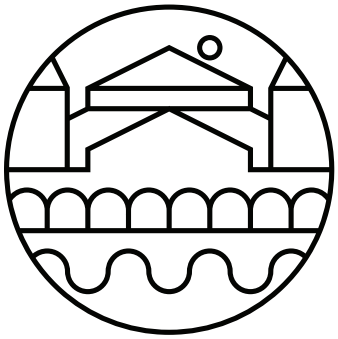
XXXV
SIMPÓSIO BRASILEIRO DE
REDES DE COMPUTADORES
E SISTEMAS DISTRIBUÍDOS

15 a 19 de maio de 2017
Belém - Pará

Anais XXII WGRS 2017



XXXV
SIMPÓSIO BRASILEIRO DE
REDES DE COMPUTADORES
E SISTEMAS DISTRIBUÍDOS

15 a 19 de maio de 2017
Belém - Pará

Anais do XXII WGRS 2017
Workshop de Gerência e Operação de Redes e Serviços

Editora
Sociedade Brasileira de Computação (SBC)

Organização
Augusto José Venâncio Neto (UFRN)
Cristiano Bonato Both (UFCSPA)
Ronaldo Alves Ferreira (UFMS)
Antônio Jorge Gomes Abelém (UFPA)
Eduardo Coelho Cerqueira (UFPA)

Realização
Sociedade Brasileira de Computação (SBC)
Universidade Federal do Pará (UFPA)
Laboratório Nacional de Redes de Computadores (LARC)

Copyright ©2017 da Sociedade Brasileira de Computação
Todos os direitos reservados

Capa: Catarina Nefertari (PCT-UFPA)
Produção Editorial: Denis Lima do Rosário (UFPA)

Cópias Adicionais:
Sociedade Brasileira de Computação (SBC)
Av. Bento Gonçalves, 9500- Setor 4 - Prédio 43.412 - Sala 219
Bairro Agronomia - CEP 91.509-900 - Porto Alegre - RS
Fone: (51) 3308-6835
E-mail: sbc@sbc.org.br

XXII Workshop de Gerência e Operação de Redes e Serviços (22: 2017: Belém, Pa).

Anais / XXII Workshop de Gerência e Operação de Redes e Serviços – WGRS; organizado por Antônio Jorge Gomes Abelém, Eduardo Coelho Cerqueira, Ronaldo Alves Ferreira, Augusto José Venâncio Neto, Cristiano Bonato Both - Porto Alegre: SBC, 2017

164 p. il. 21 cm.
Vários autores
Inclui bibliografias

1. Redes de Computadores. 2. Sistemas Distribuídos. I. Abelém, Antônio Jorge Gomes II. Cerqueira, Eduardo Coelho III. Ferreira, Ronaldo Alves IV. Neto, Augusto José Venâncio V. Both, Cristiano Bonato VI. Título.

Sociedade Brasileira da Computação

Presidência
Lisandro Zambenedetti Granville (UFRGS), Presidente
Thais Vasconcelos Batista (UFRN),Vice-Presidente

Diretorias
Renata de Matos Galante (UFGRS), Diretora Administrativa
Carlos André Guimarães Ferraz (UFPE), Diretor de Finanças
Antônio Jorge Gomes Abelém (UFPA), Diretor de Eventos e Comissões Especiais
Avelino Francisco Zorzo (PUC-RS), Diretor de Educação
José Viterbo Filho (UFF), Diretor de Publicações
Claudia Lage Rebello da Motta (UFRJ), Diretora de Planejamento e Programas Especiais
Marcelo Duduchi Feitosa (CEETEPS), Diretor de Secretarias Regionais
Eliana Almeida (UFAL), Diretora de Divulgação e Marketing
Roberto da Silva Bigonha (UFMG), Diretor de Relações Profissionais
Ricardo de Oliveira Anido (UNICAMP), Diretor de Competições Científicas
Raimundo José de Araújo Macêdo (UFBA), Diretor de Cooperação com Sociedades Científicas
Sérgio Castelo Branco Soares (UFPE), Diretor de Articulação com Empresas

Contato
Av. Bento Gonçalves, 9500
Setor 4 - Prédio 43.412 - Sala 219
Bairro Agronomia
91.509-900 – Porto Alegre RS
CNPJ: 29.532.264/0001-78
http://www.sbrc.org.br

Laboratório Nacional de Redes de Computadores (LARC)

Diretora do Conselho Técnico-Científico
Rossana Maria de C. Andrade (UFC)

Vice-Diretor do Conselho Técnico-Científico
Ronaldo Alves Ferreira (UFMS)

Diretor Executivo
Paulo André da Silva Gonçalves (UFPE)

Vice-Diretor Executivo
Elias P. Duarte Jr. (UFPR)

Membros Institucionais
SESU/MEC, INPE/MCT, UFRGS, UFMG, UFPE, UFCG (ex-UFPB Campus Campina Grande), UFRJ, USP, PUC-Rio, UNICAMP, LNCC, IME, UFSC, UTFPR, UFC, UFF, UFSCar, IFCE (CEFET-CE), UFRN, UFES, UFBA, UNIFACS, UECE, UFPR, UFPA, UFAM, UFABC, PUCPR, UFMS, UnB, PUC-RS, PUCMG, UNIRIO, UFS e UFU.

Contato
Universidade Federal de Pernambuco - UFPE
Centro de Informática - CIn
Av. Jornalista Anibal Fernandes, s/n
Cidade Universitária
50.740-560 - Recife - PE
http://www.larc.org.br

Organização do SBRC 2017

Coordenadores Gerais
Antônio Jorge Gomes Abelém (UFPA)
Eduardo Coelho Cerqueira (UFPA)

Coordenadores do Comitê de Programa
Edmundo Roberto Mauro Madeira (UNICAMP)
Michele Nogueira Lima (UFPR)

Coordenador de Palestras e Tutoriais
Edmundo Souza e Silva (UFRJ)

Coordenador de Painéis e Debates
Luciano Paschoal Gaspary (UFRGS)

Coordenadores de Minicursos
Heitor Soares Ramos (UFAL)
Stênio Flávio de Lacerda Fernandes (UFPE)

Coordenadora de Workshops
Ronaldo Alves Ferreira (UFMS)

Coordenador do Salão de Ferramentas
Fabio Luciano Verdi (UFSCar)

Comitê de Organização Local
Adailton Lima (UFPA)
Alessandra Natasha (CESUPA)
Davis Oliveria (SERPRO)
Denis Rosário (UFPA)
Elisangela Aguiar (SERPRO)
João Santana (UFRA)
Josivaldo Araújo (UFPA)
Marcos Seruffo (UFPA)
Paulo Henrique Bezerra (IFPA)
Rômulo Pinheiro (UNAMA)
Ronedo Ferreira (META)
Thiêgo Nunes (IFPA)
Vagner Nascimento (UNAMA)

Comite Consultivo
Allan Edgard Silva Freitas (IFBA)
Antonio Alfredo Ferreira Loureiro (UFMG)
Christian Esteve Rothenberg (UNICAMP)
Fabíola Gonçalves Pereira Greve (UFBA)
Frank Augusto Siqueira (UFSC)
Jussara Marques de Almeida (UFMG)
Magnos Martinello (UFES)
Antonio Marinho Pilla Barcellos (UFRGS)
Moisés Renato Nunes Ribeiro (UFES)
Rossana Maria de Castro Andrade (UFC)

Mensagem dos Coordenadores Gerais

Sejam bem-vindos ao 35o Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2017) e a acolhedora cidade das mangueiras - Belém / Pará.

Organizar uma edição do SBRC pela segunda vez no Norte do Brasil é um desafio e um privilégio por poder contribuir com a comunidade de Redes de Computadores e Sistemas Distribuídos do Brasil e do exterior. O SBRC se destaca como um importante celeiro para a discussão, troca de conhecimento e apresentação de trabalhos científicos de qualidade.

A programação do SBRC 2017 está diversificada e discute temas relevantes no cenário nacional e internacional. A contribuição da comunidade científica brasileira foi de fundamental importância para manter a qualidade técnica dos trabalhos e fortalecer a ciência, tecnologia e inovação no Brasil.

Após um cuidadoso processo de avaliação, foram selecionados 77 artigos completos organizados em 26 sessões técnicas e 10 ferramentas para apresentação durante o Salão de Ferramentas. Além disso, o evento contou com 3 palestras e 3 tutoriais proferidos por pesquisadores internacionalmente renomados, 3 painéis de discussões e debates, todos sobre temas super atuais, 6 minicursos envolvendo Big Data, sistemas de transportes inteligentes, rádios definidos por software, fiscalização e neutralidade da rede, mecanismos de autenticação e autorização para nuvens computacionais e comunicação por luz visível, bem como 10 workshops.

O prêmio “Destaque da SBRC” e uma série de homenagens foram prestadas para personalidades que contribuíram e contribuem com a área. O apoio incondicional da SBC, do LARC, do Comitê Consultivo da SBRC e da Comissão Especial de Redes de Computadores e Sistemas Distribuídos da SBC foram determinantes para o sucesso do evento. A realização do evento também contou com o importante apoio do Comitê Gestor da Internet no Brasil (CGI.br), do CNPq, da CAPES, do Parque de Ciência e Tecnologia Guamá, da Connecta Networking, da Dantec Telecom, da RNP e do Google. Nosso especial agradecimento à Universidade Federal do Pará (UFPA) e ao Instituto Federal do Pará (IFPA) pelo indispensável suporte à realização do evento.

Nosso agradecimento também para os competentes e incansáveis coordenadores do comitê do programa (Michele Nogueira/UFRP – Edmundo Madeira/UNICAMP), aos coordenadores dos minicursos (Stênio Fernandes/UFPE – Heitor Ramos/UFAL), ao coordenador dos workshops (Ronaldo Ferreira/UFMS), ao coordenador de painéis e debates (Luciano Gaspary/UFRGS), ao coordenador do Salão de Ferramentas (Fabio Verdi/UFSCar) e ao coordenador de palestras e tutoriais (Edmundo Souza e Silva/UFRJ). Destacamos o excelente trabalho do comitê de organização local coordenado por Denis do Rosário.

Por fim, desejamos a todos uma produtiva semana em Belém.

Antônio Abelém e Eduardo Cerqueira

Coordenadores Gerais do SBRC 2017

Mensagem do Coordenador de Workshops

É com grande prazer que os convido a prestigiar os workshops do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC) nos dias 15, 16 e 19 de maio de 2017. Tradicionalmente, os workshops abrem e fecham a semana do SBRC e são responsáveis por atrair uma parcela expressiva de participantes para o Simpósio. Como coordenador de workshops, dividi com os coordenadores gerais do SBRC a nobre tarefa de selecionar os workshops que melhor representam a comunidade e que fortaleçam novas linhas de pesquisa ou mantenham em evidência linhas de pesquisa tradicionais.

Em resposta à chamada aberta de workshops, recebemos dez propostas de alta qualidade, das quais nove foram selecionadas. Além disso, mantivemos a longa colaboração com a RNP por meio da organização do WRNP, que já é uma tradição na segunda e terça-feira da semana do SBRC. Dentre as propostas aceitas, sete são reedições de workshops tradicionais do SBRC que já são considerados parte do circuito nacional de divulgação científica nas várias subáreas de Redes de Computadores e Sistemas Distribuídos, como o WGRS (Workshop de Gerência e Operação de Redes e Serviços), o WTF (Workshop de Testes e Tolerância a Falhas), o WCGA (Workshop em Clouds, Grids e Aplicações), o WP2P+ (Workshop de Redes P2P, Dinâmicas, Sociais e Orientadas a Conteúdo), o WPEIF (Workshop de Pesquisa Experimental da Internet do Futuro), o WoSiDA (Workshop de Sistemas Distribuídos Autonômicos) e o WoCCES (Workshop de Comunicação de Sistemas Embarcados Críticos). Como novidade, teremos dois novos workshops com programação diversificada e grande apelo social, o CoUrb (Workshop de Computação Urbana) e o WTICp/D (Workshop de TIC para Desenvolvimento).

Temos certeza que 2017 será mais um ano de sucesso para os workshops do SBRC pelo importante papel de agregação que eles exercem na comunidade científica de Redes de Computadores e Sistemas Distribuídos no Brasil.

Aproveitamos para agradecer o apoio recebido de diversos membros da comunidade e, em particular, a cada coordenador de workshop, pelo brilhante trabalho. Como coordenador dos workshops, agradeço imensamente o apoio recebido da Organização Geral do SBRC 2017.

Esperamos que vocês aproveitem não somente os workshops, mas também todo o SBRC e as inúmeras atrações de Belém.

Ronaldo Alves Ferreira

Coordenador de Workshops do SBRC 2017

Mensagem dos Coordenadores do XXII WGRS 2017

O Workshop de Gerência e Operação de Redes e Serviços chega à sua vigésima segunda edição. Ele tem representado, desde sua primeira edição em 1996, um fórum nacional importante para a troca de ideias, discussão e avaliação de projetos teóricos e práticos em sua área de concentração. A programação do XXII WGRS inclui 12 artigos completos selecionados pelo Comitê de Programa e uma palestras convidadas. Na primeira sessão os artigos tem seu principal foco trabalhos que se enquadram no campo de pesquisa das Redes Definidas por Software. Na segunda sessão, o foco principal se dá a contribuições no âmbito de gerenciamento de redes. Finalmente, a terceira sessão técnica tem como foco principal temas propostas relacionadas a áreas relacionadas a Internet das Coisas. Em complemento as sessões técnicas ocorrerá uma palestra convidada, que trata de um tema atual sobre gerenciamento e operação de redes e serviços. O professor Joel Rodrigues (Inatel) abordará os desafios e oportunidades no gerenciamento de redes IoT, sob o tema "IoT Network Management: Challenges and Opportunities". É gratificante constatar que o programa do XXII WGRS reflete a preocupação dos pesquisadores brasileiros em manterem projetos e pesquisa compatível com o desenvolvimento mundial da área de Gerência e Operação de Redes e Serviços.

Gostaríamos de agradecer a todos os membros do Comitê de Programa pelo seu compromisso em retornar avaliações de qualidade em um prazo justo. Sem a pronta atuação de cada membro o WGRS não teria acontecido. Agradeço de forma especial a Sociedade Brasileira de Computação (SBC), que sempre estive pronta para conversar sobre a melhor forma de organizar o Workshop. Finalmente, gostaria de agradecer aos organizadores do SBRC 2017 pelo excelente apoio oferecido aos coordenadores dos Workshops associados ao Simpósio pela organização do SBRC.

Augusto Venâncio Neto e Cristiano Bonato Both

Coordenadores WGRS

Comitê de Programa

- Alberto Schaeffer Filho (UFRGS)
- Aldri Luiz dos Santos (UFPR)
- Antonio Marcos Alberti (INATEL)
- Artur Ziviani (LNCC)
- Augusto Venâncio Neto (UFRN)
- Bruno Schulze (LNCC)
- Carlos Westphall (UFSC)
- Célio Neves de Albuquerque (UFF)
- Cristiano Bonato Both (UFCSPA)
- Daniel Fernandes Macedo (UFMG)
- Eduardo Cerqueira (UFPA)
- Fátima Duarte-Figueiredo (PUC Minas)
- Fernando Matos (UFPB)
- Joaquim Celestino Júnior (UECE)
- Joel Rodrigues (INATEL)
- Jéferson Campos Nobre (Unisinos)
- Juliano Araujo Wickboldt (UFRGS)
- Jussara Almeida (UFMG)
- Lisandro Zambenedetti Granville (UFRGS)
- Luiz Fernando Bittencourt (UNICAMP)
- Luis Henrique Costa (UFRJ)
- Luiz Henrique Andrade Correia (UFLA)
- Marcelo Marota (UFRGS)
- Marcelo Rubinstein (UERJ)
- Mauro Fonseca (PUCPR)
- Paulo André da Silva Gonçalves (UFPE)
- Ronaldo Ferreira (UFMS)
- Vinicius da Cunha M. Borges (UFG)

Sumário

XXII Workshop de Gerência e Operação de Redes e Serviços (WGRS)

SBRC 2017

Sessão Técnica 1 – Redes Definidas por Software

Analisando o Comportamento de Ambientes de Experimentação baseados em OpenFlow

Rodrigo A. de Oliveira[1], **Yona Lopes**[2], **Débora C. M. Saade**[2], **Natalia C. Fernandes**[1]

[1]Departamento de Engenharia de Telecomunicações – PPGEET
[2] Departamento de Ciência da Computação – Instituto de Computação
Laboratório MídiaCom
Universidade Federal Fluminense (UFF) – Niterói – RJ – Brasil

{rao,yona,debora,natalia}@midiacom.uff.br

Abstract. *The choice of a Software Defined Networking (SDN) experimentation environment proper to the kind of experiment is a frequent challenge for researchers because of the range of existing possibilities, peculiarities of each environment, and restricted budget for purchasing devices. The objective of this study is to analyze the different experimentation environments from the standpoint of the existing OpenFlow switch solutions, considering issues such as performance and costs. Several performance parameters were evaluated in environments with similar network configurations, but different OpenFlow switch, controller and host implementations. The proposal of the article is not to define the best solution, but to provide inputs for choosing the environment that best suits the research requirements, considering performance, controllability, and costs.*

Resumo. *A escolha de um ambiente de experimentação para Redes Definidas por Software (Software Defined Networking - SDN) adequado ao tipo de experimento é um desafio frequente para pesquisadores devido à gama de possibilidades existentes, peculiaridades de cada ambiente e orçamento restrito para compra de equipamentos. O objetivo deste estudo é analisar os diferentes ambientes de experimentação sob a ótica das soluções existente de switch OpenFlow, considerando questões como desempenho e custos. Vários parâmetros de desempenho foram avaliados em ambientes com configurações semelhantes de rede, mas implementações do switch OpenFlow, controlador e clientes diversas. A proposta do artigo não é afirmar qual a melhor solução, mas sim prover insumos para a escolha do ambiente que melhor se adapte às necessidades da pesquisa, considerando desempenho, controlabilidade e custos.*

1. Introdução

As Redes Definidas por *Software* (*Software Defined Networks* - SDN) constituem um novo paradigma, mudando a forma como as redes de computadores são criadas, modificadas e gerenciadas [Feamster et al. 2014]. Esse novo paradigma abre uma nova perspectiva em ambientes de controle lógico da rede e novas aplicações de rede, que passam a poder ser desenvolvidas de forma simples e livre dos limites da arquitetura atual da Internet. O protocolo OpenFlow [McKeown et al. 2008] surge neste ambiente como uma forma de padronização da configuração por meio de um controle unificado de comutadores, roteadores e pontos de acesso sem fio. Usando o OpenFlow, os dispositivos de rede

passam a conter uma interface de programação simples, que permite o acesso ao controle de fluxo, ou à tabela de comutação do *hardware*.

Dada a sua importância disruptiva, as SDNs e as aplicações de rede desenvolvidas baseadas no protocolo OpenFlow são hoje alvo de grandes investimentos em pesquisas. Estas pesquisas avançam para o estado da arte com soluções inovadoras que trazem diversos benefícios para o setor acadêmico e privado. No entanto, para que estas vantagens sejam de fato reais, a avaliação realizada precisa ser o mais consistente possível. De fato, a importância de testes conclusivos feitos em bons ambientes e de evolução no plano de dados SDN é tamanha que novas propostas, como a de Katta et al. [Katta et al. 2014], surgem na tentativa de aprimorar este plano de dados. Apesar da importância, é comum se deparar na literatura com análises limitadas ou resultados incorretos devido ao mal uso de ferramentas de análise, simulação e emulação.

Este trabalho visa analisar e verificar os pontos fortes e fracos de ambientes de experimentação OpenFlow, provendo insumos para que pesquisadores possam escolher adequadamente a ferramenta que será usada de acordo com as suas necessidades. Os testes realizados visam verificar o desempenho dos ambientes de experimentação, com foco na infraestrutura adotada e através dos resultados obtidos ponderar as características destes ambientes para contribuir na escolha de ambientes a serem usados por pesquisadores.

As próximas seções estão organizadas da seguinte forma: a Seção 2 trata dos trabalhos relacionados; a Seção 3 descreve os tipos de ambientes de experimentação e a dificuldade de escolha; a Seção 4 apresenta a avaliação de desempenho realizada, descrevendo os objetivos e a configuração dos ambientes de teste detalhadamente; a Seção 5 analisa os resultados obtidos; a Seção 6 apresenta uma estimativa de custos aproximada para cada ambiente de teste, assim como uma análise qualitativa dos ambientes de experimentação utilizados e por fim, a Seção 7 apresenta as conclusões e propostas de trabalhos futuros.

2. Trabalhos Relacionados

As SDNs apresentam um modelo de arquitetura inovador capaz de entregar provisionamento automatizado, virtualização e programação de rede para *data centers* e redes corporativas, ganhando continuamente terreno no mercado corporativo e em provedores de serviços em nuvem para *data center*. De fato, diversos artigos têm sido publicados apresentando meios de utilização do OpenFlow [Heller et al. 2010] [Das et al. 2010], o que demonstra a variedade de campos onde a separação entre o plano de dados do plano de controle tem grande utilidade.

No meio acadêmico, diversas pesquisas para aprimorar o conceito e entendimento da utilidade da tecnologia SDN são realizadas, abrangendo diferentes enfoques, como análises dos tipos de controladores, do protocolo OpenFlow [Macedo et al. 2015] e dos equipamentos de diferentes fornecedores que suportam este protocolo [Feamster et al. 2014]. Alguns trabalhos focam suas análises de desempenho em comparações sobre o comportamento dos *switches* quando utilizados com o protocolo OpenFlow e no modo *legacy*, avaliando se as implementações do protocolo OpenFlow, são, de fato, realizadas em *hardware* ou em instâncias de um *software switch* adaptadas ao mecanismo interno do dispositivo [Costa et al. 2016] .

Outras avaliações de desempenho de equipamentos OpenFlow focam em uma

camada de abstração de hardware convertendo dispositivos não habilitados para OpenFlow em dispositivos habilitados [Zal and Kleban 2014], com uso de plataforma de teste *OFLOPS*. Algumas avaliações apresentam somente a análise pontual do dispositivo OpenFlow instalado num desktop com sistema operacional *Linux* [Bianco et al. 2010], mensurando como é o comportamento do plano de dados neste cenário, sem comparação com outros meios possíveis de instalação de um dispositivo OpenFlow.

De maneira muito similar outros trabalhos envolvem o emulador Mininet, analisando a sua usabilidade [de Oliveira et al. 2014] ou verificando as suas limitações em diferentes ambientes [Keti and Askar 2015].

Os *testbeds* públicos têm crescido e estimulado inúmeros pesquisadores de rede para executar seus protótipos e experimentos. Com isso, pesquisas abrangentes sobre os desafios encontrados nestes *testbeds* são apresentados [Huang et al. 2016], porém, comparações que envolvam os diversos ambientes possíveis de simulações em OpenFlow com foco em desempenho, capacidade, custos e facilidades de uso, ainda são pouco exploradas.

Neste artigo avaliam-se diferentes meios de soluções de implementações de *switch* OpenFlow para definir quais são mais próprias para a realização de experimentos de novas ferramentas, ressaltando as suas limitações de desempenho e de custos. São feitas medidas que avaliam ambientes baseados em *switches* OpenFlow comerciais, implementados em computadores pessoais, emulados e ainda construídos em *testbeds* públicos. Diferentemente dos outros artigos, a proposta não é meramente avaliar o plano de dados OpenFlow, mas fornecer informações que sejam úteis na definição da plataforma de teste adequada a um experimento específico.

3. Tipos de Ambientes de Experimentação

O paradigma SDN tem ganhado força nos últimos anos e atraído cada vez mais a atenção da comunidade acadêmica e da indústria [Costa et al. 2016]. Com isso, muitos equipamentos de rede passaram a suportar o protocolo OpenFlow. Da mesma forma, o interesse em projetos de pesquisa relacionados ao uso do OpenFlow como *framework* para novas soluções também aumentou [Masoudi and Ghaffari]. Assim, o desenvolvimento de redes definidas por *software* chamou a atenção da comunidade científica nos últimos anos, principalmente devido à flexibilidade que o modelo de controle centralizado apresenta, o que facilita o desenvolvimento de soluções de acordo com a demanda da rede e/ou dos usuários [Ortiz et al. 2016].

Importante salientar que os resultados obtidos nos experimentos realizados são diretamente relacionados ao ambiente utilizado na experimentação. Por exemplo, o ambiente de experimentação tem um efeito notável sobre o tempo necessário para a construção de uma topologia [Keti and Askar 2015], para configuração de um fluxo, para comutação para a porta backup em caso de falha, etc. Além disso, existem diferenças significativas de desempenho e funcionalidade nas implementações em *hardware* do OpenFlow [Keti and Askar 2015]. Resultados inteiros podem ser invalidados caso experimentados em ambientes que não sejam próprios para a avaliação requerida. Caso o ambiente tenha uma limitação que afete diretamente os objetivos dos testes, falsos negativos podem ser gerados e uma boa solução pode ser descartada. Além disso, falsos positivos podem ser encontrados caso a escolha do ambiente de experimentação não seja pensada

previamente.

3.1. Ambiente OpenFlow Simulado e Emulado

Um simulador de redes SDN reproduz o comportamento dos equipamentos de uma rede OpenFlow, através de um modelo, para verificar o comportamento destes. Com isso, um sistema real pode ser avaliado numericamente a partir de um modelo de simulação. Este modelo precisa ser criteriosamente construído para representar o ambiente a ser simulado de forma fiel. Testes cuja a preparação em ambiente reais seja cara, custosa ou demorada têm na simulação uma forma viável de execução. Apesar dos vários simuladores de redes disponíveis, são poucos que oferecem suporte ao OpenFlow. O *Network Simulator* 3 (NS-3)[1] é um exemplo de simulador que é compatível com o OpenFlow, mas que usualmente não é tão utilizado quanto as outras ferramentas de análise devido à sua complexidade para programação do ambiente OpenFlow.

Um emulador de redes SDN é um sistema computacional que imita o funcionamento de *switches* OpenFlow, controladores e/ou *hosts*, tendo resposta em tempo real. O Mininet é um dos emuladores mais utilizados atualmente para a avaliação de redes SDN já que permite a emulação de uma rede OpenFlow com a utilização de apenas uma máquina [Ortiz et al. 2016]. O Mininet utiliza primitivas de virtualização do sistema operacional onde o usuário pode criar, interagir, personalizar e compartilhar um protótipo de rede OpenFlow com *switches*, *hosts* e controladores [Conterato et al. 2013]. Além disso, esse emulador utiliza controladores reais e pode ser integrado com equipamentos externos à emulação.

No entanto, como qualquer outro *software* de emulação, o Mininet tem limitações devido às restrições da plataforma de *hardware/software* onde é executado, o que impede que ele escale para grandes redes [Ortiz et al. 2016]. Essa limitação é devido ao uso de um processo *shell* para emulação e da necessidade de execução de um processo para cada *switch* OpenFlow virtual e/ou *host* em espaço de usuário. O Mininet, até o momento ainda não fornece desempenho e qualidade fiéis a uma rede real [Keti and Askar 2015]. De fato, essa fidelidade depende da disponibilidade de recursos de processamento e memória para emular todos os *switches*, *hosts* e enlaces da rede em tempo real [Conterato et al. 2013].

3.2. Ambiente OpenFlow Real

Atualmente, diversos fabricantes já disponibilizam *switches* com suporte ao protocolo OpenFlow. Fabricantes de equipamentos, operadores de rede e de empresas como *Facebook, Microsoft e Google* já em 2011, formaram a *Open Networking Foundation* (ONF)[2], com o objetivo de promover o OpenFlow e as redes SDN. O desempenho de um *switch* OpenFlow real pode ser influenciado pela característica do hardware utilizado, já que pode-se implementar o *switch* OpenFlow como *Software Switch* (SS), ou *Hardware Switch* (HS). O SS implementa as tabelas de fluxo, campos, entradas, etc, como uma estrutura de dados através de software, geralmente utilizando sistemas UNIX/Linux e memória comum (SRAM - *Static Random Access Memory* - ou DRAM - *Dynamic RAM*). Nesse modelo, os dados são processados através da CPU, e por consequência, o desempenho deste é limitado pelo poder de processamento da(s) CPU(s) utilizada(s). Nos HS geralmente são utilizados uma memória especializada (*Ternary Content Addressable Memory*

[1]https://www.nsnam.org/

[2]https://www.opennetworking.org/

- TCAM) e sistemas operacionais proprietários para implementar o protocolo OpenFlow e as tabelas de fluxo. Assim, os pacotes são processados através de componentes especializados de *hardware*, orientados para implementação de tarefas específicas em domínios bem definidos, denominados *Application Specific Integrated Circuits* (ASIC). Com isso, a queda de desempenho é mínima.

Uma comparação imediata é que os HSs são considerados mais rápidos do que SSs pois utilizam memórias especializadas para realizar a checagem de regras em paralelo, apesar de possuírem menor capacidade de armazenamento. Com isso, os HSs suportam um número limitado de regras enquanto SS suportam uma quantidade de regras muito maior. Com isso, geralmente, os SSs utilizam altas taxas de processamento para alcançar um menor desempenho que os HSs. No entanto, SSs possuem uma maior flexibilidade para implementar ações mais complexas [Goransson and Black 2014]. Dentro deste cenário, o *Open vSwitch* (OvS) se encaixa como um SS. O OvS é um *switch* virtual multicamada projetado para permitir a automatização maciça da rede através da extensão programática, enquanto ainda suporta interfaces e protocolos de gerenciamento padrão. Ressalta-se que alguns *switches* de mercado, como o WRT54GL da *LinkSys*, utilizam o OvS para implementação. Devido a essas características bem diferentes, trabalhos como o de Katta et al. [Katta et al. 2014] propoem modelos híbridos de *switch* no intuito de combinar as vantagens dos HS e SS em um modelo híbrido que consiga lidar com o *cache* de regras de forma mais eficiente.

Ressalta-se que podem ser montados *testbeds*, privados ou públicos, para conduzir os experimentos tanto com *hardware switches* como com *software switches*. Os *switches* de mercado podem usar ambos os modelos, SS e HS, e ainda podem ser utilizadas máquinas com o OvS emulando os switches. Exemplos de *testbeds* públicos são o FIBRE (*Future Internet Brazilian Environment for Experimentation*)[3], o OFELIA (*OpenFlow in Europe: Linking Infrastructure and Applications*)[4] e o PlanetLab[5]. Estes *testbeds* incluem tanto *hardware switches* quanto *software switches* em suas arquiteturas.

O *testbed* FIBRE utilizado nas simulações, é uma infraestrutura de pesquisa focada em experimentação em redes e é aberta para a utilização por estudantes e pesquisadores brasileiros. Dentre suas funcionalidades plataforma permite a execução de experimentos distribuídos geograficamente em larga escala por meio de alocação de nós sem fio e instanciação de máquinas virtuais nas ilhas de experimentação, seleção de topologias virtuais e alocação de espaço de fluxos OpenFlow tanto nas ilhas de experimentação quanto no backbone e uso de conectividade de 1 Gb/s entre as ilhas de experimentação.

3.3. Escolha do ambiente

A experimentação com redes OpenFlow apresenta muitas possibilidades que diferem muito em custo, desempenho, flexibilidade, facilidade de implementação, etc. Cada uma das características abordadas impacta de forma diferente nos testes realizados. O uso de uma solução mais veloz no processamento pode fazer diferença em testes que precisem medir o tempo de processamento entre a descoberta de uma falha em uma porta e comutação para a backup, por exemplo. Nesse caso, o uso de um emulador, como o

[3]http://fibre.org.br/

[4]http://www.fp7-ofelia.eu/

[5]https://www.planet-lab.org/

Mininet, pode alterar de forma significativa os resultados.

A escolha do ambiente de testes precisa ser feita com base nos impactos que o ambiente terá sobre o resultado do experimento. Muitos trabalhos não passam por essa escolha e podem ter resultados invalidados por conta do uso de um ambiente não adequado. Por esse motivo, a avaliação dos ambientes OpenFlow para experimentação é essencial, pois resulta em uma maior assertividade dos testes realizados quando escolhido de forma adequada para cada solução. Por tal motivo as simulações foram realizadas em ambientes de simulações distintos como emuladores, *testbeds* privados e públicos. Uma vez que a mesma simulação é realizada nos diferentes ambientes torna-se possível avaliar o impacto de tal escolha no resultado da experimentação.

4. Ambiente de Avaliação

O intuito dos testes realizados por este trabalho é avaliar o comportamento de diversas soluções existentes para construção de ambientes OpenFlow para testes de novas soluções. Os critérios de comparações utilizados se baseiam em simulações que têm enfoque no desempenho do dispositivo OpenFlow. Para isso, são utilizados testes que medem a capacidade do dispositivo tratar requisições simultâneas e distintas, elevado tráfego de dados, protocolos de transporte e requisições solicitadas pelo controlador. Para simulações que necessitam de limitação da largura de banda são utilizados arquivos *bash* configurados e armazenados dentro dos *hosts* clientes com as premissas necessárias para limitação da banda de tráfego.

Essas escolhas são insumos que apontam o caminho para distinguir a eficiência de cada plataforma de experimentação estudada, uma vez que tais análises focam em como será o tratamento do dispositivo OpenFlow para cada solicitação a ele realizada.

4.1. Topologia dos Testes

Para realização dos testes foi projetada uma topologia para atender testes em emuladores, *testbeds* e ambientes físicos igualmente, sendo, esse último, definido por limitações financeiras para obtenção do *hardware*, o que limitou a quantidade de *hosts* e *switches* em todos os ambientes. A topologia consiste de 2 *hosts*, 1 controlador Ryu e uma solução com OvS conforme ilustra a Figura 1. A topologia concebida atende aos testes planejados para análise das soluções e ao mesmo tempo é de fácil implementação. Usando duas máquinas como cliente é possível realizar diferentes testes de comunicação entre os pares de clientes, focando tanto no plano de dados como em testes que estimulem sobrecarregar o dispositivo OpenFlow em utilizar o plano de controle para se comunicar com o controlador.

Os testes duraram 20 segundos, tempo suficiente para realizar as principais simulações de criação de pacotes, geração de tráfego e monitoração de dispositivos, e os resultados são apresentados com um intervalo de confiança de 95%. Foram realizados os testes em cinco ambientes usualmente utilizados para testes com *Openflow*, sendo eles: emulação com o Mininet; uso de *testbeds* públicos, especificamente com FIBRE; montagem de *testbed* privado usando um *switch* comercial, especificamente o Pronto (Pica8 - P3295)[6]; e montagem de *testbed* privado usando um OvS instalado diretamente num

[6]http://www.pica8.com/documents/pica8-datasheet-48x1gbe-p3290-p3295.pdf

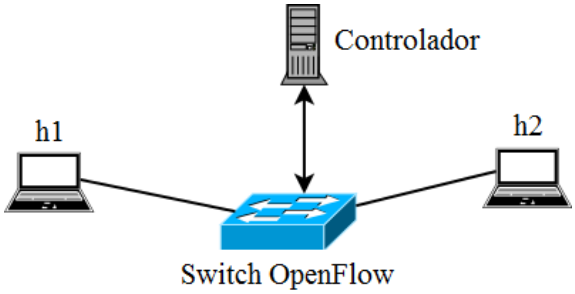


Figura 1. Topologia básica de avaliação.

desktop com 4 GB RAM e outro em máquina com 8 GB RAM, de forma a mensurar a influência na capacidade de desempenho com a variação de memória utilizada. O FIBRE e o *switch* Pronto disponível só trabalham com o protocolo OpenFlow na versão 1.0, com isso, a fim de manter a premissa nos outros ambientes analisados, todos os testes foram realizados com o OpenFlow na versão 1.0.

4.2. Configuração dos Testes

A estrutura física que compõem o *hardware* utilizado para as simulações tem papel fundamental no que tange a influência nos resultados obtidos, máquinas compostas de melhores processadores e memória tendem a desempenhar resultados superiores nas simulações. Por tal motivo, buscou-se usar a mesma configuração de *hardware* no ambiente de instalação da máquina virtual Mininet, no *switch* OvS utilizado no *testbed* privado e no *hardware* utilizado para o controlador Ryu. Qualquer mudança neste *hardware* dificulta a avaliação justa dos experimentos.

Os testes baseados em emulação foram feitos utilizando o Mininet 2.2.1[7], instalado na plataforma *VirtualBox* usando 2G RAM e 4vcpu, localizada em desktop com 4GB RAM, 55GB HD e sistema operacional XUBUNTU 14.04.1, 32bits i686, Intel(R) Core(TM) i7-2600 3.40GHZ (1 placa com 4 núcleos e 8 threads). Já a construção dos *testbeds* privados foi feita usando, no primeiro caso, um *switch* comercial e, no segundo caso, o OvS 2.0.2 instalado em desktop com as mesmas configurações da máquina citada anteriormente. Alterou-se a memória dessa máquina entre 4 e 8 GB de RAM para avaliar o impacto causado pela memória. Para o ambiente de emulação com o Mininet, foi utilizada a mesma versão OvS 2.0.2 de *switch* OpenFlow.

Para os testes com *testbeds* privados, os *hosts* foram criados com as seguintes configurações: placa Mini-TX dual-core Intel® Atom Processor N2800 (2 núcleos e 4 threads), 2 GB RAM, XUBUNTU 14.04.01, 32 bits, i686. O uso dessas máquinas desonerou os custos financeiros para realização das simulações em *testbeds* privados. No caso do Mininet, os *hosts* foram simulados no próprio ambiente.

Os testes utilizando o FIBRE contaram com equipamentos OpenFlow baseados em placas NetFPGA e *switches* Pronto. Os *hosts* criados possuem sistema operacional GNU/Linux Debian 6.0 e são executados em máquinas virtuais criadas pela própria plataforma, com configuração Linux 3.2.0-4-amd64 SMP Debian 3.2.65-1+deb7u2 x86 64, com 120 MB de memória e 8 GB de HD.

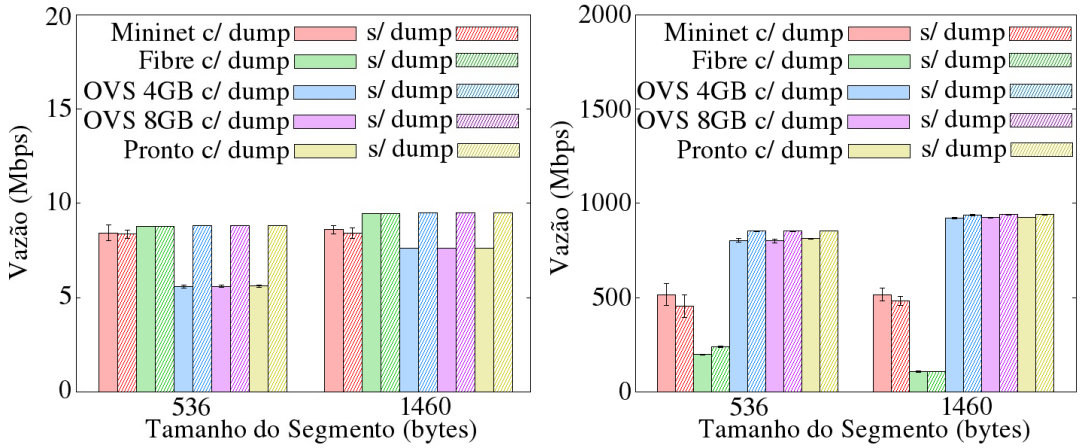
[7]Máquina virtual obtida no site mininet.org

Em todos os testes, foi utilizado o controlador Ryu 4.2.2 em desktop com 4GB RAM, 55GB HD, SO: XUBUNTU 14.04.1, 32bits i686, Intel(R) Core(TM) i7-2600 3.40GHZ (1 placa com 4 núcleos e 8 threads). No caso do ambiente Mininet foi utilizada a mesma versão de controlador, instalada na máquina virtual correspondente ao Mininet. Devido ao uso da linguagem de programação Python e suporte às versões 1.0, 1.2, 1.3 e 1.4 do protocolo OpenFlow o que amplia sua utilização para simulações futuras; optou-se pelo controlador Ryu. A escolha de controladores distintos influencia no tempo de processamento do plano de controle, porém o estudo realizado não foca na comparação de controladores OpenFlow.

Para realizar os testes, foi utilizado o gerador de tráfego *Iperf* (V2.0.5)[8], o gerador de pacotes *Scapy* (V2.2.0)[9], o *wireshark*[10] e o tcpdump[11], instalados nos *hosts* pertinentes e no servidor do controlador conforme necessário.

5. Resultados Obtidos

O primeiro teste realizado avalia o impacto de processamento dos dispositivos sobre os resultados do teste de rede. Para tanto, gerou-se um tráfego TCP com o Iperf entre os *hosts* h1 e h2. Para avaliar o impacto do processamento tanto nos *hosts* como nos *switches*, variou-se o tamanho do segmento entre 536 e 1460 B, pois segmentos menores levam a geração de mais pacotes. A banda do enlace também foi variada entre 10 Mbps e 1 Gbps, para variar o número de pacotes gerados/encaminhados. Outra análise realizada foi o impacto do uso do tcpdump como ferramenta de monitoração. Novamente, esse é mais um fator que gera sobrecarga na máquina e que tem influência no resultado. O tcpdump foi colocado em ambos os *hosts*. A aplicação usada no controlador simula um *switch* padrão, definindo entradas de fluxos com base apenas no MAC de destino.



(a) 10 Mbps (b) 1 Gbps

Figura 2. Observação da influência dos valores de tamanho máximo de segmento e uso de monitoração via tcpdump.

No caso das simulações realizadas em ambiente de laboratório real, representado pelo *switch* Pronto e OvS instalado em desktop físico, a utilização do tcpdump influencia

[8]https://iperf.fr/
[9]http://www.secdev.org/projects/scapy/
[10]https://www.wireshark.org/
[11]http://www.tcpdump.org/

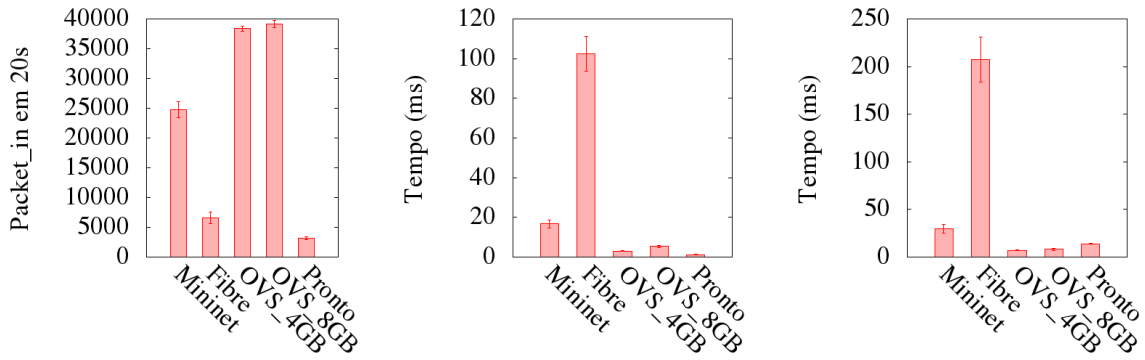
consideravelmente na vazão obtida, apresentando maior impacto proporcional em enlaces de menor capacidade. Essa característica detectada está relacionada ao *hardware* utilizado para os *hosts*. Pois, o equipamento utilizado como *host* era de capacidade menor, compatível com o custo financeiro mais baixo, porém proporciona um processamento aquém do desejável. Em contrapartida, nos ambientes Mininet e FIBRE, o comportamento foi inconsistente com o esperado, sendo importante salientar que são ambientes onde não é possível ter controle sobre sua composição, uma vez que a plataforma FIBRE é compartilhada com outros pesquisadores em quantidade não verificável, que podem usar o ambiente no mesmo momento, impossibilitando controlar a banda disponível e o ambiente Mininet pode sofrer influências de outras tarefas do sistema operacional.

Na Figura 2(a), o uso de tcpdump em nada influenciou o ambiente FIBRE e, na Figura 2(b) pacotes com tamanho de segmento máximo menor, inesperadamente, apresentaram maior vazão. O ambiente FIBRE que se assemelha com um ambiente de uso real, no qual existe pouca controlabilidade dos clientes que o utilizam, o fato de executar os testes de maneira sequencial, ou seja, iniciando a simulação com o tamanho máximo de segmento menor, e variando largura de banda, e uso de tcpdump e somente posteriormente a mesma sequência de testes porém com uso de tamanho máximo de segmento maior, pode ter ocasionado o resultado encontrado. Assim, ao utilizar este tipo de ambiente de experimentação é importante executar os testes de forma mesclada, para aumentar a probabilidade que os diferentes cenários sejam expostos às mesmas condições externas. No Mininet, o valor de tamanho de segmento máximo e o uso de tcpdump, considerando a margem de erro, em nada influenciaram na vazão dos testes de Iperf. Contudo, ao se estressar o ambiente sem limitar a sua largura de banda, observou-se a mesma limitação, sendo que obteve-se vazão de 9 Gbps sem uso de tcpdump e de 4.5 Gbps com o uso de tcpdump. Isso é um indicativo importante, pois, no Mininet, a capacidade de processamento é dividida pelos elementos que se colocam na emulação. Se fossem usados mais *hosts* ativos ou mais *switches*; *links* com 1 Gbps, ou menos, provavelmente seriam afetados, causando uma perda considerável na capacidade do enlace.

A próxima análise realizada foi relativa aos impactos sobre o plano de controle dependendo da plataforma de experimentação que está em uso. Especificamente, buscou-se aumentar o número de eventos de *packet_in*, que são gerados todas as vezes que o *switch* recebe um pacote para o qual não existe entrada de fluxo. A quantidade de eventos gerados é impactada pela capacidade do *host* de gerar pacotes com endereços MAC diferentes, pela capacidade de processamento do *switch* e pela capacidade de resposta do controlador. Os pacotes são gerados no *host* 1 utilizando a biblioteca *SCAPY* em todos os ambientes, gerando pacotes ARP para o *host* 2, variando o MAC de origem. Para medir a quantidade de *packet_in* gerados foi habilitado o uso de *wireshark* em h1, h2 e controlador. Uma máquina para o *host* 1 com configurações mais robustas foi utilizada nos testes de OvS em desktop e switch Pronto: Intel ® Core i5-M480 2.67 GHz (1 placa com 4 Cores e 8 threads), 4 GB RAM, 500GB of HD, XUBUNTU 14.04.01, 32 bit, i686.

O *host* 1, montado para os testes de OvS em desktop e *switch* Pronto foi o que gerou maior quantidade de pacotes ARP, em média 39000 pacotes ARPs, indicando que um desktop com configurações mais robustas favorece simulações que dependem do desempenho de máquinas que geram pacotes.

Nos testes de OvS em desktop, os pacotes ARP ocasionaram a mesma quantidade



(a) Eventos de *packet_in* processados pelo *switch*.

(b) Tempo de processamento do plano de controle.

(c) Tempo de requisição ARP entre h1 e h2.

Figura 3. Análise do processamento de eventos de *packet_in* nos diversos ambientes de teste.

de *packet_in* para o controlador Ryu. Para esse teste, o ambiente OvS em desktop obteve o melhor resultado. Porém no caso dos testes em *switch* Pronto, os pacotes ARP gerados em h1, que deveriam criar a mesma quantidade de *packet_in* pelo *switch*, não ocorreu, pois houve perda de pacotes ARP, que não foram convertidos em eventos de *packet_in*. A máquina virtual do Mininet criou em média 25000 *packet_in* provenientes de 25000 chamadas ARP, não havendo perda de pacotes ao longo do processo. O host h1, montado via plataforma FIBRE, criou em média 34000 pacotes ARP, porém devido a perda de pacotes ARP a quantidade de *packet_in* proveniente do *switch* Pronto para o controlador Ryu foi inferior, indicando que este ambiente impactou negativamente o processamento no plano de controle. Tais testes foram mensurados através da análise do arquivo wireshark gerado no *host* h1, verificando a quantidade de pacotes ARP gerados (com MAC de origem distintos) e no arquivo wireshark gerado no controlador, verificando a quantidade de *packet_in* que chegaram no mesmo.

A Figura 3(b) mostra a análise do tempo de processamento entre chegada dos eventos de *packet_in* no controlador e a geração do evento *flow_mod* pelo controlador, que seta uma nova entrada de fluxo no *switch*. Nesse teste, observa-se um desempenho inferior do ambiente FIBRE com o plano de controle. Embora o Mininet tenha processado aproximadamente 4 vezes o número de *packet_in* que o ambiente FIBRE, ele apresenta um tempo de processamento no plano de controle 5 vezes menor que o FIBRE. Os ambientes de *testbed* privados apresentaram um tempo menor que o Mininet e o FIBRE, mesmo no caso do OvS em desktop no qual transitou um número elevado de pacotes no plano de controle. Devido a perda de pacotes ARP, no caso do teste com *switch* Pronto, a quantidade de pacotes no plano de controle foi reduzida, o que proporcionou um processamento rápido.

Mediu-se, como mostrado na Figura 3(c), o tempo entre a requisição ARP feita por h1, com MACs distintos, até a mensagem chegar em h2 e seu retorno até h1. Nesse caso, o ambiente FIBRE foi o que apresentou maiores tempos. Uma vez que o ambiente FIBRE pode estar sendo compartilhado por outros pesquisadores, tal situação pode ter impactado o tempo de processamento do switch para criar um número de entradas de fluxo elevada.

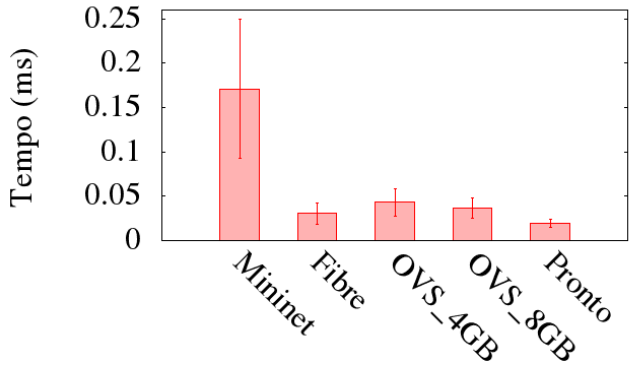


Figura 4. Observação de *JITTER*.

Por fim, a Figura 4 analisa as conexões do tipo UDP, porém o foco está sobre a variação do atraso entre os pacotes sucessivos de dados (*jitter*). Os testes utilizaram tamanho máximo de segmento de 1460 bytes, enlaces de 1 Gbps e taxa de envio de 100 Mbps. Nesta análise, mais uma vez, fica claro como a plataforma Mininet possui desempenho inferior comparado com os outros cenários.

6. Custo Financeiro das Soluções e Análise Qualitativa

Além de observar fatores relacionados ao desempenho, é importante também relacionar os custos financeiros envolvidos para implementação dos ambientes de teste, uma vez que sempre existem limitações financeiras nos projetos. Foi realizado um levantamento, aproximado, nos preços dos elementos utilizados para montagem dos ambientes de experimentação analisados, entre eles o *switch* Pronto Pica8 P-3295[12], os computadores baseados em placas Mini-ITX DN2800MT [13] e as placas HP Ethernet 1Gb 4-port 331T Adapter[14], que foram usadas na emulação de *switches* em computadores com processador Intel Core i7 2600 [15].

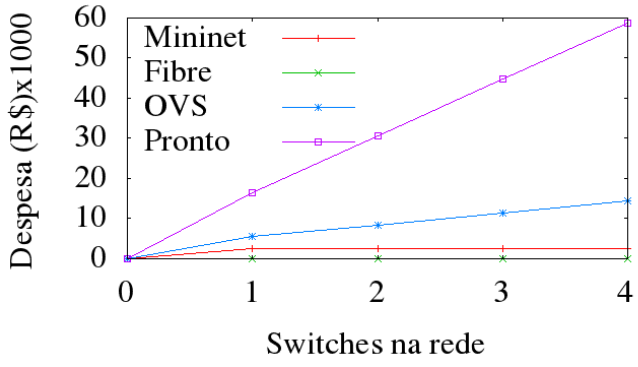


Figura 5. Demonstrativo de gastos por quantidade de *switches* com um *host* conectado por *switch* no ambiente de experimentação.

[12]http://store.netgate.com/Pica8/P-3295.aspx

[13]http://www.atera.com.br/produto/DN2800MT/Placa+m%C3%A3e+Intel+DN2800MT+c-+Atom+Dual+Core+N2800+e+NM10

[14]http://eutec.com.br/placa-de-rede-hp-ethernet-1gb-4-port-331t-adapter-647594-b21

[15]http://ark.intel.com/PT-BR/products/52213/Intel-Core-i7-2600-Processor-8M-Cache-up-to-3_80-GHz

Os valores apresentados na Figura 5 foram obtidos a partir das seguintes estimativas. Com o uso do FIBRE estima-se custo zero de montagem, somente sendo necessário algum dispositivo com acesso à internet. Com o Mininet o custo é determinado pela aquisição de um computador com configurações de memória entre 4GB ou 8GB de RAM e Intel Core i7 2600 (1 placa com 4 núcleos e 8 threads), monitor, teclado e mouse. Nos testes com OVS em desktop o custo é determinado pela aquisição de um desktop para o controlador e um por *switch*; 1 placa Mini-ITX por host implementado; e 1 placa HP Ethernet 1Gb 4-port 331T Adapter por *switch* implementado em computador. Nos testes com *switch* Pronto estima-se o custo referente ao uso de um desktop para funcionamento do controlador Ryu, 1 placa Mini-ITX por *host* e dos *switches* Pronto.

Na Figura 5 apresenta-se a evolução de custo para cada solução em função da quantidade de *switches* a serem utilizados, considerando o uso de um *host* por *switch* e um controlador por ambiente. Para o ambiente FIBRE e Mininet, o aumento de *switches* nas simulações não afeta os custos, uma vez que se trata de uma plataforma digital de uso compartilhado e um emulador de redes, respectivamente. Naturalmente, o aumento do número de *switches* é limitado em ambos casos, seja pelo número máximo de *switches* disponíveis, no caso do FIBRE, ou pela capacidade de processamento e memória da máquina, no caso do Mininet.

Tabela 1. Comparação de parâmetros analisados nos seus respectivos ambientes de simulação.

Parâmetros	Mininet	FIBRE	Mini-itx+ Desktop 4GB	Mini-Itx+ Desktop 8GB	Switch Pronto
Desempenho do *switch*	Médio	Alto	Alto	Alto	Alto
Desempenho do plano de controle	Baixo	Alto	Alto	Alto	Alto
Desempenho do cliente (host)	Baixo	Alto	Baixo	Baixo	Baixo
Controle de vazão	Baixo	Baixo	Alto	Alto	Alto
Celeridade para implantação	Alto	Baixo	Médio	Médio	Médio
Complexidade para as medições	Baixo	Alto	Médio	Médio	Médio
Custos de implantação	Baixo	N/A	Médio	Médio	Alto
Usa outras versões do OpenFlow	Sim	Não	Sim	Sim	Não
Permite controle in band	Não	Não	Sim	Sim	Não
Permite controle out band	Sim	Sim	Sim	Sim	Sim

A Tabela 1 visa comparar os diversos parâmetros analisados nos ambientes de experimentação averiguados. Outro foco é apresentar o desafio tanto no que tange a implantação quanto a obtenção dos elementos que possibilitam a análise dos ambientes. Os parâmetros foram classificados entre alto, médio e baixo de forma a mensurar a aptidão de cada parâmetro da tabela ao seu respectivo ambiente de experimentação. Em conjunto, foram apresentadas algumas linhas informando a disponibilidade ou não de itens possíveis de existir em ambientes de experimentação.

7. Conclusões e Trabalhos Futuros

Neste artigo, o comportamento de diferentes plataformas de experimentação baseadas em OpenFlow foi avaliado. A escolha do ambiente é uma das premissas para se

alcançar um resultado mais fiel. Uma vez que o desempenho do ambiente afeta diretamente o resultado da validação de novas propopostas, testes que avaliem o tempo de processamento do ambiente, vazão, atraso, etc, são essenciais para uma avaliação mais assertiva.

A utilização do ambiente Mininet pode ser uma excelente escolha para o início de aprendizagem sobre o funcionamento do OpenFlow e do conceito de SDN, tendo a vantagem de ser uma máquina virtual obtida diretamente em sites especializados e facilmente colocada em qualquer computador com um *software* de virtualização. Porém, esta pode ser uma solução muito limitada se o usuário visa realizar testes de desempenho. Como observado, o Mininet apresentou, na maioria dos casos, um desempenho muito inferior quando comparado aos demais ambientes de teste.

Soluções que fazem uso de desktops físicos para implementação de *software* de *switch* OpenFlow apresentam bom desempenho e tornando-se uma boa opção para testes e estudos de novas soluções em SDN. O custo de utilização desse mecanismo de implementação apresenta valores financeiros abaixo do que os conseguidos com a utilização de um *switch* físico habilitado para OpenFlow e, como verificado nas simulações, a variação de memória no desktop utilizado não afeta de forma consistente os resultados obtidos. Contudo, dependendo do porte do teste, esta pode ser uma solução inviável em termos de investimento inicial para montagem da rede de teste.

O ambiente FIBRE apresentou aspectos positivos e negativos. O ambiente utiliza tanto *switches* OpenFlow comerciais quanto *switches* emulados em computadores, mas tem alta variabilidade no desempenho observado, pois o ambiente é compartilhado entre pesquisadores. Não existe controle de banda mínima ou de disponibilidade de recursos nos servidores de máquinas virtuais. Por outro lado, o FIBRE tem nós espalhados por diversos locais do Brasil e do mundo, tornando o cenário mais real para experimentação. Além disso, apresentou bons resultados com relação à capacidade dos planos de dado e de controle e custo de implementação zero, tornando-se uma boa ferramenta, dependendo dos requisitos do teste.

Portanto, existem várias possibilidades de ambientes de experimentação para aplicações baseadas em OpenFlow, mas nem todos os ambientes são propícios para qualquer tipo de teste. Além disso, ficam claras as restrições de cada um dos ambientes e quais são os impactos dele sobre os resultados observados. Assim, as conclusões obtidas em cada tipo de ambiente devem sempre ser balizadas pelas restrições específicas do cenário.

Como trabalhos futuros, pretende-se realizar o teste com hosts mais eficientes, permitindo uma comparação mais justa na geração de novos fluxos. Utilizar geradores de pacotes com maior capacidade e menor custo computacional, realizar testes do Mininet usando um controlador externo, desenvolver mais de um *switch* por *desktop* para avaliação do comportamento e avaliar outros testebeds públicos.

8. Agradecimentos

Agradecemos a todas as pessoas que, direta ou indiretamente contribuíram com carinho e atenção durante a construção desse trabalho, especialmente ao CNPq, CAPES e FAPERJ pelo apoio financeiro e a todos os envolvidos da Rede Nacional de Ensino e Pesquisa (RNP) pelo apoio com os testes realizados na plataforma FIBRE.

Referências

Bianco, A., Birke, R., Giraudo, L., and Palacin, M. (2010). OpenFlow switching: Data plane performance. In *ICC, 2010 IEEE*, pages 1–5. IEEE.

Conterato, M., de Oliveira, I., Ferreto, T., and De Rose, C. A. (2013). Avaliação do suporte à simulação de redes OpenFlow no NS-3. *Anais da 11a. Escola Regional de Redes de Computadores*.

Costa, L. C., Vieira, A. B., de Britto, E., Silva, D. F., Gomes, G., Correia, L. H., and Vieira, L. F. (2016). Avaliação de Desempenho de Planos de Dados OpenFlow.

Das, S., Parulkar, G., McKeown, N., Singh, P., Getachew, D., and Ong, L. (2010). Packet and circuit network convergence with OpenFlow. In *Optical Fiber Communication Conference*, page OTuG1. Optical Society of America.

de Oliveira, R. L. S., Schweitzer, C. M., Shinoda, A. A., and Prete, L. R. (2014). Using Mininet for emulation and prototyping software-defined networks. In *COLCOM, 2014 IEEE on*, pages 1–6. IEEE.

Feamster, N., Rexford, J., and Zegura, E. (2014). The road to SDN: an intellectual history of programmable networks. *ACM SIGCOMM Computer Communication Review*, 44(2):87–98.

Goransson, P. and Black, C. (2014). *Software Defined Networks: A Comprehensive Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition.

Heller, B., Seetharaman, S., Mahadevan, P., Yiakoumis, Y., Sharma, P., Banerjee, S., and McKeown, N. (2010). ElasticTree: Saving Energy in Data Center Networks. In *NSDI*, volume 10, pages 249–264.

Huang, T., Yu, F. R., Zhang, C., Liu, J., Zhang, J., and Liu, J. (2016). A Survey on Large-Scale Software Defined Networking (SDN) Testbeds: Approaches and Challenges. *IEEE Surveys & Tutorials*.

Katta, N., Alipourfard, O., Rexford, J., and Walker, D. (2014). Infinite CacheFlow in software-defined networks. In *Proceedings of the third workshop on Hot topics in software defined networking*, pages 175–180. ACM.

Keti, F. and Askar, S. (2015). Emulation of Software Defined Networks Using Mininet in Different Simulation Environments. In *2015 6th International Conference on Intelligent Systems, Modelling and Simulation*, pages 205–210. IEEE.

Macedo, D. F., Guedes, D., Vieira, L. F., Vieira, M. A., and Nogueira, M. (2015). Programmable networks-from software-defined radio to software-defined networking. *IEEE Surveys & Tutorials*, 17(2):1102–1125.

Masoudi, R. and Ghaffari, A. Software defined networks: A survey.

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*, pages 69–74.

Ortiz, J., J.Londoño, and Novillo, F. (2016). Evaluation of performance and scalability of Mininet in scenarios with large data centers.

Zal, M. and Kleban, J. (2014). Performance Evaluation of OpenFlow Devices.

Fator de Resiliência para Aprimoramento Topológico em Redes Definidas por Software

Pedro Montibeler[1], Fernando Farias[1], Antônio Abelém[1]

[1]Grupo de Estudos em Redes de Computadores e Comunicação Multimídia
Instituto de Informática – Universidade Federal do Pará (UFPA)
pedro.salvador@itec.ufpa.br, {fernnf, abelem}@ufpa.br

Abstract. *Software Defined Networks is a paradigm that flexibilizes the management of networking, which separates the control and forwarding planes. This separation introduces new concerns towards the resilience of the network, which now presents different vulnerabilities related to the interaction between these planes. A resilience factor for Software Defined Networks is proposed, using multiple metrics to analyze intrinsic features of its architecture, serving as an indication for its resilience. Beyond that, topological augmentation algorithms are employed to increase the resilience of test topologies, as indicated by the proposed factor. Tests results demonstrate an improvement of the topologies' resilience characteristics.*

Resumo. *Redes Definidas por Software é um paradigma que flexibiliza a gerência de redes de computadores ao separar os planos de controle e de dados. Essa separação introduz novas preocupações quanto a resiliência da rede, que passa a apresentar diferentes vulnerabilidades relacionadas a interação entre os planos. É proposto um fator de resiliência para Redes Definidas por Software, utilizando múltiplas métricas para analisar características intrínsecas da arquitetura, servindo como indicativo de resiliência da rede. Além disso, algoritmos de aprimoramento topológico são empregados para aperfeiçoar a resiliência das topologias utilizadas. Os resultados demonstram melhoria nas características de resiliência.*

1. Introdução

Redes Definidas por *Software* (SDN) é um modelo de redes de computadores onde o plano de controle, tradicionalmente embutido nos comutadores, é separado do plano de encaminhamento, que agora abstrai suas funções de forma que possam ser programadas por *software* a partir do plano de controle [ONF 2012]. A programabilidade da rede a partir da visão centralizada do plano de controle flexibiliza e simplifica a gerência dos dispositivos de rede por parte dos administradores e desenvolvedores, facilitando a pesquisa e desenvolvimento de tecnologias na área [Farias et al. 2011].

A dependência do plano de controle centralizado, no entanto, traz novas

preocupações quanto à resiliência da rede. Falhas de dispositivos, que os tornem inoperantes ou incomunicáveis, estão sujeitas a acontecer não apenas no plano de encaminhamento, como também no plano de controle, sendo a conectividade entre esses planos outra vulnerabilidade. Falhas no plano de encaminhamento, tais como quedas de enlaces, devem ser detectadas e recuperadas pelo plano de controle, pois caso a falha prejudique a interação entre os controladores e os comutadores, os efeitos negativos na operação da rede podem ser agravados.

Por isso, uma rede que segue o modelo SDN deve ser capaz de tolerar falhas em entidades de seus diferentes planos [Ros e Ruiz 2014]. Logo, há a necessidade de se avaliar aspectos de resiliência durante o planejamento de uma SDN, considerando as características intrínsecas de sua arquitetura, buscando preservar o funcionamento de seus planos e sua correta interação.

Nesse contexto, trabalhos recentes tem estudado a otimização do posicionamento dos controladores para melhorar diferentes características da rede, como a resiliência. Comumente é utilizado apenas uma métrica para caracterizar a resiliência da rede [Ros e Ruiz 2014] [Müller et al. 2014], e mesmo quando são utilizadas múltiplas métricas, a otimização se limita ao plano de controle, embora o plano de dados influencie diretamente nas características da rede [Heller, Sherwood e McKeown 2012] [Hock et al. 2013].

O objetivo deste trabalho é contribuir com um fator de resiliência para o auxílio da tomada de decisões no planejamento de SDNs, onde são selecionadas e correlacionadas múltiplas métricas pertinentes às propriedades de uma SDN. Outra contribuição é a aplicação de algoritmos de aprimoramento topológico em SDNs buscando melhorar suas características de resiliência, em específico, algoritmos de Redistribuição de Arestas (*Edge Rewiring*). Na análise de resultados, observa-se que os algoritmos implementados melhoraram as características de resiliência das topologias testadas.

Além desta seção introdutória, este artigo está dividido da seguinte forma. Na Seção 2 são discutidos os trabalhos relacionados ao tema. A Seção 3 detalha o fator de resiliência proposto. A Seção 4 demonstra sua aplicação como um parâmetro de algoritmos de aprimoramento topológico. Por fim, a Seção 5 apresenta as conclusões gerais e trabalhos futuros.

2. Trabalhos Relacionados

Na análise de SDNs, trabalhos recentes têm estudado o Problema do Posicionamento do Controlador (*Controller Placement Problem*) [Heller, Sherwood e McKeown 2012], onde é evidenciado que o posicionamento dos controladores na rede afetam características de conectividade entre os planos e latência de comunicação. Nesse contexto, os trabalhos seguintes consideram características de resiliência em suas

análises.

