviadas.*

1. Introdução

Com a evolução da capacidade de processamento de dispositivos móveis, diversos trabalhos estão sendo desenvolvidos visando a criação de ambientes inteligentes e dinâmicos, que possibilitam a integração da tecnologia de forma natural à vida humana. Tal cenário, conhecido como computação ubíqua, disponibiliza de forma transparente, acesso a novos serviços [Perera et al., 2014].

No âmbito da computação ubíqua muitos problemas podem ser abordados, dentre eles é possível destacar o gerenciamento e controle de privacidade. Este requisito propõe que as aplicações ubíquas satisfaçam de forma contínua e independente o usuário, sem atrapalhá-lo [Weiser 1991]. Considerando o controle e gerenciamento de privacidade, esse trabalho, portanto, propõe um módulo de notificações que tem como objetivo tornar o ambiente informativo e dinâmico de maneira sensível ao contexto.

*Este trabalho foi desenvolvido com recursos do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) em conjunto com a Federação das Indústrias do Estado do Rio Grande do Sul (FIERGS) em parceria com o IEL/SENAI, Edital Inova Talentos 2014.

Para tanto, foi considerado o perfil do usuário, sua localização, o tipo de ambiente, critérios de tempo, prioridade de mensagem e as preferências do usuário, para então definir o formato de envio da mensagem.

Ambientes de ensino foram utilizados como cenário da aplicação, considerando somente estabelecimentos onde há em funcionamento um sistema de gerenciamento acadêmico (SGA). O SGA irá gerenciar informações nas instituições, como o controle de frequência de alunos, data de provas e trabalhos, notas, conteúdo das aulas, entre outros dados.

Este artigo está estruturado para melhor entendimento da seguinte forma: na Seção 2 são descritos os trabalhos relacionados. O modelo proposto neste artigo é descrito na Seção 3. E por fim, é apresentado os resultados esperados na Seção 4.

2. Trabalhos relacionados

Na área da computação ubíqua, existem projetos que tratam sobre controle de privacidade direcionado ao usuário, aos dispositivos, serviços ou à comunicação que os mesmos possuem em diversos ambientes. Entretanto, apesar de já existirem trabalhos na literatura que tratam de controle [Li 2009 e Pereira 2011], estes não estão relacionados ao ambiente que é quem dita as regras, faltando detalhes de como o controle será desenvolvido e gerenciado. No trabalho desenvolvido por [Coyle et al., 2008] é definido um sistema de notificações que decide quando e onde devem ser enviadas as mensagens, conforme sua prioridade e informações de contexto. Apesar de utilizar aprendizado para melhorar a identificação de mensagens relevantes, o sistema não possui um mecanismo que reconheça, de forma automática, ambientes visitados frequentemente pelo usuário.

Além de aspectos de gerenciamento e controle de privacidade, para o envio de notificações baseado no contexto do usuário e do ambiente, o modelo proposto nesse artigo também descreve mecanismos de aprendizado, para identificar novos ambientes e melhorar as decisões em locais frequentemente visitados pelo usuário.

3. Modelo Proposto

O modelo de envio de notificações proposto será implementado como um módulo que estende o UbiPri, um middleware de tratamento de privacidade e controle de ambientes ubíquos. O UbiPri tem como função o controle e identificação taxonômica das informações para controle de privacidade em ambientes ubíquos. O middleware utiliza critérios como perfis, localização dos usuários e variáveis temporais a fim de aplicar ações diversas no ambiente [Leithardt, 2013].

O objetivo do módulo de notificações é gerenciar informações e notificar os usuários do sistema, com base na sensibilidade de contexto. Para isso, é considerado o ambiente em que o usuário se encontra, suas preferências, a forma de enviar notificações (via SMS, e-mail ou pela aplicação), a prioridade das mensagens e a disponibilidade dos dispositivos dos usuários.

A Figura 1 apresenta a integração do módulo de notificações ao UbiPri. O modelo proposto é responsável pela busca de informações relevantes na base de dados do SGA para serem repassadas ao usuário. Além disso, tem como função utilizar alguns

critérios do UbiPri para auxiliar no envio de notificações. O UbiPri por sua vez, irá utilizar os usuários existentes do SGA para identificação dos perfis.

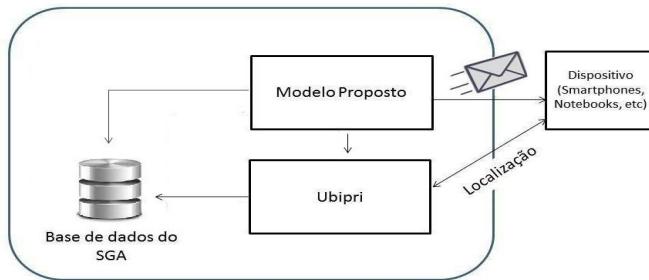


Figura 1 - Integração do modelo proposto com o UbiPri.

Para o processo de tomada de decisão pelo módulo de notificações, considerando meios e mecanismos de envio de mensagens ao usuário, serão utilizados alguns critérios definidos pelo UbiPri. O primeiro critério é o perfil de usuário, nele será utilizado os perfis de professor, estudante e administrador. Para o administrador, por exemplo, poderá ser enviado um aviso de tentativa de acesso indevido ao sistema e para um estudante, deverá ser enviado seu limite de frequência ou alguma outra informação relacionada às suas aulas, como o conteúdo a ser abordado.

Referente ao segundo critério, foram definidos três tipos de ambientes: restrito, público e privado. Para tanto foi considerado como ambiente público as instituições de ensino, onde haverá uma política *default* para os usuários. Na instituição de ensino a tomada de decisão será predefinida, sendo conhecido previamente os recursos necessários para o envio de notificações (disponibilidade de internet, dos dispositivos, horário das aulas e localização das salas). Quando for identificado outro ambiente, acessado frequentemente, é iniciado um processo de aprendizagem. O usuário então poderá configurar suas preferências, podendo tornar o ambiente como privado, aplicando as regras pertinentes à nova configuração.

Qualquer outro ambiente, além de algum local já configurado como privado ou público, por padrão, será considerado como restrito. Além disso, caso o usuário esteja em um ambiente restrito, as notificações serão armazenadas em uma base de dados contendo o histórico para serem visualizadas posteriormente.

Por fim, o terceiro critério se refere às variáveis temporais, onde o modelo proposto, consultando o SGA, identifica os dias letivos e os turnos das aulas na instituição de ensino. Assim, será possível determinar o momento certo em que o usuário será notificado.

Definido o perfil do usuário, o tipo de ambiente e as variáveis temporais, ainda não temos critérios suficientes para flexibilizar o envio de notificações. Para isto será proposto dois novos critérios complementares: a prioridade das mensagens, relacionadas à relevância de seu conteúdo e as preferências de usuário. A prioridade de mensagem poderá sobrepor os bloqueios definidos em ambientes restritos, caso o conteúdo seja de extrema urgência para o usuário, como por exemplo, uma aula cancelada. As preferências do usuário também poderão sobrepor determinados

critérios, pois são regras personalizadas. Como exemplo, seria o usuário marcar determinado tipo de mensagem para não ser visualizada novamente.

A partir dos critérios mencionados nessa seção, será possível inferir o modelo de envio de mensagens. Por exemplo, caso o dispositivo não tenha acesso à internet em um ambiente privado, é possível efetuar as notificações de média e alta prioridade via SMS. Já para ambientes restritos, quando houver a necessidade de enviar um aviso de alta prioridade, poderá ser enviada uma notificação via e-mail. Ainda, em outros casos, a mensagem será armazenada para ser visualizada posteriormente.

O modelo proposto encontra-se em fase de concepção, onde para a comunicação entre o SGA e o sistema de notificações será utilizado um modelo publisher/subscriber, onde será consumido de um *message queue* os eventos a serem notificados aos usuários. Utilizaremos o Play Framework para desenvolvimento em Java do webservice da aplicação. Pretende-se utilizar também o algoritmo *decision table* para controle das regras, que serão armazenadas em uma base de dados PostGreSQL. A base será consultada através de um *daemon* responsável pela tomada de decisões e o envio das notificações.

Os critérios utilizados nesse modelo proposto, tem como principal objetivo evitar ao máximo que a privacidade do usuário seja de alguma forma invadida, garantindo a informação correta e na granularidade esperada, no momento apropriado.

4. Resultados Esperados

O sistema proposto mostra-se uma inovadora forma de gerenciamento de notificações sensíveis ao contexto do usuário. Ao tratar diferentes tipos de notificações, espera-se que não seja ultrapassado o limite que separa as informações relevantes, do envio massivo de informações com pouca utilidade. Para isso, é importante que a partir dos critérios mencionados nesse artigo e a possibilidade de personalizar o recebimento de mensagem, seja possível evitar o envio de informações irrelevantes para o usuário. Espera-se expandir em trabalhos futuros, o caso de estudo para outros cenários.

5. Referências

- Coyle, Stephen Knox—Ross Shannon—Lorcan, and Adrian K. Clear (2008) “Scatterbox: Context-aware message management”, In: Revue d’Intelligence Artificielle, Article Vol 22/5 - 2008 - p.549-568.
- Leithardt, Valderi,et al. (2013) “Mobile Architecture for Identifying Users in Ubiquitous Environments Focused on Percontrol”, In: UBICOMM. p. 145-151.
- N. Li, N. Zhang, S.K. Das, B. Thuraisingham (2009) “Privacy preservation in wireless sensor networks: A state-of-the-art survey”, In: Ad Hoc Networks 7 1501-1514
- Pereira, J.S. Silva, J. Granjal, R. Silva, E. Monteiro, Q.Pan (2011) “A Taxonomy of Wireless Sensor Networks with QoS”, In: NTMS, Paris, p. 1-4
- Perera, Charith, et al. (2014) "Context aware computing for the internet of things: A survey", Communications Surveys & Tutorials, In: IEEE 16.1: 414-454.
- Weiser, M. (1991) “The computer of the 21st Century”, Scientific American, p. 265.

