UNIVERSIDADE FEDERAL FLUMINENSE

VITOR DOS SANTOS FARIAS

FIBREOSS – Um sistema de gerência para o testbed FIBRE.

NITERÓI

2016

UNIVERSIDADE FEDERAL FLUMINENSE

VITOR DOS SANTOS FARIAS

FIBREOSS – Um sistema de gerência para o testbed FIBRE.

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Telecomunicações da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Engenharia de Telecomunicações. Área de concentração: Sistemas de Telecomunicações

Orientador:
NATALIA CASTRO FERNANDES

NITERÓI

2016

Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de Computação da UFF

F224 Farias, Vitor dos Santos
FIBREOSS : um sistema de gerêrncia para o testbed FIBRE / Vitor dos Santos Farias. – Niterói, RJ : [s.n.], 2016.
151 f.

Dissertação (Mestrado em Engenharia Elétrica e de Telecomunicações) - Universidade Federal Fluminense, 2016.
Orientador: Natalia Castro Fernandes.

1. Sistema de telecomunicação. 2. Testbed. 3. Rede de computadores. I. Título.

CDD 621.382

VITOR DOS SANTOS FARIAS

FIBREOSS – Um sistema de gerência para o testbed FIBRE

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Telecomunicações da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Engenharia de Telecomunicações. Área de concentração: Sistemas de Telecomunicações

Aprovada em Novembro de 2016.

BANCA EXAMINADORA

Prof. NATALIA CASTRO FERNANDES, D.Sc.
Universidade Federal Fluminense (UFF) - Orientador

Prof. JOSÉ FERREIRA DE REZENDE, D.Sc.
Universidade Federal do Rio de Janeiro (UFRJ)

Prof. LUIZ CLAUDIO SCHARA MAGALHÃES, D.Sc.
Universidade Federal Fluminense (UFF)

Prof. RICARDO CAMPANHA CARRANO, D.Sc.
Universidade Federal Fluminense (UFF)

Niterói
2016

"Sunt facta verbis difficiliora."
(Marcus Tullius Cicero, Epistulae Ad Quintum Fratrem)

À minha família.

Agradecimentos

Agradeço aos meus pais, Carla e José Luiz, que tornaram possível cada etapa da minha jornada até este momento. Agradeço também aos amigos e a minha namorada, Maria Isabel, que entenderam a minha ausência durante produção deste trabalho.

À minha orientadora, Professora Natalia e também aos professores Carrano, Jacqueline, Schara e Yona pelo incentivo em seguir a carreira acadêmica.

Agradeço também aos meus gestores na TIM celular S.A., Regina Missias, Jayme Tadeu e João Portugal que possibilitaram a flexibilidade de horário para continuar meus estudos.

À toda equipe do FIBRE, em especial à Iara Machado, pelo voto de confiança em defender este projeto na seleção do PMON, também ao Daniel Marques e Marcos Schwarz que contribuíram com informações indispensáveis a este trabalho.

Este projeto foi financiado pelo Programa de Monitoramento de Redes (PMON) da RNP.

Este projeto faz uso de resultados produzidos pelo testbed FIBRE.

Resumo

A federação de *testbeds* para pesquisas rede de computadores trouxe um grande potencial de inovação criando um ambiente onde recursos são compartilhados entre pesquisadores ao redor do mundo. Porém, manter um laboratório distribuído de larga escala em funcionamento não é uma tarefa trivial. A existência de diversos domínios administrativos e a natureza heterogênea dos recursos torna a operação complexa.

Este trabalho compartilha o conhecimento adquirido na operação e manutenção do *testbed* FIBRE e desenvolve uma solução de monitoração chamada FIBREOSS para auxiliar os administradores do *testbed*. Essa solução tem como objetivo auxiliar a detecção de falhas, fornecer um ponto centralizado de agregação de informações provenientes de monitoração e realizar um controle do nível de serviço testando as capacidades do *testbed* com uma profundidade e detalhamento que não são possíveis utilizando ferramentas tradicionais de monitoração.

O sistema de monitoração proposto é aplicado ao FIBRE. O intuito do FIBREOSS é observar serviços e sistemas que não puderam ser monitorados com ferramentas tradicionais como: os serviços de experimentação OpenFlow, experimentação sem-fio e a disponibilidade dos serviços do *testbed*. Com essa adição à monitoração foi possível obter uma medição mais precisa da disponibilidade do *testbed* encontrando falhas que não eram observáveis antes da implantação do FIBREOSS. Como principais contribuições deste trabalho estão: uma comparação da monitoração com as ferramentas tradicionais com a monitoração do FIBREOSS, estatísticas de falhas do *testbed* e testes de escalabilidade do FIBREOSS

Palavras-chave: Testbed, Monitoração, FIBRE, SDN

Abstract

Federated testbeds for computer network research brought a great innovation potential by creating an environment where resources are shared among researchers around the world. However, maintaining a distributed large-scale laboratory functional is not a trivial task. The operation becomes a complex task due to the existence of many different administrative domains and the heterogeneous nature of the resources.

In this work we share operation and maintenance knowledge obtained with the FIBRE testbed and develop a monitoring solution named FIBREOSS to assist testbed administrators. This solution aims to help fault detection, providing a centralized point of aggregation of monitoring information and performs a service level control by testing the capabilities of testbed with a depth and detail that are not possible using traditional monitoring tools.

We integrate the proposed monitoring system to FIBRE. The purpose of FIBREOSS is to observe services and systems that could not be monitored using traditional tools such as the OpenFlow experimentation services, wireless experimentation and the availability of services of the testbed. With this addition to monitoring it was possible to obtain a more accurate measurement of the availability of testbed finding flaws that were not observable before.

Keywords: Testbed, Monitoração, FIBRE, SDN

Lista de Figuras

2.1 Blocos de construção de um testbed genérico. 6

3.1 Distribuição geográfica das ilhas do FIBRE. [1] 20

3.2 Componentes de uma ilha do FIBRE-BR. Adaptado de [2]. 21

3.3 Separação de plano de dados e controle em uma ilha do FIBRE. 22

3.4 Arquitetura do FIBRE. Adaptado de [1, 3]. 23

3.5 Esquema de comunicação no OMF. 26

3.6 Arquitetura e fluxograma de uso do OMF. 26

3.7 Arquitetura do OCF. 28

3.8 Modelo de gestão do FIBRE. Adaptado de [4] 29

3.9 Arquitetura do ZenOSS. 30

3.10 Maddash - Tabela de Perda de pacotes. 32

4.1 Arquitetura do NOVI. 35

4.2 Arquitetura do TopHat. 36

4.3 Arquitetura do MOST4FIRE. 38

4.4 Arquitetura do GEMINI. 40

5.1 Visão em alto nível do FIBREOSS e uma ilha do FIBRE. 45

5.2 Organização de servidores do FIBREOSS. 47

5.3 Fluxo de dados no FIBREOSS. 48

5.4 Arquitetura detalhada do FIBREOSS mostrando as ações realizadas pelos componentes. 50

5.5 Arquitetura detalhada do FIBREOSS mostrando o fluxo de dados no sistemas. 51

5.6 Fluxograma do agente de teste do OMF - Parte1. 55
5.7 Fluxograma do agente de teste do OMF - Parte2. 56
5.8 Fluxograma do agente de teste do OCF - Parte1. 59
5.9 Fluxograma do agente de teste do OCF - Parte2. 60
5.10 Fluxograma do agente de teste do OCF - Parte3. 61
5.11 Topologia dos testes do plano de dados dentro de uma ilha. 62
5.12 Topologia dos testes do plano de dados em um experimento federado. . . . 63
5.13 Modelo de dependência simplificado. 65
5.14 Exemplo do cálculo da disponibilidade de uma ilha. 66
5.15 Exemplo do diagrama em alto nível de dependências com componentes internos. 67
5.16 Exemplo do cálculo da disponibilidade do serviço/grupo infraestrutura. . . 68
5.17 Árvore de dependência detalhada, mostrando serviços e componentes. . . . 68
5.18 Exemplo de cálculo da disponibilidade com uma NetFPGA indisponíveis. . 71
5.19 Exemplo de cálculo da disponibilidade com duas NetFPGAs e dois nós sem-fio indisponíveis. 72
5.20 Exemplo de cálculo da disponibilidade com LDAP e VPN indisponíveis. . . 73
5.21 Exemplo de cálculo da disponibilidade com duas NetFPGAs e dois nós sem-fio indisponíveis. 74
5.22 Exemplo de cálculo da disponibilidade com o Flowvisor indisponível. 75
6.1 Comparação de medições de disponibilidade na ilha da RNP. 77
6.2 Teste de transferência no plano de dados da VM conectada à NetFPGA1 na ilha da RNP. 78
6.3 Teste de transferência no plano de dados da VM conectada à NetFPGA3 na ilha da RNP. 78
6.4 Teste de transferência no plano de dados da VM conectada à NetFPGA2 na ilha da RNP. 78
6.5 Comparação de medições de disponibilidade na ilha do NOC. 79

6.6 Comparação de medições de disponibilidade na ilha da USP. 80

6.7 Teste de transferência no plano de dados da VM conectada à NetFPGA1 na ilha da USP. 80

6.8 Teste de transferência no plano de dados da VM conectada à NetFPGA2 na ilha da USP. 81

6.9 Teste de transferência no plano de dados da VM conectada à NetFPGA3 na ilha da USP. 81

6.10 Teste de transferência no plano de dados da VM conectada à NetFPGA3 na ilha da USP. 81

6.11 Teste de transferência no plano de dados da VM conectada à NetFPGA3 na ilha da USP. 82

6.12 Comparação de medições de disponibilidade na ilha da UFF. 82

6.13 Comparação de medições de disponibilidade na ilha da UFG. 83

6.14 Teste de transferência no plano de dados da VM conectada à NetFPGA1 na ilha da USP. 83

6.15 Teste de transferência no plano de dados da VM conectada à NetFPGA2 na ilha da USP. 84

6.16 Comparação de medições de disponibilidade na ilha da UFPA. 84

6.17 Comparação de medições de disponibilidade na ilha da UFPE. 85

6.18 Comparação de medições de disponibilidade na ilha da UFRJ. 86

6.19 Teste de transferência no plano de dados da VM conectada à NetFPGA1 na ilha da UFRJ. 86

6.20 Comparação de medições de disponibilidade na ilha da UFSCar. 87

6.21 Disponibilidade média de dispositivos na ilha do CPqD. 88

6.22 Disponibilidade média de dispositivos na ilha do NOC. 89

6.23 Disponibilidade média de dispositivos na ilha do RNP. 89

6.24 Disponibilidade média de dispositivos na ilha do UFF. 90

6.25 Disponibilidade média de dispositivos na ilha do UFG. 90

6.26 Disponibilidade média de dispositivos na ilha do UFPA. 91

6.27 Disponibilidade média de dispositivos na ilha do UFPE. 91

6.28 Disponibilidade média de dispositivos na ilha do UFRJ. 92

6.29 Disponibilidade média de dispositivos na ilha do UFSCar. 92

6.30 Disponibilidade média de dispositivos na ilha do USP. 93

6.31 Queda gradual de banda na conexão na NetFPGA2 na ilha da RNP. . . . 93

6.32 Recuperação banda na conexão na NetFPGA2 na ilha da RNP após reiniciar a VM nos dias 27/09/2016, 28/09/2016 e término da medição no dia 29/09/2016 94

6.33 Distribuição de erros no OCF. 95

6.34 Tempo de processamento do teste de ping. Curvas de nível com diferentes proporções de dispositivos conectados 96

6.35 Tempo de processamento do teste de *ping* no plano de controle em função do número de *threads*. 97

6.36 Tempo de processamento do teste de atraso no plano de dados em função do número de *threads*. 98

6.37 Tempo de processamento do teste de banda no plano de dados em função do número de *threads*. 98

6.38 Tempo de processamento do teste de conexão com a Internet das máquinas virtuais em função do número de *threads*. 99

A.1 Adições ao portal da ilha 111

A.2 Método para avaliação da interseção de um conjunto de eventos. 116

A.3 Tela de inserção de comentários - Servidor fora 117

A.4 Tela de exibição de relatórios - Ping 118

A.5 Tela de inserção de comentários - Desempenho da rede 118

A.6 Tela de status da federação FIBRE-BR 119

A.7 Tela de monitoração do plano de dados Openflow - Parte1 119

A.8 Tela de monitoração do plano de dados Openflow - Parte2 120

A.9 Tela de eventos correntes 121

Lista de Tabelas

2.1 Linha do tempo: Início dos principais projetos de *testbeds* voltados para o desenvolvimento da Internet. 8

3.1 Componentes do FIBRE. 24

5.1 Relação de eventos gerados pelos agentes customizados do FIBREOSS. . . 64

5.2 Hierarquia dos serviços de uma ilha do FIBRE 70

6.1 Disponibilidade média das ilhas no período 88

6.2 Estatística de mensagens do OCF - funções do VTAM 95

Sumário

1 Introdução **1**

1.1 Motivação 2

1.2 Objetivos 3

1.3 Principais contribuições 3

1.4 Organização do trabalho 4

2 Testbeds e o futuro da Internet **5**

2.1 Principais iniciativas de testbeds de redes de computadores 8

2.2 GENI 9

2.2.1 Monitoração da infraestrutura 9

2.2.2 Monitoração dos experimentos no GENI 10

2.2.2.1 INSTOOLS 10

2.2.2.2 LAMP 10

2.2.3 Monitoração completa 11

2.2.3.1 GIMI 11

2.2.4 PlanetLab 11

2.2.5 ORBIT 12

2.3 FIRE 12

2.3.1 Monitoração no FIRE 13

2.3.2 BonFIRE 14

2.3.3 OFELIA 14

2.3.4 FEDERICA 15

2.3.5 *OneLab* . . . 16
2.3.6 NITOS-NITLab . . . 17
2.4 G-Lab . . . 18

3 Projeto FIBRE **19**
3.1 Componentes e arquitetura . . . 20
3.2 OMF . . . 25
3.3 OCF . . . 27
3.4 Operação do *testbed* . . . 27
3.5 ZenOSS . . . 29
3.6 perfSONAR . . . 31
3.7 Problemas enfrentados no FIBRE . . . 32

4 Sistemas de monitoração utilizados em outros testbeds **34**
4.1 NOVI . . . 34
4.2 TopHat . . . 36
4.3 MOST4FIRE . . . 37
4.4 GEMINI . . . 39
4.5 Avaliação das ferramentas . . . 40

5 FIBREOSS - especificação **42**
5.1 Especificação de funcionalidades e requisitos de monitoração de usuários e de operadores . . . 42
5.1.1 Requisitos dos Operadores . . . 42
5.1.2 Requisitos de usuários . . . 43
5.1.3 Requisitos do comitê gestor . . . 44
5.2 Especificação da infraestrutura de servidores do sistema FIBREOSS . . . 45
5.2.1 Detalhamento da Arquitetura . . . 47

5.3 Especificação do módulo de comunicação com o ZenOSS 51
5.3.1 Painel de comentários 52
5.3.2 Relatório de disponibilidade (Teste de Ping) 52
5.4 Especificação do módulo de comunicação com o perfSONAR 53
5.5 Especificação do módulo de comunicação com o OMF 53
5.6 Especificação do módulo de comunicação com o OCF 56
5.7 Teste do plano de dados OpenFlow 61
5.8 Alarmes gerados 63
5.9 Proposta de relatório de disponibilidade 65
5.9.0.1 Exemplos de cálculo 70

6 Resultados **76**

6.1 Análise 76
6.1.1 Disponibilidade 77
6.1.1.1 RNP 77
6.1.1.2 NOC 79
6.1.1.3 USP 80
6.1.1.4 UFF 82
6.1.1.5 UFG 83
6.1.1.6 UFPA 84
6.1.1.7 UFPE 85
6.1.1.8 UFRJ 86
6.1.1.9 UFSCar 87
6.1.1.10 Ilhas inativas 87
6.1.1.11 Resumo 88
6.1.2 Erros 94
6.2 Desempenho e escalabilidade 95

6.2.1 Impacto 99

7 Conclusão **100**

7.1 Trabalhos Futuros 101

Referências **103**

Apêndice A – Detalhes de Implantação **110**

A.1 FIBREOSS - desenvolvimento 110

A.1.1 Introdução 110

A.1.2 Montagem de um servidor piloto para agregar alarmes da ilha UFF 110

A.1.3 Desenvolvimento de uma interface de teste para os módulos 111

A.1.4 Desenvolvimento do módulo de comunicação com o ZenOSS 112

A.1.5 Desenvolvimento do módulo de comunicação com o perfSONAR . . 112

A.1.6 Desenvolvimento do módulo de comunicação e sondas do OMF . . . 112

A.1.7 Desenvolvimento do módulo de comunicação e sondas do OCF . . . 112

A.1.8 Desenvolvimento do módulo de teste de ping 113

A.1.9 Desenvolvimento do relatório de disponibilidade 113

A.1.10 Teste do plano de dados no domínio Openflow 113

A.1.11 Cálculo da disponibilidade utilizando medidas pontuais e intervalos 115

A.1.12 Desenvolvimento de uma interface de usuário para o sistema 116

Apêndice B – Implementação do serviço web **122**

B.1 Modificações no LS-WEB para visualização dos itens monitorados 122

B.2 Implementação da interface de comunicação com o ZenOSS 124

Apêndice C – Relatório de requisitos **128**

C.1 Especificação de funcionalidades e requisitos de monitoração de usuários e de operadores 128

C.1.1 Requisitos dos Operadores 128

C.1.2 Requisitos de usuários 131

C.1.3 Requisitos do sistema de correlação de alarmes 132

Capítulo 1

Introdução

A Internet transformou a comunicação entre as pessoas de tal forma que ela é hoje uma parte essencial nos domínios econômico e social. Uma ferramenta que permite que pessoas ao redor do mundo compartilhem informações, realizem transações financeiras, cirurgias à distância [5], possibilite a construção de uma infraestrutura de energia [6] e cidades [7] mais inteligentes, entre muitas outras aplicações com potencial revolucionário.

Para abrir caminho às novas tecnologias e suportar melhor o que existe hoje, a Internet precisa continuar evoluindo. Conexões com alta disponibilidade e baixa latência são necessárias para aplicações críticas como supervisão de instalações industriais ou aplicações em saúde e mecanismos de anonimização de tráfego são importantes para proteger a privacidade das pessoas em um mundo cercado de vigilância eletrônica. Além disso o custo de operação precisa continuar diminuindo para que a Internet continue o processo de expansão.

Porém a Internet como conhecemos não tem condições de evoluir rápido o suficiente para acompanhar a demanda dos usuários e as inovações em termos de serviços. A Internet não foi projetada para evoluir [8, 9]. Por isso, novas arquiteturas foram propostas para permitir inovação e eliminar os problemas atuais da Internet. Computação em nuvem, Redes definidas por software Software Defined Networks (SDN), Network Function Virtualization (NFV), comunicações Machine to Machine (M2M) e Internet das coisas Internet of Things (IoT) são todas tecnologias com potencial transformador que estão sendo amadurecidas.

Para testar essas novas propostas e possibilitar uma transição segura para a Internet do futuro, é necessário um ambiente controlado onde experimentos sejam realizados. Um *testbed* é um ambiente que contém equipamentos, simuladores, instrumentação e outras

ferramentas necessárias para conduzir um teste [10]. A ideia é criar um *playground* onde vários pesquisadores põem à prova novas tecnologias enquanto medem seu desempenho. Por isso instituições ao redor do mundo estão juntado forças para criar laboratórios capazes de simular a Internet. Porém, a cooperação e orquestração de serviços hospedados ao redor do mundo e sob diferentes administrações não é uma tarefa simples. A cooperação torna-se possível através da federação de redes, que é o nome dado a dois ou mais domínios de rede independentes interconectados para a criação de um ambiente maior [11].

Além disso, *testbeds* foram originalmente criados para universidades, porém com o tempo, o serviço foi oferecido para empresas como uma forma de desenvolver novas tecnologias para a indústria e manter a infraestrutura. Entretanto, para que possa ser mantido como um serviço comercial, o vão entre protótipo e experimento precisa ser preenchido. Para satisfazer a alta disponibilidade exigida pela indústria, um amadurecimento da operação e uma melhor abordagem na correção de falhas deve ocorrer.

1.1 Motivação

O *testbed* Future Internet Brazilian Environment for Experimentation (FIBRE) é hoje a maior iniciativa para ambiente de teste em redes de computadores no Brasil, federado com a Europa e Estados Unidos. Devido à natureza pioneira do trabalho, a estrutura ainda está amadurecendo e necessita do desenvolvimento de novas ferramentas de monitoração.

A monitoração da infraestrutura de federações de recursos é de essencial importância para manter o serviço funcional. Contudo, em função da natureza distribuída dos recursos e da existência de várias entidades administradoras, a validação da disponibilidade dos recursos e dos serviços é complexa. No caso específico do FIBRE, manter a infraestrutura funcional passa não apenas pela verificação da disponibilidade dos recursos, mas pela disponibilidade de conjuntos mínimos de serviços. Para operadores de ilhas de recursos é de primordial importância detectar falhas com rapidez, assim como descobrir soluções com eficiência. Já para gerentes da federação de recursos é importante avaliar a disponibilidade de cada ilha e da federação, garantindo uma boa experiência de uso pelos usuários.

O desenvolvimento de uma ferramenta específica foi necessário para suprir a demanda de monitoração para uma infraestrutura heterogênea. O estado da arte em soluções de monitoração é focado principalmente em infraestruturas homogêneas administradas por uma única entidade. O uso de tais ferramentas limita as capacidades do *testbed* impedindo a inserção de dispositivos de rede que fujam ao padrão estabelecido. Portanto,

a possibilidade de reunir informações provenientes de diferentes dispositivos e diferentes soluções de monitoração torna o *testbed* mais flexível e facilita federação de novos conjuntos de recursos.

Nenhuma ferramenta de monitoração tradicional pôde suprir todas as necessidades de *testbeds* federados. Coletar e agregar medições de diversas ferramentas também é complexo, pois cada ferramenta tem diferentes arquiteturas, funcionalidades, uso e apresentação. É necessária uma adaptação dos dados dessas diversas fontes antes de agregá-los.

1.2 Objetivos

O primeiro objetivo deste trabalho é desenvolver um arcabouço de monitoração e gerência para o FIBRE, que leve em consideração a natureza heterogênea dos recursos e que seja aplicável a uma federação de recursos para experimentação. A proposta a ser desenvolvida deve avaliar o estado de recursos e serviços, gerando alarmes para operadores possibilitando a resolução rápida de problemas, além de permitir uma visão gerencial sobre a saúde das ilhas e sobre a experiência de uso pelos experimentadores. A ferramenta deve ainda prover meios gráficos para a observação do estado da federação de recursos, assim como para justificar problemas e agendar manutenções programadas.

Além disso, um levantamento do estado da arte e uma análise de problemas mais frequentes do *testbed* FIBRE é feita nessa dissertação. Diversos problemas e soluções são documentadas com auxílio dos operadores e do arcabouço desenvolvido.

1.3 Principais contribuições

São identificadas as seguintes contribuições:

- O levantamento do estado da arte em monitoração de *testbeds*.
- Uma proposta de análise da disponibilidade do *testbed* FIBRE.
- Uma interface gráfica para a visualização dos dados de monitoração.
- O desenvolvimento de um núcleo de agregação de informações de monitoração para o *testbed* FIBRE.
- O desenvolvimento de sondas de teste para o arcabouço de controle de experimentos OCF e OMF.

- A análise de falhas durante o um período de 45 dias de operação do FIBRE.
- Estatísticas de erro e teste do arcabouço de controle de experimentos OCF.

1.4 Organização do trabalho

Esta dissertação está estruturada em sete capítulos da seguinte forma: no capítulo 2 é apresentado um levantamento das principais iniciativas de *testbeds* de redes de computadores e seus sistemas de monitoração são apresentados. O Capítulo 3 apresenta as funcionalidades, características e arquitetura do *testbed* FIBRE. No Capítulo 4 os sistemas de monitoração de *testbeds* mais relevantes para o presente estudo são apresentados em detalhes. O sistema de monitoração FIBREOSS é apresentado no Capítulo 5. Sua especificação, arquitetura e funcionalidades são descritas. Então, o Capítulo 6 mostra os resultados da monitoração do *testbed* FIBRE no período de 17 de setembro a 1º de outubro de 2016. Além disso, é feita uma análise da escalabilidade e descrição do impacto do sistema de monitoração. Por fim, o Capítulo 7 conclui a dissertação.

Capítulo 2

Testbeds e o futuro da Internet

Para possibilitar a evolução desejada para a Internet, tornou-se necessária a existência de um *playground* onde pesquisadores e estudantes pudessem entender e experimentar novas aplicações e arquiteturas. Esse era um novo passo na transição de uma pesquisa acadêmica baseada somente em modelos matemáticos ou simulações para uma pesquisa que inclui implementações reais como complementos às simulações.O projeto *Clean Slate Program* foi criado para tornar-se a face da Internet do futuro. Seu primeiro *testbed* ficou conhecido como Global Environment for Network Innovations (GENI), um laboratório virtual distribuído nos Estados Unidos [12]. Desde então, várias iniciativas ao redor do mundo sugiram para criar um laboratório distribuído onde experimentos pudessem coexistir com tráfego de uma rede de produção. Esta seção mostra os componentes básicos de um *testbed* e as principais iniciativas ao redor do mundo fazendo uma breve descrição de suas principais características.

A Figura 2.1 mostra os blocos de construção de um *testbed* genérico federado à outras instituições de pesquisa e com suporte à experimentação de múltiplos usuários. Esse diagrama de componentes essenciais foi estabelecido a partir da observação de diversos *testbeds* de redes de computadores que serão mostrados nesse capítulo. Nessa figura, destacam-se:

- Interface com o usuário - Possibilita que o usuário utilize o serviço oferecido pelo *testbed*.
- Agendamento - Responsável pela divisão de recursos entre os usuários, especialmente quando os mesmos não podem ser compartilhados para evitar interferências.
- Gestão de identidade – Permite identificar usuários, recursos e serviços, além de validar todo o controle de acesso. Em ambientes federados, é importante que haja

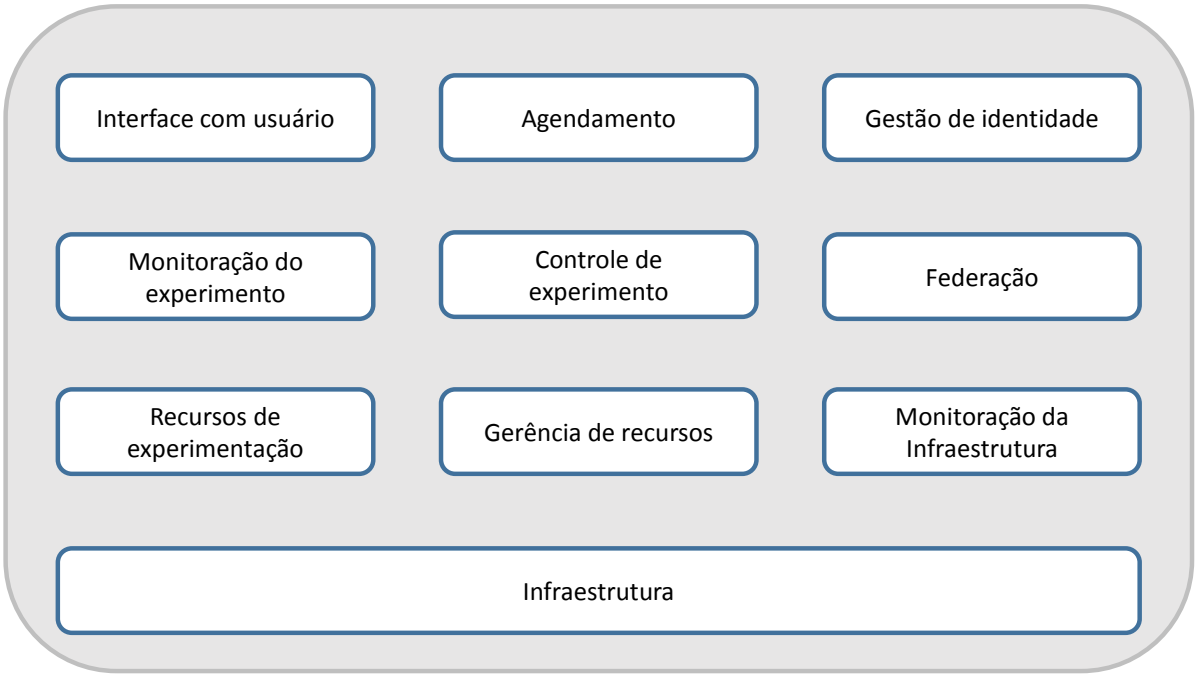


Figura 2.1: Blocos de construção de um testbed genérico.

integração entre as plataformas envolvidas de modo que o usuário não precise realizar *login* várias vezes ou criar diversos cadastros.

- Monitoração do experimento - Monitora e disponibiliza os resultados dos experimentos do usuário.
- Controle de experimento - Necessário para garantir a repetibilidade e boa execução do experimento.
- Federação - Responsável pela integração de recursos de entidades administrativas distintas.
- Recursos de experimentação - Recursos que podem ser agendados e utilizados pelo usuário para a realização dos testes.
- Gerência de recursos - Responsável pela alocação e configuração dos recursos de experimentação.
- Monitoração da infraestrutura - Fornece aos administradores do *testbed* as informações necessárias para efetuar melhorias e reparos no sistema.
- Infraestrutura de controle - Diferente dos recursos de experimentação, a infraestrutura de controle só está disponível para administradores do *testbed* e é composta por todas as máquinas e serviços necessários para o controle do testbed.

Outros termos utilizados neste capítulo para referir-se a componentes são: agregados, *slices* e *slivers*. Agregados são conjuntos de componentes administrados por uma organização. Um *slice* é uma estrutura virtual que representa um conjunto de *slivers*, que por sua vez são instâncias virtuais ou partes de um recurso. Os *slices* são oferecidos aos experimentadores através de um arcabouço de controle que dá suporte aos serviços federados.

O objeto de estudo deste trabalho tem foco no bloco de monitoração da infraestrutura.

2.1 Principais iniciativas de testbeds de redes de computadores

Tabela 2.1: Linha do tempo: Início dos principais projetos de *testbeds* voltados para o desenvolvimento da Internet.

2003	Open-Access Research Testbed for Next-Generation Wireless Networks (ORBIT).
2003	PlanetLab.
2006	Global Environment for Network Innovations (GENI).
2008	PlanetLab Europe (PLE).
2008	G-Lab.
2008	Federated E-infrastructure Dedicated to European Researchers Innovating in Computing network Architectures (FEDERICA).
2009	Network Implementation Laboratory (NITOS-NITLab).
2010	Future Internet Research and Experimentation (FIRE).
2010	OpenFlow in Europe: Linking Infrastructure and Applications (OFELIA).
2011	Future Internet Brazilian Environment for Experimentation (FIBRE).
2011	Building service testbeds for Future Internet Research and Experimentation (BonFIRE).

A Tabela 2.1 é uma lista que abrange as maiores iniciativas para criação de *testbeds* para experimentação em redes de computadores. Uma lista com 174 projetos ativos e

finalizados encontra-se em [13]. A seguir, serão brevemente descritos os principais *testbeds* de redes de computadores da atualidade.

2.2 GENI

Global Environment for Network Innovations (GENI) [14] é uma das primeiras iniciativas[1] para formação um laboratório de pesquisas em Internet do futuro e o maior em escala. Localizado nos Estados Unidos da América e patrocinado pela National Science Foundation (NSF), o projeto tem como objetivo o desenvolvimento e validação de novas tecnologias que resolvam as preocupações existentes com a ossificação da Internet [15].

As funcionalidades para o experimentador providas pelo GENI incluem:

- *Sliceability* - Experimentos podem ser virtualizados através do conceito de *slices*, que o GENI pega emprestado do *testbed PlanetLab*, suportando assim múltiplos experimentos simultâneos.
- Programabilidade profunda - Conceito fundamental como estratégia contra a ossificação. Um experimentador tem a habilidade de controlar o comportamento de sistemas de computação, armazenamento, roteamento e encaminhamento de rede, e não somente de dispositivos na ponta da rede, possibilitando flexibilidade para o surgimento de inovação.
- Usuários voluntários - Existe a opção de ter pessoas não ligadas ao estudo utilizando os serviços prestados pelo experimento quando a interação do usuário é necessária para gerar dados relevantes para a pesquisa. Desde que sejam atendidos requisitos de segurança e privacidade pré-estabelecidos.

2.2.1 Monitoração da infraestrutura

- GENI Meta Operation Center (GMOC)

  O centro de meta operações do GENI, é responsável por prover os processos necessários para trazer consistência e repetibilidade aos experimentos usando os recursos da infraestrutura federada do GENI [16]. Para a monitoração da infraestrutura, o Network Operations Center (NOC) do GENI monitora a infraestrutura genérica

[1] Primeiros Project Execution Plans (PEPs) em Janeiro de 2006.

do testbed utilizando dados coletados com Simple Network Management Protocol (SNMP) [17].

2.2.2 Monitoração dos experimentos no GENI

O projeto GEMINI [18] tem em seu escopo de monitoração todo o ciclo de vida de experimentos. Foi criado para o arcabouço de controle ProtoGENI, sendo o resultado da agregação de dois projetos anteriores Instrumentation Tools for a GENI Prototype (INSTOOLS) e Leveraging and Abstracting Measurements with perfSONAR (LAMP). O INSTOOLS era focado em medições passivas e o LAMP era focado em medições ativas. Assim, os projetos foram vistos como soluções complementares.

2.2.2.1 INSTOOLS

Para suprir as necessidades de monitoração das redes virtuais criadas pelos usuários com propósito de experimentação, foi criado o projeto Instrumentation Tools for a GENI Prototype (INSTOOLS). Este sistema de monitoração é feito para capturar, gravar e mostrar informações de um *slice* [2]. A ideia é disponibilizar um conjunto de ferramentas prontas para monitorar o experimento que seja de fácil uso.

O conjunto de ferramentas inclui tabelas de roteamento, gráficos de uso de recursos, tráfego no elemento de rede e gráficos para protocolos específicos [17]. Para realizar a monitoração, o método utilizado foi a inserção de um Measurement Controller (MC) em cada agregado do *slice* que o usuário deseja monitorar. Esses medidores usam Secure Shell (SSH) e Simple Network Management Protocol (SNMP) para coletar dados dos Measurement Points (MPs) e enviam os dados para um servidor centralizado onde essas medidas são armazenadas. [19, 20]

2.2.2.2 LAMP

O sistema Leveraging and Abstracting Measurements with perfSONAR (LAMP) [21] foi construído baseado no arcabouço de medição do perfSONAR e oferece medidas coletadas com as principais ferramentas do mesmo. Os serviços do perfSONAR foram adaptados para funcionar junto à infraestrutura de experimentação do GENI como um sistema centralizado.

[2] rede virtual disponível ao usuário

Sua arquitetura envolve três componentes: o MP, um portal *web* e o Unified Network Information Services (UNIS) [22], onde os Measurement Points (MPs) são elementos de rede executando os serviços do perfSONAR. O portal *web* disponibiliza as medições e permite que configurações sejam feitas pelo usuário. E o UNIS armazena e disponibiliza todos os metadados das medições e topologias.

2.2.3 Monitoração completa

2.2.3.1 GIMI

O projeto Large-scale GENI Instrumentation and Measurement Infrastructure (GIMI) [23] foi desenvolvido para o *testbed* Open Resource Control Architecture (ORCA) [24] do GENI. O ORCA é um arcabouço de controle para experimentos federados em redes heterogêneas do *testbed* ExoGENI, [25] integrado ao GENI. Seu objetivo é ser uma solução de monitoração completa para operadores e usuários do *testbed*.

Seus principais objetivos são:

- Fornecer serviços de instrumentação fáceis de usar para os experimentadores.
- Capacidade de medidas gerais para operadores do testbed.
- Construir e operar um arquivo de medições abrangente que possa ser utilizado pelo GIMI e pelo GEMINI.

O projeto utiliza o Orbit Measurement Library (OML) [26] como arcabouço para coleta de medições e a interface do Internet Remote Emulation Experiment Laboratory (IREEL) como interface de visualização. Este portal permite que estudantes conduzam experimentos de rede em tempo real e visualizem medições do mesmo [27].

2.2.4 PlanetLab

O PlanetLab é um rede global para experimentação em redes de computadores. Atualmente com 1353 dispositivos instalados em 717 *sites* onde múltiplos experimentos são executados de maneira concorrente.[28]

Seus principais objetivos são:

- Prover uma plataforma de pesquisa para experimentação com serviços de rede em escala planetária.

- Prover uma plataforma para que novos serviços de redes sejam testados por uma comunidade de usuários reais.
- Catalizar a evolução da Internet para uma arquitetura orientada à serviços.

O PlanetLab tornou-se uma ferramenta de pesquisa importante em redes de computadores e sistemas distribuídos. Sua bibliografia inclui cerca de 200 citações com a vasta maioria publicada no SIGCOMM, SOSP, OSDI, NSDI, INFOCOMM, Usenix, EuroSys, IPTPS, IMC, and HotNets.[29, 30]

2.2.5 ORBIT

Fundado em 2003, o Open-Access Research Testbed for Next-Generation Wireless Networks (ORBIT) [31] é um sistema de teste de campo e emulador de redes sem-fio projetado para realizar experimentos reprodutíveis enquanto proporciona uma avaliação realística de protocolos e aplicações. Sua principal instalação é o *RADIO GRID TESTBED*, um arranjo programável de 20x20 nós sem-fio que podem ser interconectados em topologias específicas e com modelos de canais de rádio reprodutíveis. Além disso, uma instalação externa fornece um conjunto configurável de tecnologias Worldwide Interoperability for Microwave Access (WiMAX), Long Term Evolution (LTE)[3] e 802.11[4] em um ambiente real. [31]

Para gerenciar a infraestrutura de experimentação do laboratório, a equipe do Winlab na universidade de Rutgers desenvolveu o OMF [32], que posteriormente foi adotado em outras plataformas de teste ao redor do mundo como o NITOS [33], FIBRE [34] e GENI [35].

2.3 FIRE

O Future Internet Research and Experimentation (FIRE) [36] foi criado em 2010[5] pela Comissão Européia para ser o principal laboratório de pesquisa e desenvolvimento de tecnologias para Internet do futuro na Europa. Financiado pelo Seventh Framework Programme (FP7), tem como objetivo criar um ambiente aberto de pesquisa e desenvolvimento dentro do contexto de Internet do futuro para estudar novas ideias e conceitos através de experimentos de larga escala com novos paradigmas, arquiteturas e conceitos

[3] WiMAX e LTE. São tecnologias de 4ª geração em telefonia móvel
[4] Comercialmente conhecido como WiFi
[5] Por mais que haja uma contradição aparente, já que a referência para este parágrafo é de 2007, o *website* oficial diz:"FIRE [...] has been growing since its inception in 2010."

de rede [37, 38]. Assim como o GENI, o projeto FIRE é um *testbed* que incorporou diversos outros *testbeds* sob sua administração. Cada um com suas próprias particularidades e ferramentas de suporte à operação.

Outros projetos sob a administração do FIRE no escopo de pesquisa de Internet do Futuro incluem: [39]

- Community Networks Testbed for the Future Internet (CONFINE) [40]
- Cognitive Radio Experimentation World (CREW) [41]
- Experimedia [42]
- OpenLab [43]
- BonFIRE [44]
- OFELIA [45]
- SmartSantander [46]
- OneLab [47]
- Panlab II [48]
- FEDERICA [49]
- TEFIS [50]

2.3.1 Monitoração no FIRE

Uma das propostas de monitoração para o FIRE é o MOST4FIRE, o qual é um sistema voltado para a monitoração e medição de recursos em *testbeds* federados em redes heterogêneas. O sistema inclui em seu escopo de monitoração vários níveis da infraestrutura, desde a infraestrutura física, a infraestrutura virtual e o nível de serviço e aplicação. Os artigos [51, 52] também descrevem a capacidade realizar monitoração utilizando três Application Programming Interface (API)s: Uma Infra API, que interage com os *testbeds*, uma API de usuário, para interagir com os usuários finais e a Inter API, que permite a interoperabilidade com outros sistemas de monitoração os outras instâncias do MOST4FIRE.

2.3.2 BonFIRE

O projeto Building service testbeds for Future Internet Research and Experimentation (BonFIRE) [44] foi fundado pela European Commission (EC) FP7 em uma chamada pública a fundação BonFIRE foi criada em dezembro de 2013 para operar as instalações de teste do projeto BonFIRE que oferece Infrastructure As A Service (IaaS) para propósito de pesquisa e desenvolvimento voltada para comunidade de Internet of Services (IoS).

O serviço de nuvem, sua plataforma, infraestrutura e modelo de arquitetura em camadas são os pontos fortes do projeto. O *testbed* é disponibilizado usando a entrega de iaas). Sua metodologia fácil de usar, ferramentas e serviços suportam uma federação em nuvem, gerenciamento de máquina virtual, modelagem de serviços, gerenciamento de ciclo de vida de experimentos, monitoramento de qualidade e análise de serviço. [53] [54]

Funcionalidades para o experimentador:

- Gerenciamento de recursos
- Monitoração de infraestruturas Físicas e Virtuais
- Descrição de experimentos em um único documento
- Agendamento de uso de recursos

Para monitoração da infraestrutura, o BonFIRE utiliza o Zabbix. Ele foi lançado em 2001 e uma empresa foi criada em 2005 para fornecer suporte técnico e serviços. Zabbix é um sistema de monitoramento nível *enterprise* desenvolvido para monitorar disponibilidade e desempenho de componentes de uma infraestrutura de tecnologia da informação. Além disso, é um *software* de código aberto. [55]

Segundo a página oficial, é possível monitorar infraestruturas físicas e virtuais com alto desempenho e escalabilidade e com múltiplas opções de visualização de dados como mapas, gráficos e resumos.

2.3.3 OFELIA

O OpenFlow in Europe: Linking Infrastructure and Applications (OFELIA) [45] foi um projeto parte do programa Information and Communication Technology (ICT) FP7, da União Européia[6] cujo objetivo era criar um ambiente de experimentação inovador onde

[6] Agora sob administração do FIRE

os usuários podem não só experimentar em uma rede de teste como também controlar a rede em si de forma precisa e dinâmica. O projeto resultou no testbed OFELIA, no qual, para realizar seus experimentos, utiliza-se SDN, mais especificamente o arcabouço OpenFlow [56, 57]. O projeto OFELIA desenvolveu o arcabouço de controle OCF que é utilizado no projeto FIBRE e será visto em detalhes na Seção 3.3

Para monitoração dos experimentos, o OFELIA propõe algumas ferramentas, sendo elas o Cbench e o OFLOPS.