Na proposta de Ros e Ruiz [2014] é investigado, especificamente, o impacto do posicionamento dos controladores na capacidade de sobrevivência da SDN. Em seu trabalho, buscam determinar a posição e quantidade de controladores em uma topologia de forma a garantir um valor mínimo na métrica de confiabilidade k-terminal (*k-terminal reliability*). No trabalho de Müller [2014] é proposto uma estratégia de posicionamento que esquematize o balanceamento de carga dos controladores, e otimize a resiliência da rede analisando os caminhos disjuntos entre os nós. Esses trabalhos caracterizam a resiliência somente com uma métrica relacionada a capacidade de sobrevivência da rede, que apesar de relevante, é insuficiente em SDN.

Nas propostas de Hock [2013] e Lange [2015], o problema do posicionamento do controlador é trabalhado utilizando diversas métricas, como latência entre dispositivos, balanceamento de carga entre os controladores, e a conectividade entre os planos. São consideradas múltiplas características pertinentes a resiliência em SDN, como a capacidade de sobrevivência e a latência de comunicação entre os dispositivos dos diferentes planos. No entanto, propõem otimizar unicamente o plano de controle, embora a estrutura topológica do plano de dados influencie diretamente na solução [Heller, Sherwood e McKeown 2012] [Hock et al. 2013] [Ros e Ruiz 2014].

Como exemplificado, os trabalhos em SDN comumente utilizam apenas uma métrica para caracterizar a rede quanto à sua resiliência, e mesmo quando investigam múltiplas métricas, as otimizações realizadas buscam somente alterar a posição e quantidade dos controladores na rede, sem interferir na estrutura topológica do plano de dados, o que influencia significativamente na solução, como observado pelos próprios autores.

Neste trabalho, múltiplas métricas pertinentes a resiliência de uma SDN são verificadas e consideradas simultaneamente, resultando em um fator para aferir a resiliência de uma topologia. O resultado pode ser utilizado para comparação objetiva entre topologias, ou aplicado como função objetivo em um algoritmo de posicionamento de controladores, ou ainda, como investigado neste trabalho, usado como parâmetro para algoritmos de aprimoramento topológico. Em nossa proposta utiliza-se algoritmos de redistribuição de arestas, investigados em trabalhos como de Beygelzimer [2005] e Bai Liang [2015], para realizar alterações topológicas em ambos os planos de uma SDN, buscando melhorar suas características de resiliência.

3. Proposta

Um fator de resiliência pode ser usado como uma ferramenta auxiliar durante a fase de planejamento de uma SDN, possibilitando a comparação objetiva de diferentes topologias, ou como função objetivo em um algoritmo de aprimoramento. Quanto mais características desejáveis forem consideradas na análise de um fator de resiliência, mais

informativos serão seus resultados. Portanto, o fator de resiliência para SDNs proposto deve retornar um valor que seja indicativo das propriedades da rede, correlacionando múltiplas métricas que analisem as características mais pertinentes da estrutura topológica da rede no que se refere a resiliência.

Para isso é verificado a seguir quais as características desejadas para uma SDN resiliente considerando as peculiaridades de sua arquitetura. Em seguida são escolhidas as métricas a serem empregadas como indicativos para essas características, por fim correlacionando-as em um fator de resiliência.

Definido um fator de resiliência, é possível empregá-lo como métrica em um algoritmo visando aprimorar as características de uma SDN. Com esse fim, são selecionados algoritmos de aprimoramento topológico que aplicarão transformações em ambos os planos da topologia SDN, buscando aumentar a resiliência da mesma.

3.1. Características de resiliência em SDN

Métricas de redes complexas e conceitos da teoria de grafos são, atualmente, uns dos principais elementos aplicados na análise de resiliência em redes de comunicação legadas. Comumente, são verificadas a conectividade da estrutura topológica da rede e a existência de caminhos alternativos entre os dispositivos, como indicativos para a capacidade de sobrevivência da rede [Habibi e Phung 2012]. Nas redes legadas, os planos de controle e encaminhamento estão juntos, distribuídos entre os comutadores. Nesse contexto, a análise é realizada trivialmente ao representar a topologia de rede como um grafo, onde os comutadores são os vértices e os enlaces as arestas, sem que haja a necessidade de discriminar os dispositivos, por estes exercerem papéis equivalentes na rede. Por outro lado, em SDN, os dispositivos de controle e encaminhamento não são necessariamente os mesmos. Logo, a análise deve ser diferenciada.

Assim como nas redes de comunicações legadas, é desejável que SDNs possuam uma estrutura topológica tolerantes a falhas, que permita a ocorrência de falhas de nós ou enlaces de forma que a rede não se torne desconexa. Isso se aplica intuitivamente em redes de computadores tradicionais, pois o seu propósito é a transmissão de dados e a comunicação entre todos os dispositivos da rede deve ser preservada para que essa transmissão possa acontecer.

Em uma SDN, mesmo com a separação de planos e a presença de controladores na rede, essa característica continua a ter a mesma importância. Sendo os controladores responsáveis pela lógica de encaminhamento da rede, o correto funcionamento do plano de encaminhamento depende das instruções transmitidas pelos controladores aos comutadores através do plano de controle. Não apenas a conectividade entre os comutadores deve ser tolerante a falhas, para que não se interrompa a transmissão de dados, como a conectividade entre os planos também deve ser tolerante a falhas, para que essa interação entre os planos não seja interrompida. Por último, é importante

lembrar que o plano de controle é logicamente centralizado, portanto mesmo quando múltiplos controladores são empregados, estes devem ser capazes de se comunicar para cooperar nas tomadas de decisão, e sua conectividade também deve ser tolerante a falhas. Conclui-se que, tal como em redes tradicionais, é interessante que a comunicação entre todos os dispositivos de uma SDN seja tolerante a falhas, independente de seu papel na arquitetura.

Relacionada a tolerância a falhas, há também a questão de redundância: apenas uma instância de controlador na rede pode ser suficiente para sua operação, mas a falha desse dispositivo deixará a rede sem plano de controle. Portanto, para aumentar a tolerância a falhas no plano de controle, é interessante que uma SDN possua múltiplos controladores em operação na rede.

Sendo o plano de controle responsável pela lógica de encaminhamento, a detecção de uma falha no plano de encaminhamento e a consequente atualização das regras é responsabilidade dos controladores. Com isso, a latência de comunicação entre os controladores e os comutadores, mais do que uma questão de desempenho, é uma questão de recuperação de falhas. A eficiência da comunicação entre os planos é determinante no tempo de recuperação de falhas da rede, afetando o tempo de resposta do plano de controles a cenários de falha. O plano de controle é logicamente centralizado e, portanto, a comunicação entre múltiplos controladores também deve ser eficiente para agilizar a sincronia entre suas diferentes instâncias. Conclui-se que é interessante para uma SDN que a eficiência de comunicação entre os múltiplos controladores e entre os controladores e os comutadores seja a maior possível.

3.2. Fator de Resiliência de Média de Múltiplas Métricas (FRM3)

A teoria das redes complexas é um tema interdisciplinar que abrange diversas áreas do conhecimento, como computação, matemática, sociologia, física e biologia. Uma rede é representada por um grafo, um conjunto de vértices conectados por arestas, modelando sistemas do mundo real, como a infraestrutura física de uma rede de computadores, para a resolução de problemas específicos [Metz et al. 2007]. Com o estudo de redes complexas, pode-se verificar propriedades de sistemas baseando-se nas relações entre seus componentes, aplicando conceitos e algoritmos da teoria matemática dos grafos. Em redes de computadores, utilizam-se métricas de grafos no planejamento de redes tradicionais, como na verificação de aspectos de resiliência em topologias de rede [Habibi e Phung 2012], controle de congestionamento, entre outros. Para a análise em SDN, as métricas selecionadas são métricas de grafos.

Na tolerância a falhas, uma métrica comumente utilizada é a k-conectividade [Habibi e Phung 2012]. A análise de k-conectividade retorna um valor inteiro que corresponde a quantidade mínima de vértices que precisam ser removidos do grafo para que o grafo induzido constituído dos vértices restantes seja desconexo. Com essa métrica se obtém um indicativo da capacidade de sobrevivência da rede a cenários de

falhas arbitrárias de dispositivos.

Quanto a eficiência de comunicação, a métrica de eficiência é um indicativo da eficiência da estrutura topológica da rede [Latora e Marchiori 2001]. Ela retorna um valor entre zero e um que é inversamente proporcional a média dos comprimentos dos menores caminhos entre os vértices do grafo. Essa métrica é um indicativo da proximidade geral dos dispositivos na topologia de rede.

Combinando as características de tolerância a falhas e eficiência, a métrica de vulnerabilidade indica o decaimento da eficiência com a remoção de um vértice arbitrário [Latora e Marchiori 2001]. O valor retornado é a maior alteração relativa do valor da métrica de eficiência com a remoção de um vértice do grafo. A métrica de vulnerabilidade indica o quanto a eficiência da comunicação entre os vértices de um grafo pode ser afetado pela remoção de um de seus nós.

Considerando as características de resiliência desejáveis especificamente para uma SDN e utilizando das métricas indicativas dessas características, é proposto um Fator de Resiliência de Média de Múltiplas Métricas (FRMMM ou FRM3). O fator proposto é constituído por cinco componentes, ou fatores, que analisam diferentes características da topologia. Cada fator retorna um valor entre zero e um, possibilitando que uma média possa facilmente ser calculada sem que um fator influencie o resultado mais do que os outros. Para as equações a seguir, define-se que um grafo simples $G=(V,E)$ é constituído de um conjunto V de vértices, que representam os dispositivos da rede, e um conjunto E de arestas que conectam dois vértices do grafo, representando os enlaces da rede. O primeiro fator é o Fator de Redundância, que considera o número de dispositivos controladores na rede Nc :

$$FRed=1-\frac{1}{Nc},\ Nc \geqslant 1 \tag{1}$$

O Fator de Redundância retorna um valor proporcional ao número de controladores na rede, onde quanto maior a redundância, maior o valor. O número de controladores deve ser no mínimo um para que a rede possa ser considerada uma SDN. Esse fator avalia de forma simples a redundância no plano de controle, onde mais controladores significa melhor redundância.

O segundo é o Fator de Conectividade, que considera a k-conectividade da topologia. Sendo $Csdn$ um valor indicativo da capacidade de sobrevivência da topologia, onde $Kglobal$ é o valor de k-conectividade do grafo da topologia, no qual ambos os planos são considerados, e Kpd o valor de k-conectividade somente do plano de dados, ou seja, do grafo induzido da topologia que contém apenas os comutadores:

$$Csdn=\frac{Kglobal+Kpd}{2} \tag{2}$$

O valor de $Csdn$ é a média simples de dois valores de k-conectividade da rede.

Dois valores são utilizados pois o número de falhas que deixem o plano de controle desconexo do plano de dados não é necessariamente o mesmo número de falhas que deixem o plano de dados desconexo. Portanto são analisados a k-conectividade do grafo, que testa a capacidade de sobrevivência entre todos os dispositivos, e do grafo induzido do plano de dados, constituído somente pelos dispositivos encaminhadores, tal como em uma rede tradicional. Com esse valor, se obtém o Fator de Conectividade:

$$FC = 1 - \frac{1}{Csdn} \tag{3}$$

O terceiro é o Fator de Eficiência de Rede, que considera o valor da métrica de eficiência. No entanto, como a eficiência de comunicação que se deseja investigar é a do plano de controle e entre os planos, são considerados apenas os caminhos que conectem um controlador ou a outro controlador ou a um comutador. Sendo Vc o conjunto dos vértices que correspondem a controladores, $Nmc(Vc, V)$ o número de menores caminhos que conectem todo vértice controlador a todo outro vértice e $d(i, j)$ o comprimento do menor caminho entre quaisquer vértices i e j :

$$E(Vc) = \frac{1}{Nmc(Vc, V)} \sum_{i \neq j} \frac{1}{d(i, j)}, \ i \in Vc, \ j \in V \tag{4}$$

Essa fórmula restringe o cálculo de eficiência aos caminhos que conectem os controladores aos outros dispositivos da rede, sejam outros controladores ou comutadores. Dessa forma o valor retornado é um indicativo da eficiência de comunicação do plano de controle na rede, que é a característica desejável a se analisar. Com a Equação 4, o Fator de Eficiência de Rede é obtido simplesmente com:

$$FE = E(Vc) \tag{5}$$

O quarto é o Fator de Vulnerabilidade, que considera o valor da métrica de vulnerabilidade, com o diferencial de essa ser obtida usando a eficiência obtida na Equação 4. Dessa forma, o impacto analisado é na eficiência do plano de controle. A métrica de vulnerabilidade, ao contrário da eficiência, pode assumir valores positivos ou negativos, num cenário onde a remoção de um nó aumenta o valor da eficiência da rede, e pode assumir um valor absoluto maior do que um, num cenário onde o impacto foi maior de cem por cento. Considerando isso, para valores entre um positivo e um negativo, o valor é transformado para um valor proporcional entre zero e um. Para um valor absoluto maior do que um, o valor é limitado ao valor mínimo de zero e máximo de um. Dessa forma, valores de impacto de até cem por cento positivo ou negativo são considerados e transformados para obedecerem os limites estabelecidos, sem perda de informação. Sendo $V(Vc)$ o valor de vulnerabilidade do plano de controle, a vulnerabilidade transformada é obtida conforme a seguinte fórmula:

$$Vt = \begin{cases} \frac{V(Vc)+1}{2}, & |V(Vc)| \leqslant 1, \\ 1, & V(Vc) > 1, \\ 0, & V(Vc) < -1. \end{cases} \quad (6)$$

Com esse valor, o Fator de Vulnerabilidade é obtido por:

$$FV = 1 - Vt \quad (7)$$

Com essa fórmula, quanto maior o valor de vulnerabilidade, pior é o impacto de uma falha na eficiência do plano de controle, e menor o valor retornado pelo fator.

O quinto é o Fator de Eficiência de Domínio, que considera a eficiência de cada controlador aos comutadores sob seu controle direto. Em implementações do paradigma SDN, como na arquitetura *OpenFlow* [McKeown et al. 2008], cada dispositivo encaminhador é alocado a um controlador, que será responsável por trocar as mensagens de controle com esse dispositivo. É possível configurar múltiplos controladores em cada dispositivo, como sobressalentes para o controlador principal, com quem o dispositivo deve se comunicar (ONF, 2015). Nesse cenário, é interessante verificar a eficiência de comunicação de cada controlador com seus dispositivos alocados, que será referenciado a partir de agora como o domínio do controlador. Sendo c um vértice correspondente a um controlador que tenha dispositivos em seu domínio, $Vd(c)$ o conjunto de vértices correspondentes a esses dispositivos e $Nmc(c, Vd(c))$ o número de menores caminhos que conectem o vértice controlador a todos os dispositivos de seu domínio, a eficiência desse domínio é obtido por:

$$E(c) = \frac{1}{Nmc(c, Vd(c))} \sum \frac{1}{d(c,i)}, \quad c \in Vc, \quad i \in Vd(c) \quad (8)$$

Calculando a eficiência de cada controlador com seu domínio, o Fator de Eficiência de Domínio é obtido como o mínimo dessas eficiências:

$$FEd = min(E(c)), \quad c \in Vc \quad (9)$$

O Fator de Eficiência de Rede e o Fator de Eficiência de Domínio fornecem indicativos de características diferentes da rede. Um valor alto no Fator de Eficiência de Domínio indica que os controladores estão distribuídos na topologia, cada um tendendo ao centro de seus respectivos domínios, o que é desejável, pois a proximidade de cada controlador aos seus dispositivos alocados tende a uma maior eficiência em sua comunicação. Um valor alto no Fator de Eficiência de Rede também é desejável, pois se todos os controladores estiverem numa posição central na topologia, na ocorrência de falha de um controlador, outro controlador sobressalente poderá absorver os dispositivos afetados em seu domínio com pouco impacto na eficiência de comunicação entre os planos. Um equilíbrio entre estes fatores é o cenário ideal: cada controlador tende ao

centro de seu domínio, assim como tende ao centro da rede.

Combinando esses cinco fatores, obtém-se o Fator de Resiliência de Média de Múltiplas Métricas:

$$FRM3 = \frac{PRed \times FRed + PC \times FC + PE \times FE + PV \times FV + PEd \times FEd}{PRed + PC + PE + PV + PEd} \quad (10)$$

Cada fator é multiplicado por um peso que pode ser definido arbitrariamente conforme as características mais prioritárias para uma determinada análise. A análise realizada na seção seguinte é realizada sem a atribuição de prioridades, sendo todos os pesos definidos com valor um, tornando a equação uma média simples entre os fatores.

3.3. Algoritmos de aprimoramento topológico

Um algoritmo de aprimoramento topológico por redistribuição de arestas se propõe a transformar uma topologia, usando de alguma estratégia para remanejar as arestas existentes [Beygelzimer et al. 2005]. No contexto de SDN, é possível utilizar estes algoritmos para alterar conjuntamente as conexões dos controladores e dos comutadores da rede, aprimorando toda a estrutura topológica em uma determinada função objetivo. Neste trabalho foram implementados três algoritmos:

- Redistribuição de Aresta Randômica: Uma aresta aleatória é removida, e uma aresta aleatória é adicionada, desde que não seja a mesma aresta [Beygelzimer et al. 2005].
- Redistribuição de Aresta Randômica Preferencial: Uma aresta aleatória é removida, e o vértice de menor grau conectado por esta aresta é conectado a outro vértice aleatório com uma nova aresta [Beygelzimer et al. 2005].
- *Smart Rewire*: Duas arestas aleatórias são removidas, e entre os vértices afetados, duas novas arestas são adicionadas, uma entre os dois vértices de maior grau, e outra entre os vértices de menor grau [Bai Liang et al. 2015].

O algoritmo de Redistribuição Randômica realiza uma redistribuição completamente randômica e é utilizado como base para comparação. O algoritmo de Redistribuição Preferencial busca preservar o grau dos vértices menos conectados durante a distribuição e apresentou melhores resultados no trabalho de referência. O algoritmo *Smart Rewire* emprega similarmente uma estratégia que busca preservar o grau dos vértices mais e menos conectados.

Esses algoritmos foram selecionados pela simplicidade e versatilidade, que permitem que o FRM3 fosse utilizado como função objetivo para um algoritmo de "subida de encosta" (*hill-climbing*). Os algoritmos realizam alterações na topologia até que ultrapassem um determinado número de tentativas sem que haja nenhum sucesso em aumentar o valor de FRM3. No experimento realizado, o valor escolhido é de dez tentativas, sendo que valores maiores não demonstraram ganhos significativos nos experimentos realizados. Transformações que desconectem o grafo não são permitidas, e

transformações que gerem efeitos negativos ao valor de FRM3 são revertidas.

4. Resultados

Para explorar a aplicação do fator de resiliência proposto, este foi empregado como métrica da função objetivo de algoritmos de aprimoramento topológico, para analisar e aprimorar as características de resiliência das topologias de teste. Para esse fim, foram utilizadas topologias disponibilizadas publicamente na *Internet Topology Zoo*[1], sob os quais experimentos foram conduzidos e resultados obtidos conforme descrito na subseção a seguir. Os valores retornados pelo FRM3 são então analisados, verificando como cada característica topológica influenciou nos resultados. Posteriormente são descritos como os algoritmos de aprimoramento topológico que foram aplicados nas topologias afetaram os valores retornados pelo FRM3.

4.1. Metodologia

Para os testes realizados foram utilizadas as topologias disponibilizadas na *Internet Topology Zoo* para o plano de dados. Como tais topologias são de redes tradicionais, fica faltando definir adequadamente o plano de controle. Neste contexto, para cada topologia foram adicionados vértices controladores posicionados aleatoriamente, ou seja, foram selecionados vértices aleatoriamente e conectados um vértice controlador a cada um. A quantidade de controladores a serem adicionados em uma topologia foi definido como uma porcentagem do número de nós da rede, especificamente 25%, 50% e 75%. Para definir os dispositivos alocados para cada controlador, ou seja, o domínio de cada controlador, cada dispositivo foi simplesmente alocado ao controlador mais próximo conforme o comprimento do menor caminho. Caso um dispositivo esteja a mesma distância de múltiplos controladores, este é alocado aleatoriamente a um desses controladores.

Como o posicionamento dos controladores na rede é feito aleatoriamente, e como é discutido neste trabalho e observado nos trabalhos relacionados que o posicionamento dos controladores na rede afetam os valores das métricas de resiliência, cada topologia possui o posicionamento dos controladores repetidos trinta vezes para se obter o caso médio da avaliação do FRM3, diminuindo a influência de melhores e piores casos nos valores observados. Foram utilizadas 39 topologias do *Internet Topology Zoo*, contendo desde 5 até 16 comutadores, os quais após a inserção dos controladores em diferentes proporções e em diferentes posições, resultaram em mais de 3500 diferentes topologias SDN.

Foi desenvolvido um programa em Java que gerou as topologias SDN a partir das topologias da *Internet Topology Zoo* como descrito anteriormente, obteve as métricas e calculou o FRM3 conforme detalhado na seção anterior, aplicando os algoritmos implementados de aprimoramento topológico nas topologias geradas,

1 Disponível em: http://www.topology-zoo.org/

retornando o novo FRM3 após o aprimoramento. Os gráficos gerados a seguir foram obtidos a partir desses testes, realizados em um servidor com as seguintes especificações: Intel(R) Xeon(R) CPU E5-2650 0 @ 2.00GHz, com 65 GB RAM, e sistema 64-bit. Sistema Operacional GNU/Linux Ubuntu, Kernel 3.13.0-48-generic.

4.2. Análise de resultados

Os valores dos fatores e do FRM3 antes e depois da aplicação dos algoritmos descritos na subseção 3.3 são ilustrados na Figura 1.

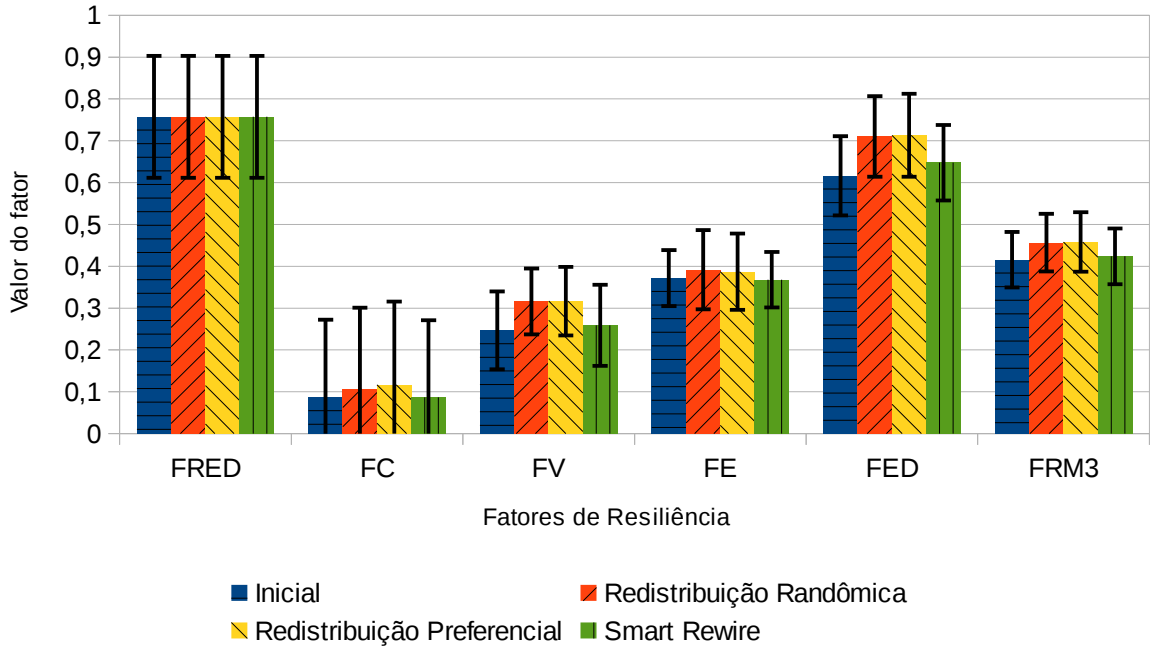


Figura 1. Média e desvio padrão dos valores dos fatores de resiliência nas topologias de experimentação antes e após a aplicação dos algoritmos de aprimoramento.

A maioria das topologias de teste possui múltiplos controladores, elevando o valor do Fator de Redundância, e essa redundância de controladores na rede naturalmente faz com que os comutadores tenham mais opções de alocação para um controlador, aumentando as chances de que tenha um controlador mais próximo, fazendo com que o Fator de Eficiência de Domínio também seja alto. No entanto, o posicionamento impensado dos controladores e a própria estrutura topológica do plano de dados resulta em Fatores de Conectividade, Eficiência de Rede e Vulnerabilidade mais baixos. Não há a preocupação em aproximar os controladores do centro da rede, e a fraca conectividade do plano de controle e de dados faz com que o valor final de FRM3 seja apenas mediano.

Com a aplicação dos algoritmos há um ganho em todos os fatores, com a exceção do Fator de Redundância, pois os algoritmos não alteram o número de controladores na rede. Os algoritmos de Redistribuição Randômica e Preferencial aumentaram a média do FRM3 em 9,7% e 10% respectivamente, apresentando ganhos em múltiplos fatores. O algoritmo de Redistribuição Randômica obteve melhorias em 89,6% das topologias, e o algoritmo de Redistribuição Preferencial aprimorou 91,8% das topologias. O algoritmo de *Smart Rewire* aumentou a média de FRM3 em 1,8%, obtendo resultados em 60,8% das topologias. A estratégia de redistribuição mais restritiva do *Smart Rewire* limitou a exploração por uma solução nas topologias testadas.

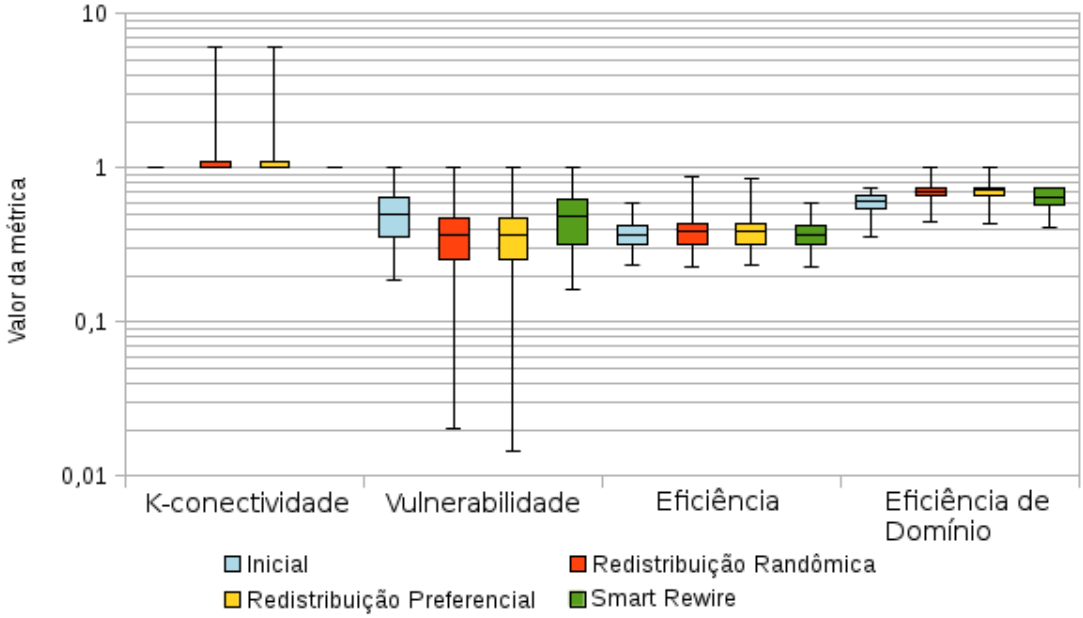


Figura 2. Distribuição em *boxplot* dos valores das métricas de análise antes e após a aplicação dos algoritmos de aprimoramento.

As formas que o ganho obtido no FRM3 reflete nas topologias pode ser visualizado na Figura 2. O impacto dos algoritmos de aprimoramento nas diferentes métricas reafirma os baixos ganhos do algoritmo de *Smart Rewire*, que não aprimorou a capacidade de sobrevivência das topologias, aumentou a média de eficiência da comunicação dos controladores para seus comutadores alocados em apenas 5% e diminuiu em apenas 4,7% em média a vulnerabilidade dessa comunicação. Os algoritmos de Redistribuição Randômica e Preferencial no entanto, aumentaram a k-conectividade da rede respectivamente em 6,4% e 6,9% das topologias, efetivamente aprimorando a capacidade de sobrevivência da rede à cenários de falha de dispositivo arbitrário. Também obtiveram aumento de 5,3% e 4% da média de eficiência de comunicação da rede, e aumento de 15,2% e 15,6% da média de eficiência de comunicação dos controladores para seus respectivos comutadores alocados, efetivamente diminuindo as distâncias dos menores caminhos da rede; adicionalmente

diminuindo em média 27,2% e 27,5% a vulnerabilidade dessa comunicação a falhas arbitrárias, através de caminhos disjuntos mais eficientes entre os dispositivos. O comportamento desses indicativos demonstra a eficácia do fator proposto como um indicativo informativo para a análise da rede, e para a aplicação como função objetivo.

Os algoritmos de Redistribuição Randômica e Preferencial apresentaram desempenho equivalente no aprimoramento das topologias testadas. O algoritmo *Smart Rewire* apresentou os menores ganhos devido a criteriosa estratégia de remanejamento de arestas aplicada, que limitou as transformações realizadas.

5. Conclusões e Trabalhos Futuros

É proposto um Fator de Resiliência de Múltiplas Métricas para Redes Definidas por *Software* que correlaciona múltiplas métricas de redes complexas, analisando diversas características de resiliência de uma topologia do paradigma SDN, levando em consideração os aspectos específicos a sua arquitetura. O fator proposto pode ser uma útil ferramenta na fase de planejamento de rede, permitindo uma análise objetiva e comparativa entre propostas de topologias e na aplicação computacional de algoritmos de aprimoramento topológico. Trabalhos relacionados demonstram pouca preocupação em correlacionar múltiplas métricas simultaneamente, e propõe soluções que otimizam apenas o posicionamento dos dispositivos no plano de controle, apesar de observarem que essa otimização é diretamente dependente da topologia formada por ambos os planos.

Topologias obtidas a partir de topologias disponibilizadas na *Internet Topology Zoo* foram analisadas e usadas como base para algoritmos de aprimoramento topológico de Redistribuição de Arestas, que obtiveram ganhos no fator proposto ao realizarem otimizações em características de resiliência de ambos os planos das topologias.

Como trabalhos futuros, pretende-se investigar de que forma o ajuste nos pesos dos componentes do fator proposto pode contribuir na aplicação dos algoritmos de aprimoramento topológico, além de comparar o desempenho de mais algoritmos, em mais topologias, para o aprimoramento topológico de Redes Definidas por *Software*.

Referências

Bai Liang, Xiao Yan-Dong, Hou Lv-Lin et al. (2015) “Smart Rewiring: Improving Network Robustness Faster”, Chin. Phys. Lett., 32(07):078901.

Beygelzimer A., Grinstein G., Linsker R. e Rish I. (2005) “Improving network robustness by edge modification”, Physica A: Statistical Mechanics and its Applications, 357(3-4):593–612, Novembro.

Farias F. N. N., Júnior J. M. D., Salvatti J. J., Silva S., Abelém A. J. G., Salvador M.R., e Stanton M.A. (2011) “Pesquisa Experimental para a Internet do Futuro: Uma Proposta Utilizando Virtualização e o Framework Openflow”, Minicurso, Simpósio

Brasileiro de Redes de Computadores e Sistemas Distribuídos.

Habibi D. e Phung Q. V. (2012). “Graph Theory for Survivability Design in Communication Networks”, New Frontiers in Graph Theory, Dr. Yagang Zhang (Ed.), ISBN: 978-953-51-0115-4, InTech, http://www.intechopen.com/books/new-frontiers-in-graph-theory/graph-theory-for-survivability-design-in-communication-networks

Heller B., Sherwood R., McKeown N. (2012) “The Controller Placement Problem”, HotSDN’12, Helsinki, Finland, Agosto.

Hock D., Hartmann M., Gebert S., Jarschel M., Zinner T., Tran-Gia P. (2013) “Pareto-Optimal Resilient Controller Placement in SDN-based Core Networks”, Proceedings of the 25th International Teletraffic Congress (ITC).

Lange S., Gebert S., Zinner, T., Tran-Gia P., Hock D., Jarschel M., Hoffman M. (2015) “Heuristic Approaches to the Controller Placement Problem in Large Scale SDN Networks”, IEEE Transactions on Network and Service Management, Volume: 12, Issue 1, ISSN: 1932-4537, Março.

Latora V.; Marchiori M. (2001) “Efficient behavior of small-world networks”, v. 87, n. 19, p. 198701. Disponível em: <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.87.198701>.

McKeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Shenker S., e Turner J. (2008) “Openflow: enabling innovation in campus networks”, SIGCOMM Comput. Commun. Rev., 38 (2):69–74, Março.

Metz J., Calvo R., Seno E. R. M., Romero R. A. F., Liang Z. (2007) “Redes Complexas: conceitos e aplicações”, Relatórios Técnicos do Instituto de Ciências Matemáticas e de Computação, USP, Janeiro.

Müller L. F., Oliveira R. R., Luizelli M. C., Gaspary L. P., Barcellos M. P. (2014) “Survivor: An enhanced controller placement strategy for improving SDN survivability”, Global Communications Conference (GLOBECOM) IEEE, ISBN: 978-1-4799-3512-3, Texas, Austin, USA, Dezembro.

ONF. (2015) “Openflow Switch Specification 1.5.1”, https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf, Março.

ONF. (2012) “Software-Defined Networking: The New Norm for Networks”, Open Networking Foundation, Abril.

Ros F. J., Ruiz P. M. (2014) “Five Nines of Southbound Reliability in Software-Defined Networks”, HotSDN’14, Chicago, IL, USA, Agosto.

Gerenciamento de recursos de serviços de Voz sobre IP baseado em Redes Definidas por *Software*

Paulo Roberto Vieira Jr [1], **Adriano Fiorese** [1], **Guilherme P. Koslovski**[1], **Anderson H. S. Marcondes**[1]

[1]Programa de Pós-Graduação em Computação Aplicada (PPGCA)
Universidade do Estado de Santa Catarina (UDESC) — Joinville, SC — Brasil

paulorvj@gmail.com, {adriano.fiorese, guilherme.koslovski}@udesc.br
anderson.marcondes@sfs.ifc.edu.br

Abstract. *The Voice Over IP technology (VoIP) allows people to communicate voice using the Internet as a means of transmission. VoIP calls between clients using the same Codec consume communication and computacional resources from VoIP server. Software Defined Networking (SDN) has been disseminating the decoupling between control and data plane in network technologies. Therefore, this paper aims to present a management approach based on SDN to reduce the use of resources in a VoIP server. Scenarios were emulated with and without the use of SDN to assess CPU utilization and bandwidth used in a VoIP server. The experimental results obtained indicate reduction on the consumption of the mentioned resources.*

Resumo. *A tecnologia Voz sobre IP (VoIP) permite que pessoas possam se comunicar com voz utilizando a Internet como meio de transmissão. Chamadas VoIP entre clientes que possuem o mesmo Codec consomem recursos de comunicação e computacionais dos servidores VoIP. Software Defined Networking (SDN) difundiu o desacoplamento entre o plano de controle e de dados. Assim, este trabalho apresenta uma solução baseada em SDN para reduzir a utilização de recursos em um servidor VoIP. Foram emulados cenários com e sem o uso de SDN, quantificando a utilização de CPU e largura de banda em um servidor VoIP. Resultados experimentais obtidos indicam redução no consumo dos recursos citados.*

1. Introdução

As redes de comutação de pacotes, inicialmente projetadas para realização de tarefas sem requisitos de qualidade de serviço, como tráfego de correio eletrônico e transferência de arquivos, foram rapidamente difundidas. Um grande número de *switches*, roteadores, *firewalls* e outros equipamentos foram e ainda são implantados, desenvolvidos por diversos fornecedores. Usualmente, cada equipamento deve ser configurado pelo administrador através de comandos de baixo nível no *software* embarcado no equipamento. Esse crescimento tornou as redes fechadas, proprietárias e verticalmente integradas, ossificando a infraestrutura [Handley 2006] e limitando a experimentação de novos protocolos e soluções de gerenciamento [McKeown et al. 2008].

Os protocolos inicialmente propostos cumpriram seu papel para o objetivo traçado. Entretanto, no contexto atual, as redes de computadores desempenham papéis

que vão além do inicialmente projetado e são essenciais para empresas, universidades, centros de pesquisa, ambientes governamentais e atividades diárias. Além do tráfego inicialmente previsto, as redes transportam pacotes com dados oriundos de diversas fontes, como vídeos, voz e outras aplicações que requerem qualidade de serviço (QoS, *Quality of Service*) durante sua execução. Em resposta às limitações da infraestrutura física de redes tradicionais, surgiu o paradigma das Redes Definidas por *Software* (*Software Defined Networking* - SDN), quebrando a integração vertical e separando a camada de dados da camada de controle. Assim, novas ideias e protocolos podem ser experimentados sem prejudicar o tráfego de produção [Jain and Paul 2013] [Kreutz et al. 2014].

O restante do trabalho está organizado da seguinte forma: A Seção 2 apresenta o embasamento teórico para desenvolvimento do trabalho. A Seção 3 discute os trabalhos relacionados, enquanto a Seção 4 apresenta a solução proposta. A Seção 5 apresenta a análise experimental e a Seção 6 as considerações finais.

2. Voz Sobre IP e Redes Definidas por *Software*

A área de telecomunicações sofreu alterações importantes relacionadas com a representação e transporte dos dados de voz. A disponibilização de Voz sobre o Protocolo Internet (*Voice over Internet Protocol*, VoIP) [Goode 2002] vem aproveitando a expansão das redes de comutação de pacotes para permitir o tráfego de voz utilizando a Internet como meio de transmissão. Atualmente, VoIP é utilizado por usuários domésticos e por empresas geograficamente distribuídas, permitindo a comunicação através de diversos dispositivos (*e.g.*, telefones IP, *smartphones*, celulares, *tablets*, *softphone*).

Vários fatores influenciam a qualidade de uma chamada VoIP, como: o protocolo de sinalização, a escolha do codificador/decodificador (*Codec*), atributos de segurança, a habilidade de cruzar *firewalls* e servidores de tradução de endereços (*Network Address Translation* - NAT), entre outros. Sobretudo, fatores como a configuração do servidor VoIP ou a localização dos clientes na rede, fazem com que os pacotes de voz trafeguem pelo servidor VoIP consumindo CPU e largura de banda.

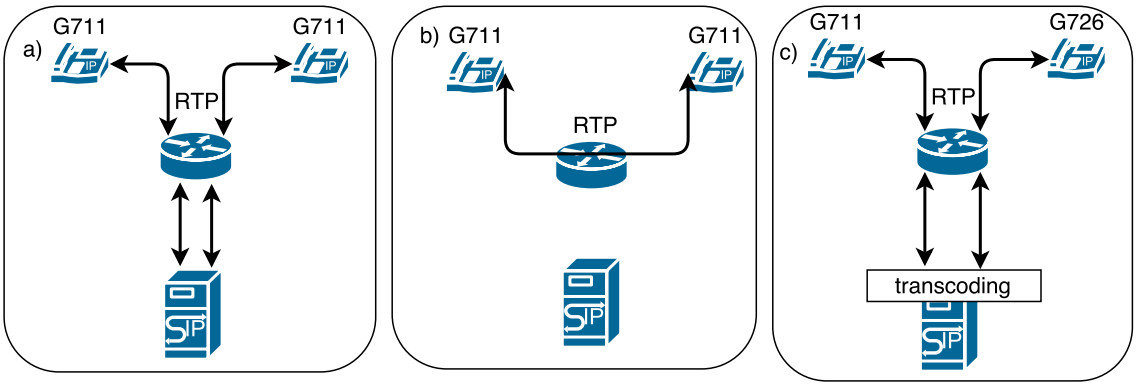


Figura 1. Exemplos de tráfego VoIP com *Codec* único (a e b) e com *Codecs* distintos (c).

A Figura 1 apresenta três maneiras pelas quais pode ocorrer o tráfego de dados em chamadas VoIP: (a) Os dados de uma chamada com o mesmo *Codec* de áudio (G711, por exemplo) são trafegados de um cliente para o outro através do servidor VoIP. Nesse

caso o servidor está no caminho do áudio e dois canais são criados, um para cada cliente. O servidor VoIP lê o áudio de um canal e escreve no outro, aumentando o uso de memória e de CPU durante a leitura e escrita dos canais. Essa configuração é comumente utilizada para monitorar chamadas, coletando informações relacionadas com a qualidade do atendimento prestado por centrais de vendas. (b) O tráfego de sinalização de clientes com o mesmo *Codec* de áudio e os dados de voz são trafegados diretamente entre os clientes. Nesse caso, o servidor VoIP não está no caminho do áudio e não é possível gravar a chamada, acrescentar música de fundo, transferir ou colocar a chamada em espera. Entretanto, essa abordagem reduz o uso de CPU, memória alocada e largura de banda utilizada pelo servidor. (c) Clientes com *Codecs* diferentes e os dados são trafegados sempre pelo servidor VoIP. Semelhante ao cenário (a), o servidor está no caminho do áudio e dois canais são criados, um para cada cliente. Entretanto, o servidor VoIP efetua a tradução dos pacotes de voz de acordo com os *Codecs* utilizados para que os clientes possam receber e ouvir o áudio [Goode 2002], aumentando o consumo dos recursos do servidor.

Clientes localizados em domínios administrativos protegidos por *firewall* ou utilizando NAT aumentam a complexidade do problema. O servidor VoIP não é capaz de resolver os endereços IPs privados dos clientes em uma chamada. Assim, todo o tráfego de voz é baseado nos IPs públicos dos clientes e nesse caso, o tráfego passa em sua totalidade pelo servidor VoIP, consumindo CPU e largura de banda [Karapantazis and Pavlidou 2009, Lin et al. 2010, Park et al. 2008, Chen et al. 2008, Yeryomin et al. 2008, Khlifi et al. 2006, Khirul et al. 2007].

Especificamente, o consumo de largura de banda é uma preocupação quando se discute sobre a adoção da tecnologia VoIP. Empresas com matriz e filiais podem adotar a tecnologia VoIP para reduzir custos com chamadas telefônicas. Porém, muitos planos de utilização da Internet oferecem largura de banda variada e muitas vezes com diferentes taxas para *download* e *upload*. A largura de banda disponível é importante para o provisionamento do serviço influenciando diretamente na quantidade e qualidade de chamadas simultâneas que são suportadas.

Nesse contexto, o presente trabalho tem como objetivo o desenvolvimento de uma solução baseada em SDN que permita reduzir o consumo de recursos na disponibilização do serviço VoIP, aplicável em ambiente LAN. A solução é baseada em SDN e faz com que o tráfego de áudio apresentado na Figura 1 (a) ocorra da maneira apresentada na Figura 1 (b), ou seja, permite que chamadas entre clientes que utilizem o mesmo *Codec* possam ser realizadas com o tráfego de voz fluindo diretamente entre os clientes. Dessa maneira há redução no consumo de CPU e largura de banda no servidor e consequentemente no segmento de rede onde o servidor VoIP está localizado.

VoIP é uma tecnologia que permite o tráfego de voz sobre redes comutadas, inclusive na Internet [Karapantazis and Pavlidou 2009]. Dentre as funcionalidades oferecidas pela tecnologia, destacam-se chamadas de longa distância e internacionais, *voicemail*, identificação de chamadas, conferência, monitoração de conversas, chamada em espera, encaminhamento de chamadas e integração com a rede pública de telefonia (*Public Switched Telephone Network* - PSTN), entre outras. Em suma, oferece serviços com a eficiência de uma rede IP e com a qualidade de uma rede tradicional de telefonia. Além dessas funcionalidades a principal razão para sua adoção é a redução de custos, uma vez que uma única rede suporta ambos os serviços: voz e dados.

O funcionamento de chamadas VoIP é similar a uma chamada telefônica tradicional e envolve sinalização, codificação e decodificação. Uma chamada pode ser iniciada em um telefone IP, no qual a voz é digitalizada e transformada em pacotes através de codificadores. Esses pacotes são encaminhados na rede IP passando por roteadores e *switches* até o servidor VoIP ou diretamente até o destinatário (conforme representando pela Figura 1).

2.1. Estabelecimento de Sessões VoIP

O estabelecimento de conexões e o tráfego de dados VoIP é usualmente regido pelos protocolos *Session Initiation Protocol* (SIP), *Real-time Transport Protocol* (RTP) e *Real-Time Transport Control Protocol* (RTCP). SIP é um protocolo de controle (sinalização) da camada de aplicação que pode estabelecer, modificar e terminar sessões com um ou mais participantes [Rosenberg et al. 2002]. Em suma, qualquer tipo de sessão pode ser estabelecida utilizando o protocolo SIP, como por exemplo distribuição de multimídia e conferências, além de chamadas VoIP. Baseado em UDP, SIP utiliza o protocolo *Session Description Protocol* (SDP) para representar os metadados necessários para o estabelecimento das sessões [G. Camarillo 2006].

Por sua vez, RTP é um protocolo da camada de aplicação que provê entrega de dados fim-a-fim com características de tempo-real, como áudio e vídeo interativo. O pacote RTP carrega, além dos dados de seu *payload* a identificação do tipo de mídia, número de sequência, data e hora e informações para monitoramento da entrega [Schulzrinne et al. 2003]. Utilizando UDP, RTP não garante qualidade de serviço e nem provê um mecanismo para assegurar a entrega temporal da informação. Assim um número de sequência é incluído no pacote RTP permitindo ao receptor a reconstrução da sequência de pacotes quando necessário.

Para estabelecimento de uma chamada de áudio, um par de portas lógicas são utilizadas. Inicialmente, uma porta UPD é direcionada para o serviço RTP para que possa enviar e receber áudio. Uma segunda porta, com numeração em sequência à porta utilizada pelo RTP, é necessária para controle dos pacotes RTP, realizado por meio do protocolo RTCP. O último é utilizado para transmissão periódica de pacotes de controle para todos os participantes da sessão, provendo informações sobre a qualidade da transmissão dos dados.

2.2. Codificação e Decodificação de Voz

Um *Codec* é um algoritmo que permite o transporte de um sinal analógico através de linhas/canais digitais. Existem vários *Codecs* que diferem em complexidade, largura de banda necessária e qualidade de representação da voz. Quanto maior a qualidade obtida pelo *Codec*, maior é a largura de banda necessária para tráfego dos dados [Karapantazis and Pavlidou 2009]. Diferentes *Codecs* podem ser utilizados em uma chamada. Porém, é necessária a transcodificação da chamada, ou seja os dados precisam ser decodificados e recodificados para que o áudio seja ouvido nos terminais da chamada. Este processo consome processamento nos servidores e clientes VoIP, afetando em diversos casos a qualidade percebida da voz [Goode 2002].

A *International Telecommunication Union-Telecommunication* (ITU-T) é o órgão que controla a aprovação de um *Codec*. A ITU-T avalia e atribui uma pontuação ao *Codec*

após um processo de testes. A pontuação é chamada de *Mean Opinion Score* (MOS) com valores de 1 (ruim) até 5 (excelente) [Union 2005b].

2.3. Redes Definidas por *Software*

SDN é um paradigma de rede emergente que separa a camada de controle da camada de dados, prometendo melhorar a utilização dos recursos de rede, simplificar o gerenciamento de rede, reduzir custos de operação e promover inovação e evolução [Akyildiz et al. 2014]. Em uma arquitetura SDN, o nível mais baixo representa a infraestrutura de rede ou camada de dados, ou seja, nessa camada encontram-se apenas os dispositivos que apenas encaminham o tráfego baseado nas decisões da camada de controle. A camada intermediária é a camada de controle e o elemento chave dessa camada é o controlador, responsável por gerenciar todos os dispositivos da rede, adicionando, removendo ou alterando as entradas nas tabelas de encaminhamento.

A comunicação entre o controlador e os dispositivos encaminhadores pode ser feita através de diversos protocolos. Entretanto, OpenFlow destaca-se como um protocolo aberto que padroniza a comunicação entre as camadas de controle e de dados, definindo como um *software* pode programar a tabela de fluxos em diferentes *switches* [McKeown et al. 2008]. De acordo com a especificação do protocolo [Open Networking Foundation 2012], a arquitetura de um *switch* SDN deve possuir os seguintes componentes: uma ou mais tabelas de fluxos, uma tabela de grupo, um canal de comunicação seguro para um controlador, uma tabela de métricas e um ou mais controladores. Cada tabela contém um conjunto de registros de fluxos e cada fluxo consiste em regras de correspondência, contadores e um conjunto de ações que devem ser aplicadas nos pacotes[1] correspondentes.

O controlador é responsável por adicionar, atualizar e remover registros da tabela de fluxos e pode ser centralizado ou distribuído. Em SDN, as aplicações gerenciam a rede através das interfaces de programação (*Application Programming Interface* - API) definidas pelos controladores. Estas APIs fornecem um nível de abstração que permite às aplicações não depender de detalhes específicos da infraestrutura, facilitando a utilização das funcionalidades disponibilizadas pelo controlador para executarem suas funções adequadamente. Como exemplos de aplicações nesta camada da arquitetura SDN pode-se citar os protocolos de roteamento, balanceamento de carga, monitoração, detecção de ataques, aplicação de políticas e *firewall*.

Para o propósito deste trabalho, um fluxo corresponde à tupla de informações de identificação dos terminais de comunicação VoIP (cliente origem e destino), e deve ser instalado nos *switches* SDN que fazem parte do caminho entre eles. Essas informações referem-se aos endereços MAC, IP e portas lógicas utilizadas na sessão SIP, no tráfego de dados de voz por meio do protocolo RTP e no tráfego de dados de controle por meio do protocolo RTCP.

3. Trabalhos Relacionados

Até a data de escrita deste trabalho não encontramos nenhuma literatura estreitamente relacionada. Apenas trabalhos que de alguma forma melhoram o desempenho de redes VoIP

[1] Uma série de bytes compreendendo um cabeçalho, um *payload* e um *trailer*, nessa ordem, tratado como unidade para fins de processamento e encaminhamento. O tipo de pacote padrão é um quadro *Ethernet*, porém outros tipos de pacote também são suportados.

ou trabalhos que dimensionam a carga de um servidor VoIP sem o uso de SDN. Portanto, as opções de comparação de resultados são limitadas. [Maribondo and Fernandes 2016] propuseram um método de adaptação de *Codecs* em redes SDN (*Codec Adaptation over SDN* - CAoS). O trabalho é baseado na arquitetura SDN e segundo os autores implementa uma abordagem para evitar a degradação da voz em redes corporativas. A solução proposta reduz os efeitos do congestionamentos de rede em SDN, reduzindo a qualidade da voz em chamadas para um *Codec* de menor consumo de largura de banda quando ocorre congestionamentos. De acordo com os resultados obtidos, o CAoS permite que uma quantidade maior de chamadas possam ser efetuadas em períodos de muita utilização sem degradar exageradamente a qualidade da voz. A presente proposta pode ser incorporada ao CAoS. Originalmente, os autores não quantificaram ou consideraram a sobrecarga de servidores VoIP, fator que pode degradar o QoS das chamadas.

Em [Lee et al. 2014] é apresentado um *framework* chamado meSDN que habilita virtualização WLAN, QoS orientado as aplicações e melhora da eficiência energética em dispositivos Android. A solução é baseada em SDN e utiliza o Open Virtual Switch (OVS) [Pfaff et al. 2015] para monitorar e gerenciar o tráfego de aplicações móveis. Os autores argumentam que estender as capacidades de uma rede SDN à dispositivos móveis pode prover soluções para muitos problemas de rede como QoS, virtualização e diagnóstico de falhas. A solução proposta na Seção 4 reduz o consumo de largura de banda de servidores VoIP, podendo ser combinado com meSDN para reduzir o tráfego de dados.

Garg *et al.* apresentaram um estudo sobre as limitações das redes sem fio 802.11 (a/b) em suportar chamadas VoIP [Garg and Kappes 2003]. O estudo mostrou a quantidade de chamadas simultâneas que podem ser colocadas em uma célula 802.11 (a/b) da rede. Por exemplo, com a utilização do *Codec* g711 com 20ms de áudio, uma célula 802.11 (a/b) pode suportar somente de 3 a 12 chamadas. Esse fator limitante endossa o desenvolvimento da presente proposta: ao reduzir o tráfego de dados até servidores VoIP, a comunicação em uma célula pode ser utilizado ao máximo para tráfego de dados de chamadas diretamente entre clientes.

Em [Costa et al. 2015] é investigada a adequação do servidor *Private Branch Exchange* (PBX) Asterisk para fornecer capacidades de comunicação VoIP com um MOS aceitável para um grande número de chamadas. A métrica de probabilidade de bloqueio é usado para medir a capacidade do servidor PBX, enquanto o MOS é utilizado para avaliar a qualidade das chamadas de voz. Os resultados do trabalho mostraram que o servidor PBX Asterisk pode efetivamente lidar com mais de 160 chamadas de voz simultâneas com uma probabilidade de bloqueio de menos de 5%, proporcionando chamadas de voz com MOS acima da média 4. Conforme será discutido na Seção 5, o número máximo de chamadas suportadas por um servidor pode ser aumentado com o redirecionamento de tráfego (e consequentemente de carga) com SDN. Ainda, em [Costa et al. 2015], o MOS é medido por uma ferramenta chamada VoIPMonitor, que observa o tráfego VoIP e faz o cálculo do MOS seguindo a recomendação da ITU-T [Union 2005a]. A presente proposta é agnóstica as tecnologias de monitoramento utilizadas, podendo ser conectada com VoIPMonitor.

Por fim, em [Ali et al. 2009] é apresentado um estudo dos requisitos máximos de largura de banda e latência máximo de serviços VoIP em redes de *Worldwide Interopera-*

bility for Microwave Access - WiMAX, incluindo uma análise dos efeitos da compressão de cabeçalho dos pacotes, supressão do cabeçalho do *payload* e outros fatores que afetam a largura de banda. Usando SDN, a aplicação desenvolvida neste trabalho (Seção 4) reescreve informações em pacotes para diminuir o tráfego de dados até o servidor.

4. Módulo de Gerenciamento SDNVoIP

Esta seção apresenta SDNVoIP, uma aplicação para controle de tráfego VoIP em SDN. A aplicação gerencia o tráfego VoIP, reduzindo a utilização de CPU em servidores e a largura de banda utilizada em redes SDN. Para reduzir o uso de CPU nos servidores, SDNVoIP reconfigura as chamadas que utilizem o mesmo *Codec* para realizarem uma comunicação fim-a-fim, sem passar pelo servidor. Ou seja, o cenário da Figura 1(a) é transformado no cenário (b), sem a necessidade de reconfigurar os servidores VoIP.

4.1. Identificação e Encaminhamento de Fluxos

Para o entendimento de um fluxo VoIP baseado nas regras de correspondência das portas da camada de transporte, é necessária a compreensão do mapeamento entre clientes e o servidor VoIP. Assim, o mapeamento de portas em uma chamada normal é apresentado na Figura 2. Por exemplo, o cliente origem e cliente destino comunicam-se na porta SIP 20000 e 15000, respectivamente, com a porta SIP 5060 do servidor VoIP. Nesse caso, todo o áudio trafegado entre os clientes passa pelo servidor VoIP. Uma vez estabelecida a sessão SIP, o servidor recebe na porta RTP 18564 o áudio enviado pelo cliente origem pela porta 25000 e encaminha o áudio recebido pela porta 16181 para o cliente destino que está aguardando na porta RTP 30000. Como as portas RTP e RTCP são definidas em pares, os pacotes RTCP do cliente origem na porta 25001 são recebidos pelo servidor na porta 18565 e encaminhados do servidor na porta 16182 para o cliente destino que aguarda na porta RTCP 30001.

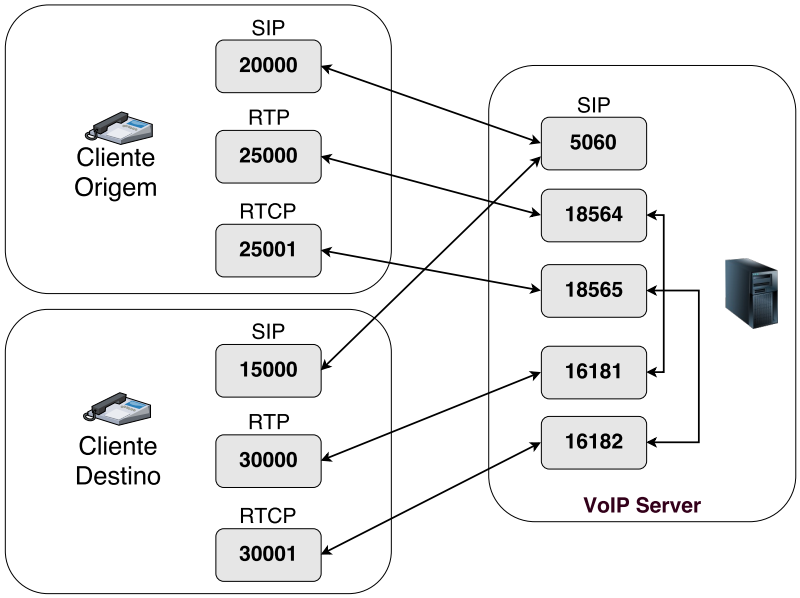


Figura 2. Mapeamento de portas em uma chamada VoIP normal.

Para caracterizar um fluxo de dados, o tráfego SIP, as portas RTP e RTCP utilizadas originalmente na comunicação entre cliente origem-servidor VoIP e servidor VoIP-

cliente destino são descobertas. Da mesma forma, os endereços de enlace e IP envolvidos, são identificados. Essa descoberta subsidia a instalação dos fluxos com as ações. Assim, após o SDNVoIP completar a instalação das entradas nas tabelas de encaminhamento dos *switches*, todo o tráfego RTP e RTCP é direcionado diretamente entre os clientes. Ou seja, os pacotes enviados pela porta 25000 do cliente origem são encaminhados diretamente para a porta 30000 do cliente destino e os pacotes enviados pela porta 25001 do cliente origem são encaminhados para a porta 30001 do cliente destino.

4.2. Implementação do Módulo

SDNVoIP foi desenvolvido como um módulo do controlador Floodlight [Project Floodlight 2016]. Dentre os controladores SDN existentes, Floodlight foi selecionado por apresentar um estágio maduro de desenvolvimento e ser adotado em diversos cenários experimentais e industriais. Quanto a implementação, *Floodlight* é um controlador centralizado desenvolvido em linguagem *Java*, que suporta *switches* físicos e virtuais, baseados no protocolo OpenFlow. O módulo SDNVoIP mantém informações dos clientes que estão executando chamadas, permitindo a instalação dos fluxos nos *switches* OpenFlow no caminho entre eles. Assim, pode-se fazer a instalação dos fluxos em dois momentos: quando o cliente destino envia a mensagem SIP/SDP de resposta para o servidor VoIP, respondendo a chamada feita pelo cliente origem; ou quando inicia o tráfego RTP entre o cliente origem e cliente destino.

Quatro fluxos por chamada são instalados nos *switches* SDN. Dois fluxos para o caminho no sentido origem-destino e destino-origem dos pacotes RTP; e dois fluxos para o caminho no sentido origem-destino e destino-origem dos pacotes RTCP. Para cada fluxo VoIP, sete ações [Open Networking Foundation 2012] são aplicadas nos pacotes RTP e RTCP que trafegam no *switch* correspondente:

(i) Alterar o endereço MAC de origem para o endereço MAC do servidor VoIP;
(ii) Alterar o endereço MAC de destino para o endereço MAC do cliente destino;
(iii) Alterar o endereço IP de origem para o endereço IP do servidor VoIP;
(iv) Alterar o endereço IP de destino para o endereço IP do cliente destino;
(v) Alterar a porta RTP/RTCP de origem para a porta RTP/RTCP do servidor VoIP;
(vi) Alterar a porta RTP/RTCP de destino para a porta RTP/RTCP do cliente destino;
(vii) Saída do pacote em uma porta física do *switch*.

As entradas nas tabelas de fluxos devem ser instaladas nos *switches* que compõem o caminho da chamada com um tempo de expiração. Quando o término de uma chamada é sinalizado pelos clientes, a entrada é removida. Em caso de interrupção abrupta da chamada, as entradas relacionadas são automaticamente removidas dos *switches* após o tempo de expiração.

5. Análise Experimental

Esta seção apresenta a análise experimental realizada com SDNVoIP. Duas métricas são coletadas: o consumo de CPU nos servidores VoIP e a largura de banda consumida para tráfego de chamadas VoIP.

5.1. Ambiente de Testes

Para o desenvolvimento deste trabalho foram utilizados os seguintes recursos: *(i)* Como controlador foi utilizado o Floodlight em sua versão 1.2 e algumas modificações em seu código foram realizadas para que os protocolos SIP, RTP e RTCP fossem manipulados pelo módulo SDNVoIP. *(ii)* Como servidor VoIP foi utilizado o Asterisk versão 1.13, executado em um computador com processador Intel QuadCore de 3.2GHz com 8GB de memória RAM. Para gerar a carga (chamadas VoIP) no servidor VoIP foi utilizado o *softphone* Linphone versão 3.6.1. *(iii)* O *switch* utilizado foi o TP-Link WR1043 com *firmware* personalizado com o Open Wireless Router (OpenWRT) versão 15.01 e Open Virtual Switch (OVS) versão 2.3. A versão 1.3 do OpenFlow foi a utilizada.

5.2. Cenários Experimentais

Dois cenários foram utilizados para realização dos experimentos, com e sem a utilização de SDN, ambos em ambiente LAN.

Chamadas VoIP sem SDN

O cenário sem SDN, apresentado na Figura 3, possui o servidor VoIP, o *switch* TP-Link WR1043 (com *firmware* nativo) e um PC onde foram inicializadas as instâncias do *softphone* Linphone. Um *script* auxiliar faz a inicialização do *softphone* e recebe como parâmetro a quantidade de instâncias que devem ser executadas no *Linux Host*. Cada instância do *softphone* possui um cliente com nome de usuário u seguido por um número sequencial, assim os clientes possuem como nome de usuário $u1$, $u2$ e assim sucessivamente até o limite informado como parâmetro.

As chamadas são geradas por outro *script*, de acordo com a seguinte lógica: cliente $u51$ realiza uma chamada para o cliente $u1$, cliente $u52$ faz uma chamada para o cliente $u2$ e assim até o cliente final, $u100$ chamar o cliente $u50$. Assim, 50 chamadas simultâneas são geradas entre 100 clientes. O *softphone* foi configurado para atender às chamadas recebidas automaticamente e logo após o atendimento deverá realizar a reprodução de um arquivo de áudio. Todas as chamadas utilizaram o mesmo *Codec* (G711) e todo o áudio é trafegado pelo servidor VoIP. Dessa maneira a carga gerada (uso da CPU) no servidor VoIP é estabelecida com chamadas simultâneas entre dois clientes, trafegando áudio e simulando uma conversa em que os dois participantes falam e ouvem.

Chamadas VoIP com SDN

O cenário com SDN é similar ao cenário apresentado na Figura 3. As únicas diferenças são a adição do controlador Floodlight e a reconfiguração do *switch* TP-Link WR1043 com OpenWRT e OVS habilitados e comunicando-se com o controlador (*out-of-band*). Assim como no cenário sem SDN, todo o áudio entre os clientes é trafegado pelo servidor VoIP. Entretanto, após o controlador obter todas as informações necessárias, quatro fluxos para cada chamada são instalados no *switch*, fazendo com que o áudio das chamadas não trafegue pelo servidor VoIP, fluindo diretamente entre os clientes.

Embora a solução atue em LAN, um cenário como o de empresas com matriz e filiais (ambiente WAN), pode obter benefícios da solução apresentada no presente trabalho. A Figura 4 apresenta uma topologia de rede de uma empresa que possui uma matriz e uma filial. Uma chamada entre um cliente origem (CO) e um cliente destino (CD), ambos localizados na Filial 1, possui uma rota passando pelos *switches* F, A e D. Em uma rede não

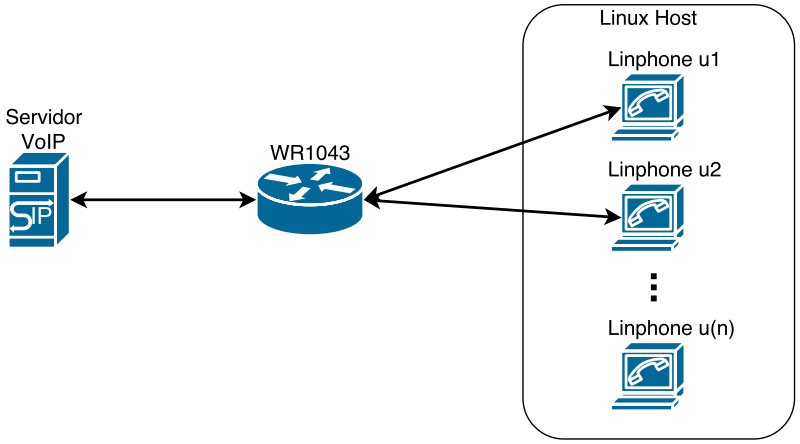


Figura 3. Cenário experimental sem SDN.

SDN, todo o tráfego de voz de CO para CD e vice-versa irá trafegar pelo servidor VoIP localizado na matriz, aumentando a utilização de CPU e consumindo largura de banda.

Já com implantação de *switches* OpenFlow e a utilização do módulo SDNVoIP, pode-se reduzir a utilização de CPU do servidor VoIP e o consumo de largura de banda no *link* Internet, mantendo-se uma arquitetura centralizada para o servidor VoIP, facilitando sua manutenção em termos de configuração. O módulo detecta a chamada feita entre CO e CD e instala os fluxos necessários no *switch* F fazendo com que o tráfego de voz ocorra diretamente entre os clientes na filial 1. Em suma, em ambiente WAN, pode-se obter redução na utilização de CPU no servidor VoIP e redução na utilização da largura de banda dos *links* de interconexão, fazendo com que o tráfego de voz ocorra somente na subrede onde estão localizados o CO e CD, quando for o caso.

