SBRC 2019

Gramado | RS

XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos

6 a 10 de maio

Gramado | RS

ANAIS DO WPEIF 2019

sbrc2019.sbc.org.br

XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos

6 a 10 de maio

Gramado | RS

ANAIS DO WPEIF 2019

sbrc2019.sbc.org.br

EDITORA
Sociedade Brasileira de Computação (SBC)

ORGANIZAÇÃO
Alberto Egon Schaeffer Filho (UFRGS)
Weverton Luis da Costa Cordeiro (UFRGS)
Stênio Fernandes (UFPE)
Flávio de Oliveira Silva (UFU)
Antônio Jorge Gomes Abelém (UFPA)
José Ferreira Rezende (RNP)

REALIZAÇÃO
Sociedade Brasileira de Computação (SBC)
Instituto de Informática - Universidade Federal do Rio Grande do Sul (UFRGS)
Laboratório Nacional de Redes de Computadores (LARC)

Copyright ©2019 da Sociedade Brasileira de Computação
Todos os direitos reservados

Capa: Dórian Fogliatto
Produção Editorial: Carlos Raniery P. dos Santos (UFSM)

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

X Workshop de Pesquisa Experimental da Internet do Futuro (10: 2019: Gramado, RS).

Anais / X Workshop de Pesquisa Experimental da Internet do Futuro - WPEIF; organizado por Alberto Egon Schaeffer Filho, Weverton Luis da Costa Cordeiro, Stênio Fernandes, Flávio de Oliveira Silva, Antônio Jorge Gomes Abelém, José Ferreira Rezende - Porto Alegre: SBC, 2019

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

1. Redes de Computadores. 2. Sistemas Distribuídos. I. Schaeffer-Filho, Alberto Egon II. Cordeiro, Weverton Luis da Costa III. Fernandes, Stênio IV. Silva, Flávio de Oliveira V. Abelém, Antônio Jorge Gomes VI. Rezende, José Ferreira VII Título.

Sociedade Brasileira de 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
Renata Mendes de Araujo (UPM), Diretora 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
Ricardo de Oliveira Anido (UNICAMP), Diretor de Relações Profissionais
Esther Colombini (UNICAMP), Diretora de Competições Científicas
Raimundo José de Araújo Macêdo (UFBA), Diretor de Cooperação com Sociedades Científicas
Cláudia Cappeli (UNIRIO), Diretora 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.sbc.org.br

Laboratório Nacional de Redes de Computadores (LARC)

Diretor do Conselho Técnico-Científico
Paulo André da Silva Gonçalves (UFPE)

Vice-Diretora do Conselho Técnico-Científico
Rossana Maria de Castro Andrade (UFC)

Diretor Executivo
Ronaldo Alves Ferreira (UFMS)

Vice-Diretor Executivo
Danielo Gonçalves Gomes (UFC)

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 Mato Grosso do Sul (UFMS)
Faculdade de Computação
Caixa Postal 549
CEP 79.070-900 Campo Grande - MS - Brasil
http://www.larc.org.br

Organização do SBRC 2019

Coordenadores Gerais
Alberto Egon Schaeffer Filho (UFRGS)
Weverton Luis da Costa Cordeiro (UFRGS)

Coordenadores do Comitê de Programa
Antônio Jorge Gomes Abelém (UFPA)
Fabíola Gonçalves Pereira Greve (UFBA)

Coordenador de Palestras e Tutoriais
Ítalo Cunha (UFMG)

Coordenador de Painéis e Debates
Artur Ziviani (LNCC)

Coordenador de Minicursos
Miguel Elias Mitre Campista (UFRJ)

Coordenador de Workshops
Stênio Fernandes (UFPE)

Coordenador do Salão de Ferramentas
Leandro Villas (UNICAMP)

Coordenador do Concurso de Teses e Dissertações
Daniel Fernandes Macedo (UFMG)

Coordenadores do Hackaton
Raquel Lopes (UFMG)
Luis Carlos de Bona (UFPR)

Comitê de Organização Local
Avelino Zorzo (PUCRS)
Carlos Raniery Paula dos Santos (UFSM)
Cristiano Bonato Both (Unisinos)
Guilherme Rodrigues (IFSUL Charqueadas)
Jéferson Campos Nobre (UFRGS)
Juliano Wickboldt (UFRGS)
Marcelo Caggiani Luizelli (Unipampa)
Marcelo da Silva Conterato (PUCRS)
Rafael Pereira Esteves (IFRS)
Rodrigo Mansilha (Unipampa)
Tiago Ferreto (PUCRS)
Vinícius Guimarães (IFSUL Charquadas)

Comite Consultivo
Fabíola Gonçalves Pereira Greve (UFBA)

Paulo André da Silva Gonçalves (UFPE)
Fábio Luciano Verdi (UFSCAR)
Jó Ueyama (USP)
Antônio Jorge Gomes Abelém (UFPA)
Eduardo Cerqueira (UFPA)
Luiz Fernando Bittencourt (Unicamp)
Rossana Andrade (UFC)
Michele Nogueira (UFPR)
Edmundo Madeira (UNICAMP)

Mensagem dos Coordenadores Gerais

É com grande alegria e orgulho que, após 9 anos, estamos trazendo o Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC) novamente para a cidade de Gramado. O SBRC 2019 acontece em um ano que é especial por várias razões: primeiro, o aniversário de 50 anos da ARPANET, a primeira rede a implementar o conjunto de protocolos TCP/IP, que acabaram se estabelecendo mais tarde como os fundamentos da Internet; os 30 anos da invenção da World Wide Web pelo cientista Inglês Tim Berners-Lee; a comemoração dos 30 anos do .br (country code top-level domain), o qual tem atualmente mais de 4 milhões de domínios registrados; por fim, nesse ano comemora-se os 30 anos da Rede Nacional de Ensino e Pesquisa (RNP), que iniciou em 1989 como um projeto de pesquisa. É, portanto, um imenso privilégio termos a oportunidade de sediar o SBRC em um ano que carrega tanto significado para a área de redes de computadores e sistemas distribuídos.

A 37a edição do evento se apoia no histórico de sucesso do SBRC, que tradicionalmente inclui sessões técnicas, minicursos, painéis e debates, workshops, salão de ferramentas e palestras. Além de contar novamente com um Hackathon e com um Concurso de Teses e Dissertações, teremos muitas novidades no SBRC 2019, incluindo as reuniões de mentoria, o evento MUSAS e o workshop de estudantes latino-americanos. As reuniões de mentoria permitem oportunizar conversas 1-a-1 entre estudantes de pós-graduação e pesquisadores de excelência do Brasil e do exterior. O MUSAS (MUlheres em redeS de computadores e sistemAs diStribuídos) visa fomentar conexões entre mulheres atuando nas áreas de redes de computadores e de sistemas distribuídos, no Brasil e no exterior. Já o workshop de estudantes latino-americanos (LANCOMM) serve como um ponto de encontro para estudantes da região apresentarem e discutirem o andamento de suas pesquisas, e obterem feedback construtivo de pesquisadores estrangeiros de grande prestígio.

Neste ano, a trilha principal do SBRC recebeu 228 submissões completas e que entraram no processo de avaliação. Todos os artigos receberam pelo menos 3 revisões, e após um rigoroso processo seletivo, 80 artigos foram aceitos para publicação e organizados em 23 sessões técnicas. O Salão de Ferramentas selecionou 12 trabalhos que a serem demonstrados ao longo do SBRC 2019. Por sua vez, o Concurso de Teses e Dissertações selecionou 8 dissertações de mestrado e 8 teses de doutorado para apresentação durante o evento. Além disso, esse ano o evento conta com 4 palestrantes internacionais, 3 painéis, 1 tutorial, 5 minicursos e 11 workshops.

A organização de um evento com o porte do SBRC é um processo longo, que demanda muita energia e empenho. O evento desse ano só foi possível graças ao apoio e suporte incondicional de muitos grupos de pessoas e instituições. Agradecemos em particular o apoio da SBC, do LARC, do Comitê Consultivo do SBRC e da Comissão Especial de Redes de Computadores e Sistemas Distribuídos da SBC. O evento contou com o apoio do Comitê Gestor da Internet no Brasil (CGI.br) e do Núcleo de Informação e Coordenação do Ponto BR (NIC.br), da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), da Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS) e da ACM SIGCOMM. Contamos novamente também com

o apoio institucional da Rede Nacional de Ensino e Pesquisa (RNP). Além disso, o SBRC 2019 recebeu o apoio de importantes patrocinadores, incluindo SAP, HUAWEI, DATACOM, Google, Bedu.tech, BRDigital e AdylNet. Também agradecemos à Universidade Federal do Rio Grande do Sul (UFRGS), e em especial ao Instituto de Informática (INF), pelo suporte incondicional.

Agradecemos a todas as pessoas que trabalharam diretamente na organização do evento. Obrigado pelo excelente trabalho de todos os membros do comitê de organização: Antônio Jorge Gomes Abelém/UFPA e Fabíola Gonçalves Pereira Greve/UFBA (Coordenadores do Comitê de Programa), Stênio Fernandes/UFPE (Coordenador de Workshops), Ítalo Cunha/UFMG (Coordenador de Palestras, Tutoriais e Mentoria), Artur Ziviani/LNCC (Coordenador de Painéis), Miguel Elias Mitre Campista/UFRJ (Coordenador de Minicursos), Leandro Villas/UNICAMP (Coordenador do Salão de Ferramentas), Daniel Fernandes Macedo/UFMG (Coordenador do Concurso de Teses e Dissertações), e Raquel Lopes/UFCG e Luis Carlos De Bona/UFPR (Coordenadores do Hackathon). Agradecemos também aos membros do nosso comitê local, que nos ajudaram na operacionalização de diversas tarefas relacionadas à organização: Avelino Zorzo/PUCRS, Carlos Raniery Paula dos Santos/UFSM, Cristiano Bonato Both/Unisinos, Guilherme Rodrigues/IFSUL Charqueadas, Jéferson Campos Nobre/UFRGS, Juliano Wickboldt/UFRGS, Marcelo Caggiani Luizelli/Unipampa, Marcelo da Silva Conterato/PUCRS, Rafael Esteves/IFRS, Rodrigo Mansilha/Unipampa, Tiago Ferreto/PUCRS, e Vinícius Guimarães/IFSUL Charqueadas. Agradecemos ao Luis Otávio Luz Soares, técnico administrativo do INF/UFRGS, pela imensa ajuda nos mais variados aspectos relacionados à organização do evento, assim como pelo apoio de Leandro Disconzi Vieira e Carlos Alberto da Silveira Junior, também técnicos administrativos da UFRGS. Por fim, agradecemos à nossa equipe local, formada por alunos de graduação, pós-graduação e pós-doutorando da UFRGS, pela dedicação e pronta ajuda sempre que precisamos: Arthur Selle Jacobs, Augusto Zanella Bardini, Bruno Dalmazo, Fernanda da Silva Bonetti, Guilherme Bueno de Oliveira, Guilherme Rotth Zibetti, Isadora Pedrini Possebon, Leonardo Lauryel, Libardo Andrey Quintero González, Lucas Bondan, Lucas Castanheira, Luciano Zembruzki, Mateus Saquetti, Jonatas Marques, Rafael Hengen Ribeiro, e Ricardo Parizotto.

Desejamos a todos um ótimo evento e uma semana produtiva e com muitas trocas de ideias em Gramado.

Alberto Egon Schaeffer Filho e Weverton Luis da Costa Cordeiro
Coordenadores Gerais do SBRC 2019

Mensagem do Coordenador de Workshops

Os Workshops do SBRC são uma importante oportunidade para aprofundamento no conhecimento de temas especializados ou emergentes na nossa comunidade científica. Como em anos anteriores, a coordenação manteve a *Chamada de Propostas de Workshops do SBRC*, que tem estimulado a comunidade brasileira de Redes e Sistemas Distribuídos a discutir a viabilidade de workshops com temas de pesquisa mais tradicionais, bem como estimular o debate para adoção de eventos que tratam de temas mais emergentes.

Tivemos uma diversidade salutar no Comitê de Avaliação em termos de localização geográfica e experiência, gerando, portanto, opiniões complementares sobre as propostas submetidas. Consideramos todas as propostas de altíssima qualidade, com temas, focos e escopo diversos. Nesta edição de 2019, buscamos equilibrar os benefícios trazidos aos participantes com a capacidade de alocação de espaços no evento. Desta forma, foram selecionados 10 Workshops de alta qualidade, além do tradicional Workshop da RNP (*WRNP*). Dentre as propostas aceitas, oito são reedições de workshops tradicionais do SBRC, a saber: **Gerência e Operação de Redes e Serviços** (*WGRS*), **Testes e Tolerância a Falhas** (*WTF*), **Pesquisa Experimental da Internet do Futuro** (*WPEIF*), **Segurança Cibernética em Dispositivos Conectados** (*WSCDC*), **Trabalhos de Iniciação Científica e Graduação** (*WTICG*), **Blockchain: Teoria, Tecnologias e Aplicações** (*WBlockchain*), **Clouds e Aplicações** (*WCGA*), e **Computação Urbana** (*CoUrb*). Como novidade, teremos dois novos workshops, a saber: o Workshop de **Teoria, Tecnologias e Aplicações de Slicing para Infraestruturas Softwarizadas** (*WSlice*) e o **Latin American Student Workshop on Data Communication Networks** (*LANCOMM Student Workshop*), que tem como público-alvo principal os alunos de pós-graduação e pesquisadores da América Latina, além de estreitar os relacionamentos da nossa comunidade com a Association for Computing Machinery (*ACM*).

Como coordenador dos Workshops do SBRC 2019, gostaria de agradecer a todos os envolvidos na seleção das propostas. Primeiramente aos coordenadores gerais do SBRC 2019, Weverton Cordeiro (UFRGS) e Alberto Egon Schaeffer Filho (UFRGS), pelo convite para a coordenação desta chamada de trabalhos, além de todo o apoio necessário para sua condução apropriada. Agradeço também a todos os membros do comitê de avaliação pelo imenso esforço nas revisões e discussões de alta qualidade de todas as propostas submetidas. Por fim, agradeço aos coordenadores dos workshops aceitos, pela dedicação no cumprimento dos prazos e na condução dos trabalhos internos de seleção dos artigos para seus respectivos eventos, mantendo a alta qualidade geral do SBRC. Os inscritos nos diversos workshops esperam ansiosamente pelas apresentações dos trabalhos e pelas frutíferas discussões que serão geradas.

Stênio Fernandes
Coordenador dos Workshops do SBRC 2019

Comitê de Programa

- Aloizio Pereira da Silva (Northeastern University/ PAWR PPO)
- Cristiano Both (Universidade do Vale do Rio dos Sinos)
- Daniel Corujo (Universidade de Aveiro)
- Diogo Gomes (Universidade de Aveiro)
- Dorgival Guedes (UFMG)
- Fabio David (Universidade Federal do Rio de Janeiro)
- Fabio David (UFRJ)
- João Paulo Barraca (Universidade de Aveiro)
- Michael Stanton (RNP)
- Natalia Castro Fernandes (Universidade Federal Fluminense)
- Pedro Rosa (Universidade Federal de Uberlandia)
- Rodrigo Miani (Universidade Federal de Uberlândia)

Sumário

Sessão Técnica 1

OSDFramework: an Open-and-Secure data Framework to enable interoperable applications in IoT ecosystems 2
Rodrigo Elias Francisco (UFU), Flávio de Oliveira Silva (UFU)

NovaGenesis NameBindings: Avaliação de Virtualização em Escala 8
Victor Hugo D. D'Ávila (Inatel), Élcio C. do Rosário (Inatel), José R. dos Santos (Inatel), Antônio M. Alberti (Inatel)

Implementação de um switch em P4 com suporte simultâneo a múltiplas arquiteturas de Internet do Futuro 14
José Augusto T. Gavazz (UFSCar), Michael dos Santos (UFSCar), Paulo Ditarso Maciel Jr. (UFSCar), Fábio Luciano Verdi (UFSCar), José A. S. Monteiro (UFPE)

vSDNLight: Uma Proposta de Arquitetura Leve para Provisionamento de Redes Virtuais Definidas por Softwares 20
Fernando N. N. Farias (UFPA), Antônio J. G. Abelém (UFPA)

Solução de Nodos de Baixo Armazenamento para o Futuro da Internet 26
Ramon Cordeiro (UFPA), Leonardo da Costa (UFPA), Antônio Abelém (UFPA)

Network Function Virtualization in Content-Centric Networks 32
José Castillo Lema (USP), Augusto Venâncio Neto (UFRN), Flávio de Oliveira (UFU), Sergio Takeo Kofuji (USP)

Evolução do testbed FIBRE 39
Gustavo Hermínio de Araújo (RNP), Marcos Felipe Schwarz (RNP), Luiz Eduardo Folly de Campos (RNP), Leandro Neumann Ciuffo (RNP), Michael Prieto Hernandez (RNP), Gustavo Neves Dias (RNP)

FUTEBOL - Orchestrated Optical/Wireless Experiments Across Brazil and Europe 49
Daniel F. Macedo (UFMG), Luis H. Cantelli Reis (UFMG), José Marcos S. Nogueira (UFMG), Cristiano B. Both (Unisinos), Luiz da Silva (UFRGS), Juliano Wickboldt (UFRGS), Johann Marquez-Barja (University of Antwerpen)

Projeto NECOS: Rumo ao Fatiamento Leve de Recursos em Infraestruturas de Nuvens Federadas 57
Fernando N. N. Farias (UFPA), Billy A. Pinheiro (UFPA), Antônio J. G. Abelém (UFPA), Paulo D. Maciel Jr. (UFSCar), André L. B. Rocha (UFSCar), Fábio L. Verdi (UFSCar), Augusto V. Neto (UFRN), Sand L. Correa (UFG),

Marcelo Nascimento (CPqD), Rafael Pasquini (UFU), Christian Esteve Rothenberg (UNICAMP)

5GINFIRE: Enabling an NFV Based Experimentation of Vertical Industries in the 5G Context 65
Flávio de Oliveira Silva (UFU), Pedro Frosi Rosa (UFU), Halid Hrasnica (Eurescom GmbH), Anastasius Gavras (Eurescom GmbH)

X Workshop de Pesquisa Experimental da Internet do Futuro
SBRC 2019
Sessão Técnica

OSDFramework: an Open-and-Secure data Framework to enable interoperable applications in IoT ecosystems

Rodrigo Elias Francisco[1,2], **Flávio de Oliveira Silva**[1]

[1]Faculty of Computer Science – Federal University of Uberlandia (UFU)
Uberlandia – MG – Brazil

[2]Goiano Federal Institute - Campus Morrinhos
Morrinhos - GO - Brazil

rodrigo.francisco@ifgoiano.edu.br, flavio@ufu.br

Abstract. *There is in the field of IoT the interest in innovating to allow the appearance of several applications with the great mass of data generated. However, there are difficulties regarding interoperability, security, and privacy involving data, platforms, and communication. Therefore, this article proposes OSDFramework, which implements the Open-and-Secure Data concept with a repository for semantically interoperable secure data sharing. The architecture for the framework is described and depends on the solution of challenges involving data transformations of different models for interoperability, semantic and privacy, and designing security protocols for sending and receiving data in the possible way concerned to the user.*

1. Introduction

The Internet of Things (IoT) is a theme worked by several academic and business institutions, as it enables innovations in different fields of application, such as E-Health and Industry 4.0. The interest in to use technology to control processes in the various domains has always existed, however, advances in electronics, computing, and data communication have made IoT viable, which will mass the use of technology.

Cloud Computing makes it possible to obtain economic and technical benefits, such as elasticity. There are scientific papers that address Cloud-IoT integration, for example, [Botta et al. 2016] mentioned research opportunities in this subject involving solving problems on Service Level Agreement, security, privacy and the need to improve the possibilities of migration between Cloud services.

Thus, a market was created around IoT technologies and applications that are organized in specific ecosystems involving certain business organizations that work with their already established standards and offer solutions with vertical integration regarding IoT and Cloud resources. This brings problems, e.g., situations where there is interest in creating IoT (System-of-Systems) applications that aggregate data from different Clouds that bring difficulties with semantic and technical interoperability since the standards have differences involving communication, security, privacy, format, and semantics of data related the Cloud. Thinking about the heterogeneity of IoT applications and communication infrastructures for IoT, the work of [Yaqoob et al. 2017] describes the interdependence of interoperability with scalability, flexibility, mobility, and security management.

This complexity in IoT research, involving many different factors such as data, hardware, software, communication, encourages the creation of experimentation environments. For example, the FIESTA-IoT project, according to [Nguyen et al. 2018], work towards a global IoT Experimentation-as-Service (EaaS) infrastructure/ecosystem, providing an experimental infrastructure with software, semantic techniques, processes of certification and best practices to facilitate the integration of IoT resources, testbeds infrastructure, and applications.

There is the use of specific patterns linked to open-data in the research, e.g., the [Dave et al. 2018] article worked around the integration between Building Information Modeling (BIM) and IoT, using the standardized Open Messaging Interface) and Open Data Format (O-DF). In spite of these technical difficulties, the use of open-data can be useful for the creation of these applications that consume data of different Clouds, being each Cloud related to a silo of data. However, it is necessary that the idea of open-data consider security. Therefore, this research introduces the concept of Open-and-Secure data applied to the IoT context.

The topic discussed in this paper addresses several application domains, however a use case is described for explanation, that it is an Intelligent City where there are several silos of data (such as transport, health, environmental monitoring) and there is the interest in creating an innovation ecosystem where new applications can arise from the consumption of such data.

This work is organized as follows: Section 2 describes related work regarding open-data. Section 3 describes the proposed framework and its mains requirements. Section 4 presents some final considerations and presents the future directions of this research.

2. Related Work

The literature presents important contributions and reflections for the development of this subject, since the use of open-data has already been perceived as being of interest to several IoT applications for some time. For example, the work of [Doukas and Antonelli 2014] explores a case of use of Smart Cities in the city of Barcelona on an application involving urban traffic and parking that used open data, still in this application domain the article of [Yacchirema et al. 2018] worked on a system intelligent for sleep monitoring that involved data on environmental pollution and climatic conditions related to the open data catalog of the city of Valencia in Spain, and on the other hand the work of [Kim-Hung et al. 2017] mentions the use of open data sources in Industrial Internet of Thing (IIoT).

Despite the possibilities that open-data brings about creating new applications composing data from others, there are security and privacy issues that are the targets of research projects. The FIESTA-IoT project presented by [Nguyen et al. 2018], eg, aims at an infrastructure/ecosystem for global IoT EaaS in order to allow experiments on multiple testbeds interoperably in a Cloud-IoT environment and applications that facilitate sharing (open-data) and reuse, it presents security aspects such as defining security by design and the use of an access control based on the session token on the service activation layer that allows the end user to directly interact with the structure by of web services.

The research of [Attila et al. 2016], to address the interconnection between IoT

and Health Information Systems with a Telemedicine Hub that supports open data, used an authentication and authorization mechanism, worked with randomly generated token for each Web request to validate and authorize the user and the Oauth authentication and authorization standard, and the definition and modification of data privacy parameters such as private, public, and anonymous. In addition, the Open Data protocol (OData) was used as a standard for creating and consuming queryable and interoperable terminals implemented with Representational State Transfer (RESTful) Application Programming Interface (API).

These concerns and definitions become more complex depending on the volume of data the system is working on, the type of information it is intended to generate, and the costs of computing and data communication, which in certain situations involves the use of Big Data and data analysis with advanced techniques of Artificial Intelligence and Data Mining. The work of [Mossucca et al. 2016], eg, proposed for the public transport domain, works with Big Data and the Extract Transform Load (ETL) concept, in order to obtain data from different sources as IoT devices and social network, transformation involving data cleansing and anonymization, and open data sharing, which indicates that security and privacy must properly be part of the architecture of any solution. However, these requirements are not always taken into account from the outset.

3. Proposed Architecture

The proposed strategy about open-data in order to share data from different IoT applications (with semantically different data silos) in a secure way involves the creation of a common cloud-deployable repository that receives, manages, and shares such data. Figure 1 illustrates the proposal with a situation that aims to create App#3 that uses data shared by IoT applications App#1 and App#2, which meets to the Smart City use case described in Section 1.

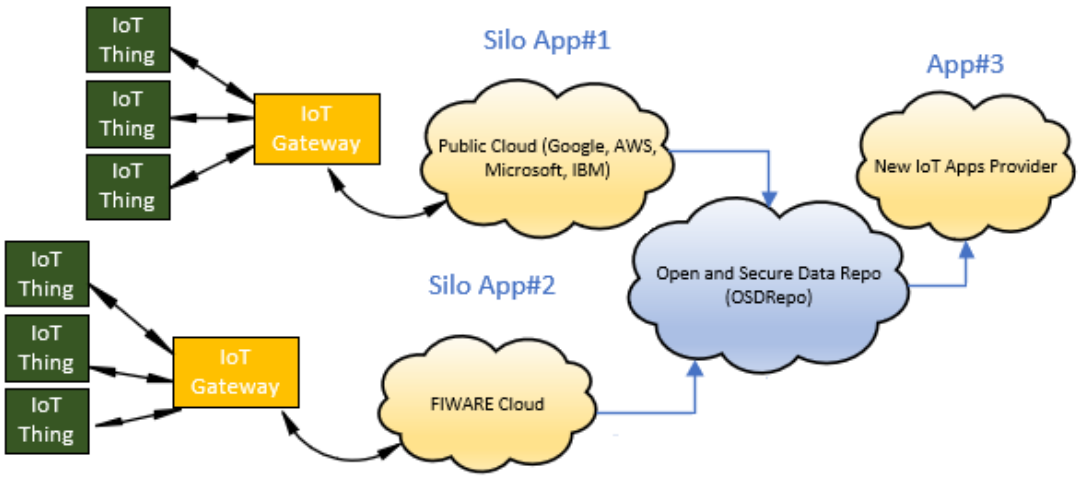


Figure 1. Creating App#3 with shared open data using the repository (OSDRepo)

To define the architecture, the following requirements have been proposed.

- Secure service to receive data

- Handle data with Common Data Model

- Translate Native Data Model to Common Data Model respecting privacy
- Allow maintain historical data (publish/subscribe mechanism)
- Write data in a secure way
- Read data in a secure way
- Access control and permissions to different cloud providers
- Secure service to share data
- Make integration process easy for both parties

The survey of these requirements made it possible to propose OSDFramework, which aims to enable IoT applications to open their data in a secure and controlled manner in order to contribute to the creation of new applications built using such data. The framework was thought for implementation in Cloud environments considering that such environments concentrate and store the IoT application data, which can be understood by the Figure 2 that describes the architecture of OSDFramework at a high level, in a way that not be difficult for both parties.