O Cbench [58] é uma ferramenta de *benchmark* para controladores OpenFlow. Ela emula requisições dos *switches*, enviando-as para o controlador de forma a avaliar o seu desempenho. Com o Cbench, é possível medir a latência de instalação do fluxo e o número de fluxos que ele é capaz de instalar por segundo. [59]

O OFLOPS [60] é um arcabouço de teste de desempenho para *switches* OpenFlow. Ele é construído como um controlador individual que executa de forma modular múltiplos testes indiferentes à implementação do *switch*. [61]

2.3.4 FEDERICA

Fundado em 2008, Federated E-infrastructure Dedicated to European Researchers Innovating in Computing network Architectures (FEDERICA) [49] é um projeto Europeu que fornece uma infraestrutura 'agnóstica a tecnologia' baseada em circuitos *Gigabit Ethernet*, equipamentos de transmissão e nós com capacidade de virtualização para abrigar atividades experimentais de novas arquiteturas e protocolos para a Internet. [62]

Funcionalidades para o experimentador:

- O usuário tem controle total sobre os elementos virtuais e links. Também pode instalar qualquer sistema operacional e protocolos de rede nos nós e configurar como desejar os roteadores e switches virtuais.
- O Network Operations Center (NOC) pode configurar um recurso virtual para apresentar um comportamento replicável
- A arquitetura pode ser integrada via federação.

Para monitoração de infraestrutura, o FEDERICA utiliza dos sistemas, G3 e perfSONAR.

G3 [63] é um sistema desenvolvido pelo CESNET [64] para monitorar continuamente infraestruturas de rede de larga escala. Seus mecanismos de medição e processamento de dados permitem a visualização da dinâmica da rede. Também possui mecanismos de processamento de dados que garantem adaptabilidade automatizada para reconfiguração de dispositivos e mantém um mapeamento estrutural dos dispositivos medidos. Sua interface de usuário agrega dados enquanto consulta o armazenamento.

A interface de usuário é utilizada por administradores de rede. Ela utiliza SNMP e permite uma monitoração em conjunto com uma ferramenta externa de relatório em uma página *web* acessível a usuários finais. Algumas das métricas disponíveis são: Uso do backbone IP/Multi Protocol Label Switching (MPLS) como capacidade do canal, taxa de transferência e utilização do canal. [65]

O perfSONAR [66] é um kit de medição para redes feito para incluir medições em infraestruturas federadas e ajudar a estabelecer as expectativas de fim-a-fim do usuários. Foi desenvolvido a partir de uma colaboração internacional liderada pelas instituições: Internet2 [67], ESnet [68], Universidade de Indiana [69] e GEANT [70].

O PerfSONAR também é utilizado pelo FIBRE e será visto em mais detalhes na Seção 3.6

2.3.5 *OneLab*

O Onelab oferece uma ampla gama de serviços e recursos serviços com foco principal em Internet do Futuro. Combina muitos *testbeds* de uso especializado e também dois grandes como o PlanetLab Europe (PLE) e o *testbed* NITOS, que é voltado para redes sem-fio. O software que controla sua infraestrutura é livre e de código aberto e o PLE funciona em um sistema de federação que permite compartilhamento de recursos com outra instituições [71].

Dentro desse contexto, a equipe do Onelab propôs o TopHat, que foi desenvolvido para ser o sistema de medição ativa do PlanetLab Europe (PLE), que também é parte do Onelab. O TopHat Usa como parte de sua infraestrutura de medição o European Traffic Observatory Measurement Infrastructure (ETOMIC) [72, 73] e DIMES [74, 75] para medições especializadas. Seu objetivo é dar suporte ao ciclo de vida completo do experimento. Na fase de instalação, o sistema auxilia o usuário a escolher os elementos de rede de acordo com métricas como atraso e número de saltos. Quando o experimento está ativo, ele permite que o usuário monitore o desempenho do experimento, faça medições

de acordo com sua necessidade e faça adaptações no experimento. Uma vez acabado o experimento, o TopHat provê acesso e visualização para dados de monitoração arquivados [76]. O TopHat dá suporte somente à monitoração do experimento enquanto agrega informações de monitoração da infraestrutura do *testbed* obtidas pelo CoMon [77].

Ainda nessa linha de monitoração do PlanetLab Europe, cabe observar que muitos serviços de monitoração do foram propostos, mas acabaram descontinuados [78]. Seguem alguns exemplos:

- CoMon - O CoMon [79] é um sistema que fornece estatísticas de monitoração para o PlanetLab em nível de elemento de rede e nível de *slice*. Pode ser usado para visualizar as causas de problemas de desempenho em elementos de rede e examinar os perfis de uso de recursos dos usuários. Sua página de *status* fornece múltiplas visualizações do *testbed* incluindo visualizações centradas em elementos de rede, centradas em *slice*, etc.

  O sistema encontra-se desativado devido a crescente dificuldade de manutenção.

- CoTop - Trata-se de uma ferramenta similar ao top do Linux, porém mostrando o consumo de recurso de *slices*.

- Ganglia – É um sistema de monitoração distribuído que mostra informações detalhadas de uso de CPU, memória e métricas de rede em todos os elementos de rede. Todas as métricas numéricas são agregadas em cada nível da árvore de monitoração. Históricos também são salvos para visualizar tendências.

- MapCenter - Mostra o estado corrente de conjuntos de computadores. Mostra um mapa geográfico da disponibilidade dos elementos de rede.

- SliceStat - Servidor que mostra o consumo de recursos por *slice* em nós individuais no PlanetLab.

2.3.6 NITOS-NITLab

NITOS-NITLab é um *testbed* com 50 nós sem-fio WiFi instalados em um ambiente externo no campus da universidade de Thessaly. É remotamente e publicamente acessível para qualquer pesquisador que deseje utilizar seus recursos após registro e aprovação pelos administradores. Seus recursos são controlados utilizando Orbit Management Framework (OMF) para controle, Orbit Measurement Library (OML) para monitoração de experimentos e *NITOS Scheduler* para agendamento de uso dos recursos. [33] [80]

NITOS desenvolveu os nós Ícarus, que são nós sem fio Wifi com discos rígidos de estado sólido e processador Intel com 4 núcleos. Esses nós sem fio são utilizados no projeto FIBREe serão explicados em mais detalhes na seção 3.7 [81]

2.4 G-Lab

G-Lab [82] foi criado em 2008 e é um projeto Alemão com o objetivo de desenvolver uma Internet segura e confiável. A missão do G-Lab é de manter uma estrutura onde todas as crescentes demandas por novas arquiteturas, protocolos e outros estudos da Internet do futuro possam ser testados. O G-lab possui uma rede com cerca de 170 elementos e alguns de seus projetos mais importantes são: o *Future Internet Lab*, que possui 60 elementos de alto desempenho e interfaces de 400 Gigabit com topologias virtuais configuráveis e o Virtual Routers Architecture, Management and Applications (VirtuRAMA), que trabalha em projetos de virtualização de enlaces de rede e roteadores. [83]

O sistema utilizado para realizar a monitoração de toda a infraestrutura é o Nagios. Um servidor virtual dedicado é responsável pela coleta de dados de monitoração de elementos de rede individuais e serviços e notificar administradores por e-mail quando houver problemas.

Informações monitoradas incluem: Uso de CPU, memória, disco, etc. As informações são coletadas de todas as máquinas virtuais. Estado de saúde de todos os elementos de rede e disponibilidade de todos os elementos de rede e processadores de serviço. [84]

Capítulo 3

Projeto FIBRE

O *testbed* Future Internet Brazilian Environment for Experimentation (FIBRE) foi criado em 2010 como parte de um acordo de cooperação internacional entre Brasil e União Européia. Na época o acrônimo significava *Future Internet testbeds experimentation between Brazil and Europe*, porém quando o projeto terminou em 2014 o governo Brasileiro ficou encarregado pela infraestrutura e o acrônimo mudou para *Future Internet Brazilian Environment for Experimentation*. Hoje, é a principal iniciativa no Brasil para um *testbed* de rede aberto. [85, 86]

O *testbed* funciona como um laboratório aberto para estudantes e pesquisadores testarem novas aplicações e modelos de arquitetura de rede. Ele possui múltiplas ilhas montadas em universidades e centros de pesquisa pelo Brasil e conexões com parceiros no exterior. A Figura 3.1 mostra as ilhas do FIBRE implementadas no Brasil, onde as siglas representam instituições que abrigam ilhas:

- Universidade Federal do Pará (UFPA)
- Universidade Federal de Pernambuco (UFPE)
- Universidade de Salvador (UNIFACS)
- Rede de Nacional de ensino e Pesquisa (RNP)
- Universidade Federal de Goiás (UFG)
- Universidade Federal de Minas Gerais (UFMG)
- Universidade Federal de Uberlândia (UFU)
- Universidade Federal do Espírito Santo (UFES)

- Universidade Federal Fluminense (UFF)
- Universidade Federal do Rio de Janeiro (UFRJ)
- Universidade Federal de São Carlos (UFSCar)
- Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPqD)
- Universidade federal de São Paulo (USP)
- Universidade Federal do Rio Grande do Sul (UFRGS)

Figura 3.1: Distribuição geográfica das ilhas do FIBRE. [1]

3.1 Componentes e arquitetura

A Figura 3.2 mostra os componentes de uma ilha típica do FIBRE. Dois domínios de experimentação foram estabelecidos: Sem-fio e Openflow. O domínio sem-fio é feito seguindo a organização do *testbed* NITOS, que foi apresentado na Seção 2.3.6. Esse domínio utiliza o sistema de controle Orbit Management Framework (OMF), nós sem-fio Ícarus e diversos

recursos sem-fio que os administradores de ilhas adicionaram em suas plataformas. O domínio Openflow em cada ilha do FIBRE consiste em três máquinas com placas Networking Field-Programmable Gate Array (NetFPGA) configuradas como *switches* Openflow, um *switch* Openflow puro e máquinas virtuais criadas para o experimentador e conectadas aos *switches*. As ilhas locais são interconectadas pela FIBREnet, que foi construída sobre o *backbone* da Rede de Nacional de ensino e Pesquisa (RNP).

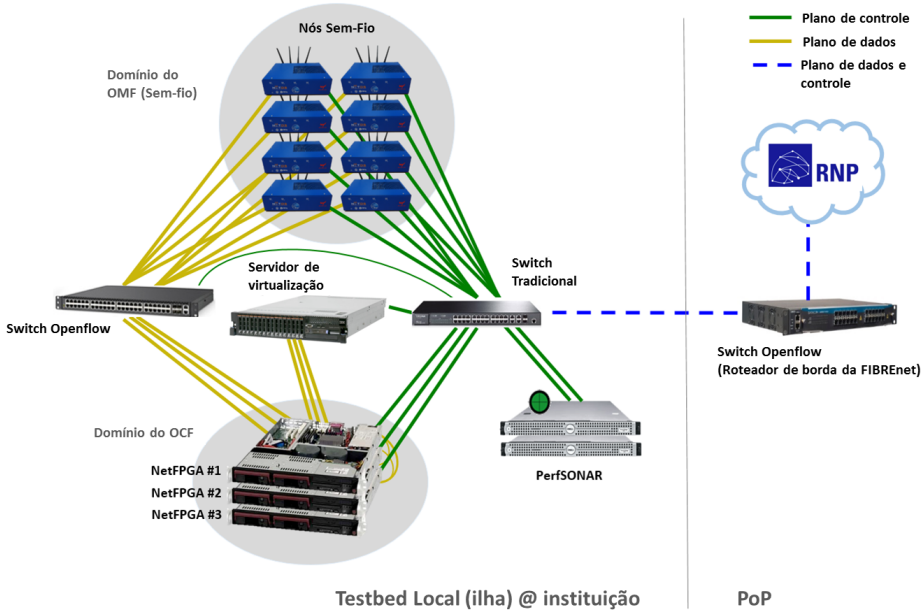


Figura 3.2: Componentes de uma ilha do FIBRE-BR. Adaptado de [2].

A Figura 3.4 mostra a arquitetura do FIBRE divida em três pilhas: recursos OpenFlow, máquinas virtuais e nós sem fio. Blocos marcados com a cor azul fazem parte do domínio OpenFlow, blocos marcados com a cor verde, do domínio sem-fio e blocos brancos são neutros, onde:

- Expedient - Permite que o usuário gerencie os recursos OpenFlow de seu *slice*;
- MySlice - Portal genérico que permite o acesso do usuário a diversos recursos de um *testbed* federado [87];
- Portal Orbit Management Framework (OMF) - Permite que o usuário gerencie os recursos sem-fio, alocando recursos para experimentos;
- Optin OpenFlow Aggregate Manager (OFAM) - Gerenciador de Agregados de recursos OpenFlow;

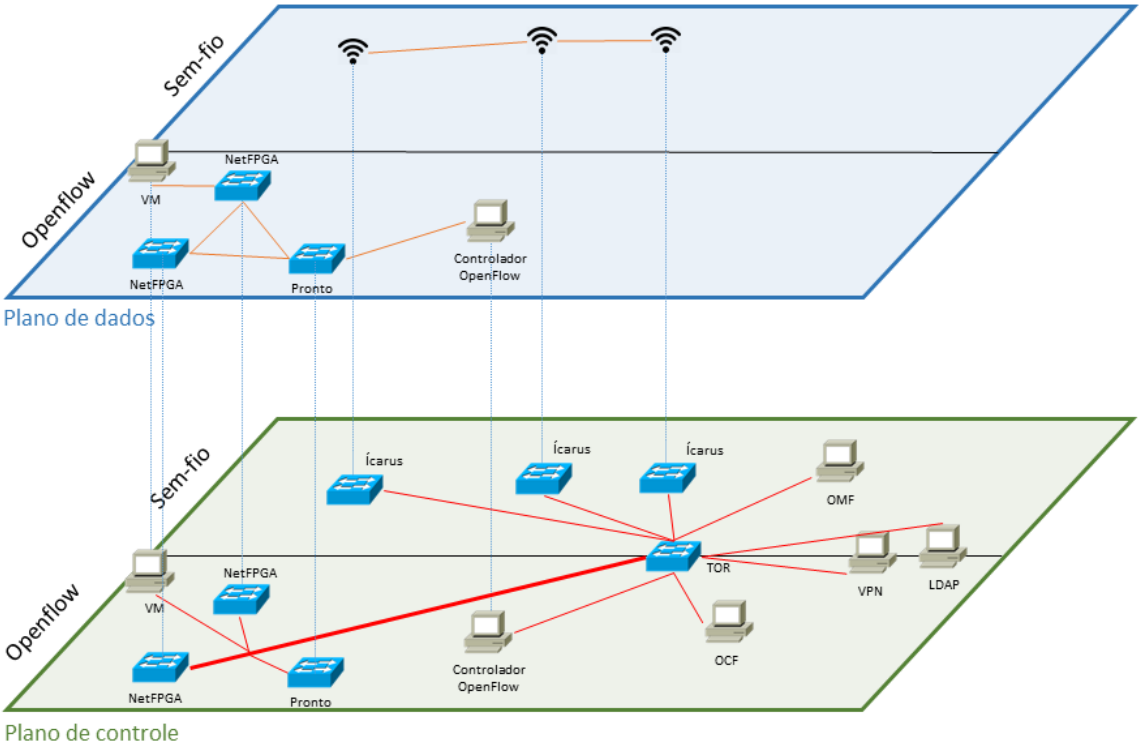


Figura 3.3: Separação de plano de dados e controle em uma ilha do FIBRE.

- Virtualization Aggregate Manager (VTAM) - Gerenciador de Agregados de máquinas virtuais;
- Labora *Scheduler* - Sistema de agendamento de uso de recursos sem-fio ;
- OMF Resource Controller (RC) - Controla os recursos do OMF;
- Flowvisor - Proxy entre *switches* OpenFlow e controladores, permitindo a coexistência simultânea de múltiplos controladores OpenFlow, desde que cada um esteja em um espaço de fluxos diferente [88];
- OFELIA Xen Agent (OXA) - Agente do OCF responsável pela criação e supervisão das máquinas virtuais [89];
- Xen - Tecnologia de virtualização utilizada do FIBRE;
- *Switches* OpenFlow - Disponíveis ao usuário para experimentação;
- Máquinas Virtuais - Máquinas criadas pelo usuário usando a interface do Expedient;
- Nós Ícarus - Pontos de acesso WiFi disponíveis para experimentação.

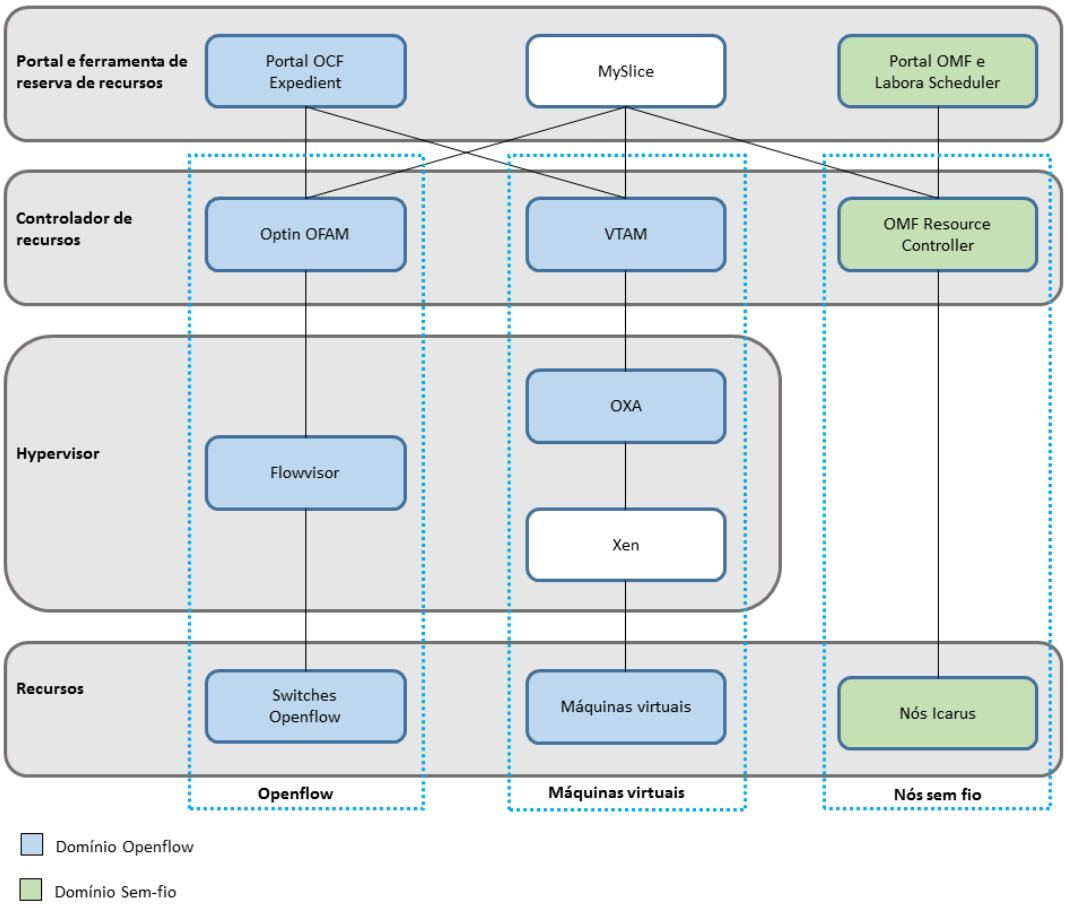


Figura 3.4: Arquitetura do FIBRE. Adaptado de [1, 3].

A Tabela 3.1 faz a correlação dos módulos do modelo genérico de *testbed* apresentado na Figura 2.1 com os módulos do FIBRE.

Tabela 3.1: Componentes do FIBRE.

Módulo	Domínio OpenFlow	Domínio sem fio	Controle do *testbed*
Interface com o usuário	Expedient e MySlice	Portal do OMF e MySlice	-
Agendamento	-	Labora *Scheduler*	-
Gestão de identidade	LDAP e CAFe	LDAP e CAFe	-
Monitoração do experimento	-	OML	-
Controle de experimento	-	OMF EC	-
Federação	SFA	SFA	-
Recursos de experimentação	*Switches* OpenFlow, NetFPGAs e VMs	Nós Ícarus, Mini ITX, AP sob trilho	-
Gerência de recursos	Optin OFAM, Flowvisor, VTAM, OXA	OMF RC	-
Monitoração da infraestrutura	-	-	ZenOSS e PerfSONAR
Infraestrutura	-	-	Xen, switches e máquinas do PerfSONAR

Onde:

- o Lightweight Directory Access Protocol (LDAP) e a Comunidade Acadêmica Federada (CAFe) são usados para manter uma base central de usuários;
- o Orbit Measurement Library (OML) auxilia na coleta das medições dos experimentos no OMF;
- o Slice-based Federation Architecture (SFA) é o esquema de federação utilizado em diversos *testbeds* ao redor do mundo;
- a Mini ITX é um computador miniatura com pontos de acesso sem fio;

- Access Point (AP) sob trilho - Um computador com ponto de acesso e câmeras montado sob um trilho em um corredor de 60m.

Como foi mostrado, a infraestrutura atual do FIBRE suporta dois domínios de experimentos bem distintos e separados, sendo eles o OpenFlow e o sem-fio. Como consequência, o FIBRE usa dois arcabouços de controle e gerência, sendo um para cada domínio: OFELIA Control Framework (OCF) e Orbit Management Framework (OMF), respectivamente.

3.2 OMF

O Orbit Management Framework (OMF) é um arcabouços de controle, medida e gerência para plataformas experimentais (*testbeds*). Esse arcabouço permite a execução de experimentos remotamente e recolhe resultados de diferentes tipos de dispositivos [90]. Para recolher resultados, o OMF disponibiliza uma biblioteca chamada Orbit Measurement Library (OML), que pode ser utilizada para coleta de dados de diferentes variáveis com um período de coleta especificado em código pelo experimentador. Os dados recolhidos são armazenados na base de dados local da ilha onde o experimento está sendo executado.

Embora possa ser usado para o controle e gerência de equipamentos de *testbeds* genéricos, o OMF, em sua versão 5.4, é usado no FIBRE apenas para controlar recursos para experimentos com redes sem-fio. Os recursos de rede OpenFlow cabeada são gerenciados pelo arcabouço OCF, que será discutido em seguida.

Originalmente, o OMF disponibiliza uma interface *web* para reserva de recursos, mas optou-se por desenvolver um novo portal para o FIBRE, o qual tem um escalonador chamado de Labora Scheduler - WEB (LS-WEB) [91]. Cada ilha possui um portal local e o *testbed* federado é acessado pelo portal federado, que reúne os recursos de diferentes ilhas.

Toda a comunicação no OMF é feita através do modelo Publish/Subscribe (PubSub), utilizando o eXtensible Messaging and Presence Protocol (XMPP) (Openfire) como servidor de comunicação. Um experimento usa o controlador de experimentos (Experiment Controller (EC)) para executar o *script* de experimento e configurar os recursos, de acordo com o esquema mostrado na Figura 3.5. O controlador de experimentos no *testbed* do FIBRE é a máquina virtual *omf-console* instalada na ilha [92]. Assim, cada ilha possui o seu próprio EC, assim como o seu próprio portal.

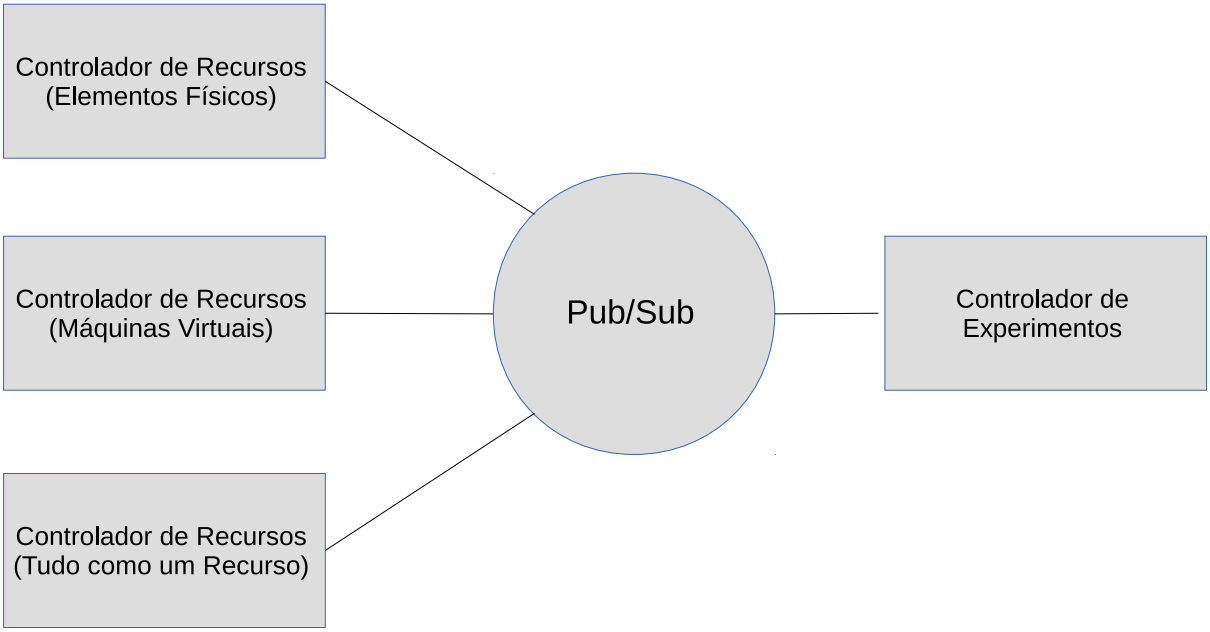


Figura 3.5: Esquema de comunicação no OMF.

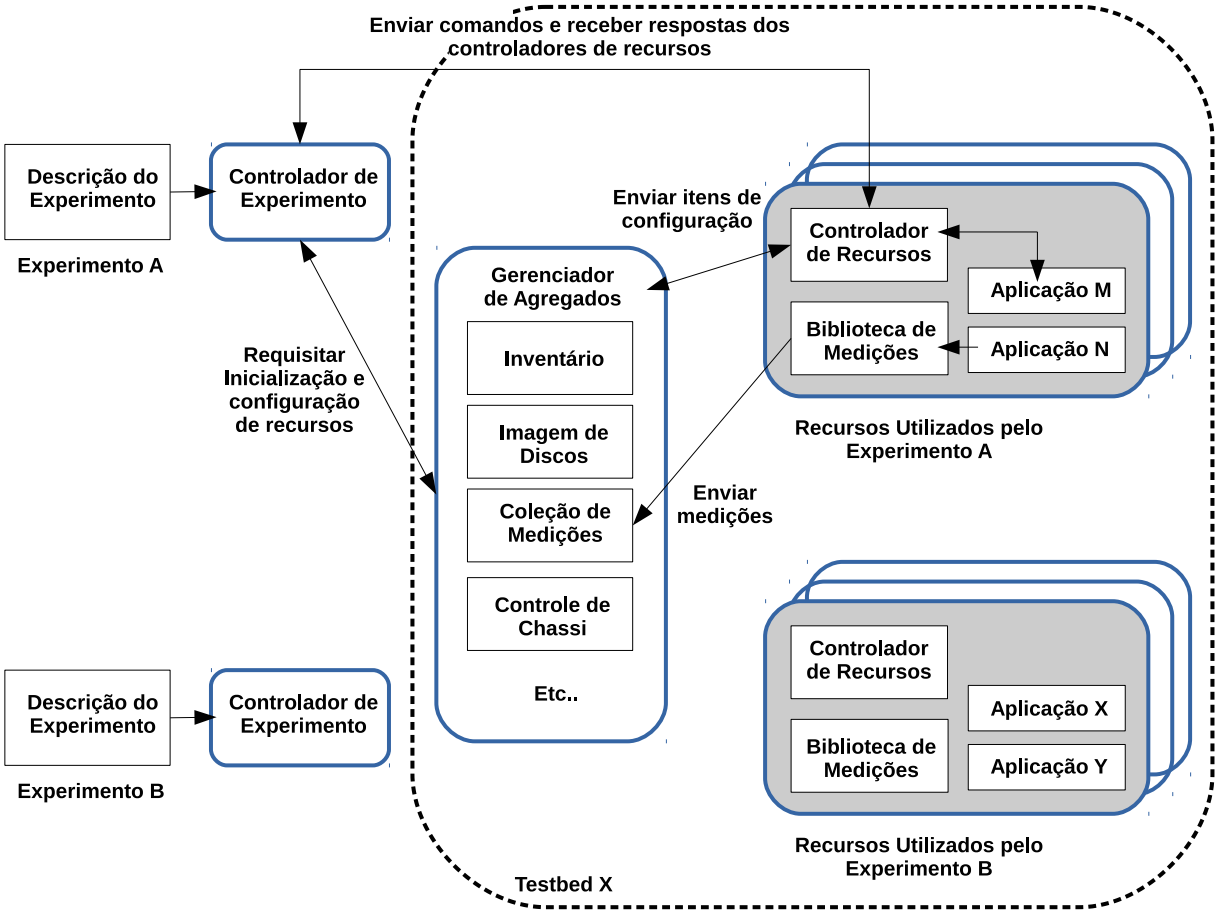


Figura 3.6: Arquitetura e fluxograma de uso do OMF.

A Figura 3.6 mostra a arquitetura do OMF e o fluxograma de uso do *testbed* pelo

usuário. O experimentador passa uma descrição de experimento ao controlador de experimento. O controlador de experimento interpreta a descrição e envia requisições para o gerenciador de agregado, iniciando e configurando os recursos pedidos pelo experimentador. Quando os recursos estiverem prontos, o controlador de experimentos envia comandos para os controladores de recursos. A partir desse momento, o experimento está em execução [93].

3.3 OCF

O OFELIA Control Framework (OCF) é o sistema de controle e gerência desenvolvido pelo projeto OFELIA. Seu objetivo é controlar *testbeds* para experimentos com OpenFlow. A Figura 3.7 mostra a arquitetura do OCF e as interfaces de comunicação entre os módulos. Seus principais componentes são:

- *Expedient* - a interface gráfica do OCF;
- *VT Manager* - gerenciador de máquinas virtuais;
- *Opt-in manager* - gerenciador de *switches* OpenFlow.

As interfaces norte e sul desses componentes são implementadas utilizando Hyper Text Transmission Protocol Secure (HTTPS) e eXtended Markup Language - Remote Procedure Call (XML-RPC) [94]. Cabe observar que o Flowvisor, embora esteja presente na figura, é um módulo externo, responsável pelo *slicing* de uma rede OpenFlow. O *Opt-in manager* interage com o Flowvisor, fazendo a configuração do *slice* para o usuário. Por utilizar o sistema de *slices*, onde cada usuário possui uma rede virtual própria, foi decidido que não havia necessidade de agendamento de recursos.

3.4 Operação do *testbed*

A monitoração correta de uma federação de recursos é crucial para manter o serviço funcional. Porém, devido à natureza distribuída dos recurso e existência de diversas entidades administrativas, a validação da disponibilidade e garantia do serviço é complexa.

No caso específico do FIBRE, manter a infraestrutura funcional exige a verificação da disponibilidade dos recursos e a disponibilidade de serviços básicos. Para os operadores de ilhas, é importante poder rapidamente detectar falhas e resolver problemas de

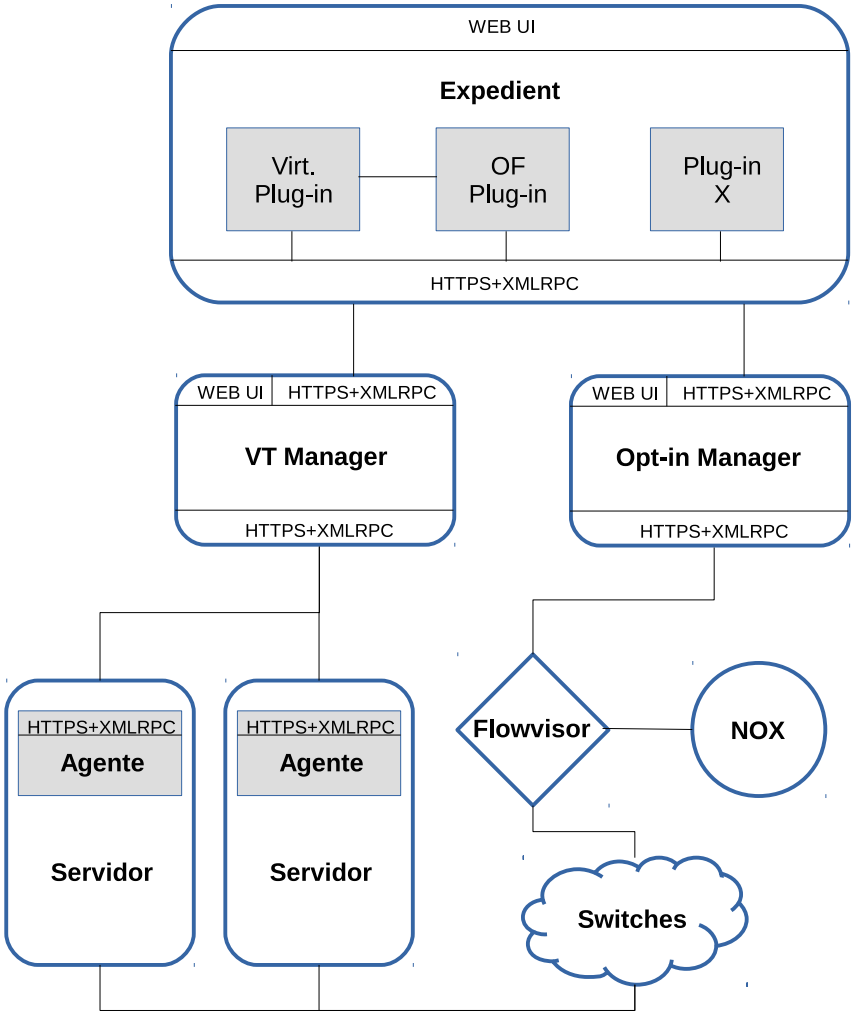


Figura 3.7: Arquitetura do OCF.

uma maneira eficiente. Para os gerentes da federação de recursos é importante avaliar a disponibilidade da ilha e garantir uma boa experiência de usuário.

A Figura 3.8 mostra o modelo de gestão adotado no FIBRE. O comitê de gestão é o corpo de tomada de decisões mais alto e é responsável pelo aconselhamento em assuntos estratégicos e administrativos. É formado por um representante de cada instituição provedora de recursos computacionais ao *testbed*. Já o comitê técnico conduz a evolução técnica do *testbed*, criando recomendações e guias de projeto. Sob o grupo técnico, está o grupo de operação e envolvimento do usuário, que está encarregado da manutenção e suporte ao usuário. Ele deve assegurar o bom funcionamento do *testbed* e promover atividades de disseminação. O NOC é responsável pela correção de falhas e suporte aos operadores das ilhas. Além disso, existe o time de desenvolvimento, que é responsável pela manutenção corretiva e evolucionária de todo o *software* suportado pelo FIBRE.

Para auxiliar a operação do *testbed*, algumas ferramentas padrão de monitoração são

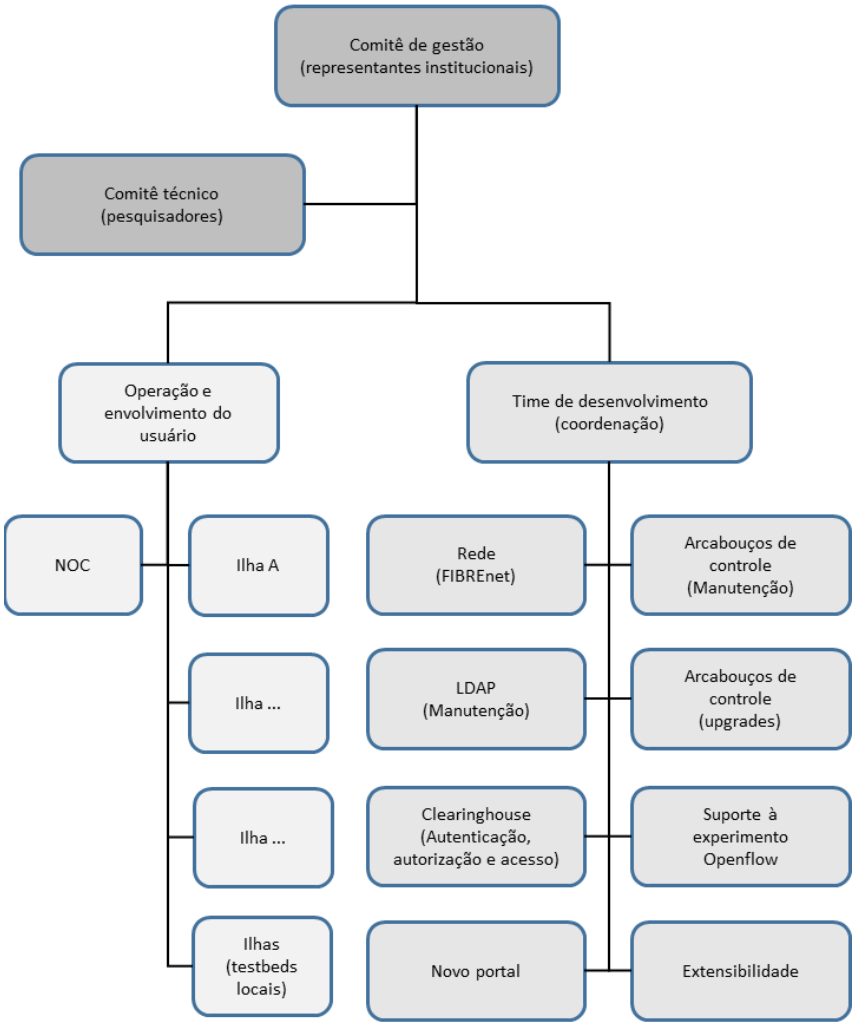


Figura 3.8: Modelo de gestão do FIBRE. Adaptado de [4]

utilizadas, dentre as mais relevantes estão o Zenoss Core [95] e perfSONAR [96]. Eles foram escolhidos por serrem programas de código aberto e com maturidade suficiente para monitorar infraestruturas de rede de nível empresarial.

3.5 ZenOSS

O ZenOSS é uma plataforma de monitoração de infraestrutura de tecnologia da informação [97] e é usado no projeto FIBRE para centralizar as informações e facilitar o trabalho do Network Operations Center (NOC) [98]. O ZenOSS (ZenOSS Core) é uma ferramenta livre e de código aberto e uma plataforma de gerenciamento de rede baseada no servidor de aplicações Zope. O ZenOSS Core é licenciado sob a () General Public License (GPL) versão 2 e provê uma interface *web* que permite que administradores monitorem disponibilidade, configuração, inventário, desempenho e eventos.

O ZenOSS gera alarmes e eventos que são armazenados em um banco de dados (EventArchive), além de gerar relatórios de disponibilidade baseado no tipo de evento. É possível configurar e extrair informações maneira programática através de uma Application Programming Interface (API) JavaScript Object Notation (JSON) que é acessível via Uniform Resource Locator (URL).

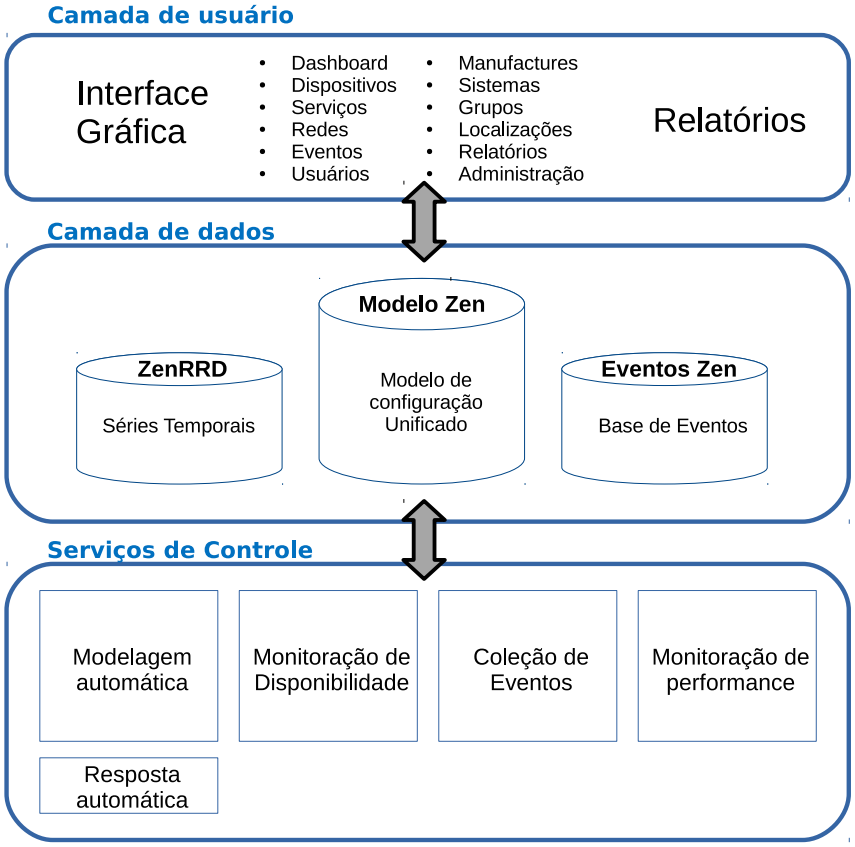


Figura 3.9: Arquitetura do ZenOSS.

A Figura 3.9 mostra a arquitetura do ZenOSS e seus principais componentes [99]:

- *ZenOSS User Layer* – interface gráfica baseada em browser;
- *Data Layer* – A infraestrutura de dados agrega três tipos distintos de informação, cada um armazenado em um mecanismo diferente:
  - ZenModel - Modelo de objeto unificado de ambiente de TI e configurações.
  - ZenRRD - Round Robin Database (RRD) do Zenoss. Usado para guardar informações históricas e gerar gráficos.
  - ZenEvents - Banco de dados para eventos ativos e históricos (Baseado em My Structured Query Language (MySQL)).

- *Collection & Control Services* – Uma série de processos que possibilitam diversas tarefas de controle, incluindo modelos de configuração, monitoramento de desempenho, monitoramento de disponibilidade, coleções de eventos e respostas automáticas.

3.6 perfSONAR

O perfSONAR é um kit de ferramentas de monitoramento de redes. Atualmente, é usado no *testbed* do FIBRE para mostrar informações de perda e atraso unidirecional dos links entre as ilhas considerando apenas o plano de controle[1] [100]. Uma vez que essas informações são relevantes na monitoração do *testbed* e não estavam disponíveis pelo ZenOSS, optou-se pelo uso também do PerfSONAR.

O *Dashboard* utilizado é o Monitoring and Debugging Dashboard (Maddash). O Maddash permite coleta, monitoração e apresentação de dados bidimensionais como um conjunto de matrizes. Os resultados podem ser acessados por uma API REST que fornece os componentes para a apresentação dos dados. O Maddash está sendo desenvolvido como parte do projeto perfSONAR e é uma plataforma relativamente agnóstica ao tipo de informação mostrada [101].