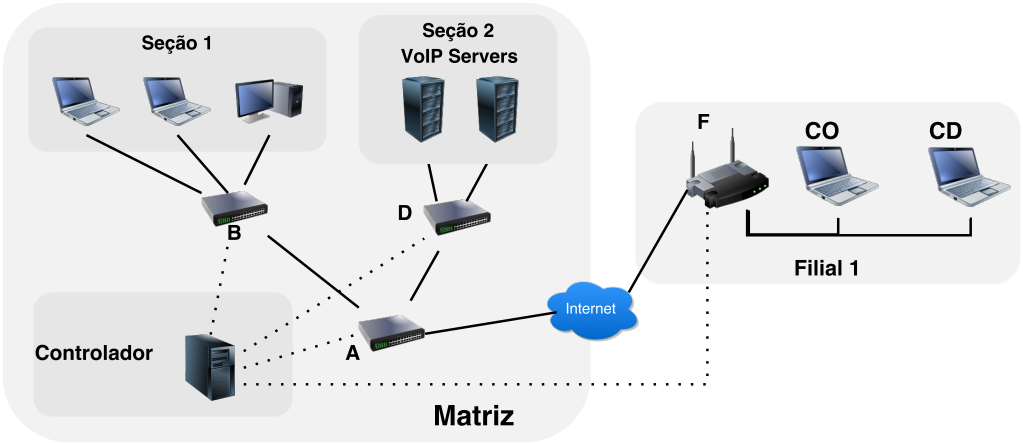


Figura 4. Cenário experimental com SDN: chamada entre clientes na Filial 1.

5.3. Análise dos Resultados

A taxa de utilização de CPU foi coletada com a ferramenta *mpstat* em intervalos de 1 segundo. Foram experimentados 5 minutos de conversação em chamadas simultâneas e os experimentos repetidos dez vezes, sobre os quais calculou-se a média aritmética e desvio padrão. A Tabela 1 apresenta os dados obtidos a respeito da utilização de CPU em

um servidor VoIP com as respectivas cargas. Observa-se que 300 chamadas simultâneas utilizando o mesmo *Codec* e todo o tráfego passando pelo servidor VoIP, gerando uma taxa de utilização de CPU de aproximadamente 27%. Já com o uso de SDN, todo o tráfego gerado pelas chamadas é encaminhado diretamente entre os clientes, fazendo com que a CPU do servidor VoIP fique aproximadamente 99% do tempo ociosa.

Tabela 1. Comparação do uso de CPU de um servidor VoIP.

Sem SDN						
Chamadas simultâneas	**50**	**100**	**150**	**200**	**250**	**300**
Uso de CPU%	4,03	9,22	13,92	19,35	22,78	27,92
Variância	0,53	5,59	1,33	1,04	1,37	2,74
Desv. Padrão	0,13	0,37	0,21	0,17	0,21	0,30
Com SDN						
Uso de CPU%	0,75	0,73	0,71	0,67	0,71	0,66
Variância	0,01	0,00	0,00	0,00	0,00	0,00
Desv. Padrão	0,08	0,01	0,03	0,06	0,03	0,02

A Figura 5 apresenta graficamente a taxa de utilização de CPU. O eixo x mostra a quantidade de chamadas simultâneas, enquanto que o eixo y apresenta a utilização de CPU em %. As barras verticais apresentam em preto a taxa de utilização de CPU sem o uso de SDN, enquanto que as barras na cor cinza apresentam a utilização de CPU com o uso de SDN. A solução com o uso de SDN reduz a utilização de CPU. Dessa maneira a CPU permanece mais tempo disponível para outras tarefas como a transcodificação entre clientes que efetuam chamadas com *Codecs* diferentes. Quanto maior a quantidade de chamadas, maior é o consumo de CPU. A solução apresentada neste trabalho com o uso de SDN pode reduzir o uso CPU para cargas maiores, desde que os clientes possuam o mesmo *Codec*.

A Tabela 2 apresenta as métricas de largura de banda sem o uso de SDN coletadas no servidor Asterisk utilizado, sendo rx a vazão de recepção, $rx\ pps$ a taxa de recepção de pacotes, em pacotes por segundo, tx a vazão de transmissão e $tx\ pps$ a taxa de transmissão em pacotes por segundo. Observa-se que para 50 chamadas simultâneas a vazão típica de recepção e transmissão é de 1,03Mbps e a vazão máxima de recepção foi de 1,04Mbps e 1,05Mbps para a transmissão. As taxas típicas de recepção e transmissão de pacotes foram de 5.09Kbps e 5.04Kbps respectivamente.

Tabela 2. Largura de banda em um servidor VoIP sem o uso de SDN.

Chamadas	**50**	**100**	**150**	**200**	**250**	**300**
Rx	1,03Mbps	2,05Mbps	3,07Mbps	4,10Mbps	5,23Mbps	6,05Mbps
Rx pps	5,10Kbps	10,13Kbps	15,19Kbps	20,23Kbps	25,84Kbps	28,44Kbps
Tx	1,03Mbps	2,07Mbps	3,10Mbps	4,14Mbps	5,28Mbps	6,11Mbps
Tx pps	5,04Kbps	10,08Kbps	15,12Kbps	20,16Kbps	25,73Kbps	28,32Kbps
Max Rx	1,04Mbps	2,05Mbps	3,09Mbps	4,11Mbps	6,15Mbps	6,48Mbps
Max Tx	1,05Mbps	2,07Mbps	3,12Mbps	4,14Mbps	6,20Mbps	6,54Mbps

Já para 300 chamadas simultâneas observa-se que a vazão típica de recepção e transmissão é de 6,05Mbps e 6,11Mbps respectivamente, e a vazão máxima de recepção

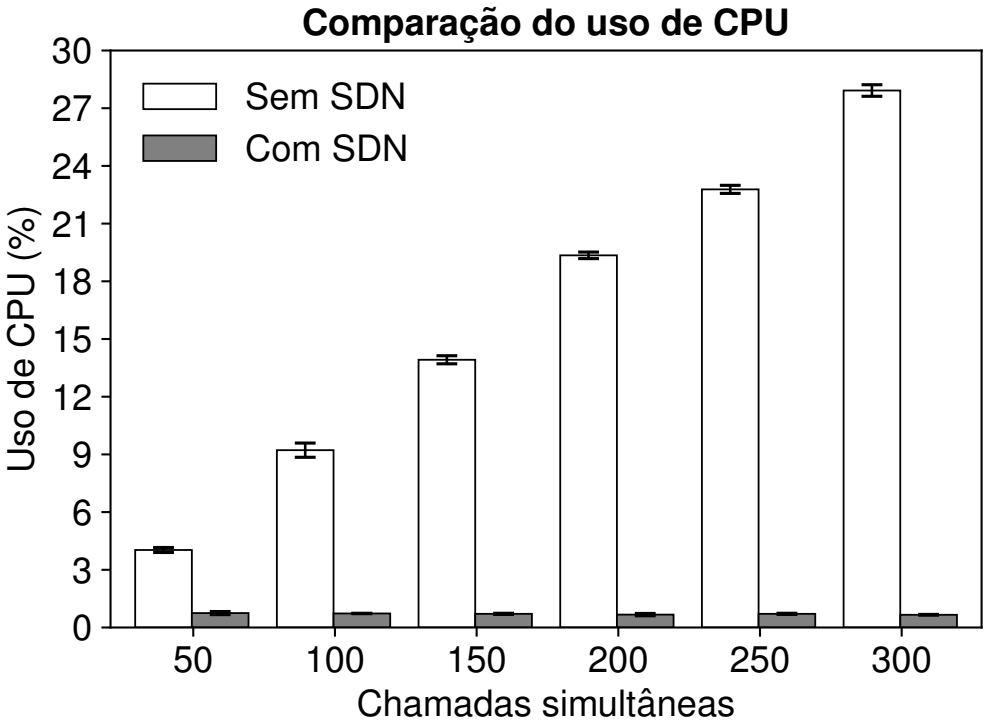


Figura 5. Comparação do uso de CPU em um servidor VoIP.

foi de 6,48Mbps e 6,54Mbps para a transmissão. As taxas de recepção e transmissão de pacotes foram de 28,44Kbps e 28,32Kbps respectivamente. A solução baseada em SDN proposta no presente trabalho reduz no maior cenário experimentado (300 chamadas com duração de 5 minutos), 6,05Mbps e 6,11Mbps de largura de banda de recepção e transmissão, respectivamente no segmento de rede do servidor VoIP.

6. Considerações Finais

O presente trabalho buscou agregar as vantagens e benefícios do uso de SDN na área de telecomunicações através da proposição do módulo SDNVoIP para o gerenciamento de chamadas VoIP em ambientes de rede de computadores que suportem OpenFlow. Os resultados obtidos demonstram que é possível reduzir a utilização de CPU em um servidor VoIP através da criação/utilização de fluxos que redirecionam o tráfego de voz diretamente entre as partes envolvidas que utilizam o mesmo *Codec*. Da mesma forma, há também a redução no consumo de largura de banda no segmento de rede do servidor VoIP.

A redução da utilização de CPU permite que haja maior disponibilidade de processador para chamadas que precisem de tradução de *Codec*, assim como a redução no consumo de largura de banda pode permitir que mais chamadas sejam completadas. Por fim, é possível redimensionar a quantidade de chamadas suportadas pelos servidores VoIP existentes em uma rede, sem a necessidade de aquisição de outro servidor para suportar mais chamadas.

Como trabalhos futuros, o módulo SDNVoIP pode ser modificado para redirecionar o tráfego de clientes que possuem *Codecs* diferentes e precisam ser transcodificadas,

para um servidor VoIP com menor utilização de CPU. Outra possibilidade é redirecionar o tráfego para um servidor VoIP com melhor localização em um ambiente de nuvem. Por fim, outra linha é o redirecionamento do tráfego de chamadas para o servidor VoIP com um caminho de rede com menos tráfego, evitando congestionamentos.

Agradecimentos

O trabalho foi parcialmente financiado pelos editais de apoio da UDESC e desenvolvido no LabP2D.

Referências

Akyildiz, I. F., Lee, A., Wang, P., Luo, M., and Chou, W. (2014). A roadmap for traffic engineering in sdn-openflow networks. *Computer Networks*, 71:1–30.

Ali, A. A., Vassilaras, S., and Ntagkounakis, K. (2009). A comparative study of bandwidth requirements of voip codecs over wimax access networks. In *2009 Third International Conference on Next Generation Mobile Applications, Services and Technologies*, pages 197–203. IEEE.

Chen, W.-E., Huang, Y.-L., and Chao, H.-C. (2008). Nat traversing solutions for sip applications. *EURASIP Journal on Wireless Communications and Networking*, 2008(1):1–9.

Costa, L. R., Nunes, L. S. N., Bordim, J. L., and Nakano, K. (2015). Asterisk pbx capacity evaluation. In *Parallel and Distributed Processing Symposium Workshop (IPDPSW), 2015 IEEE International*, pages 519–524. IEEE.

G. Camarillo, E. (2006). Session description protocol (sdp) format for binary floor control protocol (bfcp) streams. RFC 4583, RFC Editor.

Garg, S. and Kappes, M. (2003). Can i add a voip call? In *Communications, 2003. ICC'03. IEEE International Conference on*, volume 2, pages 779–783. IEEE.

Goode, B. (2002). Voice over internet protocol (voip). *Proceedings of the IEEE*, 90(9):1495–1517.

Handley, M. (2006). Why the internet only just works. *BT Technology Journal*, 24(3):119–129.

Jain, R. and Paul, S. (2013). Network virtualization and software defined networking for cloud computing: A survey. *IEEE Communications Magazine*, 51(11):24–31.

Karapantazis, S. and Pavlidou, F. N. (2009). VoIP: A comprehensive survey on a promising technology. *Computer Networks*, 53(12):2050–2090.

Khirul, I., Islam, M. K., and Hasan, K. A. (2007). An efficient approach for nat traversal problem on security of voice over internet protocol. In *10th International Conference on Computer and information technology, 2007*, pages 1–5. IEEE.

Khlifi, H., Gregoire, J., and Phillips, J. (2006). Voip and nat/firewalls: issues, traversal techniques, and a real-world solution. *IEEE Communications Magazine*, 44(7):93.

Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. (2014). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76.

Lee, J., Uddin, M., Tourrilhes, J., Sen, S., Banerjee, S., Arndt, M., Kim, K.-H., and Nadeem, T. (2014). mesdn: mobile extension of sdn. In *Proceedings of the fifth international workshop on Mobile cloud computing & services*, pages 7–14. ACM.

Lin, Y.-D., Tseng, C.-C., Ho, C.-Y., and Wu, Y.-H. (2010). How nat-compatible are voip applications? *IEEE Communications Magazine*, 48(12):58–65.

Maribondo, P. D. and Fernandes, N. C. (2016). Avoiding voice traffic degradation in ip enterprise networks using caos. In *Proceedings of the 2016 workshop on Fostering Latin-American Research in Data Communication Networks*, pages 34–36. ACM.

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74.

Open Networking Foundation (2012). OpenFlow Switch Specification. `http://www.cs.yale.edu/homes/yu-minlan/teach/csci599-fall12/papers/openflow-spec-v1.3.0.pdf`.

Park, C., Jeong, K., Kim, S., and Lee, Y. (2008). Nat issues in the remote management of home network devices. *IEEE network*, 22(5):48–55.

Pfaff, B., Pettit, J., Koponen, T., Jackson, E. J., Zhou, A., Rajahalme, J., Gross, J., Wang, A., Stringer, J., Shelar, P., Amidon, K., and Casado, M. (2015). The design and implementation of open vswitch. In *Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation*, NSDI'15, pages 117–130, Berkeley, CA, USA. USENIX Association.

Project Floodlight (2016). Project floodlight: Open source software for building software-defined networks. `http://www.projectfloodlight.org/floodlight`.

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E. (2002). Sip session initiation protocol. RFC 3261, RFC Editor.

Schulzrinne, H., Casner, R., and Frederick, V. J. (2003). RTP: A transport protocol for real-time applications. RFC 3550, IETF.

Union, I. T. (2005a). Methods for subjective determination of transmission quality. `http://www.itu.int/rec/T-REC-P.800-199608-I/en`.

Union, I. T. (2005b). Testing requirements and methodology. `https://www.itu.int/en/ITU-T/C-I/Pages/HFT-mobile-tests/Methodologies.aspx`.

Yeryomin, Y., Evers, F., and Seitz, J. (2008). Solving the firewall and nat traversal issues for sip-based voip. In *Telecommunications (ICT), International Conference on*, pages 1–6. IEEE.

Carrier-Grade SDN-Controlled WLAN-Sharing: a Performance Evaluation of OpenFlow-Enabled Commodity-based Hardware Networking Nodes

Maxweel S. Carmo[1,2], Thalyson L. de Souza[1], Alisson Medeiros[1], Augusto V. Neto[1,3], Sandino Jardim[1,2]

[1]Departamento de Informática e Matemática Aplicada
Universidade Federal do Rio Grande do Norte (UFRN), Natal-RN, Brasil

[2]Instituto de Ciências Exatas e da Terra
Universidade Federal de Mato grosso (UFMT), Barra do Garças-MT, Brasil

[3]Instituto de Telecomunicações – Aveiro, Portugal

max@ufmt.br, {thalysonluiz,alisson}@ppgsc.ufrn.br,
augusto@dimap.ufrn.br, sandino@ufmt.br

Abstract. *Advances in Internet of Things and smart cities approaches are resulting in an unprecedented number of devices requiring connectivity. Due to the ubiquitous presence of Wi-Fi networks, connectivity solutions obtained from residential WLAN-sharing are becoming increasingly popular, although they are still very restricted in scope, and offer a general connectivity service for users seeking to surf the Internet. We argue that with a proper network control plane and orchestration mechanisms like those provided by SDN, ISPs would be able to leverage the available Wi-Fi infrastructure so that it can offer network services that are tailored to the specific needs of mobile nodes and users. In this paper a general framework is set up to achieve this goal. As our approach explores low-cost commodity Wi-Fi access points and switches, we also include a study of the effects of SDN on a local network built on this class of equipment.*

1. Introduction

The development of 5G networks [Andrews *et al*. 2014] is being driven by the connectivity requirements of Internet of Things (IoT) [Tschofenig *et al*. 2015] applications, and its categories include both massive machine-type communications (mMTC) and ultra-reliable low-latency communications (URLLC). In mMTC, large numbers of low-cost devices can be defined with high scalability requirements and increased battery lifetime, whereas URLLC relates to mission-critical applications, in which an uninterrupted and robust exchange of data is of the utmost importance. In view of both aspects of high density and demand for network services with regard to IoT applications, ubiquitous connectivity plays a major role. The research community and industry are constantly searching for new solutions to suit these key requirements. According to the latest Cisco Visual Networking Index (VNI) Report [Forecast 2016], 51% of the total amount of mobile data traffic was offloaded through IEEE 802.11-based Wireless Local Area Network (WLAN) infrastructures in 2015, surpassing that of cellular traffic for the first time. Thus, the concept of the global wireless community

network is emerging as a highly promising asset among the available ubiquitous connectivity services. The guiding principle behind this concept is to create a global Wi-Fi ecosystem that can provide seamlessly, ubiquitous access to the Internet at millions of global hotspots. The success of the concept of a global wireless community network depends to a great extent on both the participation and support of Internet Service Providers (ISPs) and their corresponding customers. On the one hand, ISPs must provide carrier-grade tools to deliver a Wi-Fi service to millions of global hotspots, and thus be able to provision seamless access ubiquitously. On the other hand, customers must agree to share their own Wi-Fi WLAN with community members, thus turning a simple home WLAN into a dual-access Wi-Fi-shared infrastructure, for private (i.e. the owner) and public (i.e. community members) use. These features can make the WLAN-shared concept economically viable, as well as establishing another carrier-grade profitable service by reselling Internet connectivity [Biczók *et al*. 2011].

Traditionally, home WLANs embody commodity-based hardware networking devices, which means standard-issue off the shelf multifunctional nodes (i.e. the modem, switch, router and wireless access point functions in the same device); there are no outstanding features embedded in them and they are widely available for purchase at a low cost. However, these devices either lack appropriate tools or have inflexible proprietary software with very limited network resource control capabilities. Moreover, the customer must bear the cost of operating in-device tools, which are needed for setup network control functions (e.g., bandwidth reservation, classifying policies and queue disciplines). This is an easy task for a network professional but can be a complete nightmare for a home user who mainly wants to surf on the Internet without any concern about particular aspects of the system. For this reason, it makes sense for the WLAN-shared system to be under the carrier-grade control plane, which is likely to have expertise in this subject.

With regard to the question of provisioning dynamic resource control in WLAN-shared systems, the Software-defined Networking (SDN) [Boucadair and Jacquenet 2014] has emerged as an asset for ISPs. Through their approach to SDN network programmability, central controllers have the capacity to initialize, control, change, and manage the network behavior of the targeted networking devices dynamically, via open interfaces. The WLAN-shared systems that adopt the network programmable approach can be controlled by ensuring that the corresponding ISP installation embeds SDN controllers that feature appropriate carrier-grade network control applications, whilst home Wi-Fi routers must embed an open SDN control interface such as OpenFlow [ONF 2013]. Through this ecosystem, a given ISP can deploy customized SDN controllers featuring fine-grained resource control mechanisms that are tailored to its WLAN-shared community. In this way, it can obtain the capacity to provision optimal resource allocation at the carrier-grade level.

This paper makes an advance on our previous work [Carmo *et al*. 2017] by examining a real testbed that comprises a network topology of nine OpenFlow-capable commodity-based networking devices of different hardware platforms (instead of a single isolated router that was studied previously). The aim is to assess the suitability and resulting performance when carrying out SDN network programmable functions on commodity-based networking devices (beyond the domain of WLAN home devices), but from the perspective of a holistic network with redundant communication opportunities. In this kind of scenario, targeted devices run other networking services

apart from Wi-Fi APs, e.g. dynamic routing in the presence of redundant paths, which can provide a more accurate view of the SDN capabilities that can be exploited by commodity-based routers, and are suitable for different use cases.

The remainder of this paper is structured as follows Section II provides an overview of WLAN-shared Internet access. Section III introduces our SDN-based framework, while the preliminary results obtained from an assessment of the WLAN switches are discussed in section IV. Finally, the last section concludes the article.

2. Overview of WLAN-Shared Internet Access Solutions

The wide availability of home Wi-Fi networks has led to initiatives aimed at expanding Wi-Fi coverage for a community of customers. Among these initiatives (e.g., Comcast Xfinity[1] and WLAN to Go[2]), the most notable and successful WLAN-shared provider is FON[3]. During the last 10 years, FON has built the world's largest Wi-Fi network community, which comprises millions of people who have agreed to share their WLAN broadband connection with external users for seamless experience when accessing. FON relies on the partnership of leading carriers and Wi-Fi providers around the globe, and seeks to boost seamless connectivity service provisioning for member customers at millions of hotspots worldwide. An OpenWrt customized version has been deployed specifically for use in the FON Community to provide home customers with a FON WLAN-shared service. Following the installation of this FON-tailored OpenWrt version, the commodity-based Wi-Fi router becomes a "Fonera 2.0", and thus allows consumers to share their broadband connection and connect to other FON Spots around the world. With the Fonera device, two different Wi-Fi signals are created, one for private access and another for public access. In the former, the FON Spot owner has exclusive access and is thus able to leverage encrypted traffic service. In the latter, only registered FON users have the right to access an open (unencrypted) network system.

Although this system enhances opportunities for seamless connectivity, in our view existing WLAN-shared infrastructures are far from ideal as a means of providing an appropriate support for emerging scenarios like those involving IoT, smart cities, and the ever- increasing demand for connectivity by mobile users. For instance, FON only makes available one network per AP for public usage. We believe that to offer appropriate network services that can cater for the individual demands of mobile (and heterogeneous) nodes, it would be more appropriate to make multiple virtual networks available, each tailored to meet specific needs, i.e., the different V-WLANs could have different bandwidth capabilities and deploy different network functions to cope with the required behavior of the system, technical requirements and so on.

In addition, the one-size-fits-all public network made available by current WLAN-sharing systems, is designed for mobile users that want to surf the Internet. There is no differentiation of services on the basis of applications or types of devices. Moreover, there is no concern about optimizing the network resources at a global level to efficiently serve a larger number of users/devices. For instance, a number of people

[1] https://www.xfinity.com

[2] http://www.telekom.de/wlan-to-go

[3] https://fon.com

spend most of the day at work, which means their WLAN infrastructure is probably left with the minimum use. Thus, ISPs could set the private V-WLAN with minimum bandwidth allocations, and leave the remainder of the resources for the Public V-WLAN with the prospect of providing enhanced connectivity for the community members. When the WLAN owner reports back to his network, the ISP can re-adapt the bandwidth, and even use smart mechanisms to allocate the required resources in advance. In addition, there is no support for mobility management in the current systems, which can for instance, help devices to connect to the best public network available at a particular moment. All these requirements call for a networking control plane that is able to act on a globally perceived network topology to meet the diverse needs of the mobile nodes.

3. SDN-Controlled WLAN-Shared Framework

As highlighted in Section I, the WLAN-shared software for resource control envisions controlling the behavior of the corresponding networking devices to provision optimal use by private and open community members. In our opinion deploying the control plane at the carrier-grade level is the most appropriate manner to provide a fine-grained approach that can enable the WLAN owner to keep "surfing" the net with an acceptable quality of experience. Moreover, WLAN-shared residential infrastructures feature commodity-based Wi-Fi access points that are generally performance-constrained and low-cost to suit the needs of the end-user. Hence, the addition of the SDN approach to these types of nodes will certainly provide the prospect of overcoming additional performance issues, and ensuring suitability. Figure 1 illustrates our proposed carrier-grade SDN-controlled WLAN-shared architecture.

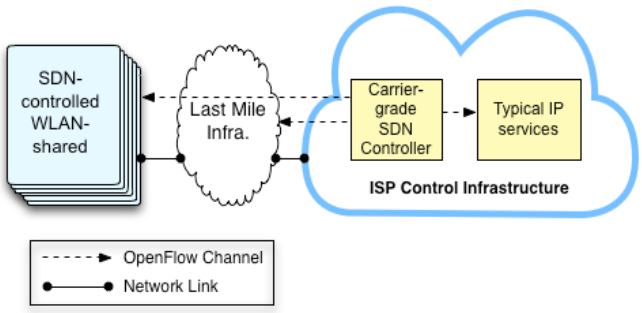


Figure 1. Carrier-grade SDN-Controlled WLAN-shared Ecosystem

The architecture depicted in Figure 1 shows the key systems of our proposed Carrier-Grade SDN-Controlled WLAN-Shared ecosystem. The WLAN-shared system is responsible for the provision of wireless connectivity, generally through Private (customer owner) and Public (community-members) Wi-Fi networks that seek to provide broadband communications to mobile and/or fixed nodes. Traditionally, ISPs include a multifunctional device to manage the connectivity at the WAN link of varying data rates (from 54Mb/s to 300 Mb/s). With regard to our SDN-controlled WLAN-Shared system, on the customer side there is a multifunctional device provided by the ISP, which is responsible for the provisioning of different Wi-Fi networks and “net-programmable” capabilities. Our view of a targeted customer network infrastructure is shown in Figure 2.

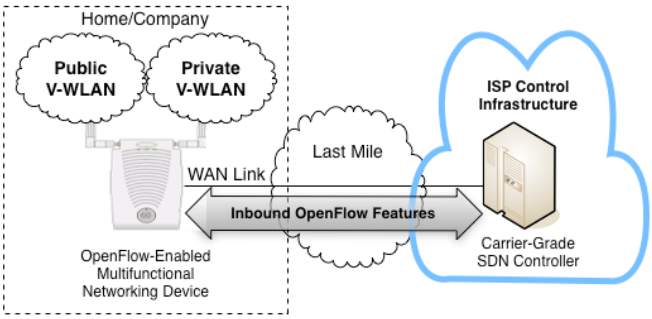


Figure 2. SDN-controlled WLAN-shared system on the customer side, interoperating with the carrier-grade SDN controller

The multifunctional device must support an SDN interface to comply with our SDN-based WLAN-shared ecosystem, in this case, OpenFlow. There is a wide selection of APs available in the market, with prices ranging from tens to even hundreds or thousands of dollars. A number of low-cost commodity-based Wi-Fi router manufacturers already supply proprietary embedded operating systems featuring OpenFlow, mostly version 1.0. Alternatively, an open source environment can be adopted through OpenWrt[4], along with Open VSwitch (which offers support for all versions of the OpenFlow protocol) [Pfaff *et al*. 2015].

The last mile infrastructure mostly comprises broadband telecommunication devices, which are responsible for carrying signals from the broad telecommunication backbone to and from the home or business networks. As robust devices are used in these types of systems (and have high-performance and networking capabilities), SDN is not a problem, and thus their study is beyond the scope of this paper. Figure 3 depicts the ISP-level Control system infrastructure.

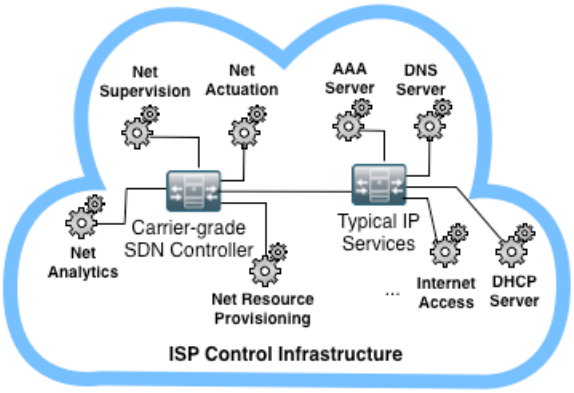


Figure 3. ISP-level control, key infrastructural systems and networking management functions

The ISP system adopts an approach to a fully integrated Internet system approach that allows it to provide a wide range of IP services and applications to a limited amount of customers seeking personal and business access to the Internet. The list of typical IP services includes the following: Internet access, domain name hosting,

[4]https://openwrt.org

network broadband access, Authentication Authorization and Accounting – AAA, web hosting, emailing, transit, among other services. In attempting to cope with SDN, the ISP infrastructure must integrate the SDN controller so that it can act in the network control plane.

We believe that even though the ISP incurs extra costs because it requires investing in additional infrastructure, it will be able to exploit the customer network infrastructure and resell Wi-Fi Internet access. Apart from providing traditional Internet-access for users, the Internet of Everything (IoE) [Evans 2012] application scenario can benefit from this wireless networking opportunity. IoE is turbinated by devices, ranging from connected coffee makers, cars, or sensors on cattle to connected machines in a production plant, and can thus leverage broadband wireless/wired home network systems. Network operators can exploit this scenario to apply opportunistic Internet access, by sharing the bandwidth of WLANs with nearby devices. In addition the network providers can handle the systems to maintain WLAN QoS, for example by using NFV technology [Hawilo et al. 2014, Mijumbi et al. 2016] to virtualize different wireless networks, and allocate resources to each of them by taking account of their needs, contracts, and so on. Thus, network providers can offset the additional costs, and earn more revenue at the same time.

4. Performance Assessment

We carried out a set of experiments to analyze whether it was feasible for ISP to enable the SDN functions to be used with networking nodes that feature commodity-based hardware capabilities, while taking into account the SDN-controlled WLAN-shared scope. To the best of our knowledge, there are no studies in the literature that make a performance evaluation of OpenFlow in a network topology comprising a set of interconnected commodity-based networking devices. However, the literature has authors who include a single device at the center of their targeted assessment. Examples of initiatives taken in this area include [Gharakheili *et al*. 2015], where the authors designed, implemented and evaluated an SDN-controlled system that allows a third-party to define which subscribers can easily customize Internet sharing within their household. The evaluations demonstrated the feasibility and utility of the proposal in the real world, and included home router equipment featuring an OpenFlow protocol. In [Lima *et al*. 2015], the authors evaluated the OpenFlow performance of a commodity-based wireless router running Open vSwitch linked to different SDN controllers. The evaluation was of a set of client hosts connected at the router, with the aim of sending data traffic towards a sink server. The key networking metrics included: throughput, delay, jitter, and packet loss. The outcome suggested that commodity-based wireless routers should be used in smaller networks, such as homes and small -to -medium sized organizations.

Although the dual-access WLAN-shared service is provisioned by a Wi-Fi home AP, there are a number of environments featuring a Wi-Fi network composed of several APs that form a bridge (such as restaurants, hotels, clubs, and the like). This has led us to carry out studies in both suitability and performance, which involves having commodity-based networking devices, interconnected with each other to form a wired mesh topology that is used to deploy OpenFlow networking functions. With this goal in mind, the methodology employed for the performance assessments defines a real testbed

for allowing highly accurate benchmarking perspectives. The network topology of the testbed is depicted in Figure 4.

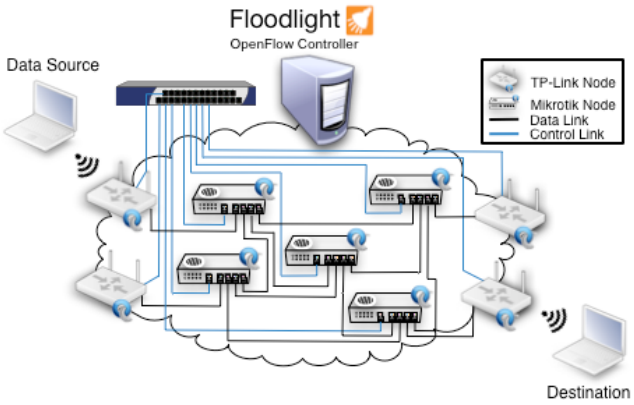


Figure 4. The Testbed Topology adopted in the experiments

The testbed configuration set includes 9 commodity-based networking nodes that are marketed throughout the world at a low cost. The network border embodies 4 TP-Link type nodes set at IEEE 802.11n access points, whilst the core network infrastructure features 5 Mikrotik type nodes that target switching functions. The control data communication takes place through dedicated links. We replaced the original firmware devices with the OpenWrt operating system and installed the Open vSwitch to implement OpenFlow capabilities. The configuration of the network elements is shown in Table I.

Table I. Key Specifications of hardware

Manufacturer	Hardware Capabilities		
	Model	*CPU*	*RAM*
TP-LINK	TL-WR1043ND v3	720 MHz	64 MB
Mikrotik	951G-2HnD	600 MHz	128 MB

The methodology employed for the performance evaluation entailed incorporating a number of individual flow traffic data to obtain the specific load rates (30, 60, 90, and 120% of the maximum network bandwidth capacity). In carrying this out, the D-ITG tool [16] which was running on the client and server machines, generated UDP data flows at a constant rate of 407 Kbps each, which amounted to a total of 293 UDP flows per heavy workload (120%). A specific network application is deployed in compliance with the Floodlight[5] SDN controller. All the new packets arriving at the border switches were directed to the application, which in turn installed a corresponding flow for each of the switches along the path between the client and server machines. Two different sets of experiments are carried out to check the networking behavior at both flow-level and level of video streaming quality.

[5]www.projectfloodlight.org/floodlight

4.1. Flow-Level Networking Set of Experiments

The goal of the experiments was to evaluate how the OpenFlow software layer affects the performance of the devices in terms of packet loss and throughput.

Figure 5 shows a comparison between the packet loss ratios in scenarios where OpenFlow is enabled and disabled. In the case of any offered load, the network experiences a higher packet loss when OpenFlow is running; this is mainly due to the time needed for the controller to configure the switches along the path (the packets are buffered at the ingress switch till the controller installs the flows at the appropriate switches). Figure 6 depicts the corresponding bitrate. In an offered load of 60%, the OF-enable scenario has a performance that is just 5% lower than when OF is disabled. The difference increases to 13% for 90% of the offered load.

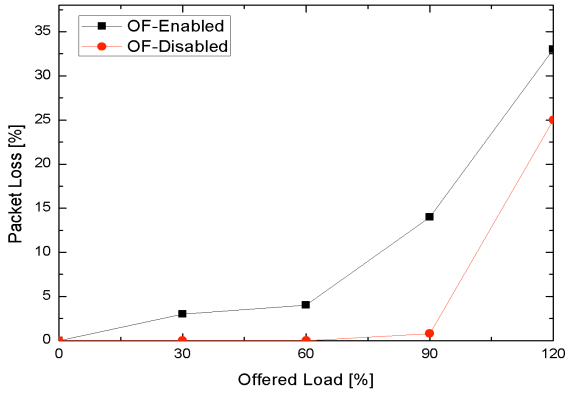


Figure 5. Impact of the variations in the offered load on the behavior of the packet loss

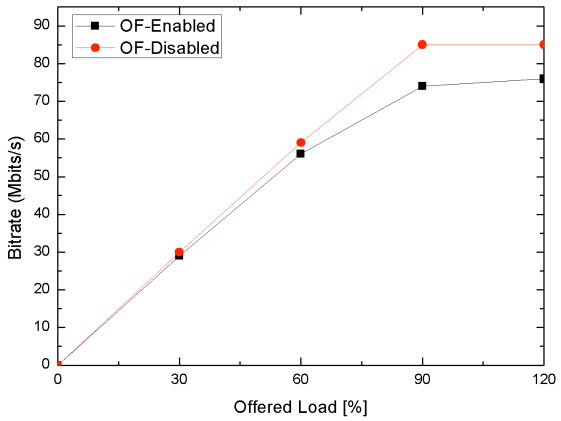


Figure 6. Impact of the variations in the offered load on the throughput behavior

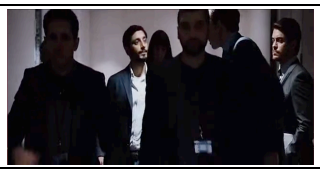
4.2. Video Streaming Quality Set of Experiments

We carried out a second set of experiments in the testbed to assess its performance when a multimedia streaming session is connected along with the D-ITG background data traffic. This set of experiments is run at the testbed where it is configured with 120% of offered load, and seeks to allow subjective benchmarking in heavy saturation networking conditions. The methodology employed involves scaling

the testbed with a multimedia streaming service based on the VLC Media Player[6], in addition to D-ITG data flow traffic model. The “Source Node” of the testbed is set so that it can stream a H.264 encoded real video file using RTP/UDP towards the “Destination” node. No Quality of Service (QoS) control functions are included, since this paper is only concerned with assessing the OpenFlow performance on the testbed.

The video streaming session is invoked by the "Destination" node during the 15 seconds of the running time for the experiment, and keeps running till the end of the experiment (i.e., 33 seconds). With regard to the methodology employed for assessing the impact caused by the over-saturation networking conditions during the video streaming session, we observed degradation of the streaming-level quality caused by an event occurrence during the workflow experiment. Moreover, we calculated the time taken in both testbed configurations (i.e., OF-enabled and OF-disabled) for the VLC client to setup the video streaming session with the VLC server and check the influence of OpenFlow on the additional features of OpenFlow in the setup time of the session. Table II shows three video samples obtained during both sets of experiments.

Table II. Video samples during the workflow experiment

OF-enabled	***OF-disabled***
Video Sample 1.1	Video Sample 2.1
	
Video Sample 1.2	Video Sample 2.2
	
Video Sample 1.3	Video Sample 2.3
	

With regard to the OF-enabled testbed configuration, a grey screen with no viewable video motion (Video Sample 1.1 of Table II) appears after 6 seconds of the request for the VLC client stream session, probably on account of the VLC bufferization. The grey screen remains the same for about 4.2 seconds, and then starts showing a low-quality video streaming (Video Sample 1.2 of Table II) for about 4.3 seconds. From this time on (14.9 seconds), the video starts streaming with good quality (Video Sample 1.3 of Table II) until the end of the workflow experiment. As expected, the over-saturation networking conditions caused degradation occurrences in the video quality (e.g., frame loss), but these were very slight and short-term which meant that on the whole, it was a very good streaming session.

[6]http://videolan.org/vlc

The same screen patterns experienced during the OF-disabled set of workflow experiments also occurred in the OF-disabled one. The grey screen (Video Sample 2.1 of Table II) appears after 5.7 seconds of the VLC client request, and switches to a low-quality video (Video Sample 2.2 of Table II) after 3.9 seconds. After 3.8 seconds (making a total of 13.4 seconds for the streaming setup time), the video streams maintain a good quality till the end of the workflow experiment with slight and short-term quality degradation occurrences for the same reason i.e. networking over-saturation.

The numerical results of the multimedia assessment reveal that there are slight differences in performance in the video streaming on top of OpenFlow at both the OF-enabled and OF-disabled testbed configurations. On the one hand, the session setup time for the OF-enabled set of experiments was 6 seconds, whereas for the OF-disabled set it was 5.7 seconds (5%). With regard to the video streaming session with good quality, the OF-disabled set of experiments took around 19.1 seconds, whilst the OF-enabled set averaged 20.5 seconds. Hence, video streaming on top of the OpenFlow approach and in oversaturated networking conditions displays, slightly higher latency (5% in session setup time, and 7% for VLC buffering when starting the video with good quality) than when running in a conventional system (i.e., without OpenFlow). On the other hand, the video streaming quality remains the same.

5. Conclusion and Suggestions for Future Work

In this study, a general framework was established that was based on SDN and aimed at allowing ISPs to leverage the already existing WLAN infrastructure in urban centers. The purpose of this was to offer tailored connectivity services to mobile nodes through dynamic sharing of these networks between the owner and third parties. The framework includes low-cost commodity hardware, switches and wireless routers. A set of experiments was conducted to show that these devices were suitable for supporting the framework, and the OpenFlow protocol was used to control a network that consists of these devices. Although it has incurred some performance penalties, the SDN approach is still a good choice in view of the benefits it can provide to WLAN sharing-based Internet access.

The next steps of our work will focus on prototyping the carrier-grade SDN-controlled WLAN-shared approach by means of a Wi-Fi router featuring commodity-based hardware capabilities. Afterwards, we will assess the prototype to determine its suitability and performance.

Acknowledgments

This work was partly supported by the SMART (CNPq Universal grant number 457051/2014-0) and Smart Metropolis research projects. The authors would like to thank the Coordination for the Improvement of Personnel in Higher Education (CAPES), the National Council for Scientific and Technological Development (CNPq) and the Mato Grosso State Research Foundation (FAPEMAT). The experiments were carried out at NPITI/UPLab of Instituto Metropole Digital (IMD) and we are grateful to all the personnel at the laboratory for their help.

References

Andrews, J.G., Buzzi, S., Choi, W., Hanly, S.V., Lozano, A., Soong, A.C. and Zhang, J.C. What will 5G be? Selected Areas in Communications, IEEE Journal on, 32(6), 2014

Tschofenig, H., Arkko, J., Thaler, D., and McPherson, D. “Architectural Considerations in Smart Object Networking”. Internet Architecture Board (IAB) Request for Comments (RFC) number 7452, March 2015

Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015–2020 White Paper. Feb. 2016

Biczók, G., Toka, L., Gulyás, A., Trinh, T., and Vidács, A. “Incentivizing the global wireless village”. Computer Networks, Vol. 55, Issue 2, Feb. 2011, pp. 439-456. DOI=http://dx.doi.org/10.1016/j.comnet.2010.08.014

Boucadair, M., Jacquenet, C. “Software-Defined Networking: A Perspective from within a Service Provider Environment”, Internet Engineering Task Force (IETF) Request for Comments (RFC) number 7149, March 2014

Open Networking Foundation, "The OpenFlow Switch Specification, Version 1.4.0", October 2013. [online]. Aveilable at: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf

Carmo, M., de Souza, T., and A. Neto, “On Expanding Carrier-Grade SDN Control-Plane to WLAN customer-side Systems: Performance Evaluation of Commodity-Based Wireless Nodes”, 5th International Workshop on ADVANCEs in ICT INfrastructures and Services (ADVANCE’17), Evry, France, Jan 2017

Gharakheili, H., Exton, L., Sivaraman, V., Matthews, J., and Russell, C. "Third-party customization of residential Internet sharing using SDN," 2015 International Telecommunication Networks and Applications Conference (ITNAC), Sydney, NSW, 2015, pp. 214-219. Doi: 10.1109/ATNAC.2015.7366815

Lima, L., Azevedo, D., and Fernandes, S. "Performance evaluation of OpenFlow in commodity wireless routers," 2015 Latin American Network Operations and Management Symposium (LANOMS), Joao Pessoa, Brazil, 2015, pp. 17-22. Doi: 10.1109/LANOMS.2015.7332664

Pfaff, B., Pettit, J., Koponen, T., Jackson, E. J., Zhou, A., Rajahalme, J., ... & Amidon, K. (2015, May). The Design and Implementation of Open vSwitch. InNSDI (pp. 117-130).

Evans, D. "The internet of everything: How more relevant and valuable connections will change the world." Cisco IBSG (2012): 1-9

Hawilo, H., et al. "NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC)." IEEE Network 28.6 (2014): 18-26

Mijumbi, R., Serrat, J., Gorricho, J. L., Bouten, N., De Turck, F., & Boutaba, R. (2016). Network function virtualization: State-of-the-art and research challenges. IEEE Communications Surveys & Tutorials, 18(1), 236-262.

XXII Workshop de Gerência e Operação de Redes e Serviços (WGRS)

SBRC 2017

Sessão Técnica 2 – Aprovisionamento de Redes e Planejamento de Capacidade

Coleta e Caracterização de um Conjunto de Dados de Tráfego Real de Redes de Acesso em Banda Larga*

Martin Andreoni Lopez[1], Renato Souza Silva[2], Igor Drummond Alvarenga[1], Diogo Menezes Ferrazani Mattos[1] e Otto Carlos Muniz Bandeira Duarte[1]

[1]Grupo de Teleinformática e Automação (GTA)
{martin,alvarenga,menezes,otto}@gta.ufrj.br
[2]Laboratório de Redes de Alta Velocidade (RAVEL)
renato@ravel.ufrj.br
Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro – RJ – Brasil

Resumo. *A segurança do acesso à Internet banda larga reside na implantação de políticas de perímetro e na adoção de listas de controle de acesso. Essas medidas são precárias, pois se baseiam em perfis comuns e pouco atualizados de ameaças aos usuários residenciais. Este artigo analisa e caracteriza o tráfego de usuários residenciais de redes fixas de acesso à Internet em banda larga de uma grande operadora de comunicações, por um período de uma semana, e obtém o perfil dos alarmes gerados por um sistema de detecção de intrusão sobre esse tráfego. Os resultados demonstram que a caracterização proposta permite classificar os fluxos, com uma sensibilidade de* 93% *a alertas, diferenciando os fluxos legítimos dos fluxos que geram alarmes, validando o conjunto de dados coletado, e permite reduzir em* 73% *o tráfego direcionado ao analisador de tráfego, tornando a segurança da rede de acesso mais dinâmica e eficiente.*

Abstract. *Broadband Internet access security lies in the implementation of perimeter policies and in the adoption of access control lists. These measures are precarious because they are based on common and poorly updated profiles, lacking residential users threat information. This article analyzes and characterizes residential user traffic from fixed broadband Internet access networks of a large communications operator, for a period of one week, and obtains the profile of the security alarms generated by an intrusion detection system on this traffic. The results show that the proposed characterization allows classification of the flows, with an alert sensitivity of 93% in the differentiation of the legitimate flows and the alarm generating flows, thus, validating the collected dataset, and allows a 73% reduction for the traffic directed to the traffic analyzer, enabling more dynamic and efficient access network security.*

1. Introdução

O acesso à Internet está presente em 45,5% dos domicílios brasileiros [IBGE 2016]. No entanto, os provedores de infraestrutura de acesso à Internet e os órgãos reguladores encaram a segurança desse serviço de forma precária e com ações paliativas para mitigar possíveis prejuízos. A segurança das redes de acesso à Internet

*Este trabalho foi realizado com recursos do CAPES, CNPq e FAPERJ.

é implementada de forma coletiva, por perímetros de segurança [Bhatt et al. 2014]. Geralmente, não existem medidas individuais por parte dos provedores para garantir a segurança dos elementos das redes residenciais. As restrições instaladas ficam obsoletas conforme o uso da rede evolui, como por exemplo, com a adoção da computação em nuvem e da Internet das coisas que trazem novas ameaças à segurança do usuário residencial [Puthal et al. 2016]. Assim, há a necessidade de analisar sistematizadamente alertas de segurança para, então, atualizar estas restrições de perímetro de segurança da rede.

A definição de novas políticas de segurança depende do conhecimento das ameaças de segurança que estão presentes na rede para garantir maior precisão no combate e na mitigação dos efeitos dos ataques [Heidemann e Papadopoulos 2009]. Uma das maiores dificuldades nas pesquisas relacionadas à análise de tráfego na Internet e à extração de conhecimento de situações de ameaças é a obtenção de bases de dados reais (*datasets*). Apesar de existirem algumas bases de dados disponíveis para pesquisa [CAIDA 2007, D. Kotz e Abyzov 2004, NSL-KDD 2009], essas bases, muitas das vezes, não são atuais ou foram criadas a partir de padrões de ataques e ameaças sintéticas. Outras limitações ao acesso de dados reais recaem sobre questões regulatórias e de sigilo dos dados coletados, já que os dados contem informações sensíveis ou confidenciais.

Este artigo analisa e a caracteriza o tráfego em uma rede de acesso à Internet banda larga, diferenciando o tráfego normal do tráfego que gera alertas de segurança. Emprega-se um conjunto de dados real e anonimizado de uma importante operadora de telecomunicações. O conjunto de dados criado é baseado na captura de 5TB de dados de acesso de 373 usuários residenciais de banda larga na cidade do Rio de Janeiro. O tráfego é tratado e analisado para a criação do conjunto de dados contém tráfego legítimo, ataques e outras ameaças de segurança. O tráfego foi analisado em um sistema de detecção de intrusão (*Intrusion Detection System* - IDS) e, então, resumido em um conjunto de dados de características de fluxos associadas a uma classe de alarme do IDS ou à classe de tráfego legítimo. Por fim, o artigo avalia o desempenho de um classificador de tráfego, baseado no algoritmo de árvore de decisão, para identificar em tempo real os fluxos suspeitos.

Em trabalhos anteriores [Andreoni Lopez et al. 2017, Lobato et al. 2016], foi estudada a adequação de classificadores e da computação por fluxo na identificação de anomalias em rede. Outras propostas criam conjuntos de dados baseados em ataques sintéticos e já desatualizados para o estudo da segurança em redes [NSL-KDD 2009]. Há ainda propostas de criação de potes de mel [Song et al. 2013] e de previsão do comportamento de atacantes baseada em processos estocásticos [Chen et al. 2015]. Este artigo, por sua vez, analisa e caracteriza um conjunto de dados reais constituído de pacotes de dados capturados, no período de uma semana, de usuários residenciais de banda larga fixa de uma importante operadora de telecomunicações. A caracterização provê a diferenciação de dois tipos de tráfegos: um tráfego normal e um tráfego que gera alertas de segurança quando tratados por uma ferramenta de análise tráfego. Os resultados obtidos mostram que, observando somente as estatísticas do fluxo, é possível identificar, com 93% de precisão, fluxos que geram alertas de segurança. A adoção da classificação proposta tem o potencial de reduzir em até 73% do tráfego encaminhado às ferramentas de análise de pacotes, inclusive àquelas que verificam camadas superiores.

O restante do artigo está organizado da seguinte forma. A Seção 2 aborda os trabalhos relacionados. O problema de processamento, caracterização de fluxos, caracte-

rização de alertas e o procedimento de coleta dos dados são apresentados na Seção 3. A Seção 4 explicita o procedimento de análise e apresenta os seus resultados. A Seção 5 conclui o trabalho.

2. Trabalhos Relacionados

Construir um conjunto de dados que permita a análise do perfil de uso fiel da rede é um desafio, pois as redes estão em constantes mudanças e os dados podem ser capturados em diferentes locais da rede. Heidemann e Papadopoulos evidenciam quais são os locais ideais da rede para a captura dos dados, abordam as dificuldades em anonimizar e extrair conhecimento de dados anonimizados e discutem os principais conjuntos de dados disponíveis [Heidemann e Papadopoulos 2009]. Dentre os conjuntos de dados para pesquisa em segurança, o mais utilizado é o KDD [NSL-KDD 2009], cuja caracterização foi realizada por Tavallaee *et al.* [Tavallaee et al. 2009].

Além da localização do ponto de coleta, outros fator importante para evitar a contaminação tendenciosa da base de dados é o tamanho da amostra. Shiravi *et al.* discutem a criação e a análise de uma base de dados real, voltada para a detecção de intrusões [Shiravi et al. 2012]. Os autores também geraram três categorias de bases de dados com tráfego real que se encontram disponíveis para a comunidade acadêmica. A principal desvantagem dessas bases de dados reside no fato de que todo o tráfego anômalo foi coletado a partir de ataques simulados em ambientes controlados. Este artigo, por sua vez, utiliza um IDS para identificar o tráfego malicioso e anômalo. Embora não haja a garantia de completude nessa abordagem, evitam-se os problemas oriundos da inserção de ataques artificiais, como a criação de um conjunto dados muito tendenciosos.

Com o surgimento de novos serviços e diferentes tipos de tráfegos, os ataques se modernizam. Bases de dados antigas como em [NSL-KDD 2009] já não podem ser utilizadas para aferir sistemas de detecção de intrusões mais modernos. A abordagem proposta em [Shiravi et al. 2012] analisa diferentes amostras de serviços TCP (HTTP, SSH, FTP, SMTP, IMAP e POP3) em relação ao número de requisições no tempo e compara as curvas obtidas com distribuições de probabilidade conhecidas. As distribuições similares são então utilizadas para montar perfis de ataques que podem ser posteriormente reproduzidos sinteticamente. O conjunto de dados do presente artigo contempla, entre outros, os mesmos serviços e pode ser utilizado para geração de perfis de ataques atuais.

A utilização de potes de mel (*honeypots*) para a montagem de bases de dados reais é explorada em alguns trabalhos de pesquisa. Chen *et al.* analisam uma base de dados criada a partir de 491 *honeypots* TCP, para estudar as características estocásticas dos ataques [Chen et al. 2015]. Song *et al.* propõem modificar os potes de mel para emular sistemas proativos que, inclusive, visitam páginas maliciosas e se juntam a *botnets* [Song et al. 2013]. Além de aumentar o realismo dos dados coletados, as propostas buscam superar a dificuldade de diferenciação tráfego legítimo e do malicioso, considerando que todo o tráfego coletado nos potes de mel é proveniente de ataques. No entanto não é possível garantir, ou verificar, essa premissa. A utilização de IDS para identificação de tráfego malicioso é utilizada no presente artigo como solução para o mesmo problema. A principal diferença entre os métodos reside na probabilidade de ocorrência de falsos positivos e falsos negativos no conjunto de dados, respectivamente prevalentes em [Song et al. 2013, Chen et al. 2015], e no presente trabalho.

Drapper-Gil *et al.* apresentam a caracterização do tráfego de redes privadas virtuais baseada em características temporais dos fluxos [Draper-Gil et al. 2016]. Para tanto, os autores propõem o uso de 8 características para a classificação dos fluxos em 14 diferentes tipos, incluindo fluxos de VPN e não VPN. Os resultados mostram que o algoritmo de aprendizado de máquina por árvores de decisão apresenta um desempenho um pouco melhor do que o algoritmo de k-vizinhos mais próximos. No entanto, os autores não avaliam se as características usadas para a classificação são as que melhor descrevem o conjunto de dados, nem avaliam o perfil de uso da rede. A redução de dimensionalidade do conjunto de dados pode ser realizadas através de métodos de seleção de características, selecionando as que melhor descrevem os dados [Andreoni Lopez et al. 2017], ou através de técnicas que levam os dados a outro espaço vetorial [Pascoal et al. 2012]. O presente artigo utiliza as técnicas descritas nesses trabalhos para realizar uma avaliação de relevância de atributos em relação à identificação de anomalias, resultando também em um conjunto de oito características principais.

Kato *et al.* propõem o uso de aprendizagem profunda (*deep learning*) para realizar a caracterização do tráfego da rede, já que advogam que a aprendizagem profunda é capaz de extrair padrões mais complexos do que outras técnicas [Kato et al. 2017]. Por sua vez, Nie *et al.* usam uma rede bayesiana para detectar anomalias e modelam a matriz de tráfego da rede [Nie et al. 2016]. Os resultados mostram que a previsão de tráfego é acurada, mas concentram-se no ambiente de computação em nuvem e não visam a predição de tráfego de usuários finais. O presente artigo oferece um conjunto de dados complementar ao utilizado por Kato *et al.*, uma vez que é composto somente por tráfego de usuários finais.

O conjunto de dados deste artigo é real e corresponde à captura, durante uma semana entre os dias 24 de fevereiro e 4 março de 2017, de pacotes de acesso à Internet de 373 usuários de banda larga fixa de uma importante operadora de telecomunicações na Zona Sul da cidade do Rio de Janeiro. São observadas as recomendações sobre a localização dos pontos de captura em [Heidemann e Papadopoulos 2009], bem como adotada como base a metodologia de caracterização de [Tavallaee et al. 2009]. Os alarmes de segurança são identificados pela análise dos dados através de uma ferramenta de detecção de intrusão, em contrapartida às abordagens sintéticas [Shiravi et al. 2012] e baseadas em potes de mel [Song et al. 2013, Chen et al. 2015]. Dessa forma, pretende-se possibilitar o estudo de métodos de classificação em dados atualizados, de forma a complementar e atualizar estudos como [Shiravi et al. 2012, Kato et al. 2017].

3. A Análise do Tráfego e o Conjunto de Dados de Acesso de Usuários Residenciais à Internet

A caracterização do perfil dos alertas em uma rede de acesso exige o monitoramento, o processamento e o gerenciamento de grandes volumes de dados gerados em tempo real. Esse grande volume de dados é processado por sistemas de detecção de intrusão (*Intrusion Detection System* - IDS) e, também, pela correlação com as informações de fluxos na rede [Wu et al. 2014, Lobato et al. 2016]. Ferramentas conhecidas como Gerenciamento e Correlação de Eventos de Segurança (*Security Information and Event Management*- SIEM) realizam este tipo de tarefa a um custo econômico elevado e ainda assim podem gerar atrasos. Em média, a reação a ameaças de segurança é tomada após 123 horas da ocorrência e, no caso de detecção do vazamento de informações, a demora na identificação dessa falha de segurança chega a 206 dias [Clay 2015]. Ao conhecer o

perfil mais comum dos alertas de segurança, é possível identificar mais rapidamente o vazamento de informações e as vulnerabilidades exploradas.

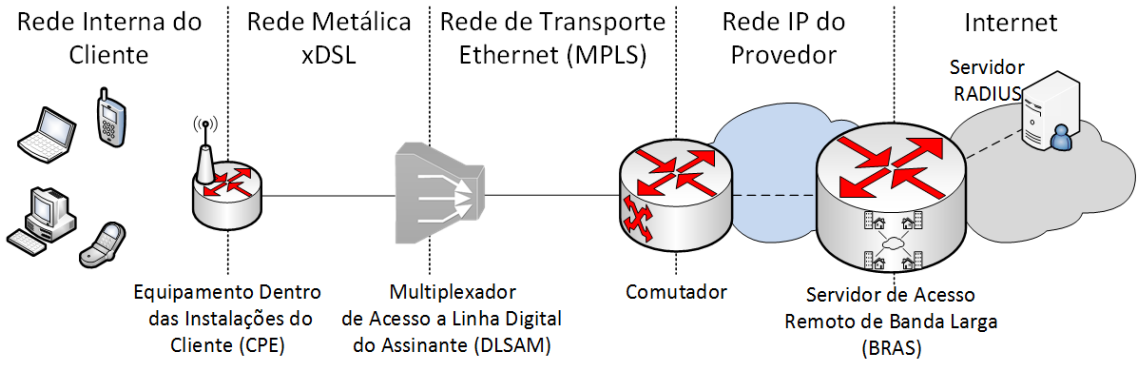


Figura 1. Topologia típica da rede de acesso banda larga. A conexão entre o *Home Gateway* e a Internet é autenticada, contabilizada e registrada pelo servidor *Radius*. O tráfego é encapsulado em sessões PPPoE (*Point-to-Point Protocol over Ethernet*) entre a casa do usuário e o BRAS (*Broadband Remote Access Server*). A inspeção e a coleta do tráfego ocorre após o BRAS.

A Figura 1 mostra uma topologia de acesso típica para o serviço de banda larga composta por um equipamento dentro das instalações do usuário, *home gateway* ou CPE (*Customer Premises Equipment*), ligado a um multiplexador de acesso (*Digital Subscriber Line Asymmetric Multiplexer* - DSLAM), uma rede de transporte, como por exemplo uma rede MPLS (*Multiprotocol Label Switching*), e um agregador de seções (*Broadband Remote Access Server* - BRAS) que autentica a sessão dos usuários através de um servidor RADIUS, responsável também pela auditoria de uso da rede. Assim, em uma rede de acesso para usuários de banda larga fixa, a inspeção é realizada somente após a agregação do tráfego, já que não há nós que permitam a inspeção dos dados nas premissas do usuário ou no perímetro mais próximo dos usuários.

O tráfego a ser analisado é composto pelo tráfego agregado proveniente da alta capilaridade de diferentes usuários, com uma grande variedade de perfis de serviços acessados por cada usuário e gerando um grande volume de dados. Portanto, o problema de caracterização do perfil de alertas consiste um problema complexo de análise de grandes massas de dados (*big data*) [Costa et al. 2012], que requer ferramentas de processamento apropriadas. A ideia central deste artigo é gerar, analisar e caracterizar um conjunto de dados que represente o mais fielmente possível o perfil de uso dos usuários de banda larga fixa residencial com a finalidade de treinar classificadores de tráfego. Como foi mencionado anteriormente, o conjunto de dados corresponde ao tráfego de acesso de 373 usuários de banda larga fixa de uma grande operadora de telecomunicações na Zona Sul da cidade do Rio de Janeiro. Os dados foram coletados nas premissas da operadora e foram anonimizados para garantir o sigilo e a privacidade dos usuários. As análises realizadas descartam verificações do conteúdo dos pacotes. A base de dados analisada foi criada a partir da captura de pacotes brutos, contendo informações reais de tráfego IP (*Internet Protocol*) dos usuários residenciais. O tráfego foi coletado e gravado de forma ininterrupta por uma semana através do *software* `tcpdump`[1]. O processo de coleta e gravação

[1]Disponível em http://www.tcpdump.org.

dos arquivos não utilizou quaisquer filtros de pacotes e, portanto, todos os pacotes da rede foram gravados na sua forma bruta (*raw data*) diretamente na base de dados. A estrutura física de coleta foi configurada espelhando o tráfego agregado de um DSLAM em uma outra porta do comutador *metro-ethernet* da rede de transporte. O espelhamento da porta do DSLAM no comutador permite que todo o tráfego originado ou destinado para o DSLAM seja clonado para um computador executando o sistema operacional Linux Ubuntu, o qual foi conectado a esta segunda porta para coletar e gravar em formato *pcap* todos os pacotes. Para garantir o armazenamento em alta velocidade e para permitir o fácil transporte dos dados, a base de dados foi gravada em um disco rígido externo com interface USB 3.0. A Figura 2 mostra a topologia básica e a estrutura montada para a coleta dos dados. Vale ressaltar que, embora a rede da operadora considerada seja muito maior que a amostragem tomada neste trabalho, os dados coletados representam o consumo real de usuários residenciais e não é o escopo do trabalho analisar todo o tráfego da operadora.

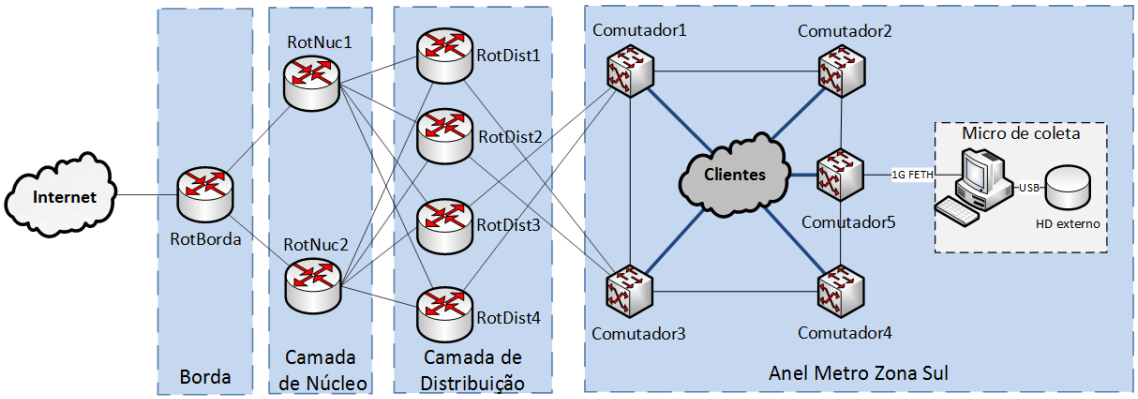


Figura 2. Topologia da estrutura de coleta de dados da porta principal do DSLAM com 373 clientes de banda larga.

O procedimento usado na captura dos dados garantiu não haver perdas de pacotes no espelhamento de portas a 1Gb/s montado para coleta e gravação dos pacotes. Assim, 100% do tráfego gerado pelos 373 clientes foi coletado e gravado na base de dados, totalizando 5TB de informações. Apesar da velocidade média disponível em cada porta do DSLAM ser de aproximadamente 12Mb/s, gerando um tráfego agregado hipotético superior a 4Gb/s, foi verificado que durante todo o processo de captura, o tráfego real agregado não superou a taxa de 800Mb/s. O tráfego agregado conta com tráfegos de ida e volta (*uplink* e *downlink*). Vale notar que os dados capturados são somente de usuários residenciais, portanto todo tráfego é proveniente de sessões banda larga fixa.

4. A Análise dos Dados

A análise dos dados capturados da rede da operadora de telecomunicações foi dividida em três etapas. A primeira etapa trata os arquivos de captura de dados brutos através de um sistema de detecção de intrusão (*Intrusion Detection System* - IDS) de rede e, posteriormente, gera um resumo dos dados na forma de fluxos. A segunda etapa analisa a distribuição das principais características do conjunto de dados, evidenciando diferenças entre o tráfego normal e o tráfego que gera alerta. Por fim, a terceira etapa consiste em comparar o uso de classificadores para verificar a acurácia do classificador em separar tráfego normal do tráfego gera que alertas. Ao realizar a classificação é possível

direcionar somente a parcela do tráfego que pode gerar alerta dentre todo o tráfego da operadora de telecomunicações para o IDS, diminuindo a carga de tráfego analisada.

A primeira etapa de análise dos dados baseou-se na extração das características dos fluxos representados pelos pacotes capturados, assim como na verificação de possíveis alertas através do IDS. Por se tratar de um tráfego de clientes residenciais com acesso ADSL (*Asymmetric Digital Subscriber Line*), o tráfego capturado é encapsulado em sessões PPPoE (*Point-to-Point Protocol over Ethernet*), o que dificulta a análise dos pacotes, já que alguns IDS não realizam a inspeção desse tipo de tráfego, como, por exemplo, o SNORT [Roesch et al. 1999]. Portanto, a fim de executar a classificação do tráfego em diferentes tipos de alertas, foi usado o IDS Suricata[2], versão 3.2, com a sua base de assinaturas atualizada. Vale ressaltar que a classificação entre tráfego normal e alerta foi realizada somente com base nas assinaturas do Suricata, pois não havia conhecimento prévio sobre a origem dos dados coletados. Como os dados são reais, não é possível assegurar que todos os fluxos são legítimos ou, mesmo após a classificação do IDS, que todos os fluxos de alerta são de fato maliciosos. No entanto, a classificação gerada pelo IDS é utilizada como referência e considerada como definitiva e correta no contexto do artigo.

Paralelamente à classificação dos pacotes pelo IDS, os pacotes capturados foram desencapsulados da sessão PPPoE, usando a ferramenta `stripe`[3], e foram resumidos em fluxos, através da ferramenta `flowtbag`[4]. Ademais, foi desenvolvida uma aplicação Python que processa a saída do IDS, o relatório de fluxos que geram alertas, e correlaciona os alertas com a informação de fluxos resumida. Assim, foi possível obter um conjunto de dados de fluxos com a marcação da classe a qual pertencem. O conjunto de dados apresenta 42 características de cada fluxo e mais a classe a que pertence cada fluxo. A classe de saída, característica 43, é dada pelo tipo de alerta gerado pelo IDS, no caso de um fluxo que dispara um alerta, ou o fluxo é marcado com a classe 0 indicando que é um fluxo normal. No conjunto de dados não se adicionam os endereços IP de origem e de destino dos fluxos para garantir a *"anonimização"* dos dados.

A segunda etapa consiste em extrair conhecimento dos dados. Para tanto, foi utilizada a plataforma de análise de dados gratuita e de código aberto KNIME[5], versão 3.3.1. Em um primeiro momento, a análise dos dados foca na Análise das Componentes Principais (*Principal Component Analysis* - PCA) com o intuito de verificar quais são as características do conjunto de dados que carregam mais informação. Assim, a Figura 3 mostra as seis componentes principais do conjunto de dados. Foram escolhidas seis componentes principais, pois são aquelas em que o autovalor absoluto é muito maior que 0, o que garante a preservação de 99% da informação do conjunto de dados. As componentes foram ordenadas em relação ao autovalor associado a cada autovetor que define a componente. Tendo em vista as componentes principais, destaca-se que as características mais relevantes para a caracterização do tráfego são: porta de origem, porta de destino, volume total do fluxo, quantidade de pacotes no fluxo, volume dos subfluxos de ida e de volta e volume de dados em cabeçalhos nos fluxos de ida e de volta. A partir dessas características, analisou-se o comportamento do tráfego normal e dos alertas.