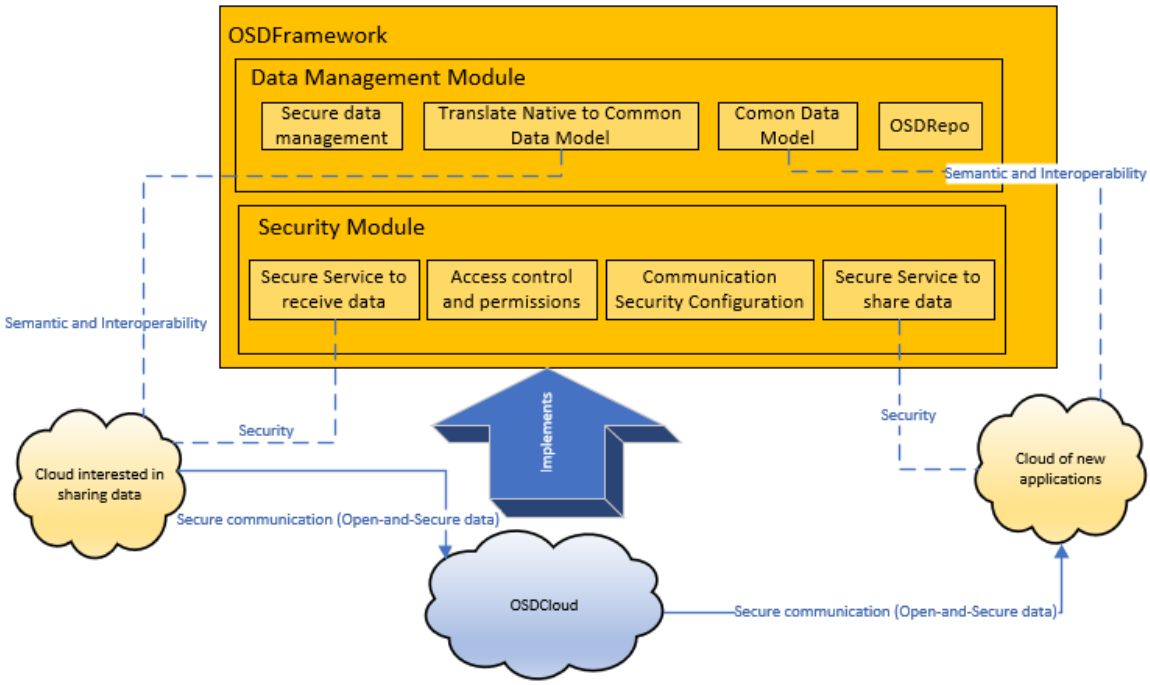


Figure 2. Architecture Design of OSDFramework

OSDFramework is divided into two modules, data management module and security module. The OSDFramework architecture was designed to deal with security and privacy issues in the process of sharing data related to the original IoT applications, since privacy eg can be achieved with strategies, e.g., the definition the appropriate granularity level of the data and the anonymization and security must be present in communication and data management. In order to be able to share data, OSDRepo has been proposed, which is a repository to open the data that is protected given the specification of the framework.

The security module includes the services of receiving and sending data and has been designed in such a way as to reduce the work of the parties with such interests. Stakeholders (to open data and receive open data) will use these services and will have to have in their computing environments resources to provide communication security (e.g. native cryptographic features) agreed with OSDCloud. In addition, the security module includes access control and permissions and the security configuration for communication agreed upon between the parties. The idea is, with these features, to flexibly work with security services, such as confidentiality, integrity, authenticity and protocols involved.

The data management module was conceived with the idea that data can come from different sources and each source have its own format and semantics, and therefore a common data model was defined that semantically represents the data to be made available in the repository and one translator from the native data model to the common data model that should consider data privacy issues. There is secure data management, which includes allowing to maintain historical data according to the publish / subscribe mechanism, and data writing and reading, which brings up research challenges on how to ensure security in this process.

4. Final Considerations and Future Directions

The idea that security should be part of the architecture discussion from the outset was used in designing this project. Thus, OSDFramework was thought of as a viable proposition, which implies being easy for entities that want to share and receive data, and secure. Several fields of application that want to innovate their ecosystem can benefit from the proposal, such as Smart Cities, Industry 4.0 and Health.

The next step of this project, which is already specified and with a defined architecture, is to solve the research problems and proceed with the implementation. The next few paragraphs present the problems to be solved.

Despite the high-level specification, it is necessary to deepen the security research applied to the framework. Security services, access control and permissions, security in sending and receiving data, protocols, encryption contribute to this step of the research. It is interesting to contribute to answer: How to define the most appropriate security configuration for a given use case?

The data management module also has challenges to be solved, such as the strategy to translate from the native data model to the common data model thinking about the diversity of schemas and technological details that may exist and computational cost thinking about an actual application.

The strategy to build the cloud repository is another important challenge. It is necessary to search in the literature the existing options that most attend the operations required for this project, such as reading and writing and with security guarantees.

References

Attila, A., Garai, Á., and Péntek, I. (2016). Common open telemedicine hub and infrastructure with interface recommendation. In *Applied Computational Intelligence and Informatics (SACI), 2016 IEEE 11th International Symposium on*, pages 385–390. IEEE.

Botta, A., De Donato, W., Persico, V., and Pescapé, A. (2016). Integration of cloud computing and internet of things: a survey. *Future Generation Computer Systems*, 56:684–700.

Dave, B., Buda, A., Nurminen, A., and Främling, K. (2018). A framework for integrating bim and iot through open standards. *Automation in Construction*, 95:35–45.

Doukas, C. and Antonelli, F. (2014). A full end-to-end platform as a service for smart city applications. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2014 IEEE 10th International Conference on*, pages 181–186. IEEE.

Kim-Hung, L., Datta, S. K., Bonnet, C., Hamon, F., and Boudonne, A. (2017). An industrial iot framework to simplify connection process using system-generated connector. In *Research and Technologies for Society and Industry (RTSI), 2017 IEEE 3rd International Forum on*, pages 1–6. IEEE.

Mossucca, L., Goga, K., Spoto, G., Bolognesi, T., and Caragnano, G. (2016). Bussola: A cloud collaborative platform of oriented services to passengers. In *Complex, Intelligent, and Software Intensive Systems (CISIS), 2016 10th International Conference on*, pages 420–425. IEEE.

Nguyen, H., Serrano, M., Gyrard, A., and Tragos, E. (2018). Fiestaiot project: Federated interoperable semantic iot/cloud testbeds and applications. In *Companion of the The Web Conference 2018 on The Web Conference 2018*, pages 425–426. International World Wide Web Conferences Steering Committee.

Yacchirema, D. C., Sarabia-Jácome, D., Palau, C. E., and Esteve, M. (2018). A smart system for sleep monitoring by integrating iot with big data analytics. *IEEE Access*, 6:35988–36001.

Yaqoob, I., Ahmed, E., Hashem, I. A. T., Ahmed, A. I. A., Gani, A., Imran, M., and Guizani, M. (2017). Internet of Things Architecture: Recent Advances, Taxonomy, Requirements, and Open Challenges. *IEEE Wireless Communications*, 24(3):10–16.

NovaGenesis NameBindings: Avaliação de Virtualização em Escala

Victor Hugo D. D'Ávila[1], **Élcio C. do Rosário**[1],
José R. dos Santos[1] **e Antônio M. Alberti**[1]

[1]ICT Lab – Instituto Nacional de Telecomunicações (INATEL)
Santa Rita do Sapucaí – MG – Brasil

victorhugo.davila@ifmg.edu.br, elcio.carlos@mtel.inatel.br,

joserodrigo@gec.inatel.br, alberti@inatel.br

Abstract. *This article presents a scale evaluation of NovaGenesis content publishing and subscription services. This study complements previous works on this proposed Future Internet architecture and validates its services through experimentation in multiple scenarios.*

Resumo. *Este artigo apresenta uma avaliação em escala dos serviços de publicação e assinatura de conteúdos da NovaGenesis. O estudo complementa trabalhos prévios dessa proposta de arquitetura de Internet do Futuro e valida seus serviços através de experimentações em múltiplos cenários.*

1. Introdução

A NovaGenesis (NG) [Alberti et al. 2017] é uma proposta de arquitetura de Internet do Futuro (IF) que possui desenho *clean slate* para nomeação, troca de dados, *cache* de rede e segurança da informação. O objetivo desse trabalho é realizar uma avaliação do mecanismo de resolução de nomes dessa arquitetura quando executado em múltiplos nós. A experimentação tornou-se importante porque a NG é estruturada por princípios diferentes da arquitetura atual de Internet e, por isso, a forma como seus elementos interagem deve ser testada em ambientes controlados e, se possível, comparada a outras propostas de IF.

A escolha do tema foi norteada pela necessidade de acompanhar as discussões sobre IF no cenário mundial. Essas discussões se baseiam na mudança de orientação do tráfego de dados que, em grande parte, deixa de ser orientado a *hosts* e passa ser orientado a conteúdos e informações [Conti et al. 2011]. Essa nova tendência, denominada *Information Centric Network*, vem sendo abordada nas propostas recentes de arquitetura de Internet e corroboram para suportar os novos requisitos de rede, as aplicações de *Internet of Things* (IoT), *streaming* massivo de vídeo, expansão do número de dispositivos móveis, mobilidade de usuários e segurança da informação.

Para alcançar o objetivo de testar em escala a NG foram utilizados softwares de virtualização e ambientes de experimentação de redes. Os resultados obtidos expõem as características da NovaGenesis e permitem acompanhar sua evolução. É esperado que a NG se destaque em alguns aspectos e seja eficiente em resolver nomes e localizar conteúdos armazenados na rede.

Esse artigo se organiza da seguinte forma: na Seção 2 são levantados os trabalhos relacionados ao tema. Na Seção 3 é definida a metodologia de trabalho para alcançar

dados com relevância científica. Os dados são exibidos e analisados na Seção 4. Na Seção 5 estão as conclusões gerais e abordagens para trabalhos futuros.

2. Trabalhos Relacionados

Os trabalhos relacionados a esse tema são aqueles que tratam de experimentação de IF, redes de computadores, plataformas de experimentação e estudos sobre as tendências para a evolução da Internet. Os trabalhos que serão mencionados a seguir se encaixam em algum desses assuntos.

O CURLING [Chai et al. 2018] é uma proposta de arquitetura de IF que otimiza a transmissão de conteúdos na rede. Nela são utilizados os conceitos de *Software-Defined Networking* (SDN), com o desacoplamento da camada de dados e de controle, e é adicionada uma terceira camada, exclusiva para controle de conteúdos. Dessa forma, passa-se a ter uma camada especializada em lidar com a troca de conteúdos e *caches* intra-redes.

O *Future Internet Brazilian Environment for Experimentation* (FIBRE) [Salmito et al.] é um *testbed* sediado no Brasil que proporciona a experimentação de arquiteturas de Internet do Futuro. O FIBRE foi base para os primeiros trabalhos da NG em ambientes de experimentação. Esse trabalho é uma continuidade e expansão dos testes realizados nesse *testbed* [Ferreira et al. 2017], em que foi definido um cenário com dois nós e avaliado o tempo de assinatura de *NameBinds* (NBs). Um NB é uma ligação entre nomes na forma $<$ X, Nome(s) $>$, ou seja, um Nome X é mapeado a um ou mais Nomes tanto em linguagem natural, quanto em nomes calculados via Função *Hash*. No trabalho citado foram avaliados o comportamento dos serviços da NG e do *testbed*, no trabalho corrente foram utilizados apenas servidores próprios.

3. Metodologia

Para realização desse trabalho foram elencados equipamentos e recursos que pudessem ser totalmente controlados e monitorados, fossem eles de *hardware* ou *software*. O principal equipamento utilizado foi um servidor com alta capacidade de processamento, descrito na Tabela 1. Como elementos de *software* foram utilizados: GNU/Linux, para o sistema operacional base do *hypervisor* e das VMs, *Kernel-based Virtual Machine* (KVM), como *hypervisor*, e a NovaGenesis, sendo executada dentro de cada VM e avaliada sob diferentes topologias.

Tabela 1. Configuração de *hardware* servidor de experimentação.

Característica	Descrição
Marca/Modelo	Dell PowerEdge T640 BCC
Processador	2x Intel R Xeon TM Silver 4114 10 cores (2.20GHz, 13.75MB Cache, 9.6GT/s)
Memória	256GB (8x32GB) RDIMM DDR4 2667 MT/s
Placa de rede	2x 10GbE: Intel I350 (10GBase-T); 2x 1GbE: Intel I350 (1000Base-T)
Armazenamento	2x SSD SATA 480GB 6 Gbps; 3x HDD SATA 4TB, 7.2K RPM

3.1. KVM

Para criação do ambiente virtual foi escolhido o *hypervisor* KVM dada a sua afinidade com plataformas de virtualização abertas e de código livre. Na Figura 1 é possível observar a organização dos elementos de *software*, desde o componente de *hardware* até o Sistema Operacional (SO) das VMs. Na camada mais baixa está o *hardware* do servidor.

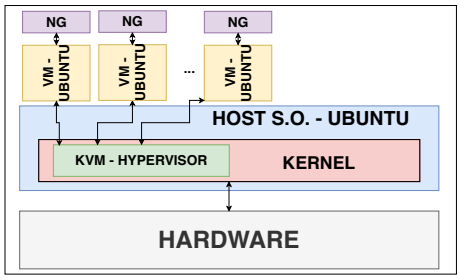


Figura 1. Topologia dos elementos de *software* da virtualização com o KVM.

Logo acima dele está o *Kernel* do SO do hospedeiro (*host*). O KVM é apresentado como parte do *kernel*. Esse é justamente seu diferencial em relação a outros hipervisores, ele é implementado diretamente no *kernel* do sistema hospedeiro para otimizar a comunicação de entrada/saída das VMs com o *hardware*. Na camada superior é possível observar o SO do *host* e as máquinas virtuais. As VMs são replicadas para atender a quantidade de nós necessários nos testes. A NG é executada em cada VM como um processo a nível de usuário.

3.2. NovaGenesis

Na NovaGenesis é utilizada uma abordagem para a troca de informações orientada a serviços. Isso facilita a expansão do número de dispositivos conectados em rede. Os três pilares da NovaGenesis são: resolução de nomes e cache de NBs e dados; ciclo de vida de serviços e conteúdos; e exposição de elementos de hardware como serviços. No primeiro pilar são propostas condições para mapear e armazenar conteúdos na rede. O segundo pilar trata todas as funcionalidades de rede como serviços e permite que eles comuniquem entre si através de *Application Programming Interfaces* (APIs). O último pilar transcende o mundo virtual e permite a representação dos elementos e dispositivos físicos como serviços de rede. Essa representação abre caminho para permitir o controle e compartilhamento de dispositivos de forma integrada [Alberti et al. 2018]. A NG é modular e seus componentes podem ser organizados em diferentes blocos, de acordo com a necessidade e/ou função que se deseja atribuir a um nó. Os serviços utilizados nesse trabalho foram:

- ***Proxy Gateway Controller Service* (PGCS):** É o serviço que inicia a exposição dos recursos e serviços de rede disponíveis em um domínio NovaGenesis. Também é o responsável pelo encapsulamento de mensagens NG sobre tecnologias de camada de enlace (e.g. Ethernet) e reconhecimento de outros nós NG em um domínio de rede.
- ***Publish/Subscribe Service* (PSS):** É o serviço responsável pelo recebimento das publicações e solicitações de assinaturas de NBs e conteúdos entre serviços e encaminhamento dos mesmos a uma instância de GIRS pré-descoberta visando balanceamento de carga.
- ***Generic Indirection Resolution Service* (GIRS):** Responsável por encontrar um *Hash Table Service* (HTS) apropriado para guardar os NBs e conteúdos associados.
- ***Hash Table Service* (HTS):** É o serviço responsável por armazenar NBs e conteúdos de forma distribuída e enviar as respostas de assinaturas ao serviço solicitante.

- **NBSimpleTestAPP**: É a aplicação criada para gerar NBs aleatórios publicando-os e assinando-os, gerando métricas para avaliar o desempenho da arquitetura.

3.3. NameBinds (NBs)

NBs são associações relacionais entre duas entidades, sejam nomes, computadores, serviços, pessoas ou elementos que se desejem mapear. Para exemplificar essa associação é possível relembrar a relação que existe entre endereços IP e nomes de domínios na Internet. O site www.example.com pode ser associado ao endereço IP: 203.0.113.1 e vice-versa. Essa associação é considerada um NB. Nesse trabalho, os NBs são gerados por uma aplicação (NBSimpleTestAPP), que possui o único objetivo de publicar e assinar NBs via PSS.

3.4. Topologias NovaGenesis

A NG é executada como um conjunto dos serviços, de forma que a execução ou não de cada serviço caracteriza uma função diferente para o nó. Foram utilizados três tipos de nós nesses experimentos:

- **Nó de controle (CORE):** Executa o PGCS, GIRS e PSS. Tem função de receber publicações e assinaturas de NBs e encaminhar para o HTS correspondente.
- **Nó de armazenamento (HTS):** Executa o PGCS e HTS. Tem função característica de cache de NBs e conteúdos. É o responsável por armazenar as publicações e responder as solicitações de assinaturas de NBs do nó de controle.
- **Nó de aplicação (APP):** Executa o PGCS e NBSimpleTestAPP. Tem característica de aplicação. É o serviço que publica o NB para o CORE e os recebe via HTS. Contabiliza o tempo gasto na assinatura dos NBs pelo HTS.

Foram utilizadas quatro topologias na execução dos testes. O cenário mais simples é composto por duas VMs, uma executando a função de (CORE) e Armazenamento (HTS) da rede e outra executando a Aplicação (NBSimpleTestAPP), como mostra a Figura 2a. Com quatro VMs é possível inserir dois nós exclusivos de armazenamento na rede (HTS), conforme ilustrado na Figura 2b. Utilizando cinco VMs aumenta-se mais um nó de armazenamento (HTS), como mostra a Figura 2c. Com seis máquinas virtuais, uma VM executa o (CORE), outra a aplicação (APP) e quatro VMs são responsáveis pelo armazenamento (HTS) de conteúdos, tal qual mostrado na Figura 2d. Um dos focos desse experimento é justamente variar a quantidade de elementos de armazenamento (HTS) na rede e observar o seu desempenho. O número de publicações realizadas pela aplicação também varia de 1, 2, 4, 8, 16 e 32 milhões. O número de assinaturas realizadas após as publicações foi constante e definido em todos cenários como 360. Para cada cenário e nível de publicações foram realizados dez experimentos.

4. Resultados e Discussões

Os principais resultados foram compilados em gráficos de tempo médio de publicações e de assinatura de NBs de acordo com o cenário de teste. A Figura 3 exibe o tempo de publicação dos NBs utilizando de 1 a 4 HTSes. A Figura 4 exibe o tempo de assinatura dos NBs utilizando também de 1 a 4 HTSes.

Com o aumento no número de HTSes é possível observar um aumento no tempo de publicação e assinatura dos NBs. Esse resultado é consequência da troca de dados entre

o nó principal (CORE) e os múltiplos nós HTS. Além disso, o GIRS passa a distribuir as solicitações de publicações e assinaturas, oriundas do (APP), para múltiplos HTSes. O

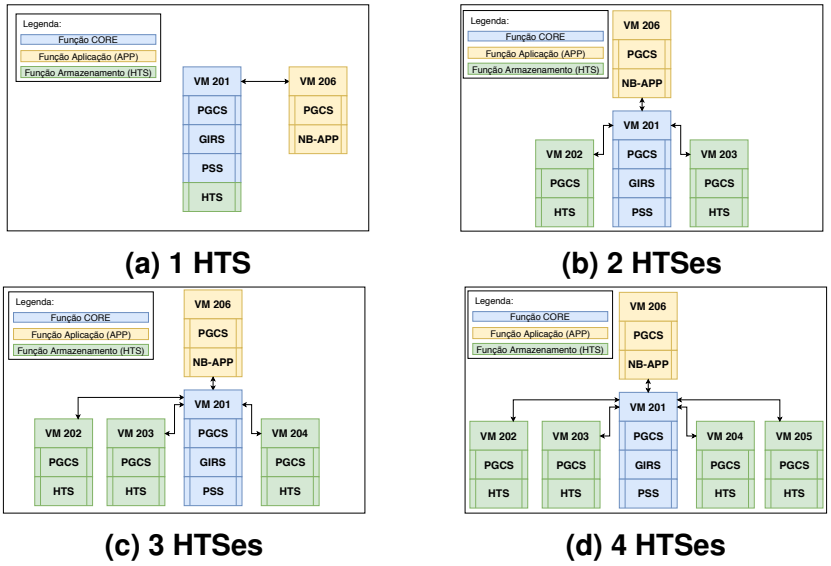


Figura 2. Cenários de VMs variando de 1 a 4 HTSes.

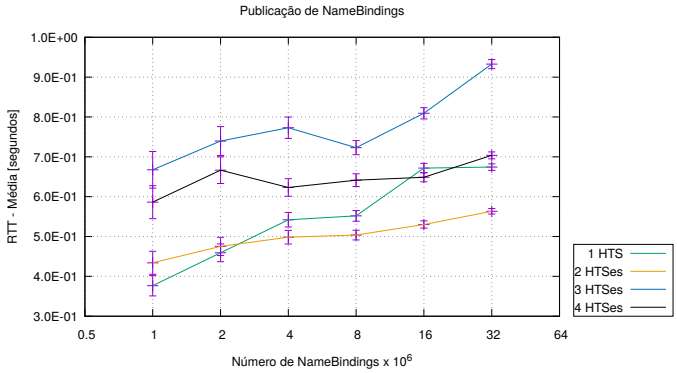


Figura 3. Tempo médio de ida e volta - *Round Trip Time* - (RTT) necessário para publicação de NBs variando de 1 a 4 HTSes.

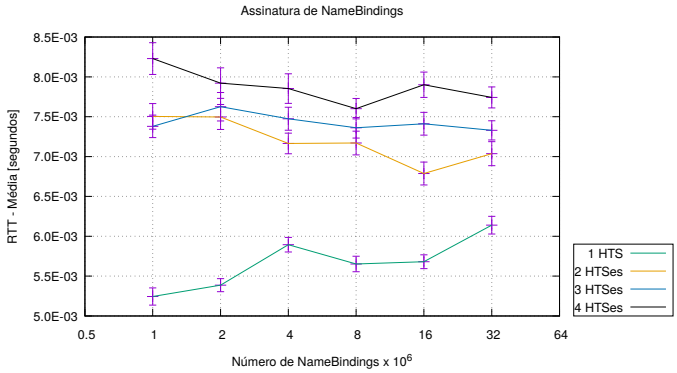


Figura 4. RTT para assinatura de NBs variando de 1 a 4 HTSes.

tempo médio de publicação é o tempo gasto para envio e confirmação de armazenamento, pelo HTS, de mil NBs. Isso é feito através de uma única mensagem NG fragmentada e transportada sobre Ethernet. O tempo médio de assinatura pode ser comparado ao modelo de um DNS local [Alberti et al. 2017]. Ou seja, o tempo médio para solicitar um NB dentre os milhões publicados. Nas avaliações aqui realizadas, esse tempo sempre foi inferior a 8,5 milissegundos, o que indica que, de forma prática, essa solução é viável para resolver nomes em um domínio local, tratando-se de uma alternativa a outros sistemas já estabelecidos para esse fim.

5. Considerações Finais

A experimentação da NovaGenesis utilizando máquinas virtuais se mostrou válida por contribuir com a evolução da arquitetura e a avaliação da mesma em cenários de múltiplos nós de *cache* intradomínio. A alteração no número de HTSes não implica necessariamente em um grande aumento de atraso, o que mostra que a solução pode ser escalável. A utilização de um ambiente totalmente virtual se mostrou flexível às demandas de execução e coordenação dos testes. Em ambientes com computadores físicos, a execução do mesmo seria mais dispendiosa e necessitaria de um tempo maior para realização. Foi dado um passo sólido ao estabelecer um protocolo de experimentação e na obtenção de dados replicáveis. Como próximos passos será possível realizar testes em plataformas diferentes, sejam elas contêineres, computadores físicos ou *testbeds* de experimentação e compará-los com os resultados aqui obtidos.

Agradecimentos

Este trabalho foi parcialmente financiado pela RNP, com recursos do MCTIC, processo Nº 01250.075413/2018-04, sob o projeto Centro de Referência em Radiocomunicações (CRR) do Instituto Nacional de Telecomunicações – Inatel, Brasil. Os autores agradecem também a FAPEMIG, CNPq, CAPES e IFMG Campus Ouro Branco.

Referências

Alberti, A. M., Casaroli, M. A. F., Singh, D., e da Rosa Righi, R. (2017). Naming and name resolution in the future internet: Introducing the NovaGenesis approach. *Future Generation Computer Systems*, 67(September):163–179.

Alberti, A. M., Bontempo, M. M., Santos, J. R., Cerqueira, A. S., e da Rosa Righi, R. (2018). NovaGenesis Applied to Information-Centric, Service-Defined, Trustable IoT/WSAN Control Plane and Spectrum Management. *MDPI - Sensors*, PP:1–34.

Chai, W. K., Pavlou, G., Kamel, G., Katsaros, K. V., Wang, N., and Member, S. (2018). A Distributed Interdomain Control System for Information-Centric Content Delivery. *IEEE Systems Journal*, PP:1–12.

Conti, M., Chong, S., Fdida, S., Jia, W., Karl, H., Lin, Y.-D. D., Mähönen, P., Maier, M., Molva, R., Uhlig, S., e Zukerman, M. (2011). Research challenges towards the Future Internet. *Computer Communications*, 34(18):2115–2134.

Ferreira, F. A., Miranda, F. S., Rosário, C. d., D'Ávila, V. H. D., e Alberti, A. M. (2017). NovaGenesis no Ambiente FIBRE: Primeiras Impressões. pages 6–9, Belém. SBRC.

Salmito, T., Ciuffo, L., Machado, I., Abelem, A. (2014). FIBRE an international testbed for future internet experimentation. Pages 969–976. Florianópolis. SBRC

Implementação de um switch em P4 com suporte simultâneo a múltiplas arquiteturas de Internet do Futuro

José Augusto T. Gavazza[1], **Michael dos Santos**[1],
Paulo Ditarso Maciel Jr.[1,2], **Fábio Luciano Verdi**[1], **José A. S. Monteiro**[3]

[1]Departamento de Computação (DComp) — UFSCar – Sorocaba

[2]Unidade Acadêmica de Informação e Comunicação — IFPB – João Pessoa

[3]Centro de Informática — UFPE — Recife

gavazza@gmail.com, michael-santos@outlook.com.br

paulo.maciel@ifpb.edu.br, verdi@ufscar.br, suruagy@cin.ufpe.br

Abstract. *New Future Internet Architectures have emerged independently in different locations, seeking to provide a new evolutionary environment for the Internet and to solve many issues that the current architecture, even with the creation of new protocols and patches, does not solve adequately. Despite efforts to deploy a single Internet architecture, what has been observed is the tendency to deploy and use different architectures that can coexist simultaneously. Therefore, there is an imminent need for a solution that makes this coexistence possible. The focus of this work is to present the implementation of a multi-architecture switch in P4 language, so that entities and applications from the same architecture can communicate.*

Resumo. *Novas arquiteturas de Internet do Futuro surgiram de forma independente em diferentes locais, buscando proporcionar um novo ambiente de evolução para a Internet e solucionar diversas questões que a arquitetura atual, mesmo com a criação de novos protocolos e patches, não resolve de maneira adequada. Apesar dos esforços em se implantar uma única arquitetura de Internet, o que se tem observado é a tendência de implantação e uso de diferentes arquiteturas que possam coexistir simultaneamente. Sendo assim, há a necessidade de uma solução que torne esta coexistência possível. O foco deste artigo é apresentar a implementação de um switch multiarquitetura em linguagem P4, para que entidades e aplicações de uma mesma arquitetura possam se comunicar.*

1. Introdução

Atualmente, existem diversas arquiteturas de rede com protótipos em diferentes estágios de implementação, cada qual com objetivos específicos de projeto e paradigmas de comunicação diferentes. Sendo assim, uma solução viável onde diferentes arquiteturas possam coexistir na mesma infraestrutura de rede é extremamente desejável.

O foco deste trabalho é implementar um switch usando a linguagem P4 capaz de distinguir diferentes arquiteturas e realizar, correta e simultaneamente, o encaminhamento

de pacotes entre entidades de mesma arquitetura de Internet. A opção pela implementação em linguagem P4 advém da possibilidade do switch ser executado em hardware programável e pela facilidade de expressar regras de encaminhamento em uma linguagem de alto nível.

O switch P4 suportará três arquiteturas de Internet: duas arquiteturas de Internet do Futuro originadas no Brasil, ETArch [de Oliveira Silva et al. 2012] e NovaGenesis [Alberti et al. 2017], além da tradicional arquitetura TCP/IP tendo em vista que esta arquitetura continuará existindo. Contudo, a solução do switch apresentado é flexível, podendo ser facilmente estendida para suportar novas arquiteturas de Internet do Futuro.