[1] O FIBRE possui um plano de controle para troca de informações entre serviços e um plano de dados para a experimentação.

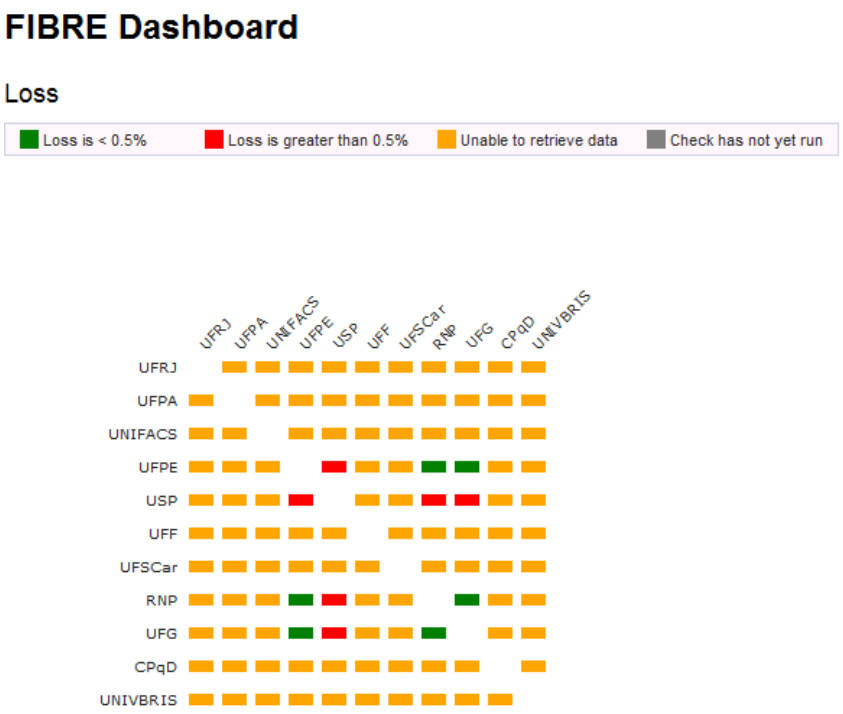


Figura 3.10: Maddash - Tabela de Perda de pacotes.

A Figura 3.10 mostra o *dashboard* da perda de pacotes na página de monitoração do Maddash no FIBRE. As ferramentas realizam os testes e guardam os resultados em arquivos centrais ou mesmo locais.

3.7 Problemas enfrentados no FIBRE

Determinar se um dado serviço está disponível ou não vai muito além de meramente testar a infraestrutura. Ferramentas como ZenOSS e perfSONAR são muito úteis para manter o controle do estado de resposta ao ping, estado de *hardware* ou mesmo dos servidores Hyper Text Transmission Protocol (HTTP). Porém para atestar o funcionamento dos domínios de experimentação, um grande número de componentes do arcabouço deve estar funcionando para que o serviço do *testbed* possa ser considerado disponível.

Por exemplo, os nós Icarus mencionados no Capítulo 3 não podem ser avaliados somente pelo teste de suas interfaces de rede, pois seu estado normal é desligado. Quando um experimento começa, o nó recebe um comando de ligar e pode realizar um *boot* pela rede se assim o experimentador desejar. Endereços de IP e sistemas operacionais podem

ser modificados. Por mais que haja uma política de "limpe a sua bagunça", sempre haverá casos de estudantes desconfigurando os recursos. E não podemos culpá-los, afinal esse é um lugar para aprender.

Outro problema existe no que diz respeito à demanda. Atualmente, as características de tráfego do *testbed* são tipicamente de picos distribuídos de maneira esparsa. O motivo para isso é que muitos dos experimentadores são estudantes que acessam o *testbed* quando em aula ou em seminários. Então, a maioria dos equipamentos fica ociosa por um bom tempo e sem prévio aviso, passa a existir uma demanda de uma sala cheia de alunos tentando criar experimentos ao mesmo tempo. Caso haja uma falha no sistema desconhecida pelos operadores que impeça o uso do *testbed* nesse momento, a primeira impressão de muitos usuários será ruim. Essa possibilidade reforça a necessidade de testes periódicos da infraestrutura. Esperar pela reclamação de um usuário para buscar a solução de um problema é prejudicial à reputação do *testbed* e pode desencorajar novos usuários. Deve-se realizar uma abordagem de solução de problemas pró-ativa.

Hoje, também não há uma solução de monitoração que possa validar o funcionamento do ambiente de experimentação OpenFlow. É preciso verificar se o Flowvisor faz o isolamento do tráfego e Quality of Service (QoS) entre os slices, se as máquinas virtuais conseguem comunicar-se normalmente e se há conexão no plano de dados entre ilhas do testbed. Nenhuma das ferramentas tradicionais de monitoração realizam esse tipo de tarefa.

Também é difícil separar que anormalidades que surgem devido à falhas que ocorrem naturalmente em software ou hardware por conta de bugs ou desgaste do equipamento de falhas sendo causadas por mau uso ou por testes com a intenção de estressar o *testbed*. Nesse ponto, as ferramentas de monitoração tradicionais deixam a desejar nas funcionalidades para auxiliar o operador a distinguir entre o problema e o funcionamento de um experimento.

Capítulo 4

Sistemas de monitoração utilizados em outros testbeds

4.1 NOVI

O Networking innovations Over Virtualized Infrastructures (NOVI) tem como objetivo investigar e experimentar acerca da monitoração, descrição formal e alocação de recursos virtualizados em uma federação de plataformas da Internet do futuro. Foi projetado para integrar as infraestruturas dos *testbeds* PlanetLab Europe (PLE) e FEDERICA [102]. O desenvolvimento do NOVI foi realizado por um projeto de três anos[1] que envolveu 70 pesquisadores e teve orçamento total de 2,36 milhões de euros [103].

Seus principais objetivos são:

- Integrar tecnologias de virtualização em nível de rede e computação com novos métodos e algoritmos para provisionar, monitorar e controlar *slices* dedicados ao estudo da Internet do Futuro;
- Estender conceitos de dados, controle, monitoração e provisionamento em um ambiente federado, de forma a auxiliar provedores a fornecer de uma maneira segura serviços em nuvem com Quality of Service (QoS)) ;
- Fornecer aos usuários finais ferramentas inteligentes para descobrir recursos virtualizados e compor serviços com o auxílio de uma linguagem comum utilizando conceitos e ferramentas da *web* semântica.

A Figura 4.1 mostra a arquitetura do NOVI em três camadas: no topo, os serviços de

[1] 1º de setembro de 2010 a 28 de fevereiro de 2013

controle e gerência do NOVI oferecem mapeamento inteligente de recursos e acesso baseado em políticas estabelecidas. Os componentes da camada central são usados para prover capacidade federada de controle e gerência entre plataformas (na Figura 4.1, assume-se que o Slice-based Federation Architecture (SFA) é implementado). E na camada física plataformas heterogêneas contém os recursos virtuais que serão alocados para usuários. A conexão no plano de dados de um *slice* federado é obtida através de um *switch* virtual do NOVI, o Nswitch.

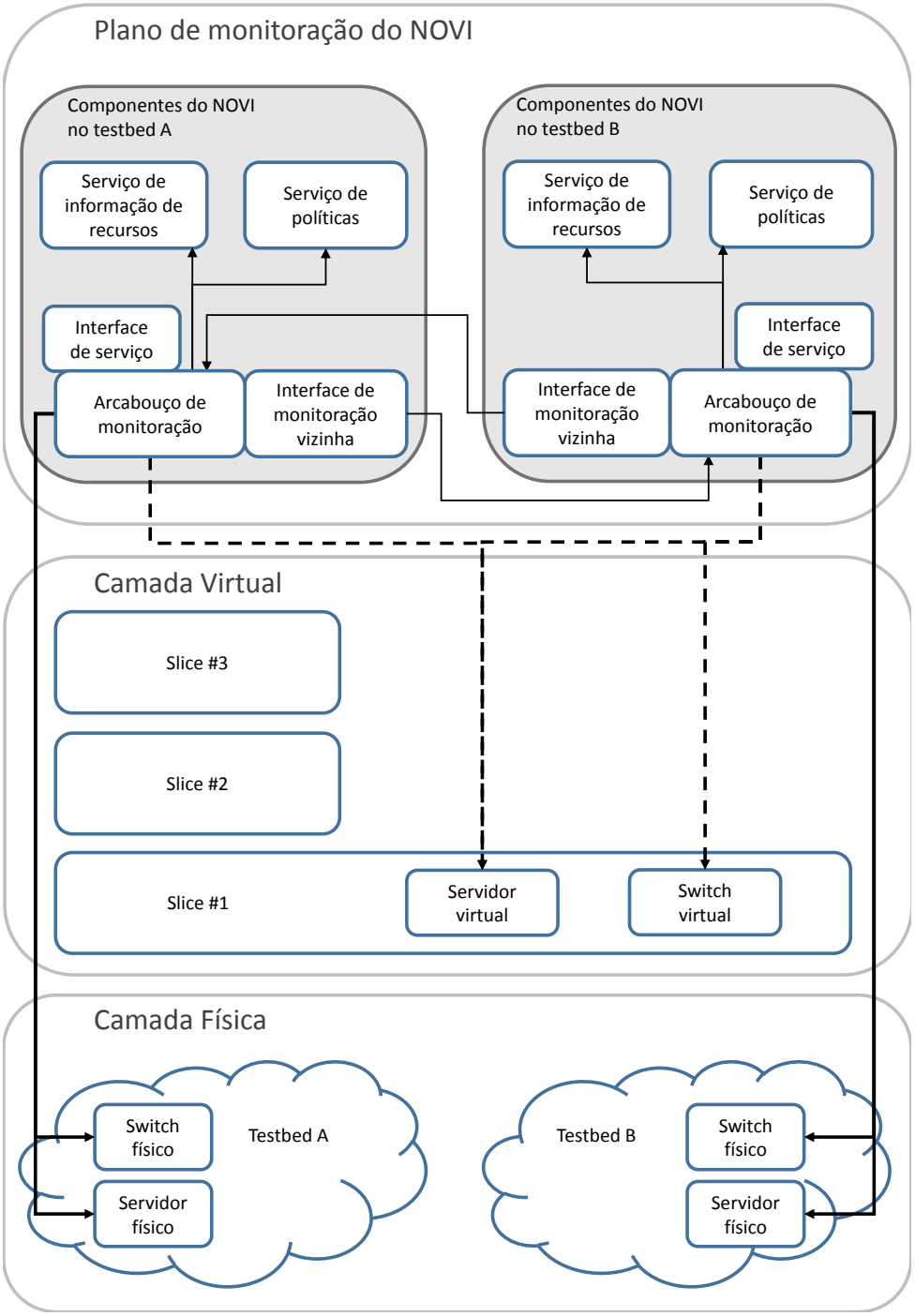


Figura 4.1: Arquitetura do NOVI.

4.2 TopHat

TopHat é o arcabouço de ferramentas de monitoração do PlanetLab Europe (PLE), que é o principal *testbed* dentro da estrutura do FIRE. Sua arquitetura é mostrada na Figura 4.2. Suas fontes de dados de monitoração são os serviços TopHat Dedicated Measurement Infrastructure (TDMI) e infraestruturas de medição que utilizam uma API eXtended Markup Language - Remote Procedure Call (XML-RPC). O TDMI consiste de medições que nenhum outro sistema faz e foram desenvolvidas pelos autores do TopHat, como: *Paris Traceroute* [104], que mede rotas na Internet de forma mais precisa, e o *DoubleTree* [105] que melhora a eficiência de uma infraestrutura de medição distribuída. Um gateway traduz as requisições originadas do TopHat para requisições específicas da plataforma e adapta os resultados com um formato que o TopHat consegue entender [76].

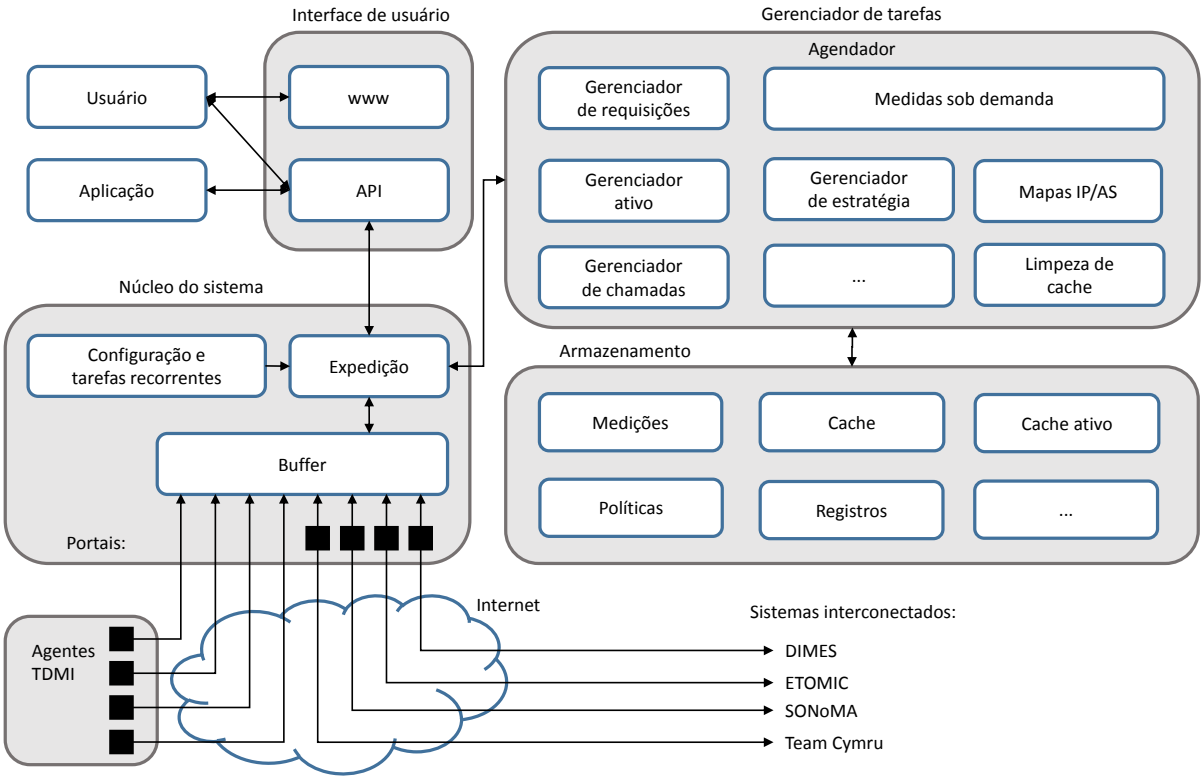


Figura 4.2: Arquitetura do TopHat.

O fluxo de dados no sistema começa com arquivos de dados sendo carregados para uma área de memória no núcleo do sistema, que associa cada um com uma tarefa de monitoração. Essas tarefas são agendadas e processadas segundo um esquema de prioridade. Para evitar a transmissão e armazenamento de informações redundantes, o sistema verifica se houve mudança na medição antes de armazenar do banco de dados.

O TopHat possui um servidor centralizado por ser uma arquitetura mais simples e portanto mais robusta. Os agentes TDMI fazem medições em uma malha completa, o que aumenta a carga no servidor ao quadrado do número de agentes.

O TopHat tem uma arquitetura simples, sendo capaz de realizar a monitoração de diversas infraestruturas diferentes, assemelhando-se a proposta adotada para a arquitetura do FIBREOSS. O TopHat traria para o FIBRE uma solução para auxiliar usuários do FIBRE a escolher os recursos que irão usar e identificar melhor problemas de infraestrutura que possam afetas seus experimentos.

4.3 MOST4FIRE

Implementado no *testbed* BonFIRE [52], parte do FIRE, o MOST4FIRE introduz o conceito monitoração como serviço, um conjunto de tarefas de monitoração à disposição do usuário. Essas medições podem ser realizadas em múltiplas camadas, desde a infraestrutura física à infraestrutura virtual, em nível de plataforma até o nível de serviço e aplicação. O sistema propõe o uso de ontologias de medição para resolver o problema da monitoração em ambientes heterogêneos [51]. Sua principal vantagem é a capacidade de agrupar medições de diferentes tipos de tecnologias através de uma camada de adaptação.

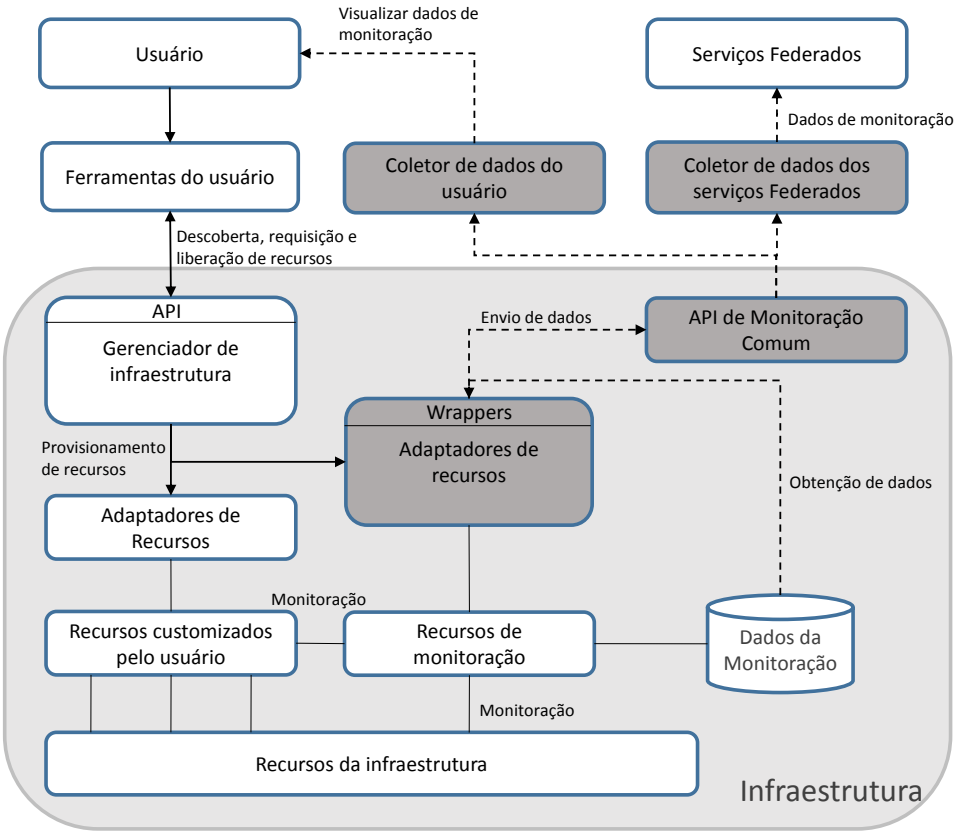


Figura 4.3: Arquitetura do MOST4FIRE.

Em sua tese de janeiro de 2016, o autor do MOST4FIRE mostra detalhes de uma arquitetura de monitoração seguindo a mesma filosofia chamado Monitoring Architecture for Federated heterogeneous Infrastructures (MAFIA) [2]. O MAFIA usa um modelo de informação baseado em uma ontologia de monitoração chamada Monitoring Ontology for Federated Infrastructures (MOFI) que usa conceitos da ontologia Monitoring and Measurement in the Next generation Technologies (MOMENT) e da ontologia do NOVI. Diversos recursos dessas duas ontologias foram usados para criar uma ontologia genérica o suficiente para ser útil na monitoração de ambientes heterogêneos [3]. Essa mesma ontologia está sendo usada para padronizar os serviços de monitoração de um projeto em andamento chamado Open Multinet [106].

A Figura 4.3 mostra a arquitetura dos componentes do arcabouço MAFIA. Os componentes marcados em cinza são responsáveis pela unificação e coleta de dados. Somente um usuário e uma infraestrutura são mostrados para simplificar o diagrama, porém na

[2] As semelhanças entre os dois e a mesma autoria indicam que seja uma evolução do MOST4FIRE, que foi publicado em 2014, enquanto o MAFIA foi publicado em 2016, embora o autor não faça menção direta ao MOST4FIRE em sua tese.

[3] A ontologia MOMENT é focada em medições de rede e a ontologia do NOVI foi completamente reutilizada.

realidade diversas infraestruturas e usuários estariam conectados simultaneamente. É importante notar que nenhuma medição é realizada pelo MAFIA. Ele é responsável pela adaptação de uma ampla gama de medições utilizando uma estrutura de conhecimento muito bem definida, tornando possível monitorar redes heterogêneas [107].

4.4 GEMINI

O GENI Measurement and Instrumentation Infrastructure (GEMINI) combina duas infraestruturas de medição utilizadas no GENI, INSTOOLS e LAMP. A Figura 4.4 mostra sua arquitetura e interação entre seus componentes. O usuário começa a acessar o GENI *Desktop* para instalar as medições que deseja; então, o GENI *Desktop* usa os serviços do nó global para orquestrar as medições requisitadas e atualizar o Unified Network Information Services (UNIS) [108] com as novas configurações. Os agentes automaticamente baixam a nova configuração do UNIS, começam a realizar medições e enviam os dados coletados para o Armazenamento de medições (Módulo Serviço de medições na Figura 4.4). O serviço de medições registra no UNIS os dados disponíveis e quando requisitado, alimenta o GENI *Desktop* e as interfaces de visualização locais (módulo visualização na Figura 4.4 com dados de monitoração. No fim do experimento, o usuário pode arquivar suas medições no serviço de arquivo externo.

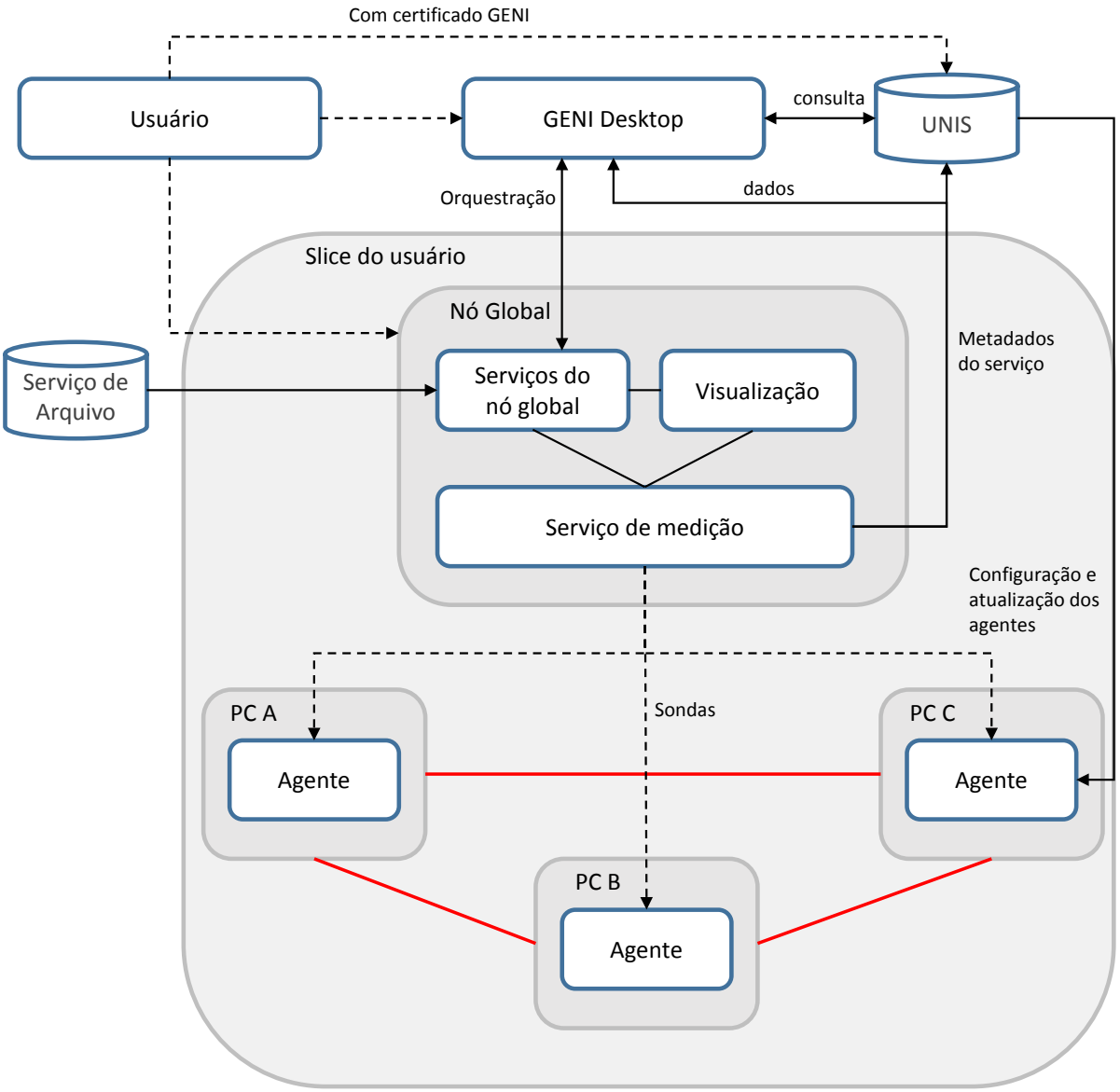


Figura 4.4: Arquitetura do GEMINI.

4.5 Avaliação das ferramentas

Durante a implantação do FIBRE, foram avaliadas diversas propostas de monitoração. Uma arquitetura de monitoração capaz de configurar, monitorar, coletar e exibir dados de monitoração da infraestrutura e de experimentação é proposta em [109], utilizando um ponto de integração de medições Measurement Data Integration Point (MDIP), baseado na arquitetura do perfSONAR.

Nenhuma das ferramentas de integração utilizada em outros *testbeds* mostrou-se completamente adequada ao FIBRE. A ferramenta TopHat, que atualmente encontra-se na

versão beta, foi testada no início da implantação do FIBRE e precisa de modificações para ser integrada. Além disso, ela não fornece suporte à operação do *testbed*.

Outras ferramentas apresentadas são muito bem ajustadas aos *testbeds* para qual foram projetadas, mas não estavam totalmente adequadas às necessidades do FIBRE. No entanto, uma adaptação desses sistemas acrescentando as funcionalidades desenvolvidas para o FIBRE seria uma ótima contribuição à monitoração. O GEMINI, por exemplo é usado no GENI para monitorar experimentos e sua adaptação ao FIBRE pode ser uma solução. As medições dos experimentos podem ser coletadas pelo FIBREOSS e contribuir para o diagnóstico de problemas. Já o projeto NOVI é também um CMF, que se propõe a fornecer todo o suporte necessário, ao invés de suportar uma infraestrutura heterogênea e flexível. Isso limita a expansão do *testbed* e ossifica a estrutura de controle e monitoração. O MAFIA propõe soluções para os problemas apresentados nesse trabalho e poderia ser utilizado para agregar dados de monitoração desde de que sejam feitas as adaptações necessárias para o FIBRE. O sistema de monitoração Open Multinet, que usa a ontologia de monitoração do MAFIA, está em desenvolvimento e é ser uma opção a ser estudada quando estiver completo.

Capítulo 5

FIBREOSS - especificação

Devido às particularidades descritas na Seção 3.7, os sistemas de gerência "pacote fechado" não cobrem todas as necessidades dos operadores. A natureza distribuída dos recursos e a existência de vários domínios administrativos torna a validação das disponibilidade dos recursos complicada e não confiável. As ferramentas de monitoração tradicionais medem o funcionamento de uma rede e de seus serviços, mas não são próprias para verificar a disponibilidade e medir o desempenho de um recurso virtual no qual será implementada a rede. Para dar conta dos problemas mencionados, um sistema customizado foi projetado para executar testes de rotina, contabilizar a disponibilidade e agregar alarmes usando as relações de dependência hierárquicas do sistema no FIBRE. Esse sistema foi nomeado como FIBRE Operation Support System (FIBREOSS).

5.1 Especificação de funcionalidades e requisitos de monitoração de usuários e de operadores

Para fins de definir os requisitos do sistema, foi necessário, primeiramente, detalhar os requisitos de monitoração de acordo com a visão de usuários e de operadores. A lista completa de requisitos encontra-se no Apêndice C

5.1.1 Requisitos dos Operadores

Operadores precisam de informações relativas ao estado de saúde do *testbed* para tomar ações corretiva e preventiva. É necessário um sistema de suporte de primeiro nível (First Level Support (FLS)) que mostre o estado atual de dispositivos e histórico de disponibilidade para auxiliar o trabalho dos operadores. As ferramentas de monitoração utilizadas

no FIBRE são capazes de monitorar a infraestrutura de rede tradicional, como servidores e enlaces, porém não é capaz de testar todos os serviços oferecidos pelo *testbed*. Por isso um painel que combine todas as informações de monitoração já disponíveis pelo ZenOSS, perfSONAR e por agentes de teste de específicos deve ser desenvolvida.

Outra funcionalidade importante para a resolução de problemas é o relatório de disponibilidade, que deve mostrar uma informação histórica do estado de saúde dos dispositivos. Essa funcionalidade é essencial na detecção de falhas intermitentes e recorrentes. Novamente, as ferramentas ZenOSS e perfSONAR possuem meios para gerar relatórios de disponibilidade de dispositivos e enlaces de rede, respectivamente, porém são muito superficiais e não mostram de maneira aprofundada o estado da infraestrutura. O uso dos enlaces no plano de controle assim como no plano de dados também fornece informações importantes para solucionar problemas de desempenho do *testbed*. Métricas como vazão perda e atraso devem ser disponibilizadas aos operadores.

Estatísticas de número de usuários ao longo do tempo e de suas atividades são importantes para entender as características de tráfego das atividades e ajustar o suporte. Também, o uso de recursos da federação deve ser monitorado para garantir que os usuários não causem problemas no *testbed*.

O uso de recursos nos servidores deve ser também monitorado para esclarecer problemas. Informações como uso de CPU, memória, I/O, uso do disco, número de processos em execução e número de máquinas virtuais ativas devem estar disponíveis. O ZenOSS faz exatamente o proposto e essas informações podem ser utilizadas para correlacionar falhas em serviços do *testbed* com problemas de *hardware*.

5.1.2 Requisitos de usuários

Nesta seção, são descritos os requisitos do FIBRE relacionados à monitoração e como eles podem ser alcançados. Cabe observar que aqui não são apresentadas ferramentas para monitoração do experimento, mas ferramentas que fornecem dados sobre disponibilidade e estabilidade do ambiente de teste que podem ser importantes para o usuário na hora de escolher quais recursos usar e em que momentos usar.

Um mapa com uma visão em alto nível do estado de saúde dos nós sem fio é importante para que usuários saibam o que esperar em termos de nível de serviço. Um sistema deve mostrar a reputação das ilhas e problemas mais frequentes. Ajudando assim, o usuário a evitar a frustração de não saber ao certo se falhas que ocorrem em seu experimento

são falhas da infraestrutura. O histórico de falhas e histórico de disponibilidade tem a função de auxiliar o usuário a conhecer a reputação das ilhas do *testbed* sem precisar ter utilizado.

No domínio sem-fio, o uso do espectro dos nós pode auxiliar o usuário a entender o meio de transmissão antes de realizar seus experimentos. Já no domínio Openflow as tabelas de fluxo nos *switches* Openflow deve estar disponível para que o usuário possa depurar as aplicações de controle desenvolvidas.

5.1.3 Requisitos do comitê gestor

Os gestores do projeto FIBRE precisam prestar contas da contribuição científica e do nível de serviço do *testbed* à sociedade, portanto a monitoração do SLA, uso de recursos e número de usuários ativos é importante para justificar a existência do projeto e planejar expansões e melhorias.

5.2 Especificação da infraestrutura de servidores do sistema FIBREOSS

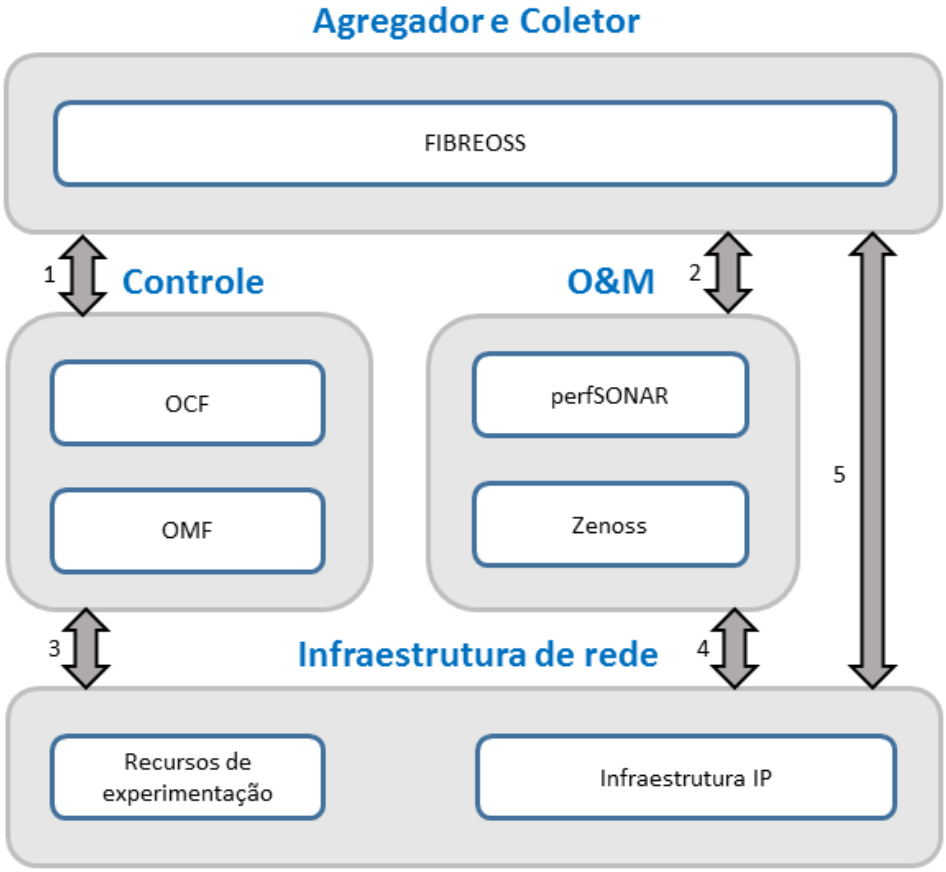


Figura 5.1: Visão em alto nível do FIBREOSS e uma ilha do FIBRE.

A Figura 5.1 mostra a interação de um servidor do FIBREOSS com os componentes do FIBRE, sendo eles o ZenOSS, o perfSONAR e os arcabouços de controle e gerência OMF e OCF. O FIBREOSS também utiliza/avalia os servidores de VPN e de LDAP, mas de forma transparente, ou seja, não são desenvolvidos módulos específicos para interação com esses serviços. O status do LDAP e da VPN ficam expostos pela interação com os serviços de monitoração já existentes e com os arcabouços de controle e gerência. Os servidores associados ao FIBREOSS devem realizar testes programáticos e obter informações de experimentos dos componentes principais do FIBRE. Dentro do bloco de infraestrutura de rede os blocos infraestrutura IP e recursos de experimentação são usados para diferenciar a infraestrutura de rede padrão dos dispositivos para experimentação.

Ainda na Figura 5.1: As indicações 1,2 e 5 mostram a interface *Southbound* do FIBREOSS. Sendo que a interface comunicação com os sistemas de controle de experimentos, representada pele indicação 1 da figura, é realizada através de conexões diretas ao banco

de dados e API XML-RPC e a indicação 2 representa a comunicação com o suporte à Operação e Manutenção (O&M). É realizada através de chamadas programáticas a API do ZenOSS e perfSONAR. Esses módulos realizam o monitoramento dos dispositivos da infraestrutura, representada pela indicação 4, através de pacotes Simple Network Management Protocol (SNMP), acesso Secure Shell (SSH), monitoração ativa e passiva da rede, entre outros. Já o monitoramento e controle dos experimentos e dispositivos do *testbed* é indicado por 3, e é feito pelos Control and Management Frameworks (CMFs). O OMF controla os nós sem fio sua comunicação é baseada em eXtensible Messaging and Presence Protocol (XMPP) [90]. O OCF instancia, configura e monitora recursos para experimentos com OpenFlow [94]. A indicação 5 mostra as medições realizadas pelo FIBREOSS através de agentes de teste customizados.

Devido ao FIBREOSS monitorar uma infraestrutura federada, estão previstos dois tipos de servidores, como mostrado na Figura 5.2[1]:

- *Concentrador*

  Responsável por recolher informações de todas as ilhas. (e.g. Ping status, delay, etc...) e está situado na ilha NOC.

- *Local*

  Recolhe informações mais específicas e interage com elementos da ilha pela rede local.

O servidor concentrador recolhe informações através de uma conexão direta com o banco de dados dos servidores locais e disponibiliza aos usuários. Essa abordagem possui a desvantagem de ser mais difícil de gerenciar, porém tem vantagens no que diz respeito à escalabilidade e também permite que a monitoração continue localmente na eventualidade de falha de conexão com as ilhas.

A conexão é feita do concentrador aos bancos de dados locais, onde os dois bancos de dados representados por Database (DB) e Round Robin Database (RRD) são respectivamente os bancos de dados de alarmes e histórico permanente de medições e o um banco de medições que tem uma tamanho fixo, onde a medição mais recente inserida substitui a mais antiga.

[1] Os servidores locais não foram implementados durante o desenvolvimento deste trabalho

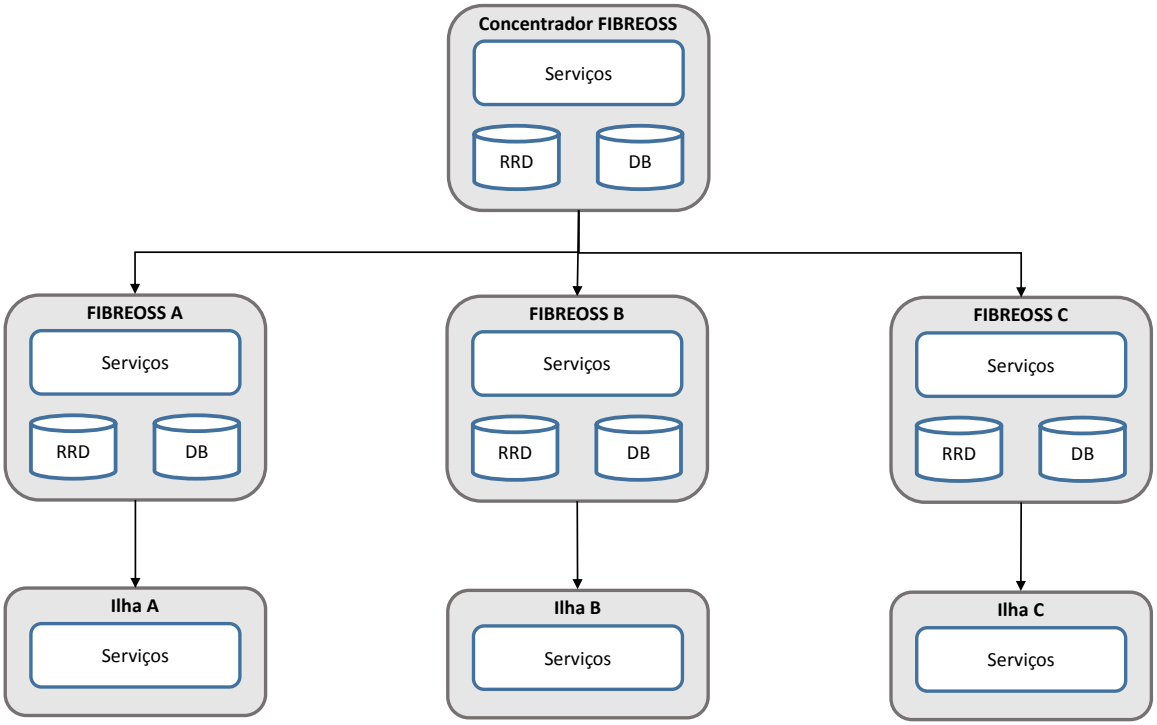


Figura 5.2: Organização de servidores do FIBREOSS.

5.2.1 Detalhamento da Arquitetura

Nesta seção será explicado como os componentes do FIBREOSS interagem, processam informações e suas principais funcionalidades:

- Teste do plano de dados do domínio Openflow
- Teste do domínio sem-fio
- Cálculo da disponibilidade
- Processamento de testes do plano de controle

A Figura 5.3 mostra uma visão de alto nível do fluxo de dados na monitoração de um serviço com o FIBREOSS. Alguns dos agentes de teste utilizados são sistemas de terceiros como o ZenOSS e o Perfsonar, os quais realizam testes genéricos na infraestrutura e o FIBREOSS recolhe essas informações e gera relatórios de disponibilidade. Não havia necessidade de refazer essas mesmas medições genéricas, por isso optou-se pela coleta. Testes mais específicos de *status* para os *frameworks* OMF e OCF exigem agentes de teste customizados, os quais foram propostos e desenvolvidos com o FIBREOSS.

Tudo relacionado ao *testbed* é experimental, por isso, o paradigma adotado no FIBREOSS para desenvolver os procedimentos do sistema de supervisão foi o de testar tudo e utilizar todos os componentes do *testbed* do mesmo jeito que um usuário deveria fazer. Esse tipo de teste ajudou a manter um controle do estado de saúde dos equipamentos, resolvendo o problema do baixo uso ou da demanda de pico dos recursos e também a melhorar a plataforma de experimentação encontrando *bugs* no sistema de maneira automatizada.

A metodologia de teste proposta, simulando o usuário, pode ser aplicada na maioria dos testbeds, uma vez usualmente utilizam um portal *web* para acesso. No FIBREOSS, foram desenvolvidos *scripts* em Python utilizando a biblioteca de testes de *browser* Selenium [110], a qual foi usada para instanciar um *browser* e acessar as interfaces de usuário para simular as ações e entradas executadas por um experimentador. Os resultados são armazenados em um banco de dados local e usados para análise posterior no caso de ocorrência de uma exceção. Uma execução diária destes scripts é executada em nível local e federado para determinar a disponibilidade dos serviços.

Então, o modo padrão de avaliar a disponibilidade de um serviço foi melhorada para gerar relatórios mais confiáveis, incluindo testes específicos que simulam as interações do usuário. De forma que os relatórios passaram a refletir a experiência do usuário.