[2]Disponível em https://suricata-ids.org.

[3]Disponível em https://github.com/theclam/stripe.

[4]Disponível em https://github.com/DanielArndt/flowtbag.

[5]Disponível em https://www.knime.org/

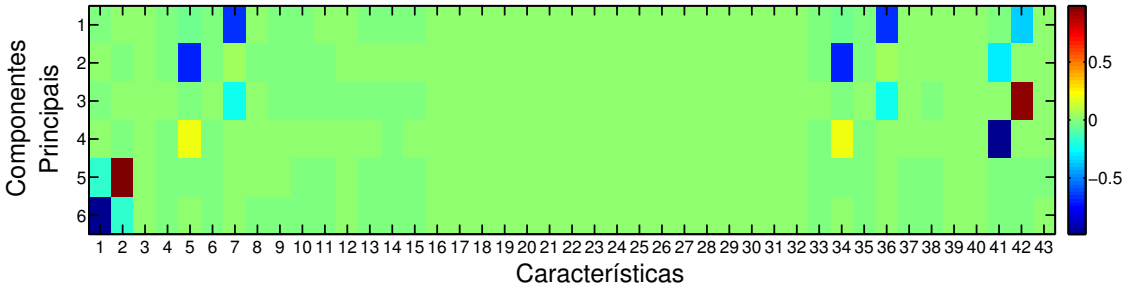
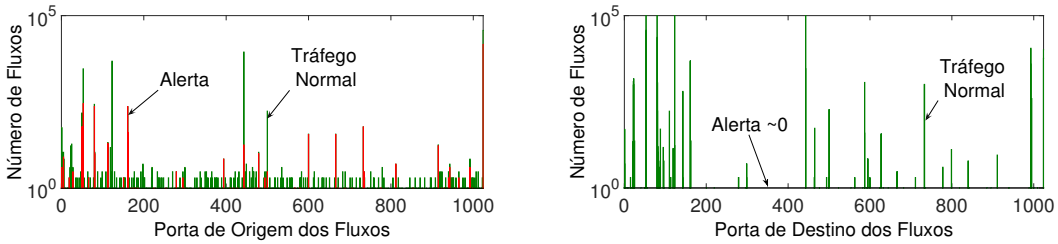


Figura 3. Visualização das seis componentes obtidas com a Análise de Componentes Principais (PCA) sobre o conjunto de dados, em ordem decrescente de autovalor. As oito características que mais se destacam nas componentes são: porta de origem (1), porta de destino (2), volume total do fluxo (5), quantidade de pacotes no fluxo (7), volumes dos subfluxos de ida (34) e de volta (36), volume de dados em cabeçalhos nos fluxos de ida (41) e de volta (42).

A primeira característica analisada foi a porta em que o tráfego ocorre. A Figura 4 apresenta as portas de origem e destino dos fluxos. A figura concentra-se sobre as 1024 primeiras portas (de 0 a 1023), pois são as portas restritas. Usualmente, essas portas são usadas por *daemons* que executam serviços com privilégios de administrador do sistema. A definição de fluxo usada considera como porta de origem a porta que inicia a conexão TCP. Como o conjunto de dados retrata usuários residenciais, é esperado que a maior parte das conexões seja destinada a portas restritas e não originadas dessas. Assim, verifica-se que o número de alertas provenientes de conexões que usam as portas restritas como destino é baixo em relação ao total de conexões nessas portas, Figura 4(b). Contudo, ao se considerar os fluxos que usam as portas restritas como porta de origem, quase a totalidade dos fluxos é marcada como alerta pelo IDS, mostrado na Figura 4(a). Outro fato marcante é que se observa que grande parte dos fluxos analisados refletem o uso do serviço de DNS (UDP 53) e os serviços HTTPS e HTTP (TCP 443 e 80). O predomínio do uso de serviços HTTPS sobre os HTTP reflete a mudança de que os principais provedores de conteúdo da Internet, tais como Google e Facebook, têm passado a usar o serviço criptografado por padrão para garantir a segurança e privacidade dos usuários.

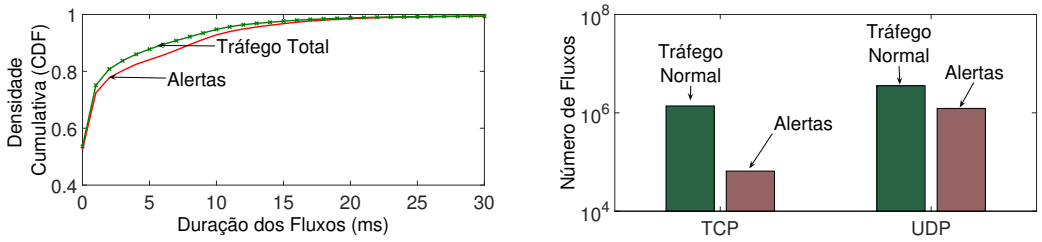


(a) Distribuição das portas de origem dos fluxos. (b) Distribuição das portas de destino dos fluxos.

Figura 4. Portas usadas nos fluxos. Comparação do uso das 1024 portas mais baixas (portas restritas) nos fluxos avaliados. Por se tratarem de usuários domésticos, o maior número de fluxos com origem nessas portas são fluxos que geram alertas.

Os resultados dos serviços mais comumente acessados na rede tornam-se mais claros ao serem comparados com a duração e os protocolos usados nos fluxos, mostrados

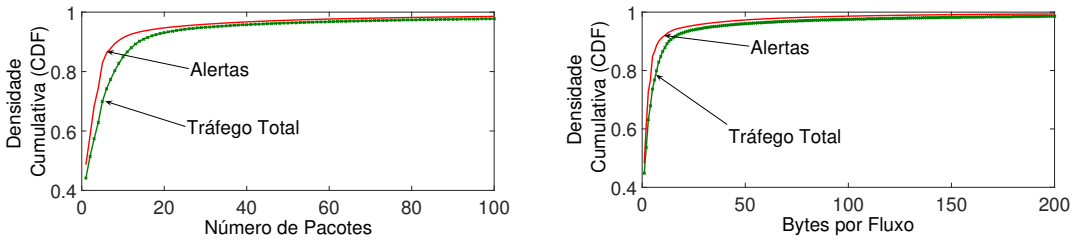
nas Figuras 5(a) e 5(b). A duração dos fluxos analisados majoritariamente é menor que 30 ms, caracterizando o uso dos serviços DNS, HTTP e HTTPS. Uma boa aproximação para duração média dos fluxos é a distribuição Erlang, com $k = 1$ e $\lambda = 3.7$. A aproximação foi calculada através da adequação da distribuição aos dados e através da minimização do erro quadrático médio entre a distribuição e os dados obtidos. Contudo, pelo teste hipótese de Kolmogorov-Smirnov[6], a hipótese de que os dados seguem tal distribuição deve ser rejeitada. Em relação aos protocolos usados, é evidente o predomínio de fluxos UDP, referentes a consultas DNS. Vale ressaltar que o número de alertas gerados por fluxos UDP é mais de 10 vezes superior ao número de alertas gerados por fluxos TCP. Outro ponto importante é que o número de fluxos que geram alertas é de aproximadamente 26% dos fluxos totais.



(a) Duração dos Fluxos. (b) Protocolos de transporte usados.

Figura 5. Função de Densidade de Probabilidades Cumulativa (CDF) para a distribuição da duração dos fluxos em *ms* e número de fluxos por protocolos de transporte. a) Os fluxos que geram alertas tendem a ser de menor duração que os fluxos totais. b) Os fluxos legítimos com UDP são numerosos em função do DNS (porta 53 UDP). O número de alertas em UDP é mais de 10 vezes maior que em fluxos TCP.

A Figura 6 mostra a caracterização do número de pacotes por fluxo, em ambos os sentidos da comunicação, ida e volta. Em ambos os sentidos, a comunicação ocorre com até 32 pacotes em 95% dos casos e com até 100 pacotes em 98, 5%. O resultado mostra que as conexões no cenário residencial são em sua grande maioria conexões com poucos pacotes. Nota-se ainda que os fluxos que geram alertas têm, em geral, menos pacotes do que fluxos do tráfego normal, pois 95% dos fluxos de alerta apresentam até 22 pacotes.

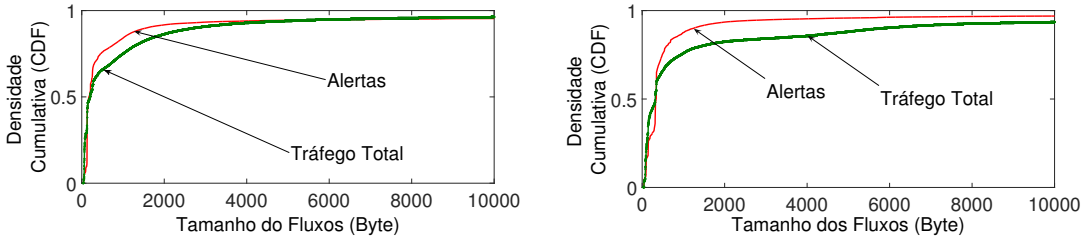


(a) Fluxos no sentido de ida. (b) Fluxos no sentido de volta.

Figura 6. Função de Densidade de Probabilidades Cumulativa (CDF) para a distribuição do número de pacotes por fluxo. Fluxos que geram alertas tendem a ter menos pacotes.

[6]O teste de Kolmogorov-Smirnov verifica se uma das distribuições de probabilidade difere da distribuição em hipótese com base em um número finito de amostras.

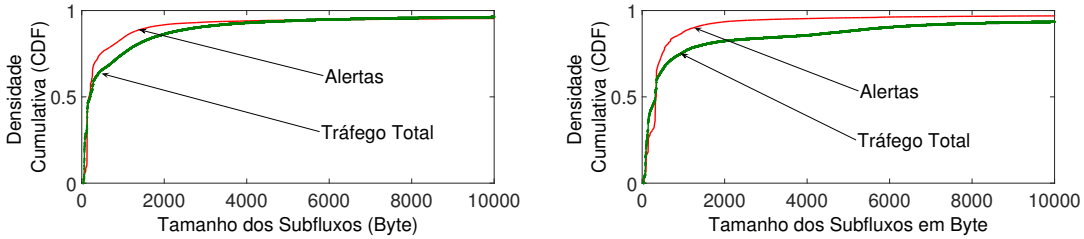
Considerando-se a quantidade de dados trafegada em cada fluxo, a Figura 7 compara os fluxos de ida e volta em relação ao volume em *bytes* trafegados. É visível a disparidade do volume de tráfego nos dois sentidos da comunicação. Enquanto, no sentido de ida, 95% do tráfego apresenta no máximo o volume de $6,4$ kB, no sentido de volta, a mesma parcela de tráfego apresenta até 18 kB. A distribuição que melhor se adequa ao volume de dados, verificando-se a que minimiza o erro quadrático médio, é distribuição lognormal, com $\mu = 5.67$ e $\sigma = 31.68$. A validação da adequação à distribuição lognormal foi realizada através do teste de Kolmogorov-Smirnov sobre uma subamostragem aleatória dos dados para uma significância estatística de 95%. Esse resultado demonstra que o perfil do usuário de banda larga residencial é o de um consumidor de conteúdo. Outro ponto interessante é que os fluxos que geram alertas têm um perfil de volume de tráfego semelhante nos sentidos de ida e de volta. Tráfegos assimétricos são mais característicos do usuário classificado como legítimo.



(a) Fluxos no sentido de ida. (b) Fluxos no sentido de volta.

Figura 7. Função de Densidade de Probabilidades Cumulativa (CDF) para a distribuição do volume em *bytes* por fluxo. Fluxos que geram alertas tendem a ter menor volume em *bytes* trafegados.

Já a Figura 8 mostra o comportamento dos subfluxos gerados em cada conexão. Tal característica foi apontada pelo método PCA como uma das características mais importantes para se descrever o conjunto de dados. No entanto, o comportamento estatístico do volume de dados dos subfluxos é o mesmo do fluxo total. Isso ocorre, pois os fluxos são majoritariamente de curta duração, evidenciado na Figura 5(a), e assim não geram subfluxos. A análise dos dados mostrou que os fluxos não passam ao estado *idle*.

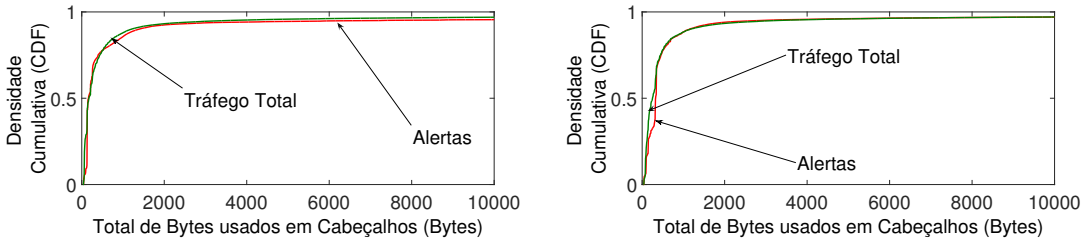


(a) Fluxos no sentido de ida. (b) Fluxos no sentido de volta.

Figura 8. Função de Densidade de Probabilidades Cumulativa (CDF) para a distribuição do volume em *bytes* por subfluxo em cada fluxo. Fluxos que geram alertas tendem a ter menor volume em *bytes* trafegados em subfluxos.

Outra característica importante é a quantidade total de dados trafegada nos cabeçalhos dos pacotes. A Figura 9 evidencia que, em ambos os sentidos dos fluxos, tanto o tráfego marcado como alerta quanto o total apresentam o mesmo comportamento. Em

especial, percebe-se uma simetria no tráfego de ida e de volta quanto ao volume de dados nos cabeçalhos.



(a) Fluxos no sentido de ida. (b) Fluxos no sentido de volta.

Figura 9. Função de Densidade de Probabilidades Cumulativa (CDF) para a distribuição do volume em *bytes* dos dados trafegados em cabeçalhos. O comportamento do tráfego que gera alertas é muito semelhante ao tráfego total.

Ao fim da etapa de extração de conhecimento dos dados, foi analisado o perfil dos alertas gerados pelo IDS. A Figura 10 mostra quais são as principais classes de alertas disparados pelo IDS. Destacam-se primeiramente os alertas de ataques ao HTTP. Nessa classe de alertas enquadram-se os ataques de injeção de SQL através de chamadas HTTP e ataques XSS (*cross-site scripting*). Tais ataques são plausíveis de serem executados por usuários residenciais, pois usam os parâmetros das chamadas HTTP para inserir algum código malicioso nos servidores e, portanto, não são barrados por regras de acesso. Outros alertas importantes são os de escaneamento de portas e vulnerabilidade (`scan`) e os de execução de aplicativos maliciosos (`trojan` e `malware`). Os escaneamentos visam, no geral, identificar portas abertas e vulnerabilidades nas premissas do usuário (*gateway* doméstico). Os alertas referentes a `trojan` e `malware` identificam atividades características de aplicativos maliciosos conhecidos que visam criar e explorar vulnerabilidades nos dispositivos dos usuários residenciais. Os demais alertas são referentes a mecanismos de roubos de informação e, também, a assinaturas de ataques bizantinos em protocolos comuns, como IMAP e Telnet[7].

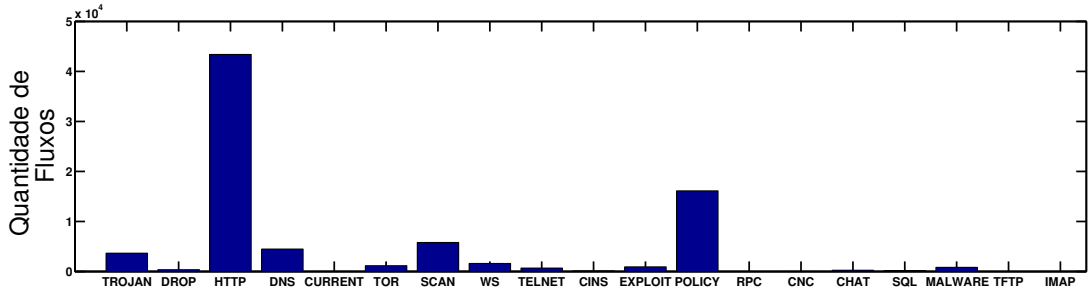


Figura 10. Distribuição dos principais tipos de alertas no tráfego analisado.

A terceira etapa da análise dos dados consiste em projetar um classificador para separar os fluxos em normais ou naqueles que podem gerar alarmes. O objetivo é realizar a classificação dos fluxos com uma precisão considerável para otimizar a análise de trafego realizado pela operadora de telecomunicações, pois somente o tráfego classificado como

[7]Principalmente usado para a configuração remota de equipamentos de rede.

suspeito deverá ser desviado para ser tratado pelo serviço de IDS. Para tanto, o foco do artigo é realizar a classificação supervisionada, em que o treinamento do classificador é feito com um conjunto de dados marcados com uma classe de saída. As métricas de avaliação do classificador verificam o quanto o classificador proposto se aproxima da classe de saída pré-determinada para cada fluxo. No caso deste artigo, a classe de saída usada no treinamento e avaliação do classificador é o resultado da análise pelo IDS.

A primeira abordagem de classificador considera uma rede neural com duas camadas ocultas com 20 neurônios em cada, com seis neurônios na camada de entrada e um neurônio na camada de saída. O algoritmo de aprendizado usado foi o *Multilayer Perceptron* (MLP) com o ajuste de pesos através do algoritmo de *backward propagation*. A modelagem da rede neural leva em consideração que a entrada dos dados é formada pelas seis componentes principais calculadas pelo PCA. A saída da rede é a probabilidade de o fluxo ser marcado como suspeito. Considera-se fluxo suspeito todo aquele que tiver saída maior que 0,5 e fluxo normal aqueles cujo valor final esteja abaixo desse limiar. A Tabela 1 mostra o desempenho da rede neural em uma avaliação cruzada em 10 rodadas[8]. Verifica-se que a acurácia da rede neural foi de $0,847$, com sensibilidade de $0,625$ na classe de alerta.

Tabela 1. Classificação com Rede Neural com 2 camadas ocultas.

	VP	**FP**	**VN**	**FN**	**Precisão**	**Sens.**	**Espec.**
Alerta	809626	396162	3234673	485174	0.671	0.625	0.891
Normal	3234673	485174	809626	396162	0.870	0.891	0.625

Tabela 2. Classificação com Árvore de Decisão.

	VP	**FP**	**VN**	**FN**	**Precisão**	**Sens.**	**Espec.**
Alerta	1209238	87499	3543336	85562	0.933	0.934	0.976
Normal	3543336	85562	1209238	87499	0.976	0.976	0.934

Em uma segunda abordagem, avaliou-se o uso do classificador baseado em Árvore de Decisão. A entrada do classificador foram as seis componentes principais extraídas do PCA. A saída do classificador é a marcação em uma das duas classes possíveis, alerta ou normal. A Tabela 2 mostra o resultado da classificação usando-se a Árvore de Decisão em uma avaliação cruzada em 10 rodadas. A acurácia atingida pelo classificador foi de $0,956$. A sensibilidade na classe de alertas foi de $0,934$ e de $0,976$ na classe normal. Esse resultado mostra que o uso desse classificador como um pré-tratamento dos fluxos reduz em até 73% a carga no analisador de tráfego da operadora de telecomunicações, com uma sensibilidade de $0,934$ no tráfego suspeito.

5. Conclusão

O conhecimento do perfil de uso da rede é importante para melhor dimensionar a rede e identificar os principais serviços utilizados. A identificação dos principais alertas de segurança na rede, por sua vez, permite conhecer quais são as principais ameaças e projetar possíveis contramedidas. Esse artigo apresentou a criação de um conjunto de

[8]VP: Verdadeiro Positivo; FP: Falso Positivo; VN: Verdadeiro Negativo; FN: Falso Negativo; Sens.: Sensibilidade; Espec: Especificidade.

dados de alertas de segurança em uma rede real de uma operadora de telecomunicações na cidade do Rio de Janeiro[9]. O conjunto de dados representa o uso do serviço de acesso banda larga fixa de 373 usuários residenciais. A análise dos dados permite identificar que os principais serviços acessados são os de DNS e serviços *web*. O perfil dos fluxos é caracterizado por conexões rápidas, de até 30 ms, com a transferência de até 18 kB em 95% dos casos. A fim de validar o conjunto de dados coletado quando à acuidade da marcação de alertas por fluxo, foram empregados métodos de aprendizado de máquina para classificação destes fluxos em uso normal e alertas. Os resultados obtidos através da aplicação de rede neural e árvore de decisão demonstram que a marcação aplicada ao conjunto de dados é consistente com padrões observáveis. Com base na caracterização do uso normal e dos alertas, esse artigo propôs o uso de um classificador baseado em árvore de decisão para identificar os fluxos suspeitos e reduzir a carga no analisador de tráfego. Os resultados mostram que o uso de um classificador simples é capaz de reduzir em até 73% o tráfego enviado ao analisador de tráfego, com a capacidade de identificar até 93% dos fluxos que geram alertas na rede.

Como trabalhos futuros, pretende-se utilizar este conjunto de dados na validação de uma arquitetura de processamento por fluxos para a classificação por árvores de decisão, verificando a precisão da análise de tráfego em tempo real. Ademais, será avaliado o desempenho de outros algoritmos de classificação e de treinamento em tempo real.

Referências

Andreoni Lopez, M., Lobato, A., Mattos, D. M. F., Alvarenga, I. D., Duarte, O. C. M. B. e Pujolle, G. (2017). Um algoritmo não supervisionado e rápido para seleção de características em classificação de tráfego. Em *XXXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC'2017 - a ser apresentado)*.

Bhatt, S., Manadhata, P. K. e Zomlot, L. (2014). The operational role of security information and event management systems. *IEEE Security Privacy*, 12(5):35–41.

CAIDA (2007). Supporting research and development of security technologies through network and security data collection. http://www.caida.org/funding/predict/. Acessado 28-03-2017.

Chen, Y.-Z., Huang, Z.-G., Xu, S. e Lai, Y.-C. (2015). Spatiotemporal patterns and predictability of cyberattacks. *PLOS ONE*, 10(5):1–19.

Clay, P. (2015). A modern threat response framework. *Network Security*, 2015(4):5–10.

Costa, L. H. M. K., de Amorim, M. D., Campista, M. E. M., Rubinstein, M. G. e Duarte, O. C. M. B. (2012). Grandes massas de dados na nuvem: Desafios e técnicas para inovação. Em *Minicursos do XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC'2012)*.

D. Kotz, T. H. e Abyzov, I. (2004). CRAWDAD: A community resource for archiving wireless data at dartmouth. http://crawdad.cs.dartmouth.edu/. Acessado 28-03-2017.

Draper-Gil, G., Lashkari, A. H., Mamun, M. S. I. e Ghorbani, A. A. (2016). Characterization of encrypted and vpn traffic using time-related features. Em *Proceedings of the*

[9]Os dados anonimizados podem ser consultados através de contato com os autores por e-mail.

2nd International Conference on Information Systems Security and Privacy, páginas 407–414.

Heidemann, J. e Papadopoulos, C. (2009). Uses and challenges for network datasets. Em *Proceedings of the IEEE Cybersecurity Applications and Technologies Conference for Homeland Security (CATCH)*, páginas 73–82, Washington, DC, USA. IEEE.

IBGE (2016). *Síntese de indicadores sociais : uma análise das condições de vida da população brasileira*, volume 36. IBGE, Rio de Janeiro.

Kato, N., Fadlullah, Z. M., Mao, B., Tang, F., Akashi, O., Inoue, T. e Mizutani, K. (2017). The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective. *IEEE Wireless Communications*, PP(99):2–9.

Lobato, A., Andreoni Lopez, M. e Duarte, O. C. M. B. (2016). Um sistema acurado de detecção de ameaças em tempo real por processamento de fluxos. Em *XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC'2016)*, Salvador, Bahia.

Nie, L., Jiang, D. e Lv, Z. (2016). Modeling network traffic for traffic matrix estimation and anomaly detection based on bayesian network in cloud computing networks. *Annals of Telecommunications*, páginas 1–9.

NSL-KDD (2009). Nsl-kdd data set for network-based intrusion detection systems. http://iscx.cs.unb.ca/NSL-KDD/. Acessado 28-03-2017.

Pascoal, C., de Oliveira, M. R., Valadas, R., Filzmoser, P., Salvador, P. e Pacheco, A. (2012). Robust feature selection and robust PCA for Internet traffic anomaly detection. Em *2012 Proceedings IEEE INFOCOM*, páginas 1755–1763.

Puthal, D., Nepal, S., Ranjan, R. e Chen, J. (2016). Threats to networking cloud and edge datacenters in the internet of things. *IEEE Cloud Computing*, 3(3):64–71.

Roesch, M. et al. (1999). SNORT: Lightweight intrusion detection for networks. Em *Lisa*, volume 99, páginas 229–238.

Shiravi, A., Shiravi, H., Tavallaee, M. e Ghorbani, A. A. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Comput. Secur.*, 31(3):357–374.

Song, J., Takakura, H., Okabe, Y. e Nakao, K. (2013). Toward a more practical unsupervised anomaly detection system. *Inf. Sci.*, 231:4–14.

Tavallaee, M., Bagheri, E., Lu, W. e Ghorbani, A. A. (2009). A detailed analysis of the kdd cup 99 data set. Em *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, páginas 1–6.

Wu, K., Zhang, K., Fan, W., Edwards, A. e Yu, P. S. (2014). RS-Forest: A rapid density estimator for streaming anomaly detection. Em *2014 IEEE International Conference on Data Mining*, páginas 600–609.

Modelagem de Custo para Redes Móveis Centralizadas de Nova Geração

Daniel da S. Souza, Carlos A. de M. Teixeira, Marcos C. da R. Seruffo, Diego L. Cardoso

[1]Instituto de Tecnologia – Universidade Federal do Pará (UFPA)
Caixa Postal 479 – Belém – PA – Brasil

{danielssouza,seruffo,diego}@ufpa.br, carlos.mattos@itec.ufpa.br

Abstract. *Upon the challenges offered by the fifth generation of mobile networks, the Centralized Radio Access Network architecture is receiving increasing attention for offering support to the ultra-dense high capacity 5G networks. This paper presents a cost evaluation methodology aimed at mobile C-RAN, covering both the capital and operational expenditures in the process. The proposed model is used in a case study in which the total cost of ownership of the distributed and centralized architectures is compared. The results points out that there is an economy of 28% in the centralized scenario and highlights the most relevant aspects in the C-RAN planning.*

Resumo. *Diante dos desafios propostos pela quinta geração de redes móveis, a arquitetura Centralized Radio Access Network (C-RAN) vem ganhando espaço por oferecer suporte à redes ultra-densas de alta capacidade de 5G. Este trabalho propõe uma metodologia de análise de custo para C-RAN, abrangendo as despesas de implantação e de operação. O modelo proposto é utilizado em um estudo de caso em que o custo total de implementação e operação das arquiteturas distribuídas e centralizadas são comparados. Os resultados apontam uma economia de 28% nos cenários centralizados e destacam os aspectos econômicos mais relevantes no planejamento da C-RAN.*

1. Introdução

Em 2021, o tráfego global em redes móveis alcançará a marca de 49 exabytes mensais, cerca de meio zetabyte anual, representando um crescimento de 700% em relação ao ano de 2016 [1]. As redes celulares de quinta geração, previstas para o ano de 2020, pretendem suprir esta crescente demanda por tráfego, oferecendo não apenas maiores velocidades, mas uma arquitetura de rede mais heterogênea e uma maior integração entre a vasta gama de dispositivos conectados simultâneamente. A 5G apresenta desafios em sua concepção, quesitos como eficiência energética e econômica devem ser otimizados para que as premissas de heterogeneidade e altas velocidades possam ser alcançadas de forma viável.

Tendo em vista os requisitos da próxima geração de redes de acesso, a arquitetura *Centralized Radio Access Network* (C-RAN) propõe a centralização, compartilhamento e alocação inteligente de recursos computacionais. Aliadas a C-RAN, a 5G trás consigo diversas tecnologias que possibilitam a consolidação desta nova realidade, tais como *Ultra-dense Networks* (UDN), *Advanced Inter-cell Interference Coordination* (ICIC),

Massive Multiple-Input Multiple-Output e (*Massive* MIMO) [2]. Os benefícios oriundos da atualização das arquiteturas tradicionais e da implantação de C-RAN são comprovados em diversos estudos, porém a implementação de tais tecnologias ocasionam em desafios em termos de custo, eficiência energética [3] e processamento [4].

Por outro lado a introdução de C-RAN no cenário atual resulta em um impacto econômico relacionado a atualização da arquitetura das redes móveis, pois o crescente quantitativo de equipamentos de backhaul e fronthaul traz consigo novos desafios e questionamentos para as operadoras em relação às margens de lucro e viabilidade econômica das novas soluções. No estudo desenvolvido em [13] são propostas soluções para otimizar as novas implantações de C-RAN em termos de minimização do custo de capital da rede, por exemplo. Duas abordagens podem ser levadas em consideração durante o planejamento da implantação de C-RAN, a primeira consiste na migração e adaptação da infraestrutura já existente para a nova arquitetura (*Brownfield*), a abordagem considerada neste trabalho propõe a implantação completa de uma nova infraestrutura (*Greenfield*). Em [10] e [11] são apresentadas metodologias abrangentes para avaliação de TCO na implantação de redes heterogêneas, porém não aplicadas especificamente à arquiteturas centralizadas.

Com o desafio proposto às operadoras de telefonia móvel de atender a demanda futura de dados e para que o planejamento e implantação das tecnologias necessárias para a 5G seja possível, é necessária a padronização de serviços e o desenvolvimento de modelos que visem o controle de aspectos econômicos. Tendo em vista as necessidade do mercado, este trabalho propõe a modelagem de Custo Total de Propriedade para topologias C-RAN voltada à redes móveis de 5G. A metodologia desenvolvida busca analisar, coletar e identificar os fatores de custos mais elevados nos segmentos de *backhaul* e *fronthaul* e, com isso, oferecer um modelo que facilite o planejamento da implantação e operação de redes centralizadas. Como validação da metodologia desenvolvida é aplicado um estudo de caso comparativo entre as arquiteturas distribuída e centralizada, em abordagem *Greenfield*, a fim de destacar as vantagens da centralização.

O restante do artigo está estruturado da seguinte forma, na seção 2 é apresentada uma visão geral do estado-da-arte de *Centralized Radio Access Network*. Em seguida a modelagem matemática de despesas de capital e de operação são descritas na seção 3. A seção 4 apresenta um estudo de caso e os resultados obtidos. Por fim, as conclusões são expostas da seção 5.

2. Centralized Radio Access Network

Na arquitetura C-RAN, o hardware de processamento de *Base Band Unit* (BBU) é movido das estações de base para um local centralizado comum, servindo um grande grupo de *Remote Radio Heads (RRHs)* que não precisam de muito mais hardware para operação, apenas a RF eletrônica [5]. A arquitetura de BBUs centralizados se comunica com as RRHs através de protocolos específicos, os mais analisados e conceituados pela literatura são o *Common Public Radio Interface (CPRI)*, o *Open Base Station Architecture Initiative (OBSAI)* ou o *Open Radio Interface (ORI)*, e assegura a transmissão dos componentes dos sinais em fase e em quadratura (I/Q), controle e sincronismo às unidades de rádio [6].

Para aplicação do C-RAN, se faz necessário um link com alta taxa de transmissão e baixa latência. O mais provável é a utilização de links cabeados, como de fibra ótica,

mas padrões wireless também podem ser utilizados, desde que alcancem os requisitos necessários entre o *fronthaul* e o *backhaul*. A Figura 1 representa os setores de *fronthaul* e *backhaul* da rede móvel. O *Fronthaul* é a ligação entre os RRHs e as BBUs [7][8]. Já a nomenclatura *backhaul*, que do mesmo modo utiliza fibra como canal de retorno, e refere-se as conexões de rede entre as estações de base e a rede principal, ou *core network* [7][8].

As redes de *fronthaul* e *backhaul* baseadas em fibra são organizadas em topologias de árvore ou em PtP. No caso de PtP, como no cenário proposto na seção IV, um terminal de linha ótica (OLT) localizado em um *Central Office* está ligado a um switch de agregação que fará a distribuição para as BBU *Pools* existentes. O *backhaul* baseado em fibra oferece capacidade praticamente ilimitada em longas distâncias. No entanto, é relativamente caro e lento para implementar em áreas onde não existe infraestrutura de fibra.

O custo de implantação de equipamentos de *fronthaul* o *backhaul* de redes tradicionais é muito alto, logo a questão custo-benefício das redes C-RAN pode ser considerada um fator primordial para a implantação da nova geração de redes móveis, proporcionando uma redução dos custos de fundação e com isso menores custos para os usuários finais. Essa economia é ocasionada pela centralização da arquitetura, tendo em comparação às redes tradicionais a eliminação de alguns equipamentos e o compartilhamento, como é o caso das BBUs, pois em uma rede tradicional é necessário uma BBU para cada estação base, porém na arquitetura centralizada, várias RRHs podem ser servidas por uma BBU *Pool*, se o limite de processamento desta não for ultrapassado. Segundo [9] uma BBU pode atender apenas seis RRHs. Por esta razão, o conceito de C-RAN é uma forma viável de reduzir as despesas de capital e de operação das operadoras.

Logo, nas redes centralizadas o aumento da distância entre BBU *Pool* e RRH leva a uma maior quantidade de cabeamento de fibra necessários no setor de *fronthaul*, o que pode traduzir-se em maiores custos de implantação para a rede de transporte. Uma modelagem de custo de implantação e operação é realizada visando um melhor planejamento para a rede móvel de arquitetura centralizada.

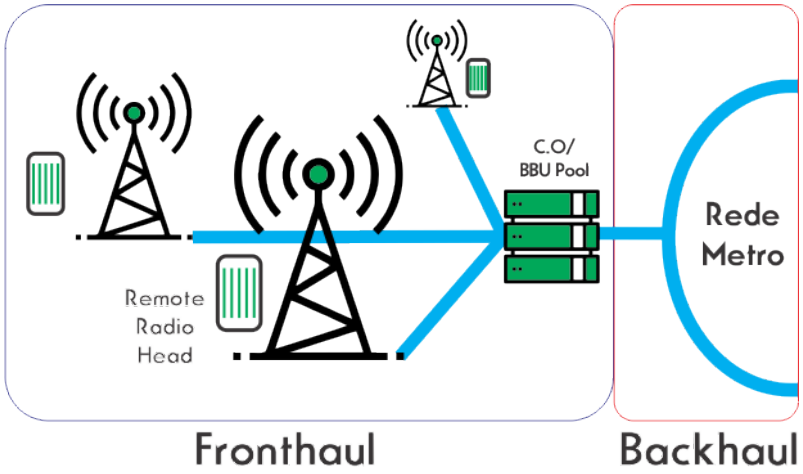


Figure 1. Cloud Radio Access Network

3. Modelagem de Custo

Nesta seção é apresentado um modelo de *Total Cost of Ownership* que abrange os custos de CAPEX e OPEX de uma rede móvel 5G em arquiteturas centralizadas. A formulação matemática detalha todos os investimentos relacionados a implantação da rede, incluindo também os gastos de cunho operacional.

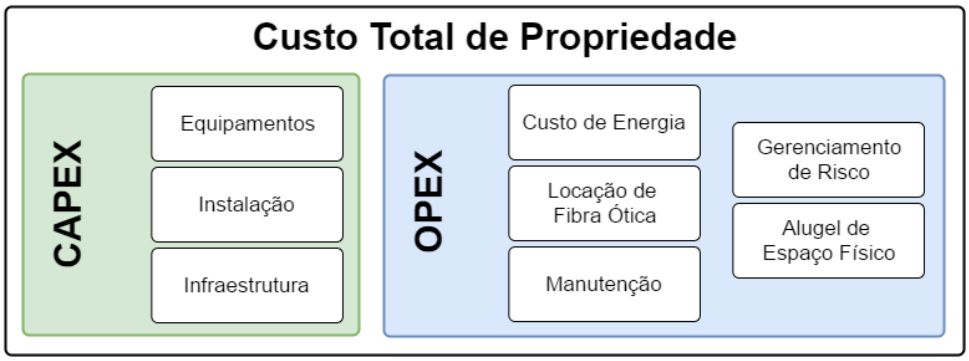


Figure 2. Custo Total de Propriedade

3.1. Capital Expenditure

O CAPEX representa os custos relacionados a fase de implantação da rede, abrangendo a compra e instalação de equipamentos, além da implantação da infraestrutura.

$$CAPEX = Equip_{cost} + Inst_{cost} + Infra_{cost} \tag{1}$$

3.1.1. Equipamentos

A equação 2 representa o total de despesas relacionadas a compra dos equipamentos e componentes de *backhaul* e *fronthaul* necessários para a implantação da rede centralizada.

$$Equip_{cost} = \sum_{i=1}^{n} N_i^{Eq} P_i^{Eq} \tag{2}$$

Em que N_i^{Eq} e P_i^{Eq} representam cada tipo de equipamento i e seus respectivos preços. O tipo de equipamentos e sua quantidade depende especificamente da arquitetura escolhida para implantação e da tecnologia de acesso que servirá de *fronthaul*.

Finalizada a fase de compra de todos os equipamentos de *backhaul* e *fronthaul*, a equipe técnica passa a fase de instalação e testes de todos os equipamentos. O custo total para a instalação de cada equipamento em suas devidas localidades é expressa na equação 3.

$$Inst_{cost} = \sum_{i=1}^{n} (T_i^{InstallPort} N_i^{Ports}) N_i^{Eq} P_{tech} \tag{3}$$

Em que $T_i^{InstallPort}$, N_i^{Ports}, N_i^{Eq} e P_{tech} representam, respectivamente, o tempo em horas necessário para se instalar uma porta do equipamento i, o número de portas a

serem instaladas para o equipamento, o quantitativo de equipmentos e o salário do técnico responsável. Sendo i a representação de cada tipo de equipamento.

3.1.2. Infraestrutura

Abrange os investimentos relacionadas ao custo total da infraestrutura de um segmento de *fronthaul* e *backhaul* em um cenário móvel utilizando-se da fibra como tecnologia de acesso. Os segmentos de fibras são colocados dentro dos dutos que estão enterradas sob o solo (tunelamento). O custo da infraestrutura de fibra inclui todas as despesas relacionadas com a escavação de dutos, compra e implantação de cabos de fibra nos dutos. Esta despesa pode ser expressa da seguinte forma:

$$Infra_{cost} = L_{trench}P_{trench} + L_{fiber}P_{fiber} + L_{lease}P_{lease} \tag{4}$$

Em que L_{trench}, P_{trench}, L_{fiber} e P_{fiber} representam a distância total do tunelamento, o preço do tunelamento por quilômetro(Km), o total de fibra ótica a ser utilizado e seu preço por Km, respectivamente. As variáveis L_{lease} e P_{lease} referem-se aos gastos iniciais referentes ao aluguel da infraestrutura fibra ótica, quando necessários.

3.2. Operational Expenditure

Os elementos que compõem o OPEX da rede remetem ao capital necessário para manter e aprimorar os recursos previamente implementados no CAPEX. São incluídos os custos de energia, manutenção e de gerenciamento de risco, além das taxas de aluguel de fibra ótica e do espaço físico necessário para abrigar os equipamentos, demostrado na equação 5.

$$OPEX = En_{cost} + FL_{cost} + Mt_{cost} + FM_{cost} + FS_{cost} \tag{5}$$

3.2.1. Custo de Energia

O consumo de energia de uma rede móvel centralizada que utiliza fibra é obtida a partir da soma do consumo de energia de todos equipamentos nos setores de *backhaul* e *fronthaul*(*Central Office, Cell sites*, BBU *Pool*, etc).

$$En_{cost} = EC_{CellSite} + EC_{CO} \tag{6}$$

O cáculo de consumo de energia de um *Cell Site* ($EC_{CellSite}$) é mostrado na equação 7.

$$EC_{CellSite} = \sum_{i=1}^{n} EC_i^{CSEquip} P_{OutdoorKwh}, \tag{7}$$

em que $EC_i^{CSEquip}$ e $P_{OutdoorKwh}$ representam, respectivamente, o gasto de energia em cada célula i e o preço por unidade de energia (isto é, kWh) consumida.

O custo do consumo de energia da *Central Office* (EC_{CO}) é mostrada pela equação 8.

$$EC_{CO} = \sum_{i=1}^{N_{CO}} R \sum_{i=1}^{n} EC_i^{COEquip} P_{IndoorKwh}, \tag{8}$$

em que $EC_i^{BBUPEquip}$ representa o gasto de energia de cada equipamento i alocado dentro de uma *Central Office*. O coeficiente de refrigeração (R) é usado para contabilizar o resfriamento na *Central Office*.

3.2.2. Locação de Fibra Ótica

O custo de locação de fibra é expresso pela equação 9.

$$FL_{cost} = L_{lease} P_{lease} \tag{9}$$

Sobre a locação de fibras, a operadora de telefonia faz pagamento de uma taxa anual para a manutenção e reparação, além das despesas iniciais descritas na seção 3.1.2. Este custo é calculado multiplicando o comprimento total de fibras alugadas em Km (L_{lease}) pela taxa anual de manutenção por Km (P_{lease}).

3.2.3. Custo de Manutenção

Os custos de manutenção representam os gastos relacionados ao monitoramento e reparo dos equipamentos do *Central Office* e dos *Cell Sites*. Inclui-se também os custos de taxa anual para as licenças de software. Os custos relacionados a manutenção são necessários para manter o *fronthaul* e *backhaul* sempre em boas situações de operação. O custo total é calculado usando a equação 10.

$$Mt_{cost} = Mt_{CS} + Mt_{CO} + Mt_{SWL}, \tag{10}$$

em que Mt_{CS} e Mt_{CO} são os custos de manutenção dos *Cell Sites* e dos *Central Office*, respectivamente. Por fim é contabilizado a taxa anual da lincenças de software (Mt_{SWL}). A equação a seguir traz o cálculo de manutenção do *Cell Site*:

$$Mt_{CS} = P_{Tech}(RT_{CS} + 2T_{Travel}), \tag{11}$$

em que P_{Tech}, RT_{CS} e T_{Travel} representam, respectivamente, o salário do técnico por hora, o tempo requerido para manutenção de cada *Cell Site* e o tempo de viagem para locação onde está cada *Cell Site*.

O *Central Office* é a localidade da rede em que estão localizados os equipamentos responsáveis para funcionamento da C-RAN, como por exemplo a OLT e as BBU *Pools*, logo os operadores consideram a necessidade de se ter várias rodadas de procedimentos de manutenção para cada *Central Office*, dependendo do número de usuários e serviços abrangidos por cada um deles. Esse cálculo de custo de manutenção é expresso da seguinte forma:

$$Mt_{CO} = RT_{COR}N_{CO}P_{Tech} + P_{Upgrade}, \quad (12)$$

em que RT_{COR},N_{CO} e $P_{Upgrade}$ representam, respectivamente, o tempo necessário em homens horas para reparação de cada *Central Office* em um ano, a quantidade de *Central Office* e o custo a ser pago para atualização de hardware e para substituição de componentes, por exemplo as baterias.

3.2.4. Gerenciamento de Risco

O gereciamento de riscos é referente às despesas de reparo no *fronthaul* e *backhaul* na ocorrência de falhas na rede.

$$FM_{cost} = \sum_{i=1}^{n}((RT_i + 2T_{Travel})N_{Tech}P_{Tech} + P_rp)ANF_iN_i^{Eq}, \quad (13)$$

em que RT_i, N_{Tech} e P_rp representam, em ordem, o tempo de reparo cada equipamento i, o número de técnicos necessários para reparo da falha e o custo da reparação caso seja necessária a compra de um novo componente. O número médio de falhas por ano de cada tipo de componente i (ANF_i) que pode ser calculado com base na taxa de falha do componente, que ao ser multiplicado pelo número de equipamentos do tipo i na rede (N_i^{Eq}) resulta no número esperado de falhas do componente do tipo i durante um ano.

3.2.5. Aluguel de Espaço Físico

O custo de espaço (FS_{cost}) é uma taxa de aluguel anual paga pela operadora da rede para alojar seus equipamentos, como mostrado na expressão 14:

$$FS_{cost} = FS_{CO} + FS_{CS} \quad (14)$$

A váriavel A_{Rack} determina o espaço necessário para um rack, levando em consideração o espaço de trabalho aceitável na frente dele para que os técnicos possam atuar no rack. O número de racks dentro de um *Central Office* é medido dividindo o número de equipamentos por *Central Office* (N_{Eq}^{CO}) pelo número de equipamentos permitidos por rack (N_{Rack}^{CO}).

$$FS_{CO} = (A_{Rack}(N_{Eq}^{CO}/N_{Rack}^{CO}))P_{IndoorM2} \quad (15)$$

Somado a equação anterior, temos agora a equação que calcula o custo do espaço físico para os *Cell Sites*, em que A_{CS}, N_{CS} e $P_{OutdoorM2}$ representam o espaço necessário para implantação de um *Cell Site*, a quantidade de *Cell Sites* e o preço do aluguel anual pago pelas operadoras de telefonia.

$$FS_{CS} = A_{CS}N_{CS}P_{OutdoorM2} \quad (16)$$

Uma vez realizada a modelagem, foi realizado um estudo de caso com o intuito de aplicar o modelo proposto em um cenário de abordagem *Greenfield*.

4. Estudo de Caso

Esta seção apresenta um estudo de caso em que o modelo de TCO apresentado é aplicado ao cenário proposto para avaliar os custos de implantação e operação da rede móvel com abordagem *Greenfield*. O cenário utilizado, que abrange uma área de 15 Km^2, é composto por 4 BBU *Pools* e 169 RRHs, sendo 70% destas localizadas em áreas residenciais e o restante em áreas comerciais, que apresentam elevados índices de tráfego [13].

A estratégia para alocação de RRHs em cenários com múltiplas BBU *Pools* apresentada em [13] utiliza um modelo de programação linear inteira para minimizar o CAPEX da rede. A alocação considera o limite de cada BBU *Pool* em termos de tamanho e carga, a máxima distância entre RRH e BBU *Pool* é respeitada para evitar atrasos na rede, por fim, o mesmo número de células comercias é alocado a cada BBU *Pool* de modo que o ganho de multiplexação estatística seja maximizado. Diferente de [13], que utilizou de distância euclidiana para estipular as distâncias entre as localidades da rede móvel, foi utilizada a distância de Táxi, abordagem apresentada por [14] que considera apenas caminhos horizontais e verticais, retratando com fidelidade um cenário urbano real. O cenário de entrada [13] e a alocação resultante são ilustrados pela Figura 3, em que estão representadas as 4 BBU *Pools* e as 169 células com seus respectivos perfis de tráfego, as células de perfil comercial são destacadas das demais. A alocação das RRH's é representada de acordo com a cor de cada BBU *Pool*.

A interconexão do *fronthaul* é feita com fibra ótica monomodo utilizando um padrão de sinal de rádio digital sobre fibra (D-RoF), como o CPRI. Distribuições de perfis de tráfego distintas são apresentadas, isso implica na ligação da BBU *Pool* com o *Cell Site*, pois dependendo da carga que cada célula possui é determinado o seu processamento para a BBU *Pool* mais próxima e com processamento disponível para alocação. A Tabela 1 resume os parâmetros de custo utilizados para calcular o TCO de *fronthaul* e *backhaul*. A normalização dos preços utilizados foi baseada no preço do quilômetro de fibra ótica. Em casos de *CostFactor* muito baixo, mesmo que o número de BBUs necessárias seja muito menor, o comprimento de fibra a ser instalado é alto tornando a arquitetura impraticável, nestes casos a diferença entre o custo de uma BBU e um quilômetro de fibra é muito pequeno.

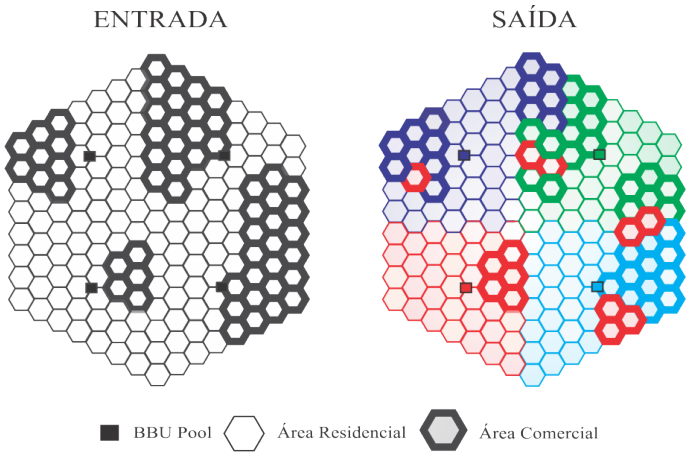


Figure 3. Cenário de entrada e alocação de recursos resultante.

Table 1. Valores Normalizados

Equipamento/Serviço	Custo Normalizado
Salário do técnico (hora)	0,33
Custo de Energia (kWh)	0,000625
Fibra (Km)	1
Tunelamento (Km)	303,13
Switch de Agregação	1,88
Antena Micro Cell	12,50
Radio Head Unit	3.75
BBU *Pool*	838,50
OLT	18,75

Segundo os resultados obtidos a partir do modelo, o cenário utiliza 592 Km de fibra ótica. Tendo em vista que o modelo sempre prioriza a otimização do uso de recursos da BBU *Pool* e a minimização do CAPEX ao realizar a alocação de RRHs, é possível observar que certas células são alocadas a BBUs de outros setores.

Uma das alternativas do modelo proposto em [11] foi utilizado para estimação do custo total de propriedade de um cenário comparativo de arquitetura tradicional, ou seja, distribuída e de topologia pré-definida baseada em fibra. A Figura 4 apresenta uma projeção de CAPEX e OPEX em um período de 20 anos para ambas as arquiteturas. É possível observar que o cenário C-RAN apresenta uma economia de aproximadamente 28% do custo total de propriedade em relação a arquitetura distribuída.

O modelo tradicional, por ser distribuído, utiliza uma BBU para cada RRH do cenário, justificando a economia na compra de equipamentos presente na C-RAN. Em contrapartida, na arquitetura C-RAN são necessários 32 BBUs, referentes às 04 BBUs *Pools* e 592 Km de fibra. Para situações em que o custo de uma BBU é expressivo, a abordagem C-RAN se torna mais atraente, pois reduz o número de BBUs necessárias em aproximadamente 81% devido ao compartilhamento de recursos computacionais.

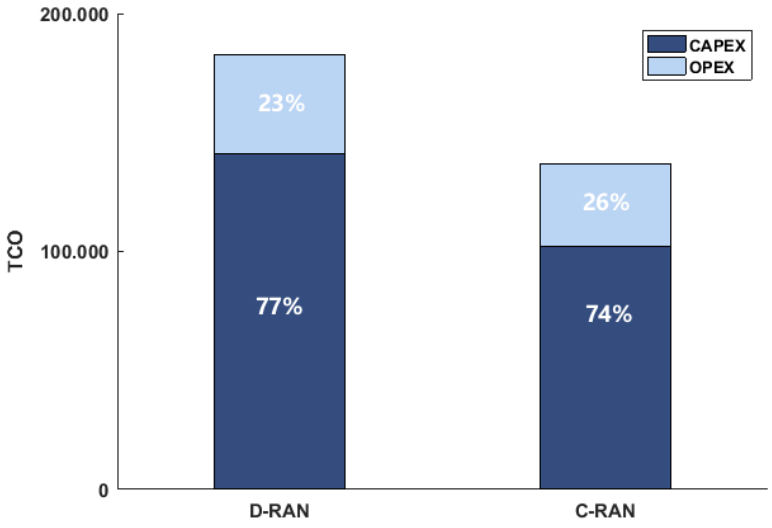


Figure 4. Projeção de TCO a 20 anos

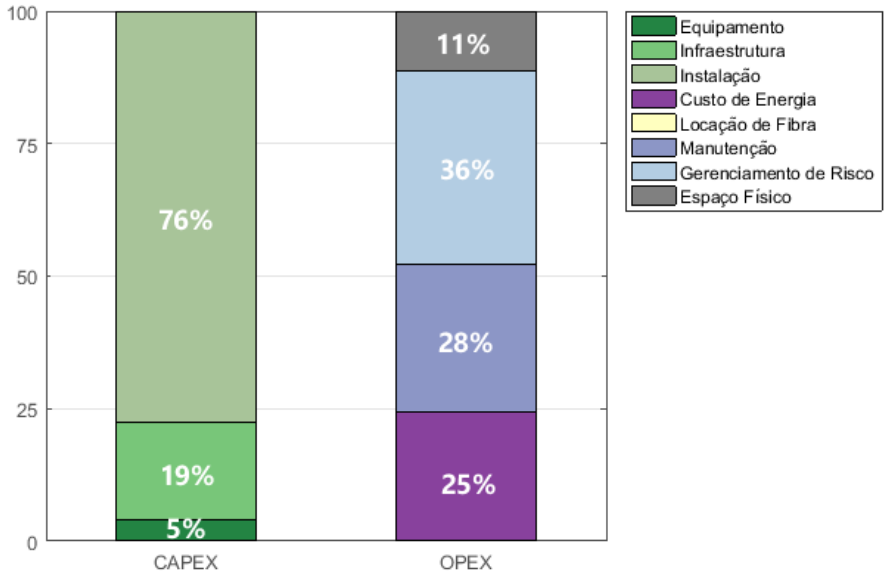


Figure 5. Custo Total de Propriedade da C-RAN

A Figura 5 ilustra os valores de CAPEX e OPEX do cenário de arquitetura centralizada, os valores são apresentados em uma perspectiva anual. É importante reforçar que, como já dito em [15], os custos relacionados a infraestrutura representam uma grande parte do capital inicial investido para implantação de um cenário de abordagem *Greenfield*, este fato é justificado pela necessidade de realizar o tunelamento de fibra em todas as localidades do *fronthaul*. No cenário em estudo, as despesas de infraestrutura representam 76% do valor total de CAPEX. Levando em consideração que não há locação de equipamentos, fibra ou dutos já existentes. Desta forma, os custos de compra de equipamentos e instalação representam uma fatia menor do CAPEX.

Em relação ao OPEX, é importante notar a economia nos custos de consumo de energia, manutenção da rede e aluguel de espaço físico. A diminuição no consumo de energia é justificada pela diminuição no quantitativo de equipamentos da rede, pela centralização do processamento e pelo compartilhamento de recursos computacionais na BBU *Pool*. Tais fatores também são responsáveis pela economia no setor de manutenção e gerenciamento de riscos da rede móvel, a arquitetura centralizada permite a operadora de telefonia móvel uma maior flexibilidade nesses quesitos, vantagem essa relacionada a possibilidade de melhor gereciamento, pois grande parte dos serviços podem ser feitos remotamente, assim como o gerenciamento de riscos. A economia no aluguel de espaço físico em comparação as arquiteturas distríbuidas atuais é justificada pela centralização das unidades de processamento de banda base, diminuindo consideravelmente o espaço necessário para implantação de um novo *Cell Site*.

Conclusões

O custo para construir, operar e modernizar a RAN está se tornando cada vez mais elevado, enquanto o lucro não está crescendo na mesma proporção. Por outro lado, a proliferação de internet de banda larga móvel também apresenta uma oportunidade única para o desenvolvimento de uma arquitetura de rede evoluída que permitirá novas

aplicações e serviços e uma maior eficiência em termos de custo benefício. Para conservar a ganho e o desenvolvimento, as operadoras móveis devem encontrar soluções para reduzir custos. Com isso a implantação de arquiteturas de próxima geração de redes móveis, como o C-RAN, se faz necessária com um gerenciamento de redes bem elaborado.

Levando em consideração que o gerenciamento de redes está associado ao controle das atividades e ao monitoramento do uso dos recursos no ambiente da rede nos anos que segue em operação. Um planejamento prévio de instalação e operação é de extrema importância, coletando dados das tarefas mais básicas de gerência e operação de redes, resumidamente, com esses dados é possível: obter as informações da rede, tratá-las para diagnosticar possíveis problemas de operação e aplicar as soluções destes problemas, tendo assim uma resposta rápida e eficiente afetando a relação custo benefício de toda a rede móvel.

Este trabalho apresenta como contribuição uma modelagem de TCO para redes móveis focada em arquiteturas C-RAN. O foco do trabalho é apresentar uma solução de *backhaul* e *fronthaul*, utilizando fibra ótica como tecnologia de acesso, que possa suportar o crescente tráfego da quinta geração de redes celulares. Foi desenvolvido um estudo de caso comparativo que apresenta um cenário de abordagem *Greenfield* de topologia ponto-a-ponto com o objetivo de destacar as vantagens da arquitetura C-RAN em relação a D-RAN. Os resultados evidenciam a considerável economia, tanto em CAPEX como em OPEX, advindas da centralização do processamento de banda base e compartilhamento de recursos computacionais.

Para ampliar as contribuições apresentadas neste trabalho é possível fazer diversas modificações e evoluções. Alternar o tipo de tecnologia de acesso usado nos cenários montados, fibra ótica ou micro-ondas, a topologia da rede, além da arquitetura ponto-a-ponto usado neste trabalho, para verificar a aplicabilidade e flexibilidade da modelagem desenvolvida.

References

[1] *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016-2021*. Cisco, 2017.

[2] M. Jaber, M. Ali Imram, R. Tafazolli, A. Tukmanov. *5G Backhaul Challenges and Emerging Research Directions: A Survey*, IEEE Access, 2016.

[3] S. Tombaz, B. Skubic, L. Wosinska, P. Monti. *Modelling Energy Performance of CRAN with Optical Transport in 5G Network Scenarios*, Journal of Optical Communications and Networking, 2016.

[4] Y. Liao, L. Song, Y. Li, Y, Zhang. *How Much Computing Capability Is Enough to Run a Cloud Radio Access Network?*, IEEE Communications Letters, Vol. 21, 2017.

[5] T. Pfeiffer. *Next Generation Mobile Fronthaul and Midhaul Architectures [Invited]*, B38 J. Opt. Commun. Netw., 2015.

[6] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, L. Dittmann. *Cloud RAN for Mobile Networks - a Technology Overview*, IEEE Communications Surveys and Tutorials, 2014.

[7] Bartelt, Jens, et al. *Fronthaul and backhaul requirements of flexibly centralized radio access networks*, IEEE Wireless Communications, 2015.

[8] Schulz, Dominic, et al. *Robust optical wireless link for the backhaul and fronthaul of small radio cells*, Journal of Lightwave technology ,2016.

[9] Ericsson Review. *Connecting the dots: small cells shape up for high performance indoor radio*, 2014.

[10] F. S. Farias. *Designing Cost-Efficient Transport Solutions for Fixed and Mobile Broadband Access Network*, UFPA, 2016.

[11] M. Mahloo, P. Monti, J. Chen, L. Wosinska. *Cost Modeling of Backhaul for Mobile Networks*, ICC'14 - W9: Workshop on Fiber-Wireless Integrated Technologies, Systems and Networks, 2014.

[12] A. Checko, H. Holm, H. Christiansen. *Optimizing Small Cell Deployment by the use of C-RANs*, European Wireless, 2014.

[13] H. Holm, A. Checko, R. Al-obaidi, H. Christiansen. *Optimal Assignment of Cells in C-RAN Deployments with Multiple BBU Pools*,2015 European Conference on Networks and Communications, 2015.

[14] Bristin R., Paul A. *Taxicab Trigonometry*, Pi Mu Epsilon Journal, vol. 8, pp. 89-85, 1985.

[15] C. Ranaweera, C. Lim, A. Nirmalathas, C. Jayasundara, and E. Wong. *Cost-optimal Placement and Backhauling of Small-cell Networks*, J. Lightw. Technol., 2015.

Uma Arquitetura de Virtualização de Funções de Rede para Proteção Automática e Eficiente contra Ataques

Leopoldo A. F. Mauricio[1,3], Igor D. Alvarenga[1], Marcelo G. Rubinstein[2] e Otto Carlos M. B. Duarte[1]

[1]Universidade Federal do Rio de Janeiro - GTA/PEE/COPPE
[2]Universidade do Estado do Rio de Janeiro - FEN/DETEL/PEL
[3]Globo.com

leopoldo@corp.globo.com,{alvarenga,otto}@gta.ufrj.br,rubi@uerj.br

Resumo. *Sistemas de prevenção de intrusão são caros e ineficientes ao bloquearem todo o tráfego de um endereço de origem. Logo, a capacidade de aplicar contramedidas somente ao tráfego malicioso se torna um importante desafio de segurança. Este artigo propõe uma arquitetura de virtualização de funções de rede (Network Functions Virtualization - NFV) que oferece proteção automática e eficiente contra ataques. Os fluxos maliciosos são dinamicamente desviados por uma cadeia de funções virtuais de rede (Virtual Network Functions - VNFs) contendo um firewall de aplicação e um IDS. Um protótipo foi construído utilizando-se a plataforma Open Platform for NFV (OPNFV). A arquitetura é capaz de bloquear 99,12% dos ataques sem afetar 93,6% do tráfego benigno.*

Abstract. *Intrusion Prevention Systems (IPSs) are expensive and inefficient when blocking all traffic from a source address. Therefore, the ability to apply countermeasures only to malicious traffic becomes an important security challenge. This article proposes a Network Function Virtualization (NFV) architecture to provide automatic, and efficient protection against attacks. Malicious flows are dynamically diverted through a Virtual Network Functions (VNFs) chain containing an application firewall and an IDS. A prototype was built using the Open Platform for NFV (OPNFV). The architecture is capable of blocking 99.12% of the attacks without affecting 93.6% of the benign traffic.*

1. Introdução

Os ataques de segurança, cada vez mais frequentes na Internet, causam grande prejuízo a pessoas e organizações [Akamai 2017, Anderson et al. 2013]. Ataques de negação de serviço (*Denial of Service* - DoS) visam consumir os recursos do alvo atacado para causar indisponibilidade, enquanto outros, por exemplo, exploram vulnerabilidades para desfigurar o conteúdo de sistemas, propagar código malicioso ou roubar dados sensíveis [Gu et al. 2008, Haggard and Lindsay 2015]. Por isso, sistemas de proteção de rede devem ter capacidade de detectar e aplicar contramedidas eficientes contra os ataques de segurança, a fim de proteger dispositivos, aplicações e redes ligadas à Internet. Além disso, os sistemas de proteção devem ser ágeis e eficientes, de forma a reduzirem o tempo entre a detecção e a reação necessária para mitigar ataques [Amann and Sommer 2015].

Muitas vezes os ataques são gerados a partir dos nós de uma *botnet* [Mtibaa et al. 2015]; um conjunto de máquinas zumbis com *malwares* que permitem a

um atacante controlar o dispositivo “infectado”. Nesse caso, os usuários reais das máquinas escravizadas não sabem que existe tráfego malicioso gerado a partir de seus dispositivos, que podem ser computadores, aparelhos de celular, aparelhos de TV, etc. Além disso, o endereço IP de origem de um tráfego identificado como malicioso pode pertencer a um *gateway* NAT (*Network Address Translation*) que pode servir grande quantidade de nós de rede. Em qualquer dos casos, o bloqueio do endereço IP de origem é uma contramedida eficaz contra ataques de negação de serviço. Entretanto, também é preciso impedir os ataques que exploram vulnerabilidades, tais como a injeção de SQL (*Structured Query Language Injection* - SQLI) e o *cross-site scripting* (XSS), sem afetar o tráfego benigno gerado pelos dispositivos. O XSS é um tipo de vulnerabilidade encontrada normalmente em aplicações web, que ativa ataques maliciosos ao injetar *client-side script* dentro das páginas web vistas por outros usuários. Assim, os sistemas de proteção devem ser eficientes para bloquear todas as atividades maliciosas sem impedir o acesso benigno às redes e aplicações.

A virtualização de funções de rede (*Network Functions Virtualization* - NFV), normatizada pelo *European Telecommunications Standards Institute* (ETSI), é uma nova tecnologia que tem como objetivo tornar as redes mais ágeis e flexíveis e reduzir os custos materiais e de operação. O NFV propõe a implantação de funções de rede virtuais (*Virtual Network Functions* - VNFs) em hardware genérico de prateleira, como alternativa ao uso de dispositivos de rede e segurança customizados [ETSI 2012]. Dessa forma, tanto o custo de capital (CAPEX) quanto o custo operacional (OPEX) são reduzidos. Isso ocorre porque dispositivos especializados não são utilizados e as despesas com dissipação de calor, consumo de eletricidade e custo de manutenção são reduzidas, através da diminuição da quantidade de dispositivos físicos utilizados.

Este artigo propõe uma arquitetura de virtualização de funções de rede para proteção automática e eficiente contra ataques de segurança. Na arquitetura proposta, um módulo de controle foi criado para gerenciar e operar VNFs que capturam amostras de tráfego, detectam ameaças e filtram atividades maliciosas de forma automática, sem impedir o tráfego benigno. A arquitetura NFV proposta aumenta a eficiência da mitigação de ataques, porque apenas o tráfego malicioso é detectado e bloqueado através do encadeamento de um sistema de detecção de intrusão (*Intrusion Detection System* - IDS) a um *firewall* de aplicação. A arquitetura foi implementada e avaliada na plataforma de código aberto para virtualização de funções de rede (*Open source Platform for Network Functions Virtualization* - OPNFV) utilizando servidores de prateleira, onde VNFs foram criadas para atuarem como IDS e *firewall* de aplicação [OPNFV 2017]. Os resultados obtidos demonstram que o sistema proposto apresenta uma alta eficiência.

O restante do artigo está organizado da seguinte forma. A Seção 2 discute trabalhos relacionados. As características da arquitetura NFV proposta são descritas na Seção 3. Na Seção 4, os detalhes de implementação do protótipo são discutidos. Os resultados da avaliação do sistema são apresentados na Seção 5. A Seção 6 apresenta as principais conclusões e sugestões para trabalhos futuros.

2. Trabalhos Relacionados

Diversos autores utilizam propriedades das redes definidas por software (*Software Defined Networks* - SDNs) e de virtualização de funções de rede para propor soluções de gerenciamento de rede e de segurança. Deng *et al.* propuseram o *framework* VNGuard,

que utiliza os conceitos de SDN e de NFV para prover e gerenciar *firewalls* virtuais denominados VNF-FWs [Deng et al. 2015]. Os componentes do VNGuard foram implementados no ClickOS [Martins et al. 2014], que é uma plataforma de software flexível, extensível e escalável. Foram realizados experimentos que mostraram que o tempo médio de processamento por pacote na VNF-FW cresce em função do aumento da quantidade de regras do *firewall*. Todavia, a implementação de um sistema automático para proteção contra ataques não é investigada pelo autores.