BlueTApp - Um Aplicativo Móvel para Registro Automático da Presença Acadêmica via Bluetooth

**Fernando Weber Albiero¹, Fábio Weber Albiero²,
João Carlos Damasceno de Lima³, Iara Augustin³**

¹ Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre – RS – Brasil

²Instituto Federal Farroupilha – Câmpus Santo Ângelo
Santo Ângelo – RS – Brasil

³Laboratório de Sistemas de Computação – Universidade Federal de Santa Maria
Santa Maria – RS – Brasil

fernando.albiero@ufrgs.br, {weber, caio, august}@inf.ufsm.br

Abstract. This paper presents a mobile application to the Android platform that, through using Bluetooth technology, automates the academic frequency record process at educational institutes.

Resumo. Este artigo apresenta um aplicativo móvel para a plataforma Android que, através do uso da tecnologia Bluetooth, automatiza o processo de registro da frequência acadêmica nas instituições de ensino.

1. Introdução

De acordo com Lei nº 9.394 [Brasil 1996], os alunos matriculados em uma disciplina devem possuir uma frequência mínima para a aprovação de 75% (setenta e cinco por cento) sobre o total de aulas ministradas e demais atividades acadêmicas. Na maioria das instituições de ensino, o registro da frequência ainda é realizado de modo tradicional, ou seja, através do uso de papel e caneta; o que demanda tempo e tem impacto direto no planejamento da aula. Assim sendo, o tempo destinado para o registro da frequência poderia ser melhor utilizado, caso esse processo fosse automatizado.

Neste contexto, este artigo apresenta um aplicativo móvel, chamado de BlueTApp, para a plataforma Android que, através do uso da tecnologia Bluetooth, automatiza o processo de registro da frequência acadêmica nas instituições de ensino. O aplicativo captura os sinais Bluetooth dos dispositivos móveis dos alunos e, através das informações obtidas a partir desses sinais, verifica a presença ou a ausência do aluno em sala de aula.

2. Trabalhos correlatos

Atualmente, no mercado, há uma série de alternativas que realizam a tarefa de automação do registro da frequência acadêmica [da Silva 2002, Heck 2013, Chamon 2014], porém, cada qual com suas particularidades. Por exemplo, a Universidade Católica de Minas desenvolveu em parceria com a Universidade Centro Leste um sistema ubíquo para o registro automático da presença acadêmica de alunos [Chamon 2014]. Bem diferente do aplicativo apresentado neste trabalho, o sistema desenvolvido pela Universidade Católica

de Minas funciona através do uso de um cartão eletrônico integrado com um microchip de rádio frequência que é alimentado por uma bateria e busca por uma rede sem fio. Uma vez descoberta a rede, o microchip envia informações para a rede, identificando e localizando o aluno que precisa permanecer, no mínimo, quinze minutos ao alcance da mesma para ter sua presença registrada.

3. BlueTApp

O BlueTApp visa automatizar e agilizar o processo de registro da frequência acadêmica nas instituições de ensino. Esse aplicativo foi desenvolvido em linguagem Java, somente para o sistema operacional Android (o BlueTApp ainda não foi desenvolvido para outros sistemas operacionais, tais como iPhone OS e Windows Phone). A ideia central do BlueTApp é capturar os sinais Bluetooth dos dispositivos móveis dos alunos e, através das informações obtidas a partir desses sinais, verificar a presença ou a ausência do aluno em sala de aula. Para tornar isso possível, o aplicativo foi desenvolvido em duas versões: uma versão para o aluno e outra para o professor.

3.1. Versão do aluno

A versão do aluno, chamada de BlueTApp Aluno, tem como objetivo padronizar o nome dos dispositivos móveis dos alunos. Essa padronização consiste em alterar o nome do dispositivo móvel para a matrícula do aluno, que é única e serve como campo identificador (ID). A padronização dos nomes dos dispositivos dos alunos ocorre da seguinte forma: primeiramente, o aluno deve informar a matrícula (veja a figura 1 (a)); posteriormente, deve autorizar (via uma solicitação) o uso do adaptador Bluetooth do aparelho. Caso essa solicitação seja aceita pelo aluno, o aplicativo verifica o estado atual do adaptador Bluetooth. Caso o adaptador esteja ligado, o aplicativo altera o nome do dispositivo para matrícula do aluno. Em contrapartida, caso o adaptador esteja desligado, o aplicativo modifica o seu estado para “ligado” e altera o nome do dispositivo para a matrícula do aluno. Após finalizada essa operação, o aplicativo torna o dispositivo visível para outros dispositivos e uma mensagem é exibida ao aluno informando que seu aparelho está pronto para o registro da frequência via Bluetooth.

Em resumo, a versão BlueTApp aluno é um ferramenta facilitadora para ativar o adaptador Bluetooth e alterar o nome do dispositivo para a matrícula do aluno, que ainda pode ser alterada de modo manual (Menu → Configurações → Conexões sem fio e rede → Configurações Bluetooth → Ativar).

3.2. Versão do professor

A versão do professor, chamada de BlueTApp Professor (veja a figura 1 (b)), é responsável pelo processo de registro acadêmico de forma automática e transparente. Dentre as principais ações, essa versão do aplicativo permite ao docente: inserir e remover turmas; inserir e remover alunos; realizar o controle da frequência via Bluetooth ou de forma manual; e, exportar os dados. A versão BlueTApp Professor opera com o banco de dados SQLite (banco de dados nativo da plataforma Android). A inserção de turmas e alunos no aplicativo resulta, respectivamente, na inserção de tabelas no banco de dados e registros nas tabelas. Por outro lado, a remoção de turmas e alunos resulta, respectivamente, na remoção de tabelas do banco de dados e registros nas tabelas. Quanto a inserção de turmas deve-se destacar que, é necessário que o professor acesse o portal da instituição de

ensino e importe os arquivos .csv referentes às turmas, as quais ele irá realizar o registro acadêmico. Esse processo de importação evita que o professor digite todos os dados referentes às turmas de forma manual. As informações contidas nos arquivos .csv são armazenadas no banco de dados. Além disso, não há um número mínimo ou máximo para o número de alunos nas turmas, entretanto, há um número máximo de turmas que podem ser cadastradas pelo aplicativo. Esse número máximo de turmas fica limitado pelo número máximo de tabelas que o banco de dados SQLite suporta.

A chamada por Bluetooth, principal ação disponibilizada pelo aplicativo BlueTApp Professor, realiza o registro da frequência de forma automática. Ao escolher esta ação, uma tela é exibida, na qual o professor, antes de realizar o registro, deve indicar o tópico da aula e a carga horária. Após a confirmação dos dados, o aplicativo verifica se o adaptador Bluetooth do dispositivo está ligado. Caso não esteja, uma solicitação de permissão é exibida. Uma vez aceita a solicitação, o aplicativo ativa o adaptador Bluetooth iniciando a busca pelos sinais dos dispositivos dos alunos. Caso esta seja a primeira chamada por Bluetooth, o aplicativo BlueTApp Professor armazena o endereço físico do adaptador Bluetooth dos dispositivos dos alunos no banco de dados. Caso contrário, o aplicativo compara o endereço físico encontrado com os endereços físicos dos sinais anteriormente armazenados; se tanto o endereço físico quanto a matrícula forem a mesma, o aluno recebe a presença. Isso acaba limitando o aplicativo ao número de 1 dispositivo por aluno. Por fim, todos os dispositivos encontrados são listados para o docente, assim como o número total de dispositivos encontrados (veja a figura 1 (c)).

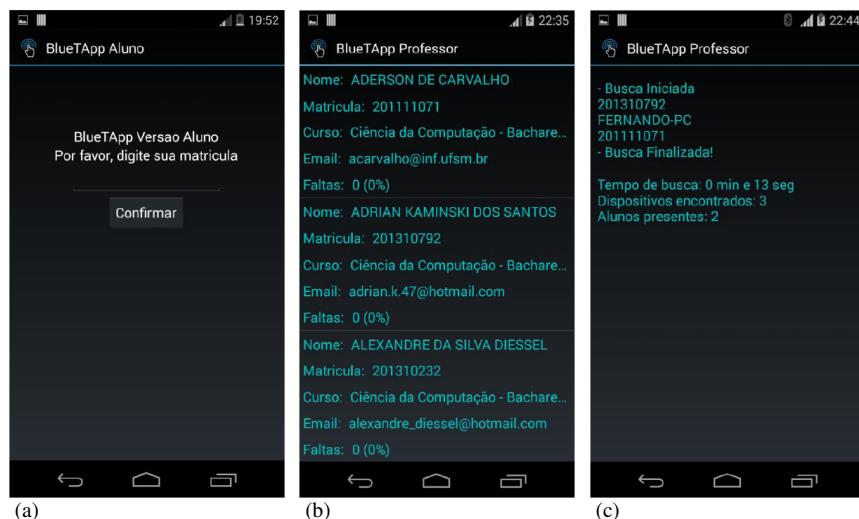


Figura 1. (a) Tela inicial do aplicativo BlueTApp Aluno; (b) BlueTApp Professor exibindo as informações sobre cada aluno; (c) BlueTApp Professor ao término da busca por dispositivos.

Quanto a comunicação entre os dispositivos, esta foi realizada sem o pareamento dos mesmos, uma vez que através de testes verificou-se que o pareamento do dispositivo do professor com os dispositivos de todos os alunos não poderia ser efetuado de forma

simultânea. Em outras palavras, o dispositivo móvel do docente precisa criar uma fila de pareamento, na ordem em que os sinais foram encontrados e enviar as solicitações de pareamento uma a uma. Por meio de testes, pode-se notar que conforme aumenta o número de dispositivos envolvidos no processo, aumenta também o tempo para a realização da descoberta de sinal e pareamento entre os dispositivos envolvidos. Em função disso, optou-se por não efetuar a troca de mensagens entre os dispositivos e realizar a automação do registro da frequência acadêmica sem realizar o pareamento.