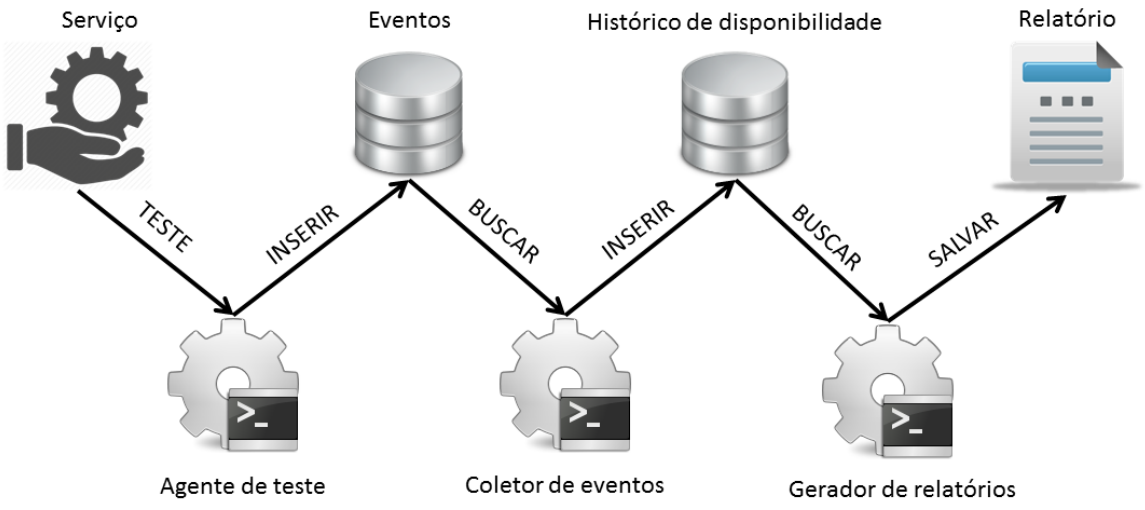


Figura 5.3: Fluxo de dados no FIBREOSS.

A arquitetura é mostrada em detalhes na Figura 5.4. Onde as setas indicam ações realizadas pelos componentes do FIBREOSS enquanto a Figura 5.5 mostra o fluxo de dados no sistema.Na indicação 1 da figura, os agentes de teste do FIBREOSS realizam a monito-

ração da infraestrutura[2] e dos arcabouços OCF e OMF, representado pela indicação 5. A infraestrutura é controlada pelos arcabouços OCF, indicado por 2 e OMF, indicado por 3 e uma monitoração genérica é realizada pelo perfSONAR, Zenoss e Zabbix[3], representado pela indicação 4. A indicação 6 representa o módulo conector, que recupera informações do OCF, OMF e dos sistemas de monitoração através e API ou conexões diretas ao banco de dados. Na marcação 7, o módulo coletor faz a adaptação dos dados e na indicação 8 representa a inserção dos dados do banco de dados do FIBREOSS. Juntamente com os agentes de teste que também fazem a inserção. Na indicação 9 o módulo de relatório recupera as medições e alarmes gerados do banco de dados e armazena os relatórios em um arquivo local, representado pela marcação 10. Na indicação 11, a interface Common Gateway Interface (CGI) disponibiliza os relatórios e medições através de uma API JSON, representada pela marcação 13. A indicação 12 representada a interface web disponibilizando ao usuário as informações. A interação do usuário com o portal é identificada pela marcação 14. A autenticação é feita utilizando LDAP.

[2]Por simplicidade o termo infraestrutura é utilizado aqui para referir-se tanto aos recursos de experimentação como a infraestrutura genérica do *testbed*

[3]Em fase de implantação no FIBRE

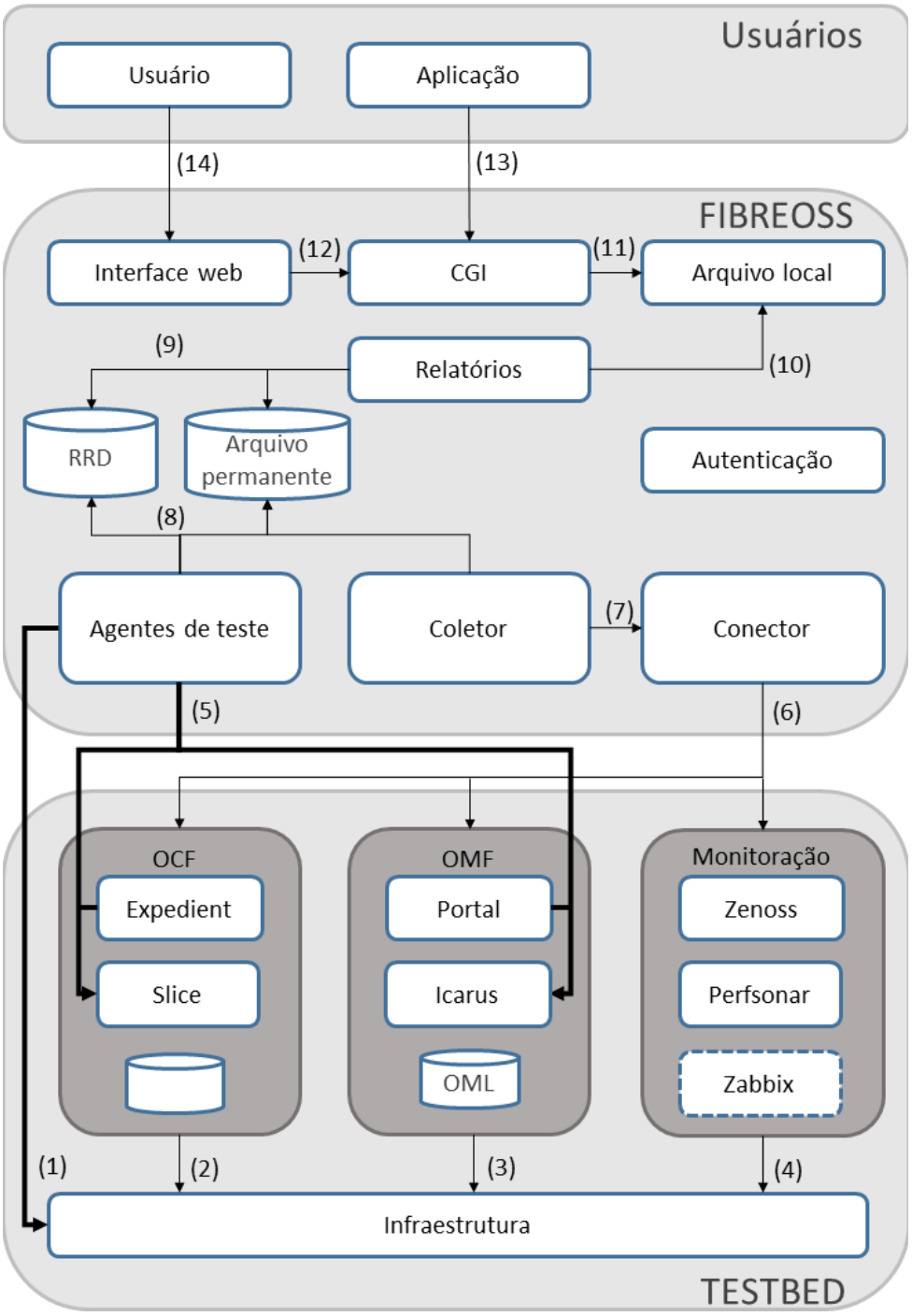


Figura 5.4: Arquitetura detalhada do FIBREOSS mostrando as ações realizadas pelos componentes.

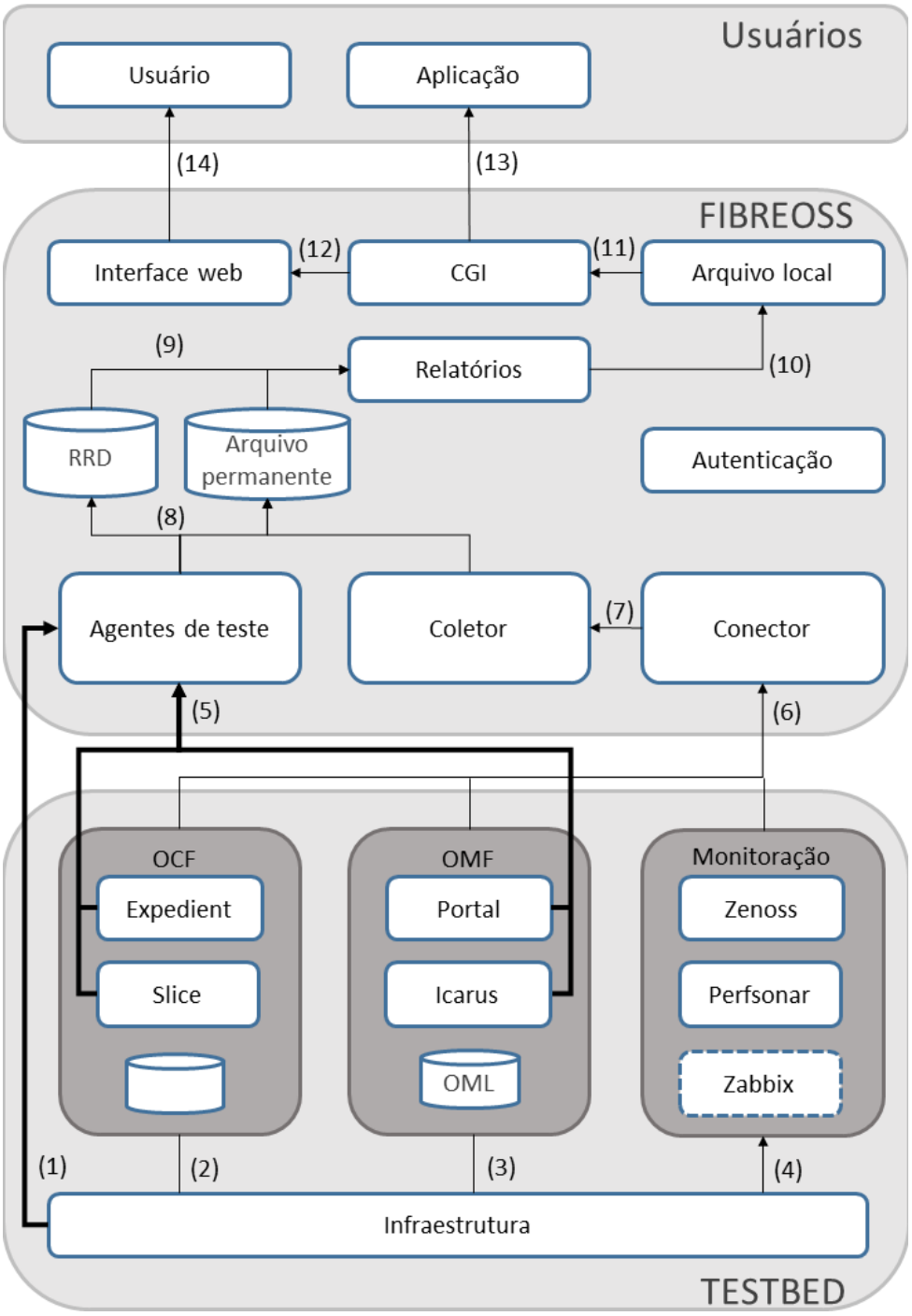


Figura 5.5: Arquitetura detalhada do FIBREOSS mostrando o fluxo de dados no sistemas.

5.3 Especificação do módulo de comunicação com o ZenOSS

O sistema de monitoração ZenOSS realiza testes da infraestrutura e gera alarmes avisando a existência de problemas. Esses alarmes são coletados para contribuir com os dados de monitoração gerados pelo testes do FIBREOSS. Além disso, o sistema de descoberta de

dispositivos do ZenOSS monta a base de dispositivos sob monitoração e disponibiliza através de sua API.Esses dados são coletados e usados pelo FIBREOSS para estabelecer o escopo de monitoração.

O *framework* de comunicação com o ZenOSS foi estudado para definir a melhor maneira de integrá-lo ao FIBREOSS. A seguir, seguem as especificações desse módulo para o FIBREOSS.

5.3.1 Painel de comentários

O objetivo desse submódulo é exibir eventos correntes e possibilitar a inserção de comentários. Um *script* escrito em PHP faz chamadas diretamente à API do ZenOSS usando cURL. O JSON retornado é processado e ordenado para ser exibido na página do LS-WEB. Os eventos recentes[4] são mostrados em uma tabela onde os operadores das ilhas do FIBRE[5] podem comentar os eventos. Esses comentários são gravados em um banco de dados e estarão disponíveis para visualização na forma de um histórico para os operadores e usuários do *testbed*. Essa funcionalidade permite um acompanhamento rápido das ações em curso montando uma relação unívoca entre alarmes e ações corretivas. Devendo ser usada em conjunto com um sistema de gerenciamento de tíquetes, onde outras atividades de operação não relacionadas à falhas são documentadas.

5.3.2 Relatório de disponibilidade (Teste de Ping)

Esse submódulo gera relatórios de disponibilidade a partir dos eventos de \Status\Ping do ZenOSS. Um *script* em Python que é chamado como tarefa agendada do sistema operacional recolhe a listagem de dispositivos monitorados e a lista eventos do EventArchive do ZenOSS de todos os dispositivos.

Diariamente, um *script* recolhe os eventos, calcula a disponibilidade diária, que é dada pelo algoritmo descrito na seção 5.9 e grava o valor no banco de dados local. Após a gravação, um *script* localizado na pasta cgi-bin recolhe os valores de disponibilidade e gera um relatório dos últimos sete dias.

[4]Eventos recentes são eventos ativos há dois dias ou menos. Essa configuração é definida na plataforma do ZenOSS.

[5]Usuários que tem o *flag Administrator* em seu cadastro.

5.4 Especificação do módulo de comunicação com o perfSONAR

O FIBRE já possui o Maddash[6] instalado e configurado. O *dashboard* pode mostrar perdas, atraso unidirecional e vazão entre todas as ilhas. Os testes realizados pelo perfSONAR no FIBRE atualmente são relacionados à medição do atraso unidirecional usando o One Way Ping (OWAMP). Essa métrica é avaliada continuamente por ter um baixo consumo de banda. Com o teste OWAMP, as métricas proporcionadas são: atraso unidirecional, perda de pacotes, *jitter* e pacotes duplicados. O teste envia 10 pacotes por segundo (0.1 segundos com pausas entre cada pacote). Para minimizar a quantidade de dados armazenados enquanto possibilita uma análise estatística mais detalhada, as medidas são salvas como histogramas. Cada histograma representa medidas feitas no período de 1 minuto. Essa técnica usa o Transmission Control Protocol (TCP) para o canal de controle e User Datagram Protocol (UDP) para enviar pacotes de teste. [7]

Com o módulo proposto no FIBREOSS, os eventos de desempenho de rede são coletados através da API JSON do MaDDash e exibidos no portal LS-WEB (portal do NOC). Nesse painel, é possível que os operadores comentem os eventos e os usuários acompanhem problemas de desempenho da rede. Os eventos são gravados em um banco de dados e usados para correlacionar problemas de desempenho em experimentos.

5.5 Especificação do módulo de comunicação com o OMF

É importante testar o OMF para assegurar o bom funcionamento do domínio sem-fio. Um rotina de testes customizados foi desenvolvida para ser incluída na colação de monitoração do FIBREOSS.

O módulo de comunicação com o OMF inclui rotinas de teste dos e recursos do *testbed* sem-fio e do portal da ilha, que também é o portal do OMF.

Um *script* em Python envia requisições HTTP simulando a interação que o usuário teria com a interface *web* do portal da ilha. A rotina do *script* deve autenticar o usuário de teste e reservar todos os nós da ilha durante um período curto de tempo, o suficiente para realizar testes no OMF. A reserva é realizada para uma data futura e a frequência e horário dos testes devem ser ajustadas para não ocupar demasiadamente recursos que os

[6] O maddash só mostra dados monitorados na rede de controle.
[7] Cabe observar que o FIBRE utiliza, no momento, apenas a visualização do atraso e das perdas.

usuários do *testbed* precisam.

Após realizada a reserva, uma rotina acessa o omf-console da ilha e tenta ligar todos os nós sem fios reservados [8] e em seguida faz um teste de ping. Os resultados são armazenados em um banco de dados postgreSQL na máquina virtual do FIBREOSS.

Cabe observar que, no momento que o script é executado, ele faz uma reserva para o futuro, de acordo com o intervalo entre esses testes que for definido pelo operador. Assim, quando o script é executado para testar o uso efetivo dos recursos, ele usa a reserva feita na última execução do script. Optou-se pelo uso de uma reserva futura para ter maior chance de reservar todos os nós da ilha, pois se algum nó já tiver sido reservado por um usuário, ele não poderá ser incluído no teste.

As Figuras 5.6 e 5.7 mostram o fluxograma do agente de teste desenvolvido para o OMF. O bloco com a letra 'A' representa o ponto de continuação no fluxograma.

[8] Usando o comando "omf tell".

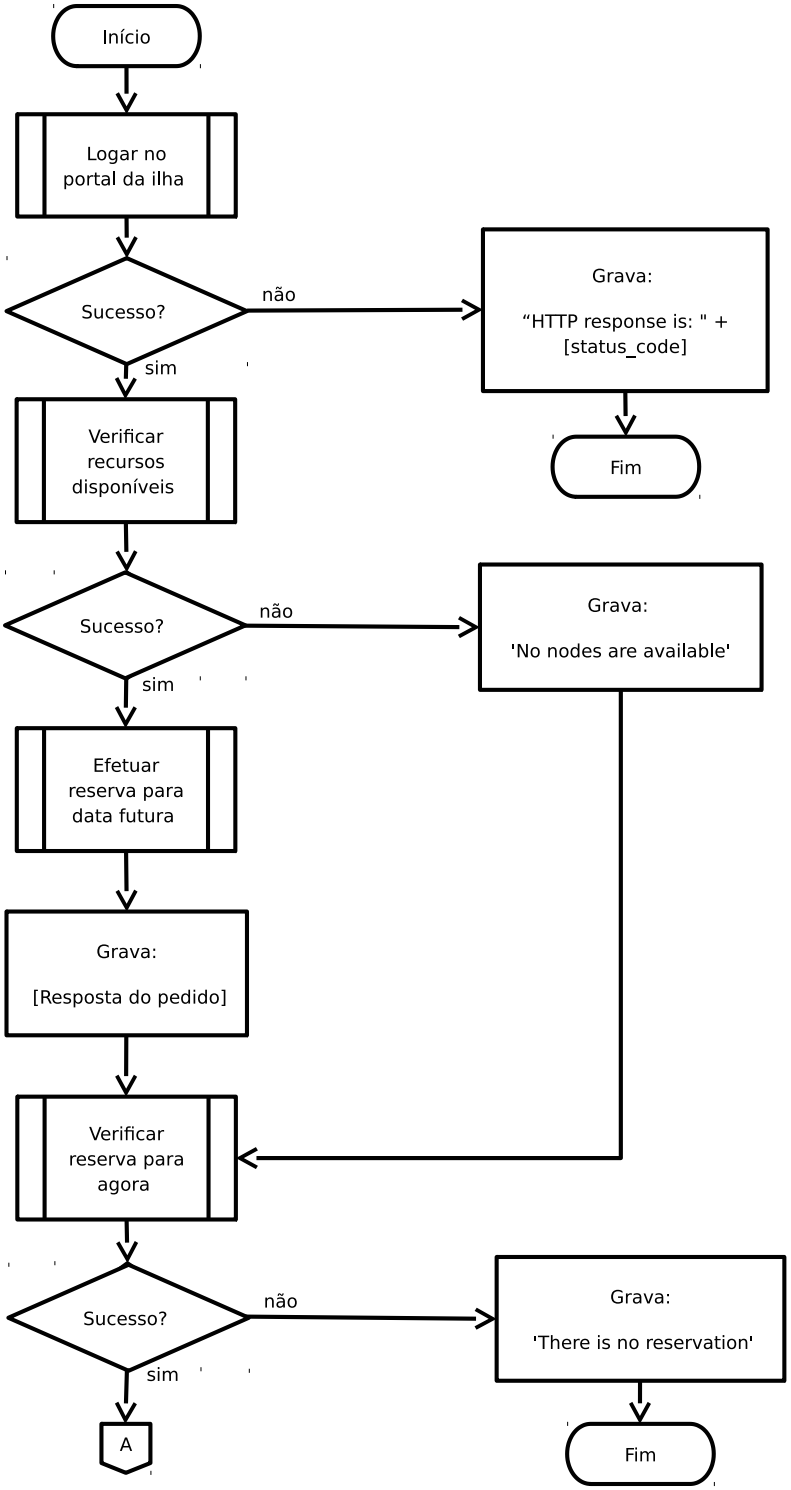


Figura 5.6: Fluxograma do agente de teste do OMF - Parte1.

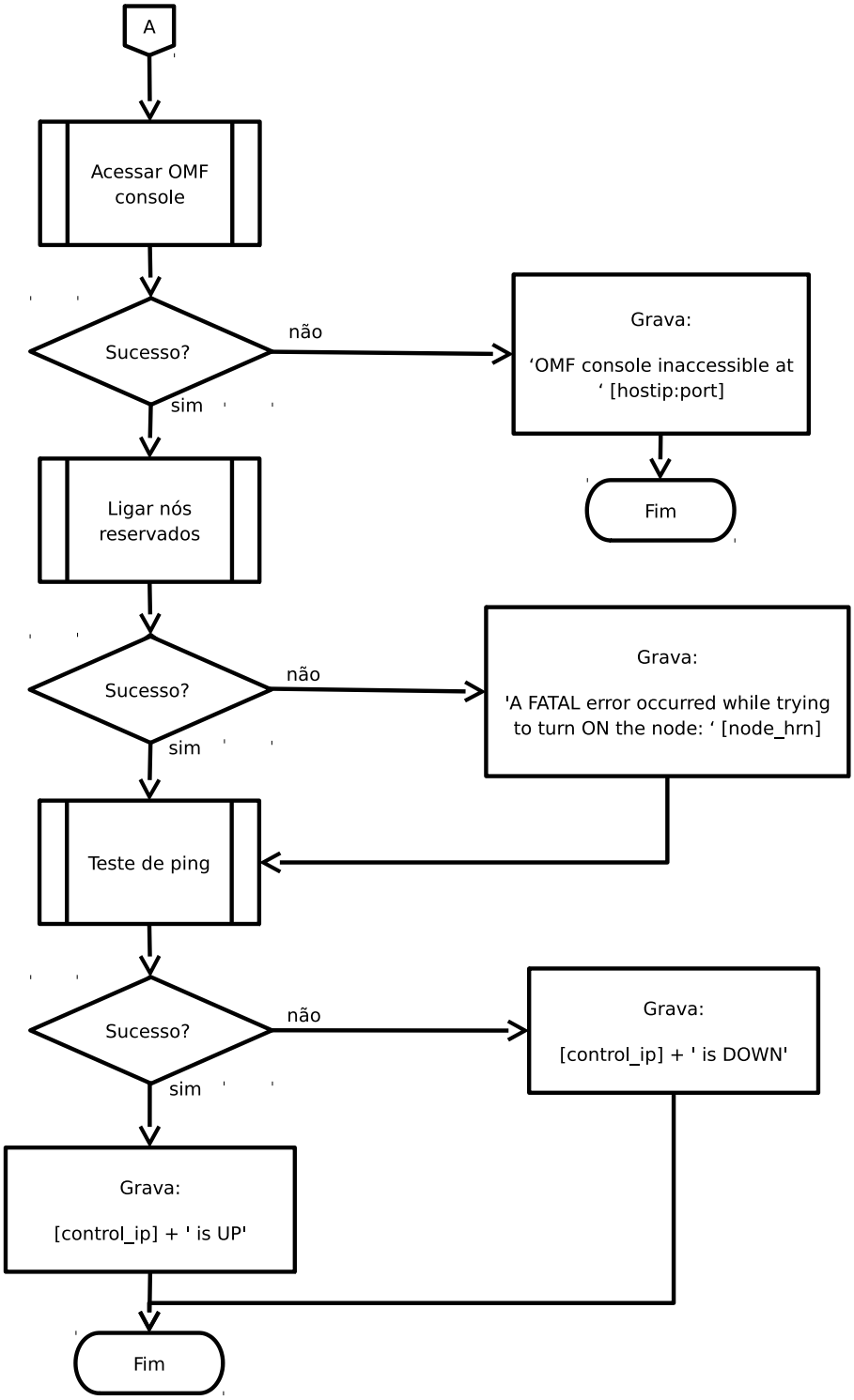


Figura 5.7: Fluxograma do agente de teste do OMF - Parte2.

5.6 Especificação do módulo de comunicação com o OCF

O domínio OpenFlow é complexo e possui muitos componentes que precisam estar funcionando para que a experimentação seja possível. Desde agentes para criação de máquinas

virtuais, controladores de recursos sem fio ao portal *web*, existem muitas funcionalidades que precisam ser testadas para que ações corretivas sejam aplicadas pelo operadores de maneira pró ativa para evitar que usuários passem pelo transtorno de fazer reclamações. Alarmes são gerados e além disso os erros gerados pelo portal do OCF são coletados diretamente do banco de dados para fazer parte da coleção de dados de monitoração do FIBREOSS.

O agente de teste desenvolvido para OCF conecta ao Expedient, que é a interface de usuário do OCF. Por ser uma interface que utiliza tecnologia Assincronous Javascript (AJAX), não seria possível o mesmo modelo utilizado para testar o OMF que conta com requisições HTTP. Uma instância do navegador Firefox é iniciada no servidor de display virtual virtual framebuffer X server for X Version 11 (Xvfb) controlado pelo *webdriver* Selenium para simular ações que seriam executadas por usuários.

Os testes foram especificados seguindo os procedimentos do *checklist* criado pelo NOC do FIBRE, no qual os operadores das ilhas do FIBRE devem realizar periodicamente o teste para avaliar a saúde de suas ilhas. Além disso, testes mais aprofundados que seriam muito exaustivos para os operadores foram acrescentados na rotina de testes agendados do FIBREOSS. [9]

A rotina de testes do Expedient:

- Criação de um slice
- Verificação dos Agregados
- Criação de máquinas virtuais
- Execução e parada de máquinas virtuais
- Reserva de recursos OpenFlow
- Configuração de um controlador OpenFlow
- Teste do plano de controle e do plano de dados
- Teste das máquinas virtuais
- Finalização e limpeza do slice

[9] Todos os testes realizados neste trabalho foram feito usando o portal federado situado no NOC para simplificar a análise. Os testes poderiam também ser feitos nos portais locais da cada uma das ilhas

Os após a execução das ações através da interface gráfica, o *script* de teste busca mensagens indicando sucesso ou falha no portal. Essas informações são recolhidas usando um *parser* HyperText Markup Language (HTML).As VMs são criadas usando os botões de controle da página. Para verifica se a criação foi bem sucedida, o *script* de teste espera um tempo pré definido [10] e verifica as mensagens mostradas pelo portal do OCF.

A Figuras 5.8,5.9 e 5.10 mostram o fluxograma do agente de teste desenvolvido para o OCF. Os blocos com as letras 'A' e 'B' representam pontos de continuação no fluxograma e o bloco com o texto '1 hora' representa um tempo de espera de 1 hora.

[10] Atualmente o tempo é de 1 hora, como mostrado no fluxograma

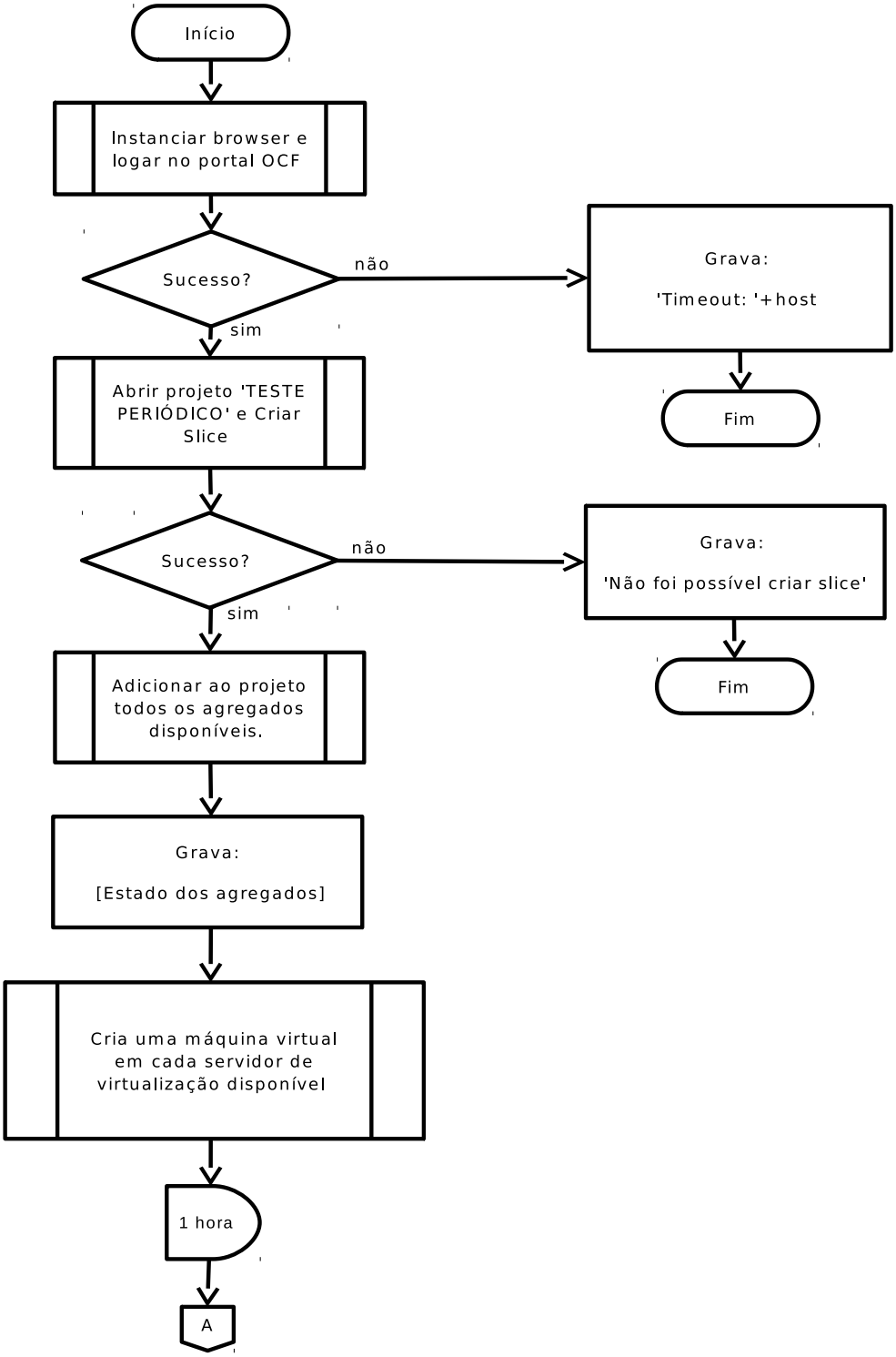


Figura 5.8: Fluxograma do agente de teste do OCF - Parte1.

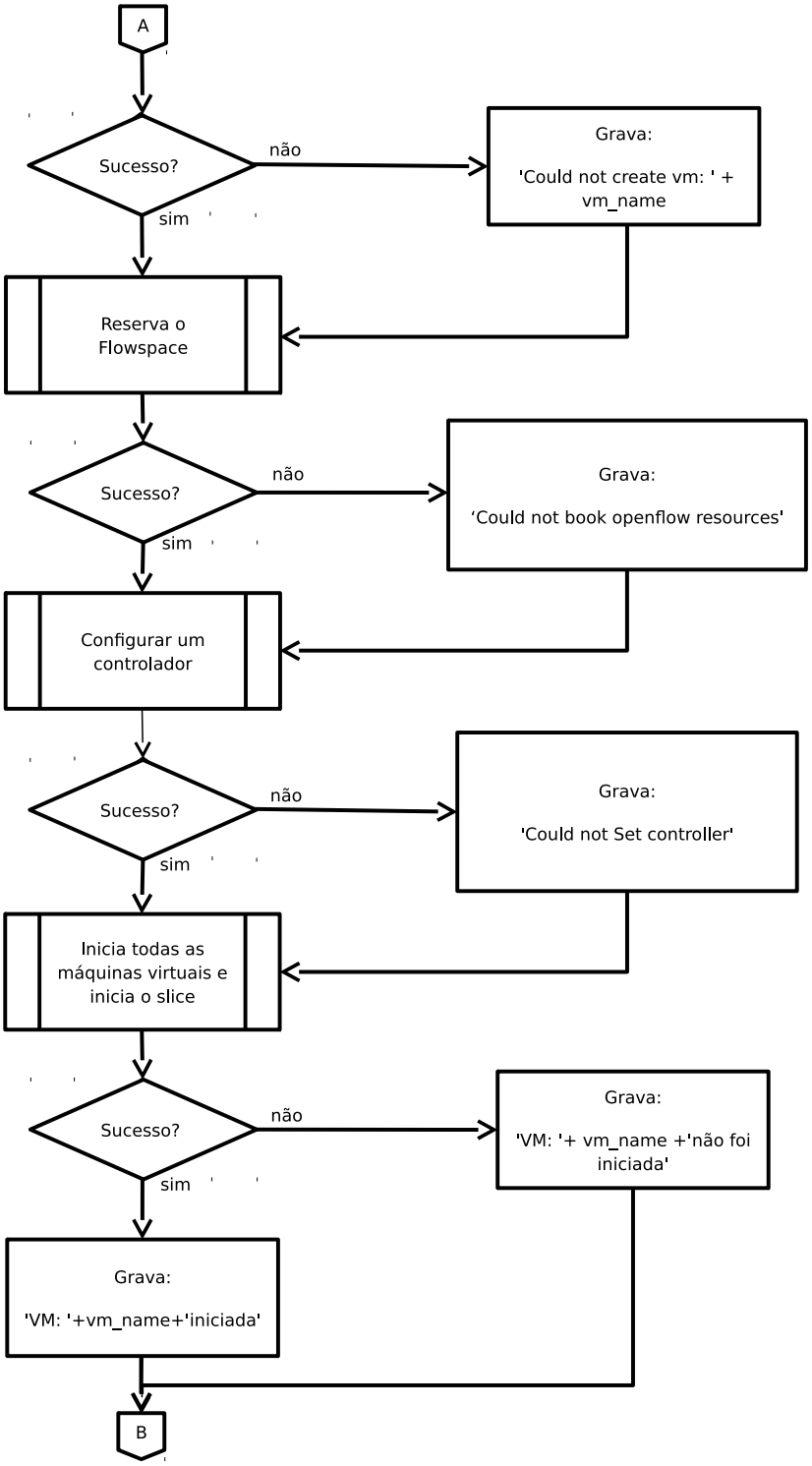


Figura 5.9: Fluxograma do agente de teste do OCF - Parte2.

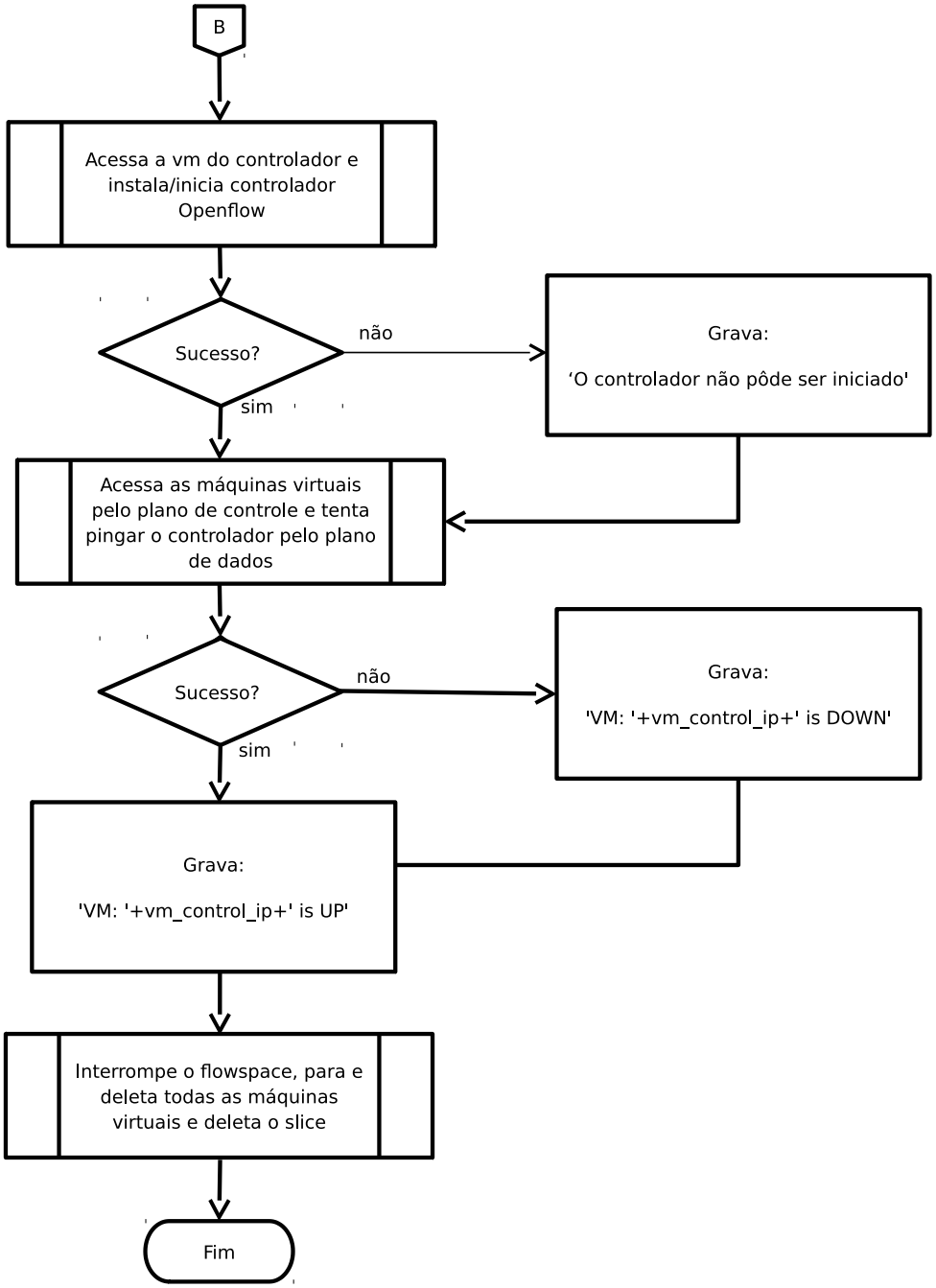


Figura 5.10: Fluxograma do agente de teste do OCF - Parte3.

5.7 Teste do plano de dados OpenFlow

O teste do plano de dados consiste na criação de um experimento de referência que utiliza as funcionalidades básicas oferecidas ao usuário, podendo assim diagnosticar problemas no serviço. Esse teste foi criado para suprir o vão deixado pelas ferramentas de monitoração tradicional.

O plano de dados é testado criando uma rede com um cliente ligado a cada um dos

switches do *slice*. Atraso, taxa máxima de transferência e velocidade da conexão com a Internet são testadas periodicamente. O arranjo experimental é baseado no tutorial de congestionamento TCP do FIBRE [111]. A Figura 5.11 mostra a topologia que foi configurada em cada ilha. Foi feito de forma a não haver *loops* e que todos as NetFPGAs estejam conectadas ao *switch* Pronto. A Figura 5.12 mostra o mesmo esquema, porém em um ambiente federado. A conexão entre cada uma das ilhas é testada.

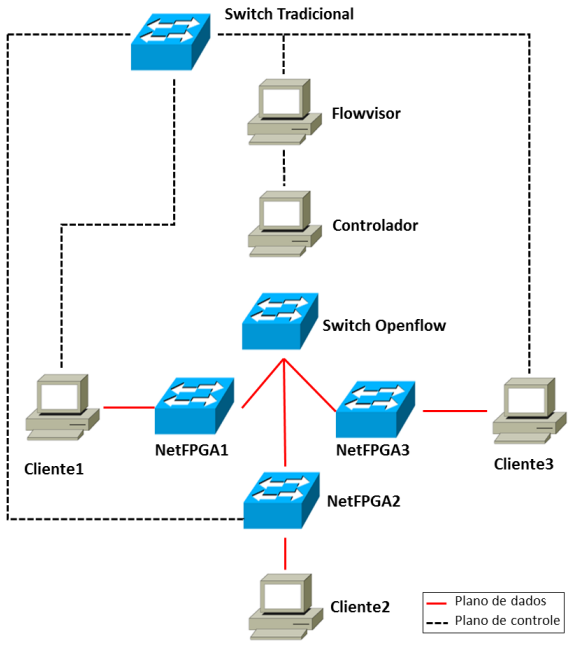


Figura 5.11: Topologia dos testes do plano de dados dentro de uma ilha.

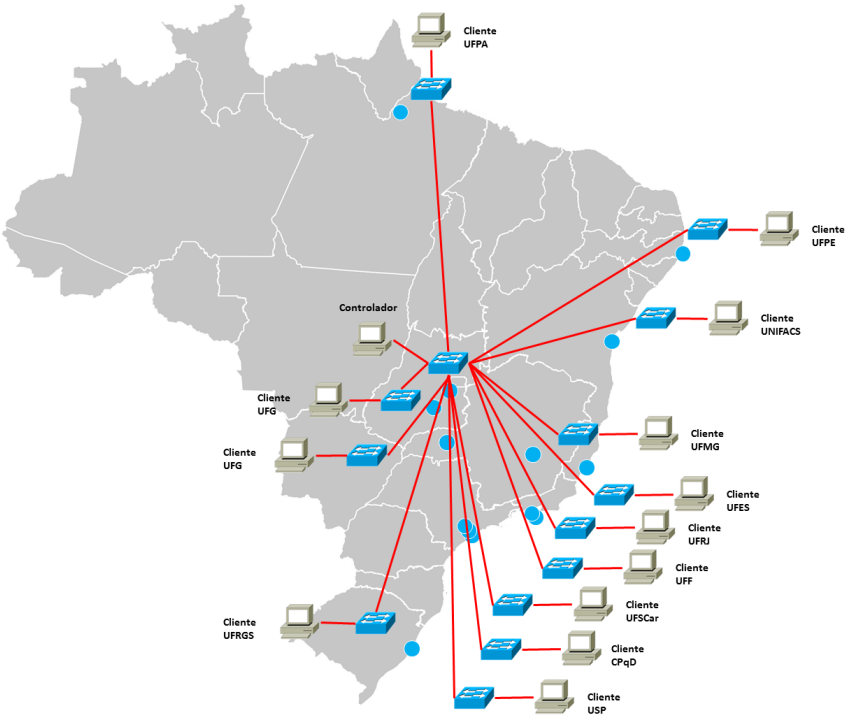


Figura 5.12: Topologia dos testes do plano de dados em um experimento federado.