Assim como no trabalho apresentado em [Gupta et al. 2015], o switch aqui desenvolvido também pode ser utilizado em IXPs (Internet Exchange Points). Entretanto, o switch em P4 descrito neste artigo não é apenas uma proposta para um IXP tradicional (IP-only), mas sim para um novo modelo de IXP que possibilite o encaminhamento simultâneo de pacotes entre entidades de mesmas arquiteturas de Internet do Futuro. Considera-se este novo IXP como um ponto de troca de Internet do Futuro (FIXP, do inglês *Future Internet eXchange Point*).

2. Programming Protocol-independent Packet Processors – P4

P4 é uma linguagem de alto nível proposta a permitir a programação do plano de dados de dispositivos de rede e a programação de processadores de pacotes independentes de protocolo, que trabalha em conjunto com protocolos de controle de SDN (Software Defined Networking), permitindo criar soluções customizadas. Dentre os seus principais objetivos pode-se citar: (i) a reconfigurabilidade em tempo de execução, permitindo a programadores alterarem como os switches processam os pacotes depois de implementados; (ii) a independência do protocolo, não vinculando os elementos de rede a nenhum protocolo específico de redes; (iii) a independência do dispositivo alvo, permitindo que as funcionalidades de processamento de pacotes sejam independentes das especificidades do hardware alvo, através da simulação em ambiente virtualizado ou da execução em hardware programável [Bosshart et al. 2014].

Um programa implementado em P4 se utiliza da definição de cabeçalhos (*headers*), que descrevem a sequência e a estrutura de uma série de campos; de analisadores (*parsers*), que definem como os campos de cada cabeçalho são identificados e validados; de tabelas (*tables*), que são mecanismos utilizados para realizar o processamento dos pacotes; das ações (*actions*) que podem ser executadas de acordo com o processamento de pacotes; e, por fim, da implementação de programas de controle (*control programs*), que determinam a ordem das tabelas que são aplicadas a um pacote. Assim, a linguagem P4 possibilita a implementação de switches mais flexíveis, possibilitando ao programador decidir como o plano de dados processa os pacotes, sem se preocupar com detalhes de implementação.

3. Arquiteturas de Internet do Futuro

Nesta seção, são apresentadas e descritas as duas arquiteturas de Internet do Futuro inicialmente suportadas pelo switch.

3.1. ETArch – Entity Title Architecture

A ETArch [de Oliveira Silva et al. 2012] é uma Arquitetura de Internet do Futuro baseada no conceito SDN [Cox et al. 2017] e que apresenta como proposta resolver as novas demandas da Internet. Para isto, oferece suporte aos novos requisitos de aplicações como *multicast*, mobilidade, qualidade de serviço (QoS) e segurança; enquanto se mantém genérica o suficiente para suportar evoluções tecnológicas.

3.2. NG – NovaGenesis

NovaGenesis (NG) [Alberti et al. 2017] é uma Arquitetura de Internet do Futuro baseada em três pilares principais: (i) nomeação, resolução de nomes e cache de rede; (ii) dispositivos físicos, serviços e ciclos de vida de conteúdo; (iii) representantes das entidades para expor suas características e para fornecer operações definidas por software. NG visa criar uma arquitetura de informação convergente definida por um serviço que integra a troca, o armazenamento e o processamento de dados. Tal arquitetura também é escalável, concentrando-se em fornecer uma melhor coerência entre dados nomeados, interações baseadas em contratos entre entidades e controle, além de gerenciamento definido pelo serviço [Alberti et al. 2017].

4. Proposta da Solução e Implementação

Para a implementação do switch P4 proposto neste trabalho, foi utilizada a revisão de 2016 da linguagem de programação P4.

A codificação do switch P4 apresenta três seções funcionais como pode ser visto na Figura 1.

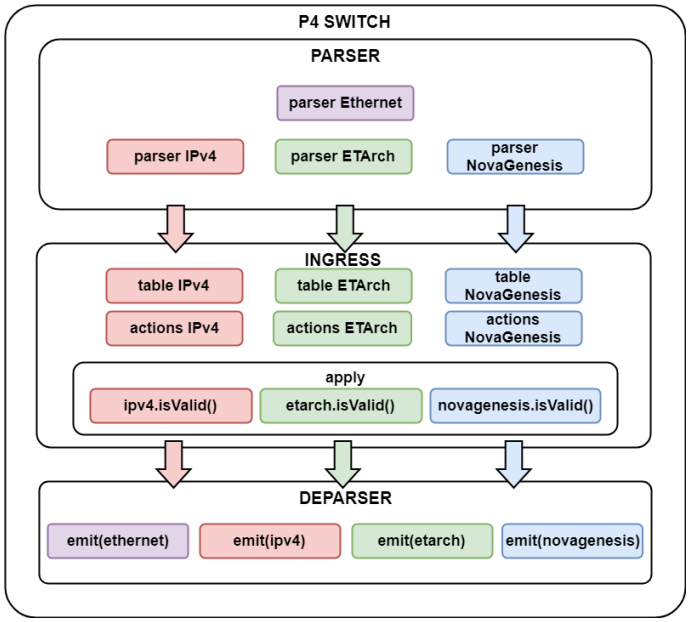


Figura 1. Representação da estrutura do código do switch P4.

A primeira seção corresponde ao analisador (*Parser*), que recebe os pacotes, extrai os cabeçalhos do protocolo Ethernet e de outros protocolos existentes e realiza a identificação da arquitetura do pacote recebido, através da análise do campo *ethertype* do

protocolo Ethernet. Na seção de ingresso (*Ingress*), são definidas as estruturas das tabelas de encaminhamento de pacotes, os campos-chave utilizados pelas regras de roteamento e a implementação das ações (*Actions*) de encaminhamento para cada uma das arquiteturas de Internet do Futuro implementadas. Por último, tem-se uma seção depuradora (*Deparser*), que remonta os campos dos cabeçalhos de cada protocolo no pacote a ser encaminhado à porta de saída correta do switch.

Embora não exista uma seção específica na linguagem P4 para a definição dos cabeçalhos das arquiteturas, os mesmos foram definidos no início do código P4 do switch, antes da codificação das principais seções funcionais. A definição das regras de encaminhamento de pacotes é configurada externamente, em arquivos individuais para cada arquitetura. A implementação foi realizada no ambiente Linux Ubuntu versão 16.10, utilizando a versão 2 do P4 *Behavioral Model* e o compilador *p4c*.

5. Prova de Conceito

A execução da prova de conceito foi realizada em um cenário virtualizado em ambiente Mininet. Foram configurados seis *hosts*, cada um com um endereço IPv4, um endereço MAC e uma *hash* de 6 bytes utilizada como identificador do título do *workspace* para a Etarch. Também foram configurados quatro switches FIXP, cada um com as regras de encaminhamento de pacotes para as arquiteturas IPv4, ETArch e NG. Dessa forma, cada dispositivo é compatível com essas arquiteturas, permitindo a cada *host* enviar e receber pacotes das três arquiteturas.

Utilizou-se a biblioteca Scapy[1] do Python para a geração e envio dos pacotes de testes, onde foram implementados os cabeçalhos das três arquiteturas abordadas. Foram criados dois programas, *send.py* e *receive.py*, que são executados nos terminais dos dispositivos. O *send.py* é utilizado para realizar a geração e envio de pacotes de uma determinada arquitetura definida pelo usuário, enquanto que o *receive.py* é utilizado para realizar o recebimento de pacotes enviados por qualquer uma das três arquiteturas definidas, independente do dispositivo que originou o pacote e também apresentar os dados do pacote recebido no terminal.

A Figura 2 ilustra a topologia utilizada para realização dos testes. Os dispositivos foram nomeados de *Host 1* a *Host 6* e a comunicação entre eles é realizada pelos quatro switches, de acordo com a figura. Cada *host* pode disparar pacotes das arquiteturas IPv4, ETArch ou NG. Nos testes realizados ficou definido que o *Host 1* envia pacotes IPv4 para o *Host 6*; o *Host 2* envia pacotes ETArch para o *Host 5*; e o *Host 3* envia pacotes NovaGenesis para o *Host 4*. A Figura 2 também apresenta os fluxos do encaminhamento dos pacotes durante os testes.

Os pacotes gerados em cada *Host* (1, 2 ou 3) são recebidos pela seção *Parser* do *Switch 1* e têm o campo Ethertype do protocolo Ethernet analisados quando chegam. Através do valor deste campo [2], o *Switch 1* identifica que o pacote pertence a uma das três arquiteturas (IPv4, ETarch ou NG). Em seguida, é invocada uma seção *Ingress* para o processamento da arquitetura correspondente, que compara o endereço de destino presente no cabeçalho do pacote e consulta as regras de encaminhamento da arquitetura correspondente, definidas na tabela de encaminhamento do *Switch 1*. Quando há correspondência

[1] https://scapy.net/.

[2] Valor 0x0800 para IPv4, valor 0x0880 para ETArch e valor 0x1234 para NG.

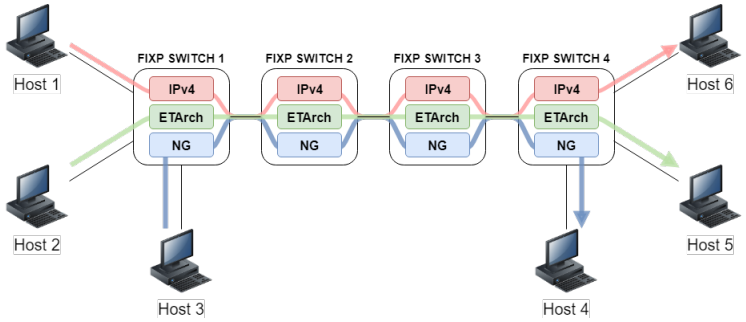


Figura 2. Encaminhamento dos pacotes durante a prova de conceito.

do destino com uma ocorrência na tabela de encaminhamento, o pacote é direcionado para a porta de saída que conecta o *Switch 1* ao *Switch 2* e o fluxo é finalizado pela seção *Deparser* que reconstitui o pacote na porta de saída determinada. Este procedimento é repetido até a chegada do pacote ao *Switch 4*, que realiza a análise do Ethertype e verifica o endereço de destino nas regras da tabela de encaminhamento, destinando assim o pacote à porta de saída onde se conecta o *Host* correspondente (6, 5 ou 4).

É importante observar que cada switch P4 atua de forma independente para cada arquitetura, proporcionando um dispositivo multiarquitetura, com suas próprias regras e tabelas de encaminhamento utilizadas de forma concorrente para cada arquitetura.

Foram executadas duas baterias de testes com o intuito de realizar uma análise inicial sobre o tempo de processamento de encaminhamento dos pacotes. As duas baterias de testes consistiram em: testes individuais para cada arquitetura e testes com as três arquiteturas funcionando simultaneamente. Em cada bateria, foram realizados experimentos com o envio de mil e dez mil pacotes, repetindo-se trinta vezes cada cenário. Os resultados do tempo médio de envio total dos pacotes (em milissegundos) podem ser observados na Tabela 1 (teste individuais) e Tabela 2 (testes simultâneos). Os intervalos de confiança para cada cenário foram obtidos com um nível de confiança de 95%.

Tabela 1. Dados do tempo médio de envio (ms) nos testes individuais.

Qtde pacotes	IPv4	ETArch	NG
1k	$7218,83 \pm 16,79$	$3638,27 \pm 27,25$	$3564,40 \pm 32,31$
10k	$72030,27 \pm 208,77$	$36258,57 \pm 154,74$	$34919,40 \pm 189,69$

Tabela 2. Dados do tempo médio de envio (ms) nos testes simultâneos.

Qtde pacotes	IPv4	ETArch	NG
1k	$21241,73 \pm 254,61$	$13114,33 \pm 183,52$	$12878,13 \pm 174,26$
10k	$179619,20 \pm 672,40$	$131786,50 \pm 583,93$	$129229,70 \pm 505,26$

Pelos dados apresentados em ambas as tabelas, nota-se que os tempos médios são estatisticamente diferentes para as arquiteturas avaliadas, com erro máximo de 1.4%, muito embora próximos quando consideradas a ETArch e NG. Pode-se observar que o IPv4 leva mais tempo para encaminhar os pacotes em comparação às demais arquiteturas. Este comportamento acontece provavelmente por uma maior inspeção de campos na arquitetura IPv4. Também é observado que, tanto individualmente quanto simultaneamente,

cada arquitetura mantém uma proporção (aparentemente linear) de crescimento de tempo da execução do cenário de mil para dez mil pacotes. Por fim, observa-se uma queda no desempenho do processo de encaminhamento de pacotes ao executar as três arquiteturas simultaneamente, quando comparados aos valores dos testes individuais.

Como resultado dos testes realizados no ambiente virtualizado, foi possível observar também que a atual implementação P4 do switch funcionou satisfatoriamente, realizando o encaminhamento de pacotes entre os dispositivos de mesma arquitetura, conforme esperado. Esta implementação cria um tratamento de pacotes dependente da arquitetura, com regras distintas de encaminhamento para cada uma das mesmas, agindo como um switch multiarquitetura.

6. Conclusão

Este trabalho apresenta a proposta e implementação de um switch multiarquiteturas programado na linguagem P4, que permite a interconexão simultânea de entidades de mesma arquitetura através da infraestrutura de rede existente. Como resultados dos testes de prova de conceito do switch P4, observa-se que o encaminhamento dos pacotes de três arquiteturas distintas de forma simultânea entre os dispositivos de mesma arquitetura foi realizado com sucesso. Como trabalho futuro, em uma segunda etapa deste trabalho, pretende-se estender a funcionalidade do switch P4 para que o mesmo possa realizar a comutação interna entre arquiteturas distintas.

Agradecimentos

Trabalho financiado pelo Projeto FAPESP “Alavancando a Internet do Futuro no Brasil através da Coexistência e Interconexão de Múltiplas Arquiteturas” (2015/24518-4). Agradecimentos também aos autores das arquiteturas ETArch e NovaGenesis pelas importantes discussões.

Referências

Alberti, A. M., Casaroli, M. A. F., Singh, D., and da Rosa Righi, R. (2017). Naming and name resolution in the future internet: Introducing the novagenesis approach. *Future Generation Computer Systems*, 67:163 – 179.

Bosshart, P., Daly, D., Gibb, G., Izzard, M., McKeown, N., Rexford, J., Schlesinger, C., Talayco, D., Vahdat, A., Varghese, G., et al. (2014). P4: Programming protocol-independent packet processors. *ACM SIGCOMM Computer Communication Review*, 44(3):87–95.

Cox, J. H., Chung, J., Donovan, S., Ivey, J., Clark, R. J., Riley, G., and Owen, H. L. (2017). Advancing Software-Defined Networks: A Survey. *IEEE Access*, 5:25487–25526.

de Oliveira Silva, F., Goncalves, M., de Souza Pereira, J., Pasquini, R., Rosa, P., and Kofuji, S. (2012). On the analysis of multicast traffic over the Entity Title Architecture. In *2012 18th IEEE International Conference on Networks (ICON)*, pages 30–35.

Gupta, A., Vanbever, L., Shahbaz, M., Donovan, S. P., Schlinker, B., Feamster, N., Rexford, J., Shenker, S., Clark, R., and Katz-Bassett, E. (2015). Sdx: A software defined internet exchange. *ACM SIGCOMM Computer Communication Review*, 44(4):551–562.

vSDNLight: Uma Proposta de Arquitetura Leve para Provisionamento de Redes Virtuais Definidas por Softwares

Fernando N. N. Farias[1], **Antônio J. G. Abelém** [1]

[1]Programa de Pós-Graduação em Ciência da Computação
Universidade Federal do Pará (UFPA)
Belém – Pará – Brasil

{fernnf,abelem}@ufpa.br

Abstract. *For building virtual software defined networks, which also are well-known as slices, is indispensable to use a hypervisor SDN. However, these solutions have introduced serious limitations of performance and scalability, because of your architecture based on a proxy. Thus, this paper presents a proposal of orchestrator that provides virtual software defined network based on a new model through the allocation of virtual switch instance on demand available in a physical infrastructure of whiteboxes.*

Resumo. *Para construção de redes virtuais definidas por softwares (vSDN), também conhecidas como slices da infraestrutura física, é necessário a utilização de soluções de hipervisores SDN. No entanto, essas soluções vêm apresentando grandes limitações de escalabilidade e desempenho, por causa de sua arquitetura baseada em proxy de serviços. Este artigo apresenta uma proposta de orquestração diferente do modelo atual provendo redes virtuais definidas por softwares através da alocação de instancias de switches virtuais sob demanda diretamente em dispositivos de comutação de baixo custo.*

1. Introdução

A redes definidas por software (SDN) [Kreutz et al. 2015] vem sendo a principal tecnologia utilizada para habilitar a virtualização de redes através de múltiplas redes lógicas isoladas em redes virtuais definidas por softwares (vSDN) [Blenk et al. 2016]. A partir do SDN foi possível otimizar a utilização de recursos físicos através da abstração de diferentes instâncias lógicas de maneira concorrente e isoladas, de forma que os recursos compartilhados são resumidamente: nós e enlaces. As vSDNs têm a habilidade de redesenhar a topologia da infraestrutura física em topologias virtuais, que por sua vez pode servir de suporte a uma aplicação ou serviço específico [Bozakov and Papadimitriou 2014].

Para habilitar o compartilhamento de uma infraestrutura física SDN por diferentes *tenants* (inquilinos) foi desenvolvida uma camada de abstração chamada de hipervisor SDN. Esta camada é responsável por monitorar, isolar e abstrair os recursos disponíveis na infraestrutura física e prover uma rede virtual para as camadas superiores através de um *slice* [Sherwood et al. 2009]. Desta forma, cada *tenant* que utilizar um *slice* pode instanciar seu próprio controlador, com a impressão que está utilizando a rede em sua plenitude quando, na verdade, ele apenas controla parte do recurso que o hipervisor lhe ofereceu.

No entanto, soluções de hipervisores, sejam elas centralizadas ou distribuídas, vêm apresentado problemas de desempenho que impactam diretamente na execução, tanto das redes virtuais como do próprio hipervisor [Blenk et al. 2016]. Isto acontece, pois, as traduções aplicadas pelo hipervisor elevam o índice de processamento que, consequentemente, degrada o desempenho do mesmo para execuções de outras funções. Além disso, esses hipervisores acumulam vários níveis de funções de controle e orquestração, causando um verdadeiro inchaço na estrutura do mesmo, ocasionando falhas que impossibilitam o aumento da escala de redes virtuais. Como exemplo, uma mensagem ou pacote pode ser descartado devido à alta utilização de recursos da rede (ex. sobrecarga de regras na tabela de encaminhamento) ou a exaustão da execução do hipervisor, como consequência, todas as redes virtuais são comprometidas e com isso o isolamento entre elas é afetado.

Devido a sua característica de abertura, o paradigma SDN vem sofrendo constantes evoluções e oferecendo oportunidades para o desenvolvimento de novas soluções em quaisquer de suas camadas. Nesse contexto, a introdução de *whiteboxes* (baseados em hardware ou software) [Farias et al. 2018b] na infraestrutura SDN permitiu a exploração de novas perspectivas de virtualização mais próximas do plano de dado. Atualmente, esses *whiteboxes* estão oferecendo dispositivos capazes de estabelecer instâncias virtuais de switches na infraestrutura física. No entanto, não existem arquiteturas definidas que explorem essas características de alocação de switches virtuais em whiteboxes para a aplicação de virtualização de redes e, com isso, apresentar um modelo de redes virtuais em SDN baseados na alocação de switches virtuais sob demanda.

Este trabalho apresenta uma proposta de arquitetura chamada de vSDNLight, que tem como objetivo propor uma arquitetura leve que define uma nova abordagem de criação de redes virtuais definidas por softwares (vSDN) através de switches whiteboxes que permitam a alocação de instâncias virtuais de switches. Neste caso o hipervisor deixa de existir entre o controlador da rede virtual e infraestrutura tal como a necessidade de traduções (i.e., *proxy*). Na proposta a camada de virtualização é parametrizada entre os switches da infraestrutura física, e o seu conjunto dessas instâncias de switches virtuais caracteriza a rede virtual.

Além desta seção introdutória, este artigo está divido em mais 3 seções. Na Seção 2, tem-se os conceitos e descrição da proposta virtualização de redes baseados na instanciação de switches virtuais. Na Seção 3, descreve-se a arquitetura do vSDNLight. Por fim, na Seção 4, tem-se as conclusões e trabalhos futuros.

2. Slice baseado em VSIs

O *vSDNlight* é um orquestrador leve para prover redes virtuais definidas por softwares, que através de agentes espalhados pela infraestrutura reduz os overheads que ocasionam problemas, tanto na escalabilidade quanto na latência das soluções atuais, afentando o desempenho desses hipervisores.

O principal desafio científico da proposta é aplicar uma nova arquitetura para a criação de slices em redes SDN. Na arquitetura cria-se um modelo de slice baseado em vínculos entre pequenas abstrações de switches virtuais nomeadas de VSI (*Virtual Switch Instance*), de forma que o conjunto dessas instâncias resultam na rede virtual ou slice. Para isso, a gerência dessas VSIs é definida por agentes que negociam com a tecnologia

de virtualização disponível no plano de dados (ex. [1]*OpenvSwitch*) a criação, atualização ou remoção das mesmas. Na Figura 1, ilustra-se a diferença entre modelo tradicional e o proposto.

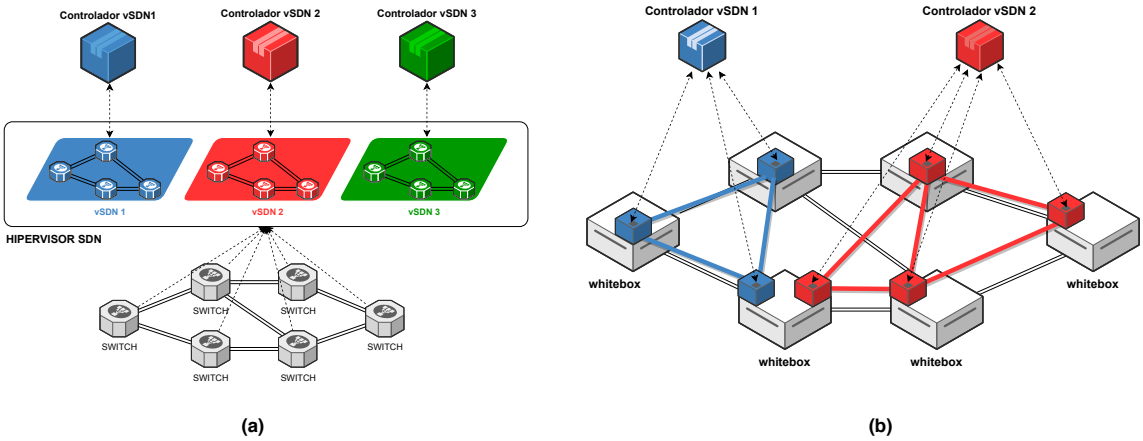


Figura 1. Diferença entre o modelo de *slice* tradicional (a) e o proposto (b).

No modelo tradicional, ilustrado na Figura 1(a), o hipervisor tem que tratar e identificar todas as mensagens recebidas de cada switch participante no *flowspace*. Logo depois, essas mensagens são reescritas e entregues ao controlador responsável por este *flowspace*. A tabela de encaminhamento é compartilhada entre os *flowspaces* nessa proposta. Além de ser um ponto de falha, isso compromete a escalabilidade, pois dependendo do número de redes virtuais essa tabela pode ser rapidamente congestionada de regras, o que inviabilizaria a sua utilização.

No modelo proposto, ilustrado pela Figura 1(b), a proposta é leve, pois não há tratamento de mensagens de controle pelo orquestrador e sim apenas o gerenciamento do ciclo de vida e monitoramento das redes virtuais, eliminando assim, a camada de virtualização no plano de controle e levando-a para mais próximo do plano de dados. Além disso, o orquestrador também coordena agentes espalhados na infraestrutura para tratar apenas das abstrações de switches virtuais, que é uma parte pequena da topologia criada, diminuindo a sobrecarga de controle e gerenciamento no sistema como um todo, se comparado a um hipervisor SDN. Desta forma, o controlador do usuário atua diretamente com o switch virtual.

Esta proposta também contribui, apresentando uma nova visualização do slice, sendo composto da junção das instâncias de virtuais de switches que compõe a topologia do slice, conforme ilustrado na Figura 2, e administrado pelo orquestrador (*vSDNLight*) que é executado no plano de gerenciamento. Na mesma Figura 2, a elipse vermelha ilustra o slice e os recursos alocados pelos agentes na infraestrutura física.

Com a remoção da camada de virtualização do plano de controle, a camada de tradução (i.e., camada de proxy) não é mais necessária, de forma que os dados pertencentes a um determinado slice são entregues diretamente a instância virtual do switch. Além disso, o controle dos slices também não são mais necessários, de forma que todo o tipo de orquestração foi migrado para plano de gerenciamento da arquitetura SDN, através do

[1]Projeto OpenvSwitch: https://www.openvswitch.org

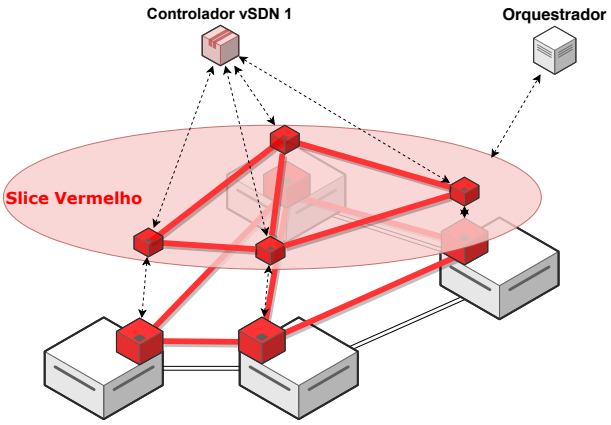


Figura 2. Visão de slice no modelo proposto.

componente *vSDNOrches*.

Por fim, a proposta ainda define dois modelos de operação do *slice*: no primeiro modo o controle e de responsabilidade do tenant, ou seja, o *slice* é controlado pelo sistema operacional de rede (NOS – *Networking Operating System*) atribuído pelo tenant com suas próprias regras de negócios. O outro modo, o controle do *slice* feito por uma interface de gerência através de um WIM (*Wide-area Network Infrastructure Manager*) [Necos 2019] aonde existe um controle da rede virtual pré-definido e apenas algumas funções especiais de controle do slice são disponibilizadas externamente.

3. Proposta do *vSDNLight*

Na arquitetura do *vSDNLight*, ilustrada na Figura 3, é possível observar os componentes responsáveis pela gerência dos recursos de virtualização disponíveis pela infraestrutura. Esta arquitetura contém dois elementos essenciais para funcionamento da orquestração: *vSDNOrches* e *vSDNAgent*. Na Figura 3, observa-se a visão geral da arquitetura destacando essas estruturas e seus componentes internos.

O *vSDNOrches* é responsável por fazer toda a orquestração dos slices e mapeá-los em forma de VSIs, além disso, ele também mapeia a interconexão entre os VSIs através de enlaces virtuais ou físicos, disponíveis na infraestrutura. A estrutura interna dele é composta pelos seguintes componentes: *Slice Manager, Slice Monitor, Topology Manager, CLI Manager e Slice Builder*.