Quanto a segurança do aplicativo BlueTApp, este apresenta algumas limitações, que ainda terão que ser sanadas. Uma limitação é: um aluno que tenha trocado seu dispositivo móvel não será reconhecido pelo BlueTApp Professor, visto que o endereço físico da interface Bluetooth não será mais o mesmo que consta no banco de dados do aplicativo do professor. Outra limitação é: um aluno fora da sala de aula, porém dentro do limite de alcance do sinal Bluetooth (que é de aproximadamente 10 metros), com um dispositivo já pré-configurado, poderá receber a presença.

4. Considerações finais

Este trabalho apresenta como contribuição um aplicativo móvel para a automação e gerenciamento do registro da frequência acadêmica nas instituições de ensino. O aplicativo também pode ser usado como ferramenta de consulta para os professores, podendo estes obterem informações dos alunos e das aulas já ministradas. Foram realizados testes afim de validar o funcionamento do aplicativo. Os testes ocorreram na Universidade Federal de Santa Maria, em algumas salas de aula. Nestes testes, foi solicitado aos alunos que instalassem a versão BlueTApp Aluno em seus dispositivos móveis. Aos alunos que estavam usando dispositivos com plataformas que não a Android, solicitou-se que efetassem de modo manual os passos realizados pelo aplicativo. Em todos os testes, o aplicativo conseguiu capturar os sinais Bluetooth de todos os dispositivos envolvidos no processo e registrar a frequência de forma automática, alcançando assim os objetivos inicialmente propostos neste trabalho.

Como trabalhos futuros, espera-se incrementar o número de funcionalidades do aplicativo, disponibilizar versões do aplicativo para o iPhone OS e Windows Phone; e, desenvolver mecanismos de segurança, como alguns protocolos, afim de evitar que o aluno receba presença sem estar em sala de aula.

Referências

- Brasil (1996). Lei n 9.394, de 20 de dezembro de 1996. Disponível em: http://www.planalto.gov.br/CCIVIL_03/leis/L9394.htm. Acesso em: 25/06/2015.
- Chamon, J. P. M. (2014). Registro ubíquo de controle acadêmico: localização em ambiente interno utilizando ciclo de trabalho dinâmico. *XXXIV Congresso da Sociedade Brasileira de Computação*.
- da Silva, F. L. M. (2002). Protótipo de hardware para controle de frequência acadêmica. Disponível em: dsc.inf.furb.br/arquivos/tccs/monografias/2002-1fernandoluizmelatidasilvavf.pdf. Acesso em: 12/08/2015.
- Heck, F. S. (2013). Sistema móvel de controle de presença. Disponível em: www.lume.ufrgs.br/bitstream/handle/10183/100288/000931702.pdf. Acesso em: 12/08/2015.

Redes Definidas por Software: Monitoramento Sensível ao Contexto

**Lucas Powaczuk¹, Leonardo da C. Marcuzzo², Luiz E. G. da Silva¹, Vania Freitas¹,
Tassiana Kautzmann¹, Roseclea D. Medina¹**

¹Programa de Pós-Graduação em Informática – Universidade Federal de Santa Maria
(PPGI/UFSM)

Avenida Roraima, 1000 – 97.105-900 – Santa Maria – RS – Brazil

²Grupo de Redes e Computação Aplicada – (GRECA/UFSM)

{lucaspw12, luizevandro.silva, 2.vania, tassik,
roseclea.medina}@gmail.com, lmarcuzzo@inf.ufsm.br

Abstract. This paper focuses on monitoring systems to Software-Defined Networks (SDN), making a analysis of three tools developed over the last year: FlowCover, SUMA and EnterpriseVisor. With this study it's proposed to highlight the main features of each tool, detecting its effectiveness in the context-awareness aspect. It starts with a review on the subject, which aims to develop a context-aware monitoring tool for SDN.

Resumo. Este trabalho enfoca sistemas de monitoramento para Redes Definidas por Software, fazendo uma análise de três ferramentas desenvolvidas no último ano: FlowCover, SUMA e EnterpriseVisor. Propõe-se com o estudo destacar as características principais de cada uma das ferramentas, detectando sua eficácia no aspecto de sensibilidade ao contexto. Parte-se de uma revisão sobre o tema, que objetiva desenvolver uma ferramenta a sensível ao contexto para monitoramento de SDNs.

1. Introdução

Redes Definidas por Software ou *Software Defined Network (SDN)* caracterizam-se como um conceito emergente no mundo de redes, caracterizando uma mudança paradigmática. Seu caráter revolucionário propõe transformações nas formas de operações conhecidas da atualidade, especialmente por separar o plano lógico dos equipamentos, promovendo a centralização do controle em uma entidade gerenciadora da rede [Kreutz et al. 2015].

Em redes tradicionais, o gerenciamento é uma tarefa complexa que na maioria das vezes requer a configuração individual de cada equipamento, fragilizando o processo de monitoramento da rede na sua totalidade. Aliado a isso, está a dificuldade em detectar e/ou resolver falhas emergentes no uso, de forma dinâmica, caracterizando uma rede nata ou pouco sensível ao contexto.

O conceito de rede sensível ao contexto indica para a capacidade de adaptação da rede de acordo com as demandas evidenciadas no contexto, através da coleta de dados/informações, “para tomar a decisão ideal para a determinada situação de forma automática” [Burceanu et al. 2013]. Considera-se contexto de rede como um conjunto de

atributos que caracterizam uma determinada rede. Conforme [Wang et al. 2014] podem ser informações dos equipamentos e nós (*nodes*), informações do *link*, como perda de pacotes, atraso e *jitter*, largura de banda do controlador, informações estatísticas dos fluxos, entre outros.

Nesta perspectiva, este trabalho enfoca sistemas de monitoramento para ambientes *SDN*, fazendo uma análise de 3 ferramentas desenvolvidas no último ano: o *FlowCover*, *SUMA* e *EnterpriseVisor*. Propõe-se com o estudo destacar as características principais de cada uma das ferramentas, detectando sua eficácia no quesito sensibilidade ao contexto. Logo, o interesse é investigar ferramentas de monitoramento com vistas a detectar características e funções que possam subsidiar a construção pretendida. Ressalta-se, desta forma para o caráter exploratório do estudo apresentado, justificando sua relevância na divulgação e socialização de dados coletados na pesquisa em desenvolvimento.

2. Sistemas de Monitoramento em SDN

Atualmente vários estudos vêm destacando ferramentas de monitoramento para ambientes *SDN*, evidenciando um campo de estudo em ascendência. [Kreutz et al. 2015] ao apresentar uma revisão sistêmica sobre o tema, destaca os seguintes sistemas de monitoramento em redes definidas por software: *BISmark*, *DCM*, *FleXam*, *FlowSense*, *Measurement Model*, *OpenNetMon*, *OpenSample*, *OpenSketch*, *OpenTM*, *PaFloMon* e *PayLess*.

Já no estudo realizado por [Yassine et al. 2015], são listados 15 ferramentas de monitoramento, dentre as quais estão: *OpenNetMon*, *iSTAMP*, *OpenTM*, *PayLess*, *FlowSense*, *Zhang*, *DREAM*, *Baadaat*, *HONE*, *PLANCK*, *OpenSample* e *OpenSketch* e as ferramentas desenvolvidas por [Jose et al. 2011], [Moshref et al. 2013] e [Dusi et al. 2014] as quais não receberam denominação específica.

A revisão desenvolvida para o presente trabalho identificou três ferramentas recentes que são: *FlowCover*, *SUMA* e *EnterpriseVisor*, as quais não foram mencionadas nas revisões sistêmicas acima mencionadas. A primeira ferramenta, o *FlowCover*, é um *framework* de monitoramento de baixo custo operacional e com alta precisão de monitoramento. O sistema utiliza pequena carga de recursos, pois ele agrupa mensagens do tipo pedido e resposta, otimizando a frequência de *pollings* (consultas) nos agentes, através do módulo Flow Stat Aggregator (figura 1). Os resultados mostraram que o uso do *FlowCover* diminuiu o *overhead* causado pelas funções de monitoramento da rede em até 50% dos casos.

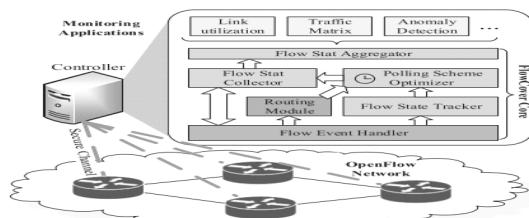


Figura 1. Funcionamento do sistema FlowCover

O segundo sistema, o *SUMA*, é um *middlebox* de monitoramento inteligente que fornece controle, filtro e monitoramento [Choi et al. 2014]. Sua característica principal é a facilidade de monitorar eventos de detecção de anomalias e de filtragem de tráfego. Sua proposta engloba eventos para verificar *status* dos *switches*, inspeção de tráfego, modificação de mensagens, detecção de anomalias de rede e identificação de possíveis ataques. Seu custo operacional é baixo em eventos de detecção e filtragem entre controladores e *switches OpenFlow*. Sua implementação em hardware e software conseguiu atingir uma capacidade de processamento de pacotes de até 10Gbps.

Por último o *EnterpriseVisor*, é um sistema de gerenciamento de recursos de rede, que funciona dividindo a rede em *slices* (partes), conforme a figura 2, monitorando e alocando os recursos dinamicamente entre as partes [Chen et al. 2014]. Seu funcionamento ocorre a partir de um sistema de tomada de decisão baseado em regras pré-definidas, de modo a utilizar mais eficientemente os recursos da rede.