5.8 Alarmes gerados

Para auxiliar os operadores a descobrir falhas e aplicar ações corretivas, alarmes são gerados e disponibilizados em um painel de alarmes da interface gráfica de usuário do FIBREOSS. A partir dos testes realizados pelo FIBREOSS e por outros sistemas de monitoração instalados no FIBRE são gerados alarmes avisando a existência de problemas.

A Tabela 5.1 mostra os eventos gerados pelos agentes de teste do FIBREOSS. Os eventos são apresentados na interface web na lista de eventos correntes. Essa lista mostra alarmes ativos. A telas de alarmes ativo é mostrada na captura de tela da Figura A.9.

Onde OK, indica que o teste foi realizado e bem sucedido, NOK indica que o teste foi a realizado e existe falha e INCONCLUSIVE indica que o teste não foi realizado com sucesso, e o estado do objeto testado é desconhecido.

São definidos 5 tipos de níveis de eventos:

- INFO - Não representa um problema, somente um *status* gerado.
- WARNING - Foi encontrado um problema, porém o mesmo não afeta os serviços.
- MINOR - O problema afeta a qualidade do serviço, mas não a disponibilidade.

Tabela 5.1: Relação de eventos gerados pelos agentes customizados do FIBREOSS.

Test ID	Return Code	Event level	Description
10001	NOK	MAJOR	Timeout Occurred
10002	NOK	MAJOR	Portal did not return 'OK'
10003	INCONCLUSIVE	WARNING	Did not show any nodes to reserve
10004	OK	INFO	Reservation OK
10005	NOK	MINOR	No nodes were reserved
10006	INCONCLUSIVE	WARNING	There is no reservation for now
10007	NOK	WARNING	Error while trying to turn ON the node
10008	OK	INFO	Node control interface is UP
10008	NOK	MAJOR	Node control interface is DOWN
20001	NOK	MAJOR	Timeout Occurred
20002	NOK	MINOR	Slice Could not be created
20003	OK	INFO	Expedient shows aggregate as Available
20003	NOK	MAJOR	Expedient shows aggregate as unavailable
20003	INCONCLUSIVE	WARNING	Could not determine aggregate state
20004	OK	INFO	Aggregate appears as available
20005	NOK	MINOR	Could not create VM in virtualization device
20006	NOK	MAJOR	Could not book OpenFlow resources
20007	OK	INFO	Controller was set
20007	NOK	MAJOR	Could not Set controller
20008	OK	INFO	Virtual machine is running
20008	NOK	WARNING	Virtual machine still has not started
20009	OK	INFO	Slice was started
20009	NOK	MAJOR	Could not start slice
20010	OK	INFO	Controller was configured and started
20010	NOK	MAJOR	Controller could not be configured and started
20010	INCONCLUSIVE	WARNING	Controller is not set
20011	OK	INFO	Device's control plane network interface is UP
20011	NOK	MAJOR	Device's control plane network interface is DOWN
20012	OK	INFO	Device's data plane network interface is UP
20012	NOK	MAJOR	Device's data plane network interface is DOWN
30001	NOK	MAJOR	Maddash is DOWN
40001	OK	INFO	Device is UP
40001	NOK	MAJOR	Device is DOWN
40101	NOK	MAJOR	Datapath connectivity lost
40102	NOK	MINOR	Datapath bandwidth under threshold
40103	NOK	MINOR	No internet connection for user VMs
40501	NOK	MAJOR	ZENOSS is DOWN

- MAJOR - O problema afeta a disponibilidade do serviço enquanto o alarme não for resolvido
- CRITICAL - Existe um problema que pode afetar a integridade física de equipamentos ou pessoas.

5.9 Proposta de relatório de disponibilidade

O objetivo desse módulo é atribuir notas às ilhas permitindo uma avaliação em nível gerencial sobre a disponibilidade dos serviços. Naturalmente, esse tipo de medida dá ênfase a certos aspectos enquanto encobre outros, devido à sua característica de resumo. Contudo, ela é interessante para a realização de um diagnóstico inicial da federação de recursos.

Foi proposto um modelo recursivo do cálculo em que os elementos do FIBRE são agrupados em serviços raiz e serviços afiliados. Cada serviço possui um determinado conjunto de componentes e serviços afiliados que dependem do serviço pai para funcionar. A Figura 5.13 mostra um exemplo em que o serviço ou grupo raiz ou serviço pai é a infraestrutura da ilha e que dá suporte a dois serviços que são de interesse direto dos usuários do *testbed*: o *testbed* de rede sem-fio e o *testbed* cabeado OpenFlow.

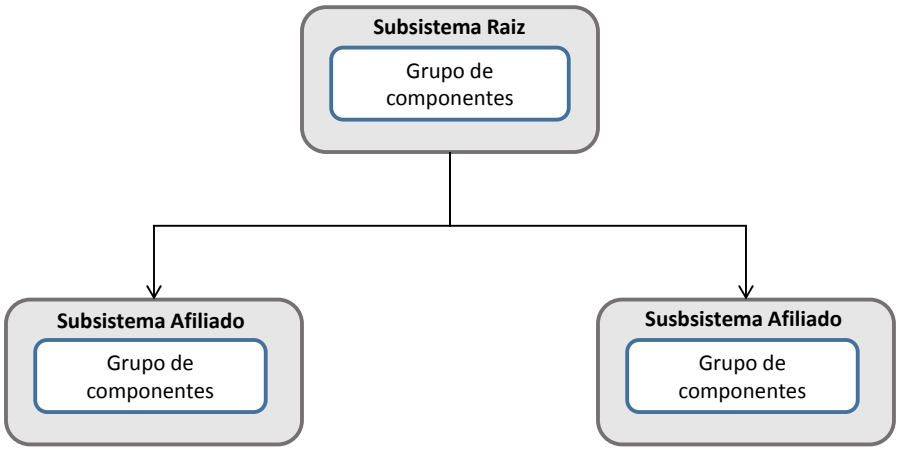


Figura 5.13: Modelo de dependência simplificado.

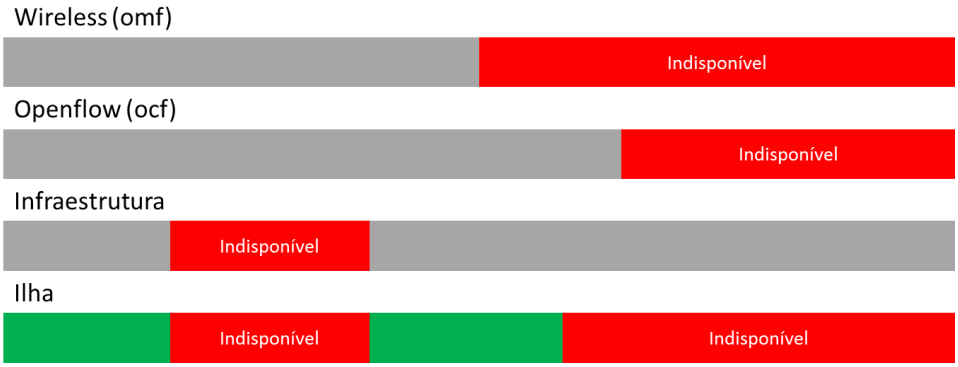


Figura 5.14: Exemplo do cálculo da disponibilidade de uma ilha.

A Figura 5.14 mostra o cálculo da disponibilidade proposto no domínio do tempo. Se a infraestrutura falhar, todos os serviços que dela dependem também ficam indisponíveis. Se um serviço afiliado como o OCF falha, a percepção dos usuários que querem fazer experimentos com OpenFlow é de que a ilha está indisponível, enquanto que para os usuários que querem realizar experimentos com rede sem-fio é de que a ilha está funcional. Supondo que a demanda pelos serviços seja idêntica, é razoável dizer que a percepção da disponibilidade da ilha foi de 50% nesse período. Daí o uso da média para o cálculo.

A média foi escolhida na proposta por permitir o resumo em um único número a disponibilidade do *testbed*, tornando a medida mais simples, porém pode ser dividida em duas métricas que separem o *testbed* OpenFlow do *testbed* sem-fio, pois dificilmente o mesmo usuário vai fazer experimentos nos dois domínios. Assim, pode ser ruim juntar as duas medições, já que um usuário interessado em realizar experimentos sem-fio, por exemplo, não se importa com o estado do *testbed* OpenFlow.

$$grupo_{disp} = minimo(componentes, media(filhos)) \tag{5.1}$$

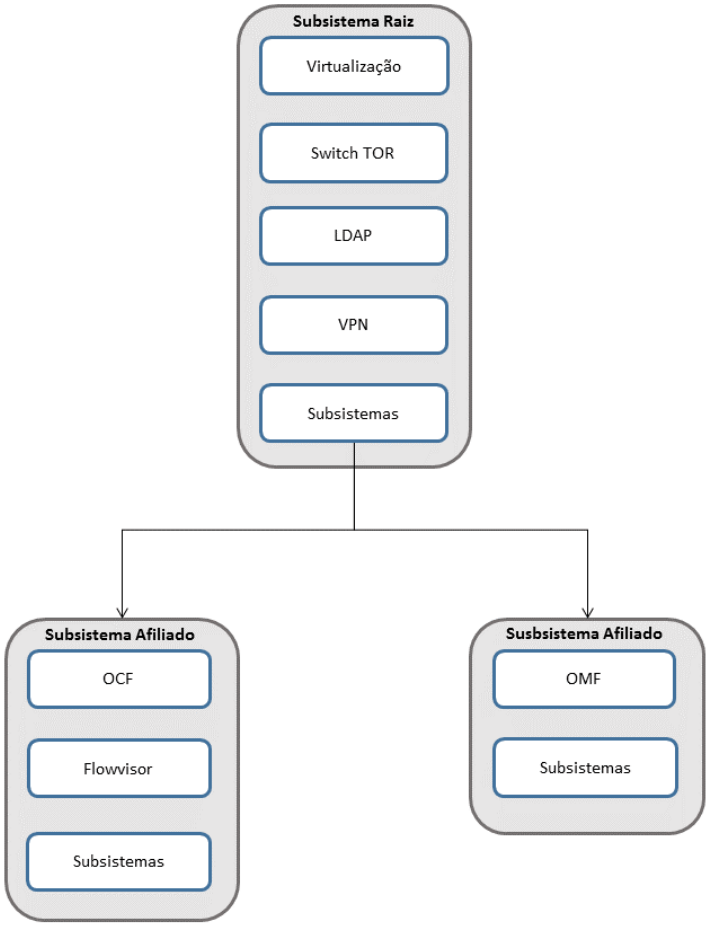


Figura 5.15: Exemplo do diagrama em alto nível de dependências com componentes internos.

A Figura 5.15 mostra uma visão mais detalhada. Os serviços e seus componentes são incluídos diagrama. Na Figura 5.16, o cálculo da disponibilidade da infraestrutura é mostrado em função do tempo. Considera-se que todos os componentes considerados no exemplo estejam em série, pois se um parar, todo o serviço fica indisponível. Assim, alguns componentes devem ser avaliados como um arranjo em série e outros componentes como um arranjo paralelo.

Figura 5.16: Exemplo do cálculo da disponibilidade do serviço/grupo infraestrutura.

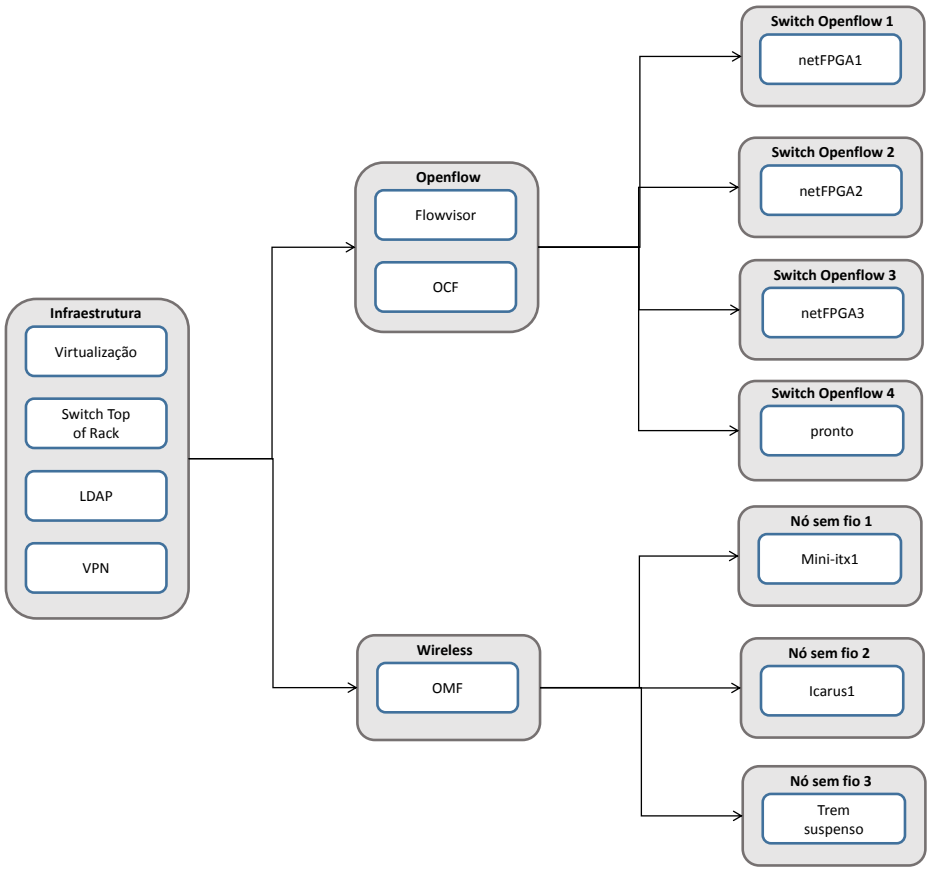


Figura 5.17: Árvore de dependência detalhada, mostrando serviços e componentes.

A Figura 5.17 mostra a mesma árvore incluindo todos os componentes atualmente instalados em uma ilha típica do FIBRE. A Tabela 5.2 mostra a hierarquia de serviços baseada na árvore de serviços.

Serviço ou subsistema aqui é compreendido como um conjunto de componentes, em

que todos os componentes que não possuem redundâncias precisam estar funcionais para que o serviço seja considerado disponível. Os blocos OpenFlow e *Wireless* na Figura 5.17 são dependentes diretos da infraestrutura e são dois serviços independentes. O serviço de *testbed* OpenFlow, por sua vez, também tem serviços afiliados, os *switches* OpenFlow, incluindo o *switch* Pronto e as NetFPGAs, que também são serviços independentes. É dito aqui que dois serviços são independentes, quando a falha em um deles não afeta o serviço irmão, porém há perda de funcionalidade. Um cenário parecido é aplicado ao bloco *wireless*: ele possui serviços afiliados contendo vários nós sem-fio.

Por exemplo, se a infraestrutura falha, todos os serviços dependentes dele também irão falhar. Se, em outra situação, um serviço como o OCF falha, os usuários precisando fazer experimentos com o OpenFlow perceberão o *testbed* como indisponível, enquanto que os usuários interessados em experimentos de rede sem-fio perceberão o *testbed* como funcional. Assumindo que a demanda seja idêntica para esses serviços, em caso de falha no OCF, por exemplo, é razoável dizer que o *testbed* está 50% disponível neste período. Isso justifica o uso da média como medida para serviços independentes. Quando é calculada a disponibilidade de um serviço em função de seus componentes, a metodologia tradicional é usada: componentes essenciais que não possuem redundância serão considerados em série com o resto do sistema (lógica E), e componentes com redundância serão considerados em paralelo com o seu par (lógica OU). Já que não há nenhuma redundância de componentes implementada atualmente no *testbed* do FIBRE, a fórmula simplesmente usa o mínimo da disponibilidade dos componentes na fatia de tempo analisada.

Tabela 5.2: Hierarquia dos serviços de uma ilha do FIBRE

Servidor	**Grupo do serviço**
vpn.uff.fibre.org.br	/Infraestrutura
tor.uff.fibre.org.br	/Infraestrutura
virt.uff.fibre.org.br	/Infraestrutura
pronto.uff.fibre.org.br	/Infraestrutura/OpenFlow
perfsonar1.uff.fibre.org.br	/Acessório/Perfsonar
perfsonar2.uff.fibre.org.br	/Acessório/Perfsonar
omf.uff.fibre.org.br	/Infraestrutura/Wireless
ocf.uff.fibre.org.br	/Infraestrutura/OpenFlow
netfpga1.uff.fibre.org.br	/Infraestrutura/OpenFlow/netFPGA
netfpga2.uff.fibre.org.br	/Infraestrutura/OpenFlow/netFPGA
netfpga3.uff.fibre.org.br	/Infraestrutura/OpenFlow/netFPGA
ldap.uff.fibre.org.br	/Infraestrutura
flowvisor.uff.fibre.org.br	/Infraestrutura/OpenFlow
dns.uff.fibre.org.br	/Acessório

Dessa forma, todos os componentes de um determinado serviço são considerados em série. Na ocorrência de uma falha o serviço inteiro ficará indisponível.

Usando esse mapa de dependência dos serviços e componentes de uma ilha e os alarmes gerados o operador pode diagnosticar qual componente está causando a indisponibilidade do serviço.

5.9.0.1 Exemplos de cálculo

Na Figura 5.18, uma NetFPGA fica fora de serviço. O componente NetFPGA1 fica indisponível, assim a disponibilidade do serviço NetFPGA1 passa a ser zero. Como foi dito anteriormente, esses serviços irmãos são independentes, a falha de uma NetFPGA não atrapalha o funcionamento das outras. O domínio OpenFlow está com todos os seus componentes funcionando, portanto a experimentação é possível, mas com perda de funcionalidade. Portanto fica estabelecido que 75% do domínio OpenFlow está disponível. Então a disponibilidade da ilha será a média entre 75% e 100%.

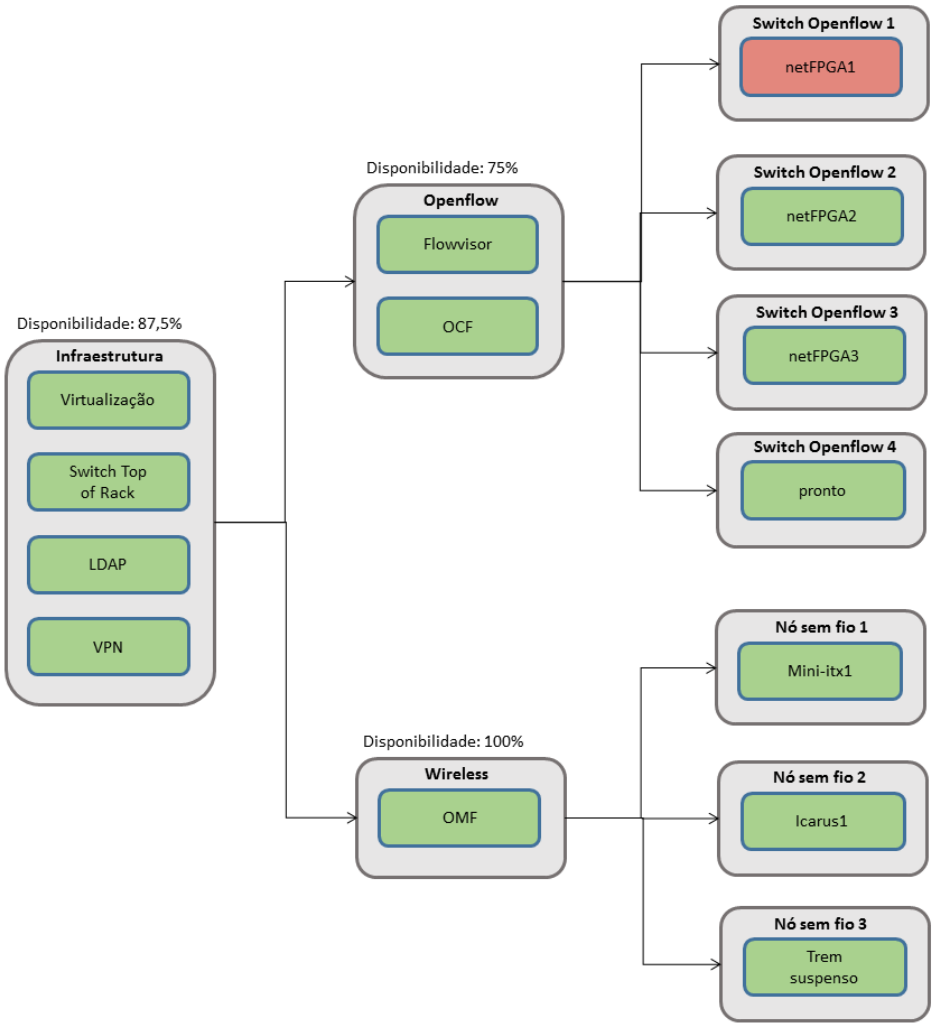


Figura 5.18: Exemplo de cálculo da disponibilidade com uma NetFPGA indisponíveis.

Na Figura 5.21, existe uma situação parecida com a da Figura 5.18, porém existem duas NetFPGAs indisponíveis, o que torna a disponibilidade do domínio OpenFlow 50%. Os dois nós sem-fio indisponíveis fazem com que a disponibilidade do domínio sem-fio fique em 33,3%. Assim, sabendo que os domínios OpenFlow e sem-fio são dois serviços independentes, fica estabelecido que a ilha está 41,7% disponível no instante do cálculo.

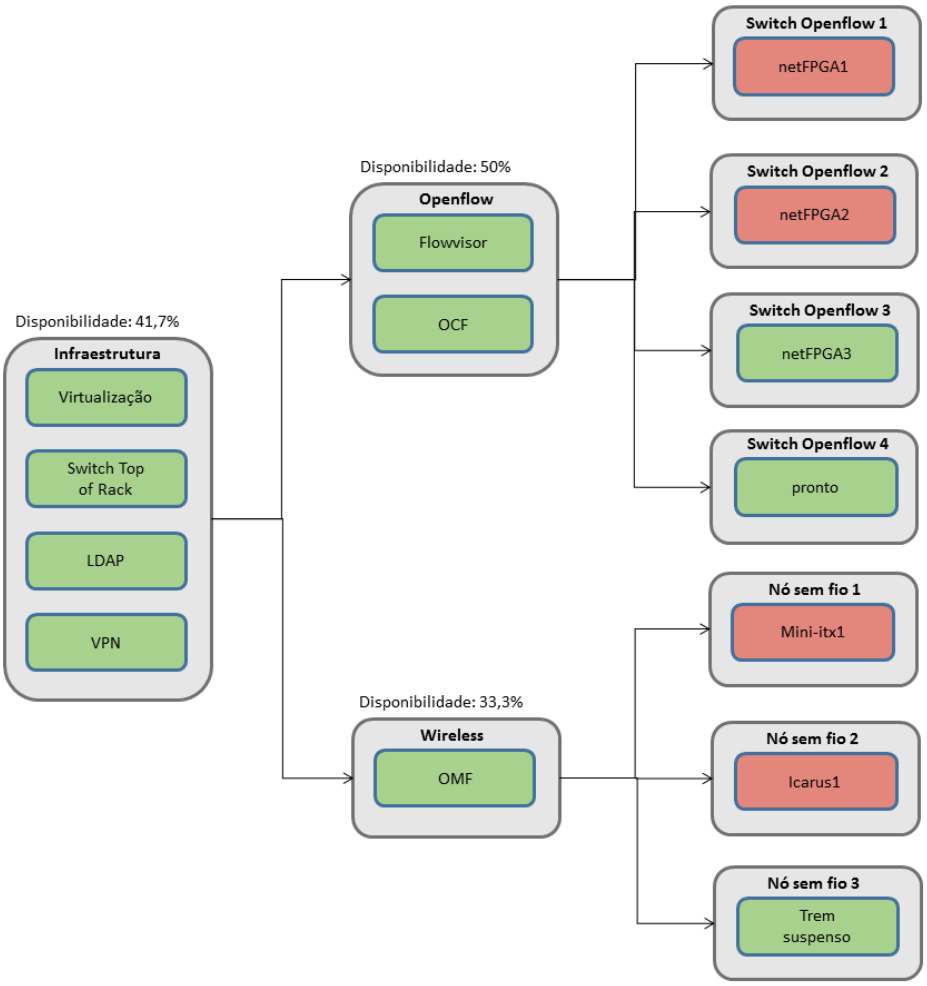


Figura 5.19: Exemplo de cálculo da disponibilidade com duas NetFPGAs e dois nós sem-fio indisponíveis.

Na Figura 5.20, os domínios sem-fio e OpenFlow estão plenamente funcionais, porém se um ou mais componentes essenciais da infraestrutura ficam indisponíveis, a ilha não pode ser utilizada pelos usuários. A conexão com a VPN indisponível impede que usuários acessem o *testbed* através do portal do FIBRE. Uma falha do LDAP impede a autenticação do usuário, impedindo seu acesso. Vale notar que qualquer experimento que já esteja em execução não será afetado nesse caso. As conexões no plano de dados continuam ativas, somente com uma possível perda de conexão com a Internet.

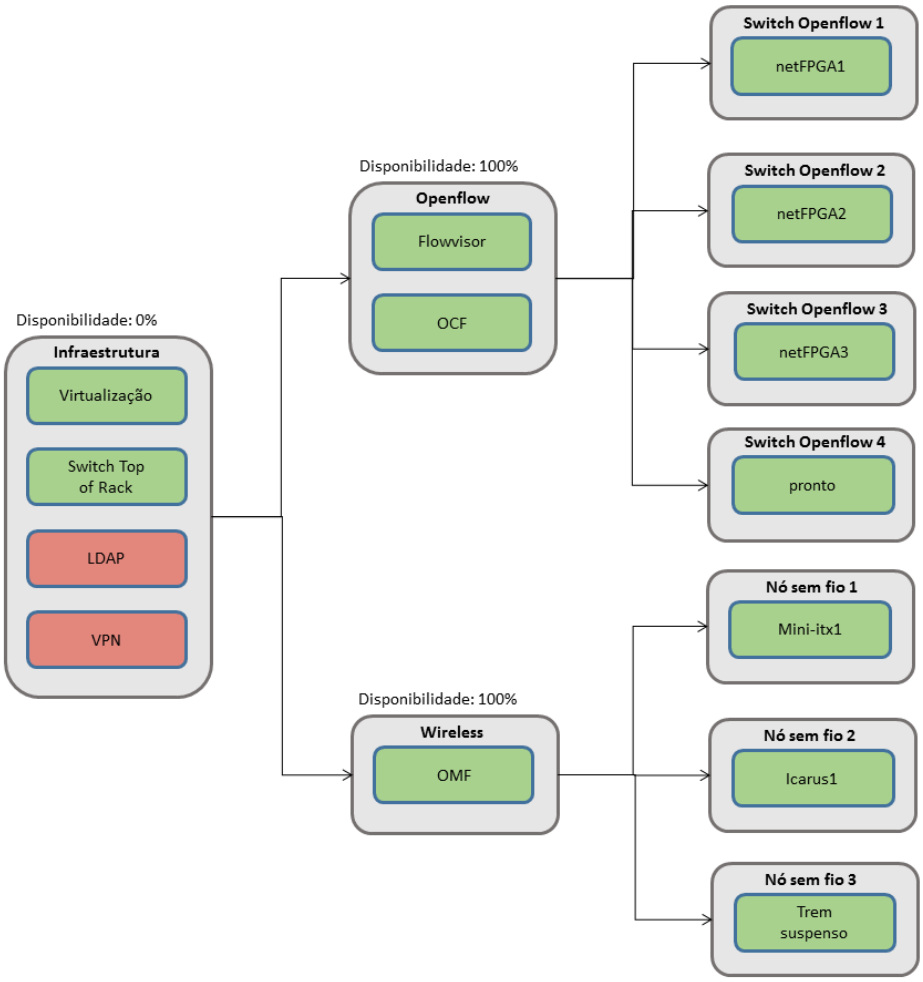


Figura 5.20: Exemplo de cálculo da disponibilidade com LDAP e VPN indisponíveis.

Na Figura 5.22, O domínio sem-fio está totalmente disponível e todos os *switches* OpenFlow estão disponíveis, porém um componente essencial do domínio OpenFlow está indisponível. O Flowvisor é um *proxy* transparente para os controladores OpenFlow e sem ele não é possível realizar nenhum experimento. Por isso a disponibilidade do domínio OpenFlow é 0%. Assim, a ilha está 50% disponível, pois 1 de seus dois domínios de experimentação está indisponível.

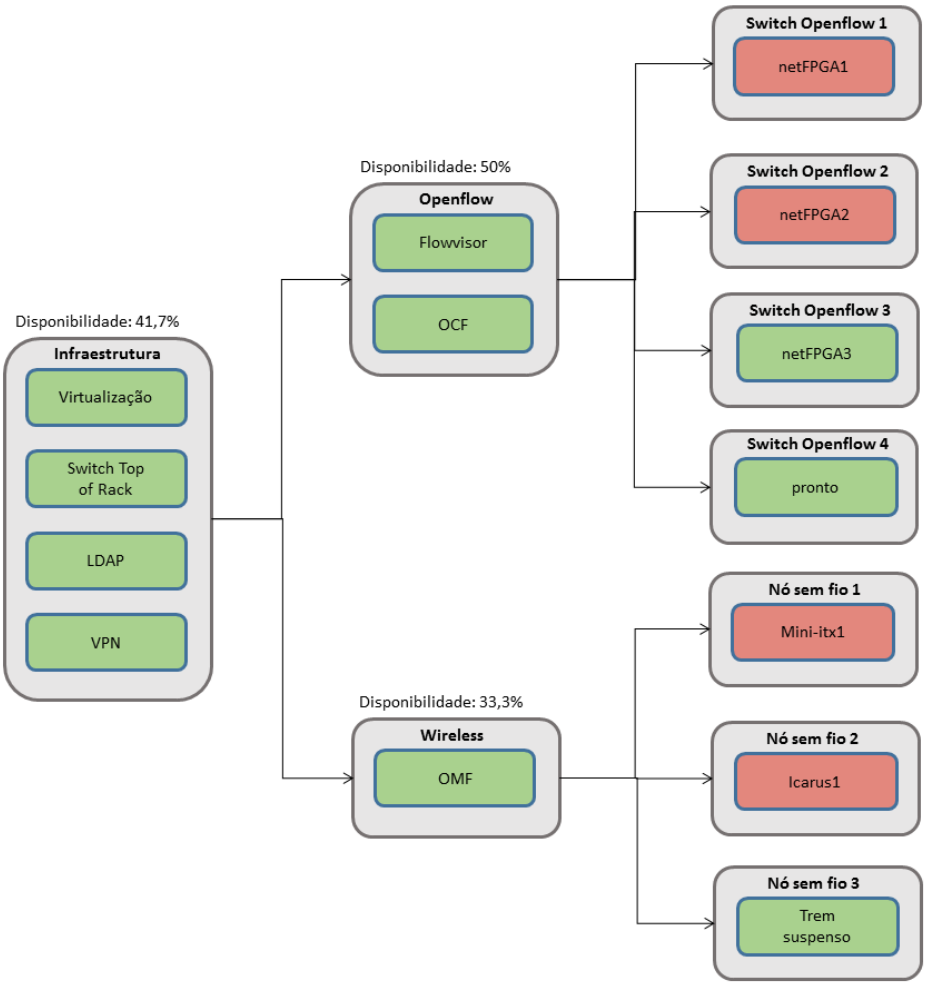


Figura 5.21: Exemplo de cálculo da disponibilidade com duas NetFPGAs e dois nós sem-fio indisponíveis.

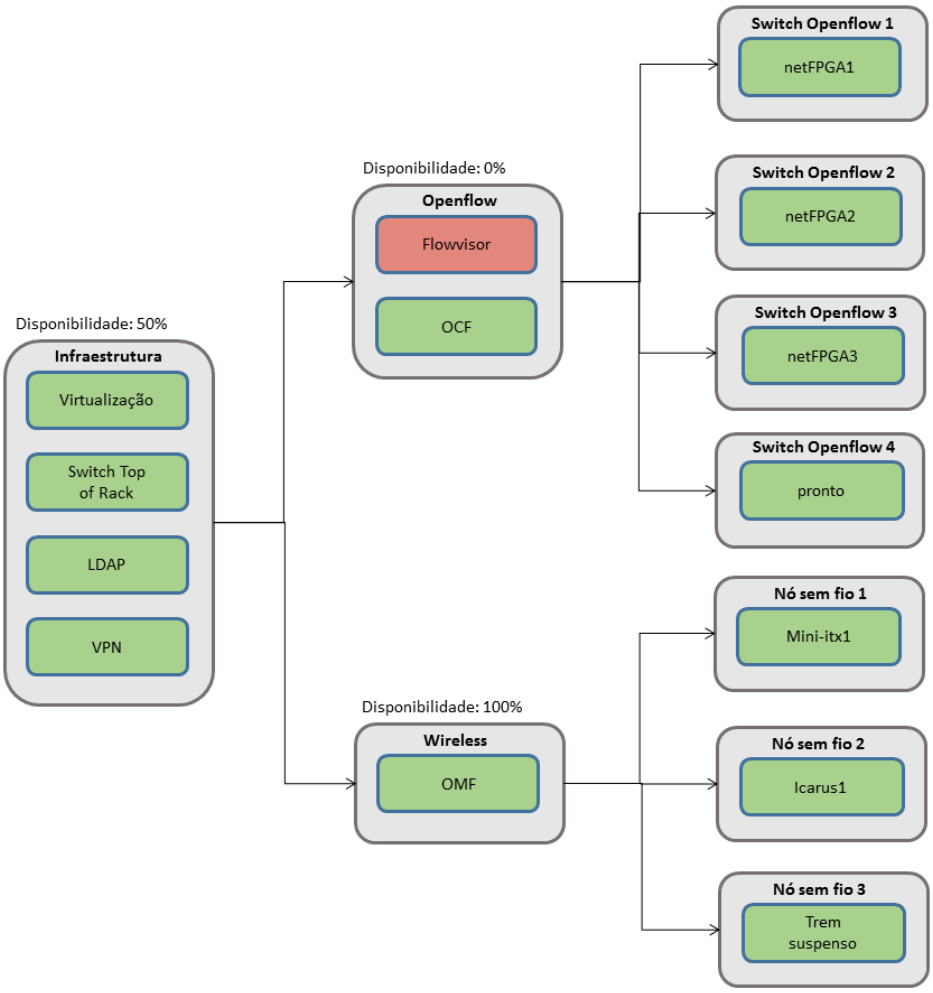


Figura 5.22: Exemplo de cálculo da disponibilidade com o Flowvisor indisponível.

Capítulo 6

Resultados

Foi realizada a implantação do sistema de monitoração da infraestrutura FIBREOSS com o objetivo de verificar o estado de funcionamento do *testbed* com mais precisão do que as ferramentas tradicionais conseguiam oferecer. Portanto, usando o método de cálculo da disponibilidade apresentado na seção 5.9 é feita uma comparação entre o teste de ping realizado pelo Zenoss e usando os testes do FIBREOSS na seção 6.1. O ambiente de testes utilizado é o *testbed* FIBRE e as medições são feitas apenas nas ilhas ativas. As medições adicionais realizadas no plano de dados do domínio OpenFlow foram o teste de ping e teste de banda máxima do canal utilizando a ferramenta iperf transmitindo a maior quantidade de dados possível no tempo determinado[1]. Detalhes da experimentação no plano de dados foram explicados na seção 5.7. Já no domínio sem-fio, testes de funcionalidade básica no portal da ilha e do OMF, além de testes dos nós sem-fio foram realizados. Esses testes permitiram constatar o funcionamento do serviço de experimentação do *testbed*. Detalhes desses testes foram explicados na Seção 5.5. Os relatórios foram produzidos usando dados do FIBREOSS e cruzados com os dados das outras plataformas de gerência. Inconsistências são marcadas no relatório para que sejam esclarecidas pelos operadores.

6.1 Análise

Para determinar se os serviços básicos do *testbed* estavam funcionais, foram realizados testes adicionais, a disponibilidade foi calculada e um relatório foi gerado para fins de controle do Service Level Agreement (SLA). Os testes apresentados foram feitos no período de 17 de setembro a 1º de outubro de 2016 na seção 6.1.1. Em seguida, uma análise dos problemas mais frequentes é apresentada na Seção 6.1.2

[1] 5 segundos foi o tempo utilizado como parâmetro

Todas as medições no plano de dados possuem um vão no dia 19/09/2016, devido à uma falha de energia que interrompeu a monitoração. Nos gráficos de disponibilidade o período é mostrado como tendo disponibilidade constante.[2]

Os gráficos de disponibilidade mostram duas curvas comparando a disponibilidade medida com as ferramentas tradicionais e as medidas realizadas com o FIBREOSS. Já os gráficos que mostram os testes do plano de dados do domínio OpenFlow mostram a taxa máxima de transferência entre uma VM de experimentação que está ligada a um *switch* OpenFlow da ilha para a VM onde funciona o controlador OpenFlow do *slice* onde o experimento está funcionando. Nem todas as falhas puderam ser investigadas de maneira aprofundada. Nesses caso é feita somente uma descrição geral da falha.

6.1.1 Disponibilidade

6.1.1.1 RNP

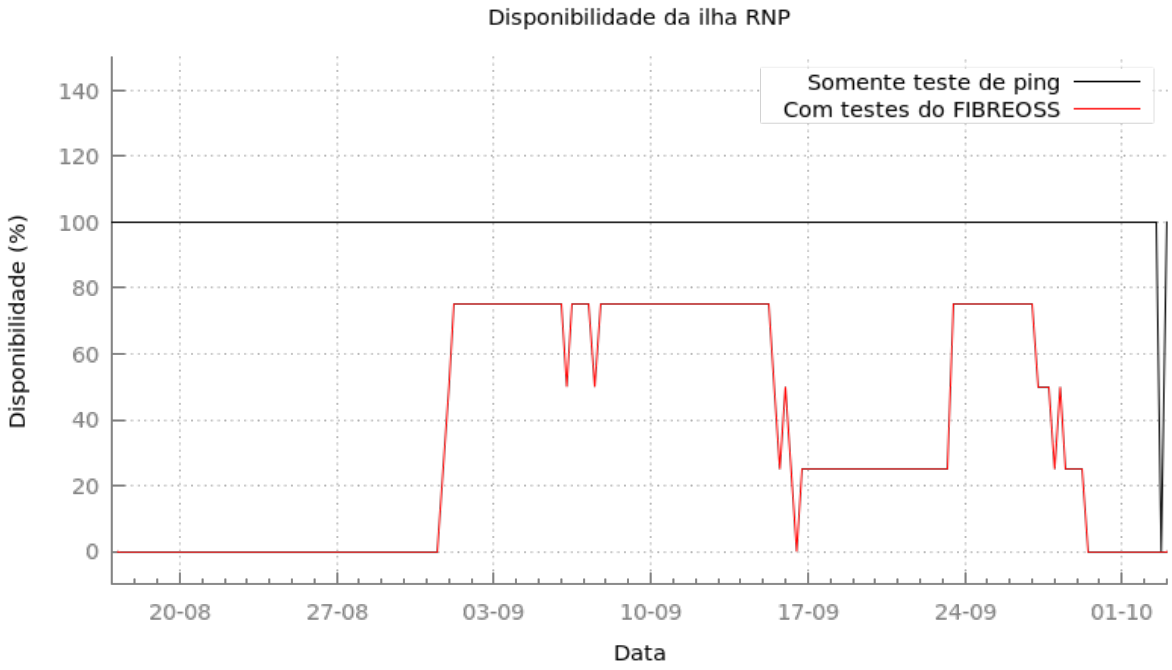


Figura 6.1: Comparação de medições de disponibilidade na ilha da RNP.

Do início do período de medição até o dia 1º de setembro as VMs de usuário estavam sendo deletadas após a criação, derrubando a disponibilidade para 0%. A disponibilidade atinge 75% no período seguinte [3] e em 16/09/2016 as VMs ligadas as NetFPGAs 1 e 3 perderam conectividade no plano de dados. (Figuras 6.2 e 6.3). A VM conectada

[2] A disponibilidade do dia anterior é repetida
[3] somente a VM conectada à NetFPGA3 não funcionava

a NetFPGA2 e a VM do controlador funcionaram nessa semana (Figura 6.4).As VMs foram re-estabelecidas e houve perda de conectividade no dia 27/09/2016. O pico em 0% observado no dia 02/10/2016 foi causado por uma perda de conectividade com o Flowvisor.

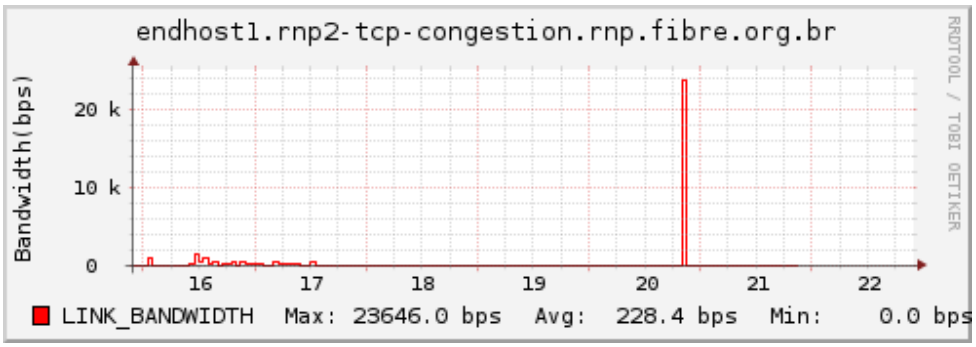


Figura 6.2: Teste de transferência no plano de dados da VM conectada à NetFPGA1 na ilha da RNP.

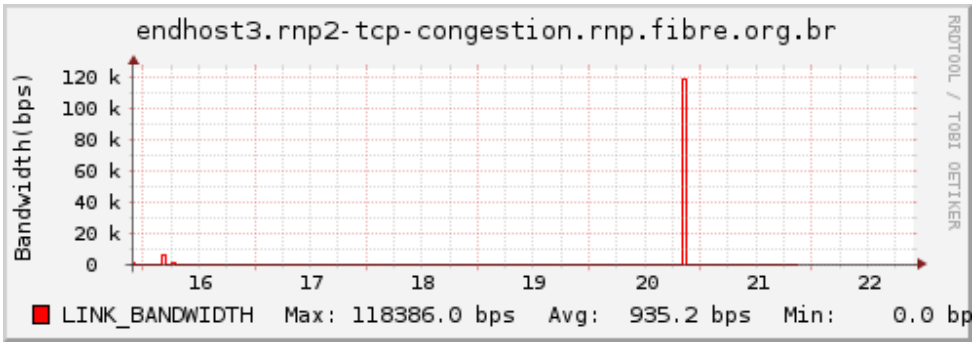


Figura 6.3: Teste de transferência no plano de dados da VM conectada à NetFPGA3 na ilha da RNP.