Zanna *et al.* mostraram que é possível integrar um sistema de detecção de intrusão com um controlador SDN para realizar detecção e bloqueio de ataques de negação de serviço [Zanna et al. 2014]. O IDS Bro foi configurado para enviar uma requisição de criação de fluxo de bloqueio para a interface de programação (*Application Programming Interface* – API) REST do controlador Ryu, a fim de filtrar os IPs geradores do tráfego malicioso. Porém, a proposta não aborda a criação e a remoção dinâmica de regras para proteção contra ataques de exploração de vulnerabilidades.

Andreoni Lopez *et al.* propuseram o BroFlow, que combina o IDS Bro e o controlador POX para bloquear automaticamente ataques com base nos IPs de origem e de destino [Andreoni Lopez et al. 2014]. Foi implementado um algoritmo para gerar ataques de negação de serviço e um módulo foi criado no IDS Bro para programar os fluxos de bloqueio na rede. Contudo, apenas contramedidas adequadas para evitar ataques de negação de serviço são implementadas, através do bloqueio de todo tráfego gerado pelo IP de origem identificado.

Lin *et al.* propuseram uma arquitetura SDN capaz de classificar tráfego e realizar prevenção de intrusão [Lin et al. 2015]. A proposta implementa uma arquitetura capaz de detectar tanto ataques de negação de serviço quanto os que exploram vulnerabilidades, através da identificação de padrões maliciosos em requisições HTTP. Contudo, o foco do artigo é a diminuição da sobrecarga de tráfego que é enviado para os controladores SDN. A automação da detecção e do bloqueio de ataques não é abordada.

Este artigo propõe uma solução automática e integrada de detecção e mitigação de ataques de segurança usando as tecnologias NFV/SDN. A solução inclui um módulo Controlador de Segurança que recebe e analisa os alertas enviados por uma VNF-IDS e decide bloquear apenas o tráfego de ataque, sem prejudicar o tráfego benigno dos usuários que se servem do mesmo IP de origem.

3. A Arquitetura NFV Proposta

A arquitetura proposta estende a arquitetura NFV do ETSI. A seguir são descritos de forma resumida os principais componentes da arquitetura NFV do ETSI apresentada na Figura 1.

Cada VNF da arquitetura NFV está associada a um sistema de gerenciamento de elementos (*Element Management System* - EMS) que monitora a utilização de seus recursos. A infraestrutura NFV (*NFV Infrastructure* - NFVI) fornece hardware e software com os recursos de rede, de computação (memória e CPU) e de armazenamento necessários para implantar, gerenciar e executar VNFs. O Gerenciamento e Orquestração (*Management and Orchestration* - MANO) do NFV é responsável pelo gerenciamento do ciclo de vida dos recursos físicos e de software da NFVI e pela gestão do ciclo de vida das VNFs.

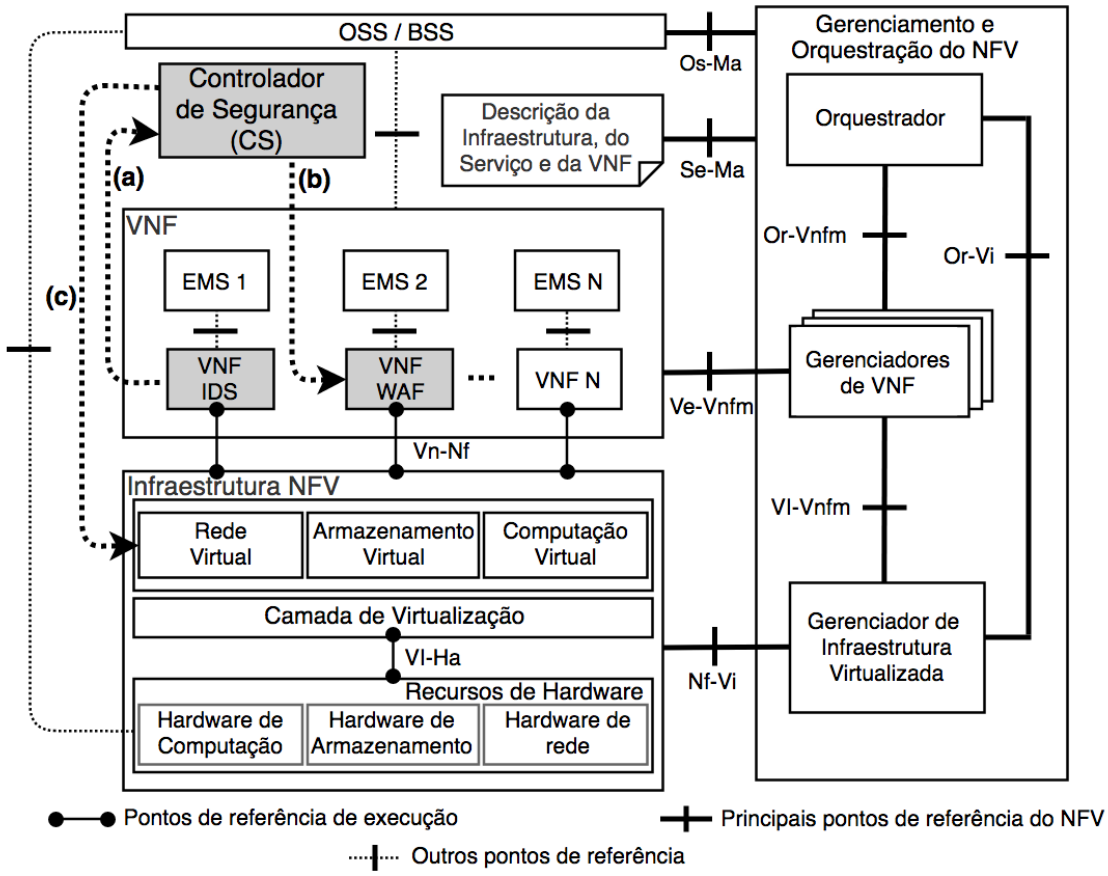


Figura 1. Arquitetura NFV estendida com os componentes de segurança propostos em cor escura. As principais interações do Controlador de Segurança proposto são: a) recebimento de alarmes; b) envio das regras atualizadas para instalação no *firewall* e c) envio de fluxos OpenFlow para serem bloqueados.

Por isso, as ferramentas necessárias para criação, atualização, migração e encerramento de VNFs estão inseridas nele.

O sistema de suporte operacional (*Operations Support System* - OSS) e o sistema de suporte empresarial (*Business Support System* - BSS) gerenciam o inventário de rede, o provisionamento de serviços, os relatórios de falhas e de faturamento etc. A interação entre os componentes da arquitetura NFV é realizada através das interfaces definidas pelo ETSI: Nf-Vi, Vi-Ha, Vn-Nf, Ve-Vnfm, Se-Ma, Os-Ma, Or-Vnfm, Or-Vi e Vi-Vnfm [ETSI 2013, ETSI 2012].

O principal objetivo da arquitetura NFV proposta é oferecer um sistema de proteção automático e eficiente contra ataques. São propostos três novos componentes na arquitetura, apresentados em cor escura na Figura 1. O principal componente é o módulo Controlador de Segurança. Ele interage com duas funções virtuais de rede específicas e com a interface de rede virtual (ver Figura 1). As duas funções virtuais de rede são um sistema de detecção de intrusão denominado VNF-IDS e um *firewall* de aplicação web denominado VNF-WAF.

A função virtual de rede de sistema de detecção de intrusão analisa os dados, detecta ameaças e envia alertas para o Controlador de Segurança. Ao receber os alertas, o

Controlador de Segurança os analisa e decide se uma atualização das regras de *firewall* é necessária. Em caso afirmativo, novas regras são enviadas à função virtual de *firewall*. O Controlador de Segurança caracteriza os alarmes e decide por uma possível estratégia de mitigação da ameaça de segurança, enviando informações para a interface de rede virtual de como determinados fluxos devem ser tratados (bloqueados etc.). Este procedimento é realizado de forma completamente automática, o que permite uma rápida reação a ataques.

4. Implementação do Sistema Proposto

A arquitetura proposta foi implementada na plataforma *Open Platform for Network Function Virtualization* (OPNFV) versão Colorado, que provê parte das funcionalidades propostas na arquitetura de virtualização de redes do ETSI. Mais especificamente, a versão Colorado provê a Infraestrutura de Virtualização de Redes (*Network Function Virtualization Infrastructure* (NFVI) e o Gerenciador de Infraestrutura Virtualizada (ver Figura 1). A OPNFV é uma plataforma de código aberto mantida por uma comunidade de desenvolvedores, que utiliza o OpenStack como Sistema Operacional (SO) de nuvem para controlar os recursos físicos da Infraestrutura NFV [OPENSTACK 2017].

Em relação ao provisionamento de redes virtuais, a plataforma OPNFV permite o uso dos controladores de redes definidas por software *OpenDaylight* (ODL) [ODL 2017] ou *Open Network Operation System* (ONOS) [ONOS 2017]. Nesta implementação, a OPNFV foi instalada com o controlador SDN *OpenDaylight*. A escolha do ODL se baseou na sua maturidade, na sua documentação de apoio e nas características de sua API. A nuvem OPNFV implantada possui três nós de computação e um nó de rede. O Controlador de Segurança e a VNF-WAF da arquitetura proposta foram implantados nos nós de computação, um componente por nó. O VNF-IDS foi implantada em uma outra máquina. Além disso, um comutador OpenFlow [Moraes et al. 2014] Open vSwitch (OVS) foi configurado em cada nó de computação. O ODL foi instalado no nó de rede da nuvem [OPENSTACK 2017].

A Figura 2 apresenta os módulos da proposta que foram implementados para mitigar ataques contra servidores web vulneráveis. As interações que ocorrem entre os módulos e a rede virtual também são apresentadas. A função virtual de rede de sistema de detecção de intrusão (VNF-IDS) proposta possui dois módulos: o módulo Detector de ameaças e o módulo Notificador de ameaças (Figura 2). O detector de ameaças foi implementado utilizando-se o Bro, que é um sistema de detecção de intrusão de rede de código aberto capaz de monitorar passivamente o tráfego de rede e procurar por atividades suspeitas [Sommer 2003]. O módulo de detecção foi configurado através de *scripts* para ser capaz de analisar e extrair informações da camada de aplicação, de modo a compará-las às atividades com padrões classificados como ataques. Dessa forma, é possível identificar ataques do tipo *cross-site scripting* (XSS) e injeção de SQL.

TAP é um tipo de operação da rede. Nesse tipo de operação, uma cópia de cada evento de rede ocorrido em uma interface é enviada para o sistema que monitora e analisa os dados. A interface de rede da VNF-IDS foi configurada em modo TAP para permitir o envio de amostras do tráfego da rede para o detector de ameaças, como mostrado na interação (1) da Figura 2. O notificador de ameaças da VNF-IDS, que foi escrito na linguagem Python, foi desenvolvido para enviar mensagens HTTP com alertas de ameaças para o controlador de Segurança (2). Na mensagem enviada, a VNF-IDS informa o

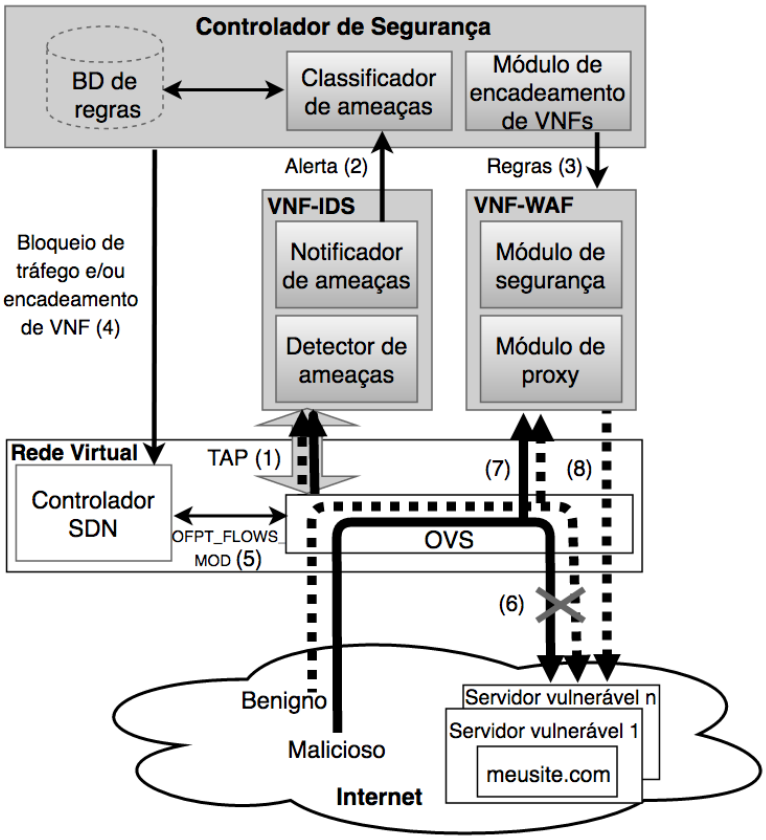


Figura 2. Arquitetura NFV implementada. Os módulos mais escuros são os propostos como extensão da arquitetura NFV do ETSI.

endereço IP que gerou o tráfego malicioso, o endereço IP atacado, a *Uniform Resource Locator* (URL) relativa ao ataque, a *Uniform Resource Identifier* (URI) e o tipo de ameaça detectada.

Um módulo classificador de ameaças, um módulo de encaminhamento de VNFs e uma base de dados (BD) de regras são implementados no controlador de segurança. Na base de dados, são armazenados os alertas de ameaça e as regras que podem ser utilizadas posteriormente no *firewall* de aplicação para bloquear ataques (Figura 2). O programa de banco de dados de código aberto MongoDB foi utilizado na criação da base de dados. A mitigação das atividades maliciosas de forma automática pode ser realizada através de uma operação de bloqueio de tráfego em todos os comutadores OVS da nuvem OPNFV ou através de uma operação de encadeamento com a VNF-FW para bloquear as atividades maliciosas, sem impedir o acesso benigno.

O Algoritmo 1 determina o tratamento dos pacotes após um ataque ser detectado. Ao receber um alerta de ataque (2), o módulo classificador de ameaças do controlador de segurança verifica se o tipo é de negação de serviço (Ψ) ou se busca explorar uma vulnerabilidade na rede ou nos sistemas protegidos pela arquitetura NFV (Φ). Se o ataque é de negação de serviço, o controlador de segurança executa uma operação de bloqueio de tráfego em todos os comutadores OVS da nuvem OPNFV. Nesse tipo de operação, o controlador de segurança envia mensagens HTTP para a API REST do *OpenDaylight* (4), que insere fluxos *OpenFlow* no OVS (5) para bloquear todo fluxo de dados gerado pelo

endereço IP de origem do ataque de negação de serviço ($\delta \rightarrow src, *$).

Algoritmo 1: TRATAMENTO DE ATAQUE.

Entrada:
$\delta = \{src, dst, tipo\}$: Ataque detectado
Ψ: Conjunto de ataques de negação de serviço
Φ: Conjunto de ataques que exploram vulnerabilidades

```
1  início
2      se δ → tipo ∈ Ψ então
3          BLOQUEIA(δ → src, *);
4      senão se δ → tipo ∉ Φ então
5          BLOQUEIA(δ → src, δ → dst);
6      senão
7          DESVIA(δ → dst);
8      fim
9      REGISTRA(δ);
10 fim
```

Se o ataque não é de negação de serviços, o classificador de ameaças verifica se a base de dados possui uma regra adequada para bloquear o ataque. Caso não exista ($\delta \rightarrow tipo \notin \Phi$), o controlador de segurança bloqueia o tráfego de ataque nos comutadores OVS ($\delta \rightarrow src, \delta \rightarrow dst$) executando os passos (4) e (5). Quando existe regra adequada na base de dados do controlador de segurança para mitigar o ataque de exploração de vulnerabilidade, o classificador de ameaças aciona o módulo de encadeamento de VNF para que uma operação de redirecionamento de tráfego seja executada, de forma a permitir que o tráfego passe pelo *firewall* instanciando na VNF-FW. Por conseguinte, o módulo de encadeamento de VNF insere a regra que mitiga o ataque no Módulo de segurança (3), configura o módulo de *proxy* da VNF-WAF e aciona o controlador SDN (4) para inserir fluxos OpenFlow de redirecionamento de tráfego no OVS ($\delta \rightarrow dst$) (5), com o propósito de desviar todo tráfego antes destinado de forma direta ao site web atacado (6) para a VNF-WAF (7).

O módulo de segurança da VNF-WAF foi construído com o ModSecurity [Ristic 2010], que é um *firewall* de aplicação web. Quando a VNF-WAF é encadeada ao fluxo de dados (7), todas as requisições HTTP destinadas ao site web vulnerável passam a ser tratadas primeiro pelo módulo de segurança e pelo módulo de *proxy* (Figura 2). O módulo de segurança remove o tráfego malicioso. O módulo de *proxy* é configurado pelo módulo de encadeamento de VNFs do controlador de segurança para atuar como "*proxy* reverso" [APACHE 2017], repassando o tráfego de rede benigno que recebe para os servidores que hospedam o site web vulnerável (8). Deste modo, a VNF-WAF remove o tráfego malicioso, sem afetar o tráfego benigno gerado a partir do mesmo endereço de origem.

5. Avaliação de Desempenho da Arquitetura Proposta

O desempenho da arquitetura proposta é avaliado através de dois experimentos. No primeiro experimento é avaliada a eficiência da detecção de ataques do módulo IDS.

Assim, é realizado uma simulação de ataques, na qual diversos tipos de ataque são realizados contra um servidor web vulnerável, de forma a verificar se os ataques praticados são detectados e se as requisições benignas continuam a ser atendidas. Em um segundo experimento, é verificado o impacto na capacidade do servidor atender às requisições HTML, ou o quanto a métrica taxa de respostas HTTP recebidas por segundo é afetada ao se utilizar a arquitetura proposta.

5.1. Capacidade da Proposta em Filtrar Ataques

De modo a avaliar o comportamento da arquitetura proposta, um cliente web envia requisições HTTP benignas e maliciosas para um servidor web. São observados quais ataques foram filtrados e quais requisições benignas passaram quando a arquitetura proposta é utilizada. Um pote de mel (*honeypot*) [Provos et al. 2004], que é uma ferramenta capaz de emular aplicações e sistemas operacionais vulneráveis, foi instalado para atuar como o servidor web. O principal propósito desse tipo de ferramenta é ser atacada e comprometida, para que novos tipos ou tendências de ataques possam ser identificados através da análise minuciosa das ações e do comportamento dos atacantes. Foram inseridas no pote de mel vulnerabilidades como XSS e injeção de SQL.

O cliente web foi configurado para enviar 220 requisições HTTP benignas e 340 requisições de ataque para o pote de mel. As requisições HTTP benignas foram enviadas com o objetivo de acessar a página principal do site web vulnerável, através do comando *WGET http://sitevulneravel/index.html*. As requisições HTTP maliciosas foram enviadas para explorar as vulnerabilidades configuradas no pote de mel. A VNF-WAF foi instalada sem regras previamente configuradas em seu módulo de segurança e nenhum *site* vulnerável foi previamente cadastrado para receber repasses de tráfego web do módulo de *p*roxy. As regras armazenadas na base de dados do Controlador de Segurança foram as necessárias para impedir todos os tipos de requisições maliciosas enviadas pelo cliente web.

A nuvem OPNFV é implantada em quatro servidores de prateleira: um nó de rede e três nós de computação. Todas as máquinas são instaladas com o sistema operacional Ubuntu 14.04.5 LTS. Nos nós de computação, a ferramenta de virtualização KVM, única opção suportada pelo OPNFV, foi utilizada. O nó de rede e um dos nós de computação da nuvem OPNFV são máquinas com processador Intel(R) Core(TM) i7-4770 CPU @ 3,40 GHz de oito núcleos, 32 GB de RAM e três interfaces Ethernet de 1 Gb/s. O segundo nó de computação possui processador Intel(R) Core(TM) i7-2660 CPU @ 3,40 GHz de oito núcleos, 32 GB de RAM e três interfaces Ethernet de 1 Gb/s. O terceiro e último nó de computação é uma máquina com dois processadores Intel(R) Xeon(R) CPU E5-2650 v2 @ 2,60 GHz de dezesseis núcleos, 32 GB de RAM e duas interfaces de 1 Gb/s.

No conjunto de nós de computação, foram criadas cinco máquinas virtuais (VMs) com acesso à Internet, uma para cada componente a seguir: Controlador de Segurança, cliente web, servidor web vulnerável, VNF-WAF e base de dados do Controlador de Segurança. Cada VM foi criada com duas CPUs virtuais e 4 GB de RAM. Por último, a VNF-IDS foi implantada numa quinta máquina com processador Intel(R) Core(TM)2 Quad CPU Q6600 @ 2,4 GHz, 4 GB de RAM e uma interface de 1 Gb/s.

Para realizar os ataques de exploração de vulnerabilidade foi desenvolvido um *script*, no qual são enviados os ataques e as requisições benignas em uma mesma ordem

Tabela 1. Exemplos de ataques detectados e não detectados.

Método	Requisição HTTP para explorar vulnerabilidade	Resposta
GET	http://siteA/?rHPbc8c=../../../../etc/passwd	200 OK
PUT	http://siteA/oRnQL com o arquivo:"9zseqh"	201 OK
GET	http://siteA/?XzuFzsw=SELECT TOP 1 name FROM sysusers	200 OK
GET	http://siteA/?rHPbc8c=ps -aux	403 Proibido

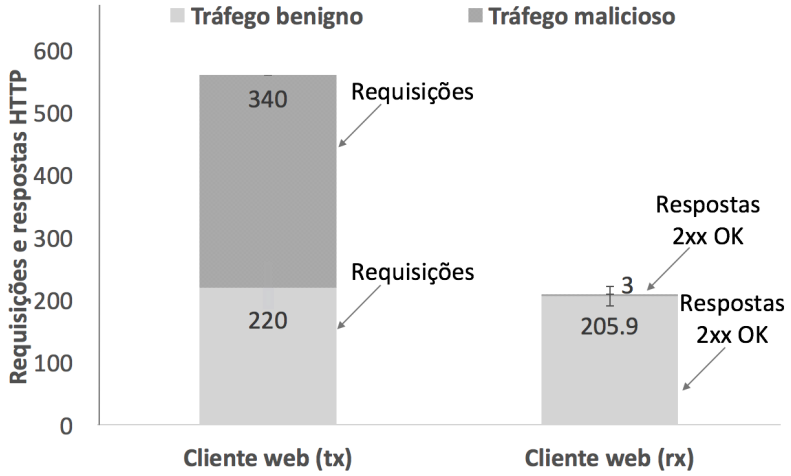
em todos os experimentos. As primeiras requisições de ataque enviadas pelo *script* em todos os experimentos estão apresentadas na Tabela 1. A primeira linha da tabela mostra que as contas de usuários existentes no sistema operacional do servidor web atacado são indevidamente listadas (200 OK) através de um GET HTTP. Na segunda linha, um ataque de inclusão remota de arquivo no servidor web atacado é realizado e comandos SQL arbitrários são enviados com sucesso para o banco de dados na terceira linha. A quarta linha mostra que uma mensagem HTTP 403 é enviada quando a arquitetura proposta bloqueia um ataque de injeção arbitrária de comandos no sistema operacional do servidor.

Em uma primeira rodada de teste, foram enviados os 340 ataques e as 220 requisições benignas para o servidor web vulnerável sem habilitar os módulos da arquitetura NFV proposta. A quantidade de respostas 2xx OK (Tabela 1) foi igual a de requisições enviadas, conforme esperado, pois não há detecção de ataques. Há necessidade de execução do teste várias vezes em função do não determinismo relacionado à remoção e à instalação de novos fluxos nos comutadores SDN. Por esse motivo, o *script* envia as requisições em dez rodadas do experimento, com os módulos da arquitetura NFV proposta habilitados. Todos os resultados apresentados são médias com intervalo de confiança de 95%. Alguns intervalos de confiança não são visualizados, por serem muito pequenos.

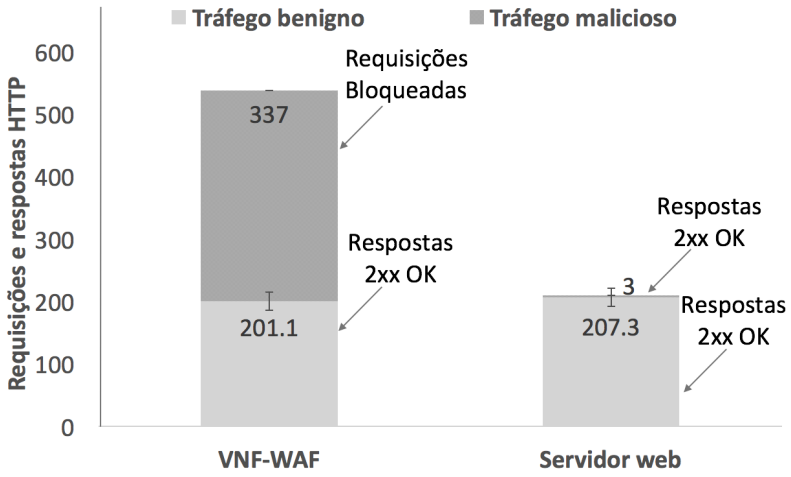
Pode-se observar pela Figura 3(a) que dos 340 ataques transmitidos pelo cliente web, apenas três alcançaram o servidor web e foram bem sucedidos (respostas 2xx OK recebidas pelo cliente web na figura). Todos os outros ataques foram detectados e bloqueados. Quando os primeiros ataques foram enviados, a VNF-IDS identificou as requisições maliciosas e notificou o Controlador de Segurança. O Controlador de Segurança automaticamente desviou o tráfego para a VNF-WAF aplicando fluxos OpenFlow no OVS, inseriu as regras de bloqueio na VNF-WAF e configurou seu Módulo de proxy. As três respostas HTTP positivas para os ataques (Figura 3(b)) foram enviadas pelo servidor web enquanto a VNF-WAF ainda não havia sido automaticamente encadeada.

Os resultados indicam que existem perdas na rede quando o Controlador de Segurança remove do comutador OVS os fluxos que permitiam a comunicação entre o cliente web e o servidor vulnerável. Esses fluxos são substituídos por outros que direcionam o tráfego para a VNF-WAF. Durante esta operação, acontecem perdas de pacotes, semelhantes às ocorridas quando um roteador de uma rede de comutação de pacotes falha. Por esse motivo, a quantidade de respostas HTTP benignas recebidas no cliente web, que em média é igual a 205,9, é menor do que as 220 requisições por ele transmitidas (ver Figura 3(a)). No entanto, 207,3 das 220 requisições benignas transmitidas alcançaram o servidor sem problemas. Portanto, a quantidade de requisições HTTP perdidas entre o cliente e o servidor vulnerável é em média igual a 12,7.

Ainda nessa linha, os resultados também mostram que 201,1 das 207,3 respostas



(a) Quantidade de requisições transmitidas e de respostas recebidas de ataques e de tráfego benigno pelo cliente.



(b) Quantidade de ataques que alcançaram o servidor web, bloqueados na VNF-WAF e quantidade de respostas enviadas pelo servidor web e que passaram pela VNF-WAF para o tráfego benigno.

Figura 3. Ataques filtrados quando a arquitetura NFV proposta é utilizada.

HTTP enviadas pelo servidor, dentro da média observada, foram respondidas quando a VNF-WAF já estava encadeada ao fluxo de dados (ver Respostas 2xx OK benignas na Figura 3(b)). Por conseguinte, aproximadamente seis respostas HTTP benignas voltaram do servidor para o cliente web sem passar pela VNF-WAF. Além disso, é possível perceber, que em média 205,9 dessas 207,3 respostas HTTP chegaram no cliente web. Portanto, a quantidade de respostas HTTP benignas perdidas entre o servidor e o cliente web é em média igual a 1,4. Por fim, a Figura 3(b) ainda mostra que, em média, depois do encadeamento da VNF, todos os 337 ataques realizados foram bloqueados.

Os resultados mostram que a arquitetura NFV proposta é eficiente, pois 99,12% dos ataques gerados foram dinamicamente bloqueados e 93,59% do tráfego benigno gerado não foi afetado quando a VNF-WAF foi dinamicamente encadeada ao fluxo de dados.

5.2. Impacto da Proposta no Desempenho da Aplicação Web

O segundo experimento consiste em variar a quantidade de requisições HTTP por segundo enviadas para o servidor web. O objetivo é verificar de que forma o encadeamento da VNF-WAF afeta a taxa de requisições e respostas HTTP por segundo. O desempenho da VNF-WAF foi avaliado no segundo experimento, quando todas as regras, adequadas para mitigar os ataques gerados pelo *script* de teste, estavam aplicadas na VNF.

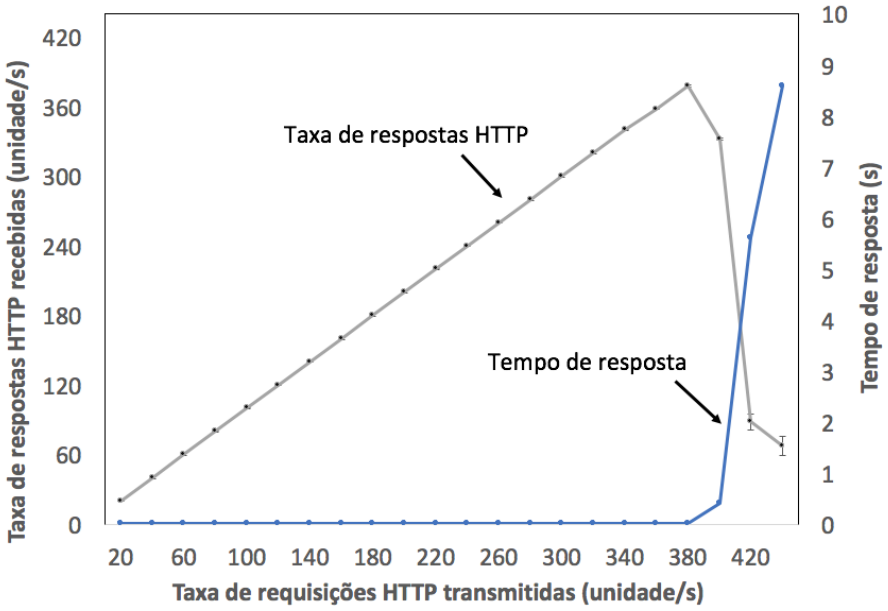


Figura 4. Desempenho da aplicação web quando a VNF-WAF não está encadeada ao fluxo de dados.

O httperf foi utilizado para gerar as requisições HTTP. Esse aplicativo permite gerar e manter taxas sustentáveis de requisições HTTP de forma a sobrecarregar um servidor. Nele é possível definir a taxa de conexões TCP por segundo, quantas requisições HTTP devem ser realizadas em cada conexão e o número máximo de requisições HTTP que o cliente deve realizar. Também é possível aumentar a taxa de conexões TCP por segundo durante um teste, definindo uma taxa inicial, o valor do incremento e a taxa máxima de conexões TCP [Mosberger and Jin 1998].

De forma a automatizar o uso do httperf, o Autobench foi utilizado. O Autobench foi configurado para variar a taxa de conexões TCP entre 10 e 220, com incremento de 10 em 10 conexões TCP por segundo [AUTOBENCH 2017]. Foram enviadas duas requisições HTTP persistentes em cada conexão TCP estabelecida com o servidor web. Portanto, a quantidade de requisições HTTP requisitadas variou entre 20 e um valor máximo de 440. Estes valores foram utilizados porque, em diversos testes preliminares, observou-se que este é um limite para postergar ao máximo o aumento significativo do tempo de resposta das transações HTTP, que é o tempo decorrido entre a requisição de um objeto e o recebimento do mesmo. Os resultados apresentados a seguir são médias com intervalo de confiança de 95%. Alguns intervalos de confiança não são visualizados nas figuras, por serem muito pequenos.

Na Figura 4 pode-se observar que a taxa máxima de requisições HTTP suportada

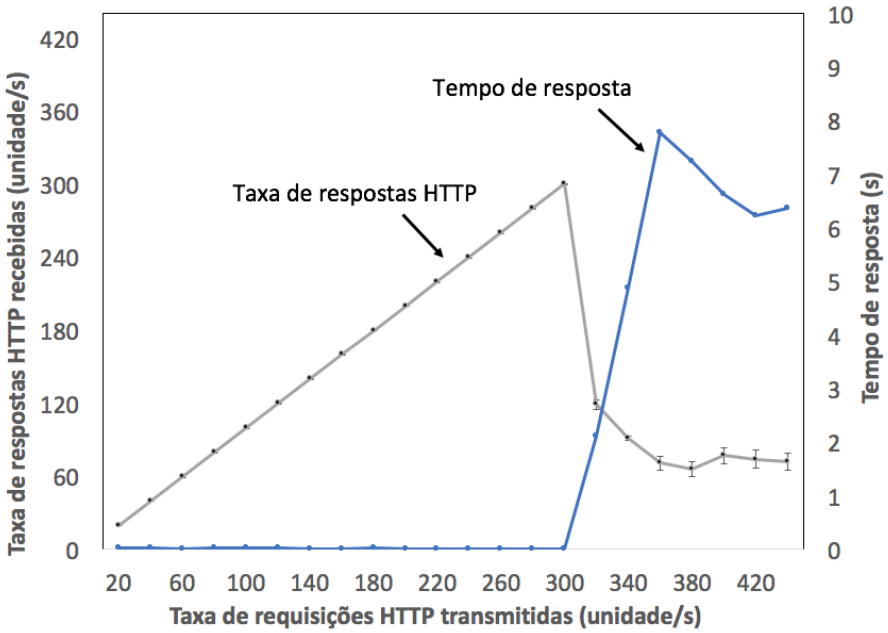


Figura 5. Desempenho da aplicação web quando a VNF-WAF está encadeada ao fluxo de dados.

pelo servidor web, sem que ocorra aumento significativo no tempo de resposta, é igual a 380 requisições por segundo. A partir deste valor, o tempo de resposta aumenta consideravelmente, saindo de aproximadamente zero para cerca de 9 s quando o cliente tenta enviar 420 requisições HTTP por segundo para o servidor.

A Figura 5 mostra que o encadeamento da VNF-WAF fez com que a capacidade de o servidor responder sem o tempo de resposta aumentar consideravelmente caísse para cerca de 300 requisições HTTP por segundo. Essa redução da capacidade foi causada pela VNF-WAF, em função das regras de ataque instanciadas. A partir desta quantidade de requisições, o tempo de resposta subiu consideravelmente, chegando a alcançar cerca de 8 s. Portanto, o encadeamento da VNF-WAF reduziu em cerca de 21% a taxa de conexão e de requisições HTTP por segundo atendidas. Entretanto, Mauricio *et al.* mostraram que é possível utilizar a escalabilidade da computação em nuvem quando é necessário escalar funções de rede virtuais criadas para atuarem como *firewalls* [Mauricio et al. 2016]. É possível escalar a VNF de forma vertical, aumentando os recursos de memória, CPU ou disco da VNF-WAF em função da demanda, ou de forma horizontal, quando o número de VNFs da arquitetura é aumentado.

6. Conclusão

A arquitetura NFV proposta mostrou-se capaz de aplicar contramedidas de forma dinâmica contra ataques. A função virtual de rede de sistema de detecção de intrusão notificou corretamente o Controlador de Segurança e seus módulos de classificação e de encadeamento conseguiram inserir corretamente o fluxo OpenFlow OFPT_FLOW_MOD no comutador SDN para desviar o tráfego para o *firewall* VNF-WAF, onde as regras de bloqueio para os ataques de exploração de vulnerabilidade foram corretamente instaladas. A arquitetura apresentou elevada eficiência, pois bloqueou 99,12% dos ataques de exploração de vulnerabilidade gerados no ambiente de teste, sem afetar consideravelmente o

tráfego benigno gerado pelo mesmo IP de origem, que continuou alcançando o servidor web. A VNF-WAF do ambiente de teste reduziu a quantidade de conexões por segundo que o servidor web consegue atender em cerca de 21%. Contudo, a escalabilidade da computação em nuvem permite que a VNF-WAF seja vertical ou horizontalmente escalada em função da demanda de tráfego. Comparada a uma solução com equipamentos tradicionais, a arquitetura NFV possui ainda outras vantagens como menor custo e maior adaptabilidade à quantidade de tráfego de rede.

Como trabalhos futuros, deseja-se estender a arquitetura NFV para implementar um módulo de varredura dinâmica de vulnerabilidades. Dessa forma, será possível identificar possíveis vulnerabilidades nas aplicações e aplicar as contramedidas descritas assim que sejam identificadas.

7. Agradecimentos

Este trabalho foi realizado com recursos da Globo (www.globo.com), INCT Internet do Futuro, CNPq, CAPES e FAPERJ.

Referências

Akamai (2017). State of the Internet - Connectivity report - Additional resources. Technical report. https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/q4-2016-state-of-the-internet-security-report.pdf.

Amann, J. and Sommer, R. (2015). Providing dynamic control to passive network security monitoring. In *International Workshop on Recent Advances in Intrusion Detection*, pages 133–152.

Anderson, R., Barton, C., Böhme, R., Clayton, R., Eeten, M. J. V., Levi, M., Moore, T., and Savage, S. (2013). *The Economics of Information Security and Privacy*, chapter Measuring the Cost of Cybercrime, pages 265–300. ISBN 978-3-642-39497-3. Springer Berlin Heidelberg.

Andreoni Lopez, M., Figueiredo, U. R., Lobato, A., and Duarte, O. C. M. B. (2014). Broflow: Um sistema eficiente de detecção e prevenção de intrusão em redes definidas por software. In *Workshop em Desempenho de Sistemas Computacionais e de Comunicação (WPerfomance)*, pages 1919–1932.

APACHE (2017). Apache http server: Reverse proxy guide. https://httpd.apache.org/docs/2.4/howto/reverse_proxy.html. Acessado 07-02-2017.

AUTOBENCH (2017). Autobench: An http benchmarking suite. https://github.com/menavaur/Autobench. Acessado 17-02-2017.

Deng, J., Hu, H., Li, H., Pan, Z., Wang, K.-C., Ahn, G.-J., Bi, J., and Park, Y. (2015). VNGuard: An NFV/SDN combination framework for provisioning and managing virtual firewalls. In *IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*, pages 107–114.

ETSI (2012). Network functions virtualisation: An introduction, benefits, enablers, challenges and call for action. Technical report.

ETSI (2013). Network functions virtualisation (NFV); architectural framework. ETSI GS NFV 002 V1.1.1.

Gu, G., Perdisci, R., Zhang, J., Lee, W., et al. (2008). Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection. In *USENIX Security Symposium*, pages 139–154.

Haggard, S. and Lindsay, J. R. (2015). North Korea and the Sony hack: Exporting instability through cyberspace. *AsiaPacific Issue*, 117:1–8.

Lin, Y.-D., Lin, P.-C., Yeh, C.-H., Wang, Y.-C., and Lai, Y.-C. (2015). An extended SDN architecture for network function virtualization with a case study on intrusion prevention. *IEEE Network*, 29(3):48–53.

Martins, J., Ahmed, M., Raiciu, C., Olteanu, V., Honda, M., Bifulco, R., and Huici, F. (2014). ClickOS and the art of network function virtualization. In *USENIX Conference on Networked Systems Design and Implementation*, pages 459–473.

Mauricio, L. A. F., Rubinstein, M. G., and Duarte, O. C. M. B. (2016). Proposing and evaluating the performance of a firewall implemented as a virtualized network function. In *International Conference on Network of the Future (NOF)*, pages 1–3.

Moraes, I. M., Mattos, D. M. F., Ferraz, L. H. G., Campista, M. E. M., Rubinstein, M. G., Costa, L. H. M. K., de Amorim, M. D., Velloso, P. B., Duarte, O. C. M. B., and Pujolle, G. (2014). FITS: A flexible virtual network testbed architecture. *Computer Networks*, 63:221–237.

Mosberger, D. and Jin, T. (1998). Httperf — a tool for measuring web server performance. *Performance Evaluation Review*, 26(3):31–37.

Mtibaa, A., Harras, K. A., and Alnuweiri, H. (2015). From botnets to mobibots: A novel malicious communication paradigm for mobile botnets. *IEEE Communications Magazine*, 53(8):61–67.

ODL (2017). Opendaylight: Open source SDN platform. https://www.opendaylight.org/. Acessado 01-02-2017.

ONOS (2017). ONOS: Open network operating system. http://onosproject.org/. Acessado 01-02-2017.

OPENSTACK (2017). Open source software for creating private and public clouds. https://www.openstack.org/. Acessado 27-01-2017.

OPNFV (2017). Open platform for NFV. https://www.opnfv.org/. Acessado 24-01-2017.

Provos, N. et al. (2004). A virtual honeypot framework. In *USENIX Security Symposium*, pages 1–14.

Ristic, I. (2010). *ModSecurity Handbook: The Complete Guide to the Popular Open Source Web Application Firewall*. Feisty Duck, 1^{st} edition. ISBN 978-1907117022.

Sommer, R. (2003). Bro: An open source network intrusion detection system. In *DFN-Arbeitstagung über Kommunikationsnetze*, pages 273–288.

Zanna, P., O'Neill, B., Radcliffe, P., Hosseini, S., and Hoque, M. S. U. (2014). Adaptive threat management through the integration of IDS into software defined networks. In *International Conference on the Network of the Future (NOF) - Workshop on Smart Cloud Networks & Systems*, pages 1–5.

Um Serviço SDN para Detecção e Solução de Problemas em Redes Domésticas

Alisson R. Alves, Henrique D. Moura, Luis H. C. Reis, Julio C. T. Guimarães
Jonas R. A. Borges, Philippe S. Silva, Daniel F. Macedo, Marcos A. M. Vieira

[1]Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte – MG – Brazil

{alissonralves, henriquemoura, luiscantelli}@dcc.ufmg.br

{julio.guimaraes, jonasrafael, phss, damacedo, mmviera}@dcc.ufmg.br

Abstract. *The number of smart devices in home networks is rapidly increasing, making it more complex to manage problems. In addition, the lack of customer knowledge and the lack of tools to automatically diagnose and fix problems aggravate the problem. In this paper, we propose a network service, based on software defined networks, for the detection and automatic solution of problems in home networks. We evaluate the service in a prototype, considering throughput, jitter, elapsed time to fix the problem, among other metrics. The results show that our service can provide increase throughput by 7%, reduction in wireless transmission delays, and reduction of human intervention in troubleshooting.*

Resumo. *As redes domésticas têm recebido quantidades cada vez maiores de dispositivos inteligentes. O resultado disso é um aumento na complexidade de gerenciamento de problemas nessas redes. Além disso, contribuem também a falta de conhecimento dos clientes e a falta de ferramentas que diagnostiquem e corrijam automaticamente tais problemas. Neste artigo, é proposto um serviço de rede para a detecção e a solução automática de problemas em redes domésticas, baseado em redes definidas por software. O serviço foi avaliado em um protótipo, considerando vazão, jitter, tempo para solução da falha, entre outras métricas. Os resultados comprovam que o serviço é capaz de proporcionar aumento na vazão em 7%, redução em atrasos de transmissão sem fio e redução da intervenção humana ao solucionar problemas.*

1. Introdução

As redes domésticas têm se tornado cada vez maiores em função da quantidade de dispositivos inteligentes conectados [Cisco 2015, Perera et al. 2014]. Dentre os dispositivos pessoais inseridos nesse tipo de rede estão *smartphones, tablets*, *notebooks, smart TVs* e dispositivos de Internet das Coisas (IoT). Resultado disso, é a densificação dessas redes. Além disso, novos serviços como *streaming*, armazenamento de dados na nuvem, dentre outros, têm se tornado bastante populares entre seus usuários [Bouchet et al. 2014]. A partir disso, surge a necessidade de maior confiabilidade e qualidade de serviços (QoS), o que torna o gerenciamento de redes domésticas ainda mais complexo.

Outros fatores contribuem no aumento da complexidade de gerenciamento de redes domésticas. Como exemplo, tem-se o volume elevado de conexões móveis e a diversidade de padrões de comunicação entre dispositivos. O aumento no tráfego em redes,

cabeadas ou sem fio, bem como a diversidade de problemas ou falhas que podem ocorrer na rede, também contribuem para maior complexidade [Yiakoumis et al. 2011]. Em geral, quando ocorre um problema, o usuário de redes domésticas não é capaz de inspecioná-lo ou solucioná-lo. Muitas vezes tais usuários são os responsáveis por configurá-las [Dong and Dulay 2011]. Essa dificuldade é agravada pela falta de ferramental que apoie a identificação precisa dos motivos que geraram um problema, assim como a solução automática deste [Fratczak et al. 2013]. Neste trabalho, considera-se problemas como baixo desempenho, falhas, ataques à rede ou eventos que possam ser representados por uma condição que tenha uma ou mais métricas de rede associadas.

O diagnóstico e a correção automática de problemas em redes domésticas são benéficos tanto para os usuários quanto para as provedoras de acesso. Para os usuários, os benefícios são uma rede mais estável e confiável, aumentando a sua satisfação. Para as provedoras, a automatização permite uma redução no seu custo operacional, pois exige uma demanda menor de serviços de manutenção, que normalmente não são faturados aos clientes. Além disso, ferramentas que facilitam o diagnóstico da causa do problema permitem um menor tempo para a sua recuperação, aumentando a produtividade da equipe de operação. Ainda, do ponto de vista de regulação, uma rede mais robusta é importante para alcançar as metas de qualidade de rede e disponibilidade que são definidos pelos órgãos reguladores, que muitas vezes podem aplicar multas em caso de descumprimento dessas metas.

Apesar de diversos trabalhos na literatura tratarem de problemas e de falhas em redes domésticas [Gheorghe et al. 2015, Kim et al. 2014b, Biswas et al. 2015, DiCioccio et al. 2012], poucos utilizam o paradigma de redes definidas por software para lidar também com redes sem fio. Em complemento, poucos solucionam automaticamente problemas na rede ou permitem adicionar novas regras de monitoramento e diagnosticar novos tipos de dispositivos de rede. Além disso, alguns trabalhos apresentam limitações em termos de flexibilidade por dependerem de plataformas proprietárias em suas soluções.

Desta maneira, neste artigo é proposto o *HomeNetRescue* (HNR), um serviço para o gerenciamento autônomo de redes domésticas voltado para a detecção, o diagnóstico e a solução automática ou minimização de problemas, que também pode atuar em aspectos de desempenho, baseado em redes definidas por software (SDN) [Guedes et al. 2012, Macedo et al. 2015]. Uma contribuição do artigo encontra-se na modelagem do protótipo do serviço para o gerenciamento autônomo que utiliza o paradigma SDN em redes sem fio. Além disso, a utilização do *HNR* é capaz de proporcionar benefícios como redução da intervenção humana ao solucionar falhas, maior confiabilidade de rede, aumento na vazão em 7%, redução em atrasos de transmissão sem fio e melhor qualidade de experiência.

As seções deste artigo estão organizadas como segue. Na Seção 2, os trabalhos relacionados são discutidos. Na Seção 3, é realizada a descrição do *HomeNetRescue* juntamente com a arquitetura Ethanol. Na Seção 4, as avaliações e os resultados obtidos são descritos. Por fim, na Seção 5, são discutidas as conclusões e os trabalhos futuros.

2. Trabalhos Relacionados

A literatura apresenta diversas ferramentas e plataformas para o diagnóstico e a solução de falhas. Tais ferramentas se diferem nos tipos de redes e equipamentos que suportam, no escopo de problemas tratadas, nas abordagens (distribuídas ou centralizadas),

se somente detectam ou também solucionam falhas, entre outras características. Deste modo, na Tabela 1, apresentamos uma comparação entre nossa proposta e outras propostas da literatura. Posteriormente, descrevemos em síntese cada trabalho e ao final da seção, discutimos a respeito das principais diferenças. Na Tabela 1, as colunas identificam as seguintes propriedades: soluções focadas em redes domésticas (RD); uso de técnicas para solucionar falhas (*troubleshooting*) (TS); emprego do paradigma SDN (SDN); permitem o gerenciamento local de rede para a detecção de falhas (GL); fornecem recursos para a detecção automática de falhas (DA); suportam redes ethernet (ET); suportam redes wireless (WI); identificam falhas nas estações dos usuários (FE); e possibilitam a adição de novas regras de monitoramento e novos tipos de dispositivos, ou seja, extensibilidade (EX). A seguir apresentamos as propostas mais relevantes.

Tabela 1. Comparação entre trabalhos na literatura.

Referência	RD	TS	SDN	GL	DA	ET	WI	FE	EX
[Kim et al. 2014b]	✓	✓	✗	✗	✓	✓	✓	✓	✓
[Biswas et al. 2015]	✓	✓	✗	✗	✗	✓	✓	✓	✗
[DiCioccio et al. 2012]	✓	✗	✗	✗	✗	✓	✓	✓	✗
[Gheorghe et al. 2015]	✗	✓	✓	✗	✗	✓	✗	✗	✗
[Sundaresan et al. 2013]	✓	✗	✗	✗	✗	✓	✓	✓	✓
[Kim et al. 2014a]	✓	✓	✗	✓	✗	✗	✓	✓	✓
Serviço proposto	✓	✓	✓	✓	✓	✓	✓	✓	✓

DYSWIS é *framework* P2P colaborativo, com uma arquitetura centralizada, para detecção e diagnóstico automático de falhas em redes de usuários finais, através do monitoramento de pacotes e relatórios de falhas de aplicações [Kim et al. 2014b]. As regras para a detecção de falhas são baseadas em consultas e sondagens distribuídas e recorrem, ainda, à colaboração de usuários para distinção de falhas na rede, bem como a identificação do ponto de origem da falha. Outra arquitetura, a *Meraki* [Biswas et al. 2015] da Cisco, realiza o gerenciamento de redes empregando um sistema em nuvem. A arquitetura fornece configuração centralizada, monitoramento e ferramentas de resolução de problemas em redes cabeadas e sem fio. É uma arquitetura *backend* composta de APs, comutadores e *firewalls* em residências ou empresas, que monitoram métricas de redes e as repassam a um sistema de gerenciamento central.

O *HomeNet Profiler* é uma aplicação cliente-servidor executada em estações clientes [DiCioccio et al. 2012]. Nesta proposta o usuário deve executar uma aplicação e, posteriormente, enviar os dados coletados a um servidor para análise. A aplicação monitora redes domésticas cabeadas e sem fio. As informações mensuradas incluem configurações de rede, desempenho, dispositivos ativos e serviços em execução, realizadas via protocolos ZeroConf [1] e UPnP [2]. Já *SDN-RADAR* consiste em uma abordagem distribuída para o monitoramento da infraestrutura de rede e a localização de problemas ou falhas de desempenho em redes de grande porte [Gheorghe et al. 2015]. Recursos SDN são utilizados no processo de identificação de enlaces de baixo desempenho mais prováveis em rede cabeadas. A ferramenta auxilia os administradores de rede a entenderem prováveis falhas de enlaces de rede, bem como a monitorar e depurar as falhas que afetam os serviços de entrega nas redes dos usuários.

[1] http://www.zeroconf.org/

[2] https://tools.ietf.org/html/rfc6970

Where's The Fault? é uma ferramenta executada em roteadores domésticos para detectar problemas de desempenho em redes domésticas cabeadas e sem fio [Sundaresan et al. 2013]. Informações de *timing* e *buffering* de rede são obtidas pelo monitoramento passivo do tráfego dos roteadores. Por fim, *Why is my Wi-Fi Slow?* (*Wi-Slow*) diagnostica o desempenho do sinal em redes Wi-Fi (abrangendo redes domésticas), utilizando sondagens em nível de usuário [Kim et al. 2014a]. Esta ferramenta auxilia na localização física e na identificação de causas que impactam o desempenho sem fio da rede via análise da perda de pacotes e do número de ACKs recebidos na rede.

Dentre os trabalhos citados, alguns propõem sistemas colaborativos para a solução de falhas, o que pode acarretar em problemas de privacidade em relação a informações de usuários da rede. Neste sentido, nossa proposta não armazena ou exibe dados a terceiros, respeitando a privacidade e a segurança dos dados dos usuários. Alguns dos trabalhos mencionados são pouco flexíveis, pois dependem de tecnologias proprietárias, consequentemente a adição de parâmetros de monitoramento e novos tipos de serviços ficam limitados aos fabricante dos componentes adotados nas redes. Por sua vez, em nossa solução, o paradigma SDN em conjunto com plataformas Linux embarcadas possibilita a implementação de recursos de controle da rede de forma mais flexível.

Nossa proposta também apresenta um diferencial ao analisar as falhas em todo o escopo de uma rede doméstica (APs, roteadores, estações). O *HomeNetRescue* pode solucionar ou minimizar automaticamente as falhas detectadas, diferentemente da maioria das abordagens apresentadas. Embora o *HomeNetRescue* apresente variadas funcionalidades, a solução herda problemas característicos de arquiteturas centralizadas. Todavia, devido a técnicas de alta disponibilidade para controladores serem um problema resolvido na literatura, o controlador não é o principal ponto de falha.

A visão global do controlador simplifica o gerenciamento da rede, possibilita obter soluções mais eficientes e controlar a rede com um grão mais fino, algo primordial para o processo de identificação e solução de falhas. Através dessa abordagem o serviço pode monitorar todos os componentes da rede em busca do seu funcionamento adequado. Ainda, tal visão possibilita às provedoras identificarem falhas nos dispositivos dos clientes, algo que atualmente depende de visita de técnicos ao domicílio. O resultado disto são reduções de custos para as provedoras (por não precisarem alocar profissionais em casos desnecessários) e para os usuários (dispensando a contratação de serviços adicionais de suporte). Outro benefício do serviço consiste em proporcionar maior índice de satisfação do cliente (QoE), pois com ele os problemas tornam-se passíveis de serem diagnosticados e resolvidos em menor tempo e automaticamente.

3. O Serviço *HomeNetRescue*

Nessa seção, é proposta e apresentada a arquitetura do *HomeNetRescue* (HNR), um serviço para o gerenciamento autônomo de redes domésticas voltado para a detecção, o diagnóstico e a solução automática ou minimização de falhas. Adicionalmente, o serviço também pode atuar em aspectos de desempenho. O HomeNetRescue baseia-se no paradigma SDN para o gerenciamento de redes sem fio e é composto por uma arquitetura modular e expansível, podendo ser empregado na resolução de diversos problemas desses ambientes. Destaca-se que as plataformas SDN são mais flexíveis do que as não SDN e isto permite ao HNR o controle com um grão mais fino no processo de gerenci-

amento. Além disso, SDN para o gerenciamento de redes sem fio é um tema em ampla investigação pela comunidade científica [Guedes et al. 2012].

O *HomeNetRescue* foi desenvolvido para ser executado sob demanda ou de forma agendada. Ainda, foi concebido para poder ser administrado por provedoras de acesso à internet, podendo ser fornecido aos seus clientes de forma gratuita ou paga. O *HomeNetRescue* pode ser executado a partir de *call centers*, quando necessário realizar diagnósticos ou soluções de problemas nas redes, mediante chamadas de suporte, ou ser executado programadamente (permanecendo em execução automática).

3.1. Arquitetura

A arquitetura do *HomeNetRescue* é composta por planos, camadas e módulos. Em seu planejamento foi considerado tratar eventos de falhas ou problemas a partir da camada de enlace até a camada de aplicação do modelo TCP/IP. Nesta, são apresentados os módulos de software juntamente com suas respectivas funcionalidades. Destaca-se também os componentes de hardware de uma rede doméstica. Considera-se uma rede doméstica composta por APs, roteadores IP, *switches* e estações clientes com e sem fio.

A arquitetura foi dividida em três planos, conforme apresentado na Figura 1. Estes planos são: $i)$ **plano de gerenciamento de problemas**: plano extensível onde são executados os algoritmos de detecção e gerenciamento de problemas. $ii)$ **plano de controle**: corresponde a um software (controlador) que pode ser executado de forma distribuída ou centralizada em um ou mais componentes da rede (ex: desktops, roteadores, entre outros); e $iii)$ **plano de dados**: responsável pelas funções de encaminhamento ou roteamento realizadas nos dispositivos de rede a serem gerenciados pelo serviço. A seguir são descritos mais detalhes sobre cada plano.

3.1.1. Plano de Gerenciamento de Problemas

Este plano permite ao administrador da rede configurar as regras, as políticas e os parâmetros de gerenciamento do *HomeNetRescue*. É composto por uma ou mais aplicações. Estas aplicações são aplicações padrão do *HomeNetRescue* ou aplicações implementadas pelo gerenciador da rede. Elas fornecem as funcionalidades de detecção e solução de problemas. Cada aplicação possui dois módulos, diagnóstico e atuador.

O módulo **Diagnóstico** permite definir políticas de gerenciamento, registro de métricas a serem monitoradas, seu intervalo de monitoramento, prioridades de aplicações e alterações de configurações nos componentes de rede. Este módulo, assim como o módulo **Atuador**, é programável, porém não é vinculado ao módulo **Eventos**, com isso a implementação de inteligência das aplicações inseridas na arquitetura é flexibilizada. Este módulo comunica-se com o módulo monitor. Como demonstração dos módulos, uma aplicação para a detecção de problemas no sinal de transmissão sem fio é apresentada. Tal aplicação pode registrar o monitoramento, a cada segundo, das métricas *Signal-to-Noise-Ratio* (SNR) e porcentagem de pacotes perdidos em um roteador da rede. Com isso, a aplicação registra no plano de controle que, caso o SNR atinja o limiar de 10 dbm ou a porcentagem de pacotes perdidos seja maior do que 10%, o plano de controle deve acionar o Atuador da aplicação com a prioridade predominante.

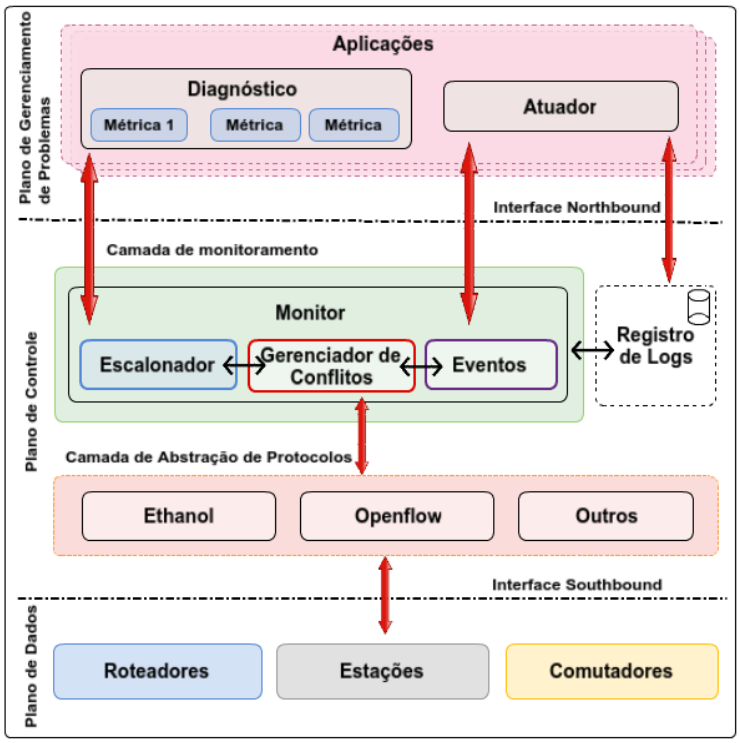


Figura 1. Arquitetura do Serviço

O módulo **Atuador** é invocado para atuar na rede quando limiares pré-definidos são atingidos. Assim, o **Atuador** pode executar funções que alterem parâmetros nos componentes de rede objetivando solucionar um problema detectado ou melhorar o desempenho. Nesse módulo, os parâmetros oriundos dos protocolos da camada de abstração são aqueles que podem ser configurados. Seguindo o exemplo anterior, o módulo **Atuador**, ao ser invocado, pode aumentar a potência de transmissão do AP afetado ou mudar o canal de transmissão, solicitar que as estações reassociem a outro AP, entre outros.

3.1.2. Plano de Controle

O plano de controle fornece uma visão global da rede e facilita a programação de aplicações de gerenciamento de problemas. O plano foi dividido em camada de monitoramento e camada de abstração de protocolos. A primeira camada realiza o monitoramento da rede, podendo escalonar aplicações conforme prioridades, gerenciar conflitos em ações de configuração, bem como disparar eventos de acordo com uma dada ocorrência. A segunda camada possibilita lidar com os protocolos de rede, a saber: Ethanol (descrito posteriormente), Openflow [3], Netconf [4], SNMP [5], entre outros. Mais informações sobre as camadas desse plano são apresentadas a seguir:

- **Monitor:** realiza o controle e o monitoramento da rede. Através dele são realizadas as coletas de dados da rede que permitem o serviço detectar e solucionar

[3]https://www.opennetworking.org/ja/sdn-resources-ja/onf-specifications/openflow
[4]https://tools.ietf.org/html/rfc6241
[5]https://www.ietf.org/rfc/rfc1157.txt

um problema. Nele, realizada-se a agregação dos dados, o processamento de métricas e a consolidação do parâmetro de tempo de monitoramento informado na aplicação. Contidos nesse módulo estão três sub-módulos:
`Escalonador:` escalona as aplicações do serviço. Dadas as aplicações, realiza solicitações periódicas às camadas inferiores (por exemplo, informações de SNR a cada minuto). Os dados coletados são repassados ao módulo de **eventos** e armazenados em **Registro de Logs**. O escalonador auxilia na otimização da coleta de dados originados de aplicações distintas. Por exemplo, se uma aplicação solicita o SNR a cada 100 ms e outra a cada 200 ms, o escalonador evita o disparo duplo de solicitações de SNR oriundos das aplicações distintas.
`Gerenciador de Conflitos:` verifica se as ações enviadas por diferentes aplicações podem causar comportamento conflitante nos dispositivos ou nos protocolos utilizados na interface *Southbound*. Por exemplo, uma aplicação pode solicitar o aumento da potência de transmissão para reduzir a taxa de perda, enquanto outra solicita a redução da potência de transmissão para reduzir a interferência entre APs. O módulo de gerenciamento de conflitos resolve tais situações em função da prioridade de cada uma. Aplicações de maior prioridade sobrescrevem as ações das aplicações de menor prioridade.
`Eventos:` analisa as medições periodicamente e, em caso de evento de rede, dispara o evento para as aplicações (módulo **atuador**) com as regras previamente inseridas. Um exemplo de evento seria a taxa de perda ser superior a 10%.

- **Registro de Logs:** registra os logs das operações executadas pelo serviço e das leituras realizadas, permitindo auditorias aos administradores das redes.
- **Camada de Abstração de Protocolos**: representa os protocolos suportados pela interface *Southbound* do serviço. O serviço foi projetado para suportar múltiplos protocolos. Nesta camada podem ser utilizados protocolos como Openflow, NetConf, SNMP, Ethanol, entre outros.

O Ethanol consiste em uma arquitetura SDN para o gerenciamento de redes IEEE 802.11 (WLAN) empresariais e domésticas [Moura et al. 2015]. É definida uma interface *Southbound* que permite o controle de pontos de acesso IEEE 802.11 compatíveis, bem como estações sem fio que implementem alguns padrões IEEE 802.11. Através destas características, a arquitetura fornece recursos para controle de *handoff* de estações entre os roteadores, controle de autenticação de usuários, criação de redes virtuais, configuração de QoS, localização de usuários na rede, entre outras. Assim como no Openflow, são utilizadas conexões seguras (Socket SSL) para a comunicação do controlador com os clientes e APs.

3.1.3. Plano de Dados

O plano de dados é composto pelos dispositivos de encaminhamento ou roteamento da rede que suportam pelo menos um dos protocolos ativos no *HomeNetRescue*. Neste artigo, são empregados planos de dados compatíveis com os protocolos OpenFlow e Ethanol. Assim, o *HomeNetRescue* suporta switches SDN (via OpenFlow), bem como APs IEEE 802.11 (via Ethanol) e estações que implementam recursos de gerenciamento do protocolo IEEE 802.11/2012 (também via Ethanol). Nos **APs** IEEE 802.11, o *HomeNetRescue* pode realizar operações como mudar a frequência e canais de operação, mo-

dificar a potência de transmissão, os parâmetros do protocolo MAC (uso de RTS/CTM, DTIM, etc), solicitar varreduras do espectro, gerenciar parâmetros de QoS, controlar a associação de estações, entre outras operações.

Nas **estações**, devido ao suporte aos padrões IEEE 802.11 mencionados, é possível receber informações sobre a interface de rede sem fio, solicitar varreduras do espectro, ajustar parâmetros da camada física tais como a potência de transmissão, alterar o *bitrate*, realizar a troca para outra rede WiFi, entre outros ajustes. Vale ressaltar que na versão atual do Ethanol as estações devem executar um software que implementa os padrões de gerenciamento do IEEE 802.11, pois o kernel do Linux ainda não possui suporte para eles. Entretanto, tal software será desnecessário quando o suporte aos padrões for nativo.

3.2. Aplicabilidade da Solução

O *HNR* foi desenvolvido para permitir o gerenciamento autônomo de problemas e de falhas em redes domésticas. Conforme mencionado na Seção 3, o serviço pode ser executado pelas provedoras de acesso à internet, sendo esta uma alternativa de gerenciamento, à distância, das redes nas residências dos usuários. Diferentemente das redes empresariais, planejadas e configuradas por administradores de rede, as redes domésticas não apresentam *hardware* e *software* padronizados, assim como administradores dedicados para lidar com os possíveis problemas que essas redes são susceptíveis. Disso, a complexidade adicional e a demanda por uma solução extensível, customizável e modular para o gerenciamento desse tipo de rede, como também para o gerenciamento de problemas e falhas.

O *HomeNetRescue* provê extensibilidade e customização ao permitir que novas aplicações sejam facilmente adicionadas ao serviço, assim como novos dispositivos e parâmetros a serem monitorados. Além disso, o *HNR* possibilita a coordenação das redes sem fio ou cabeadas de uma mesma provedora. Deste modo, a provedora poderia alocar os canais dos roteadores de seus clientes em um edifício, evitando interferências e melhorando a qualidade do serviço oferecido. Outra possibilidade seria a detecção de áreas de sombra (desvanecimento) nas residências, que poderiam ser minimizadas ajustando a potência ou gerando sugestões para os usuários substituírem seus APs. O *HNR* também poderia gerenciar o ponto de conexão entre APs e estações, requisitando que as estações reassociem a outros APs visando obter melhor qualidade de conexão.

Além dessas aplicações, o *HomeNetRescue* também poderia ser utilizado para realizar a coordenação entre APs e estações, gerenciando: coordenadamente a intensidade de sinal dos APs de uma mesma rede de modo que não se interfiram; a alocação de canais de acordo com a vizinhança, realizando a leitura dos *beacons* dos outros APs; a alteração da taxa de *bitrate* ao detectar variações inadequadas na vazão; a associação de estações entre os APs da rede, de modo a balancear a carga nos roteadores; e a alteração da potência de transmissão das estações conforme características de SNR presenciadas nelas.