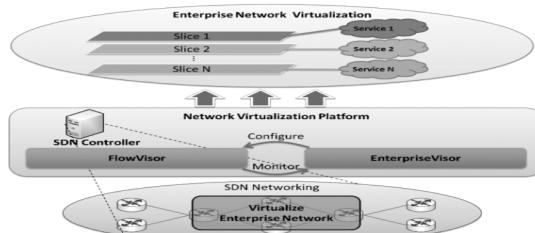


Figura 2. Sistema EnterpriseVisor segmentando a rede em slices.

O *EnterpriseVisor* utiliza o *FlowVisor*, que é um mecanismo de virtualização de rede, onde sua função é permitir que múltiplas redes lógicas possam compartilhar a mesma infraestrutura física conforme [Kreutz et al. 2015]. Através do *FlowVisor*, a rede é dividida em instâncias (*slices*) onde cada *slice* pode requisitar uma quantidade de recursos diferente das demais partes [Chen et al. 2014], melhorando a utilização de recursos ociosos.

2. Sistemas de Monitoramento em SDN

A partir da caracterização das três ferramentas de monitoramento buscou-se identificar o sistema mais indicado para o contexto de redes sensíveis. Considera-se, nesta direção, a configuração de uma rede inteligente, ou seja, que tenha capacidade de se adaptar ao ambiente/contexto, realizando tarefas de forma dinâmica e automática. O quadro abaixo apresenta as ferramentas analisadas destacando características relativas à condição do monitoramento, tipos de funções suportadas, custo operacional e funcionamento:

Tabela 1. Análise das ferramentas de monitoramento para SDNs:

Aplicação	Funções	Custo operacional	Funcionamento
FlowCover	Estáticas	Baixo (software)	Monitoramento de Flows
SUMA	Estáticas	Alto (hardware e software)	Monitoramento e detecção de anomalias
EnterpriseVisor	Dinâmicas	Alto (software)	Alocação dinâmica de recursos da rede

Dentre as ferramentas de monitoramento analisadas, o sistema mais indicado no quesito considerado foi o *EnterpriseVisor*. Com esta ferramenta é possível, dentre outros, configurar uma rede com foco na priorização de serviço/aplicação, incrementar a sensibilidade ao contexto através de seu sistema dinâmico de tomada de decisões, otimizar recursos ociosos e redistribuir estes nas demandas manifestas, através da alocação dinâmica e de forma automática. Conclui-se, desta forma, que as ferramentas de monitoramento sensíveis ao contexto para ambiente de redes definidas por software precisam ser maleáveis ao contexto exigindo uma alta capacidade de coleta e especialmente de tratamento de informações, de modo a adaptar-se ao contexto.

Referências

- Chen, J., Ma, Y., Kuo, H. and Hung, W. (2014). EnterpriseVisor : A Software - Defined Enterprise Network Resource Management Engine. p. 381–384.
- Choi, T., Song, S., Park, H., Yoon, S. and Yang, S. (2014). SUMA: Software-defined unified monitoring agent for SDN. IEEE/IFIP NOMS 2014 - IEEE/IFIP Network Operations and Management Symposium: Management in a Software Defined World,
- Kreutz, D., Rothenberg, C. E., Ieee, M., et al. (2015). Software-Defined Networking : A Comprehensive Survey. v. 103, n. 1.
- Su, Z., Wang, T., Xia, Y. and Hamdi, M. (2014). FlowCover : Low-cost Flow Monitoring Scheme in Software Defined Networks.
- Yassine, A., Rahimi, H. and Shirmohammadi, S. (2015). Software Defined Network Traffic Measurement: Current Trends and Challenges. n. April.
- L. Jose, M. Yu, and J. Rexford, “Online measurement of large traffic aggregates on commodity switches,” in Proc. of the USENIX HotICE Workshop, 2011.
- M. Moshref, M. Yu, and R. Govindan. (2013). Resource/accuracy tradeoffs in software-defined measurement, in Proc. Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, HotSDN ’13, pp. 73–78.
- M. Dusi, R. Bifulco, F. Gringoli, and F. Schneider, Reactive logic in software-defined networking: Measuring flow-table requirements, in Proc. Intern. Wireless Comm. and Mobile Computing Conf. (IWCMC), pp. 340-345, 4-8 Aug. 2014.
- Burceanu, E.; Dobre, C.; Cristea, V.; Costan, A.; Antoniu, G. (2013) “Distributed Data Storage in Support for Context-Aware Applications”. In: 12th International Symposium on Parallel and Distributed Computing, IEEE.

Da Elaboração a Implantação da Política de Segurança da Informação: uma proposta baseada no ciclo PDCA

Peter Prevedello¹, Diogo Otto Kunde¹

¹Eixo Informação e Comunicação - Instituto Federal de Educação, Ciência e Tecnologia Farroupilha

Postal 98130-000 – Júlio de Castilhos – RS - Brasil

peterprevedello@gmail.com, diogokunde@hotmail.com

Abstract. *Information security is an area of knowledge dedicated to information asset protection against unauthorized access, unauthorized changes or unavailability. From a broader way we can also consider it as risk management practice involving the commitment of the three main concepts of security: confidentiality, integrity and availability of information. In this work we present a proposal to develop and implement an Information Security Policy based on the PDCA cycle, which enables continuous process improvement and problem solving.*

Resumo. *Segurança da Informação é uma área do conhecimento dedicada à proteção de ativos da informação contra acessos não autorizados, alterações indevidas ou sua indisponibilidade. De uma forma mais ampla, podemos também considerá-la como prática de gestão de riscos que impliquem o comprometimento dos três principais conceitos da segurança: confidencialidade, integridade e disponibilidade da informação. Neste trabalho apresentaremos uma proposta de elaboração e implantação de uma Política de Segurança da Informação baseada no ciclo PDCA, que possibilita a melhoria contínua de processos e solução de problemas.*

Introdução

Com o rápido aumento de dispositivos conectados a rede, a informação se tornou o ativo mais crítico dentro das instituições e a criação e implantação de uma Política de Segurança da Informação (PSI) se tornou de extrema importância para garantir sua confidencialidade, integridade e disponibilidade. A segurança da informação preserva a confidencialidade, integridade e disponibilidade da informação por meio da aplicação de um processo de gestão de riscos e fornece confiança para as partes interessadas de que os riscos são adequadamente gerenciados [ISO/IEC 27001 2013]. Neste contexto, é inadmissível uma instituição de ensino não possuir no seu planejamento um processo de implantação de política de segurança da informação. Desta forma este trabalho tem como objetivo apresentar uma proposta de processo de elaboração e implantação de política de segurança da informação (PSI) com base no ciclo PDCA no Instituto Federal Farroupilha, campus Júlio de Castilhos-RS (IF-Farroupilha-JC).

A Norma Brasileira (NBR) ISO/IEC 27001:2013 adota o ciclo PDCA – Plan-Do-Check-Act (Planejar, Fazer, Checar e Agir) para estruturar todos os processos envolvidos em uma PSI, sendo que o PDCA é uma ferramenta gerencial que possibilita a melhoria contínua de processos e a solução de problemas [AGUIAR 2006].

O IF-Farroupilha-JC é uma instituição de educação superior, básica e profissional, especializada na oferta de educação profissional e tecnológica nas diferentes modalidades de ensino [FARROUPILHA 2015]. Atualmente o campus possui um pouco mais de vinte setores e por esse motivo a proposta prevê a elaboração e implantação de PSI por setor, com o objetivo de envolver todos os funcionários da instituição. Neste sentido, a proposta do processo de elaboração e implantação da PSI teve como base a NBR ISO 27001:2013, e trás como objetivo especificar requisitos para o estabelecimento, implementação, operação, monitoração, análise crítica, manutenção e melhoria de uma PSI [ISO/IEC 27001:2013].

Elaboração e implantação da PSI

Tendo em vista o PDCA, todo processo deve ter uma ampla fase de planejamento, onde se busca conhecer o ambiente a qual será implantada uma PSI [SÊMOLA 2014]. Sendo assim a presente proposta prevê quatro fases para sua implantação em um setor do IF-Farroupilha-JC. A Figura 1 ilustra a proposta de elaboração e implantação da PSI com base no ciclo PDCA.

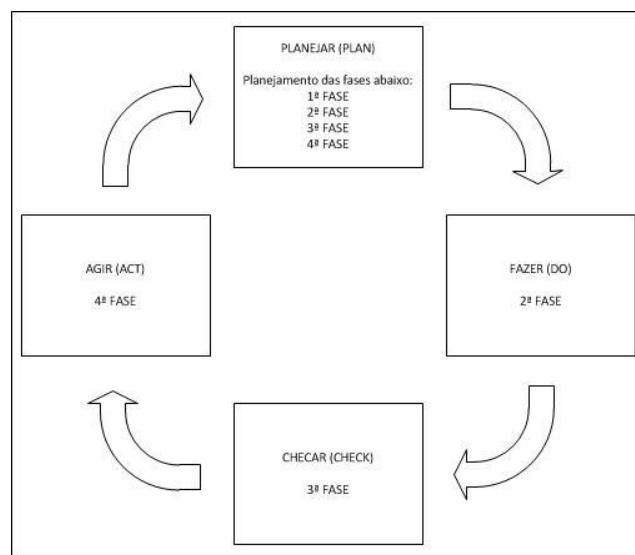


Figura 1 - Proposta de elaboração e implantação da PSI com base no ciclo PDCA.