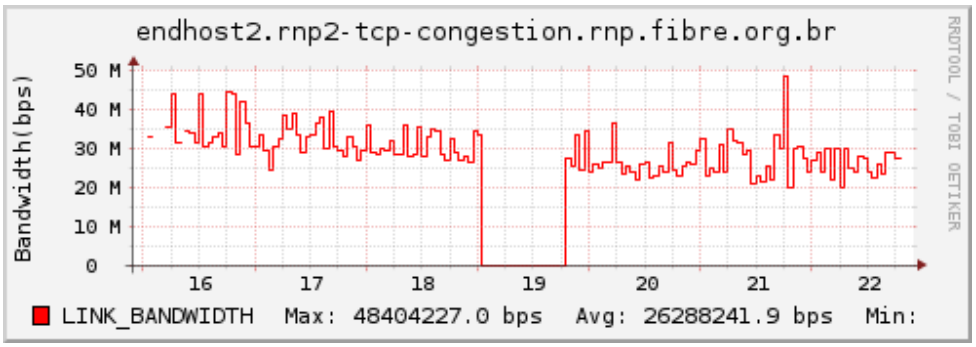


Figura 6.4: Teste de transferência no plano de dados da VM conectada à NetFPGA2 na ilha da RNP.

6.1.1.2 NOC

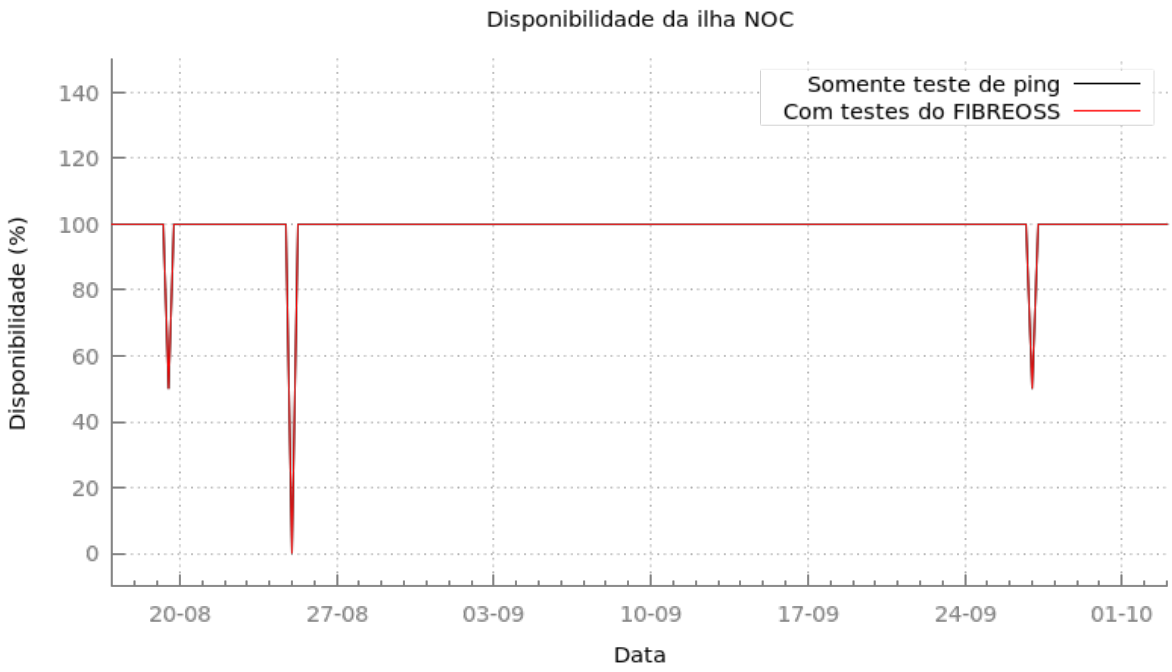


Figura 6.5: Comparação de medições de disponibilidade na ilha do NOC.

A ilha do NOC não possui recursos de experimentação e serve como portal para experimentos federados. É uma ilha de referência e teve poucas falhas e de curta duração no período. Em 19/08/2016 o portal do OMF ficou fora do ar, reduzindo a disponibilidade da ilha para 50%. No dia 25/08/2016 o LDAP ficou fora do ar, e em 27/09/2016 o Flowvisor ficou sem conectividade.

6.1.1.3 USP

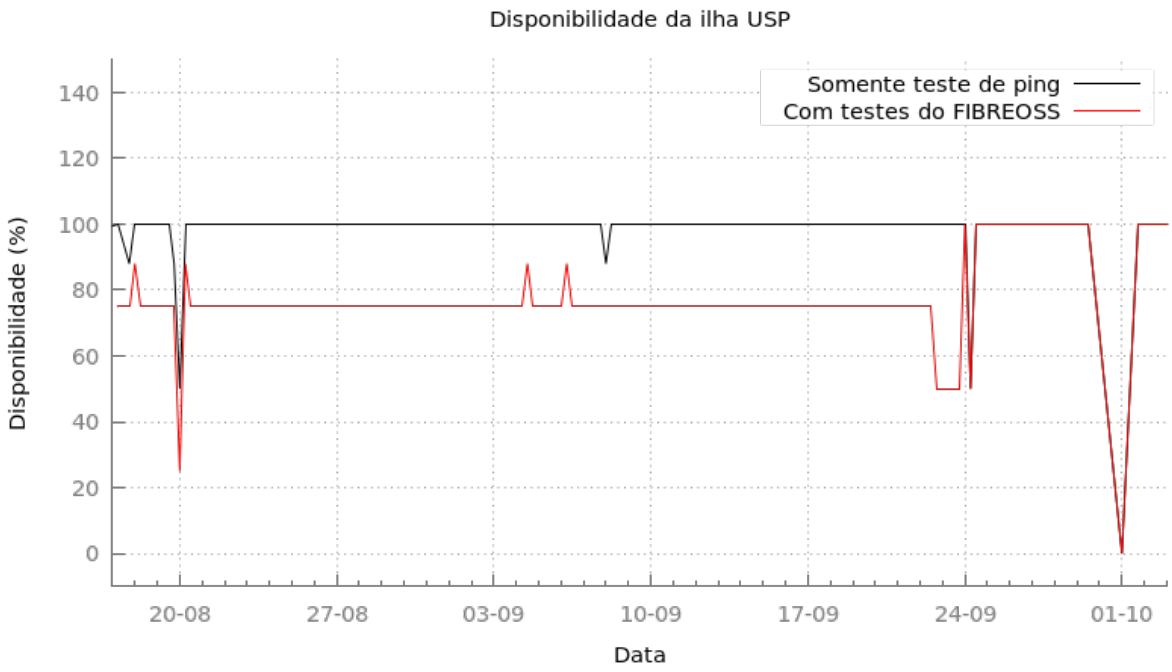


Figura 6.6: Comparação de medições de disponibilidade na ilha da USP.

A disponibilidade oscila em torno dos 75% com alguns eventos de indisponibilidade pontuais: em 20/08/2016 e 24/09/2016 o OMF fica indisponível e em 01/10/2016 o servidor de virtualização ficou indisponível. Do período inicial da medição até o dia 25/09/2016 (Figuras 6.7, 6.8 e 6.9 a VM3 fica indisponível. No dia 22/09/2016 as VMs 1 e 2 ficam indisponíveis (Figuras 6.11 e 6.10) e todas as VM voltam a funcionar após recriação do *slice*.

Nas Figuras 6.7, 6.8 a taxa máxima de transferência fica limitada a 800bps. Esse problema foi observado durante o período de medição e será discutido na seção 6.2.1.

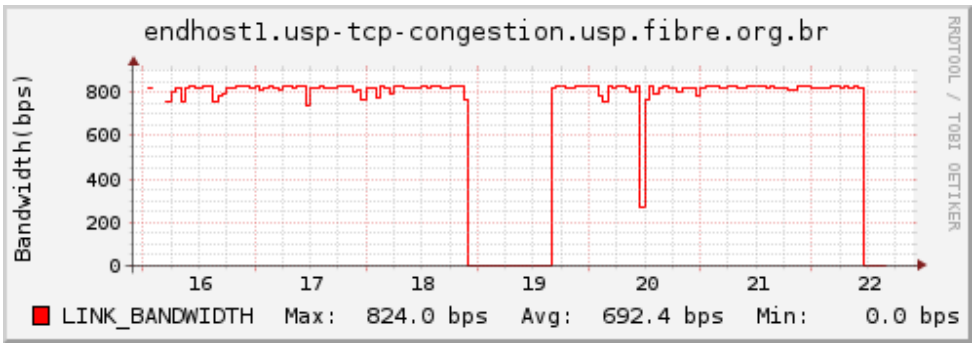


Figura 6.7: Teste de transferência no plano de dados da VM conectada à NetFPGA1 na ilha da USP.

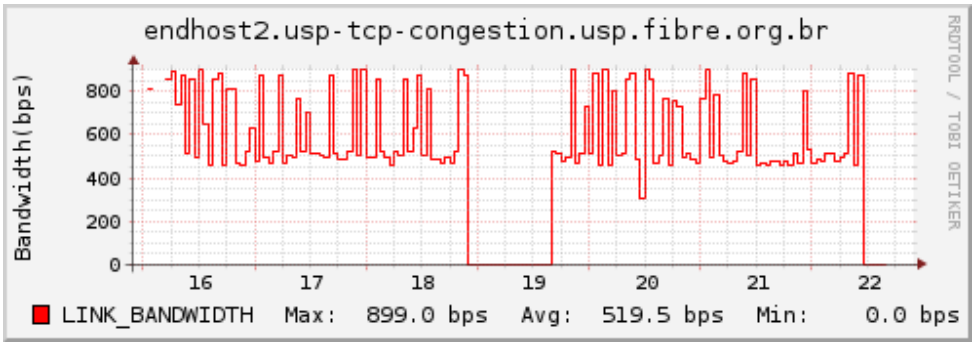


Figura 6.8: Teste de transferência no plano de dados da VM conectada à NetFPGA2 na ilha da USP.

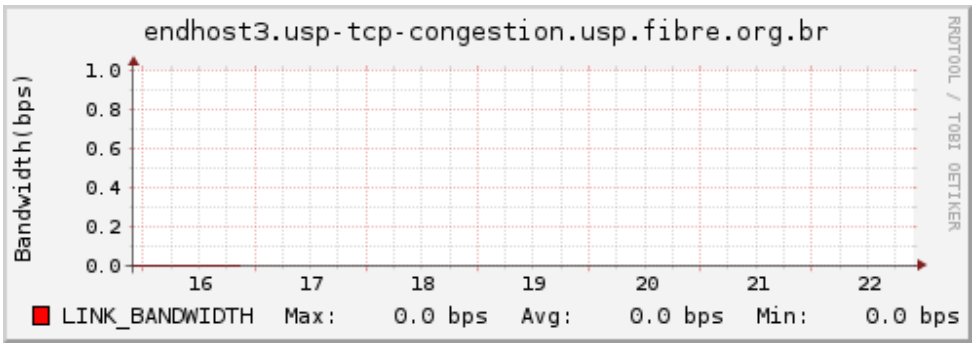


Figura 6.9: Teste de transferência no plano de dados da VM conectada à NetFPGA3 na ilha da USP.

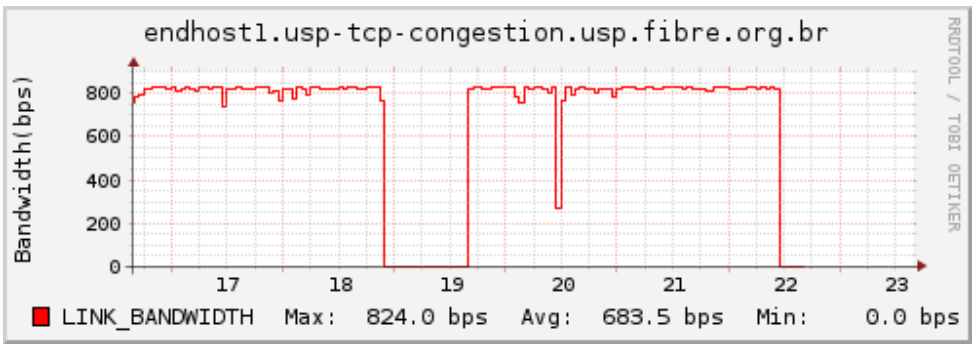


Figura 6.10: Teste de transferência no plano de dados da VM conectada à NetFPGA3 na ilha da USP.

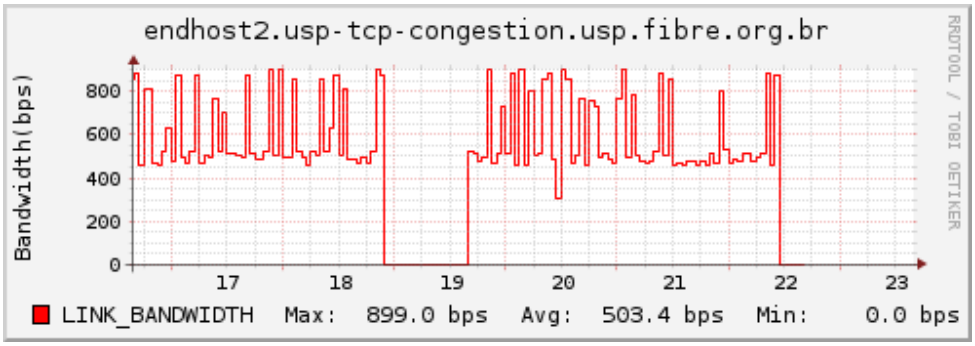


Figura 6.11: Teste de transferência no plano de dados da VM conectada à NetFPGA3 na ilha da USP.

6.1.1.4 UFF

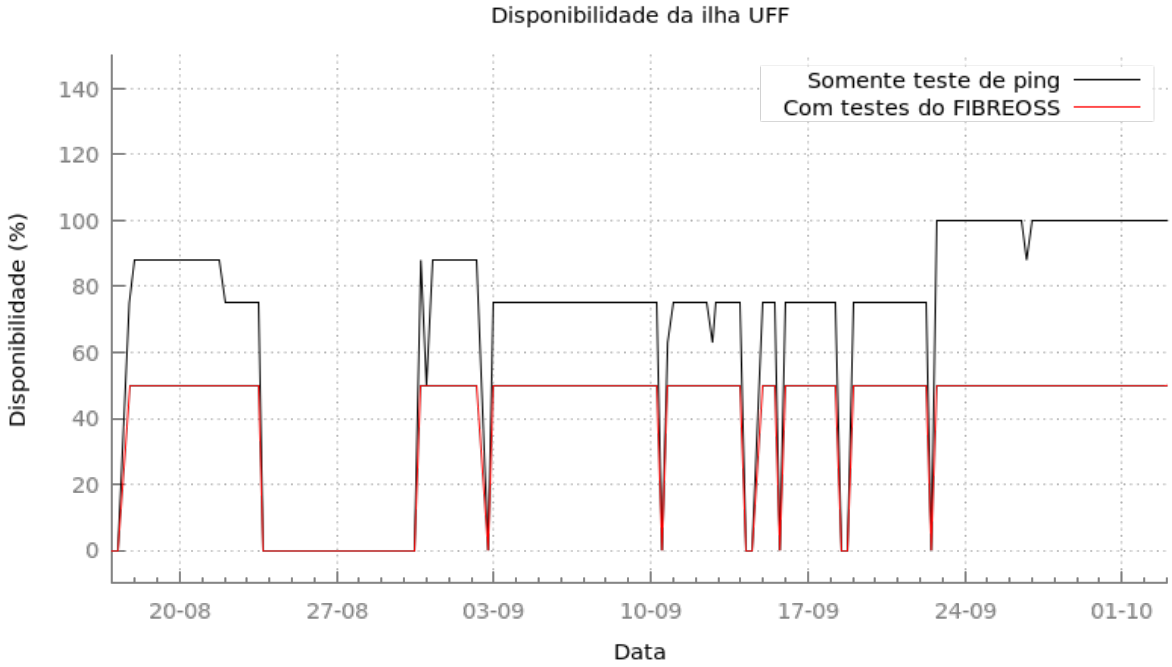


Figura 6.12: Comparação de medições de disponibilidade na ilha da UFF.

Durante o período de medição, a ilha da UFF estava recuperando-se de uma falha de disco e precisou ser reconfigurada. Durante todo o período de medição o domínio OpenFlow ficou indisponível. A disponibilidade é mantida em 50% pelo domínio sem-fio que ficou operacional. A UFF está conectada à FIBRENET através uma VPN, por isso apresenta muitos eventos pontuais de indisponibilidade causados por falhas da internet da universidade. Outros casos são causados por faltas de energia. O evento de 23/09/2016 a 30/09/2016 foi causado por uma perda de conectividade com a Internet durante uma manutenção do laboratório MIDIACOM [112]. Outras falhas são falta de energia, que são muito frequentes na UFF.

6.1.1.5 UFG

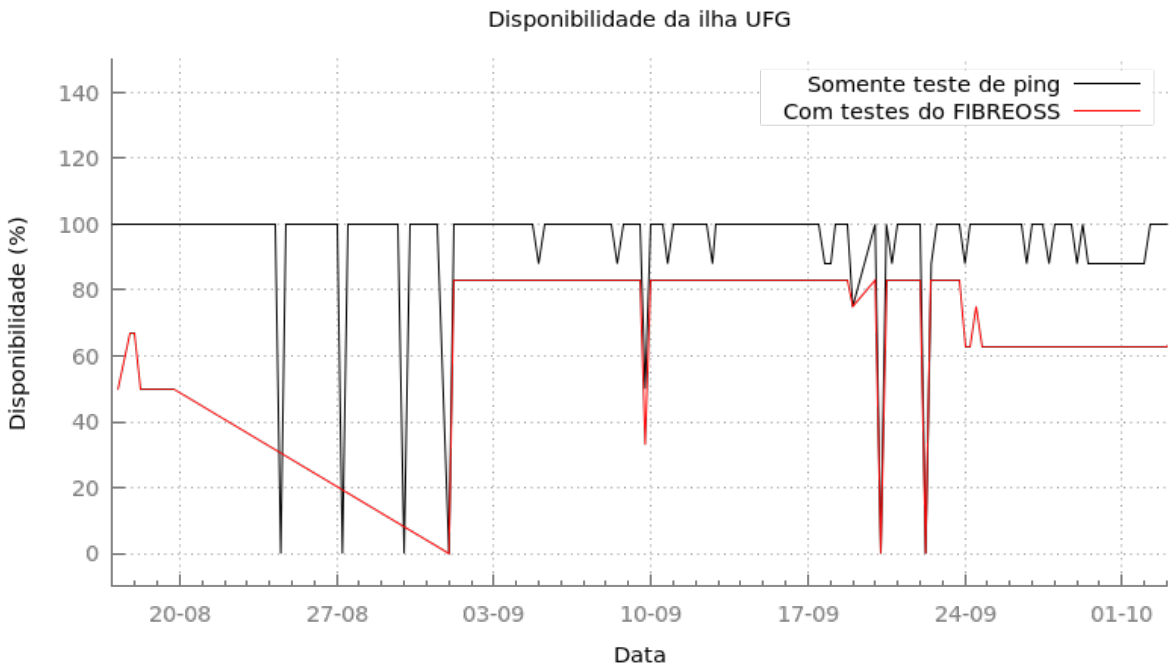


Figura 6.13: Comparação de medições de disponibilidade na ilha da UFG.

A ilha da UFG funcionou bem a maior parte do tempo, somente a VM3 não teve conectividade. Várias falhas pontuais do servidor do LDAP e do *switch* TOR derrubaram a disponibilidade da ilha, por serem serviços essenciais, porém não afetaram o plano de dados. No período de 20/08/2016 a 01/09/2016 não fio possível testar o plano de dados pois o experimento foi apagado. Um problema no serviço de virtualização fez com que as VMs fossem perdidas. Após esse período a medição é retomada. As Figuras 6.14 e 6.14 mostram o período de 16/09/2016 a 22/09/2016 e sem interferência na transmissão de dados nos dias 20/09/2016 3 22/09/2016.

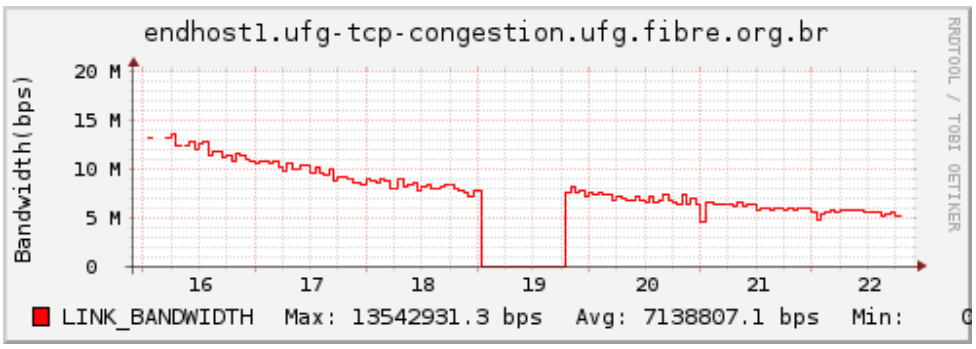


Figura 6.14: Teste de transferência no plano de dados da VM conectada à NetFPGA1 na ilha da USP.

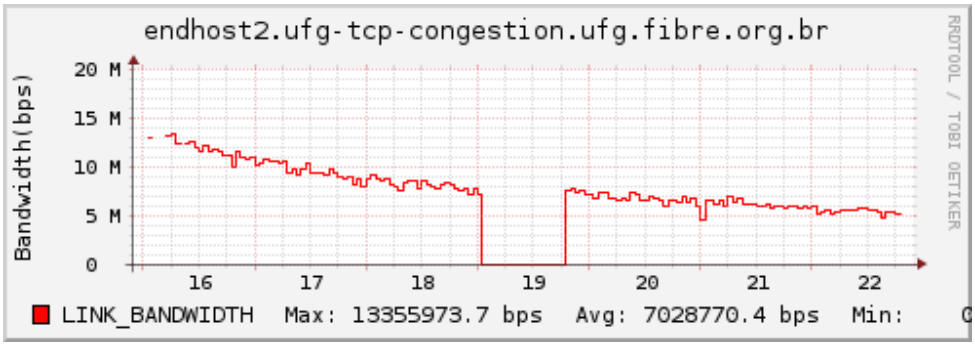


Figura 6.15: Teste de transferência no plano de dados da VM conectada à NetFPGA2 na ilha da USP.

6.1.1.6 UFPA

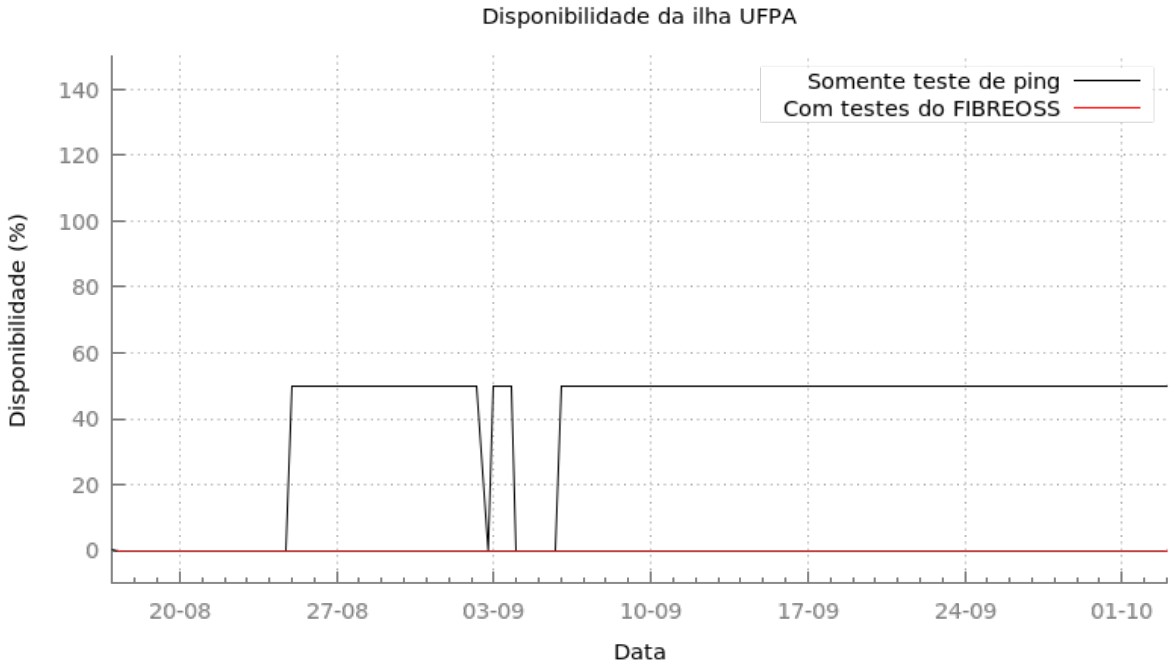


Figura 6.16: Comparação de medições de disponibilidade na ilha da UFPA.

A ilha da UFPA está com o *testbed* sem-fio em processo de instalação, portanto o portal do OMF ficou fora do ar durante o período de medição. Já o domínio OpenFlow da ilha estava inteiramente com conectividade na rede e serviços *web* funcionais, porém as VMs na ilha eram perdidas e o Flowvisor recusava conexões impossibilitando experimentos. Portanto a disponibilidade real do *testbed* no período foi de 0%.

6.1.1.7 UFPE

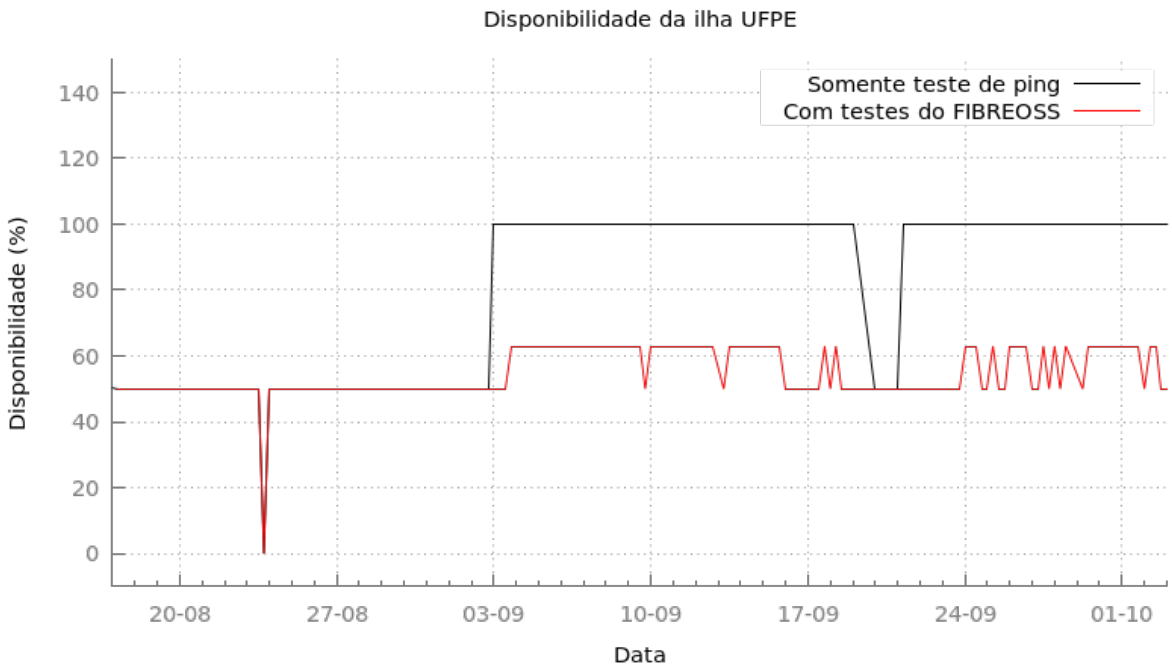


Figura 6.17: Comparação de medições de disponibilidade na ilha da UFPE.

Desde o início da medição até o dia 03/09/2016 uma falha no Flowvisor impedia a experimentação. Após esse período, o domínio OpenFlow funcionou de maneira instável apenas com a VM1.

6.1.1.8 UFRJ

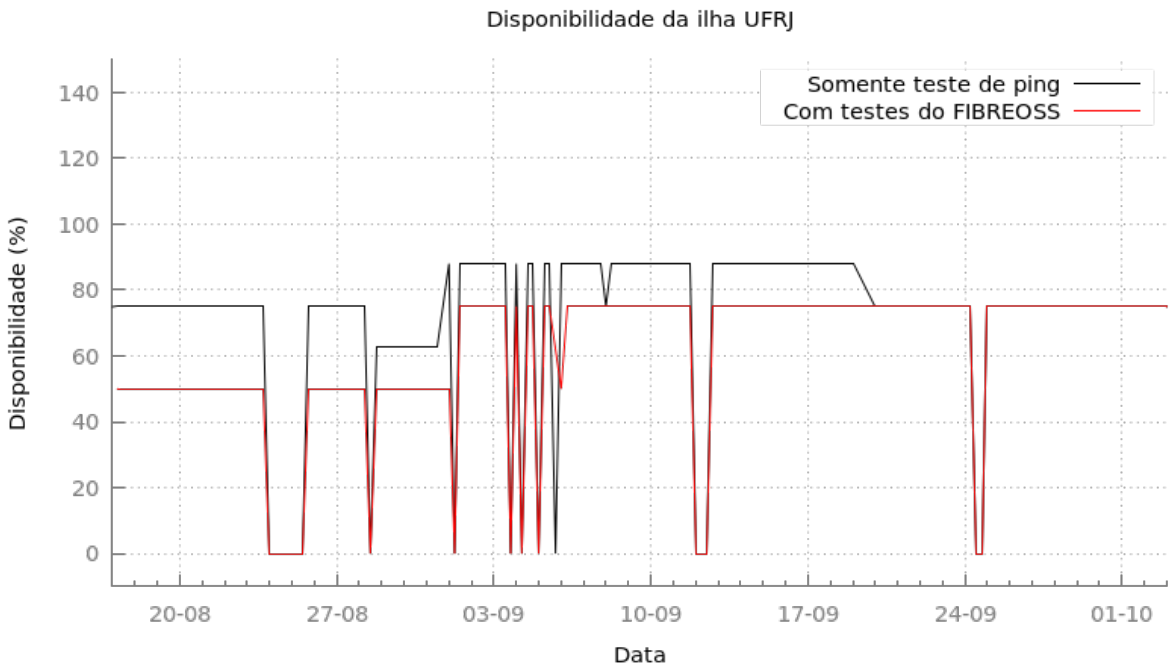


Figura 6.18: Comparação de medições de disponibilidade na ilha da UFRJ.

A UFRJ está conectada à FIBRENET através de uma VPN e por isso teve vários eventos de perda total de conectividade. Durante todo o período de medição somente a NetFPGA1 e o *switch* pronto estavam funcionais, limitando a disponibilidade em 75%. A Figura 6.19 mostra a taxa de transferência da VM1.

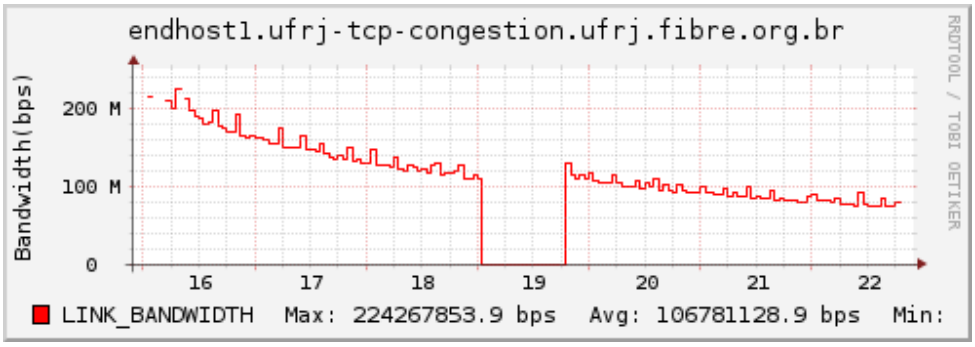


Figura 6.19: Teste de transferência no plano de dados da VM conectada à NetFPGA1 na ilha da UFRJ.

6.1.1.9 UFSCar

Figura 6.20: Comparação de medições de disponibilidade na ilha da UFSCar.

A UFSCar estava realizando mudança de prédio até o dia 15/08/2016 e voltou a funcionar com alguma instabilidade, ficando totalmente fora do ar no período de 02/09/2016 a 05/09/2016. Após essa data, apenas a NetFPGA1 no domínio OpenFlow voltou a funcionar por um dia e depois parou de responder, então todos os recursos OpenFlow ficaram indisponíveis impossibilitando a experimentação.

A UFSCar não possui um domínio sem-fio instalado, então a falha do domínio OpenFlow levou a disponibilidade a 0%.

6.1.1.10 Ilhas inativas

As ilhas do CPqD e UNIFACS ficaram indisponíveis durante todo o período de medição. A ilha UNIFACS estava realizando uma mudança prédio. Já a ilha do CPqD ficou fora do ar durante todo o período de observação por conta de um rompimento de fibra.

6.1.1.11 Resumo

Tabela 6.1: Disponibilidade média das ilhas no período

	Somente ping	Com testes do FIBREOSS
CPqD	0,0%	0,0%
NOC	98,6%	98,6%
RNP	99,4%	51,1%
UFF	63,2%	38,5%
UFG	94,1%	73,7%
UFPA	38,8%	0,0%
UFPE	81,1%	56,5%
UFRJ	68,8%	60,0%
UFSCar	37,0%	0,0%
UNIFACS	0,0%	0,0%
USP	98,6%	77,7%

As Figuras 6.21 a 6.30 mostram a disponibilidade média das máquinas das ilhas medida apenas com teste de ping.

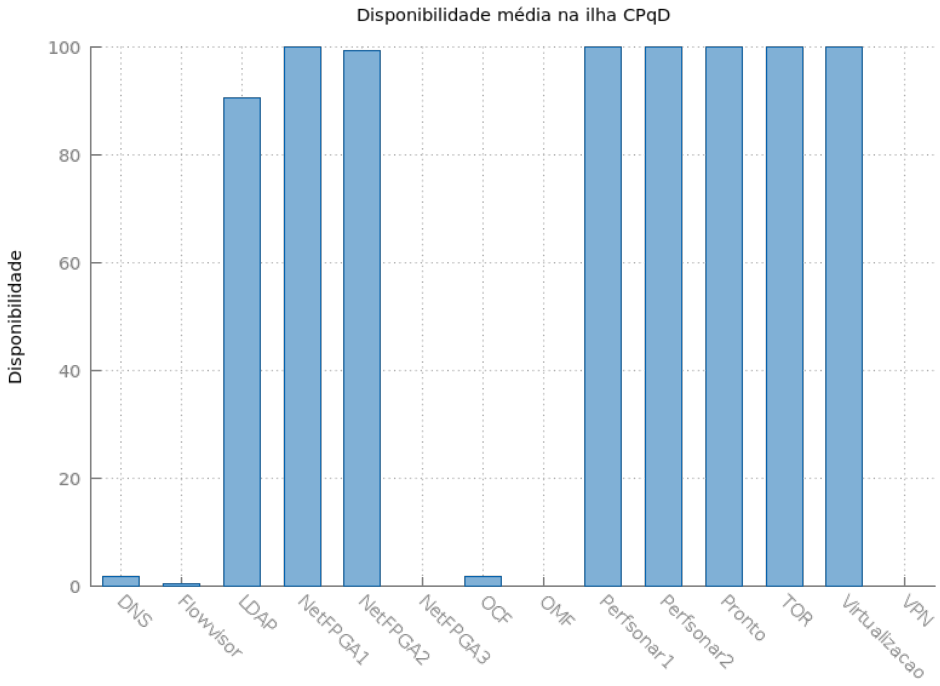


Figura 6.21: Disponibilidade média de dispositivos na ilha do CPqD.

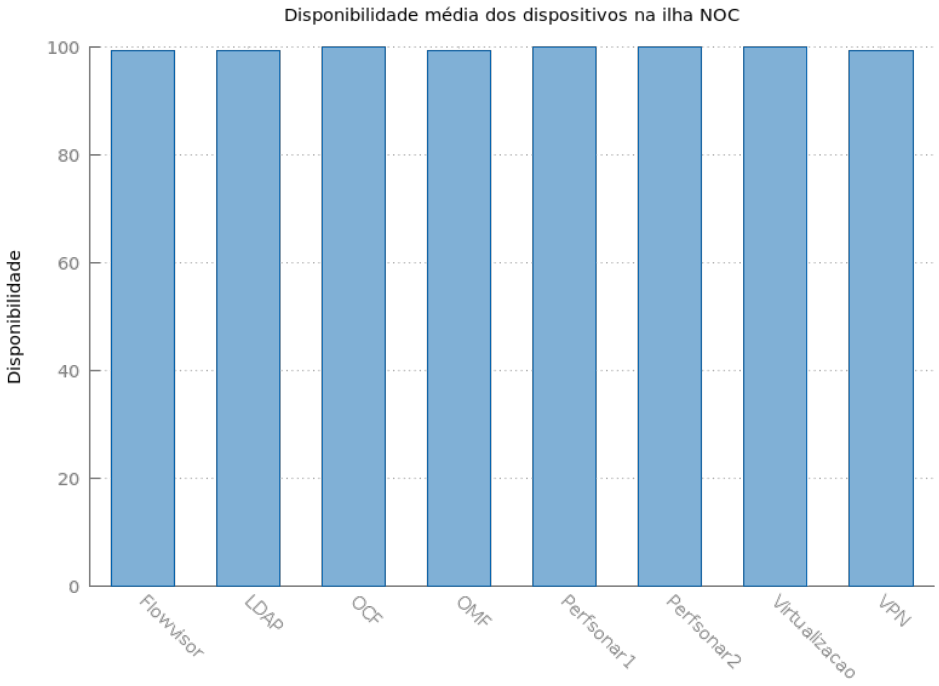


Figura 6.22: Disponibilidade média de dispositivos na ilha do NOC.

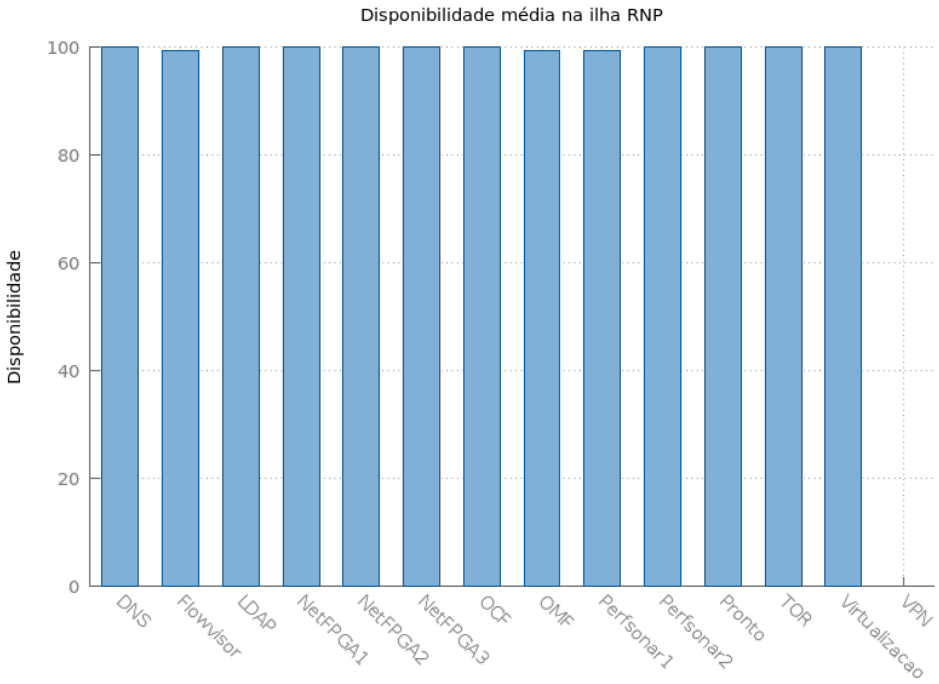


Figura 6.23: Disponibilidade média de dispositivos na ilha do RNP.

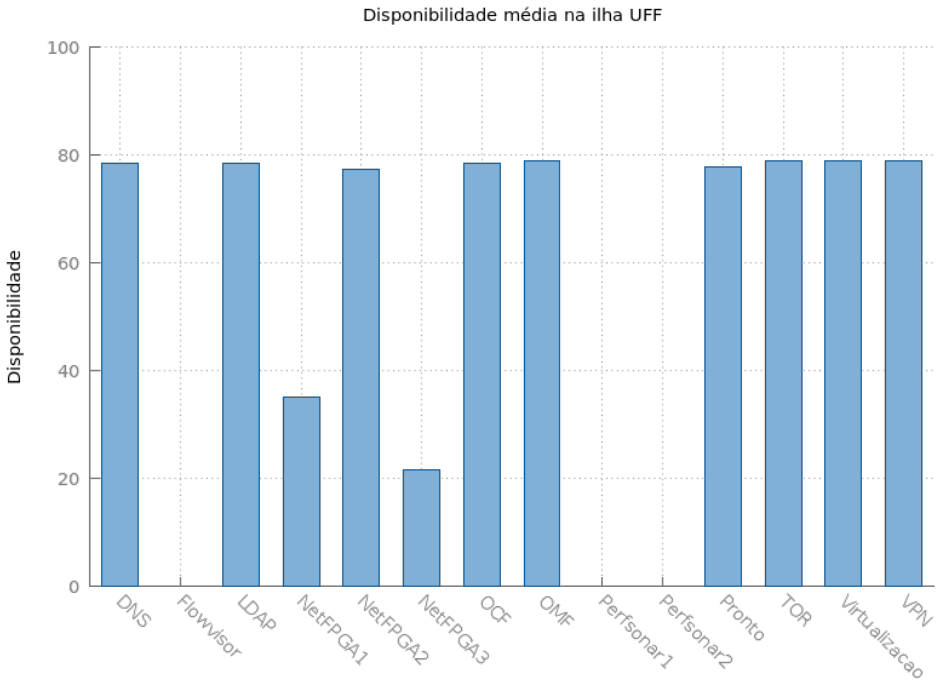


Figura 6.24: Disponibilidade média de dispositivos na ilha do UFF.