O *Slice Manager* é responsável administrar a criação, remoção e atualização dos *slices* solicitados pelos tenants. O *slice* é o efeito do mapeamento dos nós em forma de VSIs na infraestrutura física e os enlaces virtuais podem ser um compartilhamento do enlace físico através da implementação de protocolos como *802.1q VLAN* e *802.1ad Stack VLAN* ou túneis *VXLAN* ou *GENEVE*. O *Slice Manager* também permite que o slice possa ser alocado de duas maneiras, no primeiro chamado de *on-demand*, o *tenant* repassa a topologia e o Slice Manager decide em que pontos da infraestrutura, ou seja, quais switches os elementos virtuais serão alocados. Já o segundo é chamado de self-assigned, o próprio *tenant* determina em que lugar da infraestrutura os VSIs que compõem o *slice* serão mapeados. Por fim, ele também faz a administração dos tenants e como será o controle do *slice* se através de um controlador externo ou se por um WIM.

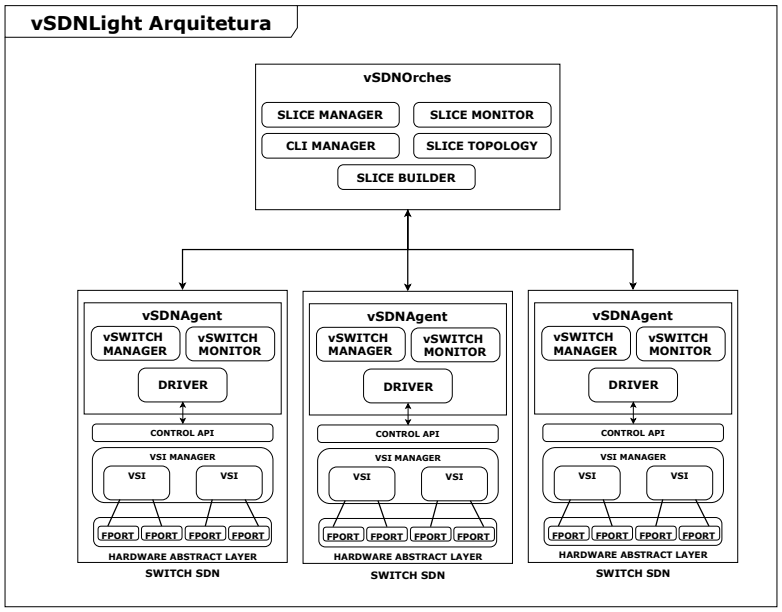


Figura 3. Arquitetura do vSDNLight.

O *Slice Monitor* permite ao vSDNOrches a capacidade de monitorar os slices e seus dispositivos físicos. Além disso, também há trocas de alertas vindos do *vSDNAgent* (ex. congestionamento de enlace e falhas tanto no switch virtual quanto o físico) ou consultas diretas aos agentes conectados aos dispositivos físicos. O *Topology Manager* é responsável por fazer à persistência dos dados da infraestrutura física e das topologias virtuais criadas. Ele é baseado em banco de dados não relacional baseado em grafos. Como as informações a serem tratadas são praticamente topologias, então esse tipo de banco de dados facilita a modelagem e otimiza as buscas de recursos. No *CLI Manager*, tem-se a interface primária com usuário onde é possível desenhar a topologia da rede virtual que será requisita ao *Slice Manager* para ser aplicada na infraestrutura física. Por fim, tem-se o *Service Bus* que o componente que permite troca de mensagens entre os componentes do *vSDNLight* com qualquer outro componente da arquitetura. No *Slice Builder*, tem-se o componente responsável por coordenar e registrar os *vSDNAgents* que entram e saem na infraestrutura. Ele é o componente responsável por negociar os recursos a serem alocados na infraestrutura. Essa negociação e definida pelo *Slice Manager* e repassada para o *Slice Builder* que aplicará a solicitação aos *vSDNAgents*.

O *vSDNOrches* é composto de 3 componentes: *vSwitch Manager*, *vSwitch Monitor* e *Driver*. No *vSwitch Manager*, tem-se o componente responsável por solicitar a criação, remoção e atualização das VSIs junto a tecnologia de virtualização disponível (ex. *OpenvSwitch* ou [2]*OF-DPA*), ele também faz integração das portas virtuais com as portas físicas, configuração dos protocolos a serem utilizados na VSI e o mapeamento do VSI ao slice solicitante. Já o *vSwitch Monitor* monitora o comportamento do VSI e encaminha para o orquestrador, além disso, ele também é capaz de enviar alertas a respeito do comportamento dos VSIs, como sobrecargas ou falhas. Já o componente Driver é responsável por fazer a comunicação do vSDNAgent com hipervisor local, o objetivo deste componente é fazer o vSDNAgent tolerar qualquer tecnologia de virtualização de switch

[2]OF-DPA Library: https://github.com/Broadcom-Switch/of-dpa

e com isso permitir que a proposta alcance a maior quantidade de switches possíveis ou fazer com que qualquer fabricante possa desenvolver a sua proposta de vSDNAgent.

4. Conclusão e Trabalhos Futuros

Esta proposta apresentou o *vSDNLight*, uma arquietura que explora a característica de *whiteboxes* de permite à criação de instâncias de switches virtuais sobre demanda. A proposta oferece uma arquitetura mais enxuta ou leve, a partir da remoção da camada de virtualização do plano de controle e a sua distribuição no plano de dados através de agentes que negociam a alocação dos switches virtuais com o switch físico. Neste caso, os elementos da rede virtual tratam diretamente com o seu controlador não precisando passar por qualquer hipervisor para fazer traduções, diminuindo o overhead do tratamento dos dados de controle quando comparado as soluções atuais.

Como trabalhos futuros, pretende-se fazer uma avaliação de desempenho do vSDNLight e compara-la com outras soluções mais conhecidas como *OpenVirteX* e *Flowvisor*. Também deseja-se avançar no desenvolvimento da arquitetura e disponibiliza-la junto ao emulador *vSDNEmul* [Farias et al. 2018a]. Este trabalho pode progredir ainda em questões como: migração de redes virtuais, tolerância a falhas e slice como serviço.

Agradecimentos

Esta pesquisa foi patrocinada pelo H2020 na 4º chamada de colaboratividade EU-BR, no acordo no 777067 (*NECOS - Novel Enablers for Cloud Slicing*), financiada pela Comissão Europeia de Tecnologia e Ministério da Ciência e Tecnologia.

Referências

Blenk, A., Basta, A., Reisslein, M., and Kellerer, W. (2016). Survey on network virtualization hypervisors for software defined networking. *IEEE Communications Surveys and Tutorials*, 18(1):655–685.

Bozakov, Z. and Papadimitriou, P. (2014). Towards a scalable software-defined network virtualization platform. In *IEEE/IFIP NOMS 2014 - IEEE/IFIP Network Operations and Management Symposium: Management in a Software Defined World*.

Farias, F. N. N., Salvador, P. M., and Abelém, A. J. G. (2018a). vSDNEmul: Emulador de Redes Definidas Por Software Usando Contêineres. In *Anais do IX Workshop de Pesquisa Experimental da Internet do Futuro (WPEIF - SBRC 2018)*, volume 9. SBC.

Farias, F. N. N., Schwarz, M. F., de Rezende, J. F., and Abelém, A. J. G. (2018b). vSDNBox: Um Hardware Especializado de Baixo Custo Gerenciado via SDN. In *Anais do XXIII Workshop de Gerência e Operação de Redes e Serviços (WGRS - SBRC 2018)*, volume 23. SBC.

Kreutz, D., Ramos, F. M. V., Veríssimo, P. E., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. (2015). Software-Defined Networking : A Comprehensive Survey. *Proceedings of the IEEE*, 103(1):14 – 76.

Necos (2019). NECOS System Architecture and Platform Specification. Acessado em: 22/03/2019. Disponivel em: http://www.maps.upc.edu/public/NECOS D3.1.final.pdf.

Sherwood, R., Gibb, G., Yap, K.-k., Appenzeller, G., Casado, M., Mckeown, N., and Parulkar, G. (2009). FlowVisor: A Network Virtualization Layer. In *Network*, page 15.

Solução de Nodos de Baixo Armazenamento para o Futuro da Internet

Ramon Cordeiro[1], **Leonardo da Costa**[1], **Antônio Abelém**[1]

[1]Faculdade de Computação – Universidade Federal do Pará (UFPA)
Belém – PA – Brazil

{ramoncord,lbc,abelem}@ufpa.br

Abstract. *In blockchain, full nodes (FNs) store all existing transactions and are responsible for validating new blocks. The amount of data stored by FNs has been increasing significantly in popular blockchains, such as that of Bitcoin. The excessive amount of data from blockchains increases the storage and processing overhead in FNs, which may cause a reduction in the number of devices that validate and store blockchain data. Also, it may put the decentralization principle of blockchain in risk. This paper proposes a less expensive data storage mechanism for blockchain FNs. The proposed mechanism aims to reduce the storage and processing overhead in FNs, and to guarantee the decentralization characteristic of the network.*

Resumo. *Em blockchain, os nodos completos (NCs) armazenam todas as transações existentes e são responsáveis por validar novos blocos. A quantidade de dados armazenados por NCs vem aumentando significativamente nas principais blockchains, como a do Bitcoin. O excesso de dados de blockchains aumenta a sobrecarga de armazenamento e processamento nos NCs, podendo causar a redução de validadores e armazenadores dos dados e colocando em risco o princípio de descentralização em blockchain. Este artigo propõe um mecanismo de armazenamento dos dados menos custoso para NCs de blockchain. O mecanismo visa diminuir a sobrecarga de armazenamento e processamento nos NCs, e garantir a característica de descentralização da rede.*

1. Introdução

Blockchain tem recebido grande atenção nos últimos anos por conta do seu sucesso na aplicação em diversas áreas, e pela garantia de confiabilidade entre partes não confiáveis [Xu 2018, Zheng et al. 2018]. A tecnologia tem sido amplamente utilizada em, por exemplo, soluções de suprimentos, governos eletrônicos e segurança de dados.

Blockchain consiste em um banco de dados descentralizado, no qual os dados são criptografados e armazenados em blocos encadeados mantidos por uma rede P2P (*peer-to-peer*) [Xu et al. 2018]. A tecnologia, vista por muitos como base para a infraestrutura da Internet em vários cenários no futuro, tem atraído grande interesse da academia e da indústria devido às características inerentes da mesma [da Costa et al. 2018]. Dentre elas, *blockchain* permite a troca segura de dados (transações) entre entes que não confiam uns nos outros. Os dados ficam armazenados permanentemente na cadeia de blocos, sendo computacionalmente muito custoso comprometer a integridade dos mesmos.

A rede P2P de uma *blockchain* é composta por uma série de nodos completos (NCs), onde cada NC armazena e verifica a validade de todos os dados já contidos na cadeia, que é gradualmente formada através de um protocolo de consenso [Kim et al. 2019]. Essa forma de armazenamento, com o passar tempo e aumento de volume de dados, causa sobrecarga de armazenamento e processamento nos NCs [Kim et al. 2019].

A necessidade de mais espaço de armazenamento para novos blocos gerados aumenta o custo de permanência ou entrada de novos NCs na rede. Como consequência, a rede pode sofrer centralização, pois somente NCs com condições para aumentar o espaço de armazenamento poderão continuar operando [Xu 2018]. Tal tendência de centralização pode também causar sobrecarga de processamento nos NCs remanescentes, visto que eles precisarão gerar novos blocos e verificar todas as transações [Xu 2018].

O objetivo deste trabalho é propor um mecanismo alternativo de armazenamento dos dados de uma *blockchain* em NCs. O mecanismo visa diminuir a sobrecarga de armazenamento nos NCs, mantendo a descentralização da rede e permitindo que dispositivos com baixa capacidade de armazenamento tornem-se NCs validadores e armazenadores da rede.

O restante desse trabalho está organizado da seguinte forma. A Seção 2 aborda os trabalhos relacionados. Em seguida, a Seção 3 apresenta a proposta, enquanto a Seção 4 realiza uma discussão sobre o mecanismo proposto. Por fim, a Seção 5 conclui o trabalho e aponta trabalhos futuros.

2. Trabalhos Relacionados

Os trabalhos da literatura que apresentam propostas para mitigar o problema de sobrecarga de armazenamento e processamento de NCs em *blockchain* normalmente utilizam algum mecanismo de compressão ou codificação de dados com o intuito de comprimir as transações já inseridas na cadeia através de técnicas advindas da teoria dos códigos [Milies 2009] ou criptografia [Shamir 1979].

Os trabalhos apresentados por [Dai et al. 2018], [Kim et al. 2019] e [Perard et al. 2018] utilizam a abordagem de codificação dos dados com o intuito de gerar economia de armazenamento. Essas propostas tem em comum a necessidade de realizar processamento na codificação e decodificação dos blocos e/ou transações. A diferença entre as propostas está na maneira com que cada mecanismo realiza o processamento.

[Dai et al. 2018] propõem um mecanismo de economia no armazenamento das transações da *blockchain* através da codificação e fatiamento dos blocos. Cada bloco gerado é particionado em sub-blocos de tamanhos iguais, que são codificados, gerando outros blocos. Um exemplo descrito por [Dai et al. 2018] em que um bloco gerado é particionado em K blocos (ex:A,B e C), em seguida é codificado em outros N blocos (p.ex:A, B, C, $A + B + C$, $A + 2B + 3C$, $A + 4B + 2C$).

O trabalho de [Kim et al. 2019] cria domínios de pares (par no conceito do autor é um nodo validador de *blockchain*) em que cada domínio é responsável por blocos específicos. Em um determinado domínio, um novo bloco é partilhado utilizando segredo de Shamir na divisão do bloco, no qual cada NC possui uma parte encriptada do bloco e uma parte de uma chave privada que é segregada entre os participantes do domínio. Im-

portante salientar que existe uma chave privada geral que é dividida (utilizando Shamir) entre todos os nodos da rede para que seja possível recuperar a chave de um domínio e consequentemente decodificar um bloco.

O trabalho de [Perard et al. 2018] propõe um novo tipo de NC, diferente de NCs tradicionais. Este novo tipo de NC armazena apenas parte da cadeia, mas atua também como validador. O armazenamento de fragmentos codificados de bloco por cada *low storage node* (LS) é obtido primeiramente particionando o bloco em partições de tamanhos iguais e gerando combinação linear dos fragmentos. A combinação linear é específica de cada bloco e cada *node*, isto é, os NCs armazenam fragmentos codificados diferentes, mesmo que aqueles sejam originários do mesmo bloco.

Os trabalhos apresentam redução da sobrecarga de armazenamento nos NCs. Entretanto, a utilização de codificação dos dados aumenta a necessidade de processamento tanto na inserção dos dados na *blockchain* quanto no processo de recuperação dos mesmos. Esta etapa dificulta que NCs com baixa capacidade computacional tornem-se validadores de transações em redes de *blockchain*. Tendo como motivação este problema, a próxima seção apresenta um mecanismo para tratar a sobrecarga de armazenamento e processamento em NCs de *blockchain*.

3. Mecanismo Proposto

Nesta seção, apresenta-se o mecanismo de armazenamento de dados da *blockchain* em NCs, objetivando reduzir a sobrecarga de armazenamento e processamento nos mesmos. *Blockchain* possui a premissa básica de descentralização, especialmente dos dados. Outra premissa básica de *blockchain* é a redundância dos dados em entes da rede que estejam interessados em contribuir com a *blockchain*. Desta forma, propor uma maneira de armazenar os dados da *blockchain* de maneira menos dispendiosa deve seguir estas premissas.

Um dos problemas das *blockchains* tradicionais é que os NCs armazenam toda a cadeia de blocos. Embora que inicialmente esta abordagem seja vantajosa, à medida que a cadeia de blocos aumenta, é necessário muito mais espaço de memória secundária disponível em cada NC. Isto pode implicar no desligamento de NCs da *blockchain* ou impossibilidade de dispositivos com menor capacidade adentrarem na rede para tornar-se NCs. A saída de NCs da rede e a impossibilidade de novos dispositivos adentrarem para desempenhar o papel de NCs pode acarretar em centralização da rede. Uma solução possível é reduzir a quantidade de dados que um NC armazena sem perder as características de elemento validador da rede e replicador dos dados (características essas que diferenciam nodos completos de clientes leves [da Costa et al. 2018]).

No mecanismo introduzido neste trabalho, propõe-se que sejam criados domínios de armazenamento, nos quais os NCs mantém redundância dos dados a serem mantidos pelo domínio. Os domínios são escaláveis de acordo com o comportamento da *blockchain*. Novos domínios podem ser criados, a partir do desmembramento de domínios que atingem o limiar de teto, ou podem ser eliminados quando este atinge o limiar inferior. Cada domínio mantém uma parte da *blockchain* e seus membros possuem as informações completas que o domínio armazena, desta forma a redundância é garantida, como mostra a figura 3.

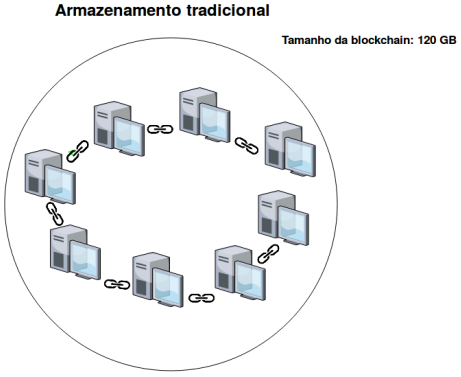
Os limiares de teto e limiar inferior, são valores previamente definidos através do consenso, em que quando o limiar de teto é atingido, um novo domínio pode ser criado, e

alguns dos membros do atual domínio se tornarão membros do novo domínio. Também, neste processo os dados do domínio anterior são divididos de maneira que parte dos dados é armazenada pelos NCs do primeiro domínio e o restante dos dados pelos membros do segundo domínio. Como inicialmente os membros do novo domínio possuem os dados de ambos os domínios, os dados do primeiro domínio podem ser apagados nestes membros.

Existe a possibilidade de um domínio perder membros de tal forma que seja atingido o limiar inferior, quando isto acontecer os membros devem ser alocados em outro domínio existente, receber os dados que os membros deste domínio possuem e replicar os seus dados com os outros NCs. Os domínios são utilizados para manter a replicação dos dados na rede, sem sobrecarregar os NCs, permitindo que NCs de menor capacidade participem da rede e esta seja mantida com a sua característica de descentralização.

Pode-se utilizar a rede do Bitcoin como exemplo de caso de uso. Atualmente esta *blockchain* possui um tamanho pouco superior a 200 Gigabytes (GB), e a rede possui em torno de 10 mil NCs. No modelo atual, cada NC do Bitcoin armazena inteiramente os mais de 200 GB, como mostra a Figura 1. Suponha-se uma situação teórica, em que no consenso seja implantado que cada domínio deve possuir pelo menos 50 NCs como limiar inferior e 200 NCs de limiar superior. Além disso, com o rearranjo da rede, suponha-se que sejam criados 100 domínios com 100 NCs cada. Neste cenário, cada domínio necessita armazenar 2 GB da *blockchain*, isto significa que cada NC armazena apenas os 2 GB delegados ao domínio à que este pertence. Neste exemplo, é perceptível a economia de espaço de armazenamento em cada um dos NCs. A Figura 2, mostra o exemplo de economia de armazenamento numa *blockchain* de 120 Gb.

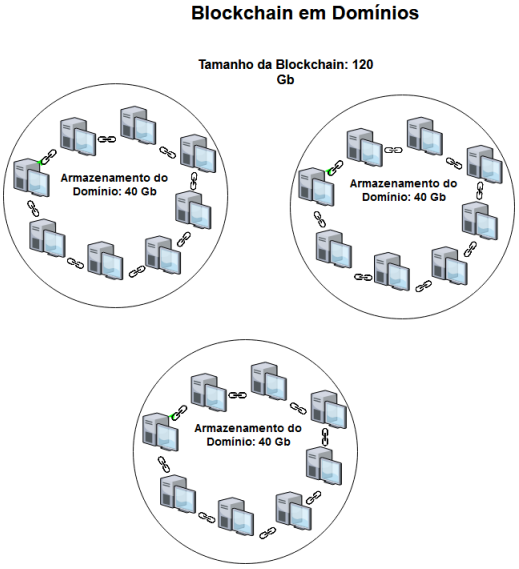
Figura 1. Armazenamento tradicional em *blockchain*.



4. Discussão

A proposta apresentada na Seção 3 possui alguns benefícios. Primeiramente, devido a redução da sobrecarga de armazenamento em cada NC, o tempo de sincronização de novos NCs deverá ser menor, pois não será necessário baixar toda a *blockchain*. O volume de dados trocados também será menor. Outra vantagem significativa é a possibilidade de utilizar dispositivos com menor capacidade de armazenamento como NC validador e mantenedor da *blockchain*, garantindo a descentralização. Isto garante também a possibilidade de utilizar dispositivos de *IoT* na *blockchain* desempenhando o papel de NCs, o que não é possível na maioria dos trabalhos apresentados na Seção 2, tendo em vista que

Figura 2. Armazenamento em forma de domínios *blockchain*.



em muitos dos trabalhos relacionados, são utilizadas técnicas de compressão de dados, normalmente baseadas em teoria dos códigos.

A utilização de compressão/descompressão no âmbito de redução de armazenamento é positiva, mas aumenta o processamento necessário em cada NC para armazenar ou recuperar informações, em alguns casos, o adicional de processamento pode inviabilizar ou dificultar a utilização de dispositivos de baixa capacidade como NC armazenador da *blockchain*. Neste aspecto, a proposta apresenta vantagem. A desvantagem das propostas apresentada na Seção 3 pode ser o tamanho dos dados armazenados por cada NC em relação a alguns trabalhos que utilizam compressão de dados no armazenamento [Dai et al. 2018, Kim et al. 2019].

5. Conclusões e Trabalhos Futuros

O mecanismo proposto é uma alternativa possível para reduzir a sobrecarga de armazenamento em *blockchain*, especialmente porque possui o foco, além do armazenamento, em manter a descentralização da rede, uma das premissas básicas de *blockchain*. A preocupação em manter a descentralização, neste caso garante a replicação dos dados, visto que o mecanismo utiliza-se da descentralização para manter os dados replicados.

Como trabalhos futuros, deverá ser implementado a proposta numa *blockchain* (preferencialmente não-permissionada e pública), realizar os testes com NCs, verificar a redução no armazenamento de dados por NC e em seguida comparar a redução com resultados obtidos em outras propostas.

Agradecimentos

Essa pesquisa teve suporte parcial da CAPES e da chamada conjunta RNP-NSF para pesquisa e desenvolvimento em cibersegurançaa (INSaNE - Improving Network Security at the Network Edge), financiada pela National Science Foundation e pelo Ministério da Ciência, Tecnologia, Inovações e Comunicações (MCTIC) através da RNP e do CTIC.

Referências

da Costa, L., Neto, A., Pinheiro, B., Araújo, R., and Abelém, A. (2018). Dlcp: Um protocolo para a operação segura de clientes leves em blockchains. In *Anais do IX Workshop de Pesquisa Experimental da Internet do Futuro (WPEIF - SBRC 2018)*, volume 9, Porto Alegre, RS, Brasil. SBC.

Dai, M., Zhang, S., Wang, H., and Jin, S. (2018). A low storage room requirement framework for distributed ledger in blockchain. *IEEE Access*, 6:22970–22975.

Kim, Y., Raman, R. K., Kim, Y.-S., Varshney, L. R., and Shanbhag, N. R. (2019). Efficient local secret sharing for distributed blockchain systems. *IEEE Communications Letters*, 23(2):282–285.

Milies, C. P. (2009). Breve introdução a teoria dos códigos corretores de erros. *Departamento de Matemática, UFMS*.

Perard, D., Lacan, J., Bachy, Y., and Detchart, J. (2018). Erasure code-based low storage blockchain node. *arXiv preprint arXiv:1805.00860*.

Shamir, A. (1979). How to share a secret. *Communications of the ACM*, 22(11):612–613.

Xu, Y. (2018). Section-blockchain: A storage reduced blockchain protocol, the foundation of an autotrophic decentralized storage architecture. In *2018 23rd International Conference on Engineering of Complex Computer Systems (ICECCS)*, pages 115–125. IEEE.

Xu, Z., Han, S., and Chen, L. (2018). Cub, a consensus unit-based storage scheme for blockchain system. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 173–184. IEEE.

Zheng, Q., Li, Y., Chen, P., and Dong, X. (2018). An innovative ipfs-based storage model for blockchain. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 704–708. IEEE.

Network Function Virtualization in Content-Centric Networks

José Castillo Lema[1], Augusto Venâncio Neto[2], Flávio de Oliveira[3], Sergio Takeo Kofuji[1]

[1]Escola Politecnica – Universidade de São Paulo

[2]Departamento de Informática e Matemática Aplicada (DIMAp) – Universidade Federal do Rio Grande no Norte

[3]Faculdade de Computação, Universidade Federal de Uberlândia

{josecastillo,kofuji}@usp.br, augusto@dimap.ufrn.br, flavio@ufu.br

Abstract. *Network Functions Virtualization (NFV) offers an alternative way to design, deploy, and manage networking functions and services by leveraging virtualization technologies to consolidate network functions into general-purpose hardware platforms. On the past years extensive effort has been made to evolve and mature NFV technologies over IP networks. However, little or no attempts at all have been made to incorporate NFV into Information-Centric Networks (ICN). This work explores the use and implementation of virtual Network Functions (VNFS) in Content-Centric Networks (CCN), and proposes the use of the Named Function Networking (NFN) paradigm as means to implement network functions and services in this kind of networks, distributing the network functions and services through the networks nodes and providing flexibility to dynamically place functions in the network as required and without the need of a central controller.*

Resumo. *Network Functions Virtualization (NFV) oferece uma maneira alternativa de projetar, implantar e gerenciar funções e serviços de rede, aproveitando as tecnologias de virtualização para consolidar funções de rede em plataformas de hardware de uso geral. Nos últimos anos, um grande esforço foi feito para desenvolver e amadurecer as tecnologias de NFV em redes IP. No entanto, pouca ou nenhuma tentativa foi feita para incorporar o NFV em redes orientadas a conteúdo (ICN). Este trabalho explora o uso e implementação de funções de rede virtualizadas (VNFS) neste tipo de redes, e propõe o uso do paradigma Named Function Networking (NFN) como meio de implementar funções e serviços de rede neste tipo de redes.*

1. General Information

Network functions, also known as ”middleboxes”, are playing an increasingly important role in modern networks, ranging from mobile networks, enterprise networks, to data-center networks. Surveys have been showing for some time that the number of network functions is comparable to that of the forwarding devices, indicating their significance (“A Survey of Enterprise Middlebox Deployments | EECS at UC Berkeley” n.d.). Network functions improve the network performance (e.g., Wide Area Network – WAN Optimizer, web proxy, load balancer), enhance the security (e.g., Intrusion Detection System – IDS / Intrusion Prevention System – IPS) or monitor the traffic (e.g., lawful interception, passive network monitor).

Conventionally, network functions were built in dedicated hardware for performance concerns, which incur high capital investment and operating expense. Furthermore, they were hard to manage. Their replacement and upgrade involve non-trial human labor. In light of this situation, Network Function Virtualization (NFV) (Li and Chen 2015) was proposed, aimed to address these issues by leveraging visualization technologies to consolidate network functions into general-purpose hardware platforms. NFV, along with Software-Defined Networking (SDN) paradigm, enables automated management of the whole life cycle of virtual network functions, leading to resource efficiency and expense reduction.

The initial perception of NFV was that virtualized capability should be implemented in data centers. But a service provider should be free to locate NFV in all possible locations, from the data center to the network node to the customer premises. This approach, known as distributed NFV, has been emphasized from the beginning as NFV was being developed and standardized, and is prominent in the NFV European Telecommunications Standards Institute (ETSI) Industry Specification Group (ISG) documents (“ETSI GS NFV-SWA 001 V1.1.1 (2014-12)” n.d.).