A primeira fase propõe a criação do comitê gestor de segurança que será responsável pela implantação e execução das fases um e três, sendo necessária a participação efetiva da direção geral do campus, demonstrando liderança e comprometimento, garantindo assim que a PSI seja compatível com o plano de desenvolvimento institucional¹ (PDI) do campus. O comitê gestor de segurança da

¹ <http://www.iffarroupilha.edu.br/site/conteudo.php?cat=168&sub=5377>

informação deverá ser definido e composto pela direção geral e pelo coordenador de tecnologia da informação, da mesma forma um comitê operacional também deverá ser criado para a implantação e execução das fases dois e quatro. O comitê operacional deverá ser composto pelos funcionários do setor onde a PSI está sendo implantada e também pelos funcionários da coordenação de tecnologia da informação [BEZERRA 2011].

Na segunda fase será realizada a gestão de riscos utilizando como referência a NBR ISO/IEC 27005:2013, onde serão analisados e avaliados estes riscos através da identificação dos ativos e identificação dos processos de negócios [CAMPOS 2007]. Feito isso, será necessário determinar a relação ativo *versus* processo, a fim de identificar a sua necessidade de segurança, de acordo com as propriedades confidencialidade, integridade e disponibilidade utilizando as seguintes perguntas [BEZERRA 2011]:

- Pode ficar indisponível por algum período de tempo? (Disponibilidade);
- Pode ser acessado e divulgado por qualquer pessoa? (Confidencialidade);
- Pode ser modificado por qualquer pessoa? (Integridade).

A partir da análise dos resultados, caso a resposta seja “não” em alguma destas perguntas, o ativo em questão necessita de segurança que garanta aquela propriedade. Para iniciar a identificação dos riscos devem-se avaliar os ativos conforme sua importância, impacto e valor financeiro [ISO/IEC 27005 2013]. Da mesma forma, é necessário identificar e avaliar as ameaças classificando-as em naturais ou humanas, conforme o Anexo C da NBR ISO/IEC 27005:2013. Após identificação das ameaças, identificar e avaliar suas vulnerabilidades utilizando como referência o Anexo ‘D’ da NBR ISO/IEC 27005:2013. Por fim, propõe-se a estimativa de riscos através da equação, conforme o Quadro 1 [ISO/IEC 27005 2013].

Quadro 1 - Equação para estimativa do risco.

$$\text{Risco (A)} = \text{SOMA} [\text{Ameaça} * \text{Vulnerabilidade}] (A) * \text{Valor do ativo (A)}.$$

O resultado desse cálculo representa o RISCO FINAL DE CADA ATIVO e com base nestes resultados realizar-se-á o tratamento dos riscos, utilizando as seções de controle de segurança da NBR ISO/IEC 27002:2013, projetada para as instituições usarem como uma referência na seleção de controles dentro do processo de implementação de PSI baseada na NBR ISO/IEC 27001:2013.

Conforme Manuel (2014), “Não se pode gerenciar aquilo que não se pode medir; por esse motivo deve-se ter um painel de indicadores com o objetivo de controlar, medir e melhorar [...]”, e nesse contexto, será realizada na terceira fase uma auditoria interna por parte do comitê gestor de segurança da informação, verificando o quanto o processo de PSI está implementado e mantido efetivamente. Para tal, este comitê realizará novamente as etapas contidas na segunda fase. O resultado dessa auditoria servirá como indicador de desempenho, a fim de apontar problemas existentes no processo, que na quarta e última fase serão tratados com o intuito de realizar correções e melhorias, utilizando as seções de controle de segurança da NBR ISO/IEC 27002:2013.

Resultados Preliminares e Trabalhos futuros

O processo descrito nesse trabalho foi aplicado parcialmente na Central de Processamento de Dados (CPD) do Setor de Tecnologia da Informação IF-Farroupilha-JC visando alguns resultados preliminares. O processo foi aplicado implantando parte da primeira fase e toda a segunda fase da proposta. Apesar da aplicação do processo no CPD, verificou-se que os ativos que apresentam maior risco são a Internet e o servidor de arquivos, devido à existência de ameaças com alto grau de probabilidade de ocorrerem, entre elas estão à falha do equipamento de telecomunicação e as ameaças representadas por seres humanos como engenharia social², acesso não autorizado ao sistema e vazamento de informações, sendo que todas essas ameaças possuem vulnerabilidades que poderão comprometer todos os serviços dependentes da Internet se vierem a ocorrer.

Como trabalhos futuros, pretende-se aplicar a proposta completa em todos os setores do IF-Farroupilha-JC com o objetivo de envolver os funcionários da instituição. Acredita-se que as pessoas são o principal agente de transformação neste processo de melhoria contínua e o estabelecimento na prática da Política de segurança da Informação.

Referências

- AGUIAR, Silvio. Integração das Ferramentas da Qualidade ao PDCA e ao Programa Seis Sigma. Nova Lima: INDG Tecnologia e Serviços Ltda., 2006.
- BEZERRA, Edson Kowask. Gestão de riscos de TI: NBR 27005. Rio de Janeiro: RNP/ESR, 2011.
- CAMPOS, André. Sistema de Segurança da informação: Controlando os Riscos. 2. ed. - Florianópolis: Visual Books, 2007.
- MANUEL, Sergio da Silva. Governança de segurança da informação: como criar oportunidades para o seu negócio. Rio de Janeiro: Brasport, 2014.
- NBR ISO/IEC 27001 - Tecnologia da Informação. Sistema de Gestão da Segurança da Informação. Associação Brasileira de Normas Técnicas. Rio de Janeiro, 2013.
- NBR ISO/IEC 27002 - Tecnologia da Informação. Código de prática para gestão da Segurança da Informação. Associação Brasileira de Normas Técnicas. Rio de Janeiro, 2013.
- NBR ISO/IEC 27005 - Tecnologia da Informação. Gestão de riscos de Segurança da Informação. Associação Brasileira de Normas Técnicas. Rio de Janeiro, 2013.
- PEIXOTO, Mário C. P. Engenharia Social e Segurança da Informação na Gestão Corporativa . Rio de Janeiro: Brasport, 2006.
- SÊMOLA, Marcos. Gestão da segurança da informação: uma visão executiva. 2. ed. – Rio de Janeiro: Elsevier, 2014.

² Segundo Peixoto (2006), a Engenharia Social é a ciência que estuda como o conhecimento do comportamento humano pode ser usado na persuasão de uma determinada pessoa, fazendo a agir conforme seu desejo.

Processamento Dinâmico de Regras Semânticas para Identificação de Situações

Lidiane Costa da Silva¹, João Ladislau B. Lopes^{2,3}, Ana Marilza Pernas¹

¹Centro de Desenvolvimento Tecnológico – Universidade Federal de Pelotas (UFPel)
Pelotas – RS – Brasil

²Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre – RS – Brasil

³Câmpus Visconde da Graça - Instituto Federal Sul-Rio-Grandense – (IFSul)
Pelotas – RS – Brasil

{lcdsilva, marilza}@inf.ufpel.edu.br, jlblopes@inf.ufrgs.br

Abstract. *A situation corresponds to an interpretation of contexts elements, relating to each other in order to provide any valid data in a specific time interval. One strategy for identifying situations is the use of semantic rules defined based on a context model specified with ontologies. Thus, it becomes possible reasoning about these valid information for certain situations of interest. In this sense, the central contribution of this paper is the possibility of a dynamic processing of semantic rules for identifying situations in an adaptative educational environment adapt on the Web.*

Resumo. *Uma situação consiste na interpretação de elementos de contextos, relacionando cada um de forma a prover alguma informação válida em um intervalo de tempo específico. Uma das estratégias para identificação de situações consiste no emprego de regras semânticas definidas com base em um modelo de contexto especificado com ontologias. Com isso, torna-se possível o raciocínio a respeito destas informações válidas para determinadas situações de interesse. Assim, o presente trabalho apresenta como contribuição central a viabilização de um processamento dinâmico de regras semânticas para a identificação de situações de alunos que utilizam um ambiente educacional adaptativo na Web.*

1. Introdução

A consciência de situação pode ser considerada uma particularização da consciência de contexto, onde situações são vistas como contextos logicamente ligados (ANAGNOSTOPOULOS, 2006). Uma situação consiste da interpretação de elementos de contextos, relacionando cada um de forma a prover alguma informação válida em um intervalo de tempo específico (LOPES et al., 2014).

Para possibilitar o processamento computacional dos dados contextuais e a consequente identificação de situações, estes devem ser representados de uma forma que seja processável por máquina, definindo um modelo que seja capaz de representar e viabilizar o processamento dos dados relativos aos contextos. Nesse sentido, o

raciocínio sobre o contexto permite a identificação de situações, sendo que uma das possibilidades de implementação do raciocínio é o emprego de regras (PERNAS, 2012).

O presente trabalho utiliza como cenário de aplicação ambientes educacionais adaptativos na Web, empregando neste cenário um modelo de contexto do aluno baseado em ontologias. Com isso, o trabalho tem como objetivo viabilizar a identificação da situação atual dos alunos, utilizando regras semânticas no processamento das informações contextuais.

O artigo está estruturado da seguinte forma: a seção 2 apresenta os trabalhos relacionados. Na seção 3 é descrito o cenário educacional e a rede de ontologias que modela o contexto do aluno. Na seção 4 é apresentada a prototipação realizada. A seção 5 apresenta as considerações finais e os trabalhos futuros.

2. Trabalhos Relacionados

Existem vários trabalhos correlatos que utilizam ontologias tanto para modelagem quanto para raciocínio sobre o contexto. Destes, destacam-se três que tratam especificamente sobre ontologias para modelagem de situação: (i) O'Brien (2009) que utiliza ontologias específicas a aplicações sensíveis à localização; (ii) Baumgartner et al. (2009), onde a ontologia tem foco nos sistemas de trânsito, sendo necessário detectar o momento específico em que os eventos ocorrem no sistema. Assim, conceitos de alto nível são especializados para modelagem da dimensão tempo; e (iii) Matheus et al. (2005) que define uma ontologia genérica para modelagem de situação e a partir dela outros conceitos podem ser definidos, possibilitando a extensão para cenários específicos. A ontologia utilizada nesse trabalho reusa alguns conceitos vindos das propostas de Baumgartner et al.(2009) e Matheus et al.(2005), principalmente por apresentarem ontologias de fácil reuso e conceitos bem fundamentados.