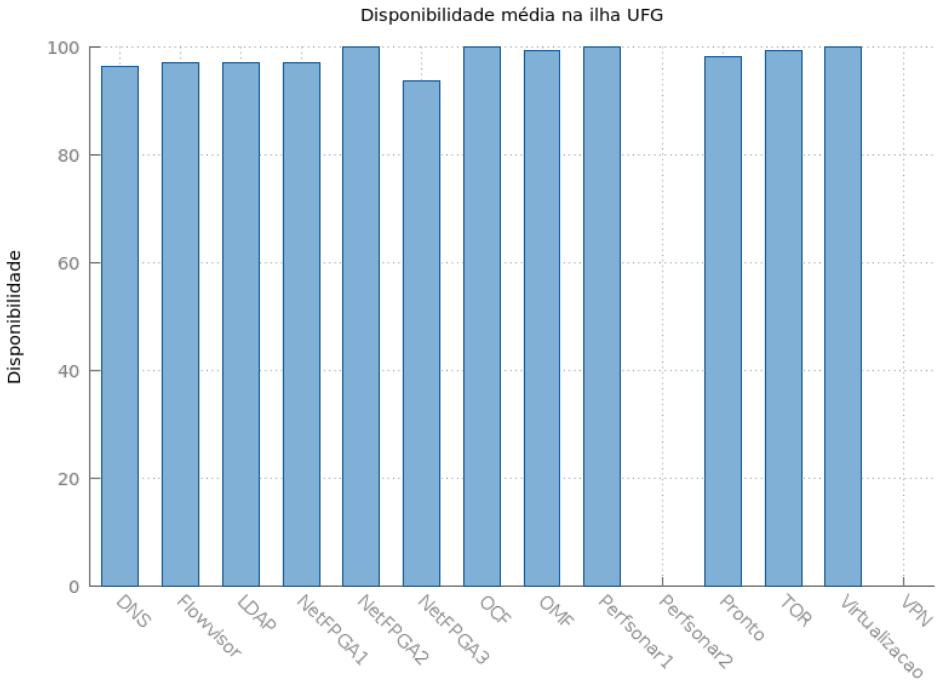


Figura 6.25: Disponibilidade média de dispositivos na ilha do UFG.

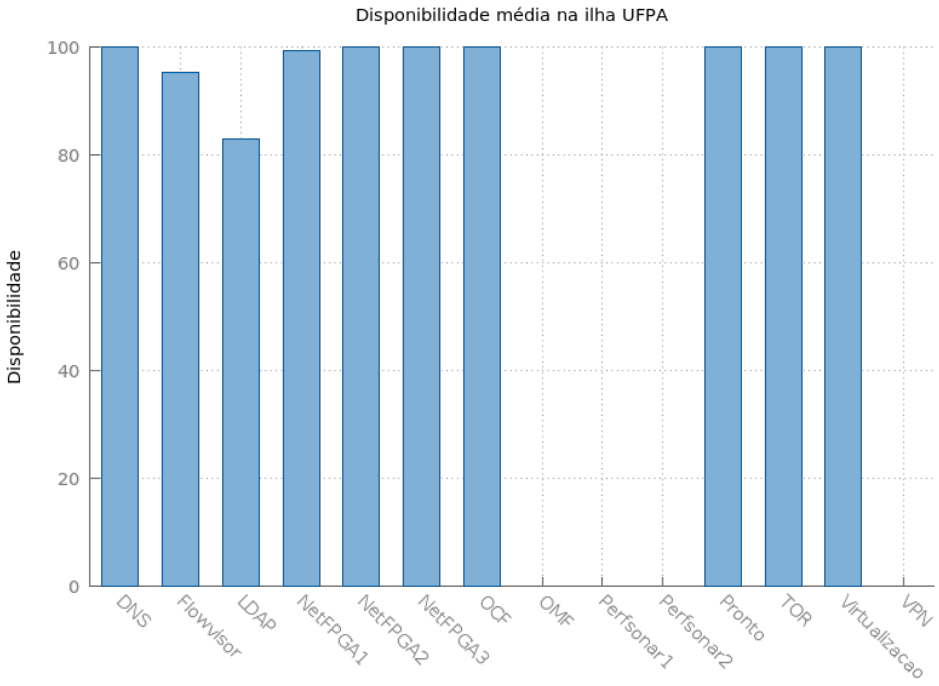


Figura 6.26: Disponibilidade média de dispositivos na ilha do UFPA.

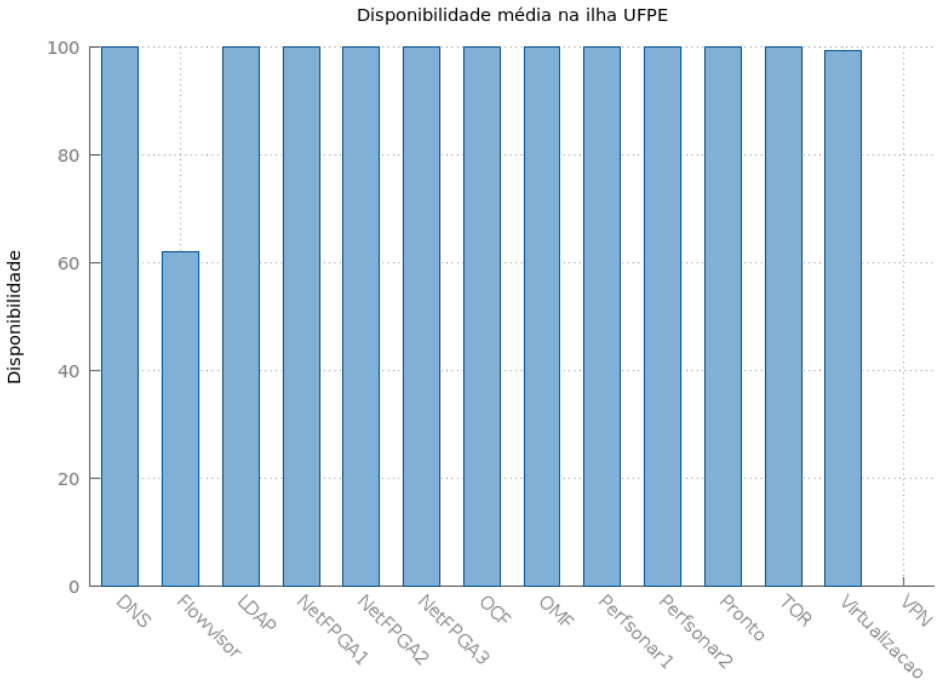


Figura 6.27: Disponibilidade média de dispositivos na ilha do UFPE.

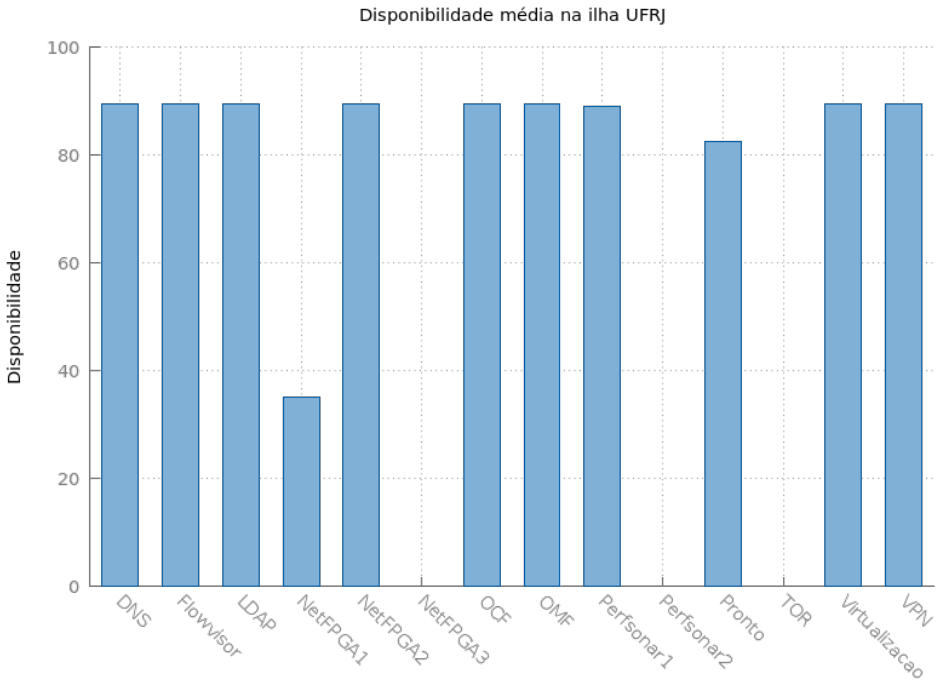


Figura 6.28: Disponibilidade média de dispositivos na ilha do UFRJ.

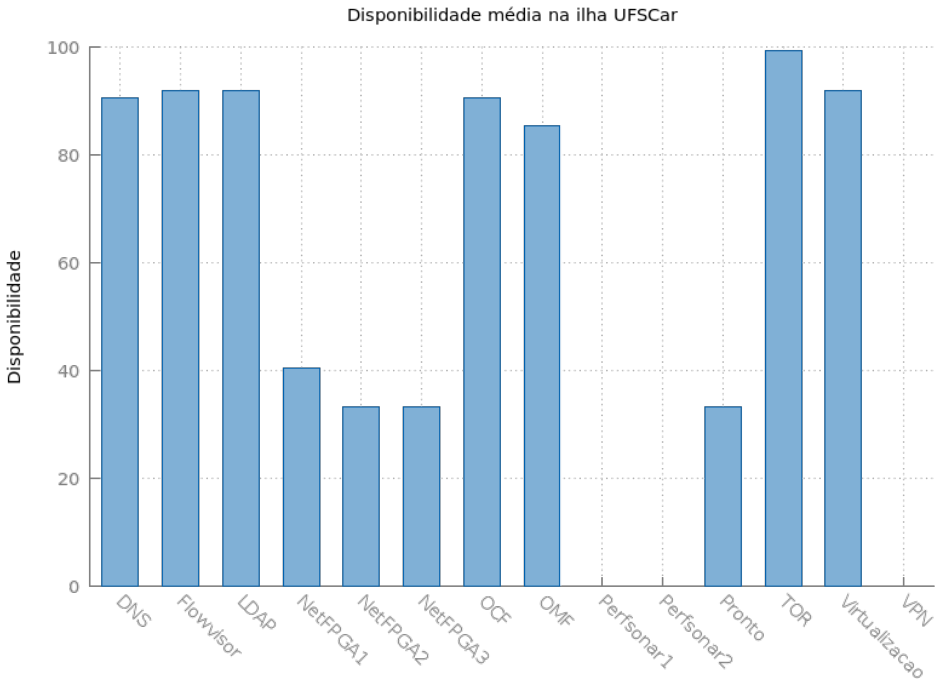


Figura 6.29: Disponibilidade média de dispositivos na ilha do UFSCar.

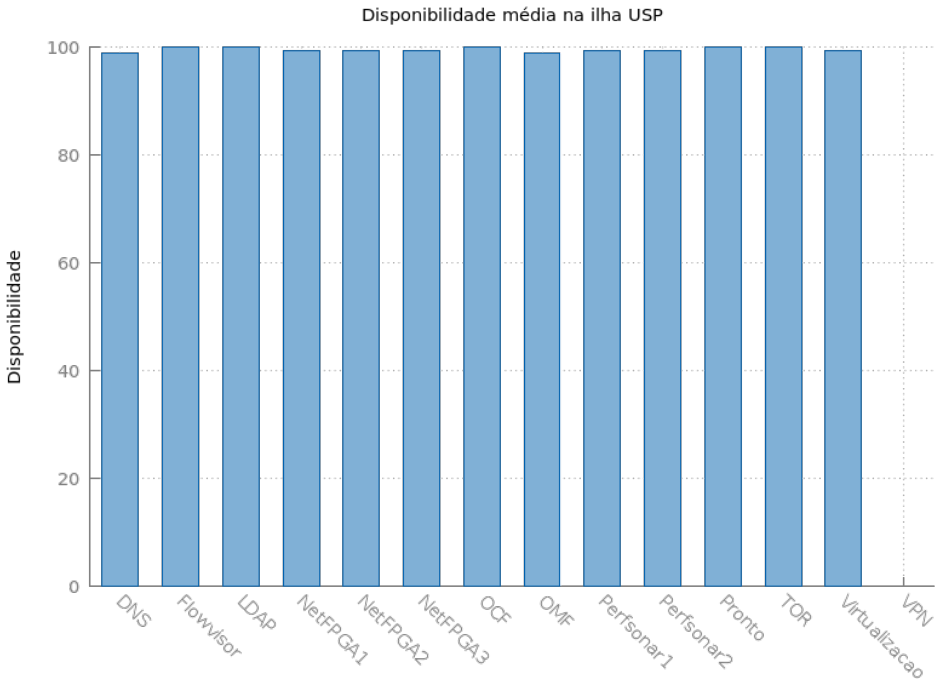


Figura 6.30: Disponibilidade média de dispositivos na ilha do USP.

Um problema observado nos resultados da monitoração do plano de dados foi a redução progressiva da taxa de transferência máxima e aumento do atraso em algumas ilhas. (Ver Figuras 6.19,6.14 e 6.15) Após reiniciar as VMs (Figuras 6.31 e 6.32) a transferência normalizava e voltada a cair gradativamente. Observou-se que o *daemon* do iperf estava forçando o uso da CPU a 100%.

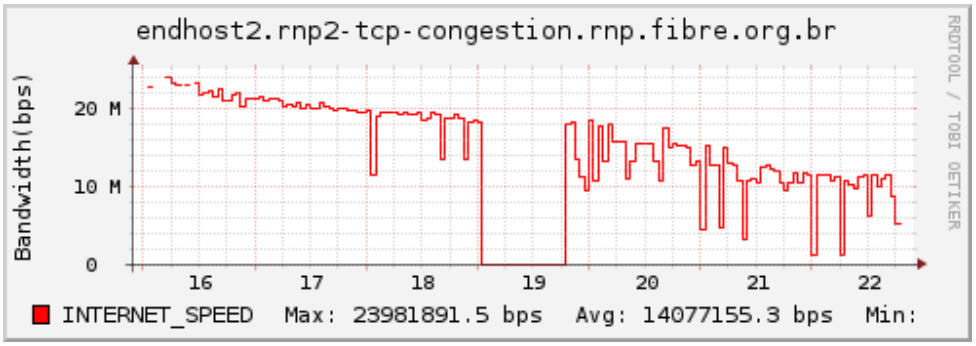


Figura 6.31: Queda gradual de banda na conexão na NetFPGA2 na ilha da RNP.

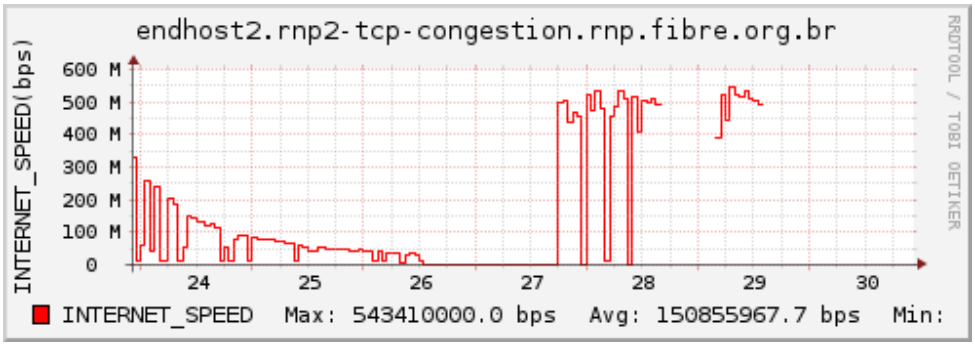


Figura 6.32: Recuperação banda na conexão na NetFPGA2 na ilha da RNP após reiniciar a VM nos dias 27/09/2016, 28/09/2016 e término da medição no dia 29/09/2016

6.1.2 Erros

A Figura 6.33 mostra as estatísticas de erro obtidas a partir do registro de mensagens do OCF desde 2014. Sendo que a maioria dos erros estava relacionado ao controle de virtualização e problemas com o Flowvisor. O controle de virtualização é responsável por criar, iniciar, parar e deletar VMs de experimentação, portanto *slices* já criados e funcionais não são afetados por problemas com o Virtualization Aggregate Manager (VTAM). Porém problemas com o Flowvisor prejudicam a conectividade no plano de dados, já que o Flowvisor funciona como um *proxy* para controladores.

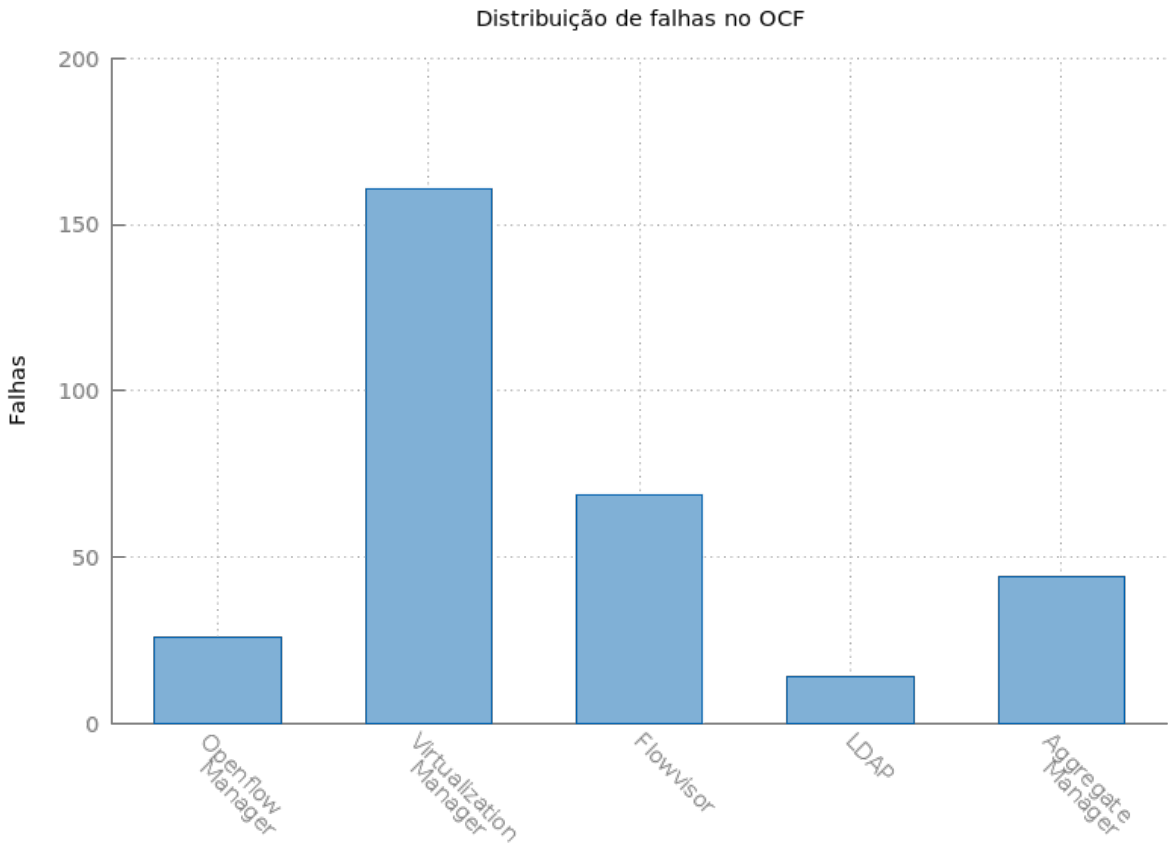


Figura 6.33: Distribuição de erros no OCF.

Dentre as mensagens relacionadas ao VTAM, a Tabela 6.2 as ações executadas pelo VTAM e sua taxa de sucesso.

Tabela 6.2: Estatística de mensagens do OCF - funções do VTAM

	Sucesso	Erro	Taxa de sucesso
VTMANAGER_START	2691	45	98%
VTMANAGER_HARDSTOP	10498	25	99%
VTMANAGER_VM_DELETE	779	26	97%
VTMANAGER_CREATE	2008	65	97%

6.2 Desempenho e escalabilidade

Para testar o serviço de medição, uma máquina virtual com 10GB de disco, duas CPUs, 2GB de memória RAM e sistema operacional Debian 7 foi instalada em uma máquina[4]

[4]Foi utilizado um nó Ícarus

com processador i7, 4GB de memória RAM e sistema operacional Debian 7.

Os testes do plano de controle foram realizados com uma lista de tarefas contendo 166 dispositivos para verificar [5]. Os testes do plano de dados continham 35 VMs para verificar. Os gráficos plotados mostram o valor médio de 100 testes. Os testes foram realizados por sondas customizadas, que são *scripts* em python paralelizados. Nos testes de escalabilidade são plotados os tempo médio de processamento da fila em função do número de *threads*.

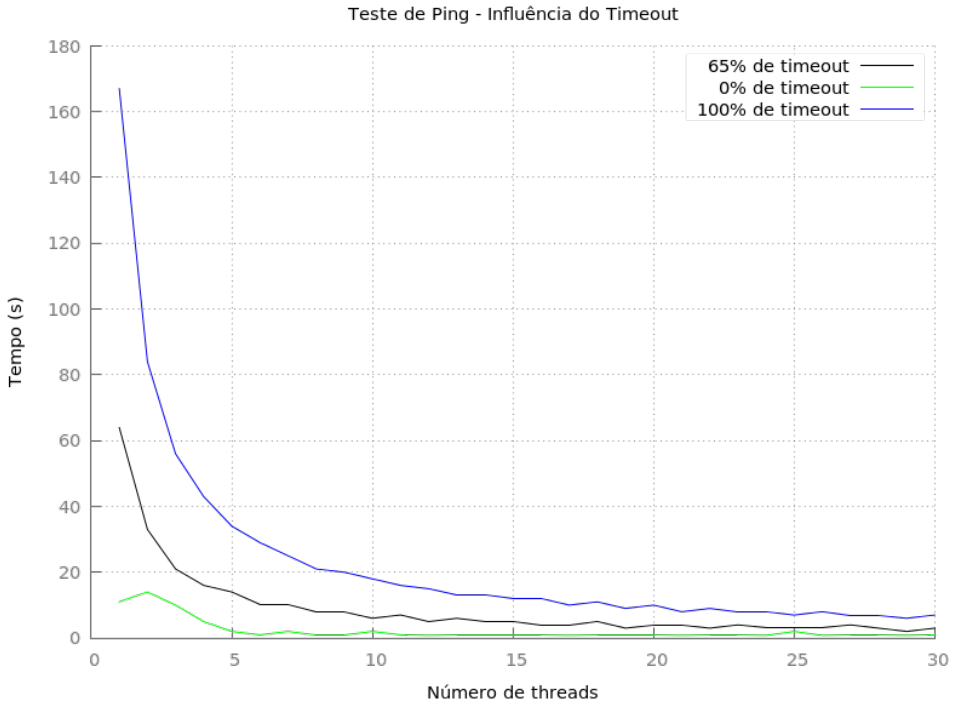


Figura 6.34: Tempo de processamento do teste de ping. Curvas de nível com diferentes proporções de dispositivos conectados

O tempo de resposta ao *ping* varia muito em função do estado do dispositivo testado. O *timeout* configurado para o teste de foi 1 segundo, portanto cada teste leva de cerca 100ms[6] a 1000ms[7]. A Figura 6.34 mostra três rodadas de medições, com todos os dispositivos desconectados (0%), levando 1 segundo por teste, com 65% dos dispositivos conectados[8] e com todos os dispositivos conectados (100%). Todos os testes mostrados a daqui em diante usam a proporção de 65% de *timeout*.

[5] Na época existiam 166 dispositivos nas ilhas ativas da federação
[6] Valor médio mínimo
[7] valor máximo
[8] observou-se que a proporção média de dispositivos conectados em toda a federação varia entre 55% e 65%

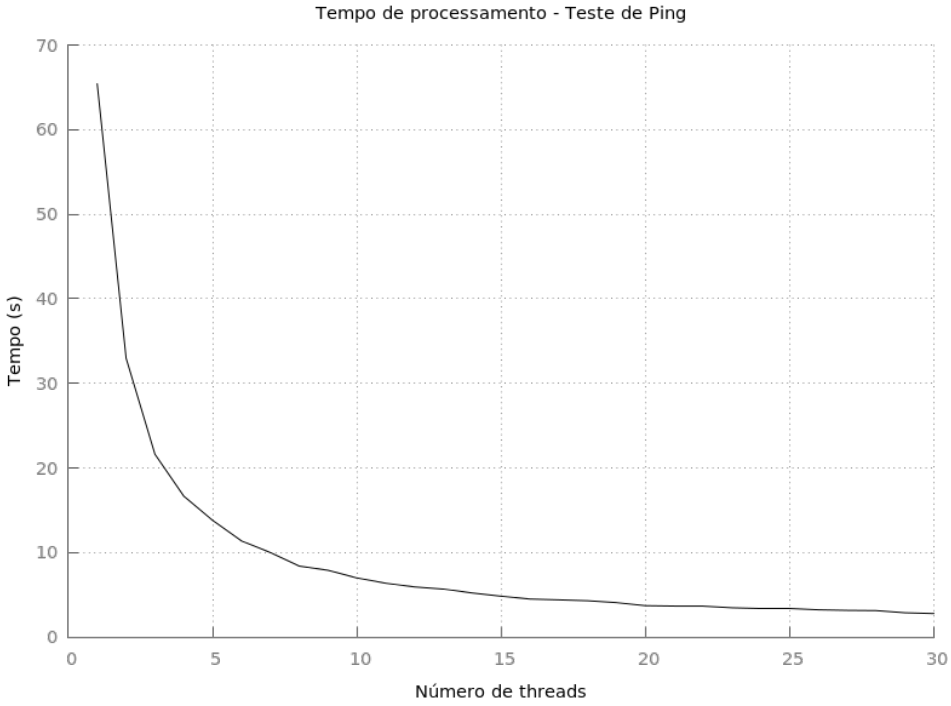


Figura 6.35: Tempo de processamento do teste de *ping* no plano de controle em função do número de *threads*.

A Figura 6.35 mostra a média de 100 testes com o tempo de processamento de uma fila de 166 tarefas em função do número de *threads*.

O erro máximo em todas as medições foi menor do que 2%, portanto a barra erro não foi impressa em nenhum dos gráficos de desempenho.

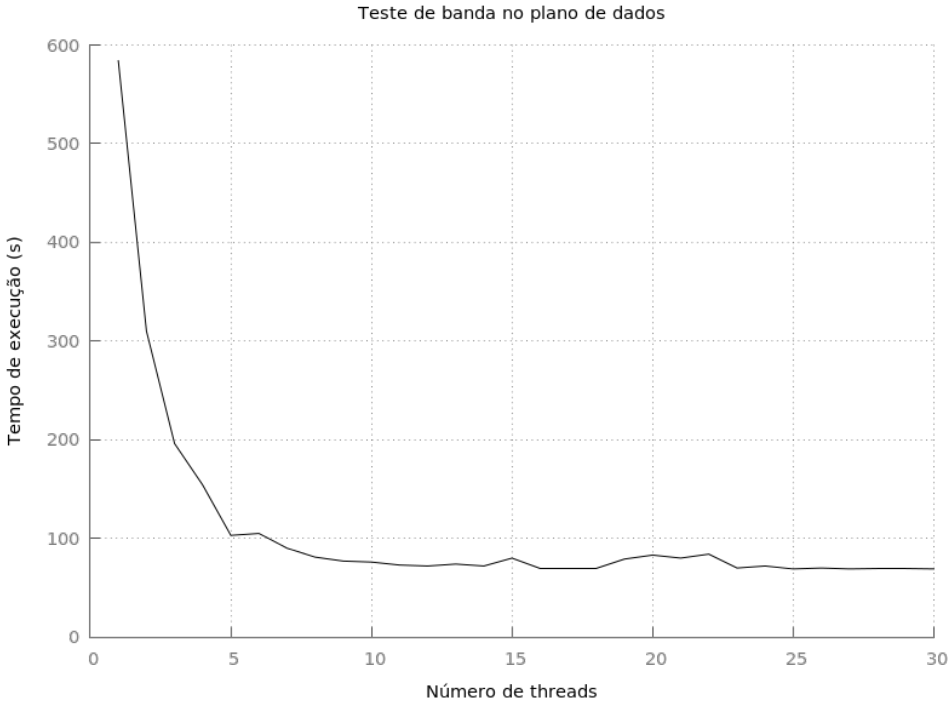


Figura 6.36: Tempo de processamento do teste de atraso no plano de dados em função do número de *threads*.

O teste de largura de banda do canal foi feito usando o iperf com a duração do teste configurada para 5 segundos. O *timeout* configurado é de 10 segundos.

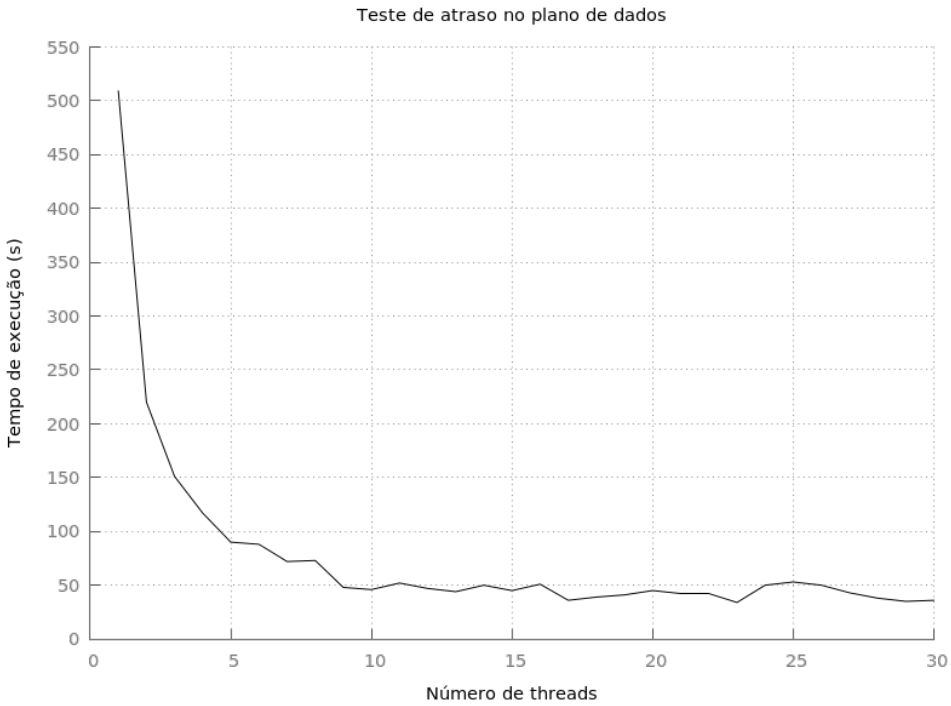


Figura 6.37: Tempo de processamento do teste de banda no plano de dados em função do número de *threads*.

O teste de atraso foi feito usando o *ping*. A média de cinco testes é gravada. O *timeout* configurado é de 10 segundos.

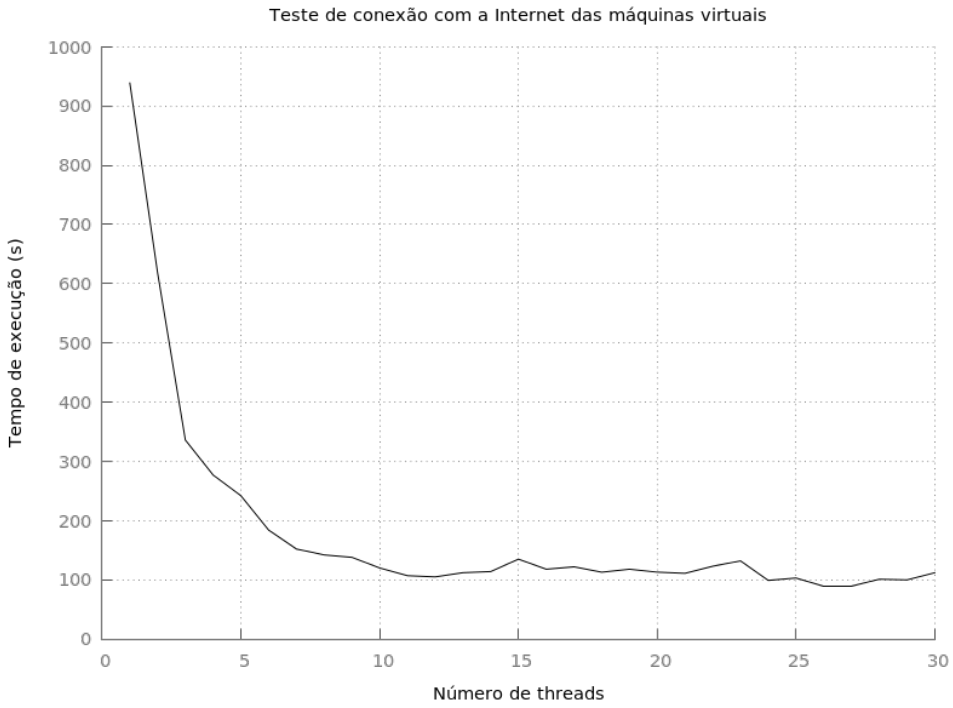


Figura 6.38: Tempo de processamento do teste de conexão com a Internet das máquinas virtuais em função do número de *threads*.

O teste de banda na conexão com a Internet é feito utilizando o *script* do Speedtest [113] para linha de comando. O *timeout* configurado é de 60 segundos.

6.2.1 Impacto

Para realizar uma monitoração fiel à percepção do usuário foi necessário simular o uso dos recursos. No caso do OMF, recursos sem-fio foram reservados todos os dias durante 30 minutos de 2h00 às 2h30 [9]. No caso do domínio OpenFlow, não foi necessária reserva de recursos, pois o sistema de *slice* permite que múltiplos experimentos sejam executados simultaneamente. Para cada ilha foi criado um *slice* e 4 VMs, cada uma com 10GB de espaço de disco, uma CPU e 128MB de memória RAM. Os testes ativos realizados no plano de dados foram de conexão com a Internet, que gera grandes quantidades de tráfego no plano de controle, conexão no plano de dados, que usa toda a banda disponível no plano de dados [10].

[9] Horário de Brasília
[10] Realizado com iperf

Capítulo 7

Conclusão

Um levantamento do estado da arte em monitoração de *testbeds* de redes de computadores e uma descrição detalhada do *testbed* FIBRE foi realizada. A partir dessas informações foi possível ter uma ideia de como pesquisadores ao redor do mundo estão mantendo seus laboratórios de redes de computadores operacionais. Também, uma descrição do estado atual do FIBRE e de suas ferramentas ajuda a compreender melhor o funcionamento do mesmo.

Os outros objetivos deste trabalho, que eram o desenvolvimento do relatório disponibilidade, interface gráfica, núcleo do arcabouço e sondas de teste permitiram uma análise mais aprofundada do funcionamento do *testbed* e detecção de problemas que não eram conhecidos ou observáveis antes da implantação do FIBREOSS.

O desenvolvimento do relatório de disponibilidade por exemplo, permite uma visão macro do estado da federação, de forma que o usuário pode conhecer melhor o nível do serviço oferecido. Enquanto que o sistema de agregação de alarmes possibilita uma visão mais objetiva dos problemas e da determinação da causa raiz.

Outro objetivo envolve o teste das ferramentas de controle de experimentos do FIBRE. Os testes apresentados na Seção 5.6 que são realizados no portal do OCF permitem que a descoberta de problemas seja realizada pró-ativamente, sem a necessidade de que exista uma reclamação de usuário ou tíquete de falha aberto. Além disso o teste do plano de dados no domínio OpenFlow permite que operadores e experimentadores tenham um experimento de referência para atestar o estado de saúde das ilhas. Problemas relacionados ao gerenciamento de máquinas virtuais foram detectados no domínio OpenFlow. Uma das sondas customizadas que testava as funcionalidades do OCF permitiu que a equipe de desenvolvimento do FIBRE descobrisse a existência de um *bug* no OCF que

não removia completamente as VMs criadas esgotando o espaço em disco do servidor de virtualização. A rotina de criação e deleção de experimentos mostrou rapidamente que havia um problema no OCF.

A análise da disponibilidade mostra que o *testbed* precisa amadurecer pois seus serviços ainda são muito instáveis. Algumas ações podem ser tomadas para melhorar a disponibilidade do *testbed*. Eliminar a necessidade de conexões através de VPN é uma delas e está sendo providenciada para a ilha da UFF. Uma conexão direta com a FIBRE-net através de fibra óptica está sendo instalada. O mesmo pode ser feito com a ilha da UFRJ. Outra medida que está sendo estudada é a substituição do OCF e OMF 5.4 pelo OMF 6 como o único arcabouço de controle de experimentos OpenFlow [114]. Além disso o sistema de monitoração de infraestrutura ZenOSS está sendo substituído pelo Zabbix.

Os arquivos relacionados ao projetos estão no github: https://github.com/vitorsfarias/fibreoss

7.1 Trabalhos Futuros

Existem alguns requisitos que puderam ser desenvolvidos durante a duração deste trabalho e que podem acrescentar mais precisão à monitoração. A verificação periódica do plano de dados do domínio sem-fio pode ser realizada através de experimentos periódicos. Porém esta é mais complexa já que o domínio sem-fio não permite experimentos simultâneos como o domínio OpenFlow. A recuperação automática de falhas também pode ser estudada, realizando cópias periódicas das VMs é possível restaurar alguns serviços ao último estado sem falhas do sistema. Também, um mecanismo de correlação de alarmes pode separar falhas no sistema de medição, evitando alarmes falsos no caso de perda de conexão do FIBREOSS.

De forma a diminuir o impacto causado pelos *healthchecks*, podem ser usados dados de experimentação de usuários para realizar a validação dos serviços. Dessa forma, a frequência dos testes pode ser reduzida se for constatado que os serviços estão sendo utilizados pelos usuários e atendendo-os corretamente. Outra maneira de diminuir o impacto é remover o *daemon* do iperf, que consome muitos recursos para garantir a precisão do relógio. Procurando uma alternativa mais leve.Além disso, para evitar o travamento das VMs um *watchdog* pode reinicia-las quando houver degradação grave de performance.

Um amadurecimento do FIBREOSS também será necessário para que ele seja mais fácil de configurar e mais rápido para processar dados. Outra opção é modificar outras

plataformas de teste existentes, como as apresentadas no Capítulo 4 para adequar-se ao FIBRE. A implantação do sistema será realizada instalando as VMs em cada umas das ilhas. Atualizações serão instaladas pelos próprios operadores. Após a implantação inicial, será necessário adequar a plataforma às atualizações feitas na ilha. Sistemas de gerência e CMFs que estão sendo atualizados precisarão de novos *scripts* de teste ou de adequações dos mesmos.

Referências

[1] "Fibre webpage," https://fibre.org.br, accessed: 2014-11-23.

[2] A. Abelem, M. Stanton, I. Machado, M. Salvador, L. Magalhaes, N. Fernandes, S. Correa, K. Cardoso, C. Marcondes, J. Martins *et al.*, "Fit@ br-a future internet testbed in brazil," *Proceedings of the Asia-Pacific Advanced Network*, vol. 36, pp. 1–8, 2013.

[3] A. Abelem and S. Fdida, "Building an Infrastructure for Experimentation Between Brazil and Europe To Enhance Research Collaboration in Future Internet," *Tnc2014 Conference*, no. April, pp. 1–17, 2014. [Online]. Available: ttps://tnc2014.terena.org/getfile/1071

[4] "Fibre modelo de governancia," http://fibre.org.br/about/governance-model/, accessed: 2015-10-15.

[5] J. Arata, H. Takahashi, P. Pitakwatchara, S. Warisawa, K. Tanoue, K. Konishi, S. Ieiri, S. Shimizu, N. Nakashima, K. Okamura, Y. Fujino, Y. Ueda, P. Chotiwan, M. Mitsuishi, and M. Hashizume, "A remote surgery experiment between japan and thailand over internet using a low latency codec system," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, April 2007, pp. 953–959.

[6] S. M. Amin and B. F. Wollenberg, "Toward a smart grid: power delivery for the 21st century," *IEEE power and energy magazine*, vol. 3, no. 5, pp. 34–41, 2005.

[7] J. M. Hernández-Muñoz, J. B. Vercher, L. Muñoz, J. A. Galache, M. Presser, L. A. H. Gómez, and J. Pettersson, "Smart cities at the forefront of the future internet," in *The Future Internet Assembly*. Springer, 2011, pp. 447–462.

[8] J. S. Turner and D. E. Taylor, "Diversifying the internet," in *GLOBECOM'05. IEEE Global Telecommunications Conference, 2005.*, vol. 2. IEEE, 2005, pp. 6–pp.

[9] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the internet impasse through virtualization," *Computer*, vol. 38, no. 4, pp. 34–41, 2005.

[10] "Ieee standard glossary of software engineering terminology," *IEEE Std 610.12-1990*, pp. 1–84, Dec 1990.

[11] S. Wahle, A. Gavras, F. Gouveia, H. Hrasnica, and T. Magedanz, "Network domain federation–infrastructure for federated testbeds," *NEM Summit*, 2008.

[12] D. Marschke, *Software Defined Networking (SDN): Anatomy of OpenFlow Volume I (Volume 1)*. Lulu Publishing Services, 3 2015. [Online]. Available: http://amazon.com/o/ASIN/1483427234/

[13] L. Casta, P. Jain, and a. M. Hausi, "The Future of Internet Applications : A Survey of Future Internet Projects."

[14] "Geni webpage," https://www.geni.net/, accessed: 2016-09-30.

[15] M. Berman, J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, and I. Seskar, "GENI : A federated testbed for innovative network experiments q," *Computer Networks*, vol. 61, no. 2014, pp. 5–23, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.bjp.2013.12.037

[16] "Gmoc webpage," http://gmoc.grnoc.iu.edu/gmoc/index/about-gmoc.html, accessed: 2016-09-22.

[17] J. Duerig, R. Ricci, C. Carpenter, Z. Fei, L. Stoller, M. Strum, J. Griffioen, H. Nasir, J. Reed, and X. Wu, "Getting Started with GENI : A User Tutorial," vol. 42, no. 1, pp. 72–77, 2012.

[18] "Gemini webpage," http://groups.geni.net/geni/wiki/GEMINI, accessed: 2016-09-30.

[19] "Instools design docs," http://groups.geni.net/geni/attachment/wiki/InstrumentationTools/instools-design-doc.pdf, accessed: 2016-09-22.

[20] J. Griffioen, Z. Fei, and H. Nasir, "Architectural Design and Specification of the INSTOOLS Measurement System," no. December, pp. 1–18, 2009.

[21] "Lamp webpage," http://groups.geni.net/geni/wiki/LAMP, accessed: 2016-09-22.

[22] M. Portnoi and M. Swany, "Unified Network Information Services (UNIS) allow users to discover network services and capabilities."

[23] "Gimi webpage," http://groups.geni.net/geni/wiki/GIMI, accessed: 2016-09-30.

[24] "Orca webpage," https://geni-orca.renci.org/trac/, accessed: 2016-09-30.

[25] "Exogeni webpage," http://www.exogeni.net/, accessed: 2016-09-30.

[26] "Oml webpage," https://mytestbed.net/projects/oml, accessed: 2016-09-30.

[27] M. Zink, M. Ott, and I. Baldine, "GIMI : Large-scale GENI Instrumentation and Measurement Infrastructure," no. option 1, p. 1.

[28] "Planetlab webpage," https://www.planet-lab.org/, accessed: 2016-11-28.