On the past years extensive effort has been made to evolve and mature NFV technologies over IP networks. However, current routing protocols deployed in IP networks constrain how packets can be deviated from a well-defined path (e.g., shortest path) and thus cannot take full advantage of the great flexibility offered by NFV. Moreover, SDN imposes the use of a central controller, which may lead to performance penalties and jeopardizes scalability capabilities (Bianco et al. 2010). Information-Centric Network (ICN) (Dannewitz et al. 2013) is a different approach that introduces Content Names to decouple the user interests from data location, improving in-network caching, management of mobility, multicast and peer-to-peer communications.

Following the ICN line of thinking, Named Function Networking (NFN) (Tschudin and Sifalakis 2014) has proposed that names should not only refer to data but also to functions and computation tasks. In NFN the network’s role becomes to resolve names to computations. By naming functions, the network starts acting like a computing

machine, capable of not only caching content but also computation results. While NFN was originally planned for tempering, processing and delivering data, named functions use can also be extended to deal with network management related functions.

The key contribution of this work is the use of named functions and the NFN paradigm as means to implement network functions and services in content-centric networks, distributing the network functions/services through the networks nodes and providing flexibility to dynamically place functions in the network as required. The Dynamic Named Functions Architecture (DNFA) was proposed, a scalable and flexible framework that allows placing functions in the network by leveraging the NFN layer, following the ETSI NFV architectural guidelines. Furthermore, DNFA explores the distributed nature of named functions and takes advantage of its preferential execution opportunism to gain gratuitous parallelism and asynchronous computations. DNFA can bring many benefits, from improving operational efficiency and reducing resource usage to shorter deployment/upgrade intervals and cost efficient realization of network functions in software deployed over commodity hardware, without the need of a central controller.

The remainder of this paper is structured as follows: Section II provides a related work to underline the value of our contribution. Section III presents the proposed framework, with deeper information of its mechanisms and architecture. Section IV explains its implementation and shows some results of evaluations in the control plane.

2. Related work

From the early definitions of ICN, several works implemented various network functions into this kind of networks: access control (“Access Control Enforcement Delegation for Information-Centric Networking Architectures” n.d.), access privacy (Mohaisen 2017), lightweight authentication/secured routing, service discovery and management operations, among others.

The authors themselves of the NFN paradigm have already explored content access (Marxer, Scherb, and Tschudin 2016) and monitoring protocol (Mansour and Tschudin 2016) implementation through NFN. However, these functions are manually deployed in absence of a framework to automate deploy, monitor and scale.

3. Functional Architecture

This section describes conceptually the functional architecture proposed by the Dynamic Named Function Architecture (DNFA), a scalable and flexible framework that allows placing functions in the network by leveraging the Named Function Networking (NFN) layer, following the European Telecommunications Standards Institute (ETSI) Network Function Virtualization (NFV) architectural guidelines.

DNFA aims to distribute network functions and services through the network infrastructure using the distributed nature of named functions. DNFA can bring many benefits, from improving operational efficiency and reducing resource usage to shorter deployment/upgrade intervals and cost efficient realization of network functions in software deployed over commodity hardware, without the need of a central controller.

The DNFA proposal owns the following contributions:

- an integrated management architecture, including an orchestrator platform, for the automated deployment, management, monitoring, optimization and lifecycle management (e.g. instantiation, configuration, update, scale up/down, migration, termination, etc.) of named functions over network infrastructures;
- the distribution of network services/functions through the network infrastructure taking advantage of the distributed nature of named functions, using its preferential execution opportunism to gain gratuitous parallelism and asynchronous computations;
- a native and purely content-based non central controller-dependant network function execution environment, allowing the creation of more agile content-centric networks;

DNFA architecture is based on the ETSI NFV architectural guidelines. Fig. 1 represents the reference architecture of a DNFA framework. The bottom left of the picture represents the NFN Infrastructure, containing the Network Elements (NEs) that support ICN resolution (NFN capable or not). This infrastructure provides the necessary resources to physically span over several locations.

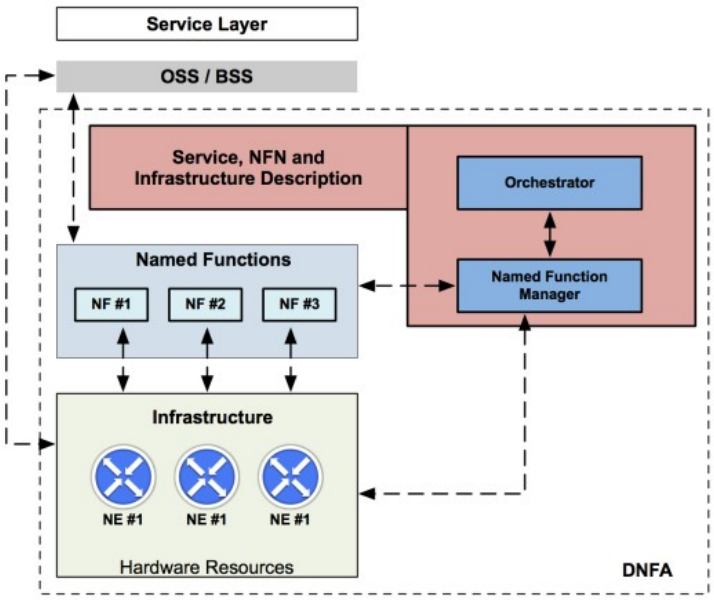


Figure 1. Functional Architecture

The middle left side of Fig. 1 contains the Named Functions, which use the resources provided by the NFN Infrastructure. The right side contains the management and orchestration elements. The Named Function Manager is responsible for the lifecycle management of named functions (placement, configuration, update, scale up/down, migration, termination, etc.). Finally, the Orchestrator is responsible for the management and implementation of the network services on the Network Infrastructure.

The top left corner are the Operation Support Systems (OSS) / Business Support Systems (BSS) of an operator as well as the Service, NFN and Infrastructure Description. The Service, NFN and Infrastructure Description provides information about a particular service (description, objective, limitations, etc.) and NFN information models. Finally, the Service Layer represents where services provided to the end user are modeled.

4. Experiments

In order to implement the reference architecture, CCN-lite ("CCN-Lite Project" n.d.) was chosen. CCN-lite is a reduced and lightweight (yet functionally interoperable) implementation of the CCN protocols. It supports named functions for letting clients express results instead of accessing only raw data. It admits Java, Scala or Python to host function execution and to interface to a NFN network.

All experiments were conducted on by network emulation experimentation through Mininet through a DPI VNF implementation (whose execution takes roughly 60 ms), considering the topology depicted in Fig. 2 with four network nodes. Relays D has the desired content and five relays A and B act as clients. Relays P is the only NFN-enabled node of the topology, and has been implemented both as a traditional IP-based DPI VNF and a NFN DPI function.

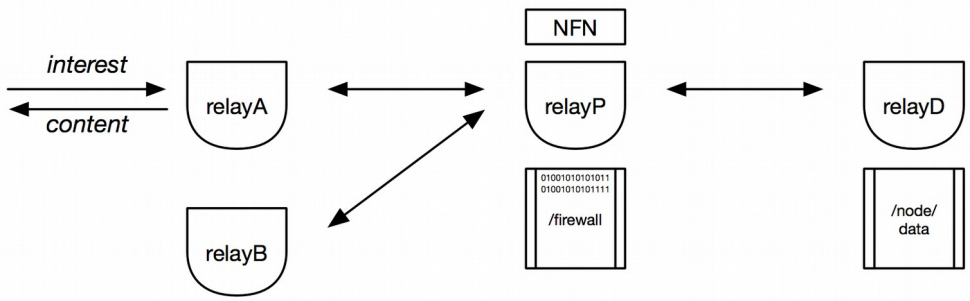


Figure 2. Hybrid CCN/NFN Infrastructure Topology

Network nodes have been assigned constant cache sizes, and a cache expiration time of 10 seconds. 20 users were spawned at a hatch rate of 2 user spawned per second, equally distributed between both relays RA and RB. Each users made consecutive and random requisitions from 100 ms to 200 ms. Content requests have been modeled as Poisson processes with $\lambda = 10$ to simulate content popularity.

Fig. 3 shows the requisitions per second (RPS) delivered by each implementation against the number of users. The CCN-based NFN implementation caches both most popular content and the computation result of the network function invocation managing to deliver more than 5 times requisitions per second than its IP-based alternative.

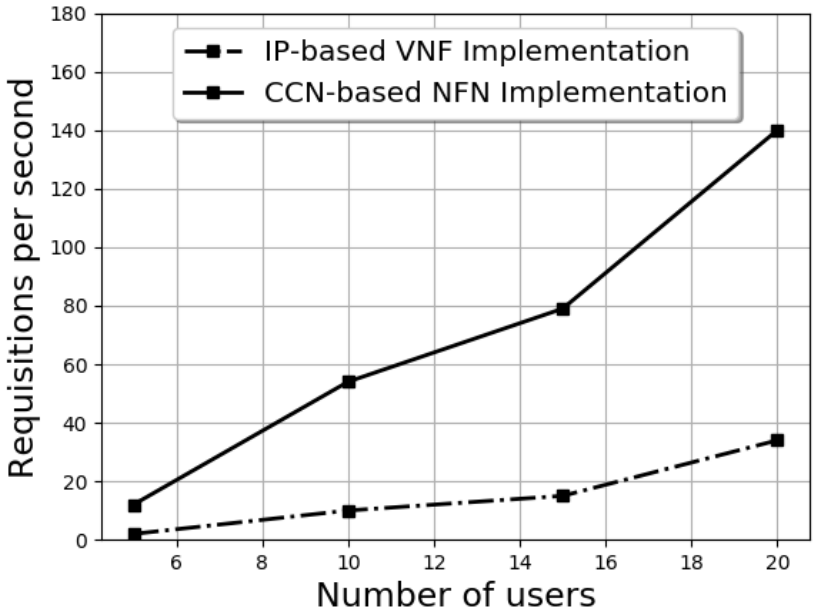


Figure 3. Requests per second

5. Conclusion

This work explored the use and implementation of VNFs in CCNs, and proposed the use of the NFN paradigm as means to implement network functions and services in this kind of networks, distributing the network functions and services through the networks nodes and providing flexibility to dynamically place functions in the network as required and without the need of a central controller. This brings many benefits, improving operational efficiency and reducing resource usage without the need of a central controller.

References

"A Survey of Enterprise Middlebox Deployments | EECS at UC Berkeley." n.d. Accessed May 12, 2016. https://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-24.html.

"Access Control Enforcement Delegation for Information-Centric Networking Architectures." n.d. Accessed December 31, 2018. https://dl.acm.org/citation.cfm?id=2342507.

Bandyopadhyay, Soma, Munmun Sengupta, Souvik Maiti, and Subhajit Dutta. 2011. "ROLE OF MIDDLEWARE FOR INTERNET OF THINGS: A STUDY." *International Journal of Computer Science and Engineering Survey* 2 (3): 94–105.

Bianco, A., R. Birke, L. Giraudo, and M. Palacin. 2010. "OpenFlow Switching: Data Plane Performance." In *2010 IEEE International Conference on Communications (ICC)*, 1–5. https://doi.org/10.1109/ICC.2010.5502016.

"CCN-Lite Project." n.d. Accessed May 2, 2016. http://www.ccn-lite.net.

Dannewitz, Christian, Dirk Kutscher, BöRje Ohlman, Stephen Farrell, Bengt Ahlgren, and Holger Karl. 2013. "Network of Information (NetInf) - An Information-Centric Networking Architecture." *Comput. Commun.* 36 (7): 721–735. https://doi.org/10.1016/j.comcom.2013.01.009.

"ETSI GS NFV-SWA 001 V1.1.1 (2014-12)." n.d. Accessed April 30, 2016. http://docplayer.net/3007035-Etsi-gs-nfv-swa-001-v1-1-1-2014-12.html.

Li, Y., and M. Chen. 2015. "Software-Defined Network Function Virtualization: A Survey." *IEEE Access* 3: 2542–53. https://doi.org/10.1109/ACCESS.2015.2499271.

Mansour, Dima, and Christian Tschudin. 2016. "Towards a Monitoring Protocol Over Information-Centric Networks." In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, 60–64. ACM-ICN '16. New York, NY, USA: ACM. https://doi.org/10.1145/2984356.2984378.

Marxer, Claudio, Christopher Scherb, and Christian Tschudin. 2016. "Access-Controlled In-Network Processing of Named Data." In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, 77–82. ACM-ICN '16. New York, NY, USA: ACM. https://doi.org/10.1145/2984356.2984366.

Mohaisen, Abedelaziz. 2017. Systems, devices, and methods for protecting access privacy of cached content. United States US9578125B2, filed May 5, 2014, and issued February 21, 2017. https://patents.google.com/patent/US9578125B2/en.

Tschudin, C., and M. Sifalakis. 2014. "Named Functions and Cached Computations." In *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th*, 851–57. https://doi.org/10.1109/CCNC.2014.6940518.

Evolução do testbed FIBRE

Gustavo Hermínio de Araújo[1], Marcos Felipe Schwarz[1], Luiz Eduardo Folly de Campos[1], Leandro Neumann Ciuffo[1], Michael Prieto Hernandez[1], Gustavo Neves Dias[1].

[1]Rede Nacional de Ensino e Pesquisa (RNP)
Rua Lauro Muller, 116 - Sala 1103 - 22.290-906 - Rio de Janeiro - RJ - Brasil

{gustavo.araujo, marcos.schwzarz, luiz.campos, leandro.ciuffo, michael.hernandez, gustavo.dias}@rnp.br

Abstract. *FIBRE (Future Internet Brazilian environment for Experimentation) is a research-driven testbed constructed in the scope of a project funded by the 2010 Brazil-EU Coordinated Call in ICT. The project ended in October 2014. However, in 2015 the Brazilian member institutions, led by the National Research and Education Network (RNP), took over FIBRE's legacy infrastructure to offer the testbed as a service. Currently, funds the secure the FIBRE testbed operational are guaranteed until the end of the year. This paper presents new features and improvements deployed in the last year, as well as strategies to decrease operating costs while new funding opportunities are prospected.*

Resumo. *FIBRE (Future Internet Brazilian environment for Experimentation) é um ambiente para experimentação (testbed) construído no âmbito de um projeto financiado pela Chamada Coordenada Brasil-UE em TIC de 2010. O projeto terminou em outubro de 2014. No entanto, em 2015, as instituições brasileiras membros do projeto, lideradas pela Rede Nacional de Ensino e Pesquisa (RNP), assumiram a infraestrutura legado do testbed FIBRE para oferecer o testbed como um serviço. Atualmente, recursos estão garantidos para a operação do testbed até o final do ano. Este artigo apresenta as novas funcionalidades e melhorias implementadas no ultimo ano, bem como estratégias para reduzir os custos operacionais enquanto novas oportunidades de financiamento são prospectadas.*

1. Introdução

A avaliação de novas arquiteturas, protocolos, métodos e tecnologias em pesquisas aplicadas em Internet do Futuro dependem de recursos dedicados para experimentação a fim de buscar soluções em áreas como desempenho, medições, segurança, mobilidade, distribuição de conteúdo etc. Diversas iniciativas e projetos financiados por governos e por redes de pesquisa e educação em diferentes países tem sido desenvolvidas desde 2003 [1], implantando infraestruturas dedicadas para dar suporte a pesquisas em Internet do Futuro. O projeto FIBRE (*Future Internet testbeds/experimentation between Brazil and Europe*) [2], selecionado em 2010 na 1ª Chamada Coordenada entre Brasil e Europa em tecnologias de informação e comunicação (TIC), construiu e opera um testbed de larga escala, que atualmente conta com a participação de 15 instituições distribuídas em

9 estados, mais o Distrito Federal. Atualmente, sendo a maior plataforma para pesquisas aplicadas em Internet do Futuro do Brasil.

2. Arquitetura do Framework de controle FIBRE

Os componentes da arquitetura são estruturados em dois níveis: (i) nível locais e (ii) nível global. O nível local é representado pelas ilhas de experimentação. Cada ilha possui controle direto de seus recursos, componentes de gerenciamento e interage com os componentes do nível local através de uma API SFA *Aggregate Manager* (SFA AM). O nível global é responsável por interagir com o usuário através do Portal de Experimentação, gerenciar as credenciais de usuário através da *ClearingHouse* e fornecer as interfaces para realizar experimentos através do OMF6. O Portal de experimentação foi desenvolvido especificamente para atender aos requisitos do testbed FIBRE, enquanto, a *ClearingHouse* foi implementada de acordo com as especificações GENI's. Na Figura 1 é possível observar os níveis da arquitetura do *framework* de controle do FIBRE e seus principais componentes.

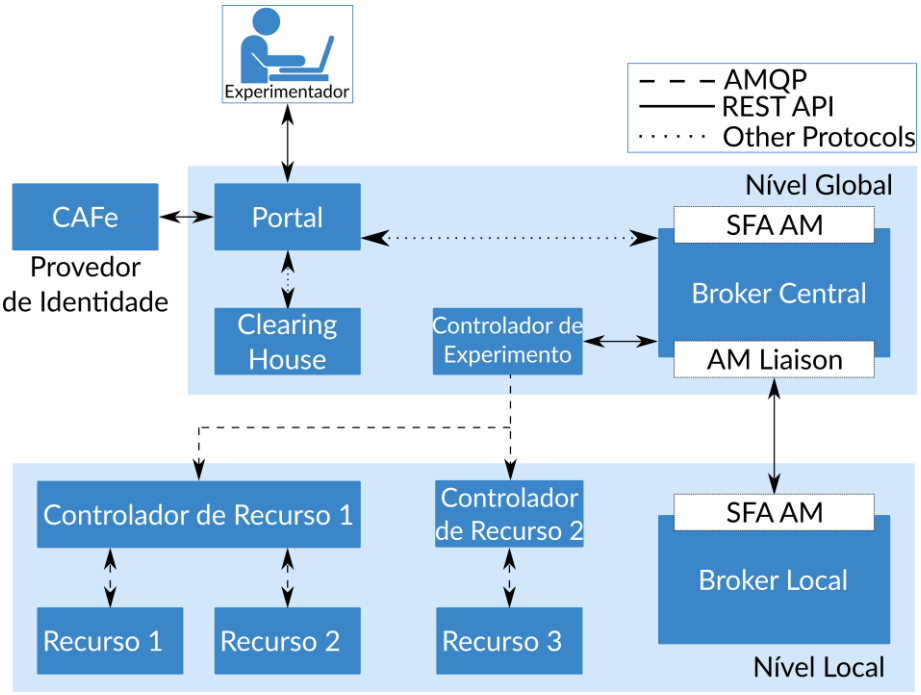


Figura 1 - Desenho da Arquitetura do Framework de controle do FIBRE.

O *Central Broker* é a entidade responsável por agregar e anunciar recursos disponíveis no testbed, realizar reserva de recursos e por prover os recursos reservados no devido tempo. Para isso, o Central Broker coleta as informações dos recursos nos *Brokers* Locais através de uma API SFA chama AM *Liaison*. Os *Brokers* locais expõem uma API do SFA AM que permite descobrir, agregar e anunciar recursos locais para o intermediário central. Cada ilha deve implantar seu Broker Local para agendar e criar reservas de seus recursos no Portal de Experimentação.

O Controlador de Experimento é a entidade de controle responsável por orquestrar experimentos descritos por scripts escritos em OEDL (*OMF Experiment Description Language*) [3]. Um sistema de mensagens de publicação e inscrição é

adotado para lidar com comunicações entre recursos e as entidades que interagem com eles. Os participantes podem criar tópicos, assiná-los e publicar mensagens para eles usando o AMQP 5, um protocolo de camada de aplicação padrão aberto para middlewares orientados a mensagens.

Controladores de recursos intermediam a troca de mensagens entre os recursos locais e da *Experiment Controller*. Eles são responsáveis por controlar o ciclo de vida dos recursos sob sua governança. Eles criam instâncias de recursos e enviam mensagens de controle arbitrárias para eles. Esses recursos locais podem representar máquinas virtuais, nós sem fio, sensores especializados ou recursos OpenFlow.

2.1. Portal de Experimentação

O Portal de Experimentação[1] é uma interface web criada especificamente para o testbed da FIBRE, que permite aos usuários alocar e interagir com os recursos de cada ilha disponível por meio de um navegador. Experimentadores podem criar e gerenciar projetos compartilhados, construir experimentos e adicione recursos suportados a eles através do portal. O portal de experimentação simplifica o uso da federação porque meio das interações necessárias entre CH, Broker e experimentadores. Na Figura 2 é possível visualizar a interface para a edição de topologias do portal de experimentação.

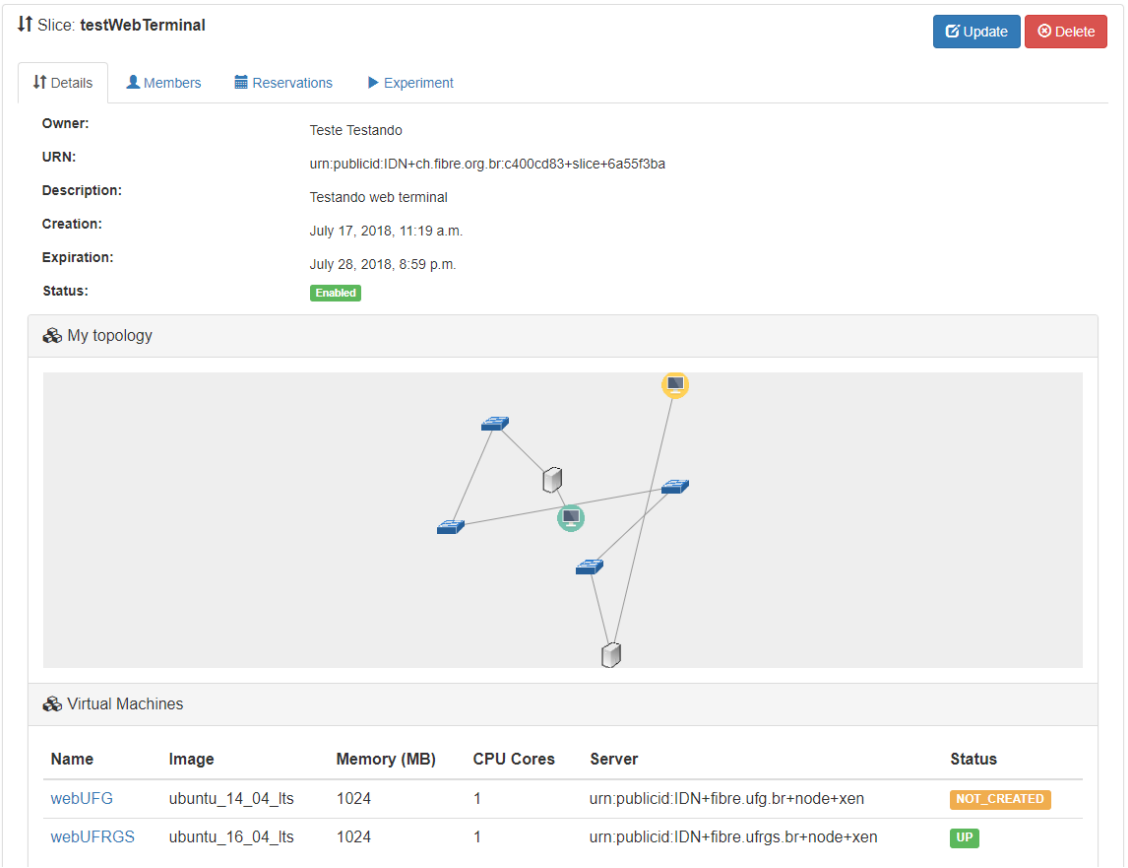


Figura 2 - Tela da Interface para Edição de Topologias.

[1] Acessível através da URL: https://portal.fibre.org.br

Para utilizar serviços, os experimentadores devem ser autenticados. Dada a diversidade de usuários que pretendemos alcançar com a FIBRE, os experimentadores podem usar a autenticação concedida pela federação brasileira de identidade acadêmica chamada Comunidade Acadêmica Federada (CAFe). No entanto, o portal também suporta autenticação local para pesquisadores que não estão contemplados com uma conta federada. O Portal de Experimentação interage com o OMF6 Central Broker para a descoberta e a reserva de recursos por meio de uma API REST exposta pelo Central Broker.

3. Evolução do Backbone FIBREnet

Os ambientes de pesquisa do testbed FIBRE, conhecidos como ilhas de experimentação, são interconectados por uma infraestrutura SDN denominada FIBREnet, composta de switches SDN OpenFlow e enlaces de rede em camada 2 sobrepostos ao backbone da RNP (Rede Ipê). A FIBREnet representa a maneira ideal de conectar as ilhas, através de links dedicados. Ela permite a execução de experimentos de forma federada, utilizando recursos de duas ou mais ilhas no mesmo experimento, e oferece a programabilidade de seus switches OpenFlow, interconectados em malha parcial, aos próprios experimentadores, permitindo que possam personalizar a arquitetura de rede que pretendem utilizar entre as ilhas participantes do experimento.

Ao longo da evolução do projeto, de 2011 a 2018, a FIBREnet implantou 10 switches, instalados em Pontos de Presença da RNP de diferentes estados, e 15 enlaces de 1Gbps entre eles, seguindo a arquitetura original do projeto FIBRE para a FIBREnet. Devido a algumas limitações, como a falta de suporte a novas versões do protocolo OpenFlow (1.3), o limite da taxa máxima de vazão em 1Gbps, e à solução específica de virtualização utilizada originalmente, foi elaborado um novo conjunto de soluções para que a FIBREnet pudesse atender aos requisitos de um testbed de experimentação com suporte às novas tecnologias e protocolos recentes e futuros. A nova solução, chamada de SDN Overlay, compreendeu o uso de equipamentos com arquitetura de hardware genérica e aberta, conhecidos como whiteboxes, o uso de software de código aberto, como o sistema operacional Linux, o switch virtual Open vSwitch [5] e o driver de aceleração de processamento de pacotes DPDK [4], e o desenvolvimento de um gerenciador de redes virtuais para administração e operação das redes virtuais que seriam ativadas sobre essa infraestrutura, chamado SOLO. Assim, a nova arquitetura da FIBREnet seria provida por switches SDN Open vSwitch de versão recente, com suporte a OpenFlow nas versões de 1.0 a 1.5, e herdando todas as demais funcionalidades providas pela pilha de software do Open vSwitch.

A arquitetura da SDN Overlay foi viabilizada como parte do projeto IDS (Infraestrutura Definida por Software), e o testbed FIBRE se tornou então seu primeiro e principal utilizador, após a migração da arquitetura original FIBREnet para esse novo modelo. Inicialmente, os whiteboxes foram implantados nos mesmos PoPs onde já estavam hospedados os switches FIBREnet. Em seguida, eles foram adicionados à rede do FIBRE com portas físicas dedicadas, porém servindo apenas como trânsito de tráfego entre o backbone da RNP e os switches originais. Então foram desenvolvidos os switches virtuais Open vSwitch nos whiteboxes, replicando de forma fiel as características dos switches físicos da rede original, como DPIDs (identificador de cada switch OpenFlow), condições de interfaces e seus respectivos circuitos, com o intuito de

reconstruir uma topologia idêntica. Como último passo, a manipulação SDN foi deslocada dos switches originais para os novos switches, retirando a parte OpenFlow dos switches originais nos PoPs e conectando suas réplicas virtuais nos whiteboxes às interfaces físicas destinadas ao FIBRE. A migração foi feita de gradualmente, de forma que ao início de 2019 todos os switches SDN FIBREnet já eram providos pela nova arquitetura Open vSwitch. Muitos dos switches originais ainda continuam na topologia do testbed, servindo como trânsito para o tráfego que usa links ópticos para chegar até as ilhas. Por fim, os experimentadores contam agora com a mesmas funcionalidades SDN providas pela arquitetura anterior, porém com a vantagem de uma nova infraestrutura que poderá ser utilizada para a evolução do testbed como um todo, incluindo novas versões de protocolos como o OpenFlow 1.3.