3. Foco do Desenvolvimento

O AdaptWeb® (Ambiente de Ensino-Aprendizagem Adaptativo na Web) (OLIVEIRA et al., 2003) foi empregado como cenário de aplicação deste trabalho.

Para o desenvolvimento do modelo de contexto que representa a situação vivenciada pelo aluno foram considerados três domínios, conforme mostra a Figura 1: (i) domínio do aluno, o qual representa as informações de contexto do aluno (ontologia Aluno) e a sua situação de aprendizagem (ontologia Situação); (ii) domínio educacional, onde são representados os recursos que compõem uma disciplina; e (iii) domínio tecnológico, que modela as informações do ambiente físico e recursos tecnológicos.

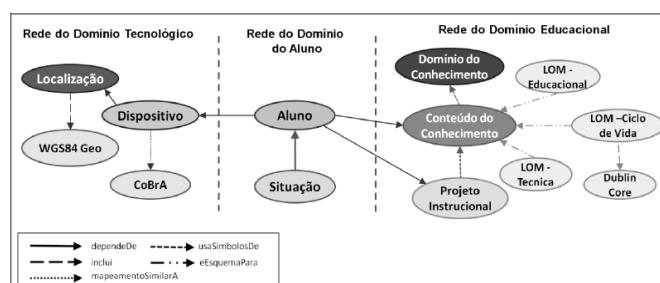


Figura 1 – Rede de Ontologias

4. Prototipação

Foi desenvolvido um protótipo para o. Em trabalhos anteriores com o mesmo foco de desenvolvimento (PERNAS, 2012), as regras para identificação de situações eram processadas de forma estática através de ferramentas como o Protégé (protege.stanford.edu). Neste trabalho buscou-se viabilizar o processamento dinâmico das regras semânticas para identificação das situações dos alunos usuários do ambiente AdaptWeb®, para tanto foi desenvolvido um protótipo, cujas tecnologias empregadas e a implementação realizada são descritas nessa seção.

A API Jena (jena.apache.org) foi utilizada como interface de programação para manipulação de ontologias, integrando o processamento de regras semânticas desenvolvidas, na perspectiva da linguagem de programação Java (www.java.com). O raciocínio sobre ontologias torna-se possível através de linguagens que agregam mecanismos de inferência e especificações com suporte a OWL (*Web Ontology Language*) (<http://www.w3.org/TR/owl-features/>). Nesse sentido, a linguagem SWRL (*Semantic Web Rule Language*) (<http://www.w3.org/Submission/SWRL/>), tem o objetivo de padronizar a definição de regras em ontologias, sendo a linguagem para definição de regras padrão recomendada pelo W3C. O raciocinador Pellet (pellet.owldl.com) foi integrado ao protótipo por possuir suporte as regras SWRL e sua possibilidade de integração com a API Jena.

Com relação à implementação do protótipo, foram definidas onze situações para compreensão do funcionamento de um sistema adaptativo à situação do aluno. Para cada uma das situações, existe uma regra, desenvolvida em SWRL, que a determina.

Para utilizar os dados contextuais contidos na ontologia, inicialmente é necessário o carregamento desta através da API Jena, sendo assim possível criar um modelo ontológico. A validação dos dados instanciados é feita através da realização de testes nas regras existentes, usando dados reais de alunos em disciplinas do AdaptWeb®. Durante a criação do modelo ontológico foi definido o raciocinador Pellet para a realização das inferências sobre as instâncias de alunos e suas respectivas situações. Logo após a criação do modelo ontológico é lido do arquivo OWL que contém os dados instanciados da ontologia. Através de uma consulta SPARQL (www.w3.org/TR/rdf-sparql-query/) são retornados todos os estudantes presentes na ontologia, o que torna possível sua seleção através da interface do protótipo, conforme mostra a Figura 2.

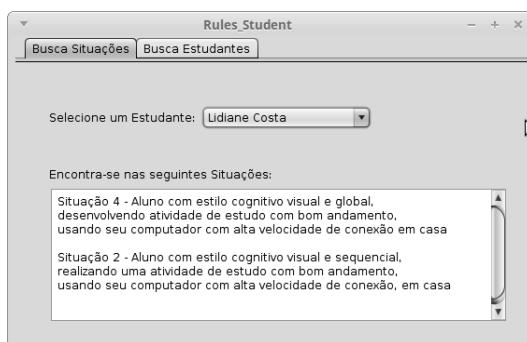


Figura 2 - Interface da Prototipação

No momento em que um estudante é selecionado, outra consulta SPARQL é disparada para a identificação das situações nas quais este aluno se encontra. Cabe salientar que esta consulta depende das inferências realizadas pelo raciocinador utilizando as regras SWRL. Na Listagem 1 é descrita uma regra SWRL, aplicada aos dados da instância aluno que configuram a situação em que a mesma se encontra.

Listagem 1. Regra em SWRL

```
Aluno(?x) ∧ estioloCaptacao (?x, visual) ∧ estioloEntendimento (?x, global) ∧ faz
(?x, ?y) ∧ Estudo(?y) ∧ bomDesempenho (?x,?y) ∧ usa (?x, ?z) ∧
ComputadorPessoal(?z) ∧ temConexao (?z, alta) ∧ localizadoEm (?x, casa) =>
temSituacao (?x, S_04)
```

5. Conclusões

O presente trabalho explora uma abordagem baseada em regras relacionadas à rede de ontologias utilizada para representação e processamento do contexto. Na avaliação desta abordagem foi empregado um cenário de aplicação na área educacional. Para validação e testes do uso das regras semânticas foi desenvolvido um protótipo para identificação de situações dos alunos usuários do ambiente AdaptWeb®. Os resultados obtidos com a execução do protótipo permitiram comprovar a viabilidade desta abordagem.

Na continuidade desta pesquisa, os seguintes trabalhos futuros podem ser realizados: (i) adicionar ao protótipo a funcionalidade de ativar e desativar regras dinamicamente; e (ii) empregar a proposta de uso de regras semânticas para identificação de situações em outros domínios.

Referências

- Anagnostopoulos, C.B., et al. “Situation Awareness: Dealing with Vague Context”. In IEEE International Conference on Pervasive Services, Jun. 2006.
- Baumgartner, N. et al. “BeAware! Situation awareness, the ontology-driven way”. Data & Knowledge Engineering, v. 69, issue 11, p. 1181-1193, 2010.
- Lopes, J. et al. “A Middleware Architecture for Dynamic Adaptation in Ubiquitous Computing”. Journal of Universal Computer Science, v.20, n.9, p.1327–1351, sep 2014.
- Matheus, C. et al. “An Assistant for Higher-Level Fusion and Situation Awareness”. In: Spie Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications, 2005. Proceedings... Orlando, FL, p. 75-85.
- O’Brien, P. “An Ontology for Mobile Situation Aware Systems”. Australian Journal of Information Systems, 2009.
- Oliveira, J. P. M. et al. “AdaptWeb: um ambiente para ensino-aprendizagem adaptativo na web”. Educar em revista, n.107, p.175–198, 2003.
- Pernas, A. M. “Sensibilidade à Situação em Sistemas Educacionais na Web”. 2012. 164p. Tese (Doutorado em Computação) - UFRGS, Porto Alegre, RS.

Laboratório Remoto de Redes de Computadores

Alexander M. Diaczenko¹, Vitor S. Brixius¹, Leandro J. Cassol¹, Luís C. M. Caruso¹, Taciano A. Rodolfo¹, Vanderson da Silva¹

¹Curso Superior de Tecnologia em Redes de Computadores

Faculdade de Tecnologia SENAI Porto Alegre

Assis Brasil, 8450 – 91.140-000 – Porto Alegre – RS – Brasil

diaczenko@gmail.com, vitorbrixius@hotmail.com,
leandro.cassol@senairs.org.br, luis.caruso@senairs.org.br,
taciano.rodolfo@senairs.org.br, vandersilvio.silva@senairs.org.br

Abstract. *The Remote Laboratory of the Faculty of Technology SENAI Porto Alegre was developed to allow rapid prototyping of computer networks at distance. This project is being performed and installed on the dependencies of this faculty. Hardware and software are under test and soon it is expected that both are fully validated and integrated.*

Resumo. *O Laboratório Remoto da Faculdade de Tecnologia SENAI Porto Alegre foi desenvolvido para permitir a prototipação rápida de redes de computadores à distância. Este projeto está sendo realizado e instalado nas dependências desta faculdade. Hardware e software estão sob teste e para breve espera-se que ambos estejam completamente validados e integrados.*

1. Introdução

A Faculdade de Tecnologia SENAI Porto Alegre oferece formação tecnológica onde a experiência profissional projetada para o egresso é antecipada para o ambiente acadêmico e aplicada como metodologia de ensino durante toda extensão do aprendizado. Além disso, desde sua criação esta Instituição faz parte da academia Cisco, oferecendo aos seus alunos dos cursos superiores de Telecomunicações e de Redes de Computadores o curso preparatório para a certificação CCNA¹.

Os laboratórios de redes de computadores da Faculdade são compartilhados entre os cursos técnicos, tecnológicos e de pós-graduação. Há, portanto, grande demanda por conexões e desconexões físicas de cabos que interligam os equipamentos. Tais conexões e desconexões são feitas constantemente, o que gera uma necessidade extra por revisões e manutenções periódicas.