4. Avaliação

Nessa seção, o *HomeNetRescue* foi avaliado em um protótipo que simula problemas presenciados em redes domésticas. Embora o HomeNetRescue tenha sido modelado para lidar com problemas apresentados nas camadas 2-5 (modelo TCP/IP), nesse artigo avaliamos principalmente eventos na camada de enlace (camada 2). Desta maneira, o foco

da avaliação concentrou-se nas anormalidades que podem colaborar para a degradação da qualidade do sinal. Assim, foram apresentadas soluções visando solucionar problemas que provoquem variações em fluxos sem fio devido a interferências, localidade dos clientes (baixa cobertura do sinal), entre outras causas. O caso de uso apresentado é composto de uma breve descrição do problema, seguido de como ele é detectado e qual a abordagem adotada em sua solução ou minimização. Visando a confiabilidade dos resultados, exceto os gráficos de *jitter* e atraso, que foram realizadas 500 leituras, os demais experimentos foram repetidos 33 vezes. Os resultados correspondem a média das repetições.

4.1. Metodologia

Para a execução dos experimentos com o *HomeNetRescue* foram utilizados: $a)$ uma máquina virtual (controlador) com processador Intel Xeon E312xx 2.2 GHz e 4 GB de RAM, rodando Linux Ubuntu 14.04 (*host system*); $b)$ três computadores pessoais, com processadores: Intel Core 2 Duo 2.53 GHz (roteador com Hostapd[6]), Intel Pentium E2220 2.40 GHz (estação) e Intel Core 2 Duo 6300 1.86 GHz (estação), todos com 2 GB de RAM e Linux Ubuntu 14.04; $c)$ uma placa *Universal Software Radio Peripheral* (USRP) B210[7] (GNU Radio) com uma placa filha FE-TX2; e $d)$ um roteador Linksys WRT54G. Os dois últimos equipamentos (c e d) foram utilizados para a geração de interferências sintéticas.

Os cenário da rede é descrito a seguir. A rede entre o HNR_{AP1} (AP HNR) e a HNR_{STA1} (estação HNR) é a rede controlada pelo o *HomeNetRescue*. Uma rede entre o INT_{AP} (AP interferente) e a INT_{STA} (estação interferente) foi montada nos cantos opostos de uma sala. Ao lado do HNR_{AP1}, a placa $USRP$ (GNU Radio) foi posicionada e configurada para gerar ruído gaussiano, com 100% de ganho, no canal de transmissão da rede controlada pelo *HomeNetRescue*. O mesmo canal é utilizado na rede do roteador Linksys WRT54G (INT_{AP}) e INT_{STA}. Sua potência de transmissão foi configurada em 18 dbm gerando tráfego no mesmo canal de transmissão. O objetivo desta disposição consistiu em induzir a rede de testes (HNR_{AP1} e HNR_{STA1}) a um alto nível de interferência, simulando assim ruídos de aparelhos domésticos (por exemplo, aparelhos micro-ondas, telefones sem fio e babás eletrônicas, etc), bem como a interferência entre APs devido a sobreposição de canais do IEEE 802.11

O tráfego de dados gerado nos experimentos foi obtido através da ferramenta cliente-servidor Bwping[8] (similar ao Iperf) . A ferramenta utiliza o protocolo de transporte UDP e é capaz de fornecer estatísticas de vazão, atraso, *jitter*, quantidade de pacotes enviados e recebidos, entre outras. Para os experimentos, a ferramenta foi configurada para gerar tráfego no limite da capacidade da transmissão sem fio.

4.2. Implementação

O *HomeNetRescue* e o controlador Ethanol foram desenvolvidos em Python 2.7 (mais de 5000 linhas de código[9]). Ambos compõem módulos executados no controlador POX Dart [10] em conjunto com o protocolo OpenFlow 1.3. Este controlador foi adotado

[6]http://linuxwireless.org/en/users/Documentation/hostapd/

[7]https://www.ettus.com/product/details/UB210-KIT

[8]https://github.com/h3dema/bwping-udp

[9]A implementação do serviço encontra-se no github, porém ainda em modo privado. Futuramente objetivamos disponibilizá-la.

[10]https://github.com/noxrepo/pox/tree/dart

por permitir rápida prototipação de aplicações, por ser implementado em Pyhton com código aberto e por ser de fácil configuração e execução em plataformas Linux. Embora o POX não seja o mais adequado para aplicações que demandem alto desempenho, para o HomeNetRescue não é um gargalo da rede, assim como não é o principal ponto de falha, pois técnicas de alta disponibilidade para controladores são consideradas um problema resolvido na literatura. Por adotar uma abordagem centralizada, tal controlador proporciona benefícios em termos de menor tempo de convergência em relação a uma solução distribuída. Adicionalmente, em um ambiente de produção do HomeNetRescue, os controladores de alta disponibilidade, ONIX e ONOS, poderiam ser utilizados.

Como mencionado, o controlador Ethanol pode usufruir dos recursos do protocolo OpenFlow. Em trabalhos futuros, outras versões do protocolo, com outras funcionalidades, podem ser utilizadas para outras aplicações. Por exemplo, uso de múltiplas tabelas no contexto de segurança, entre outras aplicações. Ainda, o controlador Ethanol também possui implementações na linguagem C (mais de 35000 linhas de código). Mensagens são trocadas entre o controlador e os agentes Ethanol. O controlador Ethanol permanece em execução enquanto as aplicações do *HomeNetRescue* requisitam informações (métricas) aos agentes em execução nos componentes da rede.

4.3. Caso de Uso: Combatendo Enlaces Sem Fio Ruins

Este caso de uso representa um cenário onde a interferência degrada a qualidade da transmissão sem fio nos APs, causando perda de pacotes, atraso na entrega de pacotes, retransmissões e diminuição da vazão de recebimento. Assim, nesse caso de uso é descrita a abordagem utilizada pelo *HomeNetRescue* para o tratamento de interferência detectada na transmissão sem fio dos APs. Neste sentido, O HNR_{AP1} foi configurado com uma potência de transmissão de 1 dbm, enquanto que o INT_{AP} foi configurado em 18 dbm, com uma antena acoplada com ganho de 2 dbm. Já a $USRP$, acoplada com uma antena de 8 dbm, foi configurada para emitir sinais gaussianos com 100% de ganho. Por limitações de espaço, o algoritmo do caso de uso foi substituído pela descrição a seguir.

O processo de verificação de interferência consiste do serviço requisitar ao HNR_{AP1}, a cada 10 s, através do controlador, as métricas qualidade do sinal e quantidade de pacotes perdidos de sua interface de rede. Isto feito, quando a quantidade de pacotes perdidos supera o limiar[11] especificado na aplicação, têm-se a necessidade de atuar na rede. Para solucionar este tipo de problema, primeiramente as métricas qualidade do sinal e quantidade de pacotes perdidos juntamente com o limiar de perda de pacotes aceitável são configuradas na aplicação do serviço. Após, o módulo monitor (Seção 3.1) requisita tais informações das camadas adjacentes verificando se a perda no HNR_{AP1} encontra-se aceitável. Em caso negativo, o atuador repassa ao HNR_{AP1} o valor da potência de transmissão a ser alterada, em dbm. Realizado o procedimento, em sequência, a aplicação registra o log da operação efetuada e a taxa de perda de pacotes, para que posteriormente, em uma nova consulta, o serviço realize e registre o cálculo de perdas de pacotes.

4.4. Resultados

Nas figuras 2, 4 e 6, diagramas de caixa foram utilizados para exibir os resultados. Neles, o eixo vertical representa a variável a ser analisada e o eixo horizontal os fatores

[11]Definido empiricamente para os experimentos, pois depende da quantidade de fluxos vigentes no HNR_{AP1}.

de interesse, indicado nas legendas de cada figura. Em cada um, a caixa é delimitada na parte superior pelo quartil Q_3 (distribuindo 25% dos dados acima) e na parte inferior pelo primeiro Q_1 (distribuindo 25% dos dados abaixo). O traço interno indica a mediana (distribuindo 50% dos dados abaixo e 50% acima). As duas linhas na horizontal que se estendem a partir da caixa são os bigodes. O intervalo interquartílico (II) é dado em função $II = Q_3 - Q_1$. O limite superior (LS) é dado em função de $LS = Q_3 + 1,5 * II$, enquanto que o limite inferior (LI), em função de $LI = Q_1 - 1,5 * II$. Por fim, os valores discrepantes ou *outliers* foram desconsiderados.

Na Figura 2, é exibido o impacto das interferências no ambiente de testes e como essas afetam as transmissões sem fio. Com isso, é demonstrada a relação entre o número de pacotes perdidos e a vazão percebida no HNR_{AP1}. Isto, para as situações onde o HNR_{AP1} não esteve sob influência da interferência; quando a interferência foi inserida somente pelo tráfego do INT_{AP} ou somente pela $USRP$; e, quando ambos geram interferências simultaneamente. Baseado nisso, conforme esperado, foi observado que à medida que se acrescenta novas fontes de interferência, a quantidade de pacotes perdidos por vazão aumenta, validando que as interferências comprometem o desempenho da rede.

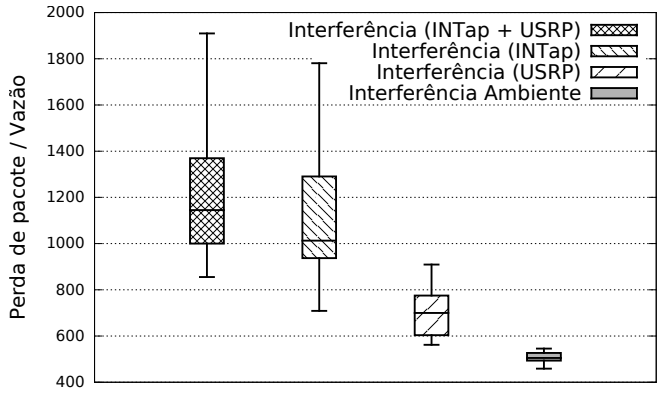


Figura 2. Razão entre número de pacotes perdidos pela vazão percebida no AP

Na Figura 3, é ilustrada a variação da vazão durante o experimento. Na fase inicial do mesmo (em amarelo entre 0 e 150 s), o HNR_{AP1} esteve com uma potência de saída de 1 dbm. Em 150 s do experimento, foi acrescentada a interferência do $USRP$. Aguardou-se alguns segundos para a vazão estabilizar. A figura mostra a vazão média depois da estabilização, no período de 170 a 250 s. Em 250 s, o controlador foi acionado, detectando um problema de vazão (através da quantidade de perdas) e aumentando a potência do HNR_{AP1} para 15 dbm (capacidade máxima). Com isso, o HNR conseguiu obter uma melhora de 7% na vazão percebida pelo cliente. O HNR também é capaz de reduzir dinamicamente tal potência, assim como utilizar outras métricas. Isso viabiliza o gerenciamento da rede com mais justiça, por exemplo, reduzir ruídos nos APs vizinhos uma vez que a vazão alcançada já atenda à necessidade da aplicação.

Vale ressaltar que ao executar o *HomeNetRescue*, o processamento realizado na CPU do HNR_{AP1} alcançou no máximo 3% da capacidade total do dispositivo. A utilização de memória não ultrapassou 3% (cerca de 40 MB). Esses valores indicam que o serviço demanda poucos recursos dos dispositivos e pode, por exemplo pode ser utilizado com o TP-Link TL-WR2543ND que possui processador 400 MHz, 64 MB de memória

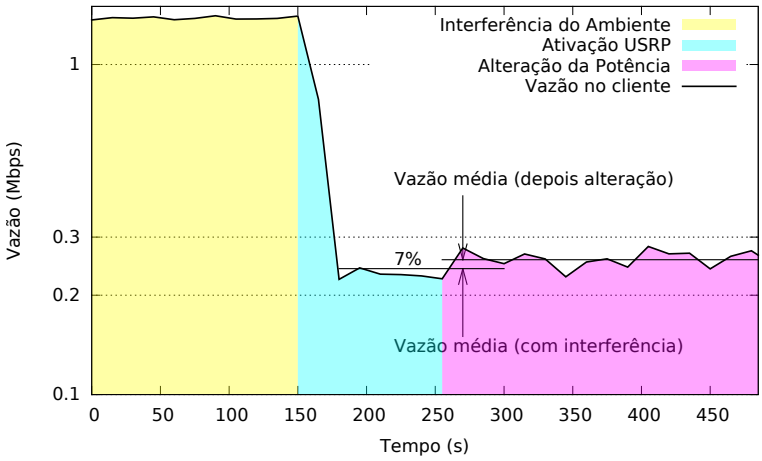


Figura 3. Exemplo de execução do *HomeNetRescue* com ganho de 7% para um fluxo frente a interferência sintética

RAM e 8 MB de memória flash. As alterações de potência de transmissão, quando realizadas, gastaram no máximo 29 ms entre a detecção do problema pelo controlador e o envio da regra ao dispositivo.

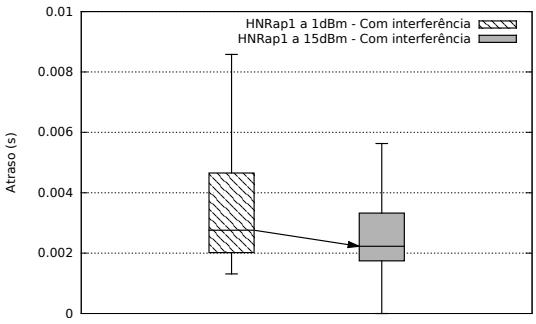


Figura 4. Atraso medido pelo cliente

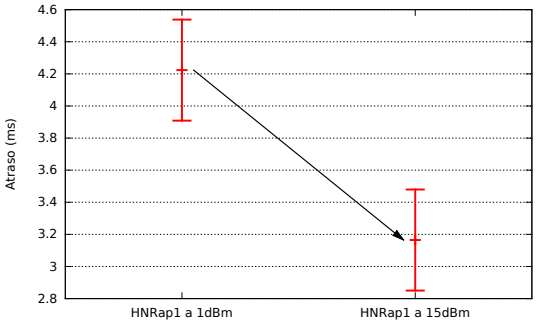


Figura 5. Atraso médio com intervalo de confiança de 95% com interferência

Na Figura 4, são apresentados os valores de atraso (em s) obtidos no experimento, medidos no cliente. Na barra da esquerda é mostrado o atraso observado pelo cliente com o HNR_{AP1} configurado para a potência de 1 dbm sob a influência do INT_{AP}. A barra à direita mostra a situação quando o controlador aumenta a potência do HNR_{AP1} para 15 dbm. Pode-se notar na figura (seta) que há uma melhoria, com a redução do atraso resultado da ação do serviço. Na Figura 5, é representado o intervalo de confiança do atraso médio para a situação com o HNR_{AP1} funcionando na potência de 1 dbm e na situação depois da alteração para 15 dbm. Não existe sobreposição entre os intervalos de confiança da média, portanto as médias são independentes.

Na Figura 6, apresenta-se os valores de *jitter* (s) obtidos no experimento. Na barra à esquerda é exibido o *jitter* observado pelo cliente com o HNR_{AP1} configurado para a potência de 1dbm sob a influência do INT_{AP} . A barra à direita mostra a situação quando o serviço aumenta a potência do HNR_{AP1} para 15 dbm, mantidas as outras condições constantes. Verifica-se que há uma redução do *jitter*. O intervalo de confiança do *jitter*

médio é mostrado na Figura 7. As médias são independentes pois não existe sobreposição dos intervalos. Dessa forma, vemos que o controlador consegue, ao modificar a potência do ponto de acesso, melhorar o *jitter* dinamicamente quase obtendo uma situação próxima ao cenário sem interferência.

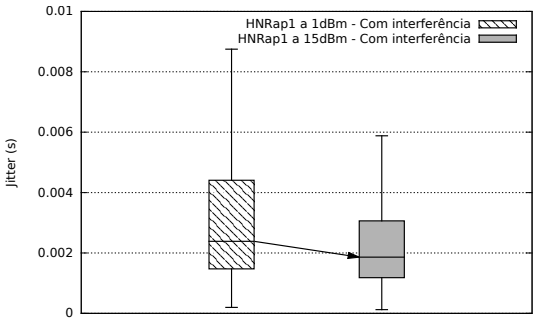


Figura 6. Jitter medido no cliente

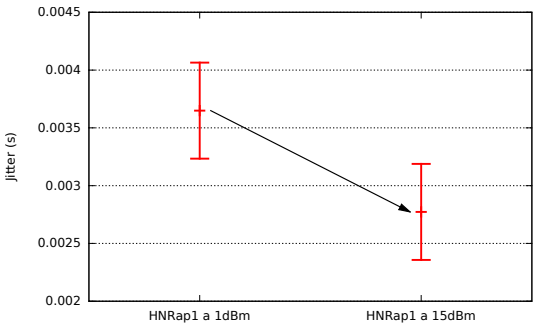


Figura 7. Jitter médio com intervalo de confiança de 95% com interferência

Finalmente, o *HomeNetRescue* melhorou o desempenho da rede. A perda de mensagens diminuiu, resolvendo o problema de perdas acima do limiar, o que impacta diretamente na vazão detectada. O objetivo do artigo concentrou-se na descrição do protótipo do HomeNetRescue em detrimento dos algoritmos de controle para tratar eventos específicos de falha ou como diferenciar tais eventos.

5. Conclusão e Trabalhos Futuros

Neste artigo, foi apresentado o *HomeNetRescue*, um serviço SDN para o gerenciamento autônomo de redes domésticas (inclusive redes sem fio) voltado para a detecção, o diagnóstico e a solução automática ou minimização de problemas que também pode atuar em aspectos de desempenho nessas redes. O *HomeNetRescue* apresenta uma arquitetura genérica e pode ser empregado pelas provedoras de acesso à Internet para que estas gerenciem problemas à distância nas redes domésticas de seus usuários. Tal serviço pode agregar a essas redes funcionalidades de detecção e solução automática de problemas de modo que estas se tornem mais estáveis e confiáveis. Isto pode proporcionar redução de custos para as provedoras e para os usuários, gerar menor demanda por serviços de suporte e menor tempo de recuperação em caso de falhas.

A partir da avaliação experimental,o *HomeNetRescue* demonstrou benefícios científicos proporcionando ganhos na vazão, redução em atrasos e no *jitter* de transmissões sem fio e redução da intervenção humana na solução de problemas. O serviço foi capaz de detectar um aumento na quantidade de perda de pacotes devido a colisões, atuando na rede automaticamente, aumentando a potência de transmissão, sendo capaz de combater enlaces sem fio ruins. Como trabalho futuros, pretende-se implementar casos de uso que demonstrem a capacidade do serviço em coordenar os dispositivos da rede, gerenciando APs em que as estações estão conectadas e atuando sobre os mecanismos de QoS.

6. Agradecimentos

Agradecimentos às instituições de amparo à pesquisa CNPq, CAPES e FAPEMIG pelo financiamento e suporte para o desenvolvimento dessa pesquisa.

Referências

[Biswas et al. 2015] Biswas, S., Bicket, J., Wong, E., Musaloiu-E, R., Bhartia, A., and Aguayo, D. (2015). Large-scale measurements of wireless network behavior. In *ACM SIGCOMM*.

[Bouchet et al. 2014] Bouchet, Javaudin, Kortebi, Adbellaouy, E., Brzozowski, Katsianis, Mayer, Guan, Lebouc, Fontaine, Cochet, Jaffré, Mengi, Celeda, Aytekin, G., and Kurt (2014). Acemind: The smart integrated home network. In *2014 International Conference on Intelligent Environments (IE)*.

[Cisco 2015] Cisco (2015). Cisco Visual Networking Index: Forecast and Methodology, 2015–2020. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf. [Online; acessado 14-Novembro-2016].

[DiCioccio et al. 2012] DiCioccio, L., Teixeira, R., and Rosenberg, C. (2012). Measuring and characterizing home networks. *SIGMETRICS Perform. Eval. Rev.*

[Dong and Dulay 2011] Dong, C. and Dulay, N. (2011). Argumentation-based fault diagnosis for home networks. In *Proceedings of the 2nd ACM SIGCOMM (HomeNets)*.

[Fratczak et al. 2013] Fratczak, T., Broadbent, M., Georgopoulos, P., and Race, N. (2013). Homevisor: Adapting home network environments. In *2013 Second European Workshop on Software Defined Networks (EWSDN)*, pages 32–37.

[Gheorghe et al. 2015] Gheorghe, G., Avanesov, T., Palattella, M.-R., Engel, T., and Popoviciu, C. (2015). SDN-RADAR: Network troubleshooting combining user experience and SDN capabilities. In *IEEE NetSoft*, pages 1–5.

[Guedes et al. 2012] Guedes, D., Vieira, L. F. M., Vieira, M. M., Rodrigues, H., and Nunes, R. V. (2012). Redes definidas por software: uma abordagem sistêmica para o desenvolvimento de pesquisas em redes de computadores. *Simpósio Brasileiro de Redes de Computadores (SBRC)*, pages 160–210.

[Kim et al. 2014a] Kim, K.-H., Nam, H., and Schulzrinne, H. (2014a). WiSlow: A Wi-Fi network performance troubleshooting tool for end users. In *IEEE INFOCOM*, pages 862–870.

[Kim et al. 2014b] Kim, K.-H., Nam, H., Singh, V., Song, D., and Schulzrinne, H. (2014b). DYSWIS: crowdsourcing a home network diagnosis. In *International Conference on Computer Communication and Networks (ICCCN)*, pages 1–10.

[Macedo et al. 2015] Macedo, D. F., Guedes, D., Vieira, L. F. M., Vieira, M. A. M., and Nogueira, M. (2015). Programmable networks: From software-defined radio to software-defined networking. *IEEE Communications Surveys Tutorials*.

[Moura et al. 2015] Moura, H., Bessa, G. V., Vieira, M. A., and Macedo, D. F. (2015). Ethanol: Software Defined Networking for 802.11 Wireless Networks. *IFIP/IEEE International Symposium on Integrated Network Management (IM)*.

[Perera et al. 2014] Perera, C., Liu, C., Jayawardena, S., and Chen, M. (2014). A survey on internet of things from industrial market perspective. *IEEE Access*, pages 1660–1679.

[Sundaresan et al. 2013] Sundaresan, S., Grunenberger, Y., Feamster, N., Papagiannaki, D., Levin, D., and Teixeira, R. (2013). WTF? locating performance problems in home networks.

[Yiakoumis et al. 2011] Yiakoumis, Y., Yap, K.-K., Katti, S., Parulkar, G., and McKeown, N. (2011). Slicing home networks. In *Proceedings of the 2nd ACM SIGCOMM (HomeNets)*, pages 1–6.

XXII Workshop de Gerência e Operação de Redes e Serviços (WGRS)

SBRC 2017

Sessão Técnica 3 – Gerenciamento de Infraestrutura/Plataforma IoT

Arquitetura RF-Miner - Uma solução para localização indoor

Eduardo L. Gomes, Mauro Fonseca, Anelise Munaretto, Carlos R. Guerber

[1]Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial (CPGEI) – Universidade Tecnológica Federal do Paraná (UFRGS)
Centro – 80.230-901 – Curitiba – PR – Brazil

{elgomes10,crguerber}@gmail.com, {maurofonseca,anelise}@utfpr.edu.br

Abstract. *The use of passive UHF RFID tags for indoor location has been widely studied due to its low cost. However, there is still a great difficulty to reach good results, mainly due the radio frequency variation in environments that have materials with reflective surfaces, such as metal. This paper proposes a localization architecture using passive UHF RFID tags and data mining techniques to identify the exact location indoors.*

Resumo. *A utilização de etiquetas RFID UHF passivas para localização indoor vem sendo amplamente estudada devido ao seu baixo custo. Porém ainda existe uma grande dificuldade em obter bons resultados, principalmente devido à variação de rádio frequência em ambientes que possuem materiais reflexivos, como por exemplo, metais. Este artigo propõe uma arquitetura de localização utilizando etiquetas RFID UHF passivas e técnicas de mineração de dados possibilitando identificar a posição exata em ambientes internos.*

1. Introdução

A tecnologia RFID (*Radio Frequency Identification* – Identificação por Rádio Frequência) é um método de identificação automática que utiliza sinais de rádio para identificar, rastrear e gerenciar produtos, documentos ou até mesmo animais e pessoas sem nenhum tipo de contato físico e nem mesmo campo visual. Isto só é possível devido a utilização de dispositivos conhecidos como transponders ou tags, que são etiquetas eletrônicas, passivas, semi-passivas ou ativas, classificadas do ponto de vista da fonte de alimentação.

A pesquisa de modelos de localização indoor tem utilizado preferencialmente a tecnologia RFID para sua implementação, já que a tecnologia GPS não apresenta resultados precisos em ambientes internos. Modelos de localização indoor são criados com vários objetivos diferentes e utilizando técnicas distintas de apuração da acuracidade das etiquetas. Existem trabalhos que têm como objetivo auxiliar a movimentação indoor de robôs utilizando como técnica o atributo S-CRR (*Swift Communication Range Recognition*) [Nakamori et al. 2012]. Outros possuem o mesmo objetivo, porém utilizam técnicas probabilísticas de apuração [Hori et al. 2008]. Da mesma maneira existem trabalhos que utilizam a mesma técnica S-CRR mas para atender objetivos diferentes, como o de identificar itens em um determinado local, ao invés de movimentação de robôs [Uchitomi et al. 2010].

A arquitetura apresentada no presente artigo tem como objetivo apurar a localização de itens em ambientes internos com características de armazenamento verticais, que é uma característica muito utilizada para aproveitar melhor o espaço de estocagem de produtos, comércio, bibliotecas e arquivos de documentos.

A metodologia utilizada para desenvolver a arquitetura se concentra em prover independência de atributos e hardware. Inicialmente foram identificados quais são os atributos que melhor contribuem na apuração da localização. Em seguida, foi realizado um teste em relação a influência da quantidade de antenas na arquitetura e o que cada antena contribui ou interfere na acuracidade. Finalmente foram utilizadas e comparadas algumas técnicas de mineração de dados com os algoritmos de classificação para identificar qual apresenta o melhor resultado. Para isso foi utilizado o software WEKA (*Waikato Environment for Knowledge Analysis*) testando os algoritmos de classificação K-Estrela, Árvores de Decisão J48, Redes Bayesiana (*Bayes Network*) e *K-Nearest Neighbor* (KNN) com diferentes parâmetros.

O artigo está organizado da seguinte forma, a seção 2 apresenta os trabalhos relacionados na área, em seguida a seção 3 apresenta a modelagem detalhada da arquitetura, e a seção 4 a implementação dessa arquitetura proposta. Na seção 5 são apresentados os resultados obtidos na aplicação da arquitetura em ambiente real, seguidos na seção 6 da análise dos resultados. Finalmente na seção 7 são apresentadas as conclusões do trabalho.

2. Localização Indoor Utilizando a Tecnologia RFID

Um sistema tradicional de localização indoor consiste em identificar precisamente itens em um determinado espaço dentro de edificações. O sistema de localização por satélite (GPS) que é amplamente utilizado para localização externa, não tem um bom desempenho para localização em ambientes internos, pois os sinais de micro-ondas são atenuados pelo teto, parede e outros objetos, o que faz a precisão reduzir drasticamente, ou nem mesmo ser possível localizar o objeto [Huang et al. 2015].

Alternativamente outras tecnologias são pesquisadas para propor soluções viáveis a fim de atender à esta necessidade, como por exemplo: Posicionamento Magnético ou Medições Inerciais que não utilizam ondas de rádio.

Entretanto o custo também é um fator extremamente relevante para a viabilidade de implantação da arquitetura apresentada. A necessidade de identificar cada item que se deseja localizar, faz com que diversas tecnologias se tornem inviáveis do ponto de vista econômico, pois o custo se multiplica pela quantidade de itens a identificar.

Por outro lado há tecnologias que utilizam ondas de rádio, como Wi-fi (WPS), Bluetooth e RFID, as quais empregam várias técnicas distintas dentre as mais conhecidas são: Ângulo de Recepção (*Angle-of-Arrival - AoA*), Diferença de Tempo de Transmissão (*Time Difference of Arrival - TDoA*), Indicador de Intensidade do Sinal Recebido (*Received Signal Strength Indicator - RSSI*) [Akre et al. 2014].

A Tecnologia RFID UHF utilizando etiquetas passivas se demonstra uma solução viável para esta necessidade. Modelos de sistemas de localização indoor utilizando RFID para diversas finalidades já foram pesquisados. Os autores em [Akre et al. 2014] apresentaram um modelo que utiliza etiquetas RFID passivas localizadas em um espaço em duas dimensões (2-D) usando a abordagem de aprendizagem. A técnica utilizada consiste em identificar o RSSI em quatro antenas e realizar medições com potências diferentes, montando uma base de dados de aprendizagem e utilizando o algoritmo KNN para obter a localização. Em seu trabalho [Ting et al. 2011] apresentaram um modelo de localização também utilizando como atributo de medição das antenas o RSSI, porém com uma técnica

mais simples de apuração, criando uma base de dados denominada na pesquisa como *Look-Up Table* (LUT), contendo as médias das medições prévias do RSSI. Em seguida, realizaram a localização através do cálculo da distância euclidiana entre a medição atual e a base LUT. A pesquisa apresentou bons resultados, porém a precisão foi de três metros para cada etiqueta, o que pode não ser útil dependendo do modelo de negócio onde o sistema de localização será implantado.

Já [Zhang et al. 2015] procuraram extrair mais características do atributo RSSI, utilizando diversas variações da potência de transmissão, orientação de etiquetas e realizando um extenso trabalho de leituras. Assim, extraíram além do RSSI puro, sua média, desvio padrão, valor mínimo e valor máximo.

Além disso, existem estudos que utilizaram diversas técnicas combinadas para criar um modelo de localização indoor, como fizeram [Huang et al. 2015], com a finalidade de apurar o resultado de maneira mais rápida e melhorar o desempenho da localização em tempo real, empregaram uma combinação de *Kalman-Filter Drift* e *Heron-Bilateration Localization Estimative* para melhorar a eficiência e precisão de métodos tradicionais como *Proximity Pattern Matching, Linear-Like RSSI-to-Distance Transformation, Trilateração e Multilateração*. Entretanto utilizaram etiquetas ativas, o que aumenta consideravelmente o custo de implantação do modelo.

Assim o trabalho apresentado neste artigo propõe uma arquitetura de sistema de localização em ambientes internos, utilizando etiquetas RFID UHF passivas de alta precisão e baixo custo. Para isso o estudo se concentra na investigação da melhor configuração da arquitetura com a quantidade de antenas e na identificação dos atributos que melhor contribuem para a acuracidade da localização, derivando do Indicador de Intensidade do Sinal Recebido (RSSI) e da Quantidade de Leituras em um Período de Tempo (*Read Count*). Foram utilizadas técnicas de mineração de dados como os algoritmos de classificação K-Estrela, Árvores de Decisão J48, Redes Bayesiana e KNN, para identificar qual técnica se comporta melhor para o cenário avaliado.

3. Arquitetura RF-Miner

A arquitetura de localização *indoor* utilizando etiquetas RFID UHF passivas proposta neste artigo é denominada RF-Miner e está subdividida em 3 módulos conforme demonstra a Figura 1.

3.1. Módulo de Leitura Física

Um sistema básico RFID é composto por 3 equipamentos: etiqueta, antenas e leitor. O módulo de leitura física nada mais é do que um sistema básico RFID que realiza as leituras e repassa a informação para o módulo de extração de características. Este módulo é independente do fabricante, pois o mesmo padroniza informações de entrada em um fluxo de saída de dados composto pela tupla *timestamp*, *RSSI*, *TagID*. É importante ressaltar que a qualidade da instalação do sistema, como qualidade de cabos e antenas bem fixadas podem interferir na apuração dos resultados.

3.2. Módulo Extração de Características

É a parte lógica da arquitetura, responsável pela preparação dos atributos e formação da base de treinamento. Para permitir que o módulo realize a derivação dos atributos é ne-

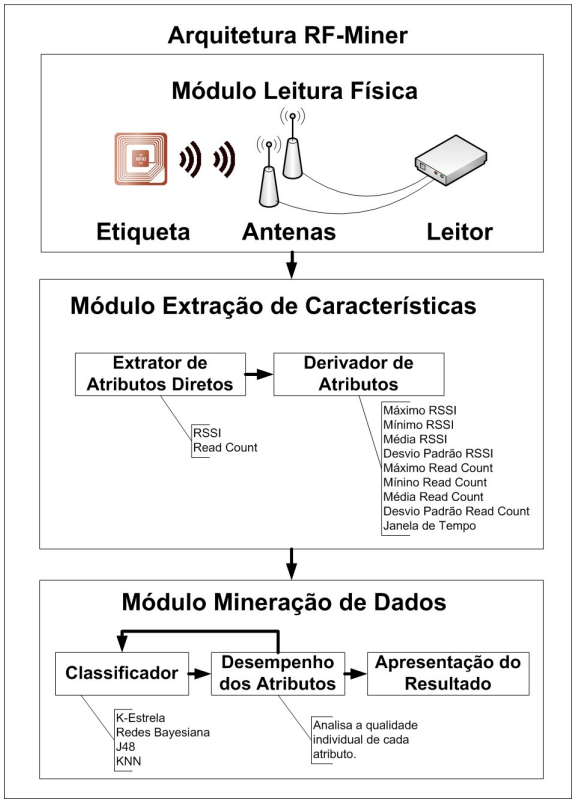


Figura 1. Arquitetura RF-Miner

cessário configurar o parâmetro denominado Tamanho da Janela, que na prática é a quantidade de tuplas que serão utilizadas para calcular as médias, mínimos, máximos e desvio padrão. Inicialmente, o módulo recebe o parâmetro tamanho da janela e o fluxo de dados do módulo de Leitura Física e extrai as informações dos atributos diretos que são: RSSI e RC (*Read Count*), baseado no tamanho da janela. Após isso, realiza a derivação destes atributos, onde inicialmente cada leitura é atribuída em uma janela de tempo, utilizando como critério a ordem sequencial da leitura. A partir do momento que cada leitura está alocada em sua janela de tempo é realizado o cálculo da média aritmética, desvio padrão e da identificação dos valores mínimos e máximos para cada atributo (RSSI e RC), criando uma base de dados com 10 (dez) atributos para cada antena conforme demonstrado na Tabela 1.

Existem 2 (dois) atributos que não estão relacionados diretamente com a antena. O número da janela de tempo e a posição da etiqueta no momento da leitura, apresentados na Tabela 2.

Após esta etapa, a base de dados e treinamento está pronta para ser utilizada no módulo de Mineração de Dados.

3.3. Módulo Mineração de Dados

Usando como entrada os dados preparados e formatados pelo módulo Extração de Características, o módulo Mineração de Dados realiza a classificação utilizando um dos algoritmos disponibilizados. Por meio do processo de Desempenho dos Atributos, avalia individualmente cada atributo com o objetivo de identificar se o mesmo contribui ou

Tabela 1. Atributos relacionados diretamente à antena

Atributo	Tipo de Dado	Descrição
rssi_antena X	Decimal	RSSI da antena
rc_antena X	Inteiro	Número de leituras da etiqueta na antena
avg_rssi_antena X	Decimal	Média aritmética do RSSI na janela de tempo
avg_rc_antena X	Decimal	Média aritmética do RC na janela de tempo
min_rssi_antena X	Decimal	Menor valor do RSSI na janela de tempo
min_rc_antena X	Decimal	Menor valor do RC na janela de tempo
max_rssi_antena X	Decimal	Maior valor do RSSI na janela de tempo
max_rc_antena X	Decimal	Maior valor do RC na janela de tempo
stddev_rssi_antena X	Decimal	Desvio padrão do RSSI na janela de tempo
stddev_rc_antena X	Decimal	Desvio padrão do RC na janela de tempo

Tabela 2. Atributos relacionados diretamente à antena

Atributo	Tipo de Dado	Descrição
Janela de tempo	Inteiro	Número da janela de tempo
Posição	Classe	Nome ou número da posição na prateleira

não para a localização naquele ambiente. Após encontrar o melhor conjunto de atributos, realiza novamente a classificação e apresenta a localização encontrada utilizando este conjunto, resultando na melhor solução encontrada para o ambiente.

4. Implementação

4.1. Preparação do Ambiente de Testes

Para a criação do ambiente de testes foram utilizadas quatro antenas UHF mono estáticas que atuam na faixa de frequência 902 a 928 MHz com um ganho de 6 dBi. O equipamento utilizado para realização das leituras foi o *ThingMagic Mercury 6*, que é um leitor RFID UHF de alto desempenho. Suporta até 4 antenas mono estáticas, entradas e saídas digitais e conexão Wi-FI. Com o objetivo de validar e implementar a arquitetura de localização indoor em um ambiente real, foi utilizado como estudo de caso real a Biblioteca Conselheiro Mafra da Universidade do Contestado – Campus Mafra. As etiquetas passivas foram dispostas lado a lado em 4 (quatro) prateleiras (P1, P2, P3, P4) de livros de 210 cm de largura. Foram utilizadas 28 (vinte e oito) etiquetas coladas no sentido vertical na lombada dos livros. As antenas foram dispostas a 120 cm de distância das etiquetas. A Figura 2 demonstra o ambiente de testes montado.

4.2. Realização das Leituras

Foram realizadas em cada célula ou *tag* alvo 100 leituras de 5 segundos divididas em 10 janelas de tempo ou seja, o parâmetro denominado Janela de Tempo utilizado nesta

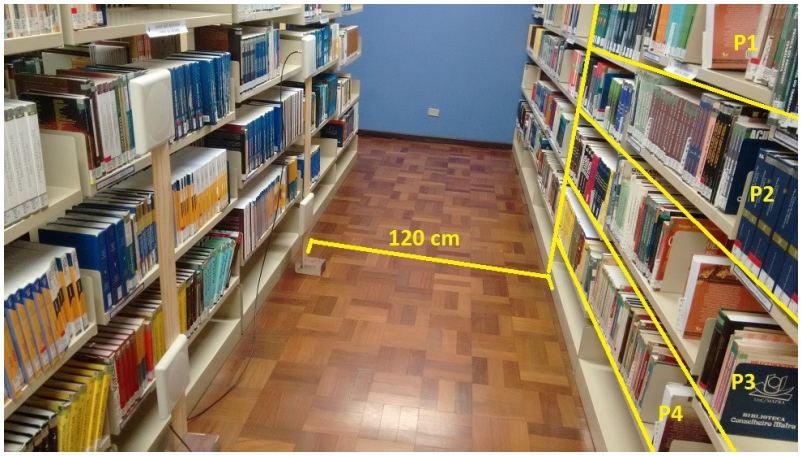


Figura 2. Ambiente de Testes

implementação foi de tamanho 10. Em cada *tag* foi medido o RSSI em dBm e a *Read Count* para cada uma das 4 antenas totalizando 2.800 leituras. Para esta etapa o módulo de Leitura Física faz a conexão por TCP/IP com o Leitor *ThingMagic 6*, envia os comandos para iniciar as leituras e recebe as respostas das etiquetas localizadas com seus respectivos atributos diretos.

4.3. Formatação da Base e Preparação dos Atributos

Após a realização das leituras o módulo Extração de Características identificou para cada janela de tempo o valor máximo, mínimo, média aritmética e desvio padrão do RSSI e do *Read Count* para cada leitura e etiqueta. Organizou os atributos conforme demonstrado nas Tabelas 1 e 2, e colocou em 2.800 registros as 100 leituras para cada uma das 28 etiquetas dentro de um arquivo do tipo ARFF (*Attribute-Relation File Format*), que foi utilizado para criação da base de dados no software WEKA.

4.4. Mineração de Dados

O módulo Mineração de Dados utiliza para a classificação o software WEKA. A técnica utilizada para os testes foi a validação cruzada (*Cross-validation*) com o parâmetro dobra (*fold*) igual a 10. A Figura 3 demonstra o comportamento desta técnica. Inicialmente o software divide a base em 10 partes iguais. Em cada iteração utiliza 9 partes para treinamento e 1 para testes. Este processo é repetido para cada uma das 10 partes sempre trocando a qual será utilizada para testes. Com esta técnica foi possível multiplicar em 10 vezes os 2.800 registros que foram utilizados na mineração de dados. Ao todo, foram 28.000 registros, sendo 25.200 para treinamento e todos os 2.800 para testes.

Para a localização através da mineração de dados foram realizados testes com 4 algoritmos diferentes. O objetivo foi identificar o desempenho dos algoritmos na arquitetura e com quais atributos existe uma maior acuracidade. Os algoritmos avaliados foram: KNN (com o parâmetro K = 1, 3 e 5), Algoritmo K-Star, Árvore de Decisão J48 e Redes Bayesianas. O módulo pode trabalhar com outros algoritmos de classificação, provando assim a generalização da arquitetura.

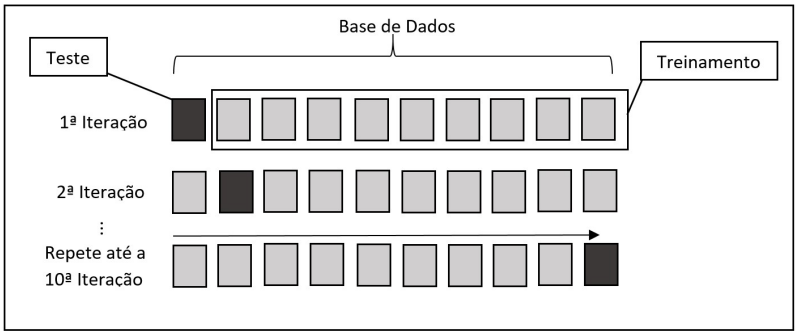


Figura 3. Validação Cruzada

5. Resultados

Para identificar quais atributos melhor contribuem para a localização da *tag* no modelo proposto, o módulo de Mineração de dados avaliou todos os atributos realizando três classificações para cada algoritmo adicionando individualmente cada grupo de atributos a cada classificação. Os resultados das três classificações são apresentados a seguir.

Na primeira classificação apresentada foi considerado apenas o atributo RSSI para cada antena, ou seja, um grupo de quatro atributos: RSSI da antena 1, 2, 3 e 4. Os resultados são apresentados na Tabela 3.

Tabela 3. Classificação utilizando quatro atributos (RSSI)

Algoritmo	Acertos	Erros	% Acerto
KNN K=1	2.660	140	95,00
K-Star	2.650	150	94,64
KNN K=3	2.650	150	94,64
KNN K=5	2.640	160	94,29
Árvore de Decisão J48	2.590	210	92,50
Redes Bayesianas	2.560	240	91,43

Na segunda classificação apresentada foram considerados os atributos RSSI em conjunto com o *Read Count* para cada antena ou seja 2 grupos de 4 atributos, totalizando oito atributos (RSSI e *Read Count* da antena 1, 2, 3 e 4). A Tabela 4 apresenta os resultados da segunda classificação.

E na terceira classificação apresentada todos os atributos foram considerados. Os atributos diretos: RSSI e *Read Count* e os atributos derivados da janela de tempo: média, mínimo, máximo e desvio padrão. Os atributos utilizados para esta classificação foram 40 no total, ou seja, 10 atributos de cada antena, como foi demonstrado na Tabela 1. A Tabela 5 apresenta os resultados obtidos na terceira classificação.

A Figura 4 apresenta o gráfico comparativo de desempenho dos algoritmos nas três situações utilizando 4 atributos, 8 atributos e 40 atributos.

Após identificar os atributos que mais contribuíram na apuração da localização das

Tabela 4. Classificação utilizando oito atributos (RSSI e Read Count)

Algoritmo	Acertos	Erros	% Acerto
K-Star	2.770	30	98,93
KNN K=3	2.750	50	98,21
KNN K=5	2.750	50	98,21
Redes Bayesianas	2.750	50	98,21
KNN K=1	2.740	60	97,86
Árvore de Decisão J48	2.640	160	94,29

Tabela 5. Classificação utilizando quarenta atributos (RSSI, Read Count e janelas de tempo)

Algoritmo	Acertos	Erros	% Acerto
Árvore de Decisão J48	2.800	0	100,00
Redes Bayesianas	2.800	0	100,00
K-Star	2.800	0	100,00
KNN K=5	2.779	21	99,25
KNN K=3	2.773	27	99,04
KNN K=1	2.764	36	98,71

etiquetas, o módulo de Mineração de Dados realizou testes considerando os atributos de uma, duas, três e quatro antenas, com o objetivo de analisar como a quantidade de antenas interfere no resultado. A Tabela 6 apresenta percentuais obtidos para cada situação.

A Figura 5 apresenta o gráfico comparativo dos percentuais de acerto em relação à quantidade de antenas.

Para validar se a arquitetura RF-Miner mantém o índice de acuracidade em tempo real, foi realizado o teste onde todos os livros foram trocados de lugar e realizado nova leitura de apenas 1 janela de tempo, que é o tempo mínimo necessário para apurar a localização de uma etiqueta, ou seja, quanto maior a janela de tempo, mais demorado irá ser a apuração do resultado, consequentemente haverá mais tuplas para o módulo de mineração de dados utilizar para a classificação. Na presente implementação, como foi utilizado leituras de 5 segundos e uma janela de tempo de tamanho 10 a apuração de 1 resultado de localização é de 50 segundos. A Tabela 7 apresenta os resultados de classificação em tempo real com apenas uma janela de tempo, com quatro antenas e todos os atributos.

Outra verificação foi realizada em 10 janelas de tempo após a troca dos livros. É importante ressaltar que para verificação em tempo real a cada janela de tempo que é adicionada na base de treinamento a primeira janela da base é removida, utilizando o conceito FIFO (*First in First Out*). Este procedimento foi adotado para manter a base

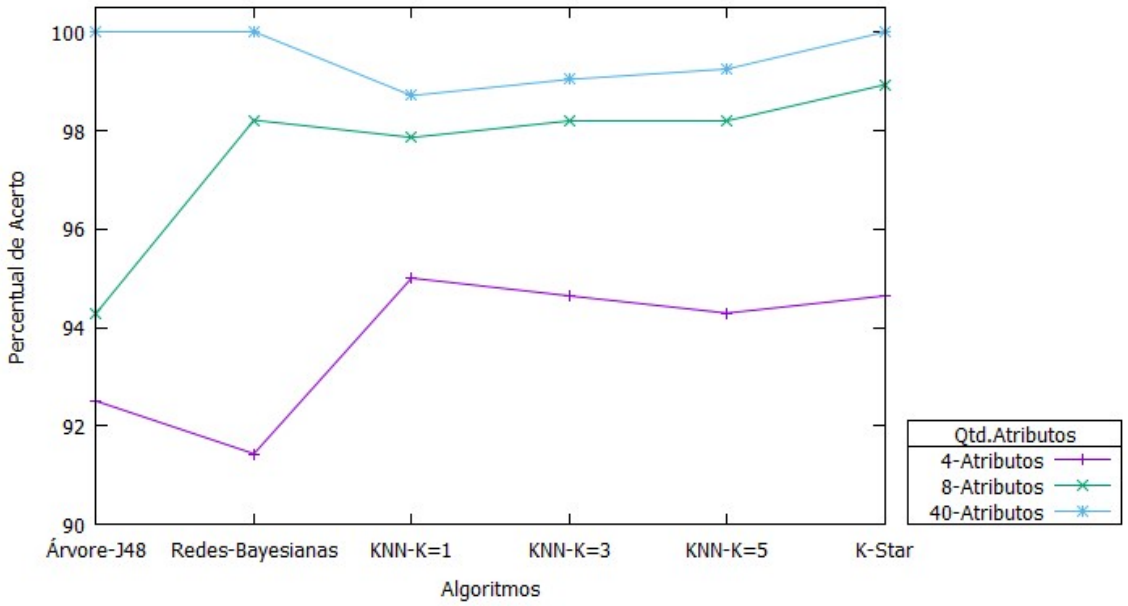


Figura 4. Comparativo de desempenho em relação à quantidade de atributos

Tabela 6. Comparativo da classificação utilizando diferentes quantidades de antenas

Algoritmo	1 Antena	2 Antenas	3 Antenas	4 Antenas
Árvore de Decisão J48	95,89%	99,96%	100,00%	100,00%
Redes Bayesianas	95,86%	100,00%	100,00%	100,00%
KNN K=1	95,21%	98,28%	98,57%	98,71%
KNN K=3	95,43%	98,46%	98,89%	99,04%
KNN K=5	95,00%	99,10%	99,25%	99,25%
K-Star	96,36%	100,00%	100,00%	100,00%

sempre atualizada mesmo diante das mudanças das etiquetas no cenário. A Tabela 8 apresenta os resultados.

6. Análise dos Resultados

Na primeira investigação da implementação da arquitetura RF-Miner o módulo de Mineração de Dados por meio do processo de Desempenho de Atributos identificou quais atributos contribuem para a precisão da localização. Analisando os resultados obtidos nas três amostragens apresentadas, Tabelas 3, 4 e 5, foi possível perceber que a utilização das janelas de tempo derivando os atributos RSSI e *Read Count* em mínimo, máximo, média e desvio padrão, contribuíram significativamente para uma melhor acuracidade da arquitetura. Portanto o papel do módulo de extração de características foi fundamental para o resultado final, pois foi possível obter 100% de acerto em três dos quatro classificadores utilizados na arquitetura.

Os algoritmos de classificação utilizados na arquitetura apresentaram resultados

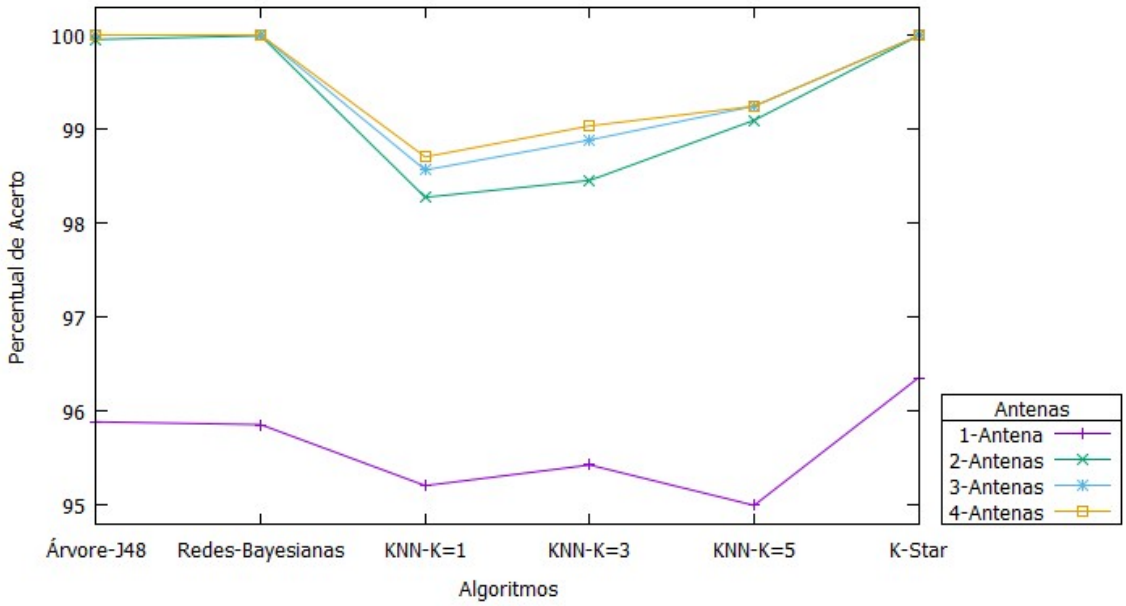


Figura 5. Comparativo de desempenho em relação à quantidade de antenas

Tabela 7. Classificação utilizando quarenta atributos (RSSI, *Read Count* e janelas de tempo) em tempo real com uma janela de tempo

Algoritmo	Acertos	Erros	% Acerto
Árvore de Decisão J48	27	1	96,43
Redes Bayesianas	25	3	89,29
K-Star	24	4	85,71
KNN K=5	22	6	78,57
KNN K=3	23	5	82,14
KNN K=1	19	9	67,85

eficientes, dos quatro algoritmos testados apenas o KNN não conseguiu 100% de acerto, embora obteve resultados acima de 99% utilizando como parâmetro K (número de vizinhos) igual a 5.

Em relação à quantidade de antenas, como esperado, foi possível perceber que quanto mais antenas melhor o resultado, porém como o custo de implementação deve ser sempre considerado, com duas antenas já é possível obter 100% de acerto na arquitetura RF-Miner. O desempenho com duas antenas é extremamente importante, pois reduz significativamente o custo de implementação da arquitetura, além de reduzir em 50% o custo das antenas, também reduz o custo dos leitores, já que leitores com capacidade para receber 2 antenas custam em torno de 30% a menos que leitores com capacidade para 4 antenas. A economia aproximada é de 40% na aquisição dos conjuntos de leitores e antenas além do menor consumo de energia para a operação do sistema.

O teste de implementação em tempo real, que considerou a leitura de apenas uma

Tabela 8. Classificação utilizando quarenta atributos (RSSI, *Read Count* e janelas de tempo) em tempo real com dez janelas de tempo

Algoritmo	Acertos	Erros	% Acerto
Árvore de Decisão J48	28	0	100,00
Redes Bayesianas	27	1	96,43
K-Star	28	0	100,00
KNN K=5	23	5	82,14
KNN K=3	23	5	82,14
KNN K=1	22	6	78,57

janela e mudança de posição de todos os livros, obteve bons resultados. Mas é perceptível que quanto mais tempo o livro fica na mesma posição melhor é a acuracidade, pois a verificação após dez janelas de tempo pode-se obter 100% de acertos em alguns classificadores.

7. Conclusões

A tecnologia RFID continua sendo muito utilizada em diversos setores, principalmente em sistemas de controle de processos, estocagem, rastreamento e antifurto. Utilizar esta versatilidade da tecnologia para implementar sistemas de localização *indoor* pode contribuir ainda mais para a viabilidade econômica de implantação do sistema.

A utilização de etiquetas passivas contribui muito com o custo de implantação, pois além de ter um preço menor em relação às etiquetas ativas, a tendência mundial é que todos os itens já possuam etiquetas desde a fabricação, percorrendo toda a cadeia de suprimentos com a mesma etiqueta que poderá ser utilizada na arquitetura. Mesmo assim deve-se analisar sempre os custos das antenas e leitores, pois devido ao fato da utilização das etiquetas passivas a distância de leitura é menor.

A arquitetura RF-Miner proposta se demonstrou uma eficiente solução para o problema de localização *indoor*. Mesmo em ambientes reais onde podem existir prateleiras de metais e outros materiais reflexivos que reduzem significativamente a acuracidade, a implementação da arquitetura apresentou excelentes resultados. O módulo que vale destacar na arquitetura RF-Miner é o de Extração de Características, pois, as técnicas de derivação dos atributos contribuíram significativamente para a precisão da implementação. Também vale destacar o módulo de Mineração de Dados, principalmente no processo de Desempenho dos Atributos, o qual sempre entrega o melhor conjunto de atributos para o algoritmo classificador realizar da melhor forma o seu papel. Estas técnicas podem ser utilizadas em outras aplicações que necessitem trabalhar com classificadores e que possuam poucos atributos diretos.

O tempo de apuração dos resultados também deve ser considerado na implementação de acordo com a regra de negócio, pois o tempo de 50 segundos pode ser muito alto e inviabilizar a aplicação. Neste caso o tamanho da janela de tempo deve ser ajustado para cada cenário, mas em casos como por exemplo a biblioteca o tempo é razoável e não interfere na aplicação.

Foi possível perceber que a qualidade da base de dados é imprescindível para o sucesso da implementação, e diante disso é necessário refinar todos os processos existentes na arquitetura, desde a qualidade de instalação dos equipamentos do módulo de leitura física até o módulo de mineração de dados para testes em outras aplicações em trabalhos futuros.

Referências

Akre, J.-M., Zhang, X., Baey, S., Kervella, B., Fladenmuller, A., Zancanaro, M., and Fonseca, M. (2014). Accurate 2-D localization of RFID tags using antenna transmission power control. *2014 IFIP Wireless Days (WD)*, pages 1–6.

Hori, T., Wada, T., Ota, Y., Uchitomi, N., Mutsuura, K., and Okada, H. (2008). A multi-sensing-range method for position estimation of passive RFID tags. *Proceedings - 4th IEEE International Conference on Wireless and Mobile Computing, Networking and Communication, WiMob 2008*, pages 208–213.

Huang, C. H., Lee, L. H., Ho, C. C., Wu, L. L., and Lai, Z. H. (2015). Real-time RFID indoor positioning system based on kalman-filter drift removal and heron-bilateration location estimation. *IEEE Transactions on Instrumentation and Measurement*, 64(3):728–739.

Nakamori, E., Tsukuda, D., Fujimoto, M., Oda, Y., Wada, T., Okada, H., and Mutsuura, K. (2012). A new indoor position estimation method of RFID tags for continuous moving navigation systems. *2012 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2012 - Conference Proceedings*, (November).

Ting, S. L., Kwok, S. K., Tsang, A. H. C. A., and Ho, G. G. T. S. (2011). The Study on Using Passive RFID Tags for Indoor Positioning. *International Journal of . . .* , 3(1):9–15.

Uchitomi, N., Inada, A., Fujimoto, M., Wada, T., Mutsuura, K., and Okada, H. (2010). Accurate indoor position estimation by Swift-Communication Range Recognition (S-CRR) method in passive RFID systems. *2010 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2010 - Conference Proceedings*, (September):15–17.

Zhang, X. b., Akre, J.-M. b., Baey, S. b., Fladenmuller, A. b., Kervella, B. b., Zancanaro, M. b. c., and Fonseca, M. (2015). Towards localization of RFID tags based on experimental analysis of RSSI. *IFIP Wireless Days*, 2015-Janua(January).

Um Mecanismo de Controle de Demanda no Provimento de Serviços de IoT Usando CoAP

Jasiel G. Bahia[1] **e Miguel Elias M. Campista**[1]

[1]Universidade Federal do Rio de Janeiro – PEE/COPPE/GTA
Caixa Postal 68.504 – 21.945-970 – Rio de Janeiro – RJ – Brasil

{jasiel,miguel}@gta.ufrj.br

Abstract. *The Internet of Things (IoT) challenges network scalability, given the huge number of connected devices. Consequently, new protocols have been proposed, being CoAP (Constrained Application Protocol) one of the most important for the application layer. In this work, we present a leading scalability and performance analysis of CoAP in a typical IoT industrial scenario to show the influence of network configuration parameters on the protocol performance. Moreover, we propose a mechanism for demand control and selection of observers. The proposed mechanism is based on the radio duty-cycle at the server and its energy consumption for operation mode switching while delivering services. The mechanism is evaluated in a simulator designed for IoT (Cooja) and the results obtained show a significant reduction of energy consumption at the server compared with traditional CoAP.*

Resumo. *A Internet das Coisas (Internet of Things - IoT) desafia a escalabilidade em rede, dado o enorme número de dispositivos interconectados. Consequentemente, novos protocolos vêm sendo propostos, sendo o CoAP (Constrained Application Protocol) um dos principais para a camada de aplicação. Este trabalho apresenta uma análise pioneira de desempenho e escalabilidade do CoAP em um cenário industrial típico de IoT para demostrar a influência dos parâmetros de configuração da rede no desempenho do protocolo. Mais ainda, este trabalho propõe um mecanismo de controle de demanda e de seleção de observadores. O mecanismo proposto baseia-se no ciclo de trabalho do rádio do servidor e no seu consumo de energia para definir o modo de operação no provimento dos serviços. O mecanismo é avaliado em um simulador específico de IoT (Cooja) e os resultados mostram uma redução significativa no consumo de energia do servidor em comparação ao CoAP tradicional.*

1. Introdução

Bilhões de pessoas utilizam a Internet no mundo para aplicações que vão desde navegação *Web* até interação através de redes sociais [Group 2016]. Ao mesmo tempo em que o número de pessoas conectadas à Internet cresce, a tecnologia embarcada em dispositivos eletrônicos evolui, possibilitando que objetos do cotidiano também sejam habilitados com recursos de comunicação e processamento. Essa evolução leva a um novo paradigma de utilização da Internet, onde os objetos também se conectam e, assim, se transformam em produtores ou consumidores de dados. Esse paradigma, conhecido como Internet das Coisas (IoT - *Internet of Things*) [Al-Fuqaha et al. 2015], já é uma realidade

na indústria, sendo difundido como a quarta revolução industrial sob nomenclaturas como Indústria 4.0, Indústria Conectada e Fábrica Inteligente [Pwc 2016]. A possibilidade de obter informações mais precisas de ambientes e ativos industriais favorece a tomada de decisões e ações mais efetivas, promovendo economia, eficiência e segurança.

A escalabilidade, adaptabilidade, eficiência energética e segurança são requisitos fundamentais de IoT. Na Internet das Coisas, os protocolos devem ser compatíveis com as limitações dos dispositivos, de modo a proporcionar um consumo eficiente da energia e dos recursos de processamento e armazenamento. Com relação ao consumo de energia, é importante otimizar o ciclo de trabalho (*duty-cycle*) do sistema de comunicação, pois esse é o que mais demanda energia do dispositivo. Devido a essas características, os protocolos empregados na Internet convencional não são adequados para uso em dispositivos de IoT [Al-Fuqaha et al. 2015]. Como parte do desenvolvimento de novos mecanismos, a análise de desempenho ante cenários típicos de IoT é necessária para que se verifique o atendimento aos requisitos. A imensa quantidade de dispositivos conectados [Cisco 2016] provoca um aumento substancial no fluxo de dados e desafia o potencial de escalabilidade das redes, especialmente quando compostas por dispositivos limitados. Os mecanismos que não consideram essas restrições podem provocar o rápido esgotamento dos recursos da rede e da energia dos dispositivos, comprometendo a disponibilidade dos serviços.

As propostas de protocolos para IoT, comumente, não realizam análise de desempenho em cenários típicos. De modo geral, os protocolos são ou avaliados qualitativamente ou por meio de experimentos que não consideram as limitações dos dispositivos, como é o caso dos trabalhos apresentados em [Thangavel et al. 2014] e [Talaminos-Barroso et al. 2016]. Nesses trabalhos, as soluções propostas para aumentar a escalabilidade e reduzir o consumo de energia envolvem acordo entre cliente e servidor sobre os critérios para o provimento dos serviços, porém não garantem a disponibilidade dos serviços para clientes prioritários. Essa disponibilidade é essencial em ambientes como os industriais, pois permite o funcionamento correto dos sistemas, o gerenciamento eficiente dos ativos e promove segurança operacional.

Este trabalho visa aumentar o desempenho e a escalabilidade de servidores CoAP (*Constrained Application Protocol*) [Shelby et al. 2014] no provimento de serviços em um cenário industrial típico de IoT. O CoAP foi escolhido por atender aos requisitos das redes IoT e possuir funcionalidades úteis para o contexto industrial, além de ser um dos protocolos de IoT mais difundidos na literatura. O primeiro objetivo deste trabalho é mostrar a influência dos parâmetros de configuração da rede no desempenho do servidor CoAP, considerando como métricas vazão, perdas, quantidade de pacotes CoAP gerados e consumo de energia. O segundo e principal objetivo é propor um mecanismo de controle de demanda e de seleção de observadores, baseando-se no ciclo de trabalho do rádio do servidor e no seu consumo de energia para estabelecer dinamicamente o modo de provimento dos serviços. Esse mecanismo aumenta a eficiência da rede, em termos de escalabilidade e consumo de energia no servidor, além de garantir a disponibilidade dos serviços para os clientes prioritários. Os experimentos foram realizados no Cooja [Thingsquare 2016], o simulador da plataforma de desenvolvimento do Contiki OS, desenvolvido para o teste de redes IoT. Considerou-se um cenário típico IoT na indústria, onde os dispositivos possuem recursos limitados e interagem entre si para realizar funções de controle. O mecanismo proposto é aplicável a qualquer cenário de IoT onde haja a ne-

cessidade de garantir a disponibilidade de um dado servidor para clientes prioritários. Os resultados dos experimentos mostram que o uso do CoAP com o mecanismo proposto reduz significativamente o consumo de energia do servidor quando comparado ao uso tradicional do CoAP, sendo esse um requisito fundamental para o aumento do tempo de vida do servidor. Com o uso do mecanismo o servidor consegue garantir a disponibilidade dos serviços para os clientes prioritários de acordo com as especificações.

O restante deste artigo está organizado da seguinte forma. A Seção 2 apresenta uma arquitetura típica da Internet das Coisas e o funcionamento do CoAP. Na Seção 3, os trabalhos relacionados são listados com ênfase nos protocolos de comunicação para IoT. Na Seção 4, o funcionamento do mecanismo de controle de demanda proposto é apresentado e, a seguir, na Seção 5, as condições de análise do mecanismo são descritas. A análise dos resultados obtidos nos experimentos é discutida na Seção 6, concluindo na Seção 7 com um resumo dos resultados e sugestões de trabalhos futuros.

2. Arquitetura IoT e Protocolo CoAP

Em face das limitações dos dispositivos de IoT, as soluções empregadas na Internet convencional não são compatíveis com os novos cenários introduzidos pelas redes LLN (*Low-power and Lossy Networks*) [Granjal et al. 2015]. Sendo assim, uma nova pilha de protocolos vêm sendo desenvolvida com o objetivo de fomentar aplicações futuras de IoT [Palattella et al. 2013]. Dentro de uma arquitetura orientada a serviço [Xu et al. 2014], a camada de aplicação tem a função de apresentar a informação ao usuário através de uma interface amigável. De acordo com os requisitos da aplicação, a camada de serviço realiza a descoberta e composição de serviços, fazendo uso da infraestrutura da rede IoT para atender às solicitações. Os pedidos podem ser gerados por meio de protocolos padrão de serviços *Web* adaptados às restrições dos dispositivos de IoT, como é o caso do CoAP.

O CoAP é um protocolo de comunicação desenvolvido para dispositivos com recursos limitados e redes LLN [Shelby et al. 2014]. Esses dispositivos possuem baixa capacidade de processamento e armazenamento, enquanto que as redes LLN apresentam alta taxa de erro e vazão típica de 10 kb/s. O CoAP foi projetado para aplicações M2M (*Machine-to-Machine*), como automação industrial. O CoAP fornece ainda um modelo de interação pedido/resposta fim-a-fim que suporta descoberta de serviços e inclui conceitos da *Web* como URI (*Uniform Resource Identifier*) e tipos de dados da Internet. Além disso, o CoAP pode ser traduzido para o HTTP, oferecendo suporte a multicast e comunicação assíncrona, com baixa sobrecarga e simplicidade para ambientes com recursos limitados. O CoAP pode funcionar utilizando *proxy* e *cache*, permitindo o acesso aos recursos CoAP via HTTP de maneira uniforme. A segurança no CoAP é estabelecida por meio da camada de transporte DTLS (*Datagram Transport Layer Security*) [Shelby et al. 2014].