A Figura 3 ilustra a nova arquitetura da FIBREnet. A camada de abaixo representa os 10 equipamentos Whitebox físicos, implantados nos mesmos PoPs da FIBREnet. A segunda camada, com enlaces em vermelho, se refere à rede virtualizada replicando a FIBREnet, criada para garantir o funcionamento transparente dos serviços oferecidos pelo FIBRE atualmente e permitir a migração gradual para esse novo modelo. A terceira camada, com 3 enlaces em verde, representa uma outra rede virtual de experimentação, instanciada por meio do novo sistema de controle e gerência do FIBRE (OMF 6).

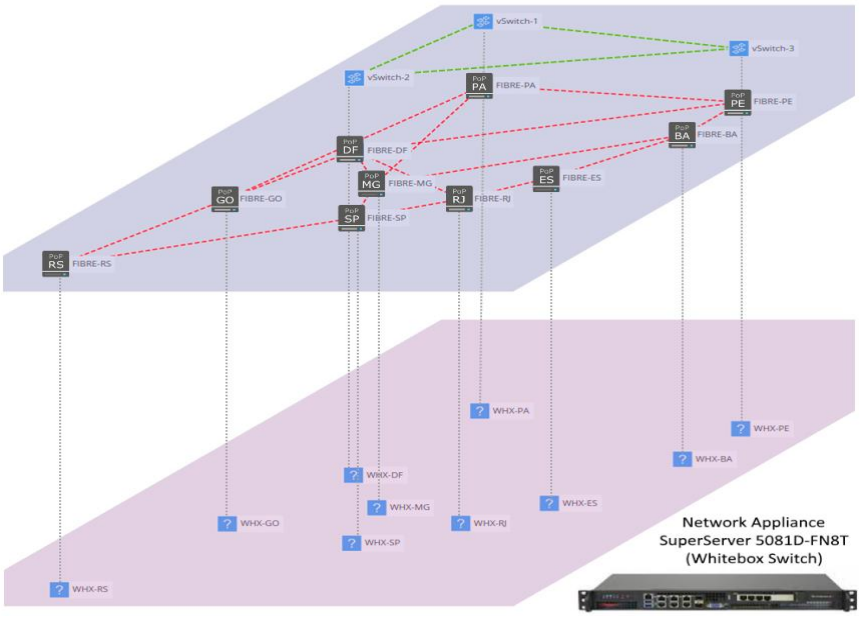


Figura 3 - Arquitetura da FIBREnet e os switches whitebox físicos utilizados na implantação.

4. Trabalhos Futuros

O Testbed FIBRE permaneceu em contínua evolução ao longo do último ano, tendo como iniciativas em andamento as apresentadas a seguir.

4.1. Automação Integrada da Operação e Desenvolvimento

Considerando a atual infraestrutura do FIBRE, que consiste em mais de 100 equipamentos em diferentes categorias de propósitos (switches OpenFlow, switches ToR, servidores de virtualização, servidores NetFPGA, servidores perfSONAR e múltiplas máquinas virtuais) distribuídos em 17 ilhas, muito do trabalho de operação está direcionado para a implantação de novas ilhas, documentação e capacitação de novos operadores, manutenções e atualizações de infraestrutura.

Dado este contexto, a equipe de operação do FIBRE tem focado em adotar boas práticas de automação para melhorar a agilidade e a replicabilidade destas atividades. Após o estudo de diversas ferramentas, foram definidos quatro estágios de automação: I) máquinas físicas (baremetal); II) configuração; III) conteinerização e IV) integração e entrega contínua (CI/CD).

O primeiro estágio consiste em habilitar a operação remota e automatizada de todos os ativos físicos, incluindo acesso ao terminal das máquinas, controle do fornecimento de energia e reinicializações, e a automação do provisionamento do sistema operacional da máquina. Para alcançar tal objetivo, foi escolhida a solução aberta MAAS da Canonical dado sua facilidade de uso e conjunto de funcionalidades. O MAAS foi validado para controlar todos os servidores e switches whitebox do FIBRE, ele será implantado em cada PoP da RNP e irá remotamente instanciar novas ilhas ou atualizar as ilhas existentes.

O segundo estágio é a gerência de configuração e consiste em preparar o Sistema Operacional para atender a todo o conjunto de ferramentas e requisitos definidos pelo FIBRE. Nesta etapa, a ferramenta selecionada foi o Ansible, dado a sua rápida curva de aprendizado e por suportar ambos equipamentos de rede e servidores. Após o servidor ser instalado na etapa anterior, o Ansible executa um conjunto pré-definido de receitas (playbooks) que são responsáveis pela instalação das aplicações e configurações de cada equipamento como instalação do docker, configuração de rede e tunings, deixando o ambiente pronto para ser usado. As receitas são criadas de forma a separar os parâmetros mínimos de cada ambiente como IPs, sistema operacional e credenciais e construir um conjunto de passos bem definidos que serão aplicados e reaproveitados em toda a infraestrutura.

O terceiro estágio consiste em substituir as máquinas virtuais de serviço do FIBRE por containers, de forma a reduzir o consumo de recursos, reduzir a quantidade de elementos gerenciáveis e facilitar a utilização de conceitos como infraestrutura imutável, que simplifica a previsibilidade e agilidade dos processos de mudança. Inicialmente será utilizado Docker como solução para executar containers, mas já está em estudo a adoção do Kubernetes, que apesar da sua difícil curva de aprendizado, possibilita uma orquestração de containers mais completa e novas funcionalidades como escalabilidade horizontal e auto regeneração (self-healing). Serão utilizados containers pré criados e testados por terceiros ou pelo próprio FIBRE usando o quarto estágio.

O quarto estágio é a integração e entrega contínua (CI/CD), que consiste em integrar o código desenvolvido ao projeto principal na mesma frequência com que as funcionalidades são desenvolvidas, sendo feito muitas vezes ao dia ao invés de apenas uma vez. O objetivo principal de utilizar a integração contínua é verificar se as

alterações ou novas funcionalidades não criaram novos defeitos no framework de controle FIBRE. Para utilizar o processo de integração contínua no contexto do testbed FIBRE, utilizamos as seguintes ferramentas:

- GitLab: realiza o controle de versão do framework de controle do testbed FIBRE.
- Jenkins: realiza a orquestração e automatização do processo de desenvolvimento de software e integração contínua.
- Nexus: utilizado como gerenciador de repositórios onde são armazenados os artefatos (e.g., instaladores, scripts de configuração) do framework de controle.
- SonnarQube: verifica a qualidade do código que foi enviado ao GitLab.
- Ansible: realiza o provisionamento de software, gerenciamento de configuração e implementação de software.

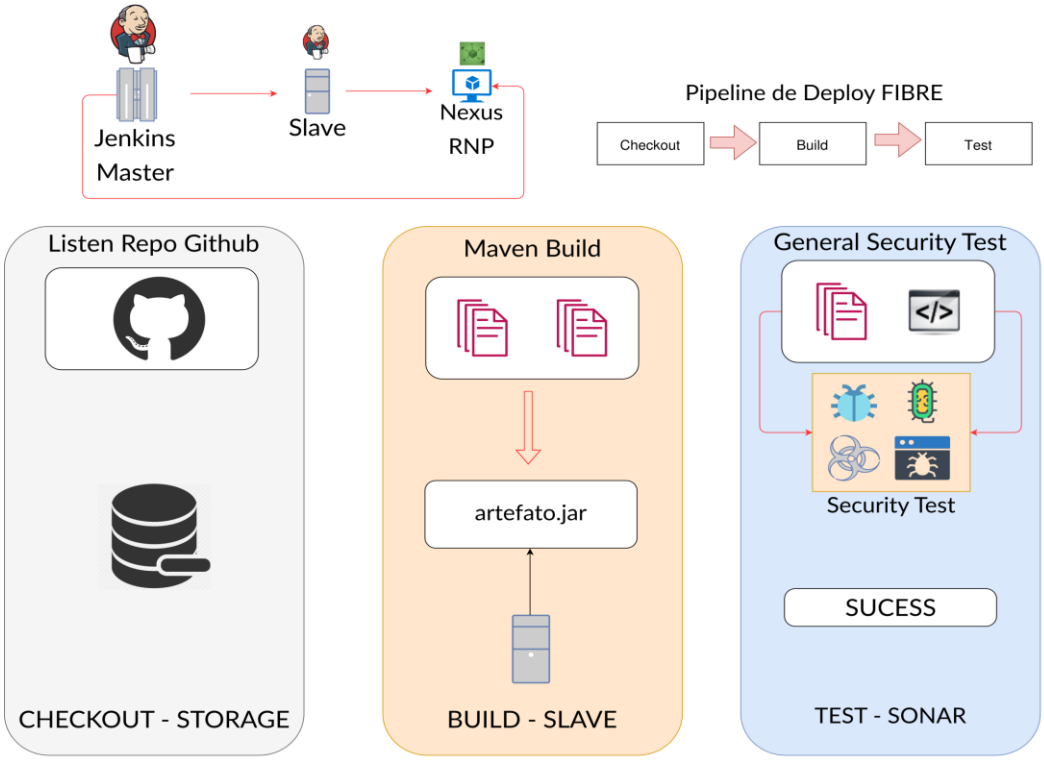


Figura 4 - Fluxo proposto para integração e entrega contínua

O fluxo completo e as ferramentas de integração contínua são exemplificadas na Figura 4. Primeiramente o desenvolvedor realiza um commit do novo código na ferramenta GitLab. Neste estágio são identificadas as modificações realizadas no código e é feito o seu armazenamento. Em seguida é iniciado o estágio de build onde o código é compilado e/ou construído e então um artefato é gerado e armazenado no gerenciador de repositório NEXUS. Em seguida, o código é submetido ao estágio de teste, onde o SonarQube realiza testes genéticos para verificar a integridade do código. Esses testes vão desde ataques de SQL Injection até DDoS e Brute Force. Ou seja, são testes

genéricos, que podem ser executados em qualquer aplicação, uma vez que têm por objetivo encontrar problemas mais comuns e básicos relacionados à segurança da informação. Por fim, os artefatos são copiados e instalados em seus respectivos alvos (e.g., as ilhas de experimentação) através da ferramenta Ansible.

4.2. FIBRE in a box/FIBRE in a laptop

Os esforços de automação da infraestrutura, além das vantagens apresentadas na seção anterior, possibilitam oferecer uma versão do FIBRE (i.e., FIBRE-In-a-Box) com baixo consumo de recursos, orientada a experimentação em um contexto local (e.g., laptop), abstraindo ao pesquisador aspectos como autenticação federada e o uso de um ambiente compartilhado.

Do ponto de vista da arquitetura o FIBRE-In-a-Box adiciona aos componentes do plano de controle do FIBRE, uma camada de rede emulada e nós de computação, encapsulados em containers. Desta forma a solução como um todo permanece autocontida e isolada sem possuir dependências de bibliotecas ou configurações externas.

FIBRE-In-a-Box permitirá aos experimentadores o planejamento, desenvolvimento e a execução de testes das aplicações de forma ágil, em ambiente local, sem as restrições do tempo alocado para o experimento ou os recursos disponíveis no ambiente de experimentação federado.

Uma vez que o experimentador validar o correto funcionamento da aplicação e precisar de um ambiente mais robusto com comportamento determinístico, no qual o experimento possa ser reprodutível e com resultados de valor científico, FIBRE-In-a-Box possibilitará a implantação do experimento no FIBRE sem intervenção do usuário.

Com a disponibilização do FIBRE-In-a-Box para o uso da comunidade científica pretende-se aproximar os pesquisadores do paradigma das redes definidas por software através de plataformas de baixo consumo de recursos, flexíveis e de fácil instalação que permitem o desenvolvimento de PoCs ao mesmo tempo que se integram com ambientes de experimentação de alta confiabilidade.

4.3. Expansão e experimentação de tecnologias associadas à Industria 4.0

Duas novas ilhas de experimentação deverão ser integradas em até 1 ano e estarão hospedadas no Laboratório de Engenharia de Sistemas (LABESys), localizado na UFSC e no Laboratório de Robótica (LAR), localizado na UFBA. Essas duas ilhas disponibilizarão um conjunto de sensores e atuadores que poderão ser programados e parametrizados pelos usuários do testbed.

Essa expansão será viabilizada por meio do projeto IND4FIBRE, coordenado pelo INESC P&D Brasil e financiado pela Agência Brasileira de Desenvolvimento Industrial (ABDI). O projeto IND4FIBRE objetiva fomentar a adoção de tecnologia associada à Industria 4.0 através da instalação de dois ambientes para experimentação, no INESC P&D Brasil e seus conveniados da Rede INESC BRASIL, as universidades

UGF e UFBA, composta de sensores e atuadores que poderão ser programados e parametrizados pelos usuários do testbed, implantação de um repositório de dados para publicação e armazenamento dos dados coletados pelos sensores, adaptação do software FlexSim para consumir os dados publicados no repositório. A Figura 5 ilustra os componentes envolvidos na solução.

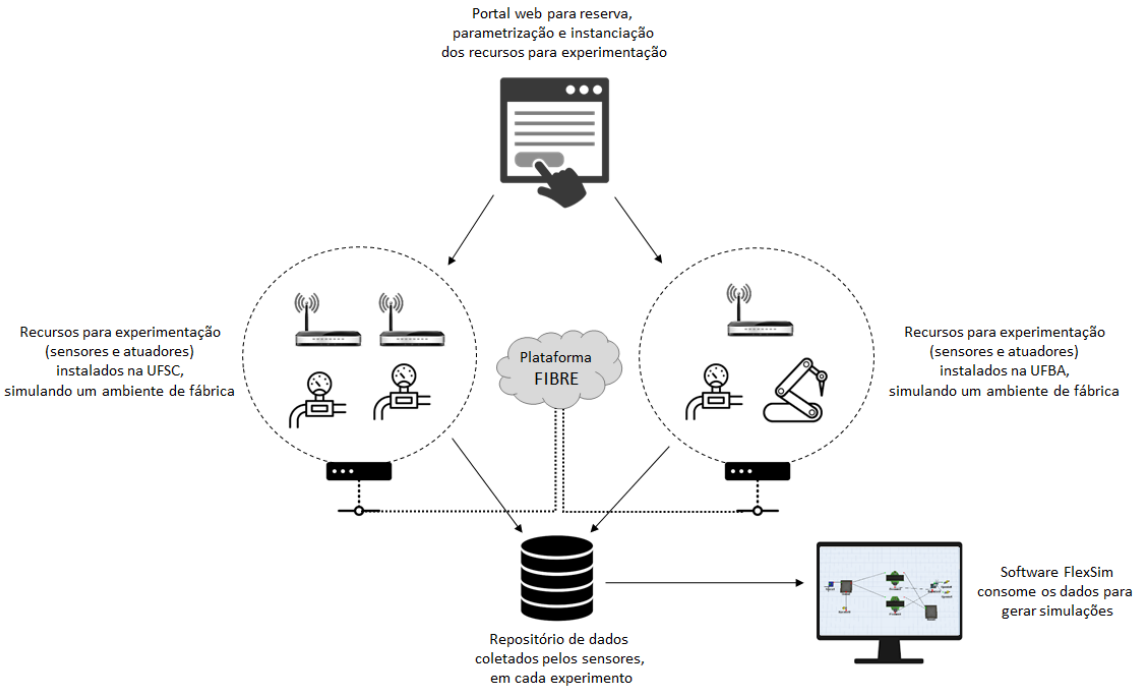


Figura 5 - Diagrama com os componentes da plataforma de experimentação.

5. Conclusão

Este artigo apresentou as evoluções implementadas no testbed FIBRE no último ano, desde a apresentação realizada no WPEIF 2018[2]. De forma a contribuir para a sustentabilidade do FIBRE, algumas ações estão sendo empreendidas, conforme relatadas no artigo: (i) Automação integrada da operação e desenvolvimento; (ii) migração da arquitetura original da rede FIBREnet para nova solução SDN Overlay; (iii) redução da barreira de entrada de novos experimentadores através da abordagem "FIBRE in a box" e (iv) expansão dos recursos programáveis disponíveis para experimentação através do apoio à submissão de projetos de pesquisa que façam uso da infraestrutura do FIBRE, como no caso do projeto IND4FIBRE.

Outra ação em andamento é a integração da base de usuários do FIBRE ao do novo testbed CloudNEXT[3], que permitirá experimentação de novas arquiteturas, plataformas e aplicações em computação em nuvem, com acesso direto ao hardware da

[2] https://www.fibre.org.br/o-fibre-no-sbrc-2018/

[3] http://cloudlab-brasil.rnp.br/testbed

infraestrutura. A proposta é que usuários do FIBRE não necessitem criar nova conta para acessar a infraestrutura do ClodNEXT. Esse novo testbed é fruto de um projeto financiado pela FINEP que está próximo de sua conclusão.

A integração do FIBRE com outros ambientes para experimentação, tais como as infraestruturas legado dos projetos NECOS, 5GINFIRE e FUTEBOL, já começaram a ser mapeadas e foram materializadas na forma de um projeto de pesquisa submetido ao edital "Chamada de Propostas FAPESP / MCTIC – 2018", ainda aguardando resultado.

Por fim, novas ações de promoção e incentivo ao uso do testbed FIBRE precisarão ser empreendidas e focadas não apenas para a comunidade acadêmica, mas também para micro e pequenas empresas do setor de TICs.

6. Referências

[1] PAN, J., PAUL, S. e JAIN, R. A survey of the research on future internet architectures. IEEE Communications Magazine, vol. 49, no. 7, págs. 26 a 36, julho de 2011. DOI: 10.1109/MCOM.2011.5936152. Acesso em: 25/03/2018. Disponível em: <http://ieeexplore.ieee.org/document/5936152/>

[2] SALMITO, T., CIUFFO, L., MACHADO I., SALVADOR S., STANTON, M. et al. FIBRE - An International Testbed for Future Internet Experimentation. Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC 2014, maio 2014, Florianópolis, Brazil. p. 969, 2014. Acesso em: 25/08/2018. Disponível em: <https://hal.archives-ouvertes.fr/hal-00990392>

[3] T. Rakotoarivelo, M. Ott, G. Jourjon, and I. Seskar. Omf: A control and management framework for networking testbeds. SIGOPS Oper. Syst. Rev., 43(4):54-59, Jan. 2010.

[4] INTEL, D. P. D. K. Data Plane Development Kit. 2014.

[5] PFAFF, Ben et al. Extending Networking into the Virtualization Layer. In: Hotnets. 2009.

FUTEBOL - Orchestrated Optical/Wireless Experiments Across Brazil and Europe

Daniel F. Macedo[1], Luis H. Cantelli Reis[1], José Marcos S. Nogueira[1], Cristiano B. Both[2], Luiz da Silva[3], Juliano Wickboldt[4], Johann Marquez-Barja[5]

[1]Universidade Federal de Minas Gerais – Brazil

[2]Unisinos – Brazil

[3]Universidade Federal do Rio Grande do Sul – Brazil

[4]Trinity College Dublin – Ireland

[5]University of Antwerpen – Belgium

{damacedo,luiscantelli,jmarcos}@dcc.ufmg.br, cbboth@unisinos.br,

dasilval@tcd.ie, jwickboldt@inf.ufrgs.br, Johann.Marquez-Barja@uantwerpen.be

Abstract. *This paper presents the experimentation facilities provided by the European Union-Brazil project "FUTEBOL - Federated Union of Telecommunications Research Facilities for an EU-Brazil Open Laboratory", from the point of view of the experimenter. The FUTEBOL testbeds to provide experimentation capabilities for optical-wireless experimentation, which is crucial for the development of 5G technologies. The main offering of FUTEBOL is, thus, to be able to set up an experiment accross many network domains – be it optical, wireless, packet or cloud –, and to orchestrate them using tools such as COPA and ABNO. Further, FUTEBOL provides a set of services that can be instantiated with very little effort, in order to compose complex 5G experiments in reduced time.*

1. Introduction

Telecommunications networks remain largely segregated in the two domains of optical networks and wireless systems, and rarely do even researchers cross the boundary between the two. We argue that the evolutionary path [Raaf et al. 2014] and the needs of future telecommunication systems, be it for high data rate applications, for the Internet of Things (IoT) paradigm, or for aggressive backhaul requirements stemming out of cell densification, require the co-design of wireless access and optical backhaul and backbone. In this context, the EU-Brazil co-funded project FUTEBOL (Federated Union of Telecommunications Research Facilities for an EU-Brazil Open Laboratory) [FUTEBOL 2019] developed a converged control framework for experimentation on wireless and optical networks and to deploy this framework in federated research facilities, which will be made openly available to the whole research community.

The proliferation of small cells deployments increases frequency reuse and is one of the main means that allow for gains in mobile network capacity. On the optical network side, network function virtualization (NFV) and software-defined

networks (SDN) are revolutionizing the way that network resources are managed. We view virtualization on the optical side and densification and capacity increase on the wireless access as major game changers in future networks that will deliver the best benefits when co-designed and experimented together. Therefore, this work describes a research infrastructure tailored to the needs of experimenters throughout the world, interested in issues that cross the boundary between wireless and optical networks. This infrastructure allows many types of experiments, as explained in more details in [Marques et al. 2017].

The focus of this paper is to present the FUTEBOL testbeds from the point of view of the user. In a nutshell, we present what are the capabilities of each testbed and what are the types of experiments that we envision in each of them. Then, we describe the FUTEBOL service catalogue and the orchestration tools, which are used by the experiment to compose and then control a multi-domain experiment. Finally, we describe a sample experiment using a graphic orchestrator called COPA.

The remainder of the paper is organized as follows. Section 2 presents each of the FUTEBOL testbeds as well as their offering. Section 3 describes the control framework, focusing on the aspects that are more relevant to the end-user. Section 4 shows a simple experiment. Section 5 concludes the paper.

2. The FUTEBOL testbeds

The FUTEBOL testbeds are heterogeneous testbeds, in which each institution provides a set of different resources. Because of that characteristic, FUTEBOL provides different types of equipment for experimentation in different 5G use cases. We describe below the main offerings of each testbed, as well as a glimpse of what type of experiments we envision to be run on each of them.

Iris TCD - the reconfigurable radio testbed at Trinity College Dublin (TCD) provides virtualized radio hardware, Cloud-RAN, Network Functions Virtualisation (NFV), and Software Defined Networking (SDN) technologies to support the experimental investigation of the interplay between legacy technologies, 5G radio and future networks. We offer the radio resources including 20 USRP N210s and 2 USRP X310s ceiling mounted nodes all equipped with SBX daughterboards, reaching frequencies between 40 MHz and 4 GHz. These platforms are connected to a computational cloud, allowing us to deploy an array of computational environments, accessible through Fed4FIRE. To expose the functionality of these platforms for a variety of applications, we employ a variety of radio hypervisors that freely enable prototyping of wireless systems, as exemplified by GNURadio, srsLTE (an open-source 3GPP 4G stack from the Irish company Software Radio Systems), etc.

UNIVBRIS - The main focus at UNIVBRIS is to provide optical-packet orchestration, through a newer optical solution on an SDN-enabled network, and a virtualized infrastructure overlayed in an optical-packet path, which enables the experimental investigation using cross-layer algorithms that could be involved in 5G network research. Furthermore, experimenters can define physical parameters in an optical resource (BV-ROADM) and create an end-to-end (E2E) path using optical-packet mapped by a tuple of ports and wavelengths. We deploy our infrastructure using OpenStack, ABNO and proprietary optical solutions from UNIVBRIS.

UFES - The UFES testbed in light of the new generation of robotics as a service (e.g, robot localization and navigation, assistive robotics), has been built to support the requirements of real-time remote control of robots over a wireless-optical SDN-enabled network. It explores the field of cloud robotics control with the aid of computer vision that requires novel orchestration features offered by the FUTEBOL Control Framework (CF) such as i) flexible innovative vertical auto-scaling providing elasticity of Virtual Network Functions (VNF) using Openstack ii) orchestrating wireless, packet and optical networks to meet simultaneously low E2E latency and high bandwidth requirements, aiming to study and overcome the bottlenecks in the cloud-fiber-wireless domain, involving cyber-physical systems use cases with stringent requirements. Our target are production processes environments that need to test wireless control of industrial manufacturing, where URLLC communication services are expected, especially in the context of the Industry 4.0 (also known as Industrial IoT) vertical.

UFMG - UFMG's focus is on wireless experimentation, and how wireless technologies interact with cloud, packet and optical networks. As such, the UFMG testbed provides resources for experimentation of the so-called last mile technologies, that is, the technologies that connect the end-users to the core networks, using standards for both licensed (4G) and unlicensed (WiFi, ZigBee, Bluetooth) bands. The resources of the testbed allow the experimenters to program the network on layers 1 and 2 of the wireless protocol stack using USRP, while the implemented SDN protocols and NFV functionality provide programmability at layers 2 and above. Further, the testbed provides resources for IoT experimentation in an indoor environment.

UFRGS - The testbed is focused on devices for wireless research, Internet of things, sensor networks and optical-wireless convergence. With wireless experimentation (WiFi, 3G, 4G, 5G) including new protocols and technologies over programmable radio platforms (i.e., SDR), the testbed also provides optical-wireless convergence experimentation with its radio-over-fiber arrangement, where it's possible to create networks (up to the layer 2) using USRP through optical infrastructure.

VTT - The VTT testbed provides means for research and development by experimentation of wireless communication. The testbed may be enhanced by adding new hardware, for example targeted at 5G research. The testbed focus area is mainly shared spectrum access, LSA (Licensed Shared Access) being the current implemented and tested software, and SAS (Spectrum Access System) as a possible future improvement.

3. The FUTEBOL Control Framework

The deployment and control of experimentation slices, as well as the control of the experiment itself, is performed using tools that are available in the FUTEBOL control framework. The FUTEBOL control framework is divided into layers, as shown in Figure 1. The lowest layers are the common layers found in any testbed, namely the infrastructure layer and the virtualization layer. The infrastructure layer is composed of the physical links, compute resources as well as network equipment that is available for the experimenters. The virtualization layer provides the means to slice the physical infrastructure, supporting arbitrary topologies and multiple experiments.

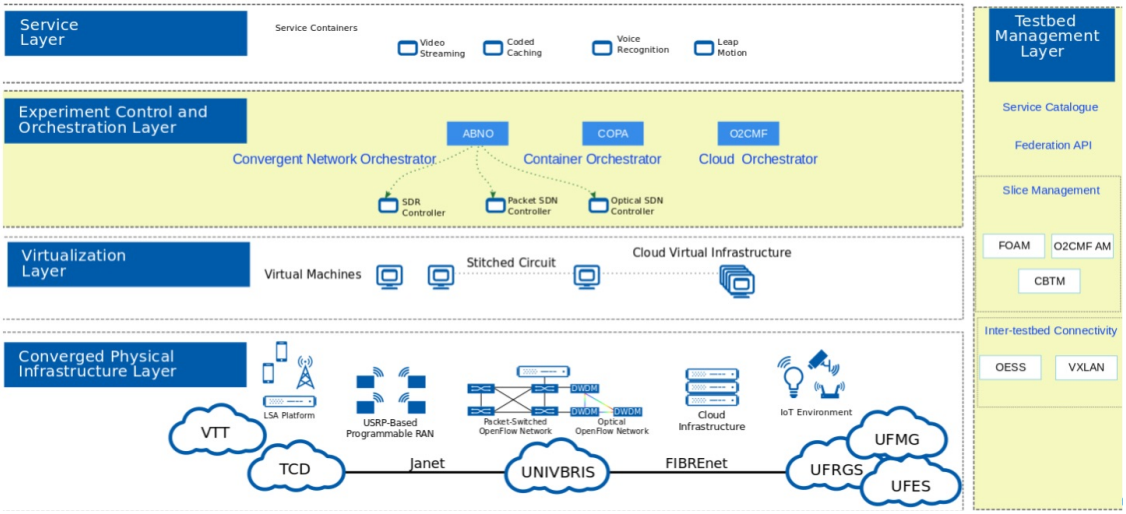


Figure 1. Structure of the FUTEBOL control framework

In this section we will describe two layers of the FUTEBOL CF, namely the testbed management layer and the orchestration layer.