O projeto do laboratório remoto é a proposta de um laboratório de redes de computadores automatizado. Comandado por uma solução que combina hardware e software, as conexões e desconexões entre as interfaces são efetuadas sem a intervenção humana. A vantagem é que o ambiente pode ser acessado pela Web, de qualquer local e hora. Outra grande vantagem é que a experiência é muito próxima da experiência real, com os problemas e limitações encontrados nos cenários reais.

¹ Cisco Certified Network Academy– certificação Cisco para profissionais de redes de computadores.

2. Arquitetura

No topo da abstração deste projeto temos vários usuários em diferentes locais interagindo via *browser* com *data centers* virtuais. Todo o equipamento real, que possibilita as interações, está concentrado em dois ou três bastidores em uma sala com acesso à Internet. A peça chave é um servidor que fornece a cada usuário ferramentas para projetar a rede e traduz as ações do usuário em conexões físicas reais.

Nos bastidores do ambiente real são instalados PCs rodando Linux, equipamentos Cisco como *switches*, rotadores com módulos seriais V.35, *access points* Wi-Fi, *firewalls* ASA. Para a automação deste ambiente são necessários equipamentos matrizes de conexões Ethernet, serial V.35 e console. Para a matriz Ethernet a solução é imediata, já que esta é a função de uma *switch*. Já para as conexões seriais esta matriz não existe, então optou-se por construir duas matrizes *crossconnect*, fazendo uso de FPGAs, com a capacidade de oferecer 10 enlaces V.35 e interligar 60 portas console. A este equipamento denominou-se comutador serial (CS).

Os equipamentos que realizarão as funções e conexões desejadas são “visíveis” aos usuários, ao passo que os equipamentos destinado à estrutura e operação do sistema são “invisíveis”. Para as conexões Ethernet um *switch* gerenciável invisível (SGI) é alocado. Cada conexão Ethernet no ambiente virtual ocupa duas portas neste *switch* e dois cabos, que vão até os equipamentos de serviço que o usuário conectou virtualmente. As conexões estruturais Ethernet utilizam um *switch* não-gerenciável invisível (SNGI).

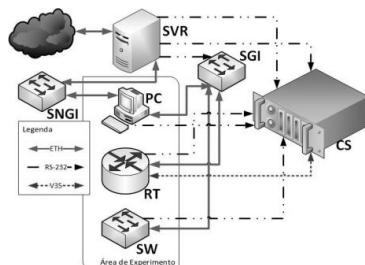


Figura 1. Diagrama simplificado do Laboratório Remoto de Redes

Na Figura 1 é apresentado um diagrama simplificado do Laboratório Remoto, exemplificando os equipamentos e as conexões. O retângulo interno delimita a área visível ao usuário. Aqui o usuário não percebe o PC, mas somente as máquinas virtuais hospedadas neste. As conexões Ethernet de serviço vão do retângulo demarcado ao SGI, onde ocorrem as conexões do usuário. As conexões estruturais vão até o SNGI.

As conexões V.35 são destinadas apenas ao serviço, sendo utilizadas duas portas V.35, uma ECD e uma ETD, no Comutador Serial, além de dois cabos compatíveis do CS até os roteadores, para realizar os enlaces desenhados virtualmente pelo usuário.

Das conexões de console, três são para a parte “invisível” e as demais para a “visível”. As consoles “invisíveis” partem do servidor, uma para a porta console do switch gerenciável invisível, outra para a console do comutador serial e a terceira para um das portas da matriz de comutação consoles do próprio comutador serial.

3. Funcionamento

O programa de controle ou Sistema do Laboratório Remoto (SLR) foi desenvolvido em PHP e oferece acesso a um administrador e a vários alunos. O administrador tem funções de cadastrar alunos e equipamentos de serviço ao *pool* de serviço. Ao aluno são oferecidas as funções de alocação de horários para seus experimentos e a alocação de equipamentos disponíveis no *pool* de serviço nos horários desejados. Um Ambiente Virtual contendo os cursos e material didático, com sugestões de topologias, é também acessível a partir do SLR.

O aluno deve marcar dia e hora, tempo necessário para seus experimentos e reservar equipamentos conforme a disponibilidade. No dia e hora marcados os equipamentos aguardarão pelo acesso do aluno. Através do *browser*, por meio de uma interface que lembra o Cisco Packet Tracer [Cisco 2015], o aluno irá posicionar no cenário os equipamentos reservados e montar a topologia desejada. Depois de pronta a topologia, o aluno irá conectar a console de um de seus PCs à console do equipamento que deseja configurar. Com cliques na imagem do PC o aluno abre um terminal, que o conecta ao equipamento alvo e por onde pode então configurá-lo. Depois de configurados todos os equipamentos, a topologia estará pronta para uso. As conexões e configurações podem ser refeitas conforme a necessidade. Após o SLR ocupa-se de limpar as configurações de todos os equipamentos.

4. Comutador serial

O comutador serial foi implementado fazendo uso de FPGAs [Xilinx 2015]. Por escolha de projeto, uma mesma placa mãe, contendo um único FPGA, pode atuar como um comutador de consoles ou V.35. O CS é um sub-bastidor fechado de 19" e 4U de altura, contendo duas placas mãe em planos sobrepostos, uma para a matriz console e outra para a matriz V.35. Uma placa auxiliar fornece a porta de console para permitir o controle do *crossconnect* de ambas as placas mãe.

As interfaces de acesso ao CS são placas filhas que adaptam os níveis elétricos dos padrões utilizados nas consolas e nas portas seriais V.35 ao nível lógico LVTTI [Jedec 2015] utilizado pela placa mãe. Cada placa mãe tem 10 conectores frontais e 10 traseiros. Cada conector possui 12 linhas de sinal, que são as entradas da matriz de comutação. Quando configurada para matriz console apenas 6 destas linhas de sinal são utilizadas atendendo 3 portas consola em uma única placa filha. Quando a matriz é configurada para V.35 todos os 12 sinais são usados para uma única porta.

5. O SLR e as Topologias Virtuais

O SLR atua sobre uma sessão de um usuário primeiramente carregando as configurações *default* nos equipamentos de rede. Para esta função, a terceira conexão de console do servidor mantida em uma das portas da matriz de consoles é conectada à porta que liga a matriz ao equipamento que se deseja reconfigurar. Estabelecida a conexão, um arquivo com a configuração *default*, associado ao equipamento alvo, é transmitido realizando a limpeza e, por fim, a conexão de console é desfeita. Esta ação é realizada para cada um dos equipamentos de rede configuráveis. Existe um atrelamento entre as portas do SGI e as portas de serviço Ethernet de cada equipamento, bem como entre as portas do CS e as portas console de serviço. Ao cadastrar um equipamento no *pool* de equipamentos de serviço, o administrador deve obrigatoriamente, para cada uma

das portas Ethernet, que desejar possibilitar conexão em serviço, indicar em que número de porta do SGI aquela estará conectada. O mesmo deve ocorrer para as portas console, cada equipamento cadastrado deverá incluir o número da porta console da matriz de consoles à qual está vinculada. Estas vinculações deverão ser cuidadosamente obedecidas por meio de conexão a cabo e não deverão ser modificadas sem a consequente alteração do registro de configuração do equipamento.

Quando do estabelecimento de um enlace, o SLR captura da topologia virtual o tipo de interface, o número das portas envolvidos no SGI ou CS e com estes dados pode formar comandos para a comutação tanto no SGI quanto no CS. O número das portas é buscado da descrição de cada um dos equipamentos fim criado quando da última alteração de seu cadastro.

Uma solução mais elaborada foi necessária na conexão de console da máquina virtual (VM) que o usuário aloca em sua topologia para acesso à configuração de seus ativos de rede. O caminho desta conexão é um acesso SSH do SLR, através do SNGI até a referida VM. Daí é estabelecida uma sessão console com o equipamento a ser configurado. Foi escolhido o AjaxTerm para rodar no SLR e oferecer ao usuário remoto a emulação de um acesso via terminal à VM real. Um menu de opções é oferecido ao usuário para limitar resultados não previstos na topologia ou configurações espúrias sobre a VM, incluindo seu hospedeiro.

6. Comentários Finais

O desenvolvimento deste projeto revelou sua multidisciplinaridade estimulando diversas interações e oferecendo oportunidades. As dificuldades iniciais passam a oferecer, após sua solução inicial, plataformas para evolução das funcionalidades do próprio Laboratório Remoto, mas também para exercício das disciplinas envolvidas. Assim, a placa mãe com sua grande generalidade permite experimentar com diversas aplicações de FPGA para aprendizado direto dos alunos. As diversas soluções encontradas em software, exigindo alguma criatividade, representam um espaço para os cursos de Sistemas Embarcados e Análise e Desenvolvimento de Sistemas experimentarem e inovarem e um cenário com responsabilidade de produção totalmente observável dentro de ambientes acadêmicos.

Referências

- Cisco Packet Tracer. Capturado em: <http://www.packettracernetwork.com>, Julho 2015.
- Xilinx, Inc. “Spartan-3E FPGA Family Data Sheet”. Capturado em: http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf, Julho 2015.
- Jedec – Global Standards for the Microelectronics Industry. “Addendum No. 1 to JESD8: Interface Standard for Low Voltage TTL- Compatible (LVTTL) VLSI Digital Circuits”. Capturado em: <http://www.jedec.org/standards-documents/docs/jesd-8-1>, Julho 2015.