O modelo de interação do CoAP é similar ao cliente/servidor do HTTP. Porém, interações M2M tipicamente possibilitam a um dispositivo agir simultaneamente como cliente e como servidor. Uma requisição CoAP é enviada pelo cliente solicitando uma ação (de acordo com o código do método) sobre um recurso (identificado pela URI) do servidor. O servidor então responde com o código de resposta, podendo incluir uma representação do recurso (p. ex., valor da temperatura). Diferentemente do HTTP, o CoAP lida com essas transações assincronamente por meio de transporte orientado a da-

tagrama, utilizando o UDP. Isso é feito logicamente usando uma camada de mensagens que suporta confiabilidade opcional (com *backoff* exponencial). Já retransmissões e reordenamento são implementados na própria camada de aplicação, excluindo a necessidade do TCP e permitindo o uso do protocolo em dispositivos limitados.

O CoAP define quatro tipos de mensagens: CON (*Confirmable*), NON (*Non-confirmable*), ACK (*Acknowledgement*) e RST (*Reset*). Os códigos dos métodos e das respostas incluídos nas mensagens caracterizam o tipo de mensagem a ser utilizada nos pedidos e respostas. Os pedidos podem ser feitos com mensagens CON (com confirmação) e NON (sem confirmação) e as respostas são enviadas como mensagens ACK, portando o conteúdo solicitado. A mensagem RST é enviada quando o servidor não consegue processar uma mensagem. O CoAP pode ser visto como um protocolo de duas camadas: uma camada de mensagens, usada para lidar com as interações assíncronas sobre UDP, e outra camada usada para lidar com interações de pedido/resposta. Essas duas camadas virtuais são integradas no cabeçalho das mensagens CoAP. Cada mensagem contém um identificador usado para detectar duplicatas e para prover confiabilidade. A confiabilidade é alcançada pelo envio de mensagens CON. Uma mensagem CON é retransmitida usando um tempo de estouro padrão e *backoff* exponencial entre retransmissões até que o recipiente envie uma mensagem ACK com o mesmo identificador da mensagem CON original. Quando o recipiente não consegue processar a mensagem CON, este responde com uma mensagem RST no lugar do ACK. Uma mensagem que não exige transmissão confiável pode ser enviada por meio de uma mensagem NON. Embora não haja uma resposta com ACK para a mensagem NON, ela possui identificação para a detecção de duplicata. Se o servidor receber o pedido, mas não puder responder imediatamente, ele envia um ACK sem conteúdo para que o cliente não continue retransmitindo. Tão logo a resposta com o conteúdo esteja pronta, o servidor envia uma nova mensagem CON, contendo o conteúdo solicitado, e o cliente confirma o recebimento com um ACK. O CoAP utiliza um subconjunto dos métodos do HTPP (GET, PUT, POST e DELETE), os quais funcionam de maneira similar, porém ajustados para uso em redes LLN.

Uma aplicação de IoT frequentemente envolve um dispositivo transmitindo informação sobre os seus sensores a outros dispositivos clientes. O CoAP suporta uma funcionalidade chamada de Observação [Hartke 2014], onde o cliente pode registrar-se como observador de um recurso (sujeito) do servidor através de um GET modificado. O servidor estabelece um relacionamento de observação entre o cliente e o recurso, sendo que cada recurso tem a sua própria lista de observadores. Cada cliente só pode se registrar uma única vez em uma lista. Essa funcionalidade reduz o congestionamento na fila de entrada do servidor, pois evita a necessidade de o cliente demandar constantemente o servidor ou ter que manter uma sessão aberta como no caso do HTTP sobre TCP. Essa extensão do CoAP permite que os clientes observem as mudanças nos recursos de seu interesse sem gerar novos pedidos, reduzindo a sobrecarga, o uso de banda, a latência e aumentando a confiabilidade em comparação com o HTTP [Ludovici et al. 2012]. Porém, é importante lembrar que o tamanho da lista de observadores está limitado à capacidade de armazenamento do dispositivo. Incorporando ao CoAP/Observação as funcionalidades de *cache* e *proxy*, é possível aumentar a escalabilidade do sistema através do registro de observadores em dispositivos intermediários, otimizando, assim, o uso dos recursos do servidor. Esse recurso de Observação do CoAP, é muito importante para a escalabilidade e para o controle do consumo de energia, garantindo um maior tempo de disponibilidade

do servidor. A Figura 1 ilustra a comunicação com o servidor quando o cliente interage sob demanda e quando o cliente se registra como observador.

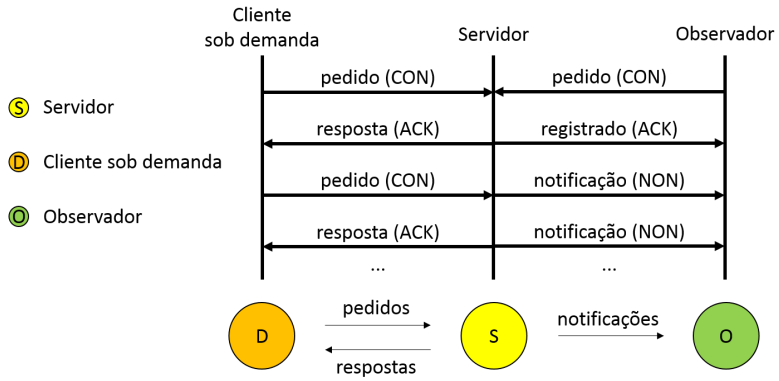


Figura 1. Interação cliente/servidor com e sem Observação.

3. Trabalhos Relacionados

O desenvolvimento de protocolos adaptados a dispositivos com recursos limitados tem sido objeto de grande interesse no meio científico. Nesta seção, são apresentados algumas propostas e análises de desempenho de protocolos de comunicação para IoT.

Sutaria et al. [Sutaria and Govindachari 2013] comparam o HTTP e o CoAP em termos de tamanho de pacote e consumo de energia por pedido. No artigo, é mostrado que um pedido para um servidor HTTP implementado no Contiki OS, quando feito com o CoAP, usa 154 bytes; quando feito com o HTTP, usa 1451 bytes, consumindo 0,774 mW e 1,333 mW, respectivamente. Já Colitti et al. [Colitti et al. 2011] mostram que o CoAP gasta menos tempo na transmissão do pedido e consome menos energia que o HTTP. Dois dos protocolos de comunicação mais difundidos para dispositivos limitados são o CoAP e o MQTT (*Message Queue Telemetry Transport*). O MQTT [Banks and Gupta 2014], desenvolvido pela IBM, é um protocolo do tipo *publish/subscribe*, assim como o CoAP, porém utiliza o TCP. O uso do TCP não é indicado para redes LLN e dispositivos mais limitados, visto que exige mais memória e processamento, além de estar sujeito à instabilidade causada pelas perdas. O MQTT não suporta diretamente serviços *Web*, pois, diferentemente do CoAP, não se consegue rastrear as transações de pedido/resposta entre cliente e servidor fim-a-fim. Para publicação dos serviços e acesso aos mesmos, o MQTT utiliza o *broker*, um dispositivo concentrador que coleta os dados dos recursos e os envia para a infraestrutura do servidor. Dessa forma, o MQTT não permite a comunicação fim-a-fim entre cliente e servidor. Além disso, os endereços utilizados pelo MQTT são definidos por longas cadeias de caracteres, sendo incompatível com a tecnologia de rede IEEE 802.15.4. O MQTT-SN [Hunkeler et al. 2008] é uma alternativa que utiliza o UDP e tenta prover uma abstração para comunicação assíncrona.

Com relação a análise de desempenho de protocolos para IoT, Thangavel et al. [Thangavel et al. 2014] analisam o desempenho do MQTT e do CoAP em termos de atraso fim-a-fim e consumo de banda, quando usados em conjunto do *middleware* proposto para prover interoperabilidade. Os resultados revelam que mensagens MQTT possuem menor atraso do que mensagens CoAP quando há baixa taxa de perda e que

a situação se inverte quando há alta taxa de perda. Quando o tamanho da mensagem é pequeno e a taxa de perda é menor ou igual a 25%, o CoAP gera menos tráfego adicional para assegurar a confiabilidade da mensagem. Kovatsch et al. [Kovatsch et al. 2011] apresentam uma implementação do CoAP para o Contiki OS, e o ContikiMAC, um gerenciador do ciclo de trabalho do rádio do dispositivo. O mecanismo é avaliado pela variação no consumo de energia e no tempo de resposta, mostrando que o ContikiMAC reduz o consumo de energia, mas aumenta a latência da rede. Zhang et al. [Zhang and Li 2014] analisam o desempenho do ContikiRPL, uma implementação do protocolo de roteamento RPL para o Contiki OS, observando o seu comportamento em diferentes arranjos da rede. Assim como neste trabalho, Zhang et al. consideram um cenário típico de IoT contendo dispositivos com recursos limitados.

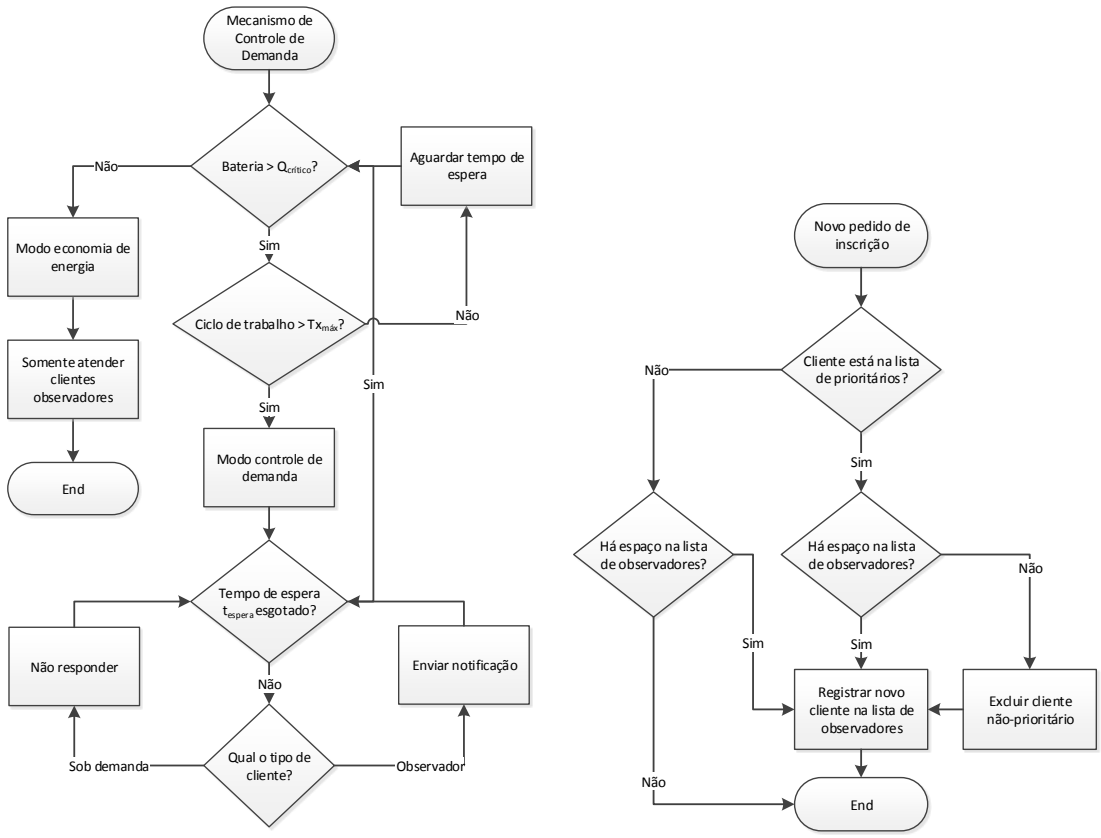
Pelo levantamento feito, não foi identificado nenhum trabalho de análise de desempenho e escalabilidade de servidores CoAP em redes IoT compostas por dispositivos com recursos limitados. Além disso, não foram encontrados outros trabalhos que mencionassem mecanismos de controle de demanda para servidores CoAP, sendo essa a principal contribuição deste trabalho.

4. Mecanismo de Controle de Demanda Proposto

Esta seção apresenta o mecanismo de controle de demanda e seleção de observadores proposto, que tem por objetivo garantir a disponibilidade de servidores CoAP para clientes prioritários em um cenário industrial IoT.

Considerando o caso em que os pedidos CoAP gerados pelos clientes sob demanda tenham uma periodicidade t_n, o número de pedidos gerados (demanda D) por um cliente, pode ser estimado, fazendo $D = \frac{t_{op}}{t_n}$, onde t_{op} é o tempo total de operação do dispositivo cliente. Já para um cliente registrado na lista de observadores de um recurso do servidor, só é gerado um pacote de pedido CoAP. Como observador, esse cliente só recebe pacotes CoAP de notificação sobre o recurso em observação com periodicidade t_p. O tamanho máximo da lista de observadores depende do espaço na memória do servidor e a redução de demanda depende do número de clientes observadores (p) registrados na lista e da relação entre t_n e t_p. Assim, verifica-se que a funcionalidade de Observação do CoAP permite uma redução na demanda, porém não exerce controle sobre a demanda dos clientes não-observadores (sob demanda). O mecanismo de controle de demanda proposto promove a redução do consumo de energia do servidor, agindo sobre o ciclo de trabalho do rádio na transmissão de pacotes CoAP. O servidor monitora o ciclo de trabalho do rádio e, quando este ultrapassa um limiar Tx_{max}, o mecanismo faz o servidor entrar em modo de controle de demanda, somente usando o rádio para notificar os observadores. Nesse modo de operação, o mecanismo busca reduzir o ciclo de trabalho na transmissão, fazendo-o retornar a um valor inferior ao limiar definido. Isso é feito da seguinte forma: enquanto o tempo de espera t_{espera} não é completado, 1) Quando receber um pedido de cliente sob demanda, não responder; 2) Quando for um pacote para cliente observador, enviar a notificação. Caso a carga da bateria esteja abaixo do nível crítico $Q_{crítico}$, o mecanismo faz o servidor entrar em modo de economia de energia. Nesse modo de operação, o rádio só é utilizado para atender a clientes observadores. Através desse mecanismo, é possível controlar o consumo de energia do servidor CoAP de forma a garantir a sua disponibilidade para os clientes observadores.

A segunda parte do mecanismo é a seleção de observadores dos recursos do servidor CoAP. Em qualquer cenário de IoT que contenha sistemas clientes críticos que devam ser priorizados, é importante garantir que esses sistemas consigam monitorar os recursos essenciais para o seu funcionamento. O mecanismo de seleção tem por objetivo selecionar quais os clientes poderão ser registrados na lista de observadores de um recurso do servidor CoAP. Para tanto, o servidor deve armazenar, para cada recurso, os endereços dos clientes prioritários. Sempre que houver um novo pedido de registro na lista de observadores, o endereço de origem do pedido é comparado com os endereços na lista de clientes prioritários do recurso. Caso esteja na lista, o cliente é registrado como observador; caso contrário, é verificado se há espaço na lista de observadores para registrá-lo. Caso um cliente prioritário deseje registrar-se como observador, mas a lista estiver cheia, o servidor deve excluir clientes não-prioritários da lista para incluir o cliente prioritário. Dessa forma, o mecanismo garante uma ordem de prioridade para monitoramento do recurso do servidor e, em conjunto com o mecanismo de controle de demanda, garante a disponibilidade para os clientes prioritários. Como já foi dito, o espaço em memória do dispositivo limita o tamanho máximo da lista de observadores de um recurso. Portanto, o dispositivo escolhido como servidor deve possuir capacidade de armazenamento suficiente para atender a todos os clientes prioritários ou o recurso deve ser disponibilizado por mais servidores. Isso deve fazer parte do projeto de automação do sistema. A Figura 2 mostra o funcionamento do mecanismo de controle de demanda e seleção de observadores.



(a) Algoritmo de controle de demanda. (b) Algoritmo de seleção de observadores.

Figura 2. Funcionamento do mecanismo proposto.

Para configurar o mecanismo, é preciso definir o ciclo de trabalho máximo (Tx_{max}) desejado para o rádio do servidor. O ciclo de trabalho representa a fração do período de tempo em que o rádio é efetivamente utilizado, ou seja, $Tx = \frac{t_{TX}}{t_{op}}$, sendo t_{TX} o tempo total de uso do rádio e t_{op} o tempo total de operação do servidor. Para definir Tx_{max}, considera-se a energia total disponível na bateria (Q_{max}), o tempo total de operação desejado (t_{op}), o período de notificação necessário (t_p), a potência consumida pelo rádio para cada notificação (P_{tx}), o respectivo tempo de uso do rádio (t_{tx}) e o número de clientes observadores (p). A energia necessária para enviar uma notificação é $Q_{tx} = (P_{tx} \times t_{tx})$ e o número de notificações por observador é $D_p = \frac{t_{op}}{t_p}$. Assim, o produto $p \times D_p \times Q_{tx}$ fornece a energia total consumida pelo rádio, ou seja, $Q_{max} = \left(p \times \frac{t_{op}}{t_p} \times P_{tx} \times t_{tx}\right)$. O valor de t_{TX} é o produto do número de notificações ($p \times D_p$) pelo tempo de uso do rádio para cada notificação (t_{tx}), ou seja, $t_{TX} = \left(p \times \frac{t_{op}}{t_p} \times t_{tx}\right)$. Dessa forma, obtém-se que $Tx_{max} = \frac{Q_{max}}{P_{tx} \times t_{op}}$. Note que quanto maior o tempo de operação desejado, menor deve ser ciclo de trabalho para a mesma energia disponível.

Os valores de Tx_{max} e de t_{espera} são os parâmetros de configuração do mecanismo que definem o seu desempenho e o grau de economia de energia. Quanto menor o t_{espera}, maior a taxa de verificação do ciclo de trabalho e mais rápido é o retorno ao atendimento dos clientes sob demanda quando o ciclo de trabalho do rádio é normalizado. Outro parâmetro que deve ser definido é o nível crítico de carga da bateria $Q_{critico}$, o qual depende do tempo residual de operação desejado para o servidor. O servidor deve continuar operando durante tempo suficiente para que as ações de manutenção sejam tomadas (p. ex., troca da bateria), sem comprometer o funcionamento do sistema cliente.

5. Análise do Mecanismo de Controle de Demanda

Esta seção apresenta as condições de análise do mecanismo de controle de demanda para servidores CoAP, as métricas utilizadas e o cenário considerado. Antes, porém, é necessário introduzir o ambiente de simulação utilizado nos experimentos.

5.1. Ambiente de simulação

A ferramenta utilizada nos experimentos foi o Cooja, que é o simulador de redes LLN da plataforma de desenvolvimento do Contiki OS [Thingsquare 2016]. O Contiki é um sistema operacional de código aberto desenvolvido especialmente para dispositivos com recursos limitados, usados tipicamente em Redes de Sensores Sem Fio e em redes IoT. O Contiki suporta inteiramente os padrões IPv4 e IPv6, além dos padrões criados recentemente para redes LLN, como 6LoWPAN, RPL e CoAP. O Contiki também possui mais de uma implementação da camada MAC para dispositivos de baixa potência (IEEE 802.15.4), sendo o ContikiMAC a principal.

A plataforma de desenvolvimento do Contiki inclui o simulador Cooja, que permite simular o comportamento de aplicações de IoT em uma ampla gama de cenários. O simulador apresenta diversos relatórios a respeito do hardware (sinalização de rádio, alcance, potência, energia, luzes, botões, etc.) e das comunicações que ocorrem entre os dispositivos da rede (mensagens, protocolos, endereços, temporização, etc.). Esses relatórios podem ser visualizados em tempo de simulação e salvos para análise posterior.

Os dispositivos usados na simulação são emulados a partir dos seus correspondentes comerciais, o que aumenta a proximidade do comportamento simulado com aquele que se espera encontrar em experimentos com dispositivos reais.

5.2. Experimentos

Os experimentos foram realizados considerando um cenário industrial típico de IoT, onde a rede é composta por dispositivos que assumem o papel de servidor ou cliente de acordo com a aplicação e a complexidade do sistema. No papel de servidor, os dispositivos disponibilizam seus recursos para os dispositivos clientes, através da Internet, podendo atender a clientes internos e externos. Considera-se que, para cada rede local, exista a figura do roteador de borda, responsável por conectar os dispositivos de sua rede entre si e a Internet, divulgando os recursos disponíveis. Um exemplo de cenário industrial típico é o sistema de monitoramento e controle de dutos de recebimento de petróleo em terminais de armazenamento. Esse sistema faz o monitoramento da vazão (recurso) no duto de petróleo que alimenta tanques, bombas de transferência e outros dutos. A prioridade no monitoramento deve ser para os sistemas diretamente afetados pelo duto de petróleo (observadores prioritários). Enquanto o servidor estiver com carga normal na bateria, ele pode atender a consultas de outros sistemas (clientes sob demanda) dentro de um limite de consultas por unidade de tempo (modo de controle de demanda). Porém, caso a bateria esteja com carga baixa, o servidor só atende aos sistemas críticos (modo de economia de energia). A Figura 3 ilustra a rede IoT considerada nos experimentos.

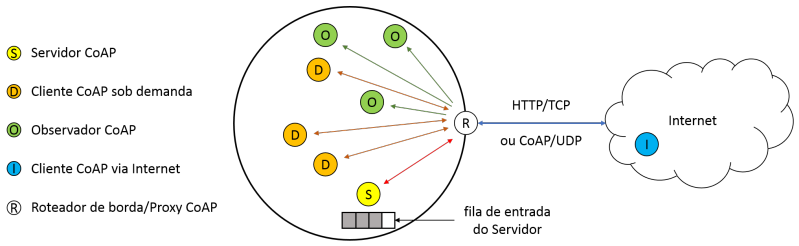


Figura 3. Um cenário industrial típico de IoT.

Como cenário base considerou-se uma rede composta por um nó no papel de roteador de borda executando o protocolo RPL, um nó no papel de servidor CoAP e um número variável de nós como clientes CoAP. A comunicação entre os clientes e o servidor ocorre sempre com a intermediação do roteador de borda , tornando indiferente ao servidor a origem do pedido, seja de cliente interno ou externo. . Para cada configuração, a duração da simulação foi de 10 minutos.

As simulações foram executadas em uma máquina virtual rodando o Contiki 3.0. A tecnologia utilizada nos dispositivos da rede foi o módulo TMote Sky (Mote IV), que utiliza a tecnologia de rede IEEE 802.15.4 e possui CPU de 16 bits, 10kB de RAM e 48 kB de memória Flash. Para reduzir o uso de memória de programa dos dispositivos, foi utilizada a seguinte configuração: NullRDC, NullMAC e tamanho de buffer uIP de 256 bytes. O tamanho da fila do servidor (buffer) deve ser definido previamente na configuração do dispositivo, de acordo com a sua capacidade de memória. Neste trabalho, o dispositivo foi configurado para permitir, no máximo, 4 mensagens CoAP em espera na fila do servidor.

Nos experimentos, os clientes CoAP podem ser clientes observadores (notificados periodicamente) ou clientes sob demanda. Para cada configuração, a periodicidade de notificações (t_p) do servidor para os observadores foi mantida, enquanto que os parâmetros variáveis foram: número total de clientes (n), periodicidade de requisições do cliente sob demanda (t_n) e número de observadores (p). O número máximo de clientes no simulador é restrito, dada a limitação do próprio padrão CoAP [Shelby et al. 2014]. Em vista disso, o número de clientes total foi variado entre $n = 1$ e $n = 10$.

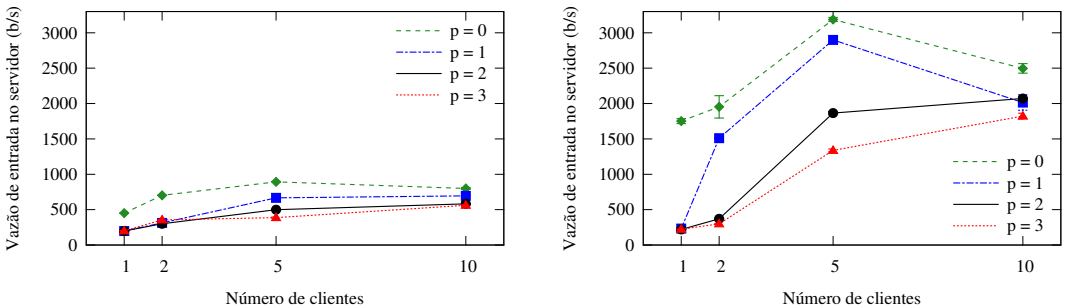
As métricas de desempenho utilizadas na análise do servidor CoAP foram: quantidade de pacotes CoAP gerados, perdas, vazão de dados e consumo de energia. Os intervalos de confiança mostrados nos gráficos são de 95%, representados por barras verticais.

6. Análise dos Resultados

Nesta seção, a influência dos parâmetros de configuração da rede no desempenho do servidor CoAP é discutida. Em seguida, os resultados dos experimentos utilizando o mecanismo de controle de demanda proposto são apresentados.

6.1. Influência dos parâmetros de configuração no desempenho do servidor CoAP

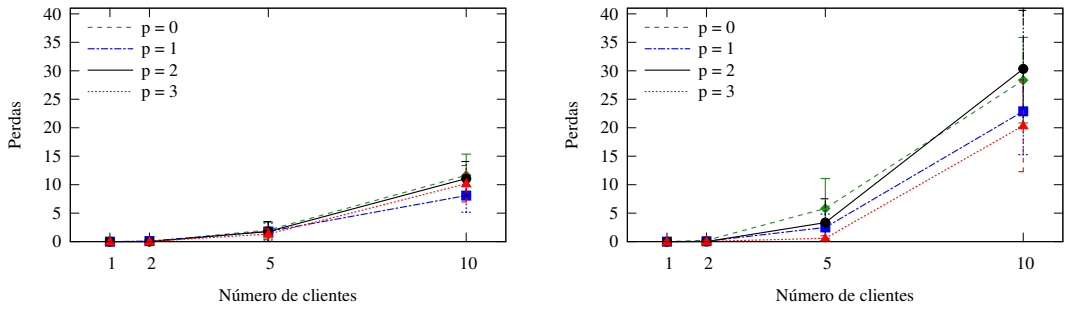
Vazão de dados: Avaliando a vazão de dados no servidor, observa-se que ela chega a atingir cerca de 3200 b/s para ($n = 5; p = 0; t_n = t_p/2$), conforme Figura 4(b). Essa taxa máxima é compatível com aquela esperada para redes LLN (Seção 2). Porém, a limitação da fila de entrada e da capacidade de processamento do servidor provoca perdas de pacotes à medida que a diferença entre o número de clientes não-observadores e observadores ($n - p$) e a demanda aumentam, reduzindo, assim, a vazão. A partir da análise do tráfego da rede, percebe-se que as perdas ocorrem sobretudo quando, antes do servidor responder a um dado pedido, o número de pedidos recebidos excede o tamanho da fila ou quando o servidor demora mais que o tempo de estouro do cliente para responder. A Figura 4 mostra também que o fluxo de dados na entrada do servidor, para o mesmo n, é reduzido com o aumento de p, especialmente quando a periodicidade de pedidos é menor que a de notificações. Como já comentado, o cliente observador, uma vez inscrito no servidor, não gera mais requisições para o recurso em observação, reduzindo assim a carga sobre a entrada do servidor. Logo, o tamanho da lista de observadores (uso de memória), o tamanho da fila de entrada no servidor (uso de memória) e a capacidade de processamento têm papel fundamental no desempenho do servidor com o aumento da demanda.



(a) $t_n = 2t_p$ (b) $t_n = t_p/2$

Figura 4. Vazão de dados no servidor.

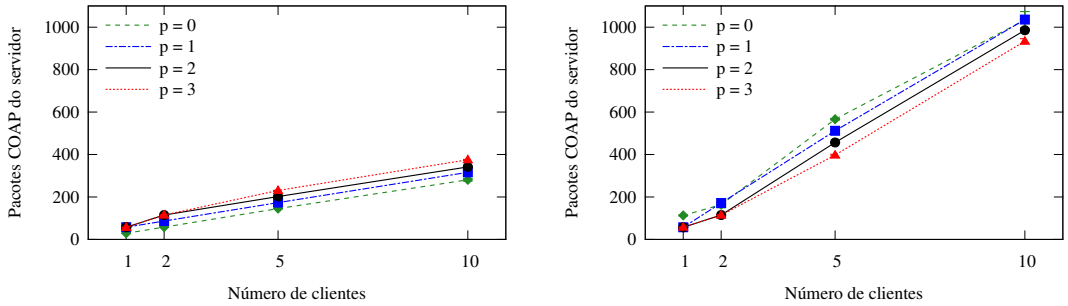
Perdas de pacotes: Na Figura 5, observa-se que o número médio de perdas se torna maior à medida que a demanda de clientes não-observadores aumenta (aumento da diferença $n - p$ e redução de t_n), chegando a representar um pouco mais de 3% dos pedidos para $(n = 10; p = 2; t_n = t_p/2)$. Pelo mesmo motivo da redução da vazão, o aumento das perdas é consequência da limitação do tamanho da fila de entrada do servidor utilizado (4 pacotes CoAP no máximo). Uma solução, portanto, é utilizar um servidor com maior capacidade de memória, permitindo um tamanho de fila maior.



(a) $t_n = 2t_p$ (b) $t_n = t_p/2$

Figura 5. Perdas de pacotes CoAP no servidor.

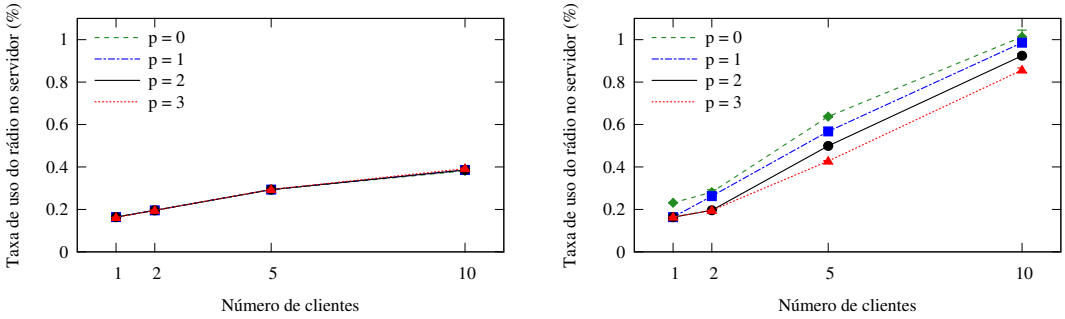
Número de pacotes CoAP: A Figura 6 mostra a evolução no número de pacotes CoAP gerados pelo servidor com o aumento da demanda (aumento de $n - p$ e redução de t_n). Observa-se que, para pedidos sob demanda com periodicidade $t_n = 2t_p$, o valor de p dita a quantidade de pacotes gerados. Porém, para $t_n = t_p/2$, o número de respostas a pedidos sob demanda é maior que o número de notificações por intervalo de tempo. Logo, a quantidade de clientes sob demanda é que dita a quantidade de pacotes gerados pelo servidor. O mecanismo de controle de demanda deve atuar nesse tipo de situação, a fim de manter o uso do rádio do servidor dentro dos limites especificados.



(a) $t_n = 2t_p$ (b) $t_n = t_p/2$

Figura 6. Pacotes CoAP transmitidos pelo servidor.

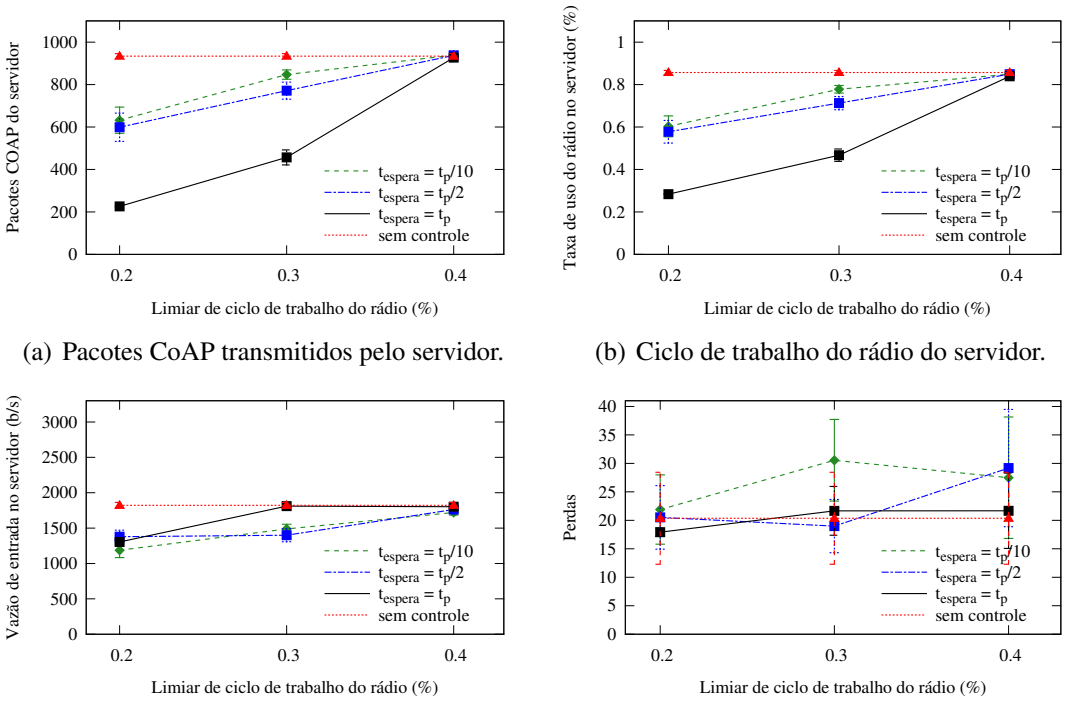
Consumo de energia: Os resultados apresentados na Figura 7 representam a percentagem de tempo em que o servidor utilizou o rádio para receber ou transmitir dados. A taxa de uso do rádio foi usada para avaliar o consumo de energia porque, além de serem diretamente proporcionais (vide folha de dados do dispositivo), a análise do ciclo de trabalho permite avaliar o quanto o sistema acompanha os parâmetros de configuração do



(a) $t_n = 2t_p$ (b) $t_n = t_p/2$

Figura 7. Ciclo de trabalho do rádio do servidor.

mecanismo proposto. Observa-se que, para $t_n = 2t_p$, o consumo de energia é praticamente o mesmo para as diversas configurações de rede. Porém, quando $t_n = t_p/2$, os clientes sob demanda se tornam os maiores contribuintes para o uso do rádio do servidor. O mecanismo de controle de demanda visa justamente impedir que o servidor seja sobrecarregado pelo aumento da demanda dos clientes não-observadores.



(a) Pacotes CoAP transmitidos pelo servidor. (b) Ciclo de trabalho do rádio do servidor.

(c) Vazão de dados no servidor. (d) Perdas de pacotes CoAP no servidor.

Figura 8. Desempenho do Mecanismo proposto.

6.2. Desempenho do Mecanismo Proposto de Controle de Demanda

Conforme observado na análise de desempenho do servidor CoAP, considerando o cenário descrito com três clientes prioritários ($p = 3$), o pior caso de demanda é aquele com menor período entre pedidos e maior número de clientes sob demanda. Sendo assim, o mecanismo de controle de demanda foi analisado para o caso ($n = 10; t_n = t_p/2; p =$

3). Os parâmetros de configuração do mecanismo foram Tx_{max} e t_{espera}, tomando como referência o ciclo de trabalho na transmissão encontrado para ($n = 10; t_n = 2t_p$). A Figura 8 mostra que o mecanismo conseguiu controlar a taxa de uso do rádio, reduzindo a quantidade de pacotes gerados (Figura 8(a)) e economizando a energia do servidor (Figura 8(b)). A redução de uso do rádio do servidor foi de cerca de 60% para a configuração ($Tx_{max} = 0.2; t_{espera} = t_p$). O uso do mecanismo provoca uma redução na vazão de entrada do servidor (Figura 8(c)), como esperado, porém não foi observado grande impacto no número de perdas na maioria dos casos. Isso mostra que, neste caso em específico, mesmo com o uso do mecanismo, os clientes sob demanda não foram tão prejudicados.

7. Conclusões e Trabalhos Futuros

As principais contribuições deste artigo são a análise de desempenho e escalabilidade de servidores CoAP em um cenário industrial típico de IoT e a proposta de um novo mecanismo de controle de demanda e seleção de observadores. Ressalta-se que este trabalho apresenta uma proposta inédita para economia de energia em redes que usam servidores CoAP. A escolha do CoAP para a aplicação do mecanismo proposto deve-se ao fato do CoAP, além de possuir a funcionalidade de Observação, atender aos requisitos de IoT e de redes LLN compostas por dispositivos com recursos limitados, condições fundamentais para uso no cenário considerado neste trabalho. Os resultados mostraram que o servidor CoAP sem o controle de demanda sofre degradação de desempenho à medida que a demanda aumenta, tendo sua energia consumida mais intensamente. O uso do mecanismo de controle de demanda em conjunto com a funcionalidade de Observação do CoAP aumenta a eficiência da rede, reduzindo o consumo de energia do servidor significativamente e garantindo a disponibilidade dos seus recursos para os clientes prioritários.

Como trabalhos futuros, pretende-se desenvolver um mecanismo de provimento de serviços adaptável ao contexto, considerando situações mais complexas como o uso de mais de um servidor. Sugere-se também a realização de testes reais, não só para avaliar o desempenho da proposta, mas também para verificar a representatividade do simulador Cooja. Finalmente, a análise de outros protocolos de comunicação para comparação com o CoAP poderá ser feita tomando este trabalho como referência.

Referências

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys Tutorials*, 17(4):2347–2376.

Banks, A. and Gupta, R. (2014). MQTT Version 3.1.1. *OASIS Standard*.

Cisco (2016). Cisco Visual Networking Index: Global Mobile Data Traffic Forecast. [Online: 15-November-2016].

Colitti, W., Steenhaut, K., Caro, N. D., Buta, B., and Dobrota, V. (2011). Evaluation of Constrained Application Protocol for Wireless Sensor Networks. In *Local Metropolitan Area Networks (LANMAN), 2011 18th IEEE Workshop on*, pages 1–6.

Granjal, J., Monteiro, E., and Sa Silva, J. (2015). Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues. *Communications Surveys Tutorials, IEEE*, 17(3):1294–1312.

Group, M. M. (2016). Internet World Stats. [Online: 18-November-2016].

Hartke, K. (2014). Observing Resources in CoAP. *Internet Engineering Task Force (IETF)*, RFC 7641.

Hunkeler, U., Truong, H., and Stanford-Clark, A. (2008). MQTT-S: A Publish/Subscribe Protocol for Wireless Sensor Networks. *Workshop on Information Assurance for Middleware Communications (IAMCOM)*.

Kovatsch, M., Duquennoy, S., and Dunkels, A. (2011). A Low-Power CoAP for Contiki. In *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on*, pages 855–860.

Ludovici, A., Garcia, E., Gimeno, X., and Auge, A. C. (2012). Adding QoS support for Timeliness to the Observe Extension of CoAP. In *2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 195–202.

Palattella, M., Accettura, N., Vilajosana, X., Watteyne, T., Grieco, L., Boggia, G., and Dohler, M. (2013). Standardized Protocol Stack for the Internet of (Important) Things. *Communications Surveys Tutorials, IEEE*, 15(3):1389–1406.

Pwc (2016). Industry 4.0: Building the Digital Enterprise. [Online: 15-November-2016].

Shelby, Z., Hartke, K., and Bormann, C. (2014). The Constrained Application Protocol (CoAP). *Internet Engineering Task Force (IETF)*, RFC 7252.

Sutaria, R. and Govindachari, R. (2013). Understanding The Internet Of Things. *Electronic Design Magazine*.

Talaminos-Barroso, A., Estudillo-Valderrama, M. A., Roa, L. M., Reina-Tosina, J., and Ortega-Ruiz, F. (2016). A Machine-to-Machine protocol benchmark for eHealth applications - Use case: Respiratory rehabilitation. *Computer Methods and Programs in Biomedicine*, 129:1–11.

Thangavel, D., Ma, X., Valera, A., Tan, H. X., and Tan, C. K. Y. (2014). Performance evaluation of MQTT and CoAP via a Common Middleware. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pages 1–6.

Thingsquare (2016). Contiki: The Open Source OS for the Internet of Things. [Online: 14-November-2016].

Xu, L. D., He, W., and Li, S. (2014). Internet of Things in Industries: A Survey. *Industrial Informatics, IEEE Transactions on*, 10(4):2233–2243.

Zhang, T. and Li, X. (2014). Evaluating and Analyzing the Performance of RPL in Contiki. In *Proceedings of the First International Workshop on Mobile Sensing, Computing and Communication*, MSCC '14, pages 19–24.

Rotas Veiculares Cientes de Contexto: Arcabouço e Análise Usando Dados Oficiais e Sensoriados por Usuários sobre Crimes

Frances A. Santos[1], Diego O. Rodrigues[1], Thiago H. Silva[2], Antonio A. F. Loureiro[3], Leandro A. Villas[1]

[1]Instituto de Computação, Universidade Estadual de Campinas. Campinas, Brasil
[2]Dep. de Informatica, Universidade Tecnológica Federal do Paraná. Curitiba, Brasil
[3]Dep. de Ciência da Computação, Univ. Federal de Minas Gerais. Belo Horizonte, Brasil

Abstract. *An increasing number of users have been adopting route recommendation systems, many of them motivated by the convenience that these systems bring to their traffic experiences. These systems observe the current traffic conditions in order to evaluate the fastest path. However, contextual information, such as pollution levels, weather conditions and security of the route, might not be taken into account in the recommendation process. With this in mind, we propose a framework to support context-aware route recommendation systems. Furthermore, we present possible approaches for development of two key components of this framework: (1) identification of contextual areas based on different urban data sources; and (2) identification of typical routes. The proposed framework might improve existing recommendation algorithms, or also enable the proposal of new ones. To validate this hypothesis, we used a set of routes suggested by Google Maps in the city of Curitiba to identify frequent patterns on these routes. Afterwards, some insecure zones were identified in Curitiba using data provided by the government of the city and also data generated by the citizens via their mobile devices. The obtained results showed the existence of an opportunity for route planners to provide differential services to users who desire it, which is an important step towards the development of context-aware vehicular networks.*

Resumo. *Cada vez mais usuários utilizam sistemas de recomendação de rotas, pois esse serviço torna a vida no trânsito mais prática. Tipicamente esses serviços levam em consideração a situação atual do trânsito para fornecer a rota mais rápida. No entanto, requisitos contextuais, como nível de poluição, condições climáticas e segurança da rota, podem não ser levados em consideração na recomendação. Nessa direção, propomos um arcabouço para apoiar o oferecimento de recomendação de rotas veiculares cientes de contexto. Além disso, apresentamos possíveis abordagens para a implementação de dois componentes chave desse arcabouço: (1) identificação de áreas contextuais em fontes de dados urbanos distintas; e (2) identificação de rotas frequentes. O arcabouço proposto possibilita o aprimoramento de algoritmos de recomendação existentes, bem como a proposição de novos. Com o intuito de estudar essa hipótese, utilizamos um conjunto de rotas sugeridas pelo Google Maps na cidade de Curitiba para identificar padrões frequentes nessas rotas. Em seguida, identificamos áreas de insegurança nessa mesma cidade utilizando dados abertos fornecidos pela prefeitura e dados sensoriados pelos usuários sobre ocorrências de crimes. Nossos resultados indicam que existe uma oportunidade para recomendadores de rotas oferecem serviços diferenciados para usuários que assim desejarem, sendo um passo importante para a construção de uma rede veicular sensível ao contexto de rotas.*

1. Introdução

Sistemas de recomendação de rotas estão sendo cada vez mais utilizados pelos motoristas. Alguns dos motivos que explicam essa tendência são a praticidade proporcionada e a possibilidade de auxiliar em problemas de congestionamento no trânsito, que têm se agravado em grandes cidades [Wang et al. 2014, Vo 2015]. De modo geral, esses serviços visam permitir aos usuários encontrarem a melhor rota para atingir um determinado destino a partir de um determinado ponto da cidade. Com isso, uma pergunta importante que surge é: como definir a melhor rota de um ponto a outro na cidade?

Comumente, os sistemas de recomendação de rotas veiculares, tais como, Waze[1] e Google Maps[2], consideram as condições históricas e atuais de trânsito para recomendação de rotas mais rápidas. No entanto, outros requisitos contextuais, como o nível de poluição, condições climáticas, eventos e segurança da rota, podem não ser levados em consideração na recomendação. Por exemplo, em 2015 um casal seguiu uma recomendação de rota fornecida pelo Waze e acabaram sendo baleados ao entrar na área da comunidade do Caramujo em Niterói, RJ [Rio 2015]. Episódios como esse mostram a importância da consideração de contexto inerente às rotas além de aspectos tradicionais, como velocidade atual da via, para fornecer *rotas veiculares cientes de contexto*. Desta forma, os usuários podem escolher uma rota considerando diferentes interesses pessoais.

Nessa direção, propomos um arcabouço que visa auxiliar no processo de recomendação de rotas veiculares cientes de contexto. O arcabouço é composto por dois componentes chave: (i) identificação de áreas contextuais em fontes de dados urbanos distintas; e (ii) identificação de rotas frequentes. Além de discutir esses componentes, também são apresentadas possíveis abordagens para as suas implementações. Por meio do arcabouço proposto, torna-se possível o aprimoramento de algoritmos de recomendação existentes, bem como a proposição de novos. Com o intuito de estudar essa hipótese, assim como ilustrar uma instanciação do arcabouço, foram coletados dados oficiais e sensoriados por usuários sobre incidências de crime em Curitiba, Brasil, e também rotas sugeridas pelo Google Maps nessa cidade. Observamos indícios de que quesitos de segurança não são considerados na recomendação de rotas, pois rotas que são frequentemente recomendadas passam por áreas reconhecidas por sua insegurança. Diante disso, verificamos que rotas alternativas, que não passam por essas áreas, não adicionam atrasos excessivos. Nossos resultados indicam que existe uma oportunidade para recomendadores de rotas oferecerem serviços diferenciados para usuários que assim desejarem, sendo um passo importante para a construção de uma rede veicular sensível ao contexto de rotas.

O restante deste trabalho está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 descreve o arcabouço proposto. A Seção 4 apresenta as bases de dados analisadas. A Seção 5 investiga o emprego do arcabouço em um cenário com dados reais. Por fim, a Seção 6 apresenta as conclusões e trabalhos futuros do estudo.

2. Trabalhos Relacionados

Existem diversas iniciativas que extraem informações contextuais da cidade a partir de fontes de dados heterogêneos para fornecer serviços mais especializados aos cidadãos, visando melhorar a qualidade de vida na cidade. Exemplos desses serviços incluem segurança [Lee et al. 2006, Calavia et al. 2012], sistemas de energia [Vaghefi et al. 2014, Jain et al. 2016] e planejamento de tráfego [Demiryurek et al. 2009, Melnikov et al. 2015].

[1] www.waze.com.

[2] maps.google.com.br.

Particularmente em relação ao planejamento de tráfego, um tópico que vem recebendo grande atenção dos pesquisadores é o de recomendação de rotas para veículos ou mesmo pedestres [Wang et al. 2014, Dai et al. 2015]. Para esse fim, são propostas várias abordagens para obter o "melhor" caminho a ser utilizado pelo usuário, tais como [ter Mors et al. 2010, Bader et al. 2011], onde uma das principais premissas adotadas é que o usuário deseja economizar tempo em sua viagem. Contudo, encontrar o *melhor caminho* é um problema que vai além de encontrar o *caminho mais rápido*. Isto porque o caminho mais rápido pode passar por vias inseguras ou com muita poluição do ar, o que pode resultar em experiências desagradáveis aos usuários.

Desta forma, além de reduzir o tempo gasto no trânsito, aplicações de recomendação de rotas também podem ter outros objetivos que dependem dos interesses dos usuários. O CrowdSafe [Shah et al. 2011], por exemplo, é um sistema para dispositivos móveis que permite aos usuários reportarem informações sobre crimes que eles sofreram ou presenciaram. Tais informações, são utilizadas para o sistema identificar áreas de insegurança na cidade, fornecer um serviço de recomendação de rotas ciente dos locais de insegurança e, também, para geração de estatísticas de crimes da cidade. O mapeamento de insegurança não difere entre as possíveis categorias de crime (e.g., furto, arrombamento, assalto, homicídio, etc) e consequentemente, pode não refletir o grau de segurança requerido. Além disso, quando a rota mais segura não for distinta da menor rota, ou ainda, quando os usuários não requerem rotas mais seguras, então a menor rota é sempre recomendada, o que pode gerar congestionamentos e outros problemas inerentes.

A abordagem proposta por [Quercia et al. 2014], para recomendação baseada em informações contextuais, considera a percepção das pessoas sobre as rotas na cidade, a fim de recomendar rotas mais agradáveis. A participação ativa das pessoas, durante todas as etapas do sistema de recomendação, é um ponto crítico dessa abordagem e, por isso, manter os usuários motivados é essencial para o sucesso desse sistema. Em [De Domenico et al. 2015], a mobilidade dos agentes (e.g., veículos) é modelada como um sistema de partículas, onde informações de diferentes contextos da cidade tais como, tráfego, poluição, crimes e eventos, representam forças exercidas sobre as partículas. Nesse modelo, os agentes são partículas que devem se mover entre pares origem-destino em uma matriz, que representa uma área geográfica de interesse. A resultante das forças de atração/repulsão deve guiar a partícula pela matriz. Como as partículas que representam os indivíduos se movem por um espaço 2D livre, i.e., sem barreiras, esse modelo não considera as limitações da malha de ruas e avenidas existente nas cidades.

Este trabalho diferencia dos anteriores por propor um arcabouço capaz de identificar padrões em rotas frequentemente utilizadas por pessoas no mundo real, utilizando contextos, ao invés de inferir melhores rotas baseando-se na rota mais curta. Como [De Domenico et al. 2015], o arcabouço proposto possibilita a identificação de áreas contextuais em fontes de dados urbanos. Contudo, realizamos a análise agregada dos dados urbanos para minimizar erros e enriquecer a qualidade da informação obtida.

3. Arcabouço para Apoiar a Recomendação de Rotas Cientes de Contexto

3.1. Visão geral

A ampla disponibilidade de fontes de dados sobre diversos aspectos da cidade permite entender e tratar problemas enfrentados pelos centros urbanos e, com isso, oferecer serviços mais sofisticados que visam melhorar a vida dos usuários na cidade. Dessa forma, estudamos uma abordagem para o oferecimento de recomendação de rotas veiculares cientes de contexto.

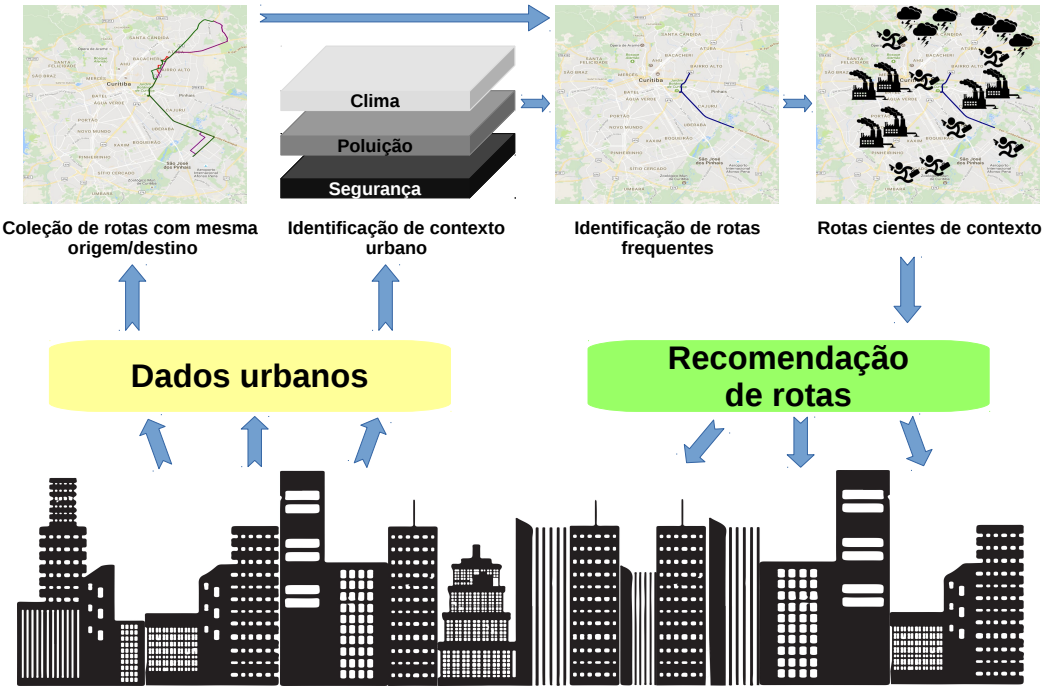


Figure 1. Visão geral do arcabouço proposto.

A Figura 1 ilustra o arcabouço proposto para apoiar serviços de recomendação de rotas cientes de contexto. Considerando o serviço que temos interesse, precisamos de um conjunto de rotas, bem como dados sobre um ou mais aspectos da cidade. Após a obtenção desses dados, que podem ocorrer de diversas formas [Silva and Loureiro 2015], eles são processados para gerar informações de interesse. Em particular, neste trabalho o arcabouço é avaliado na Seção 5 utilizando dados contribuídos por usuários e de órgãos oficiais para detectar áreas de insegurança em áreas urbanas.

A primeira informação de interesse obtida é através de um conjunto de rotas. Seja $U = \{u_1, u_2 ... u_n\}, n \geq 1$, o conjunto de usuários, $R = \{r_1, r_2, ... r_m\}, m \geq 1$, o conjunto de rotas e f_i é a frequência de uma rota $r_i \in R$, para um dado tipo de contexto (e.g., poluição do ar) definida como $0 \leq f_i \leq 1$. Cada rota $r_i \in R$ é percorrida por um usuário $u_i \in U$, que pode percorrer uma ou mais rotas. Dado R, precisamos identificar as rotas que desejamos conhecer o contexto. Essa identificação depende da recomendação que estamos interessados em oferecer. Para isso, devemos responder a algumas perguntas, tais como:

- A recomendação será feita para algum usuário específico u_i? Se sim, precisamos filtrar todas as rotas em R que podem ser utilizadas por u_i. A mesma ideia vale para um subconjunto de U. Caso contrário, consideramos todas as rotas sem aplicar um filtro específico;
- Qual a frequência f_i adequada de uma dada rota r_i que desejamos estudar o contexto? O valor da frequência depende do contexto. Considerando um contexto de poluição do ar, talvez o valor de frequência seja maior do que para o contexto de segurança, já que percorrer caminhos em áreas com elevado índice de poluição do ar pode ser mais tolerável do que caminhos que coloquem em risco a vida do usuário. Mais detalhes de como realizar esse passo são fornecidos na Seção 3.2.

Em seguida, precisamos analisar os dados sobre algum determinado aspecto da cidade para identificar informações contextuais que serão consideradas nas rotas estudadas. Note que podemos ter várias fontes de dados sobre diferentes aspectos da cidade e, assim, temos a opor-

tunidade de incorporar diversos contextos. Dessa forma, é possível criar camadas de contexto sobre vários aspectos das cidades, tais como, segurança, poluição e clima. Com isso, uma ou mais camadas podem ser consideradas na recomendação de rotas de acordo com os requisitos de interesse do usuário. Para trabalhar com diferentes contextos simultaneamente existem várias estratégias. Neste trabalho, demonstramos esse caso utilizando uma estratégia que considera a área geográfica de todos os contextos identificados para uma região da cidade. Em outras palavras, sobrepomos as áreas dos diferentes contextos no mesmo espaço geográfico. O arcabouço proposto aqui é genérico o suficiente para permitir a utilização de outras estratégias mais sofisticadas, como a utilizada em [Silva et al. 2014], que propõe um modelo para agrupar distintas fontes de dados urbanas. No entanto, uma avaliação do custo/benefício de cada abordagem está fora do escopo deste trabalho.

Nosso foco é a identificação de áreas contextuais em fontes de dados urbanos distintas, $\mathcal{F} = \{F_1, F_2, ..., F_n\}$, onde cada $F_i \in \mathcal{F}$ é um conjunto de dados sensoriados por pessoas. Esses dados possuem uma localização geográfica d e o tempo de registro do dado t. Por exemplo, uma ocorrência de crime possui o endereço de onde o crime ocorreu (d) e o momento de sua ocorrência (t). De posse do conjunto $\mathcal{F}$, o objetivo é identificar áreas contextuais que representam de forma mais adequada a área de cobertura a de dados no elemento F_i, em vez de considerar apenas o ponto específico representado por d.

Note que a cobertura mencionada é dependente do contexto a ser identificado. Por exemplo, considerando F_i como um conjunto de dados sobre criminalidade e F_j como um conjunto de dados sobre a poluição do ar, pode ser desejável considerar um valor de a menor para F_i do que para F_j. A ocorrência de um crime em d não indica apenas que o ponto d pode ser inseguro, mas sim pelo menos em alguns quarteirões. Já a medição de um nível elevado de poluição em d pode indicar que uma área maior esteja afetada, por exemplo, um bairro inteiro. A Seção 3.3 é dedicada a discutir uma estratégia para identificar áreas contextuais.

Como também ilustra a Figura 1, de posse das rotas de interesse, bem como de uma ou mais área contextual, podemos gerar rotas cientes de contexto através da combinação dessas informações. Rotas que são cientes de um determinado contexto podem ser exploradas em algoritmos de recomendação de rotas. Por exemplo, usuários podem ativamente requerer rotas que evitem áreas com alto grau de criminalidade, ou rotas que evitem vias que podem comprometer a segurança deles em condições climáticas adversas, etc. Outra possibilidade é oferecer uma recomendação automática, ou seja, sem a requisição ativa dos usuários por determinados tipos de rotas contextuais. Para isso é necessário analisar as rotas que são frequentemente utilizadas por um determinado usuário para fornecer informações que o ajude a tomar melhores decisões, como uma rota alternativa para evitar problemas respiratórios futuros.

3.2. Identificação de Rotas Frequentes

Motoristas podem percorrer rotas existentes da cidade por conta própria ou podem utilizar sistemas de recomendação de rotas para ajudá-los em suas viagens diárias. Em ambos os casos, podem existir rotas que são frequentemente utilizadas por eles, já que há uma rotina no trânsito, como observado em [Karnadi et al. 2007]. A identificação desses padrões é especialmente interessante para o arcabouço proposto, pois possibilita o estudo e a recomendação de rotas veiculares cientes de contexto

Considere um usuário específico u_i, que se desloca diariamente de sua casa para o seu trabalho e vice-versa. Seja R o conjunto de rotas, filtramos R para obter $R_i \subseteq R$ o conjunto de rotas utilizadas por u_i para percorrer esse trajeto. Seja r_i a rota mais frequente em R_i e seja $f : 0 \leq f \leq 1$, a frequência que r_i ocorre em R_i. Dependendo do contexto de interesse

do usuário e do valor de f, é possível que uma rota alternativa seja recomendada para u_i. Por exemplo, suponha que u_i tem interesse em evitar áreas com elevados índices de poluição do ar, que r_i tem frequência $f \geq 0,90$ (limite hipoteticamente alto) e que r_i passa por várias áreas poluídas (identificadas, por exemplo, com a abordagem discutida na Seção 3.3).

Com isso, o sistema de recomendação tem a oportunidade de fornecer a u_i uma nova rota $r_j \in R_i$ (ou não) para evitar áreas poluídas, mesmo que r_j seja mais longa e aumente o tempo do percurso. Caso u_i não tenha interesse em evitar áreas poluídas, ainda assim o sistema de recomendação poderia alertá-lo sobre o risco dele contrair doenças graves. No entanto, se $f \leq 0,30$ (considerado hipoteticamente um limite baixo) e r_i for um trajeto significativamente menor que r_j, então u_i provavelmente não irá optar por r_j. Similarmente, podemos considerar as rotas de todos ou um subconjunto dos usuários U. Se considerarmos o conjunto de rotas R utilizadas por todos usuários, é possível identificar um subconjunto de rotas $R' : R' \subseteq R$, onde $\forall r' \in R'$, r' é uma rota frequente a todos usuários. Dessa forma, se considerarmos as áreas contextuais que r' permeia, é possível fornecer recomendações como as que discutimos anteriormente.

Para identificar as rotas frequentes, representamos cada rota por um grafo direcionado conexo $G(V, E)$, em que um nó $v_i \in V$ é um ponto específico de uma rota (por exemplo, uma esquina) e uma aresta direcionada $e_{i,j} \in E$ representa uma via específica entre os pontos v_i e v_j. O grafo G possui um vértice que representa a origem e um vértice que representa o destino da rota, que são os únicos vértices de grau 1 de G. Se R contém n rotas, então denotamos por $\mathcal{G} = \{G_1, G_2, ..., G_n\}$, a coleção de n grafos G que representam as n rotas $r \in R$. A coleção $\mathcal{G}$ é utilizada para identificar a coleção de subgrafos maximais $\mathcal{Q} = \{Q_1, Q_2, ..., Q_m\}$, onde cada $Q_i \in \mathcal{Q}, 1 \leq i \leq m$, é um subgrafo maximal que ocorre com frequência f em $\mathcal{G}$, e Q_i representa um trecho (i.e., uma parte da rota ou a rota em si) que é comum a f rotas contidas em R.

Note que encontrar cada subgrafo maximal Q_i por meio de testes de isomorfismo é conhecido por ser um problema $\mathcal{NP}$–completo [Cook 1971]. Então, para contornar esse problema, podemos utilizar, por exemplo, o algoritmo chamado *graph-based Substructure pattern mining* (gSpan), que minimiza esse problema na sua estratégia adotada [Yan and Han 2002]. O gSpan espera como parâmetro uma lista de grafos $\mathcal{G}$ e um valor de frequência mínimo f que um determinado subgrafo deve apresentar. De posse desses parâmetros, o algoritmo retorna uma lista de subgrafos $\mathcal{Q}$.

3.3. Identificação de Contexto Urbano

Nesta seção, discutimos mais detalhes de uma etapa chave do arcabouço discutido neste estudo: a identificação de contexto. Dados urbanos, quando analisados individualmente, podem não refletir precisamente o estado atual de um contexto específico. Isto porque o dado pode possuir diversos problemas, como estar desatualizado, incorreto ou não possuir a granularidade necessária para representar corretamente um determinado contexto.

Para ilustrar um tipo de problema relacionado à localização do evento, discutimos o que é observado para dados de criminalidade. Dados oficiais sobre criminalidade podem não possuir a localização precisa do crime, por meio de geolocalização ou logradouro completo. Ao invés disso, a localização é correspondente a uma área na cidade, por exemplo, nome da rua e bairro, como é o caso dos dados fornecidos pela prefeitura de Curitiba[3]. Fontes alternativas para esse tipo de dado, como as do site `ondefuiroubado.com.br`, que permitem a usuários compartilharem a ocorrência de um crime, normalmente possuem uma localização baseada em

[3]`http://www.curitiba.pr.gov.br/dadosabertos`

geolocalização. No entanto, os usuários podem informar erroneamente a localização em que eles presenciaram algum crime, já que eles podem, por exemplo, não recordar o local exato da ocorrência do crime no momento de compartilhar o dado. Diante disso, é importante realizar a análise agregada dos dados urbanos com o intuito de minimizar erros e enriquecer a qualidade da informação obtida.

Para este fim, uma possível estratégia para a identificação de contexto urbano é a clusterização. Clusterização é a organização não-supervisionada de uma coleção de dados em grupos de acordo com alguma medida de similaridade [Jain et al. 1999]. Assim, quando é aplicado o procedimento de clusterização a um conjunto de dados F_i, são identificados subconjuntos disjuntos de dados $C_i^1, C_i^2, ..., C_i^n$ chamados *clusters*, onde dados em um mesmo *cluster* são mais semelhantes entre si do que com dados pertencentes a *clusters* diferentes [Jain et al. 1999]. A medida de similaridade adotada é a distância geográfica provida pela fórmula de haversine [Sinnott 1984].

Dentre as técnicas existentes de clusterização, o algoritmo *Density Based Spatial Clustering of Application with Noise* (DBSCAN) [Ester et al. 1996] é particularmente interessante, porque é capaz de gerar *clusters* com diferentes formatos e tamanhos e por ter bom resultado de clusterização mesmo na presença de ruídos. Por essa razão, esta é a estratégia adotada neste estudo. DBSCAN identifica clusters baseando-se na densidade da vizinhança dos pontos, onde pontos que estão em regiões de baixa densidade (de acordo com o que é especificado para o algoritmo) são chamados *outliers*. Cada *cluster* C é formado por um conjunto de pontos, onde o número mínimo de pontos é η, os quais estão em uma vizinhança de raio ε. Os parâmetros η e ε devem ser fornecidos como entrada do algoritmo DBSCAN. Seja $p_i \subseteq C$ um ponto, que pode ser um ponto *central* se p_i tiver pelo menos $\eta - 1$ pontos que estejam à distância máxima ε de p_i. Caso contrário, p_i é um ponto *periférico* de um ponto central $p_j \subseteq C$. *Outliers* são os pontos que não são centrais nem periféricos e são descartados pelo algoritmo. Essa característica é interessante, pois pode auxiliar no descarte de ruídos e dados imprecisos.

Note que a necessidade de ajustar os parâmetros η e ε do DBSCAN é fundamental para lidar com múltiplos contextos. Isto porque dois contextos distintos F_i e F_j, provavelmente possuem propriedades diferentes. Por exemplo, suponha que F_i seja o contexto de crime e F_j seja o contexto de clima. Assim, um *cluster* C_i obtido a partir de F_i representa uma área de insegurança na cidade, enquanto que C_j obtido a partir de F_j representa as condições climáticas de uma área da cidade. Provavelmente, o raio ε_j do *cluster* C_j será definido como sendo maior do que o raio ε_i do *cluster* C_i, já que uma área de insegurança é tipicamente definida na granularidade de quarteirões ou bairro, enquanto que uma condição climática pode ser referente a uma área que abrange uma cidade inteira.