3.1. Testbed management layer

The testbed management layer controls the allocation of the testbed resources. Further, it also offers a catalogue of services that provides virtual network functions (VNFs), services, and tools that facilitate converged experimentation. Each testbed will provide a different set of services in their catalogue, according to their key offering. Offering pre-installed services is very important for experimentation, since more sophisticated experiments may demand a long installation and setup time. For example, one may require to download files from github, compile them, and then configure the service.

The selection of the service that will be instantiated is based on the image that will be allocated on the virtual resource. For instance, a user may select a certain image that comes pre-loaded with an SDN controller, a video server, or even elements of the 5G core. In order to simplify the deployment of those serivces, each testbed provides a user guide that describes a basic experiment. The services come pre-loaded with configurations to realise those sample experiments.

3.2. Orchestration layer

The Experiment Control and Orchestration Layer allows the integrated manipulation of optical and wireless resources based on three tools. ABNO supports convergent network orchestration by interacting with the network controllers in the wireless and optical domains. COPA is an orchestration tool for containers developed to enable the live migration and monitoring of VNF across remote testbeds. O2CMF provides Cloud and NFV orchestration capabilities, such as automatic scaling of VNF based on pre-defined policies and monitoring of resource usage. Each tool is briefly described in the following paragraphs.

The ABNO architecture [Hammad et al. 2017] is based on IETF RFC7941 and enables network automation and programmability. The main motivation for

developing ABNO as the network orchestrator in our CF is because most existing SDN controllers rely on monolithic software, offering insufficient flexibility for heterogeneous networks. This inflexibility has led to problems for network operators, such as vendor lock-in, lack of support for non-OpenFlow networks, and lack of support for end-to-end multi-domain networks.

COPA[1] is a new orchestration and provisioning tool for containers, which we created to facilitate the migration and monitoring of virtual functions across testbeds. Live migration of full VM is only possible if they use a common hypervisor. However, we do not wish to force all testbeds to use the same hypervisor on their physical servers; moreover testbeds are already using different hypervisors for legacy reasons. Using containers to implement the virtual functions reduces the additional virtualization overhead for this second layer of abstraction.

O2CMF [Ceravolo et al. 2018] is based on OpenStack and is responsible for creating and managing VNF in a user's slice. To that end, it is composed by two elements: an AM and an orchestrator. The latter is offered to the user as a VM through of an orchestrator image. This orchestrator supports specifying an NFV experiment with monitoring and scaling policies using the TOSCA language,to send commands and to access any other VNF in the experiment.

Network functions and applications can be developed by using the Service Layer implemented in the FUTEBOL-CF architecture. This layer is responsible for deploying, for instance, SDR controllers (*e.g.*, Open Air Interface Evolved Packet Core nodes, SRS LTE eNodeB implementation). Moreover, this layer is composed of the user's applications, *e.g.*, a video server that will stream video during the experiment. Furthermore, this layer can provide experimentation tools to generate synthetic/realistic traffic and monitor the experiment network activity.

4. An orchestrated inter-testbed experiment

This experiment presents a scenario envisioned in Mobile Edge Computing [Hu et al. 2015] experiments. The experimental scenario is composed of two tiers of processing. The first and nearer to the user is the Fog, which has the characteristic of low processing power, but high network performance such as low network latency, low packet loss, and high network throughput. The second is the Cloud which has high processing power and typically worse network performance than the Fog. The main idea of this scenario is to monitor the quality of the network and computational resources, and compare the deployment and migration behavior of remote applications in such situations. To perform the monitoring and orchestration of the remote applications, we utilised COPA, a control framework application provided by FUTEBOL.

The experiment setup is illustrated in Figure 2. Currently, this experiment can be fully deployed in the FUTEBOL's testbeds. To access the hardware and virtual resources available by the consortium, we utilise JFed which provides a full list of the testbed resources and communicates directly with heterogeneous testbed control framework applications. JFed also enables the experimenter to reserve for a period of time a group of resources such as Raspberry Pis and Virtual Machines. During the reservation, we can also allocate Control Framework Services such as

[1] http://www.ict-futebol.org.br/wp-content/uploads/Deliverable5.1.pdf

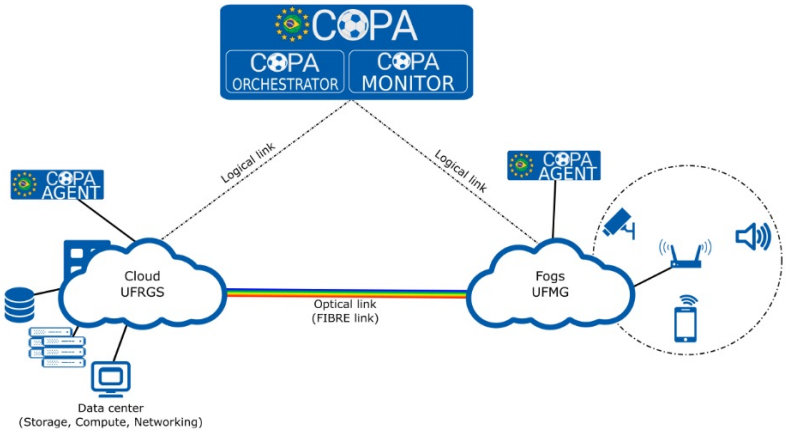
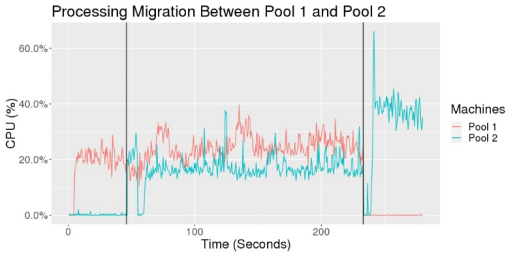


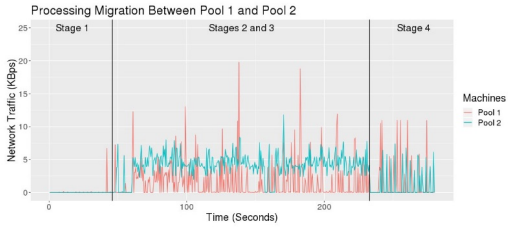
Figure 2. Experimental setup

COPA. Finally, inter-testbed connectivity is provided within certain testbeds (at the writing of this paper the links UFMG-UFRGS, UFRGS-Bristol, Bristol-TCD are operational).

Since this experiment is more focused on the use of the testbeds and not actually new scientific results, we show here a simple migration experiment. The user employs the COPA graphic user interface to perform a live migration of a container within testbeds. Each testbed will run a pool of COPA resources. COPA monitors the resource usage (link and CPU) of the pools during the entire process. Such an experiment could be built in order to do live migration of IoT services, enabling the application to process the inputs within the required QoS levels. Our previous paper shows some application-related results for such an IoT scenario [Silva et al. 2017].



(a) CPU usage in the COPA pools.



(b) Network traffic during the experiment.

Figure 3(a) shows the CPU load in Pool 1 and Pool 2 as the migration occurs. In the first part of the graph (first stage, defined by the period before the black vertical line), the usage of Pool 1 CPU is around 20%, while in Pool 2 is close to 0%. When the migration process starts, the CPU use of Pool 2 increases because it is receiving the Object Detection container, but it is not running it yet. At this stage, object detection still takes place in Pool 1. Only in the final of the second stage the object detection (after the second vertical black line) occurs in Pool 2. In the final stage, the migration process is finalised and the CPU load of Pool 2 is around 40%, while the load in Pool 1 is close to 0% as expected.

Figure 3(b) shows the network traffic of Pool 1 and Pool 2. In the left part of the figure (before the dark horizontal line), the traffic in Pool 2 is close to 0 bps. This

happens because all the network load is in Pool 1, since it is running the Object Detection container. When the migration starts, Pool 2 must receive the Object Detection container, so the network traffic increases. This is shown in the graph in the curves between the two dark horizontal lines. Pool 1 has the same behavior, once it must send the container to Pool 2 for migration. When the migration ends, the network traffic of Pool 1 decreases, while the traffic of Pool 2 increases.

5. Conclusions

This paper presented the functionalities of the FUTEBOL testbeds and its framework from the point of view of the user. 5G research requires the orchestration of complex networks scenarios, spanning multiple network domains (such as optical, wireless, packet, and cloud) and over a wide area. The FUTEBOL project aims to support such experiments by providing heterogeneous physical resources among its testbeds. Those resources support complex services, via service catalogues. Further, the resources of multiple testbeds can be interconnected, thanks to the connectivity provided by RNP and Janet. Once the experiment is running, the experimenter will need to orchestrate those multiple domains so that the service's requirements can be accomplished. For that end, FUTEBOL supports two orchestrators, called COPA and ABNO.

Acknowledgements

The research leading to these results received funding from the European Commission H2020 programme under grant agreement no. 688941 (FUTEBOL), as well from the Brazilian Ministry of Science, Technology, Innovation, and Communication (MCTIC) through RNP and CTIC.

References

Ceravolo, I. et al. (2018). O2CMF: Experiment-as-a-service for agile Fed4Fire deployment of programmable NFV. In *Optical Fiber Communication Conference*.

FUTEBOL (2019). FUTEBOL - Federated Union of Telecom- munications Research Facilities for an EU-Brazil Open Laboratory. `http://www.ict-futebol.org.br/`.

Hammad, A. et al. (2017). Demonstration of NFV content delivery using SDN-enabled virtual infrastructures. In *Optical Fiber Communication Conference*.

Hu, Y. C., Patel, M., Sabella, D., Sprecher, N., and Young, V. (2015). Mobile edge computing—A key technology towards 5G. *ETSI white paper*, 11(11):1–16.

Marques, P. et al. (2017). Experiments Overview of the EU-Brazil FUTEBOL Project. In *European Conference on Networks and Communications*.

Raaf, B., Faerber, M., Badic, B., and Frascolla, V. (2014). Key technology advancements driving mobile communications from generation to generation. *Intel Technology Journal*, 18(1).

Silva, A. P., Abreu, B. A., Silva, E. B., Carvalho, M., Nunes, M., Marotta, M., Hammad, A., Silva, C. F. M., Pinheiro, J. F. N., Both, C. B., Marquez-Barja, J. M.,

and DaSilva, L. A. (2017). Impact of fog and cloud computing on an iot service running over an optical/wireless network testbed. In *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 535–540.

Projeto NECOS: Rumo ao Fatiamento Leve de Recursos em Infraestruturas de Nuvens Federadas

Fernando N. N. Farias[1], Billy A. Pinheiro[1], Antônio J. G. Abelém[1], Paulo D. Maciel Jr.[2,5], André L. B. Rocha[2], Fábio L. Verdi[2], Augusto V. Neto[6], Sand L. Correa[4], Marcelo Nascimento[3], Rafael Pasquini[7], Christian Esteve Rothenberg[8]

[1]Universidade Federal do Pará (UFPA)

[2]Universidade Federal de São Carlos (UFSCar)

[3]Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPqD)

[4]Universidade Federal de Goiás (UFG)

[5]Instituto Federal da Paraíba (IFPB)

[6]Universidade Federal do Rio Grande do Norte (UFRN)

[7]Universidade Federal de Uberlândia (UFU)

[8]Universidade Estadual de Campinas (UNICAMP)

{fennf,billy,abelem}@ufpa.br, {beltrami,verdi}@ufscar.br, marcelon@cpqd.com.br

sand@inf.ufg.br, paulo.maciel@ifpb.edu.br, augusto@dimap.ufrn.br, rafael.pasquini@ufu.br

chesteve@dca.fee.unicamp.br

Abstract. *The Novel Enablers for Cloud Slicing (NECOS) project proposes a solution that seeks to automate the optimized cloud configuration process, providing uniform management with a high level of autonomy for the computing, connectivity and storage resources currently separated, based on the Lightweight Slice Defined Cloud (LSDC) concept. In this paper, we discuss the motivation, objectives, architecture, research challenges and initial efforts of the NECOS project through a set of use cases.*

Resumo. *O projeto Novel Enablers for Cloud Slicing (NECOS) propõe uma solução que visa automatizar o processo de configuração otimizada de nuvem e rede, fornecendo um gerenciamento uniforme com um alto nível de autonomia para os recursos de computação, conectividade e armazenamento atualmente separados, baseado no conceito LSDC (Lightweight Slice Defined Cloud). Neste artigo, discute-se a motivação, objetivos, arquitetura, desafios de pesquisa e esforços iniciais do projeto NECOS através dos casos de uso definidos.*

1. Introdução

A popularização das tecnologias de computação em nuvem (*Cloud Computing* - CC), tem incentivado o aumento no número de provedores que estão oferecendo serviços em qualquer um dos seus modelos XaaS, por exemplo, Software como Serviço (*Software*

as a Service – SaaS), Plataforma como Serviço (*Platform as a Service* – PaaS), Rede como serviço (*Network as a Service* – NaaS) e Infraestrutura como Serviço (do inglês *Infrastructure as a Service* – IaaS) [Mell and Grance 2011]. Diante desta variedade de serviços em nuvem, há um eminente desafio para que sejam fornecidos sob demanda respeitando os contratos de nível de serviço (*Service Level Aggrement* – SLA) utilizando recursos de computação, conectividade e armazenamento gerenciados separadamente.

Este desafio tem impulsionado a criação de várias soluções como o GENI[1], que automaticamente agenda e provisiona recursos heterogêneos em rede, e o RESERVOIR[2], que levou a orquestração de serviços [de Sousa et al. 2018] um pouco mais longe ao permitir a alocação e a implantação automática de recursos para os serviços à medida que são demandados. No entanto, estas soluções carecem de uma abstração única e escalável que permita atender as diferentes demandas dos serviços em um ambiente heterogêneo.

Para fornecer uma abstração única com abrangência fim-a-fim dos recursos das nuvens (processamento e armazenamento) e de redes, a plataforma NECOS[3] utiliza o conceito de *cloud network slicing* [Silva et al. 2018]. Este conceito tem sido definido de diversas maneiras na literatura, mas em geral entende-se como uma composição de recursos (computação, rede e armazenamento) físicos e virtuais criada sob demanda, contando com uma infraestrutura subjacente comum, com controle, gerenciamento e orquestração independentes. Estas *slices* devem ser mutuamente isoladas e flexíveis o suficiente para acomodar diferentes necessidades dos inquilinos *(tenants)*, fornecendo um modelo de Fatia como um Serviço (*Slice as a Service*) que provê ao *tenant* a gerência da infraestrutura virtual (*Virtual Infrastructure Manager - VIM*) sob demanda [Clayman et al. 2018].

Este artigo apresenta a versão atual da arquitetura do NECOS, os desafios identificados e os casos de uso refinados no decorrer do projeto e que hoje apresentam como principal diferencial a adição do cenário de serviços de IoT para cidades inteligentes.

2. Proposta do NECOS

Detalhes do projeto NECOS são descritos nesta seção, incluindo a abordagem LSDC (*Lightweight Slice Defined Cloud*), a arquitetura do NECOS, os principais desafios vinculados ao conceito de *cloud network slices* fim a fim (End-to-End - E2E), e por fim um workflow de sua criação.

2.1. Lightweight Slice Defined Cloud (LSDC)

A integração dos domínios de rede e computação em nuvem, como também suas operações de gerenciamento, irão permitir uma grande economia e a capacidade de ampliar a flexibilidade do provisionamento de serviços nestas áreas. Atualmente, essas operações são realizadas separadamente e, portanto, combiná-las é uma meta do NECOS com o objetivo de simplificar suas ações. Ao fazer isso, o Projeto NECOS alavancará o amadurecimento dos paradigmas SDN (*Software Defined Networking*) [Kreutz et al. 2015] e virtualização de rede que, através de um projeto e implementações adequadas, ajudará a cumprir os requisitos de inovação de maneira mais confiável, rápida e simples [Manzalini et al. 2016].

[1]http://www.geni.net/

[2]https://www.reservoir-fp7.eu

[3]http://www.h2020-necos.eu/

Na visão do NECOS, o conceito de *cloud network slicing* baseia-se em recursos e funções já disponíveis na plataforma de nuvem. A *slicing* é uma técnica associada ao particionamento de recursos, capaz de criar e redefinir partições conforme necessário. Um *slice* pode ser entendido como um agrupamento de recursos físicos ou virtuais que podem atuar como sub-nuvem e sub-rede aparentemente independentes, podendo acomodar componentes do serviço. As principais características da *cloud network slicing* incluem:

- A implantação simultânea de várias *slices* lógicas, auto-suficientes e independentes, compartilhadas ou particionadas em uma infraestrutura comum.
- Suporte a multi-serviço dinâmico, multi *tenancy* e o meio de integração entre os *players* do mercado.
- A separação de funções simplifica o provisionamento de serviços, a gerência de redes e os desafios relacionados à operacionalidade e integração, especialmente para suportar os serviços de comunicação.
- Operadores de rede, provedores de serviço Internet e proprietários de infraestrutura de nuvem podem explorar a tecnologia *slicing* para: reduzir os gastos operacionais, permitir a programabilidade e a inovação necessárias para enriquecer os serviços oferecidos, disponibilizar serviços personalizados e permitir a capacidade de programabilidade da rede para provedores OTT (*Over-The-Top* ou *content providers*) e outros *players* do mercado sem alterar a infraestrutura física.

O LSDC representa uma nova abordagem para automatizar o processo de configuração otimizada de nuvens pela criação do conceito de *cloud network slicing* em todos os recursos em conjunto de centros de dados federados, bem como, fornecer um gerenciamento uniforme dos recursos de computação, conectividade e armazenamento.

2.2. Arquitetura do NECOS

A arquitetura completa da plataforma NECOS, apresentada na Figura 1, define os componentes para o aprovisionamento de *cloud network slices* E2E em uma federação multitecnológica e de múltiplos provedores. No NECOS, um *cloud network slice* E2E pode ser composto por múltiplas partes (*cloud slices*) aprovisionadas em diferentes centros de dados da federação, e são interconectadas por *network slices*.

Para facilitar a compreensão da arquitetura do NECOS, inicialmente apresentada à comunidade no Entregável D3.1 do NECOS [NECOS 2018b], nós definimos, na Figura 1, quatro áreas distintas, isto é, a do *Tenant* em vermelho, a do Provedor de Recursos em verde, a do *Marketplace* em amarelo e a do Provedor de *Slice* em azul. A seguir, faremos uma descrição dos principais componentes da arquitetura que merecem ser destacados: o *Slice Orchestrator*, o componente de gerência e monitoramento *IMA* e o *Marketplace*.

O *Slice Orchestrator* realiza manipulação de *slices*, isto é, requisita ao *Marketplace* as diferentes partes que serão incluídas em uma *slice* E2E através da realização de uma fase de orquestração inicial. Adicionalmente, ele supervisiona o gerenciamento do ciclo de vida dos serviços que estão sendo executados nas *slices*, baseado nas informações coletadas dos diferentes recursos presentes na *slice* pelo componente *IMA*.

O componente *IMA* (do inglês, *Infrastructure & Monitoring Abstraction*) oferece uma interface *northbound* abstrata que permite ao *Slice Orchestrator* executar suas funções enquanto as informações das partes das *slices* permanecem transparentes. Como

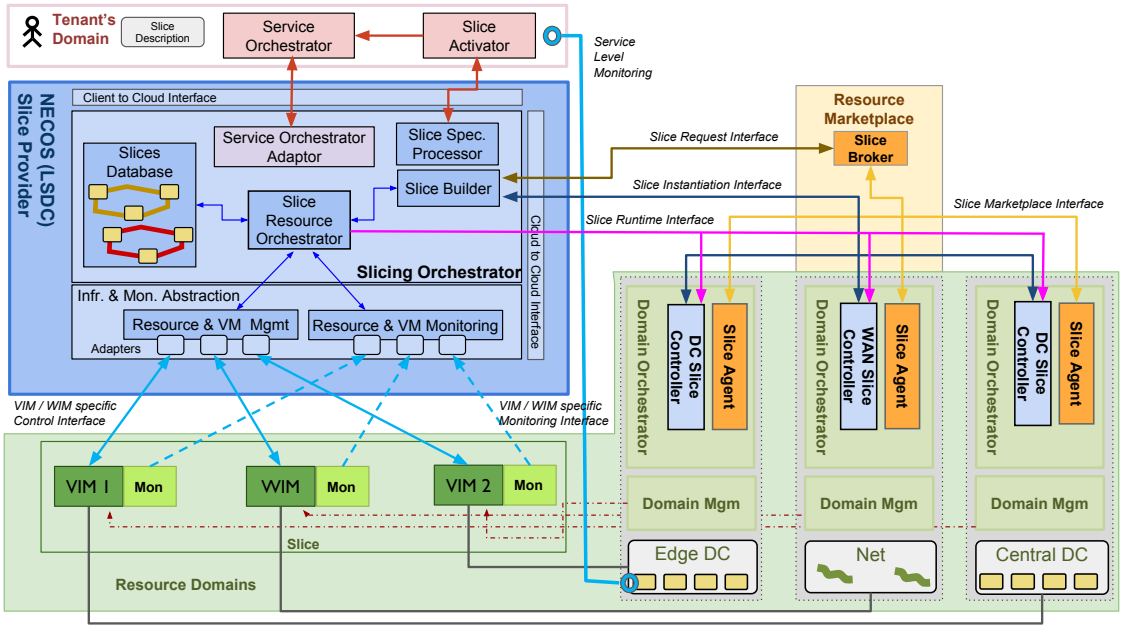


Figura 1. Arquitetura NECOS. Fonte e mais informações: NECOS D3.1.

exemplo de tais funções podemos citar: a coleta de informações sobre topologia de recursos e informações de monitoramento de recursos para cada parte da *slice*; o monitoramento e a verificação do estado dos elementos virtuais alocados a cada parte da *slice*; a coleta de indicadores KPIs (do inglês, *Key performace Indicators*) voltados para recursos como CPU, memória e armazenamento. Para alcançar estes objetivos, múltiplos adaptadores traduzem as requisições provenientes da interface *northbound* para o VIM ou WIM (do inglês, *Virtual/WAN Infrastracture Manager*). Deste modo, eles abstraem as partes das *slices* heterogeneamente na interface *southbound* do *IMA*.

A descoberta de recursos que irão compor uma *slice* E2E é responsabilidade do componente *Marketplace*, que é um sistema distribuído responsável por localizar partes de *slices* adequadas a partir de um conjunto de infraestruturas de nuvem federadas. Como mostrado na Figura 1, o *Marketplace* é formado por dois componentes, nomeados de *Slice Broker* e *Slice Agent*. O *Slice Broker* descobre as partes das *slices* através da interação com um conjunto de *Slice Agents* hospedados nos domínios de recursos envolvidos.

2.3. Desafios do Projeto

Embora a abordagem de prover slice como serviço proposta pelo NECOS traga inúmeros benefícios para tecnologias emergentes, como por exemplo, a computação em nuvem, redes móveis 5G, diversas verticais de IoT (do inglês, *Internet of Things*), entre outras, há ainda algumas lacunas e desafios que necessitam atenção para suprir eficientemente as necessidades que tangenciam este novo conceito denominado *slicing*. Alguns dos principais desafios vislumbrados são descritos a seguir.

Virtualização da Infraestrutura Física: Este conceito já vem sendo discutido há um certo tempo, entretanto com o surgimento do conceito *slicing*, aspectos importantes precisam ser melhorados, como por exemplo, o isolamento entre os recursos, a escalabilidade e o desempenho.

Elasticidade Computacional: A elasticidade de uma *slice* é de extrema importância para que o uso dos recursos seja otimizado de modo a respeitar as SLAs acordadas. Tal elasticidade pode ocorrer no nível de infraestrutura física, virtual ou até mesmo no nível da aplicação.
Heterogeneidade: A heterogeneidade inserida pelos diferentes provedores de infraestrutura é um desafio para ser estudado. Sendo assim, cabe ao *Slice Provider* abstrair as diferentes tecnologias e infraestruturas dos provedores, de modo a entregar uma visão transparente e unificada dos recursos E2E que o *Tenant* requisitou.
Segurança: A segurança é um dos desafios tradicionais no escopo de redes de computadores em geral. Sendo assim, para o conceito de *slicing* que envolve diferentes infraestruturas, e a comunicação entre diferentes provedores, os desafios são ainda maiores.
Gerência da Mobilidade: A mobilidade de dispositivos que necessitam de constante conexão é uma característica relacionada a diversos casos de uso envolvendo *slices*. Sendo assim, a gerência, a melhoria do tráfego e a melhoria dos protocolos de comunicação dos dispositivos móveis são pontos que devem ser aprimorados.

2.4. Workflow de Criação de Slices

Esta Seção apresenta um *workflow*, usando a metodologia BPMN (do inglês, *Business Process Model and Notation*), que relaciona os componentes presentes na arquitetura proposta na Figura 1. O *workflow* descrito, que é detalhado no Entregável D5.1 do NECOS [NECOS 2018c], envolve a operação de criação de uma nova *slice*.

O *workflow* apresentado na Figura 2 adota a seguinte notação: o círculo em verde denota o início enquanto o laranja denota o término do *workflow*. Os retângulos cinzas representam os componentes da arquitetura do NECOS. Os retângulos tracejados, que agrupam componentes, correspondem as áreas descritas na Seção 2.2. Por fim, as interfaces de comunicação entre os componentes são ilustradas pelos círculos em azul.

A Figura 2 ilustra todos os módulos da arquitetura NECOS envolvidos no *workflow* de Criação da Slice, as flechas representam as interações entre os componentes para

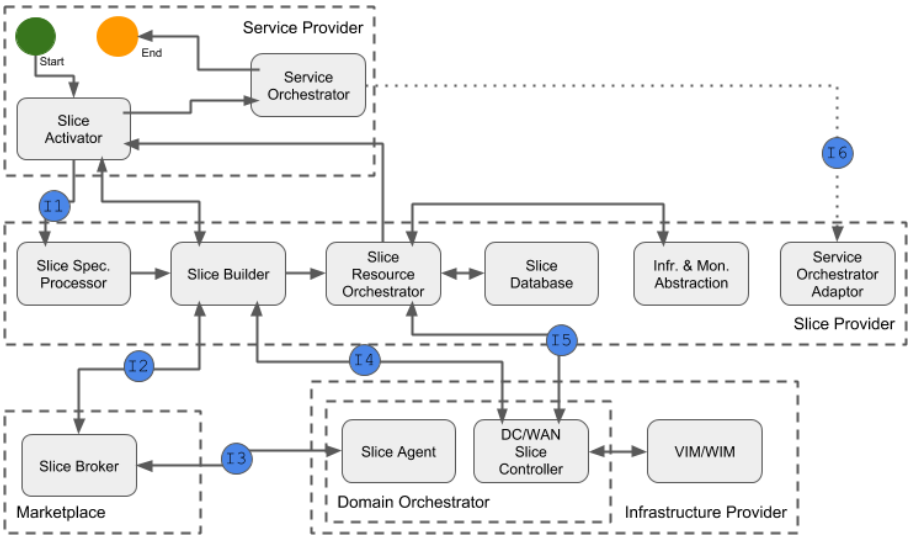


Figura 2. *Workflow* de Criação da Slice Fim-a-Fim. Fonte e mais informações: NECOS D5.1.

a criação da *slice*. O *workflow* é iniciado pelo *tenant* no *Service Provider* através do componente *Slice Activator*, sua função é iniciar a requisição desta nova *slice* para o *Slice Provider* interagindo com o componente *Slice Specification Processor* via Interface 1 (I1). O *Slice Specification Processor* irá extrair as especificações de infraestrutura e serviço requisitadas pelo *Tenant* e então envia para o *Slice Builder*. O *Slice Builder* em conjunto com os componentes *Slice Broker* (via I2) e *Slice Agent* (via I3) possui como função analisar os recursos disponíveis nos provedores de infraestrutura e então otimizar a alocação dos recursos ofertados pelo *Marketplace*. Após definir quais provedores de infraestrutura irão compor a *slice*, o *Slice Builder* se comunica com o *DC/WAN Slice Controller* de cada um dos provedores para instanciar os recursos que foram requisitados pelo *Tenant* (via I4). O *DC WAN Slice Controller* retorna para o *Slice Builder* os apontadores para os novos recursos, e então a informação é enviada para o *Slice Resource Orchestrator* que possui como responsabilidade orquestrar os recursos instanciados de modo a otimizar a utilização dos recursos presentes na *slice*. O *Slice Resource Orquestrator* é um componente de suma importância na arquitetura do NECOS, pois ele além de ser responsável pela orquestração dos recursos, ele também é responsável pela instanciação dos *VIMs/WIMs*. A instanciação destes elementos é de suma importância para o *deployment* de um novo serviço e/ou recurso virtual (Máquina Virtual, Container) para isto ele utiliza a Interface 5 (I5) na qual ele se comunica diretamente com os *DC/WAN Slice Controllers*. Além desta função, o *Slice Resource Orchestrator* gerencia o componente de monitoramento da *slice* (*IMA*) assim como o banco de dados que irá armazenar tanto as informações da *slice* quanto às KPIs monitoradas pela *IMA*. Por fim, após a *slice* estar operante, os serviços do *Tenant* podem ser instanciados e então orquestrados pelo *Service Orchestrator* via a Interface 6 (I6) na parte mais à direita da figura.