O problema da padronização de interfaces norte no paradigma SDN

Kazuki Yokoyama¹, Alexsander de Souza¹, Sérgio Cechin¹

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{kmyokoyama, asouza, cechin}@inf.ufrgs.br

Abstract. *SDN networks have already become a reality and its adoption grows each day. Therefore, many standardization efforts of related technologies have been made, but no standard has been defined to a special class of them: northbound interface. The proposal of this paper is to present the main concepts regarding SDN model specifically those related to the development of northbound interface and about the working group created by ONF to develop northbound interface APIs.*

Resumo. *Redes SDN já se tornaram uma realidade e sua adoção vem crescendo a cada dia. Nesse sentido, vários esforços de padronização de tecnologias relacionadas foram feitos, porém nenhum padrão foi definido para uma classe especial delas: a interface norte. A proposta deste trabalho é apresentar os principais conceitos envolvidos no modelo SDN especificamente aqueles relacionados ao desenvolvimento de interfaces norte e sobre o grupo de trabalho criado pela ONF para desenvolver APIs de interface norte.*

1. Introdução

As redes tradicionais possuem o plano de controle, que toma decisões sobre o tráfego de dados, e o plano de dados, que executa as decisões tomadas, coexistindo no mesmo equipamento. O paradigma SDN tem como principal característica a separação dos planos, implementando a lógica de controle em um controlador logicamente centralizado e a manipulação do tráfego ficando a cargo dos *switches* [Kreutz et al. 2015]. Essa separação permite maior flexibilidade na criação e implementação de novas políticas de gerência de redes. A adoção do modelo SDN é promovida e organizada pela ONF (*Open Networking Foundation*) que desenvolve padrões abertos.

Na arquitetura SDN pode-se destacar duas principais interfaces: sul e norte. A interface sul (*southbound interface*) provê ao controlador uma camada de abstração da infraestrutura de rede. O protocolo padronizado OpenFlow [McKeown et al. 2008] é seu principal representante e tem sido largamente adotado pela indústria. A interface norte (*northbound interface - NBI*), por sua vez, tem a função de fornecer uma abstração do conjunto de instruções de baixo nível utilizado pelos protocolos de interface sul para as aplicações de rede. Apesar de sua importância, não há padrão definido.

2. A interface norte

A interface norte localiza-se entre o controlador de rede SDN e as aplicações que o utilizam. Ela deve fornecer uma visão abstrata da rede e permitir uma expressão direta de seu comportamento e requisitos [ONF 2013b]. Dessa forma, diversos sistemas,

como os de orquestração de nuvem e os próprios operadores de rede, poderiam se comunicar em alto nível com a infraestrutura disponível sem preocuparem-se com seus detalhes de programação de baixo nível [Kreutz et al. 2015]. Com a interface norte, serviços de configuração, gerência ou provisionamento poderiam ser rapidamente postos em utilização e dinamicamente adequados às necessidades da rede.

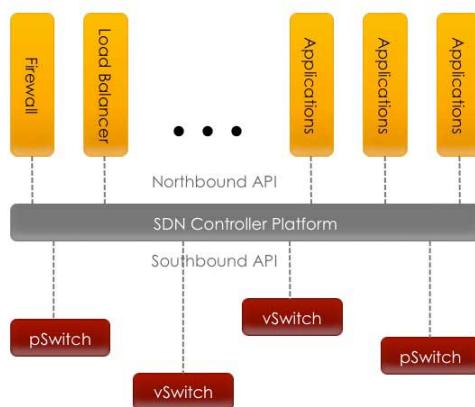


Figura 1. Interfaces do modelo SDN [Guis 2013]

Na Figura 1, podem ser vistas as interfaces norte e sul do modelo SDN. Várias aplicações podem utilizar o controlador SDN a fim de gerenciar a rede através das interfaces norte. O controlador utiliza então os protocolos de interface sul para coordenar os elementos de encaminhamento do plano de dados.

3. Padronização

Atualmente, cada fabricante e fornecedor de soluções SDN disponibiliza a sua própria interface norte. A maioria dos controladores fornece alguma API de alto nível para as aplicações que o utilizarão. O resultado é um ecossistema de aplicações dependentes de APIs fragmentadas entre diversos fabricantes. Uma das maiores propostas das redes SDN é uma abstração de rede que permita o desenvolvimento de aplicações que possam ser empregadas independentemente da estrutura física da rede. Isso pode ser dificultado se uma padronização da interface norte não for concluída.

Muita discussão tem sido levantada em torno do assunto. Alguns argumentam que a NBI é uma peça fundamental da arquitetura SDN e ressaltam a importância de ser padronizada [Salisbury 2012], outros afirmam que talvez seja muito cedo e que o melhor caminho pode ser a padronização a partir das implementações [Dix 2013]. Por fim, é possível que o padrão surja a partir da solução proprietária que melhor se destaca [Guis 2013], assim como aconteceu com o OpenFlow.

Em outubro de 2013, foi criado um grupo de trabalho da ONF, *North Bound Interface Working Group - NBI-WG*, a fim de discutir e propor uma API para interface norte.

Os objetivos iniciais do grupo são esclarecer as questões relativas às interfaces norte e contribuir com os desenvolvedores na definição de um padrão aberto [ONF 2013].

Segundo [Raza 2013], um consenso sobre o padrão NBI é uma peça essencial para o ecossistema de aplicações SDN. Além disso, uma API fragmentada e não padronizada ocupa significativamente o tempo de desenvolvimento dos sistemas, tempo que poderia ser usado para desenvolver aplicações diferenciadas e de melhor qualidade.

Não há uma única API que sirva a todos propósitos. Diferentes tipos de aplicações podem requerer distintos tipos de APIs que podem ser encontradas ou não no mesmo controlador. Essa ideia de múltiplas APIs é ilustrada na Figura 2.

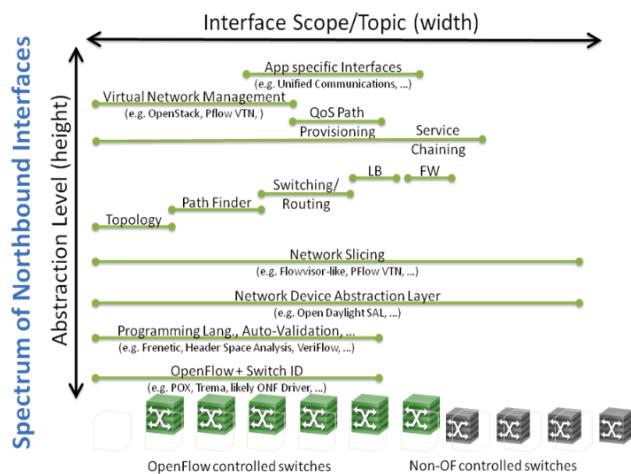


Figura 2. Diferentes tipos de interface norte [Raza 2013]

Na Figura 2, pode-se observar os diversos níveis de abstração possíveis. Diferentes aplicações necessitam de variados níveis de informações sobre a rede subjacente. Por exemplo, APIs que acessam diretamente o OpenFlow encontram-se em um nível mais baixo de abstração do que aquelas que lidam com ambientes de virtualização como o OpenStack. Ao mesmo tempo, tem-se as abrangências horizontais. Por exemplo, enquanto algumas APIs são destinadas especificamente a *switches* OpenFlow, outras podem ser empregadas em equipamentos que não suportem essa tecnologia.

É possível definir dois tipos de interface norte. O primeiro tem como público-alvo os desenvolvedores de aplicações SDN e consiste em um *kit* de desenvolvimento. O segundo é destinado aos usuários finais e provê uma espécie de virtualização da rede. O grupo de trabalho da ONF tem como objetivo tanto a definição de uma API para um conjunto básico de funcionalidades do controlador SDN, quanto de APIs para domínios específicos.

As APIs serão definidas como modelos de dados independentes de linguagem de implementação. O grupo de trabalho planeja a implementação na íntegra de pelo menos uma API assim definida. Essa implementação consistirá de duas partes, sendo uma do modelo de dados no lado do controlador SDN e a outra no lado que fará uso da API. Por

fim, a API será considerada com potencial para padronização se for implementada, ou seja, tornada código com sucesso e atender às exigências do mercado.

4. Conclusões

Como visto, as interfaces norte da arquitetura SDN foram desenvolvidas paralelamente por diversos fabricantes. Muitos fornecedores de controladores, abertos ou proprietários, também disponibilizam as suas próprias APIs.

O resultado é um ecossistema pouco integrado de aplicações SDN. Para que se usufrua de toda capacidade inovativa do modelo, é preciso que os desenvolvedores tenham uma referência ou um padrão a seguir. É possível argumentar que talvez a arquitetura SDN seja muito nova para se definir padrões mas, por outro lado, a ausência de um padrão pode significar atraso na adoção da tecnologia.

Uma das tarefas do grupo de trabalho ONF-WG é definir interfaces norte para alguns cenários de uso. No entanto, um modelo de interface com potencial para padronização ainda não foi apresentado.

Referências

- Dix, J. (2013). Clarifying the role of software-defined networking northbound APIs. <http://www.networkworld.com/article/2165901/lan-wan/clarifying-the-role-of-software-defined-networking-northbound-apis.html>. [Online; acesso em 10 jul. 2015].
- Guis, I. (2013). The sdn gold rush to the northbound api. <https://www.sdxcentral.com/articles/contributed/the-sdn-gold-rush-to-the-northbound-api/2012/11/>. [Online; acesso em 10 jul. 2015].
- Kreutz, D., Ramos, F. M. V., Verissimo, P., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(2):14–76.
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74.
- ONF (2013). Open networking foundation introduces northbound interface working group. <https://www.opennetworking.org/news-and-events/press-releases/1182-open-networking-foundation-introduces-northbound-interface-working-group>. [Online; acesso em 10 jul. 2015].
- Raza, S. e Lenrow, D. (2013). North bound interface working group (nbi-wg) charter. <https://www.opennetworking.org/images/stories/downloads/working-groups/charter-nbi.pdf>. [Online; acesso em 10 jul. 2015].
- Salisbury, B. (2012). The Northbound API - A Big Little Problem. <http://networkstatic.net/the-northbound-api-2/> [Online; acesso em 10 jul. 2015].