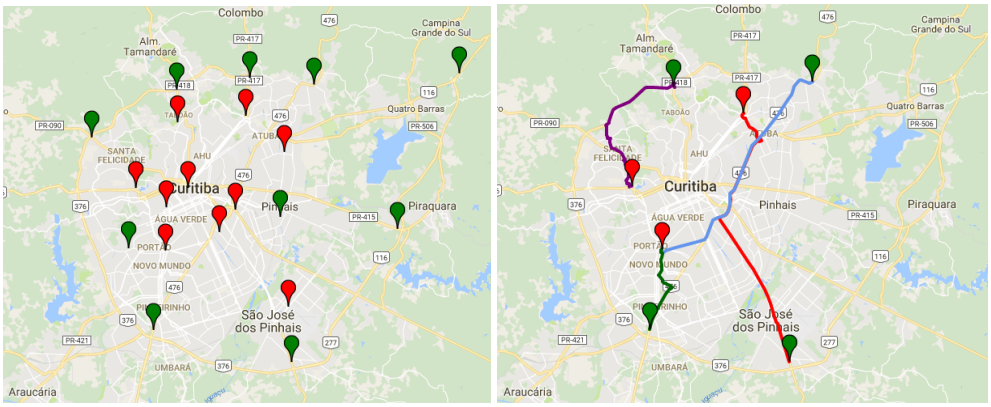
Além disso, a "validade" do *cluster* também pode diferir de acordo com o contexto que ele representa. De fato, dados sobre clima são significativos por um curto período de tempo (e.g., horas ou dia) e, por isso, devem ser atualizados com maior frequência. Enquanto dados sobre crime são relevantes por períodos maiores de tempo (e.g., semanas ou meses), já que uma área de insegurança não tende a se tornar segura em poucos dias. Assim, cada *cluster* C deve ter definido uma janela temporal $\mathcal{T} = [t_{min}, t_{max}]$, onde qualquer ponto $p \in C$ representa uma localização d e um instante de tempo t, tal que $t_{min} \leq t \leq t_{max}$.

4. Bases de Dados

Esta seção apresenta as bases de dados analisadas neste trabalho. A Seção 4.1 apresenta as rotas coletadas que foram sugeridas pelo Google Maps. A Seção 4.2 apresenta dados extra-oficiais (Subseção 4.2.1) e oficiais sobre crimes ocorridos em Curitiba (Subseção 4.2.2).

4.1. Rotas Recomendadas pelo Google Maps

Um sistema de recomendação de rotas é uma poderosa ferramenta utilizada pelas pessoas para ajudá-las durante a navegação pela cidade. As pessoas devem informar o ponto de origem (geralmente obtido de forma automática), o ponto de destino e o modo de navegação (e.g., automóvel, transporte público ou a pé) e, em seguida, o sistema recomenda uma rota baseada nas informações fornecidas. Google Maps e Waze são dois sistemas de recomendação de rotas bem conhecidos, que são comumente utilizados pelas pessoas devido à capacidade deles em recomendar "melhores" rotas. Tais sistemas, consideram informações sobre as condições históricas de trânsito e as informações em tempo real para estimar a melhor rota em um determinado momento.



(a) Pontos de origem (verde) e destino (vermelho).

(b) Exemplos de rotas recomendadas.

Figure 2. Rotas recomendadas pelo Google Maps em Curitiba.

Neste trabalho, consideramos a cidade de Curitiba como cenário para identificação de padrões na recomendação de rotas do Google Maps. Para isso, foram determinados dois conjuntos distintos de endereços para representar os pontos de origem e destino, onde cada conjunto possui dez endereços distintos. A Figura 2(a) mostra o mapa da cidade de Curitiba com os pontos de origem e destino representados em verde e vermelho, respectivamente. Com ambos conjuntos definidos, é construída a matriz de origem/destino, denotada por M_{10x10}, que é utilizada para solicitar ao Google Maps Directions API[4] recomendações de rotas a partir de cada origem a todos os destinos definidos. O modo de navegação adotado é o automóvel. A cada iteração, são obtidas 100 rotas recomendadas, com informações do percurso e as condições de trânsito ao longo dele.

Ao todo, foram realizadas 420 iterações, durante 60 dias, com intervalo de 7 horas entre as iterações. Assim, o conjunto de rotas recomendadas em Curitiba tem 42.000 rotas, em diferentes períodos do dia (i.e., manhã, tarde e noite), com instruções detalhadas em cada trecho do percurso (e.g., distância à percorrer e duração prevista), tempo estimado de chegada ao destino e distância total. A Figura 2(b) mostra quatro exemplos de rotas sugeridas pelo Google Maps, rotas representadas pelas cores azul, vermelho, verde e roxo.

4.2. Dados Sobre Ocorrências de Crimes

Considerando o contexto de segurança pública, foram coletados dois conjuntos de dados de ocorrências de crime em Curitiba: um reportado por usuários do serviço "Onde Fui Roubado"

[4]https://goo.gl/kiXllw

(dados alternativos provenientes da colaboração coletiva), e o outro disponibilizado pela Guarda Municipal (dados oficiais). A seguir, apresentamos uma breve descrição dessas bases de dados.

4.2.1. Dados Contribuídos por Usuários

O sítio "Onde Fui Roubado"[5] é um sistema online onde os usuários contribuem voluntariamente com dados sobre denúncias de crimes sofridos ou presenciados por eles. Com isso, o sistema é capaz de mapear as regiões de insegurança das cidades, o que ajuda os usuários a se prevenirem. Encontra-se disponível no sítio registros de crimes que ocorreram nos últimos quatro meses, com relação a uma data atual, em cidades brasileiras. Os principais atributos acerca de cada ocorrência são: endereço da ocorrência (logradouro, bairro, cidade, estado, país e CEP), geolocalização (latitude e longitude), descrição do crime, data e horário de registro no sistema, natureza do crime, prejuízo estimado e se foi realizado o boletim de ocorrência junto ao órgão competente. Ao longo de quatro meses, de julho à outubro de 2016, foram registradas 331 ocorrências de crime em Curitiba: roubo (36,8%), furto (26,5%), arrombamento domiciliar (7,8%), arrombamento veicular (7,5%), assalto a grupo (7,25%), roubo de veículo (6,35%), tentativa de assalto (6,0%), arrombamento de estabelecimento comercial (0,9%), sequestro relâmpago (0,3%), "saidinha" bancária (0,3%) e arrastão (0,3%). Dessas ocorrências, cerca de 39% não foram registradas oficialmente, o que pode indicar que uma parte significativa dos usuários preferem manifestar sua insatisfação com a insegurança por meio de um sistema extra-oficial. Desta forma, é clara a importância dos dados provenientes da colaboração coletiva para enriquecer o conhecimento acerca da insegurança nas cidades.

4.2.2. Dados Oficiais

A base de dados oficial[6] contém registros sobre ocorrências de crime atendidas pela Guarda Municipal de Curitiba desde 2009 e é atualizada mensalmente, onde a última atualização da base de dados utilizada neste trabalho ocorreu em 1° de novembro de 2016. Ao todo, são 182.267 registros de ocorrências criminais na região metropolitana de Curitiba, sendo 18.235 em 2016. Os principais atributos acerca de cada ocorrência são: o ano de atendimento, bairro, logradouro, se houve flagrante, a natureza da ocorrência (e.g., roubo), a sub-categoria da natureza (e.g., veículo), data e horário de registro da ocorrência no sistema e descrição da situação no momento de atendimento. Como a base de dados oficial possui registros de crimes de diversas naturezas e durante um longo período de tempo, utilizamos apenas o subconjunto de dados que ocorreram no mesmo período que os dados extra-oficiais e cuja natureza do crime seja uma categoria existente nos dados extra-oficiais, totalizando 1.313 ocorrências de crime.

5. Análise Experimental

Como prova de conceito do arcabouço proposto, nesta seção consideramos rotas recomendadas pelo Google Maps e dados de criminalidade para Curitiba, com o intuito de analisar as rotas recomendadas com relação à segurança.

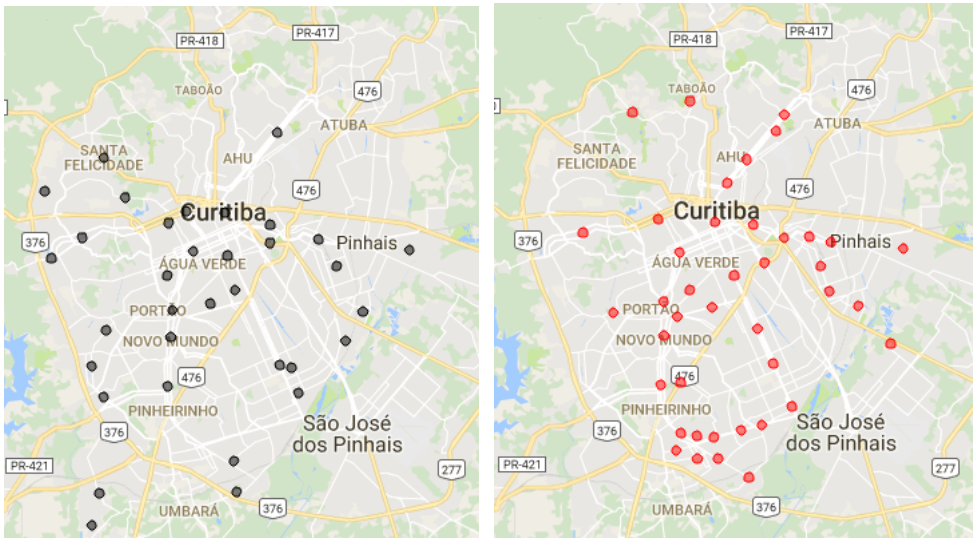
5.1. Identificação de Áreas de Insegurança

Para identificar as áreas de insegurança de Curitiba, aplicamos a metodologia proposta em nosso arcabouço (Seção 3.3). Consideramos que cada ocorrência de crime atinge diretamente uma área

[5]Disponível em: `https://goo.gl/HPOAHB`
[6]Disponível em: `https://goo.gl/Hf0aP2`

equivalente a um quarteirão (i.e., 100 m^2). Representamos cada ocorrência de crime que ocorreu em um local d no instante t pela variável o. Isto significa que para a mesma localidade d podemos ter diversas ocorrências o. Com isso, um *cluster* $C = \{o_1, o_2, ..., o_n\}$, que contém n ocorrências de crimes, possui uma área de cobertura a representando uma área de insegurança em Curitiba. O tamanho de a pode variar de acordo com a densidade da vizinhança dos pontos, podendo ser equivalente a alguns quarteirões ou até um ou mais bairros. Por simplicidade, consideramos que a é uma área circular, cujo centro de a é o centroide de C e o raio de a é o raio ε utilizado no algoritmo para a identificação de C.

Dadas essas informações, definimos neste experimento que $\varepsilon = 250$ m e $\eta = 5$, pois acreditamos que esses parâmetros resultem em *clusters* que representem razoavelmente bem áreas de insegurança na cidade. Além disso, consideramos a janela temporal de quatro meses ($\mathcal{T} = [$*Junho/2016, Outubro/2016*$]$), que é o mesmo período em que os dados sobre crimes são válidos na plataforma `ondefuiroubado.com.br`. Em seguida, filtramos os dados de crimes de acordo com o seu grau de severidade. A divisão foi feita em duas classes: (1) crimes moderados, por exemplo furto, e (2) crimes graves, por exemplo, sequestro.



(a) Crimes moderados (b) Crimes graves

Figure 3. Áreas ameaçadas por crimes moderados e crimes graves em Curitiba.

A Figura 3 mostra as áreas de insegurança em Curitiba, onde os *clusters* de crimes moderados (furto, crime tentado, arrombamento, arrastão e "saidinha") são as áreas em preto mostradas na Figura 3a e os de crimes graves (roubo, sequestro e assalto) são as áreas em vermelho na Figura 3b. Ao todo, foram identificadas 75 áreas de insegurança, sendo 34 de crimes moderados e 41 de crimes graves, havendo sobreposição de áreas em alguns casos.

5.2. Segurança de Rotas Frequentes

Nesta seção, identificamos trechos (i.e., subgrafos maximais) das rotas que são regularmente recomendadas pelo Google Maps para verificar se elas tendem a passar por áreas inseguras. Para isso, consideramos as rotas coletadas R^i com mesmo par origem/destino $i : 1 \leq i \leq 10$, como descrito na Seção 4.1. Em seguida, utilizamos a abordagem de identificação de rotas frequentes descrita na Seção 3.2. Definimos a frequência $f = 0,85$, o que significa que Q^i está contida em pelo menos 85% das rotas R^i. Assim, se Q^i permeia áreas de insegurança em Curitiba, então os usuários que solicitarem uma rota com origem/destino i, têm grandes chances de passarem por áreas inseguras.

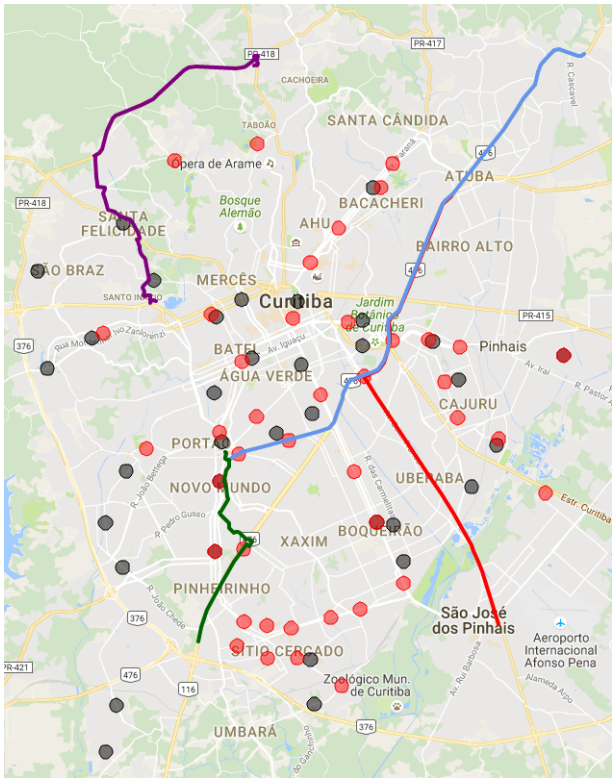


Figure 4. Trechos frequentes contidos em rotas recomendadas pelo Google Maps, que permeiam áreas de insegurança em Curitiba.

A Figura 4 traça no mapa de Curitiba alguns exemplos de trechos frequentes ($f = 0,85$), que estão contidos nas rotas exibidas na Figura 2(b) e disponíveis em R^i, juntamente com as áreas inseguras encontradas de acordo com a nossa abordagem. Todos esses exemplos ilustram rotas que passam por pelo menos uma área insegura e, em alguns casos, passam por quatro. Esse resultado é um indício de que aspectos de segurança não são considerados na recomendação de uma determinada rota. Ressaltamos que as rotas frequentemente sugeridas nem sempre passam por áreas inseguras e, em alguns casos, o padrão identificado é um trecho pequeno, o que é insuficiente para determinar se as rotas que contêm o padrão passam por áreas de insegurança.

Nessa direção, uma possibilidade é oferecer ao usuário a oportunidade de definir o grau de segurança da rota que será recomendada a ele. Como descrevemos anteriormente, áreas de insegurança são divididas em áreas com crimes moderados ou graves. Dessa forma, o usuário pode estar disposto a assumir o risco de seguir por uma rota que permeia áreas de insegurança com crimes moderados, como o trecho de cor roxa da Figura 4. Alternativamente, o usuário pode optar por uma rota livre de áreas inseguras, mesmo que isso aumente a distância do percurso e o tempo de viagem. Nesse caso, o sistema de recomendação ciente de contexto deve sugerir rotas que evitem áreas inseguras. A Figura 5 mostra possíveis opções de rotas mais seguras, cuja origem/destino são as mesmas exemplificadas na Figura 2(b).

Considerando as métricas (*i*) distância total e (*ii*) tempo de viagem, realizamos uma comparação entre as rotas que contêm os trechos frequentes ilustrados na Figura 4, denominadas r_1 (rota roxa), r_2 (rota verde), r_3 (rota azul), r_4 (rota vermelha), com as rotas alternativas ilustradas na Figura 5, denominadas r'_1, r'_2, r'_3, r'_4. A Tabela 1 mostra os resultados das métricas (*i*) e (*ii*) para as rotas estudadas. Os valores de ambas as métricas são calculados por meio da API do Google Maps, onde o tempo de viagem é obtido considerando o mesmo horário de partida

para todas as rotas. Como podemos observar, apesar das rotas mais seguras serem maiores que as rotas frequentes, elas não adicionam um atraso muito grande nas rotas, em média 9,75 min, e nenhum atraso entre as rota r_1 e r'_1. Isso ilustra que os mecanismos atuais de recomendação de rotas poderiam ser adaptados para atender a essa nova funcionalidade.

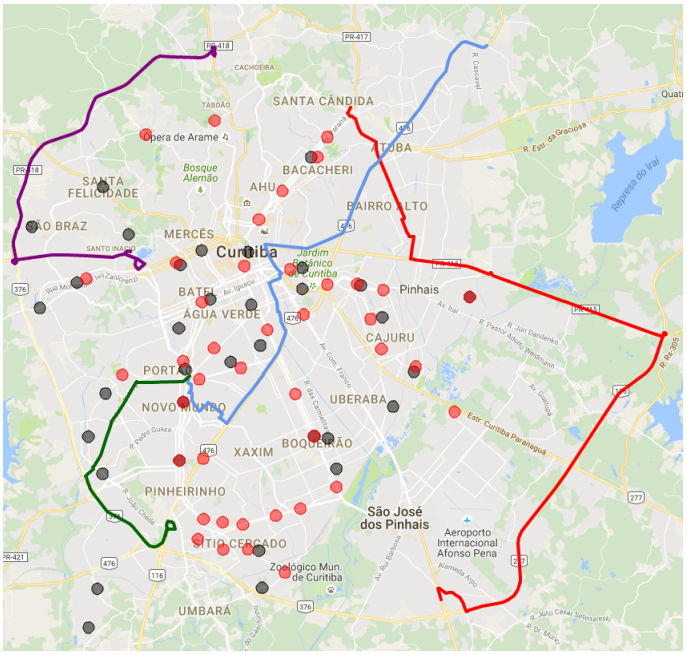


Figure 5. Opções de rotas mais seguras em Curitiba.

Table 1. Rotas frequentes *versus* rotas mais seguras

	r_1	r'_1	r_2	r'_2	r_3	r'_3	r_4	r'_4
Distância total (km)	15,2	23,1	8,5	14,1	21,8	27,6	25,1	37,5
Tempo de Viagem (min)	22	22	15	22	36	53	35	50

6. Considerações Finais e Trabalhos Futuros

O presente trabalho explora a utilização de fontes de dados distintas sobre diferentes aspectos das cidades, juntamente com rotas que são frequentemente utilizadas pelas pessoas em suas viagens diárias, para fornecer mecanismos importantes para a construção de uma rede veicular sensível ao contexto. Para atender esse objetivo, definimos um arcabouço para auxiliar no processo de recomendação de rotas veiculares. Considerando o contexto de segurança, o arcabouço proposto demonstra que pode possibilitar o aprimoramento de sistemas de recomendação de rotas, identificando rotas que são frequentemente recomendadas e que passam por áreas de insegurança. No entanto, é importante observar que o mesmo princípio poderia ser aplicado a qualquer outro tipo de contexto individualmente ou tratado de forma conjunta. Essa é uma das vantagens desta proposta: a flexibilidade em tratar diferentes contextos de forma homogênea.

Como trabalho futuro, pretende-se evoluir este arcabouço para incorporar outras fontes de dados heterogêneas, tais como, redes sociais, redes oportunistas e redes veiculares, para possibilitar a identificação de múltiplos contextos, sem depender de fontes de dados específicos.

Agradecimentos

Os autores gostariam de agradecer o apoio financeiro concedido da CAPES para o bolsista de doutorado Frances Albert Santos. Ainda, os autores também agradecem a FAPEMIG, Fundação

Araucária e CNPq por financiarem partes dos seus projetos de pesquisas. Por fim, Leandro Villas agradece o apoio financeiro da FAPESP por meio do processo no 2015/07538-1, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

References

Bader, R., Neufeld, E., Woerndl, W., and Prinz, V. (2011). Context-aware poi recommendations in an automotive scenario using multi-criteria decision making methods. In *Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation*, CaRR '11, pages 23–30, New York, NY, USA. ACM.

Calavia, L., Baladrón, C., Aguiar, J. M., Carro, B., and Sánchez-Esguevillas, A. (2012). A semantic autonomous video surveillance system for dense camera networks in smart cities. *Sensors*, 12(8):10407.

Cook, S. A. (1971). The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM.

Dai, J., Yang, B., Guo, C., and Ding, Z. (2015). Personalized route recommendation using big trajectory data. *2015 IEEE 31st International Conference on Data Engineering (ICDE)*, 00(undefined):543–554.

De Domenico, M., Lima, A., González, M. C., and Arenas, A. (2015). Personalized routing for multitudes in smart cities. *EPJ Data Science*, 4(1):1.

Demiryurek, U., Banaei-Kashani, F., and Shahabi, C. (2009). Transdec: A data-driven framework for decision-making in transportation systems. 50th Annual Transportation Research Forum, Portland, Oregon, March 16-18, 2009 207726, Transportation Research Forum.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.

Jain, A., Mangharam, R., and Behl, M. (2016). Data predictive control for peak power reduction. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*, BuildSys '16, pages 109–118, New York, NY, USA. ACM.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323.

Karnadi, F. K., Mo, Z. H., and c. Lan, K. (2007). Rapid generation of realistic mobility models for vanet. In *2007 IEEE Wireless Communications and Networking Conference*, pages 2506–2511.

Lee, U., Zhou, B., Gerla, M., Magistretti, E., Bellavista, P., and Corradi, A. (2006). Mobeyes: Smart mobs for urban monitoring with a vehicular sensor network. *Wireless Commun.*, 13(5):52–57.

Melnikov, V., Krzhizhanovskaya, V., Boukhanovsky, A., and Sloot, P. (2015). Data-driven modeling of transportation systems and traffic data analysis during a major power outage in the netherlands. *Procedia Computer Science*, 66:336–345.

Quercia, D., Schifanella, R., and Aiello, L. M. (2014). The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. *CoRR*, abs/1407.1031.

Rio, G. (2015). *Mulher morre após casal entrar por engano em comunidade em Niterói*. G1. *http://g1.globo.com/rio-de-janeiro/noticia/2015/10/mulher-morre-apos-entrar-por-engano-em-comunidade-em-niteroi-rj.html*.

Shah, S., Bao, F., Lu, C.-T., and Chen, I.-R. (2011). Crowdsafe: Crowd sourcing of crime incidents and safe routing on mobile devices. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, pages 521–524, New York, NY, USA. ACM.

Silva, T. H., de Melo, P. O. V., Almeida, J. M., Vianna, A. C., Salles, J., and Loureiro, A. A. (2014). Definição, modelagem e aplicações de camadas de sensoriamento participativo. In *Brazilian Symposium on Computer Networks and Distributed Systems (SBRC'14), Florianópolis, Brazil*.

Silva, T. H. and Loureiro, A. A. (2015). Computação urbana: Técnicas para o estudo de sociedades com redes de sensoriamento participativo. In *Anais da XXXIV Jornada de Atualização em Informática*, volume 8329, pages 68–122. Sociedade Brasileira de Computação – SBC.

Sinnott, R. W. (1984). Virtues of the Haversine. *Sky and Telescope*, 68(2):159+.

ter Mors, A. W., Witteveen, C., Zutt, J., and Kuipers, F. A. (2010). *Context-Aware Route Planning*, pages 138–149. Springer Berlin Heidelberg, Berlin, Heidelberg.

Vaghefi, A., Jafari, M., Bisse, E., Lu, Y., and Brouwer, J. (2014). Modeling and forecasting of cooling and electricity load demand. *Applied Energy*, 136(C):186–196.

Vo, H. T. (2015). Enabling smart transit with real-time trip planning. In *Proceedings of the ACM First International Workshop on Understanding the City with Urban Informatics*, UCUI '15, pages 13–18, New York, NY, USA. ACM.

Wang, H., Li, G., Hu, H., Chen, S., Shen, B., Wu, H., Li, W.-S., and Tan, K.-L. (2014). R3: A real-time route recommendation system. *Proc. VLDB Endow.*, 7(13):1549–1552.

Yan, X. and Han, J. (2002). gspan: Graph-based substructure pattern mining. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 721–724. IEEE.

Agregação Dinâmica de Enlaces em Ambientes de Redes Definidas por Software

Ronaldo Resende Rocha Junior,
Marcos A. M. Vieira, Antonio A. F. Loureiro

[1]Departamento de Ciência da Computação
Universidade Federal de Minas Gerais – Belo Horizonte, MG – Brasil

{ronaldorrjr,mmvieira,loureiro}@dcc.ufmg.br

Abstract. *The link aggregation technique is a solution to the link saturation problem. This technique combines several physical links to create a virtual link with the sum of the respective bandwidths. Since the use of Software Defined Networks (SDN) in corporate environments increases every day, the purpose of this work is to allow the SDN controller to configure the aggregation of links. Therefore, We have defined and implemented an architecture to dynamically aggregate links such that it is configurable, scalable, and autonomous. We evaluate three link aggregation algorithms: hash, traffic analysis, and round-robin. In our tests, the proposed system improves the distribution of the bandwidth, allows more aggregated interfaces, and increases the throughput between two endpoints.*

Resumo. *A técnica de agregação de enlaces é uma solução para o problema de saturação do enlace. Essa técnica combina várias interfaces físicas para criar um enlace virtual com a soma das respectivas bandas. Visto que a utilização das Redes Definidas por Software (SDN) em ambientes coorporativos aumenta a cada dia, a proposta deste trabalho é permitir o controlador SDN configurar a agregação de enlaces. Para tanto, definimos e implementamos uma arquitetura que permite agregar enlaces de forma dinâmica, configurável, escalável e autônoma. Foram avaliados três algoritmos de agregação de enlaces: hash, análise de tráfego e round-robin. Em nossos testes, o sistema proposto melhora a distribuição da banda, permite maior quantidade de interfaces agregadas e aumenta a vazão dos tráfegos entre duas pontas.*

1. Introdução

As redes de computadores modernas são sistemas complexos e, por isso, a sua administração é um grande desafio tanto para profissionais quanto para pesquisadores [Ranum et al. 1998]. Os equipamentos de redes atuais, tais como roteadores, *firewalls* e *switches*, são considerados verdadeiras caixas pretas [Moore and Nettles 2001]. Cada empresa é responsável por criar seus sistemas operacionais e, mesmo que os algoritmos de roteamento sejam os mesmos, a forma como eles são implementados varia bastante, podendo, inclusive, levar a problemas de interoperabilidade. Esse foi o caso do padrão IEEE 802.11, que devido a problemas práticos de integração dos dispositivos de diferentes fabricantes, levou à criação da aliança Wi-Fi[1].

[1]http://www.wi-fi.org/

O gargalo de tráfego é um dos principais problemas existentes em ambientes de redes clássicos [Carter and Crovella 1996]. Isso acontece, por exemplo, quando vários usuários tentam utilizar um único servidor em uma rede, saturando o enlace que conecta o *switch* àquele servidor. A Figura 1 ilustra esse problema, onde quatro servidores (conectados em um único *switch* utilizando um enlace de 1 Gbps cada) tentam acessar recursos de um servidor remoto (que está conectado ao mesmo *switch*, utilizando apenas uma conexão de 1 Gbps) ao mesmo tempo.

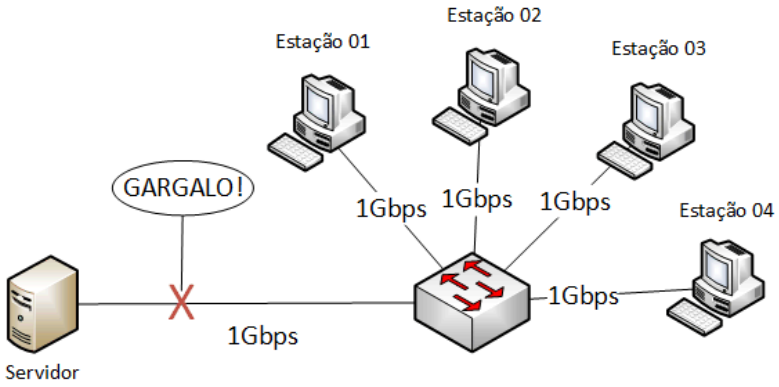


Figura 1. Exemplo de gargalo

Considerando que o fluxo de dados de cada usuário tenha o mesmo tamanho, podemos estimar que durante a concorrência do tráfego, caso todos os usuários tentem acessar o servidor em questão ao mesmo tempo, a banda média disponível para cada um será de apenas 250 Mbps. Vale salientar que essa é uma estimativa média realizada apenas para ilustrar o problema do gargalo, uma vez que em redes com fluxos reais TCP/UDP, essa taxa média dependerá do tamanho do fluxo sendo transmitido pelo usuário ao servidor [Judd 2015]. Isso demonstra que muitas vezes avaliações incorretas são feitas a respeito da lentidão em ambientes de rede, levando à troca ou *upgrade* desnecessários de equipamentos.

Uma rede com alta disponibilidade, tolerante a falhas e com alto potencial de transmissão dos seus enlaces é ideal em ambientes coorporativos [Floyd and Jacobson 1995]. Através da implantação da agregação de enlaces, podemos conseguir essas características, já que essa técnica permite transformar diversos enlaces físicos em apenas um enlace virtual. Nos *switches* clássicos, é utilizado o protocolo aberto LACP (*Link Aggregation Control Protocol*) ou os diversos protocolos proprietários dos fabricantes (Cisco: *Port Aggregation Protocol*, Huawei: *Eth-Trunk*, Juniper: *Aggregated Ethernet*, entre outros).

Este trabalho tem como objetivo propor a implementação da técnica de agregação de enlaces em controladores SDN e, assim, não será utilizado o protocolo LACP, mas a estrutura de dados do protocolo OpenFlow. Através dessa premissa, os seguintes resultados foram obtidos:

- **Melhor distribuição da banda**: utilizando diferentes estratégias para a tomada de decisão ao criar os fluxos de transmissão de pacotes, as interfaces da agregação podem ter um balanceamento de tráfego real.

- **Agregação autônoma**: a agregação pode ser criada, modificada ou removida automaticamente pelo controlador sem a necessidade de intervenção do administrador de redes.
- **Protocolo aberto**: utilizando a estrutura de dados e tabelas presentes no OpenFlow, o fluxo de transmissão de pacotes entre os enlaces de uma agregação é gerenciado pelo controlador SDN. Isso faz com que o uso de protocolos de agregação proprietários não seja mais necessário.
- **Aumento da quantidade de interfaces agregadas**: uma das limitações do protocolo LACP é a utilização máxima de oito interfaces para a agregação. Ao utilizar o LACP na plataforma *Open vSwitch*, esse valor cai para quatro portas (uma limitação referenciada na documentação da plataforma). Ao migrar a gerência da agregação desses enlaces para o controlador SDN, essa limitação deixa de existir.
- **Velocidade de transmissão entre links**: outra limitação do protocolo LACP é a necessidade de utilizar interfaces com a mesma configuração de velocidade na agregação, ou seja, interfaces com velocidade 100 Mbps não podem ser agregradas com interfaces com velocidade de 1 Gbps. Essa limitação existe por motivos de implementações do protocolo, na forma como ele realiza e mantêm a agregação configurada. Tal limitação não existiria em ambientes SDN.

O restante deste artigo está organizado como segue. Primeiramente, a Seção 2 discute os trabalhos relacionados. A Seção 3 revela a arquitetura proposta e os algoritmos de agregação de enlaces. A Seção 4 descreve os experimentos e os resultados obtidos. Por fim, a Seção 5 apresenta a conclusão e os trabalhos futuros.

2. Trabalhos Relacionados

2.1. LACP *(Link Aggregation Control Protocol)*

O protocolo aberto LACP (*Link Aggregation Control Protocol*) [Seaman 1999] permite a implantação da agregação de enlaces nos *switches* clássicos. Esse protocolo foi publicado em 1997 e sua última versão é de 2014, com o padrão 802.1AX-2014. O LACP é amplamente utilizado em ambientes clássicos de redes, mas a sua aplicação em redes SDN ainda não é muito pesquisada.

Pra criar o fluxo, o LACP analisa vários parâmetros de origem e de destino como MAC, IP, porta e etiqueta MPLS. A partir do hash desses parâmetros, o protocolo escolhe a porta para o fluxo, ou seja, todos os pacotes deste fluxo passam somente nessa interface. Quando surge um novo fluxo, um novo hash é calculado e, provavelmente, outra interface será utilizada para ele. Assim, o LACP usa todas as interfaces da agregação, espalhando os fluxos entre elas.

Para criar uma agregação de enlaces utilizando o protocolo LACP, o administrador de redes deve realizar a configuração de forma manual. Ao ser habilitado, o protocolo cria uma interface virtual e a quantidade de enlaces físicos não pode ser maior do que oito e todas as interfaces do *switch* devem estar configuradas com a mesma velocidade.

O protocolo LACP pode ser empregado em redes SDN através da plataforma *Open vSwitch* [Pfaff et al. 2009]. Essa plataforma representa a implementação de um *switch* virtual capaz de suportar vários protocolos de camada 2 e interfaces de gerenciamento como *sFlow*, espelhamento de porta e *NetFlow*. Por ser basicamente uma máquina virtual,

essa plataforma geralmente é instalada em estações ou servidores, impossibilitando sua utilização em equipamentos SDN físicos.

2.2. Outras abordagens

A plataforma *Open vSwitch* foi criada para ser um *switch* virtual capaz de executar em qualquer ambiente [Pfaff et al. 2009]. Pelo fato de ser bastante completa e robusta, logo foi incorporada nas redes SDN. Entretanto, ao considerarmos a técnica de agregação de enlaces, essa plataforma não é eficaz. A quantidade máxima de agregação de portas físicas é bastante limitada, não podendo ultrapassar quatro unidades (uma limitação da própria plataforma). Além disso, essa configuração precisa ser manual. Portanto, a solução não é escalável ou autônoma, sendo ineficiente em ambientes como data centers, que possuem centenas de equipamentos.

Em [Steinbacher and Bredel 2015], os autores propõem a implementação do protocolo LACP no Floodlight, um controlador aberto escrito em linguagem Java. Esse trabalho apenas discute a implementação teórica do protocolo, sendo que a configuração do código do protocolo em si não é elaborada.

Já os autores de [Bredel et al. 2014], [Ahn et al. 2009], [Al-Fares et al. 2010] e [Alizadeh et al. 2010] consideram o caminho de um tráfego TCP entre múltiplos enlaces, sem considerar uma possível agregação local deles. Porém, em todos eles são considerados caminhos através de múltiplos saltos. A diferença deste trabalho para os demais é a abordagem na camada dois, ou seja, a comunicação entre dois *switches* com enlaces agregados entre eles.

Por fim, em [Vencioneck et al. 2014], os autores apresentam uma nova plataforma para controlar o encaminhamento de pacotes substituindo o *Open vSwitch*. Apesar de ser inovador e possuir resultadores promissores, esse estudo é complementar ao nosso, pois não considera a agregação de enlaces entre dois dispositivos.

3. Arquitetura

Esta seção apresenta a arquitetura proposta neste trabalho, implementação, identificação dos enlaces de agregação e os algoritmos de agregação de enlance.

3.1. Descrição

A Figura 2 retrata a arquitetura proposta:

- **Aplicação (01)**: Nesta camada, foram criadas as regras de agregação de enlace. Exemplo: as políticas de encaminhamento de pacotes de acordo com a utilização de cada interface. No total foram criadas três aplicações, que são representadas por três arquivos na linguagem *Python* que são definidas nos parâmetros de entrada quando o controlador é iniciado.
- **Controlador (02)**: responsável por gerenciar a rede local. Executa as decisões da aplicação sobre as políticas de roteamento, encaminhamento, redirecionamento e balanceamento de carga. O controlador envia regras para a tabela de fluxos dos *switches*. É utilizado o controlador Ryu.
- **Interface de ligação sul (*Southbound*) (03)**: Protocolo de comunicação entre o controlador e os equipamentos de rede. Utiliza o padrão *OpenFlow*, já que existem funcões pré-definidas no contralor Ryu que facilitaram o desenvolvimento das aplicações.

- ***Switch* (04)**: responsável por comutar os pacotes. Em cada equipamento de rede, o cliente *OpenFlow* é configurado utilizando a tabela de encaminhamento de pacotes. Em nosso ambiente, são são criadas instâncias virtual do *Open vSwitch* pela plataforma Mininet.

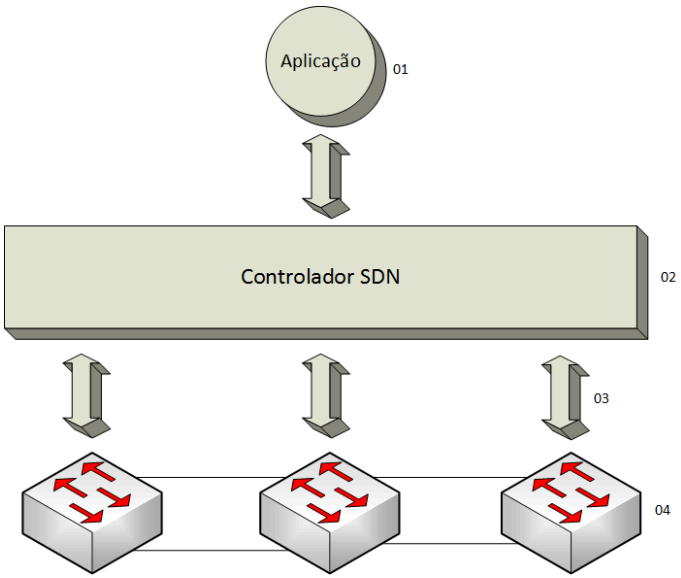


Figura 2. Arquitetura proposta neste trabalho

3.2. Implementação

O sistema foi desenvolvido utilizando o controlador Ryu, que gerencia todos os detalhes entre as conexões entre *switches*, interpretando e convertendo os pacotes de rede em objetos fáceis de usar. O aplicativo do *switch* escuta as mensagens de pacotes de entrada, mantém uma tabela interna de endereços MAC e adiciona as regras de encaminhamento nos *switches* à medida em que as conexões são identificadas utilizando o protocolo *OpenFlow*.

Essencialmente, três passos devem ser seguidos para criar uma aplicação Ryu: o registro e inicialização da aplicação ocorre antes dos *switches* serem inseridos no domínio do controlador Ryu e permite que a instância da aplicação inicialize os dados que serão compartilhados em toda a rede. Por exemplo, a aplicação *switch* inicializa uma tabela para manter informações dos endereços MAC das estações ligadas nos *switches* e as respectivas portas às quais elas estão conectadas. Em seguida acontece a inicialização de um *switch* que se conecta à controladora Ryu, ou seja, um *switch* é inserido no domínio do controlador Ryu, e assim o aplicativo pode verificar todas as suas caracterísiticas através do evento *EventOFPSwitchFeatures*. Geralmente, nesse processo, todas as regras de encaminhamento estáticas são adicionadas ao *switch*. Na aplicação *switch*, uma regra é adicionada para que qualquer pacote sem regra de encaminhamento seja enviado para o controlador. Por fim realiza-se o envio de pacotes e neste passo o aplicativo pode escolher monitorar o evento *EventOFPPacketIn*, que é responsável pelo envio de pacotes. Isso ocorre sempre quando o *switch* não possui uma regra específica para enviar aquele pacote ou caso tenha uma regra geral de encaminhamento que o instrui a enviar o pacote para o controlador analisá-lo.

Cada uma das nossas aplicações é uma modificação do aplicativo *simple_switch.py* presente na plataforma Ryu, pois nela implementamos toda a inteligência e tomada de decisão do controlador, deixando o *switch* responsável apenas por encaminhar os pacotes de acordo com as regras presentes em sua memória em tempo de execução. Essas aplicações possuem duas partes: a primeira é o processo automático de identificação dos enlaces de agregação, e a segunda o algoritmo de agregação de enlace que implementam as políticas de encaminhamento de pacotes.

3.3. Identificação dos enlaces de agregação

Quando o controlador SDN recebe a notificação da inclusão de um *switch* em seu domínio, ele verifica os enlaces desse *switch*. Caso sejam identificadas mais de uma ligação entre um mesmo par de *switches*, o controlador assume que existe uma agregação de enlaces entre esse par e cria uma tabela para ela. Com esse mecanismo, não há a necessidade da configuração manual por parte do administrador de redes. Esse processo torna a agregação de enlaces autônoma, escalável e dinâmica.

3.4. Algoritmos de agregação de enlace

Em redes SDN, temos a liberdade de implementar nosso próprio algoritmo de agregação de enlaces. Podemos classificar os algoritmos de agregação como do tipo inter-fluxo ou intra-fluxo. O tipo inter-fluxo ocorre quando cada fluxo é separado em caminhos distintos, mas não há divisão do fluxo. O tipo intra-fluxo ocorre quando existe a divisão de um fluxo em vários sub-fluxos com caminhos distintos. Diante disto, durante este trabalho, identificamos e realizamos a agregação utilizando três técnicas distintas que serão descritas a seguir.

Vale ressaltar que a plataforma de emulação Mininet não suporta a configuração do protocolo LACP entre dois *switches* virtuais. Por esse motivo, optamos por realizar a implementação de um algoritmo parecido com o LACP, batizado de Tabela *Hash*, para superar essa limitação.

3.4.1. Tabela *Hash*

As estações ligadas nos *switches* começam a transmitir dados. Quando o *switch* recebe o primeiro pacote do fluxo, ele o retransmite para o controlador, que identifica os campos disponíveis (endereço MAC de origem e destino, endereço IP de origem e destino) e calcula um *hash*, que determina qual interface será utilizada para transmitir os dados daquele fluxo. O controlador insere a regra de encaminhamento no *switch*, que encaminha os pacotes daquele fluxo pela interface calculada pelo controlador. O Algoritmo 1 representa a implementação desse procedimento.

Algorithm 1 Algoritmo implementando a política **Hash**

```
1: function CALCULA HASH(pacote)
2:     valor hash(mac origem, mac destino, ip origem, ip destino)
3:     return valor
4: end function
```

3.4.2. Análise de Tráfego

O *switch*, ao receber o primeiro pacote do fluxo, o retransmite para o controlador, que identifica a utilização da banda de cada interface da agregação para tomar a decisão de qual interface será utilizada para transmitir os dados daquele fluxo em questão. A prioridade é escolher interfaces com baixa utilização de banda, permitindo assim uma melhor distribuição do tráfego entre todas as interfaces. Apesar, dessa escolha ser realizada apenas no começo da transmissão do tráfego, eventualmente ela será refeita já que as regras criadas possuem um tempo de expiração de 30 segundos após ocorrer inatividade. O controlador insere a regra de encaminhamento no *switch*. Em seguida, o *switch* encaminha os pacotes daquele fluxo pela interface escolhida pelo controlador. Tal técnica é classificada como do tipo inter-fluxo. O Algoritmo 2 representa a implementação desse procedimento.

Algorithm 2 Algoritmo implementando a política **Análise de Tráfego**

```
1: function INTERFACE COM MENOR TRAFEGO(switch)
2:     Recupera o tráfego de todas as interfaces
3:     Identifica a interface com menor utilização de tráfego
4:     return interface
5: end function
```

3.4.3. *Round-Robin* virtual

A diferença entre esta técnica e as duas anteriores é a quantidade de regras criadas. Ao invés de criar regras para apenas um fluxo para cada transmissão de uma origem para um destino, o controlador cria regras para todas as origens e todos os destinos utilizando todas as interfaces da agregação. Porém, cada regra é criada com prioridades diferentes, fazendo com que o *switch* escolha apenas uma interface para transmitir os pacotes.

Entretanto, de tempos em tempos, o controlador muda as prioridades de todas as regras, forçando o *switch* a mudar constantemente a interface que irá utilizar para transmitir os dados. Escolhemos que essa atualização seja feita a cada 0,2 segundos. Esse valor foi determinado após a realização de vários testes para identificar qual levaria ao melhor Índice Justiça[Jain et al. 1984], que é um indicador utilizado para determinar se os usuários ou aplicativos estão recebendo uma parcela justa de banda da rede. Essa técnica é do tipo inter-fluxo, mas um fluxo pode passar por duas interfaces distintas ao longo do tempo. O Algoritmo 3 representa a implementação desse procedimento.

Algorithm 3 Algoritmo implementando a política ***Round-Robin* virtual**

```
1: while True do
2:     próximo_índice ←(índice+1)%Número_de_Regras
3:     Regra[próximo_índice].prioridade ← 50; ▷ Aumenta prioridade da próxima regra
4:     Regra[índice].prioridade ←1;            ▷ Diminui prioridade da regra atual
5:     índice ← próximo_índice;                ▷ Atualiza índice da regra atual
6:     Sleep τ ms
7: end while
```

4. Avaliação Experimental e Resultados

A proposta deste artigo inclui uma abordagem experimental. Para atingir os objetivos propostos, um ambiente virtual com três componentes foi testado: a plataforma Mininet, em sua versão mais recente, é capaz de criar redes virtuais complexas para rodar as simulações de transmissão de dados; o protocolo de comunicação Openflow, na versão 1.4, é utilizado para controlar os equipamentos SDN; a plataforma Ryu, na versão 4.9, é o controlador SDN que utiliza a linguagem Python.

Utilizamos a ferramenta *iperf* para gerar tráfego TCP e UDP entre todas as estações e servidores das topologias. Os testes variam entre apenas uma estação enviando dados até todas as estações enviando dados. O *iperf* não considera os custos adicionais dos protocolos como, por exemplo, os cabeçalhos *Ethernet*, IP, UDP ou TCP. Por isso, nos resultados apresentados, a banda máxima téorica disponível não foi atingida.

4.1. Topologia sem Agregação

Durante os testes iniciais, uma topologia sem agregação de enlaces foi criada para identificar o impacto da transmissão simultânea de pacotes entre várias origens e vários destinos. Tal topologia foi criada inspirada em um laboratório de nossa comunidade: oito estações distintas estão ligadas em um *swith* e quatro servidores estão ligados em outro *switch*. Todas as interfaces possuem velocidade máxima de transmissão de 100 Mbps e os dois *switches* estão ligados entre si. A Figura 3, substituindo a agregação por um único enlace, representa a topologia do laboratório.

A Tabela 1 representa as variações da banda (em Mbps) utilizada para as transmissões TCP de cada estação sem agregação de enlaces. A coluna banda utilizada representa o total das transmissões simultâneas. A coluna índice de Justiça (ou índice Jain) varia de a 0 (pior caso) a 1 (melhor caso) e é uma métrica para indicar o quão uniforme é a distribuição. Também fizemos experimento com UDP mas omitimos por espaço.

Transmissões	Banda utilizada por cada estação								Banda	Índice de Justiça
Simultâneas	1	2	3	4	5	6	7	8	Utilizada	
1	96,0	-	-	-	-	-	-	-	96,0	100,00%
2	53,80	27,8	-	-	-	-	-	-	96,20	99,82%
3	34,70	22,1	25,5	-	-	-	-	-	96,90	99,46%
4	20,90	14,8	17,1	14,5	-	-	-	-	96,90	99,12%
5	15,00	13,0	13,8	14,2	11,8	-	-	-	97,30	97,88%
6	7,49	15,2	12,1	12,4	12,1	11,7	-	-	97,19	98,69%
7	11,50	9,6	9,9	9,5	9,9	9,7	10,6	-	98,30	99,88%
8	11,00	8,6	9,5	9,7	11,5	9,8	8,4	7,7	98,18	98,70%

Tabela 1. Transmissões TCP simultâneas sem agregação de enlaces (Valores em Mbps)

Apesar de possuir um índice de justiça relativamente alto, quando todas as estações transmitem dados simultaneamente, elas utilizam em média uma banda de apenas 9,52 Mbps.

4.2. Tabela Hash

Em seguida, uma nova topologia foi criada no ambiente virtual utilizando a agregação de enlaces em quatro interfaces entre os *switches* interligados. Vale ressaltar que tal configuração não é possível de ser criada sem a utilização de um protocolo de agregação.

A Figura 3 representa essa nova topologia. Com isso, a banda disponível para transmissão de dados entre as estações e os servidores quadruplicou.

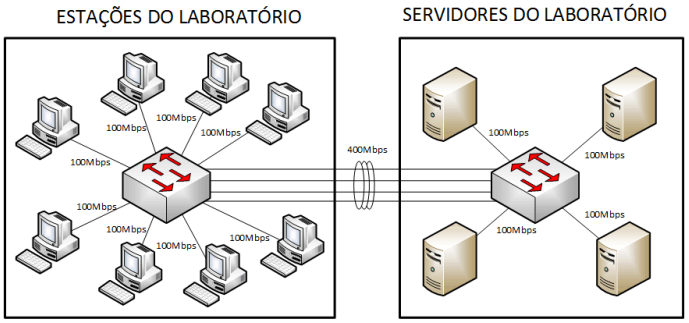


Figura 3. Topologia do laboratório com agregação de enlaces

Os resultados obtidos durante os testes dos algoritmo Tabela *Hash* são exibidos nas Tabelas 2 e 3. Nota-se que, em diversos casos, mesmo com o aumento de transmissões simultâneas, algumas estações foram beneficiadas com uma banda disponível maior do que as outras. Isso ocorre pelo fato dos fluxos serem distribuídos entre as interfaces da agregação fazendo com que a banda total consumida pela agregação não seja uniforme.

Transmissões Simultâneas	Banda utilizada por cada estação								Banda Utilizada	Índice de Justiça
	1	2	3	4	5	6	7	8		
1	83,6	-	-	-	-	-	-	-	83,6	100,00%
2	83,6	83,9	-	-	-	-	-	-	167,5	100,00%
3	75,3	29,9	30,4	-	-	-	-	-	135,6	81,85%
4	76,5	86,3	32,9	32,6	-	-	-	-	228,3	84,36%
5	19,3	81,7	82,3	16,5	20,8	-	-	-	220,6	67,01%
6	31,9	29,6	31,7	31,4	33,4	32,4	-	-	190,4	99,87%
7	11,6	34,2	87,9	87,8	34,0	24,8	29,2	-	309,5	70,67%
8	28,0	84,2	18,2	84,4	18,8	9,6	17,8	40,0	301,0	64,00%

Tabela 2. Transmissões TCP simultâneas com agregação de enlaces utilizando o algoritmo Tabela *Hash* (Valores em Mbps)

Transmissões Simultâneas	Banda utilizada por cada estação								Banda Utilizada	Índice de Justiça
	1	2	3	4	5	6	7	8		
1	88,5	-	-	-	-	-	-	-	88,5	100,00%
2	91,3	89,9	-	-	-	-	-	-	181,2	99,99%
3	53,5	95,5	42,6	-	-	-	-	-	191,6	88,69%
4	96,1	51,9	44,1	95,3	-	-	-	-	287,4	89,95%
5	43,7	45,6	95,9	52,1	52,2	-	-	-	289,5	90,00%
6	34,4	59,4	37,1	22,6	96,1	40,6	-	-	290,2	80,29%
7	46,0	95,3	44,5	53,8	48,0	51,4	43,3	-	382,3	91,20%
8	93,4	47,1	32,7	50,0	48,9	46,1	33,5	32,5	384,2	87,01%

Tabela 3. Transmissões UDP simultâneas com agregação de enlaces utilizando o algoritmo Tabela *Hash* (Valores em Mbps)

O índice de justiça é bem menor do que a topologia sem agregação, e a média da banda utilizada durante a transmissão de dados de todas as estações é de 37,62 Mbps para pacotes TCP e 48,02 Mbps para pacotes UDP.

4.3. Análise de Tráfego

Os resultados obtidos durante os testes do algoritmo Análise de Tráfego foram os piores em comparação aos outros e são exibidos nas Tabelas 4 e 5. Isso ocorre devido à criação

aleatória das regras instaladas nos *switches* a cada novo fluxo recebido. Além disso, o caminho de volta pode ser diferente do caminho de ida, o que impacta profundamente nas medições TCP.

Transmissões	Banda utilizada por cada estação								Banda	Índice de Justiça
Simultâneas	1	2	3	4	5	6	7	8	Utilizada	
1	81,7	-	-	-	-	-	-	-	81,7	100,00%
2	77,9	71,9	-	-	-	-	-	-	149,8	99,84%
3	31,3	77,1	26,7	-	-	-	-	-	135,1	79,66%
4	80,4	27,2	12,6	20,5	-	-	-	-	140,7	63,59%
5	39,1	86,1	86,8	13,5	17,0	-	-	-	242,5	69,40%
6	21,3	30,9	88,2	16,8	24,2	33,4	-	-	214,8	68,84%
7	33,1	13,3	28,5	37,0	24,4	29,2	48,9	-	214,4	90,04%
8	82,9	83,9	14,6	14,6	13,7	5,6	7,8	7,7	230,8	45,36%

Tabela 4. Transmissões TCP simultâneas com agregação de enlaces utilizando o algoritmo Análise de Tráfego (Valores em Mbps)

Transmissões	Banda utilizada por cada estação								Banda	Índice de Justiça
Simultâneas	1	2	3	4	5	6	7	8	Utilizada	
1	84,1	-	-	-	-	-	-	-	84,1	100,00%
2	93,4	93,5	-	-	-	-	-	-	186,9	100,00%
3	95,5	43,7	51,8	-	-	-	-	-	191,0	88,68%
4	96,8	46,2	50,4	96,3	-	-	-	-	289,7	89,98%
5	95,0	96,2	96,2	47,2	47,7	-	-	-	382,3	91,24%
6	96,2	49,5	38,6	96,2	57,2	46,3	-	-	384,0	88,20%
7	28,9	47,1	94,8	49,3	30,7	90,0	37,3	-	378,1	82,00%
8	24,3	50,7	22,8	28,9	91,2	45,5	20,2	92,6	376,2	74,06%

Tabela 5. Transmissões UDP simultâneas com agregação de enlaces utilizando o algoritmo Análise de Tráfego (Valores em Mbps)

A média da banda utilizada durante a transmissão de dados de todas as estações é de 28,85 Mbps para pacotes TCP e 47,02 Mbps para pacotes UDP. O índice de justiça é o menor de todos e no caso dos pacotes TCP correponde à metade do índice da topologia sem agregação.

4.4. *Round-Robin* Virtual

Os resultados obtidos utilizando o algoritmo *Round-Robin* virtual foram os mais promissores. Através dele, foi possível atingir um balanceamento de tráfego quase uniforme. As Tabelas 6 e 7 exibem os resultados desse algoritmo.

Transmissões	Banda utilizada por cada estação								Banda	Índice de Justiça
Simultâneas	1	2	3	4	5	6	7	8	Utilizada	
1	83,1	-	-	-	-	-	-	-	83,1	100,00%
2	88,9	88,9	-	-	-	-	-	-	177,8	100,00%
3	94,5	93,9	86,0	-	-	-	-	-	274,4	99,82%
4	75,5	78,9	75,0	84,8	-	-	-	-	314,2	99,75%
5	49,0	72,7	67,0	59,4	54,9	-	-	-	303,0	98,10%
6	37,5	52,7	53,9	75,2	40,7	69,0	-	-	329,0	94,15%
7	61,3	43,1	39,0	52,2	34,7	51,3	49,6	-	331,2	96,99%
8	27,5	33,9	46,3	57,3	57,8	45,2	30,9	40,2	339,1	94,02%

Tabela 6. Transmissões TCP simultâneas com agregação de enlaces utilizando o algoritmo *Round-Robin* Virtual (Valores em Mbps)

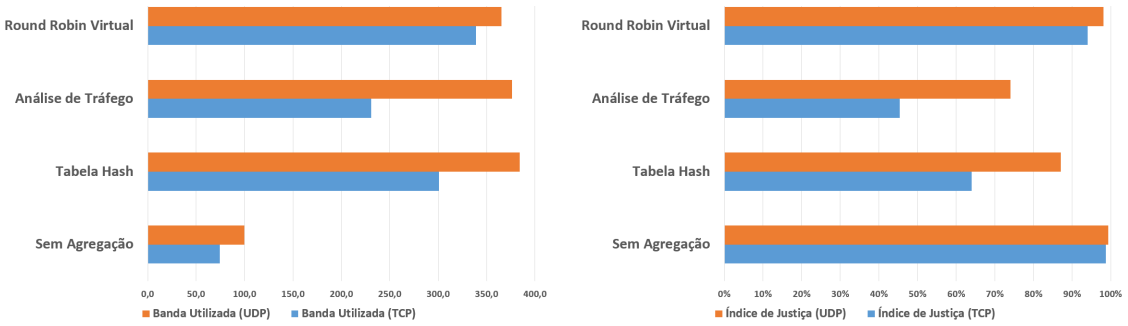
Transmissões Simultâneas	Banda utilizada por cada estação								Banda Utilizada	Índice de Justiça
	1	2	3	4	5	6	7	8		
1	99,9	-	-	-	-	-	-	-	99,9	100,00%
2	99,6	99,0	-	-	-	-	-	-	198,6	100,00%
3	94,2	90,7	90,3	-	-	-	-	-	275,2	99,96%
4	80,1	82,2	87,6	88,9	-	-	-	-	338,8	99,81%
5	70,6	70,8	61,7	71,3	65,6	-	-	-	340,0	99,69%
6	40,0	70,8	60,3	64,1	58,5	66,5	-	-	360,2	97,40%
7	51,8	38,0	58,7	55,4	44,5	53,6	62,8	-	364,8	97,80%
8	37,9	58,6	46,3	50,6	48,5	41,5	41,1	40,8	365,3	98,10%

Tabela 7. Transmissões UDP simultâneas com agregação de enlaces utilizando o algoritmo *Round-Robin* Virtual (Valores em Mbps)

O índice de justiça desse algoritmo ficou acima dos 94% nos testes TCP e acima dos 98% nos testes UDP. Além disso, as médias das bandas utilizadas durante as transmissões de dados de todas as estações foram de 42,38 Mbps e 45,66 Mbps, respectivamente. Esses valores correspondem a 84% e 91,4% da média de 50 Mbps, caso o balanceamento fosse uniforme para todas as transmissões.

4.5. Discussão

A Figura 4(a) representa o comparativo das bandas totais utilizadas e a Figura 4(b) representa o comparativo dos índices de Justiça em cada teste. Por esses resultados podemos identificar que o algoritmo *Round Robin* Virtual teve desempenhos semelhantes entre o tráfego TCP e UDP tanto na distribuição de banda quanto no índice de justiça.



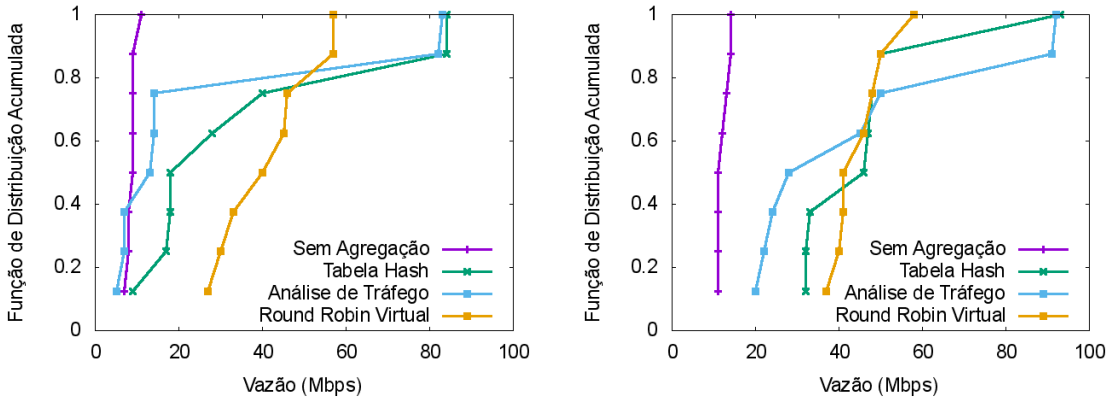
(a) Banda Total (b) Justiça

Figura 4. Comparativo dos algoritmos de agregação

As Figuras 5(a) e 5(b) representam a função de distribuição acumulada dos testes com oito fluxos simultâneos. Os gráficos demonstram que, tanto para o tráfego TCP quanto para o UDP, apenas o algoritmo *Round Robin* Virtual teve uma distribuição de tráfego tão uniforme como a topologia inicial com gargalo.

4.6. Agregação com nove enlaces

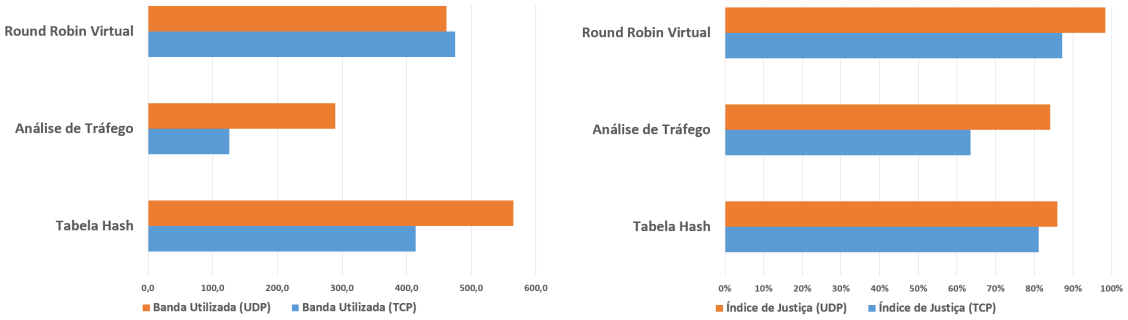
Para fins de critérios comparativos entre nossa solução e o LACP, foi criada uma nova topologia semelhante à topologia da Figura 3 com algumas modificações: a adição de uma nona estação e a configuração de nove enlaces (oito com banda de 100 Mbps e um com banda de 1 Gbps). Essas modificações foram feitas para suprir a limitação do LACP de não ser capaz de agregar mais de oito enlaces e nem de utilizar enlaces com bandas diferentes. A Figura 6(a) representa o comparativo das bandas totais utilizadas e a



(a) Tráfego TCP (b) Tráfego UDP

Figura 5. Função de Distribuição Acumulada

Figura 6(b) representa o comparativo dos índices de justiça nos testes nessa topologia. Nota-se que, em alguns resultados, a banda é superior a 400 Mbps. Isso ocorre devido ao tempo de execução dos testes, fazendo com que algum fluxo possa terminar antes de outros aumentando a vazão final.



(a) Banda Total (b) Justiça

Figura 6. Comparativo dos algoritmos de agregação

5. Conclusão e Trabalhos Futuros

Os resultados obtidos neste trabalho mostram que a nova abordagem proposta para agregação de enlaces em ambientes de redes SDN apresenta um melhor desempenho. O dinamismo, a complexidade e a escalabilidade dos ambientes de redes atuais justificam a necessidade de encontrar uma nova solução para o antigo problema de gargalo, que é a principal contribuição aqui apresentada.

A decisão de modificar o caminho dos pacotes de tempos em tempos é a melhor forma de balancear o tráfego entre todas as interfaces de uma agregação, dentre as propostas aqui discutidas. Isso fez com que a distribuição se tornasse mais justa e que não houvesse uma sobrecarga em algumas interfaces ou subutilização de outras, como pode ocorrer no protocolo LACP. Além disso, a nossa solução foi capaz de utilizar 84% da banda disponível transmitindo tráfego TCP e 91,4% transmitindo tráfego UDP. O índice

de justiça também foi bastante satisfatório, ultrapassando 94% nos tráfegos TCP e 98% nos tráfegos UDP.

Também foi implementada uma aplicação capaz de criar e remover enlaces da agregação de maneira dinâmica. Através de um simples comando, o controlador SDN consegue reconfigurar a agregação entre um par de *switches* sem a necessidade de configurações adicionais por parte do adminstrator de redes. Vale lembrar que, na solução clássica, as novas interfaces devem ser inseridas manualmente na interface virtual criada pelo protocolo.

Vale salientar que existem limitações para configurar o protocolo LACP no ambiente virtual Mininet, por isso foi realizada a implementação de um algoritmo similar que chamados de Tabela Hash. Pretendemos configurar um ambiente real para simulações, e com isso realizar os devidos testes com o estado da arte, a fim de validar nossa aboradgem.

Como trabalho futuro, pretendemos integrar ao sistema uma ferramenta de predição de tráfego para permitir engenharia de tráfego em tempo real. Além disso, a reconfiguração da agregação quando uma interface se torna indisponível precisa ser implementada. Por fim, o impacto da reescrita dos fluxos de tempos em tempos precisa ser avaliada, já que isso pode ser o causador de eventuais atrasos na transmissão de pacotes.

Referências

Ahn, J. H., Binkert, N., Davis, A., McLaren, M., and Schreiber, R. S. (2009). Hyperx: topology, routing, and packaging of efficient large-scale networks. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, pages 1–11.

Al-Fares, M., Radhakrishnan, S., Raghavan, B., Huang, N., and Vahdat, A. (2010). Hedera: Dynamic flow scheduling for data center networks. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, NSDI'10, pages 19–19, Berkeley, CA, USA. USENIX Association.

Alizadeh, M., Greenberg, A., Maltz, D., Padhye, J., Patel, P., Prabhakar, B., Sengupta, S., and Sridharan, M. (2010). Dctcp: Efficient packet transport for the commoditized data center. Technical report.

Bredel, M., Bozakov, Z., Barczyk, A., and Newman, H. (2014). Flow-based load balancing in multipathed layer-2 networks using openflow and multipath-tcp. In *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, HotSDN '14, pages 213–214, New York, NY, USA. ACM.

Carter, R. L. and Crovella, M. E. (1996). Measuring bottleneck link speed in packet-switched networks. *Performance Evaluation*, 27–28:297—-318.

Floyd, S. and Jacobson, V. (1995). Link-sharing and resource management models for packet networks. *IEEE/ACM Transactions on Networking*, 3:365–386.

Jain, R., Chiu, D., and Hawe, W. (1984). A quantitative measure of fairness and discrimination for resource allocation in shared systems. *Digital Equipment Corporation. Tech. rep., Technical Report DEC-TR-301*.

Judd, G. (2015). Attaining the promise and avoiding the pitfalls of tcp in the datacenter. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, pages 145–157, Oakland, CA. USENIX Association.

Moore, J. T. and Nettles, S. M. (2001). Towards practical programmable packets. In *in Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001*, pages 41–50.

Pfaff, B., Pettit, J., Amidon, K., Casado, M., Koponen, T., and Shenker, S. (2009). Extending networking into the virtualization layer. In *Hotnets*.

Ranum, M. J., Landfield, K., Stolarchuk, M., Sienkiewicz, M., Lambeth, A., and Wall, E. (1998). Implementing a generalized tool for network monitoring. *Information Security Technical Report*, 3:53—-64.

Seaman, M. (1999). Link aggregation control protocol. *IEEE http://grouper. ieee. org/-groups/802/3/ad/public/mar99/seaman*, 1:0399.

Steinbacher, M. and Bredel, M. (2015). Lacp meets openflow – seamless link aggregation to openflow networks. *TNC15 Networking Conference*.

Vencioneck, R. D., Vassoler, G., Martinello, M., Ribeiro, M. R. N., and Marcondes, C. (2014). Flexforward: Enabling an sdn manageable forwarding engine in open vswitch. *10th International Conference on Network and Service Management (CNSM) and Workshop*, pages 296–299.

Realização

Apoio Fomento

Apoio Institucional

Patrocinador Diamante

Núcleo de Informação e Coordenação do Ponto BR

Comitê Gestor da Internet no Brasil

Patrocinador Ouro

Patrocinador Bronze