3. Casos de Uso

O projeto NECOS identificou *Telco Cloud* e o MEC (do inglês, *Multi-access Edge Computing*) como casos de uso para definir uma arquitetura de federação de nuvem que suporta *slicing* como um serviço. Esses dois casos de uso, apresentados na Figura 3, estão atraindo bastante atenção na indústria com o desenvolvimento de padrões relacionados a eles e o aumento das ofertas comerciais.

Do ponto de vista da *Telco Cloud*, várias iniciativas estão progredindo, como o CORD no ONF e a abordagem CloudCO no BBF, conforme relatado no Entregável D2.2 [NECOS 2018a]. As arquiteturas ali definidas continuam a evoluir e espera-se que possam ser incorporadas em ofertas comerciais no curto prazo. Da mesma forma, o ambiente de computação de borda está se expandindo além do ETSI MEC, multiplicando o número de soluções, tais como o EdgeX Foundry, o grupo de trabalho TIP do Facebook em *Edge Computing*, e Akraino, para mencionar algumas.

Todas essas iniciativas reforçam o escopo do NECOS como uma solução *crossover* de fatiamento que poderia ser implantada para federar opções tão diversas, tanto na *Telco Cloud* quanto no MEC. Além dos casos de uso, o NECOS definiu alguns cenários de aplicabilidade que podem usar esse tipo de ambiente de execução. Os cenários permitiram que a noção de serviço seja executada na *slice* fornecida pelo sistema.

Dentro do contexto destes dois casos de uso, alguns cenários foram criados para obter os requisitos iniciais da plataforma NECOS e para mostrar sua aplicação em dife-

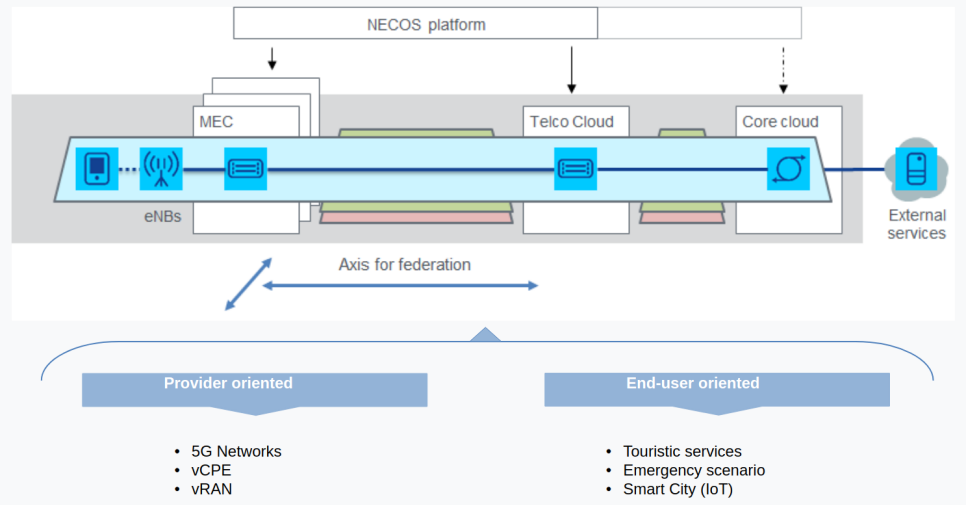


Figura 3. Casos de Uso. Fonte e mais informações: NECOS D2.2.

rentes contextos de rede. A seguir, sumarizamos estes cenários e suas derivações.

vRAN: do inglês, *virtual Radio Access Network*, o cenário vRAN aproveita as tendências de virtualização para fornecer funções de rede associadas à RAN e também pode ser estendido para suportar diferentes provedores de serviços.
Serviços 5G: este cenário aborda os requisitos das operadoras de rede para adaptar as redes existentes para fornecer serviços 5G futuros, como o Banda Larga Móvel aprimorado (eMBB), o compartilhamento de vídeo de alta definição, as comunicações maciças de tipo de máquina (mMTC) e as comunicações ultraconfiáveis e de baixa latência (uRLLC).
vCPE: do inglês, *virtual Customer Premises Equipment*, esse cenário explora os ambientes de Telecomunicações voltados para o uso de recursos de nuvem, o que permite a evolução dos *Central Offices*, aplicando a integração de elementos virtuais e não-virtuais.
Distribuição de Conteúdo Turístico: este cenário aborda a necessidade de um Centro Metropolitano de Turismo para fornecer conteúdo turístico com reconhecimento de local a um grande número de visitantes em uma área metropolitana.
Próxima Geração de Aplicações Turísticas: espera-se que os aplicativos de realidade aumentada melhorem a experiência dos viajantes e aumentem o valor do produto cultural oferecido pelas autoridades municipais.
Distribuição de Conteúdo em Cidades Inteligentes: este cenário aborda a necessidade de integrar diferentes provedores de infraestrutura em uma cidade para fornecer a entrega de conteúdo de dados em uma Cidade Inteligente. Um contribuição inicial está descrita em [Carmo et al. 2019], que propõe uma abordagem para definir em *slices* as infraestruturas de redes de acesso WiFi compartilhadas.
Monitoramento Metropolitano Integrado: este cenário aborda um Centro de Segurança Pública responsável por melhorar o gerenciamento e resolução de incidentes para um número significativo de socorristas (por exemplo, departamento de polícia, serviços de incêndio e resgate, serviços médicos de emergência e obras), bem como sua interação com a população civil em uma área metropolitana.

4. Conclusões

Neste artigo, apresenta-se a proposta NECOS que objetiva abordar as limitações das infraestruturas atuais de nuvem, propondo a abordagem *Lightweight Slice Defined Cloud* (LSDC). O LSDC atua como ativador ao *Cloud Slicing* por meio do conceito de *Slice as a Service*. Além disso, ele oferece uma estrutura de gerenciamento unificada para domínios tanto de rede quanto computacionais, e também, fornece uma orquestração inteligente para infraestruturas de nuvem federadas.

Agradecimentos

Esta pesquisa é financiada pela Comissão Europeia e Ministério da Ciência, Tecnologia, Inovação e Comunicação (MCTIC) através da RNP e CTIC no contexto da 4º chamada conjunta EU-BR, acordo #777067 (*NECOS - Novel Enablers for Cloud Slicing*). Este estudo foi financiado em parte pela CAPES - Código Financeiro 001, e pelo CNPq.

Referências

Carmo, M., Silva, F. S. D., Neto, A. V., Corujo, D., and Aguiar, R. (2019). Network-cloud slicing definitions for wi-fi sharing systems to enhance 5g ultra dense network capabilities. *Wireless Communications and Mobile Computing*, 2019.

Clayman, S., Tusa, F., and Galis, A. (2018). Extending Slices into Data Centers: the VIM on-demand model. In *2018 9th International Conference on the Network of the Future (NOF)*, pages 31–38.

de Sousa, N. F. S., Perez, D. A. L., Rosa, R. V., Santos, M. A. S., and Rothenberg, C. E. (2018). Network Service Orchestration: A Survey. `http://arxiv.org/abs/1803.06596`.

Kreutz, D., Ramos, F. M. V., Veríssimo, P. E., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. (2015). Software-Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*.

Manzalini, A., Buyukkoc, C., CHEMOUIL, P., Kuklinski, S., Callegati, F., Galis, A., Odini, M.-P., I, C.-L., Huang, J., Bursell, M., Crespi, N., Healy, E., and Sharrock, S. (2016). Towards 5G Software-Defined Ecosystems: Technical Challenges, Business Sustainability and Policy Issues. Research report.

Mell, P. M. and Grance, T. (2011). Sp 800-145. the nist definition of cloud computing. Technical report, Gaithersburg, MD, United States.

NECOS (2018a). D2.2: Consolidated definition of use cases, business models and requirements analysis. `http://www.maps.upc.edu/public/d2.2_final_v2.0.pdf`.

NECOS (2018b). D3.1: Necos system architecture and platform specification. `http://www.maps.upc.edu/public/NECOS%20D3.1%20final.pdf`.

NECOS (2018c). D5.1: Architectural update, monitoring and control policies frameworks. `http://www.maps.upc.edu/public/D5.1_final.pdf`.

Silva, F. S. D., Lemos, M. O. O., Medeiros, A., Neto, A. V., Pasquini, R., Moura, D., Rothenberg, C., Mamatas, L., Correa, S. L., Cardoso, K. V., Marcondes, C., ABelem, A., Nascimento, M., Galis, A., Contreras, L., Serrat, J., and Papadimitriou, P. (2018). Necos project: Towards lightweight slicing of cloud federated infrastructures. In *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, pages 406–414.

5GINFIRE: Enabling an NFV Based Experimentation of Vertical Industries in the 5G Context

Flávio de Oliveira Silva[1], Pedro Frosi Rosa[1], Halid Hrasnica[2], Anastasius Gavras[2]

[1]Faculty of Computing (FACOM) – Federal University of Uberlandia (UFU)
Uberlandia – MG – Brazil

[2]Eurescom GmbH
Heidelberg – Germany

{flavio,pfrosi}@ufu.br, {hrasnica,gavras}@eurescom.eu

Abstract. *5G aims to provide a wireless access network that supports the requirements of different vertical markets such as automotive and smart cities. 5G oriented services use the Internet of Things (IoT) and will use the network programmability by integrating Software-defined Networking (NFV) and Network Function Virtualization (NFV). This context presents several multi-facet challenges to experiment with these services. 5GINFIRE is a 5G oriented experimental facility where different actors of the 5G ecosystem can experimentally contribute to 5G development and deployment. This work presents the 5GGINFIRE ecosystem, highlights some of its capabilities and some 5G ongoing experiments.*

1. Introduction

At this moment the world is engaged in the design and deployment of the next generation of International Mobile Telecommunications (IMT-2020), a.k.a, 5G [Gupta and Jha 2015] [Agiwal et al. 2016].

5G aims to provide a wireless access network that supports the requirements of different vertical markets such as agriculture, automotive, energy, health, media and entertainment, public safety, manufacturing, and smart cities [Zafeiropoulos et al. 2018]. The 5G network will provide not only an enhanced Mobile Broadband (eMMB), but Ultra-Reliable Low-Latency Communication (URLLC) and Massive Machine Type Communication (mMTC) [Elayoubi et al. 2016] [Lema et al. 2017].

The Internet of Things (IoT) is a common technology used on these different verticals. IoT is an enabler for a completely new set of services, applications, and business models to all the different actors in the telecommunication ecosystem.

5G also will rely on cloud computing and in the network programmability by integrating Software-defined Networking (NFV) and Network Function Virtualization (NFV).

One characteristic of these verticals is that they do not operate and evolve in a silo mode anymore, relying on closed and proprietary technological environments, but they are becoming open ecosystem built on top of shared physical infrastructures, (re)using the same open source components and APIs and sharing resources utilizing virtualization.

This context presents a number of multi-facet challenges (such as interoperability, inter-connectivity, federation, resource sharing, slicing, resource scheduling, operations)

of unprecedented magnitude and complexity and it calls for the appropriate identification, introduction, integration and efficient operation of a series of common architectural elements and infrastructure assets that should be combined together with the vertical-specific components.

The 5GINFIRE project goal is to provide a 5G oriented experimental facility where different actors of the 5G ecosystem (startups, industry, SDOs, operators, academia, research institutes) can experimentally contribute to 5G development and deployment.

5GINGRE infrastructure enables the experimentation of Experimental Vertical Instances (EVIs), a composition of several virtual functions spanning all layers from application and services to networking with a focus on 5G verticals.

This work presents the 5GGINFIRE ecosystem, highlights some of its capabilities and some 5G ongoing experiments.

This work is organized as follows: Section 2 describes the 5GINFIRE facility. Section 3 details some 5G related experiments that uses 5GINFIRE. Section 4 presents some final considerations and presents the future directions of this research.

2. The 5GINFIRE Ecosystem

The 5GINFIRE concept focuses on the Management and Orchestration (MANO) side of the 5G network that is composed of a set of Virtual Functions of Network and Verticals (VxFs). In this context, a network slice is an end-to-end logical mobile network with the VxFs deployed on the different cloud infrastructures managed by an operator.

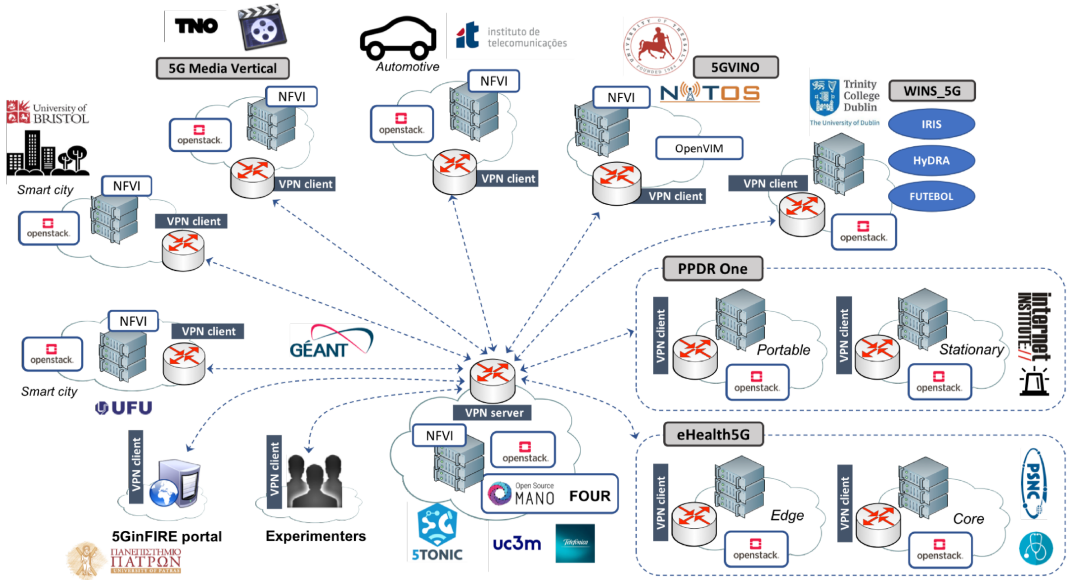


Figure 1. Overview of The 5GINFIRE Core MANO Platform

The 5GINFIRE ecosystem brought a series of particular challenges for the MANO platform to be used, especially related to the high diversity of functions to be considered and the multi-domain nature of the infrastructure to be managed.

The technical solution adopted by the project considers the utilization of a single orchestration domain, where an NFV orchestrator, implemented with Open Source MANO (OSM) [ETSI 2018], manages and coordinates the creation of Network Services (NS). A Network Service is as a composition VxFs. Each of these VxFs may be in turn deployed at any of the experimental infrastructures made available by 5GinFIRE partners, called infrastructure providers.

The experimenter creates one or more NS and using the 5GINFIRE portal [5GINFIRE 2018] is requests the deployment over the 5GinFIRE infrastructure and then it can run the 5G related experiments. The University of Patras (UoP), in Greece, hosts the 5GINIRE portal.

Figure 1 presents an overview of the 5GINFIRE Core MANO platform. The OSM release four [Hoban et al. 2018] is deployed in the 5G Telefonica Open Innovation Laboratory (5TONIC) [IMDEA 2016] made available by Telefónica Investigación y Desarrollo(TID) and Universidad Carlos III de Madrid (UC3M). This instance of OSM orchestrates the VxFs across all infrastructure providers as expressed by the NS. Considering a 5G network, it represents the MANO instance of a network operator.

Each infrastructure provider is in charge of the deployment and maintenance of a Virtualized Infrastructure Manager (VIM), compliant with the OSM software stack. On top of that, the NFV orchestrator of 5GinFIRE, deployed at 5TONIC, interacts with the VIMs of the testbed providers involved in a service deployment: it coordinates the allocation and setup of the computing, storage and network resources which are necessary for the instantiation and interconnection of the VxFs that compose the network service.

Besides the VIM, each infrastructure provider explores one of the 5G verticals. To this end, it also has a local deployment of different sensors and equipment to support specific services of a vertical.

At the Instituto de Telecomunicações in Aveiro (ITAv) the focus is the automotive vertical. The infrastructure made available by the University of Bristol (UNIVBRIS), supports experimentation activities over a smart city environment in the city of Bristol (UK). The experimentation facility at the Federal University of Uberlândia (UFU), located at Uberlândia (Brazil), enables trials with smart city applications with a particular consideration on the edge network resources.

The infrastructure provided by the NITOS testbed, hosted by the University of Thessaly (Greece), provides access to programmable resources for wireless networking, SDN and cloud computing facilities. The extension of the NITOS testbed was called 5GVINO.

A reconfigurable radio testbed at Trinity College Dublin (Ireland), called IRIS, supports radio hardware, cloud-RAN, NFV, and SDN technologies. This testbed has been extended and made available for experimentation activities in 5GinFIRE. We refer to this testbed extension as WINS$_5G$.

In the the Poznan Supercomputing and Networking Center (PSNC), in Polandm, an eHealth experimental vertical facility, called eHealth5G supports experimentation in the area of telemedicine and eHealth, offering access to: realistic eHealth equipment; a small Edge Cloud, close to eHealth devices; and a core cloud accessible via MPLS/Optical

service provider network.

NITOS and IRIS tested were created inside the Future Internet Research and Experimentation (FIRE) [Gavras et al. 2007]. Considering the number of assets already deployed on Future Internet Research and Experimentation (FIRE) testbeds, one of the goals of 5GINFIRE is to reuse these assets by enabling the deployment of the NFV network services atop of the FIRE based facilities. NITOS and IRIS testbeds are concrete examples of this reuse.

By using open calls, new infrastructure providers are engaging in the 5GINFIRE ecosystem, integrating new facilities. The TNO institute, in the Netherlands, will host the 5G Media Vertical, a testbed facility that targets the 5G Media Vertical industry. The Internet Institute (II), in Slovenia, hosts the Public Protection and Disaster Relief (PPDR) facility for outdoor and indoor experiments with 5G network architectures and services in this area.

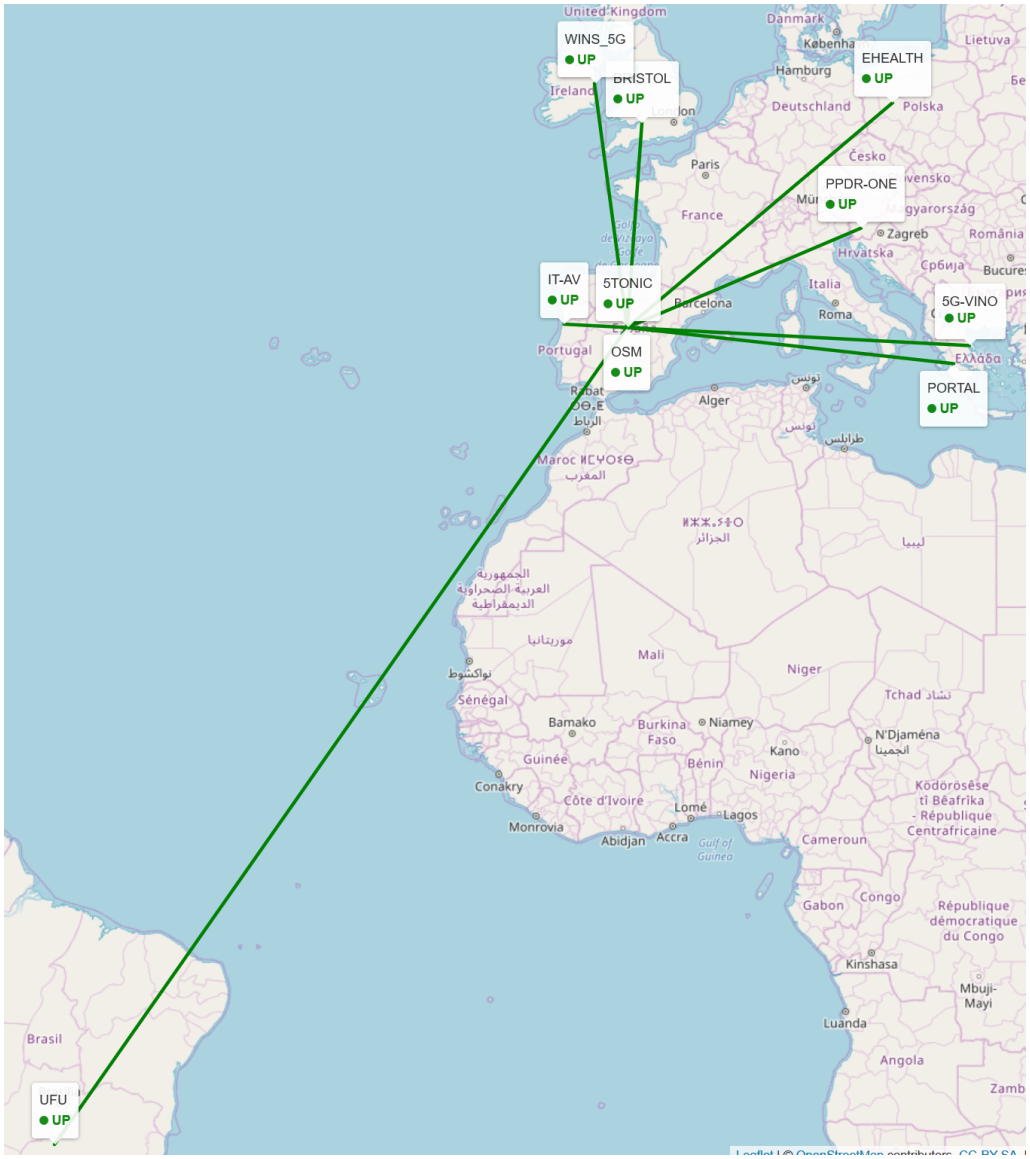


Figure 2. 5GINFIRE Facility Map

5GINFIRE spans between Europe and Brazil. Figure 2 presents the facility map provided in the health check system that monitors the different infrastructure providers already deployed and interconnected.

3. 5GINFIRE Experiments

Under the 5GINFIRE several experiments are ongoing at this moment are they are related to different verticals such as automotive and smart cities [5GINFIRE 2018].

The Context-Aware Video Controller for autonomous transport and security monitoring (CAVICO) allows controlling the video stream from the onboard camera in the function of transmission QoS and video QoE parameters measured by specific software components. The Hybrid Communications to Foster 5G Vehicular Services (SURROGATES) aims at virtualizing regular On-Board Unit (OBU) tasks requiring a high computing load, based on a hybrid communication system that allows a proactive connection between the real OBU and a virtual one.

The Vulnerable Road Users Safety (VRU-Safe) is an experimental evaluation of a network service with computing and networking capabilities, which is able to operate in a distributed manner utilizing either MEC, or Cloud capabilities (in the form of virtualized network functions), depending on the location and the network information of the involved OBUs and VRUs.

The Service Function Chaining orchestration application for low latency guarantees (SFCLola) provides tenants with a latency-aware service function chaining management while minimizing support required from infrastructure operators.

The 5G Smart City Robotic Surveillance Platform (RobotView5G) enables real-time video surveillance using WiFi and mobile networks from remote-controlled robots, drones, and other video-monitoring devices and conduct experiments with this scenario.

These are a few examples of different experiments conducted in the 5GINFIRE ecosystem. At this moment, other ones related to eHealth, mission-critical services and entertainment verticals are also taking place.

4. Concluding Remark and Future Work

5G oriented services will use the Internet of Things (IoT) by connecting different devices located in the network edge. These services also will use a programmable network based on Software-defined Networking (NFV) and Network Function Virtualization (NFV). The experimentation of such services are complex and pose several challenges.

The 5GINFIRE facility enables an NFV based experimentation of different services focusing on several vertical industries.

With an interconnected facility that spans over Europe and Brazil, the 5GINFIRE testbed supports at this moment several experiments in the automotive and smart cities scenarios.

The 5GINFIRE facility is still evolving and soon will support the 5G media vertical and mission critical in PPDR scenarios, thus improving the experimentation of 5G oriented services contributing to 5G design and deployment.

References

5GINFIRE (2018). 5ginfire Experiments.

Agiwal, M., Roy, A., and Saxena, N. (2016). Next Generation 5g Wireless Networks: A Comprehensive Survey. *IEEE Communications Surveys Tutorials*, 18(3):1617–1655.

Elayoubi, S. E., Fallgren, M., Spapis, P., Zimmermann, G., Martín-Sacristán, D., Yang, C., Jeux, S., Agyapong, P., Campoy, L., Qi, Y., and Singh, S. (2016). 5g service requirements and operational use cases: Analysis and METIS II vision. In *2016 European Conference on Networks and Communications (EuCNC)*, pages 158–162.

ETSI (2018). Open Source MANO (ETSI).

Gavras, A., Karila, A., Fdida, S., May, M., and Potts, M. (2007). Future Internet Research and Experimentation: The FIRE Initiative. *SIGCOMM Comput. Commun. Rev.*, 37(3):89–92.

Gupta, A. and Jha, R. K. (2015). A Survey of 5g Network: Architecture and Emerging Technologies. *IEEE Access*, 3:1206–1232.

Hoban, A., Israel, A., Sepúlveda, A. T., Salguero, F. J. R., de Blas, G. G., Kashalkar, K., Ceppi, M., Shuttleworth, M., Harper, M., Marchetti, M., Velandy, R., Almagia, S., and Little, V. (2018). OSM Release Four Technical Overview.

IMDEA (2016). 5tonic.

Lema, M. A., Laya, A., Mahmoodi, T., Cuevas, M., Sachs, J., Markendahl, J., and Dohler, M. (2017). Business Case and Technology Analysis for 5g Low Latency Applications. *IEEE Access*, 5:5917–5935.

Zafeiropoulos, A., Gouvas, P., Fotopoulou, E., Tsiolis, G., Xirofotos, T., Bonnet, J., Carrozzo, G., Rizou, S., Gavras, A., Barros, M. J., Costa-Perez, X., Prasad, A., Gramaglia, M., Tzanakaki, A., Simeonidou, D., Cosmas, J., Fallgren, M., Muñoz, R., and Vilalta, R. (2018). Enabling Vertical Industries Adoption of 5g Technologies: A Cartography of Evolving Solutions. In *2018 European Conference on Networks and Communications (EuCNC)*, pages 1–9.

Realização

Organização

Apoio Institucional

Patrocínio

Diamante

Ouro

Prata

Bronze