[29] L. Peterson and T. Roscoe, "The design principles of planetlab," *SIGOPS Oper. Syst. Rev.*, vol. 40, no. 1, pp. 11–16, Jan. 2006. [Online]. Available: http://doi.acm.org.ez24.periodicos.capes.gov.br/10.1145/1113361.1113367

[30] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "Planetlab: An overlay testbed for broad-coverage services," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 3, pp. 3–12, Jul. 2003. [Online]. Available: http://doi.acm.org.ez24.periodicos.capes.gov.br/10.1145/956993.956995

[31] "Orbit webpage," http://www.orbit-lab.org/, accessed: 2016-09-30.

[32] R. U. WINLAB, "Orbit testbed software architecture: Supporting experiments as a service," 2005.

[33] S. Keranidis, D. Giatsios, T. Korakis, I. Koutsopoulos, L. Tassiulas, T. Rakotoarivelo, M. Ott, and T. Parmentelat, "Experimentation on end-to-end performance aware algorithms in the federated environment of the heterogeneous planetlab and nitos testbeds," *Computer Networks*, vol. 63, pp. 48–67, 2014.

[34] "Fibre e omf," http://www.winlab.rutgers.edu/pub/docs/focus/GENI-OMF.html, accessed: 2016-09-30.

[35] "Geni e omf," http://www.fibre-ict.eu/index.php/cmf/omf, accessed: 2016-09-30.

[36] "Fire webpage," https://www.ict-fire.eu/fire/, accessed: 2016-09-30.

[37] A. Gavras, A. Karila, S. Fdida, M. May, and M. Potts, "Future internet research and experimentation: The fire initiative," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 3, pp. 89–92, Jul. 2007. [Online]. Available: http://doi.acm.org/10.1145/1273445.1273460

[38] "Fire testbeds description," http://www.ict-openlab.eu/technologies/testbeds.html, accessed: 2016-09-30.

[39] "Fire projects webpage," https://www.ict-fire.eu/projects/, accessed: 2016-09-30.

[40] "Confine webpage," https://confine-project.eu, accessed: 2016-09-30.

[41] "Crew webpage," http://www.crew-project.eu, accessed: 2016-09-30.

[42] "Experimedia webpage," http://www.experimedia.eu, accessed: 2016-09-30.

[43] "Openlab webpage," http://www.ict-openlab.eu/, accessed: 2016-09-30.

[44] "Bonfire webpage," http://www.bonfire-project.eu/about, accessed: 2016-09-30.

[45] "Ofelia webpage," http://www.fp7-ofelia.eu/, accessed: 2016-09-30.

[46] "Smartsantander webpage," http://smartsantander.eu, accessed: 2016-09-30.

[47] "Onelab webpage," http://onelab.eu, accessed: 2016-09-30.

[48] "Panlab webpage," http://www.panlab.net/, accessed: 2015-10-30.

[49] "Federica webpage," http://www.fp7-federica.eu, accessed: 2016-09-30.

[50] "Tefis webpage," http://www.tefisportal.eu/, accessed: 2015-10-23.

[51] Y. Al-Hazmi and T. Magedanz, "A flexible monitoring system for federated future internet testbeds," in *Network of the Future (NOF), 2012 Third International Conference on the*, Nov 2012, pp. 1–6.

[52] Y. Al-Hazmi, K. Campowsky, and T. Magedanz, "A monitoring system for federated clouds," in *Cloud Networking (CLOUDNET), 2012 IEEE 1st International Conference on*, Nov 2012, pp. 68–74.

[53] J. Jofre, C. Velayos, G. Landi, M. Giertych, A. C. Hume, G. Francis, and A. Vico, "Federation of the BonFIRE multi-cloud infrastructure with networking facilities," *Computer Networks*, vol. 61, pp. 184–196, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.bjp.2013.11.012

[54] A. C. Hume, Y. Al-Hazmi, B. Belter, K. Campowsky, L. M. Carril, G. Carrozzo, V. Engen, D. García-Pérez, J. Jofre Ponsatí, R. Kűbert, Y. Liang, C. Rohr, and G. Van Seghbroeck, *BonFIRE: A Multi-cloud Test Facility for Internet of Services Experimentation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 81–96. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-35576-9_11

[55] "Zabbix webpage," http://www.zabbix.com/, accessed: 2016-09-30.

[56] M. Suñé, L. Bergesio, H. Woesner, T. Rothe, A. Köpsel, D. Colle, B. Puype, D. Simeonidou, R. Nejabati, M. Channegowda, M. Kind, T. Dietz, A. Autenrieth, V. Kotronis, E. Salvadori, S. Salsano, M. Körner, and S. Sharma, "Design and implementation of the OFELIA FP7 facility : The European OpenFlow testbed," vol. 61, pp. 132–150, 2014.

[57] "Ofelia webpage," http://www.fp7-ofelia.eu/about-ofelia/, accessed: 2016-09-30.

[58] "Rob sherwood and kok-kiong yap. cbench: an open- flow controller benchmarker." http://www.openflow.org/wk/index.php/Oflops., accessed: 2016-08-27.

[59] A. Tootoonchian, S. Gorbunov, Y. Ganjali, M. Casado, and R. Sherwood, "On controller performance in software-defined networks," *Proceeding Hot-ICE'12 Proceedings of the 2nd USENIX conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services*, pp. 10–10, 2012. [Online]. Available: https://www.usenix.org/system/files/conference/hot-ice12/hotice12-final33{_}0.pdf

[60] "Oflops webpage," http://www.openflow.org/wk/index.php/Oflops, accessed: 2016-09-30.

[61] C. Rotsos, N. Sarrar, S. Uhlig, R. Sherwood, and A. W. Moore, "OFLOPS: An Open Framework for OpenFlow Switch Evaluation."

[62] M. Campanella and F. Farina, "The FEDERICA infrastructure and experience," *Computer Networks*, vol. 61, pp. 176–183, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.bjp.2013.12.029

[63] "Federica monitoring webpage," http://www.fp7-federica.eu/infrastructure/monitoring.php, accessed: 2016-09-30.

[64] "Cesnet webpage," https://www.ces.net/, accessed: 2016-09-30.

[65] J. Navrátil, T. Košnar, J. Furman, T. Mrázek, and V. Krmíček, "Monitoring of overlay networks with virtual resources," in *Proceedings. of TERENA Networking Conference*. Citeseer, 2009.

[66] "Perfsonar webpage," http://www.perfsonar.net/deploy/deployment-use-cases/, accessed: 2015-09-30.

[67] "Internet2 webpage," http://www.internet2.edu/, accessed: 2016-09-30.

[68] "Esnet webpage," https://www.es.net/, accessed: 2016-09-30.

[69] "Indiana university perfsonar webpage," http://incntre.iu.edu/research/perfSONAR, accessed: 2016-09-30.

[70] "Geant webpage," http://www.geant.org/, accessed: 2016-09-30.

[71] "Openlab: Extending fire testbeds and tools - deliverable d2.3," http://www.ict-openlab.eu/fileadmin/documents/public_deliverables/OpenLab_Deliverable_D2_3.pdf, accessed: 2016-09-30.

[72] I. Csabai, A. Fekete, P. Hága, B. Hullár, G. Kurucz, S. Laki, P. Mátray, J. Stéger, G. Vattay, F. Espina *et al.*, "Etomic advanced network monitoring system for future internet experimentation," in *International Conference on Testbeds and Research Infrastructures*. Springer, 2010, pp. 243–254.

[73] "Etomic webpage," http://www.etomic.org/index.php, accessed: 2016-09-30.

[74] Y. Shavitt and E. Shir, "Dimes: Let the internet measure itself," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 5, pp. 71–74, 2005.

[75] "Dimes webpage," http://www.netdimes.org/new/, accessed: 2016-09-30.

[76] T. Bourgeau, J. Augé, and T. Friedman, "Tophat: supporting experiments through measurement infrastructure federation," in *International Conference on Testbeds and Research Infrastructures*. Springer, 2010, pp. 542–557.

[77] J. Augé, T. Friedman, and T. Bourgeau, "Overview of tophat: Interconnecting the onelab measurement infrastructures."

[78] "Planetlab status webpage," https://www.planet-lab.org/status, accessed: 2016-09-30.

[79] "Comon webpage," http://comon.cs.princeton.edu/, accessed: 2016-09-30.

[80] "Nitos webpage," http://nitlab.inf.uth.gr/NITlab/index.php/nitos.html, accessed: 2016-09-30.

[81] "Icarus nodes - nitlab webpage," http://nitlab.inf.uth.gr/NITlab/index.php/hardware/wireless-nodes/icarus-nodes, accessed: 2014-11-23.

[82] "G-lab webpage," http://www.german-lab.de/, accessed: 2016-09-30.

[83] D. Schwerdel, B. Reuther, T. Zinner, P. Müller, and P. Tran-gia, "Future Internet research and experimentation : The G-Lab approach," *Computer Networks*, vol. 61, pp. 102–117, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.bjp.2013.12.023

[84] D. Schwerdel, D. Günther, R. Henjes, B. Reuther, and P. Müller, *German-Lab Experimental Facility*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 1–10. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15877-3_1

[85] "Fibre old webpage," http://www.fibre-ict.eu/testbeds/fibre-br, accessed: 2014-11-23.

[86] D. M. T. S. M. S. A. A. J. R. M. S. S. S. L. B. S. F. M. C. L. T. I. Machado, L. Ciuffo and D. Giatsios, "Building an infrastructure for experimentation between brazil and europe to enhance research collaboration in future internet," in *in proc. of TERENA Networking Conference, Dublin, Ireland, April 2014.*

[87] "Myslice webpage," https://www.myslice.info/, accessed: 2015-09-30.

[88] "Flowvisor - stanford openflow webpage," https://openflow.stanford.edu/display/DOCS/Flowvisor, accessed: 2015-09-30.

[89] "Fibre webpage ocf article," http://www.fibre-ict.eu/index.php/cmf/ofelia, accessed: 2014-07-11.

[90] "Omf webpage," https://omf.mytestbed.net/projects/omf/wiki/OMF_Main_Page, accessed: 2014-11-23.

[91] "Portal do fibre noc," https://portal.fibre.org.br/LS-WEB-NOC/index.php?, accessed: 2015-09-30.

[92] "Omf installation," http://mytestbed.net/doc/omf/file.INSTALLATION.html, accessed: 2015-09-30.

[93] "Omf introduction," http://omf.mytestbed.net/projects/omf/wiki/An_Introduction_to_OMF, accessed: 2015-09-30.

[94] "Ocf development page," https://github.com/fp7-ofelia/ocf/wiki/Development, accessed: 2015-09-30.

[95] "Zenoss webpage," http://www.zenoss.com/, accessed: 2015-09-30.

[96] J.-W. Hu, H.-M. Chen, T.-L. Liu, H.-M. Tseng, D. Lin, C.-S. Yang, and C. Yeh, "Implementation of alarm correlation system for hybrid networks based upon the perfsonar framework," in *Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on*, April 2010, pp. 893–898.

[97] "Zenoss api documentation," http://community.zenoss.org/community/documentation/official_documentation/api, accessed: 2015-09-30.

[98] "Fibre monitoring status," http://200.130.15.182/infrastructure/monitoring-status/, accessed: 2015-09-30.

[99] "Zenoss webpage," http://soa.sys-con.com/node/209205/mobile, accessed: 2015-09-30.

[100] "Fibre maddash webpage," http://ps.fibre.org.br/maddash-webui/, accessed: 2015-09-30.

[101] "Maddash webpage," http://software.es.net/maddash/, accessed: 2015-09-30.

[102] L. Lymberopoulos, M. Grammatikou, M. Potts, P. Grosso, A. Fekete, B. Belter, M. Campanella, and V. Maglaris, "NOVI tools and algorithms for federating virtualized infrastructures," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012.

[103] M. Grammatikou, P. Grosso, J. Stéger, B. Pietrzak, M. Campanella, C. Papagianni, G. Androulidakis, M. Potts, and T. Price, "Deliverable 1.6 Project Final Report," Tech. Rep., 2013.

[104] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira, "Avoiding traceroute anomalies with paris traceroute," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. ACM, 2006, pp. 153–158.

[105] B. Donnet, P. Raoult, T. Friedman, and M. Crovella, "Efficient algorithms for large-scale topology discovery," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1. ACM, 2005, pp. 327–338.

[106] "Open multinet webpage," https://open-multinet.info/, accessed: 2016-09-22.

[107] Y. Al-Hazmi, "Unification of monitoring interfaces of federated cloud and future internet testbed infrastructures," 2016.

[108] M. Portnoi and M. Swany, "Unified network information services (unis) allow users to discover network services and capabilities," *SIGCOMM Computer Communication Review*, vol. 27, pp. 3–14.

[109] M. M. Pinheiro, I. L. E. Macêdo, I. L. O. Souza, T. S. Hohlenweger, P. R. R. Leite, A. L. Spínola, H. Monteiro, R. A. Dourado, L. N. Sampaio, J. A. Suruagy Monteiro, and J. S. B. Martins, "An Instrumentation and Measurement Architecture Supporting Multiple Control Monitoring Frameworks."

[110] "Selenium official webpage," http://www.seleniumhq.org/.

[111] "Tutoriais do fibre," http://fibre.org.br/documentation/courseware/, accessed: 2016-10-11.

[112] "Midiacom," http://www.midiacom.uff.br/midiacom/index.php/pt-BR/, accessed: 2016-10-09.

[113] "Speedtest," https://github.com/sivel/speedtest-cli, accessed: 2016-10-11.

[114] L. CIUFFO, T. SALMITO, J. REZENDE, and I. MACHADO, "Testbed fibre: Passado, presente e perspectivas," in *Anais do WPEIF 2016 Workshop de Pesquisa Experimental da Internet do Futuro*, p. 3.

[115] "Datatables webpage," http://datatables.net/, accessed: 2015-09-30.

[116] "Noumenal designs," http://www.noumenaldesigns.com, accessed: 2015-09-30.

APÊNDICE A – Detalhes de Implantação

A.1 FIBREOSS - desenvolvimento

A.1.1 Introdução

Neste projeto será desenvolvida uma interface capaz de agregar e coletar informações do estado da infraestrutura de rede IP e da rede experimental. Esse *dashboard* unificado deve comunicar-se com os sistemas que controlam e recolhem informações da infraestrutura de servidores, o sistema de monitoração da rede e com os *frameworks* de controle dos experimentos. A recuperação dos dados será feita programaticamente via chamadas a API JSON ao ZenOSS e perfSONAR, conexões diretas com bancos de dados do OCF e interagindo com as interfaces de usuário do OMF e do OCF a partir de requisições HTTPS (*Hyper Text Transmission Protocol Secure*) e emulações de navegador para processar o JavaScript necessário.

A.1.2 Montagem de um servidor piloto para agregar alarmes da ilha UFF

Cada ilha possui um servidor XEN onde diversas plataformas funcionam em cima de máquinas virtuais. Propõe-se criar o serviço do FIBREOSS no mesmo modelo. A máquina virtual tem um servidor Web configurado e um banco de dados.

Para realizar dessa tarefa, partiu-se de uma máquina virtual clone da omf-console, montada no servidor de virtualização da ilha da UFF (XEN). Um servidor HUB está sendo instalado na ilha da RNP e disponibilizará informações resumidas sobre o status das ilhas. Esse servidor atuará como um concentrador de informações.

Um clone do omf-console do NOC foi providenciado pelo operador do NOC, Daniel Marques.

Após a validação da proposta com esse projeto, um servidor *Leaf*, será instalado em cada uma das ilhas do FIBRE. Esse servidor deve realizar tarefas mais difíceis de controlar remotamente, ou que possam comprometer a escalabilidade do serviço.

Atualmente um clone do omf-console-uff está rodando na ilha UFF e possibilitando os testes da plataforma.

A.1.3 Desenvolvimento de uma interface de teste para os módulos

Foi desenvolvida uma interface de usuário dos módulos como uma extensão do portal da ilha (LS-WEB). As adições feitas até o momento incluem entradas no menu para os usuários do portal que possuem o *flag* de administrador. Sob a aba *Maintenance* podem ser encontrados os links: *Devices Offline*, *Availability Reports* e *Performance Issues*.

Figura A.1: Adições ao portal da ilha

Detalhes sobre a implementação podem ser encontrados no Anexo B.1.

A.1.4 Desenvolvimento do módulo de comunicação com o ZenOSS

A API de comunicação com o ZenOSS foi desenvolvida usando curl para php e um script em python. Ambos acessam a API JSON do ZenOSS.

O exemplo da API JSON do ZenOSS em python foi usado como base para a interface[97]. Para adicionar uma função ao portal foi necessário modificar a interface gráfica (LS-WEB) e a API (LS-Sched).

Mais detalhes sobre a implementação estão no Anexo B.2.

A.1.5 Desenvolvimento do módulo de comunicação com o perfSONAR

A API de comunicação com o perfSONAR exibe problemas de desempenho na rede a partir de informações disponibilizadas pela API do Maddash. Na implementação proposta, um "*custom client*" deve obter medidas do *Measurement Archive*. O procedimento para adicionar o módulo no portal da ilha é o mesmo descrito na Secão A.1.4 de desenvolvimento da comunicação com o ZenOSS.

A.1.6 Desenvolvimento do módulo de comunicação e sondas do OMF

O agente de teste e o coletor de eventos estão na pasta "/var/www/cgi-bin/python-omf-scrap/"e serão disponibilizados junto com as máquinas virtuais do FIBREOSS.

Os testes realizados são armazenados na tabela 'omf_tests' do banco de dados 'fibreoss'. Esses mesmos resultados são disponibilizados para usuários e operadores no portal da ilha.

A.1.7 Desenvolvimento do módulo de comunicação e sondas do OCF

O agente de teste e o coletor de eventos estão na pasta "/var/www/cgi-bin/python-ocf-scrap/"e serão disponibilizados junto com as máquinas virtuais do FIBREOSS.

Os testes realizados são armazenados na tabela 'ocf_tests' do banco de dados 'fibreoss'. Esses mesmos resultados são disponibilizados para usuários e operadores no portal

da ilha.

A conexão direta ao banco de dados do OCF possibilita a coleta dos loga de erro na interface. Foi feita através de um túnel ssh.

A.1.8 Desenvolvimento do módulo de teste de ping

Com o uso da plataforma ZenOSS, foram identificadas algumas inconsistências no banco de dados de eventos e diversos erros nas datas de início e final. Alguns servidores que estão há muito tempo fora de serviço aparecem como online no relatório. Para contornar esse problema, foi implementado um *script multithread* que realiza testes de ping em todos dispositivos configurados no ZenOSS e armazena o resultado no banco de dados do FIBRE.

A.1.9 Desenvolvimento do relatório de disponibilidade

Dois componentes estão envolvidos da geração do relatório de disponibilidade: O coletor de eventos e o gerador de relatórios. O coletor de eventos gera a disponibilidade diária de cada elemento monitorado, após o fechamento do dia o gerador de relatórios busca os valores de disponibilidade de todos os dias da semana e gera um documento. Dessa forma, todos os dias será possível retirar um relatório com a disponibilidade dos últimos 7 dias.

O mecanismo de agregação utiliza a árvore de dependências para avaliar a disponibilidade de grupos abstratos de dispositivos como ilhas, testbeds e a federação como um todo.

Na tabela *'system_hierarchy'* do banco de dados 'fibreoss' estão contidos os nomes dos dispositivos monitorados e suas respectivas afiliações. Esta tabela é utilizada por um script em python que monta uma árvore de objetos *node* e *device* como já mostrado na Tabela 5.2. Depois que a árvore é montada, uma função recursiva calcula a disponibilidade de cada *node* ou serviço.

A.1.10 Teste do plano de dados no domínio Openflow

Em cada ilha foi criado um experimento com uma máquina virtual em cada *switch* OpenFlow disponível. Assim a partir de todos os clientes é testada o atraso e largura de banda na comunicação com o controlador e também velocidade de conexão com a Internet. Para

cada experimento, uma Virtual Local Area Network (VLAN) é designada para uso como plano dados do experimento. Nessa configuração, o cliente intermediário (*INTERMEDIATE*) executa o controlador POX e o *daemon* do iperf.

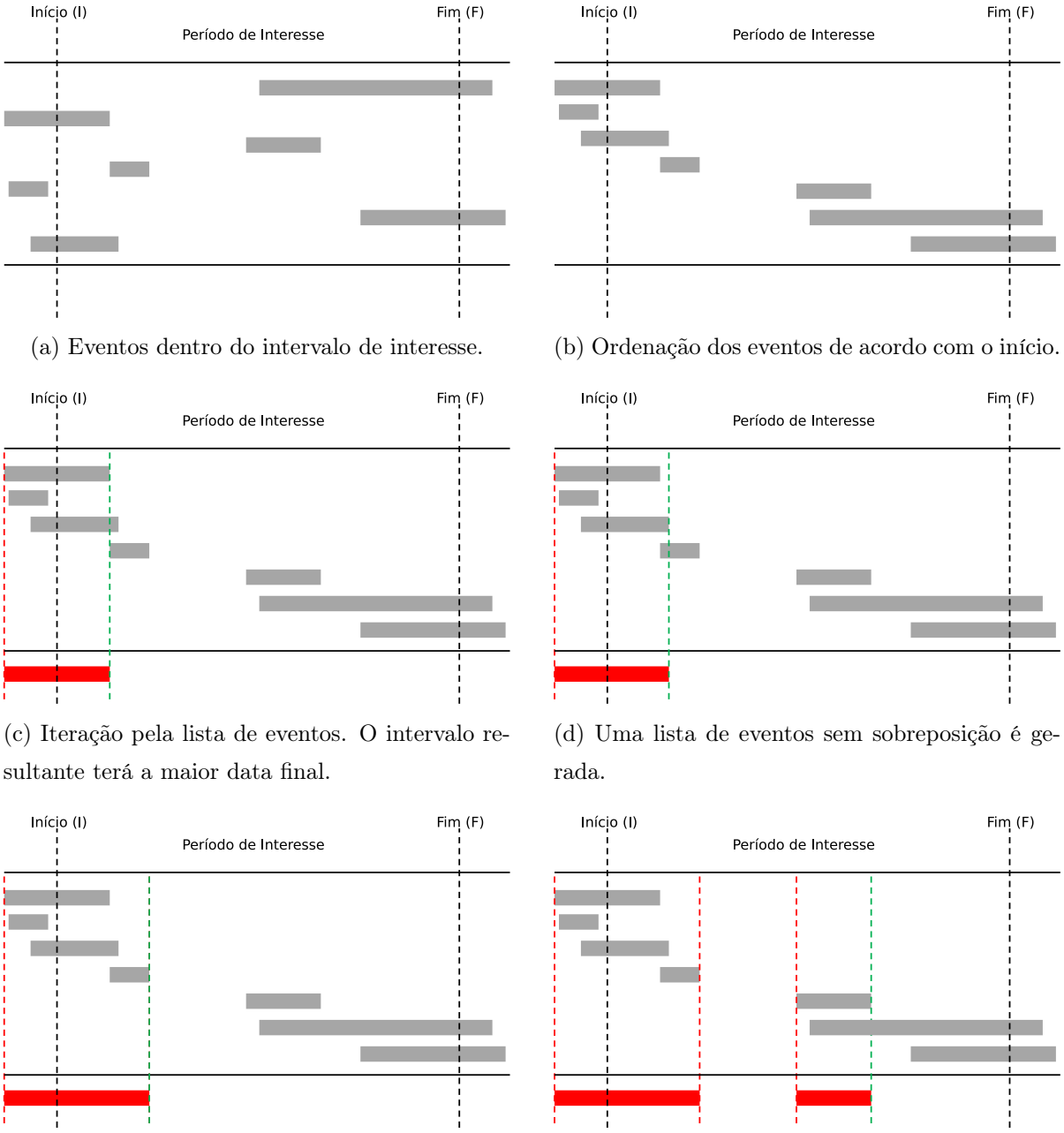
Configuração das máquinas virtuais:

```
#!/bin/bash
#ENDHOST 1
modprobe 8021q
vconfig add eth1 <VLAN>
ifconfig eth1 up
ifconfig eth1.<VLAN> 192.168.0.1
apt-get install python iperf git
wget --no-check-certificate  -O speedtest-cli \
https://raw.github.com/sivel/speedtest-cli/master/speedtest_cli.py
chmod +x speedtest-cli

#ENDHOST 2
modprobe 8021q
vconfig add eth1 <VLAN>
ifconfig eth1 up
ifconfig eth1.<VLAN> 192.168.0.3
apt-get install python iperf git
wget --no-check-certificate  -O speedtest-cli \
https://raw.github.com/sivel/speedtest-cli/master/speedtest_cli.py
chmod +x speedtest-cli

#ENDHOST 3
modprobe 8021q
vconfig add eth1 <VLAN>
ifconfig eth1 up
ifconfig eth1.<VLAN> 192.168.0.4
apt-get install python iperf git
wget --no-check-certificate  -O speedtest-cli \
https://raw.github.com/sivel/speedtest-cli/master/speedtest_cli.py
chmod +x speedtest-cli
```

```
#!/bin/bash
#INTERMEDIATE
modprobe 8021q
vconfig add eth1 <VLAN>
ifconfig eth1 up
ifconfig eth1.<VLAN> 192.168.0.2

apt-get install python iperf git
git clone http://github.com/noxrepo/pox
cd pox
git checkout dart
#start pox controller INFO:core:POX 0.3.0 (dart) is up.
python2.7 -u ./pox/pox.py forwarding.l2_learning web.webcore

#uses a lot of cpu
iperf -s -D



wget --no-check-certificate  -O speedtest-cli \
https://raw.github.com/sivel/speedtest-cli/master/speedtest_cli.py
chmod +x speedtest-cli
```

A.1.11 Cálculo da disponibilidade utilizando medidas pontuais e intervalos

A Figura A.2 mostra as etapas para avaliar a interseção de uma lista de eventos com início e fim definido. Dessa forma, pode-se verificar o somatório de diversos períodos de indisponibilidade de diversos dispositivos que estejam relacionados.



(a) Eventos dentro do intervalo de interesse.

(b) Ordenação dos eventos de acordo com o início.

(c) Iteração pela lista de eventos. O intervalo resultante terá a maior data final.

(d) Uma lista de eventos sem sobreposição é gerada.

(e) Retorna a interseção de todos os eventos.

(f) Corta a lista de resultados dentro do período de interesse.

Figura A.2: Método para avaliação da interseção de um conjunto de eventos.

A.1.12 Desenvolvimento de uma interface de usuário para o sistema

A interface com o usuário é disponibilizada no portal local das ilhas. O portal que serviu como base, o LS-WEB [91], consiste em uma API e uma interface web desenvolvidas em PHP. Portanto a interface de usuário do FIBREOSS foi desenvolvida como uma extensão

do LS-WEB para aproveitar toda a integração que o portal já possui com o FIBRE.

O FIBREOSS utiliza um banco de dados postgreSQL, que é o mesmo utilizado pelo portal para armazenar informações de usuários locais, reservas de recursos e configurações locais da ilha. Além disso, a biblioteca datatables [115] foi adicionada para formatar diversas tabelas que foram acrescentadas à interface. O datatables a partir da versão 1.10 possui licensa MIT, que permite o livre uso para projetos comerciais.

O datatables 1.10.5 foi instalado no diretório "/var/www/lib/datatables/1.10.5/".

As interfaces que já estão disponíveis estão nas Figuras: A.3, A.4, A.5, A.6, A.7 e , A.8.

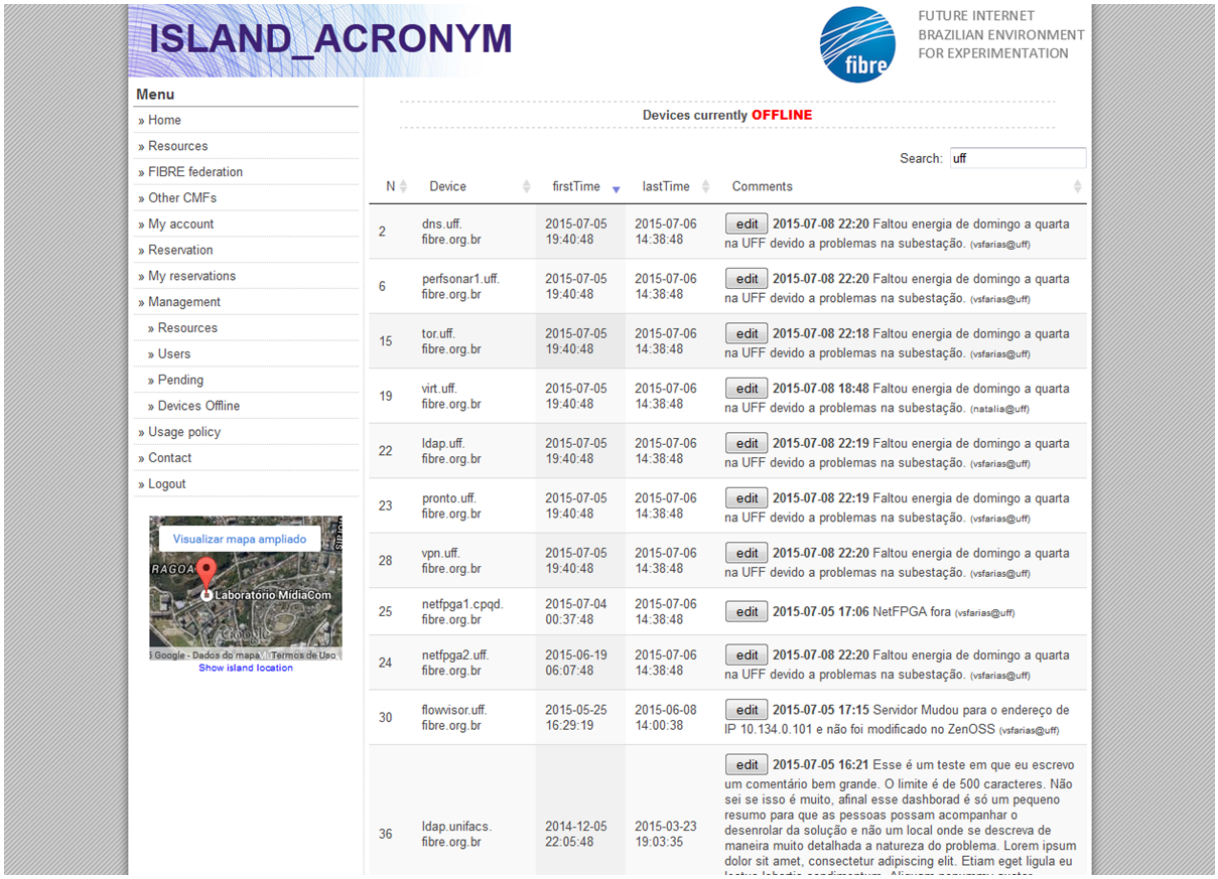


Figura A.3: Tela de inserção de comentários - Servidor fora

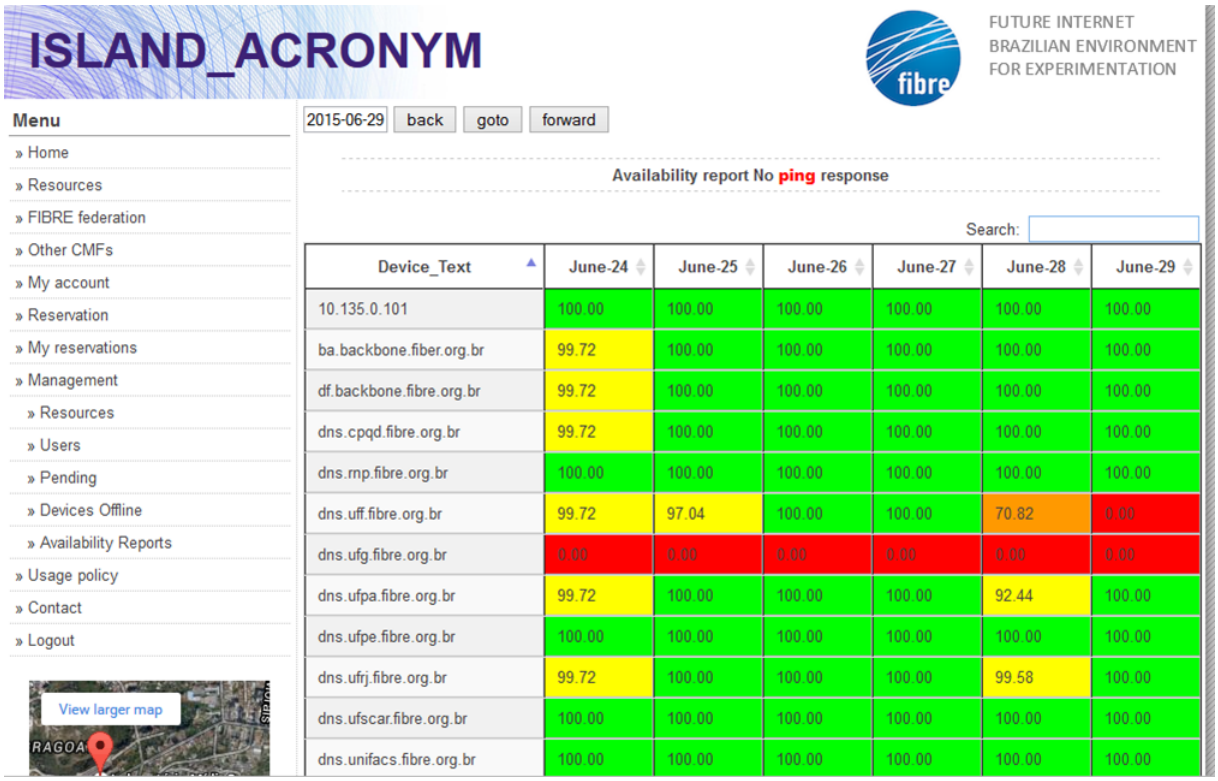


Figura A.4: Tela de exibição de relatórios - Ping

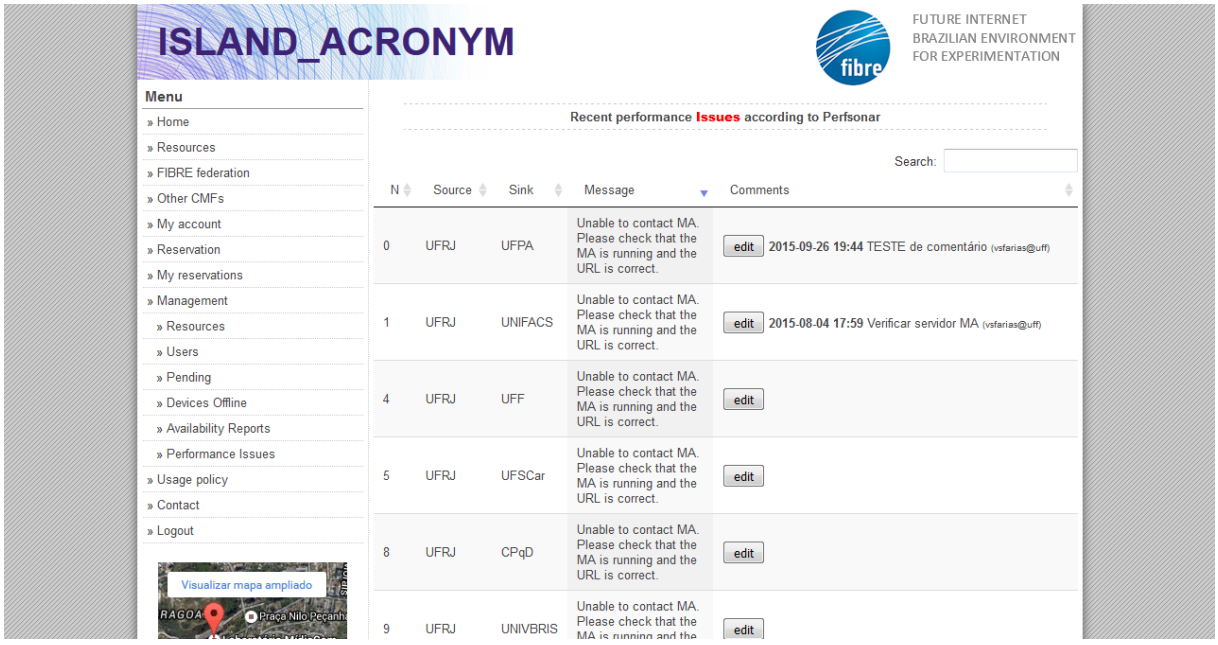


Figura A.5: Tela de inserção de comentários - Desempenho da rede

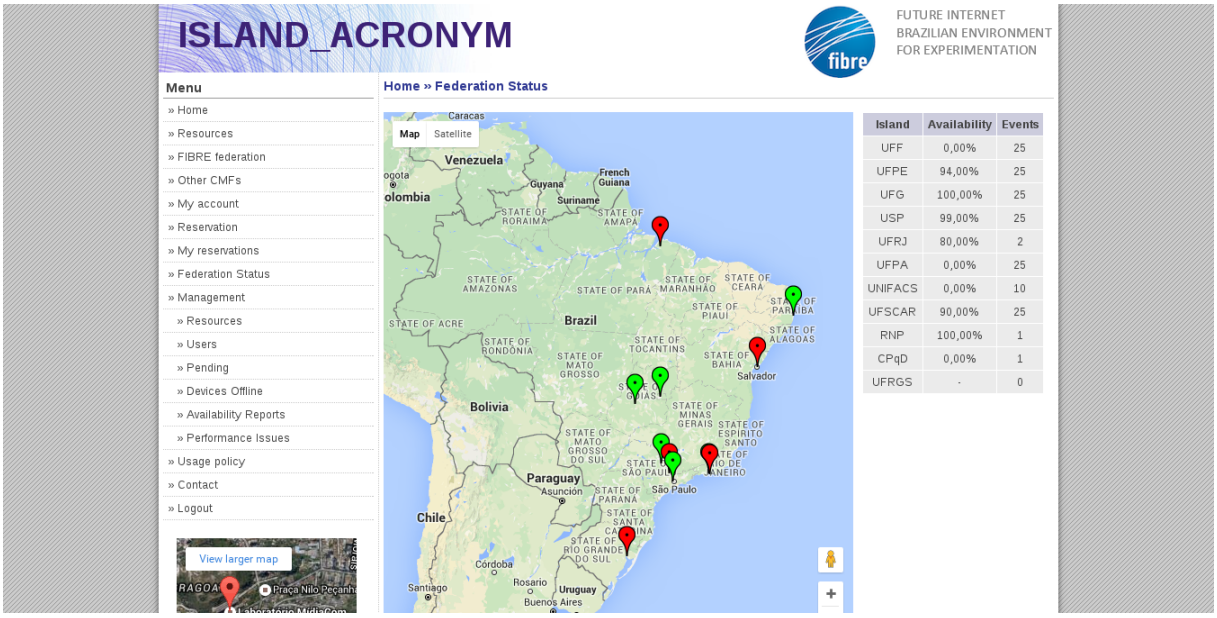


Figura A.6: Tela de status da federação FIBRE-BR

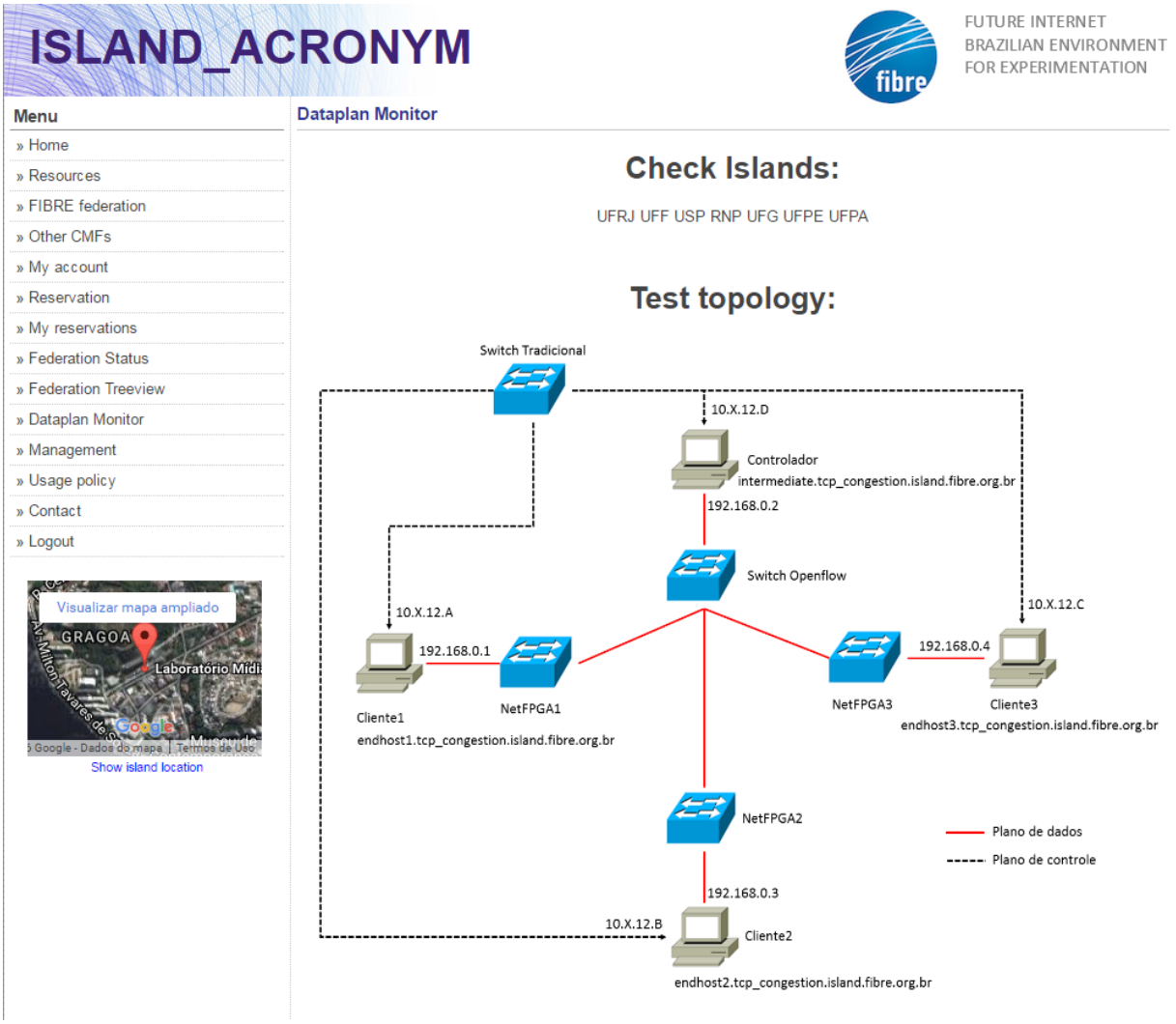


Figura A.7: Tela de monitoração do plano de dados Openflow - Parte1

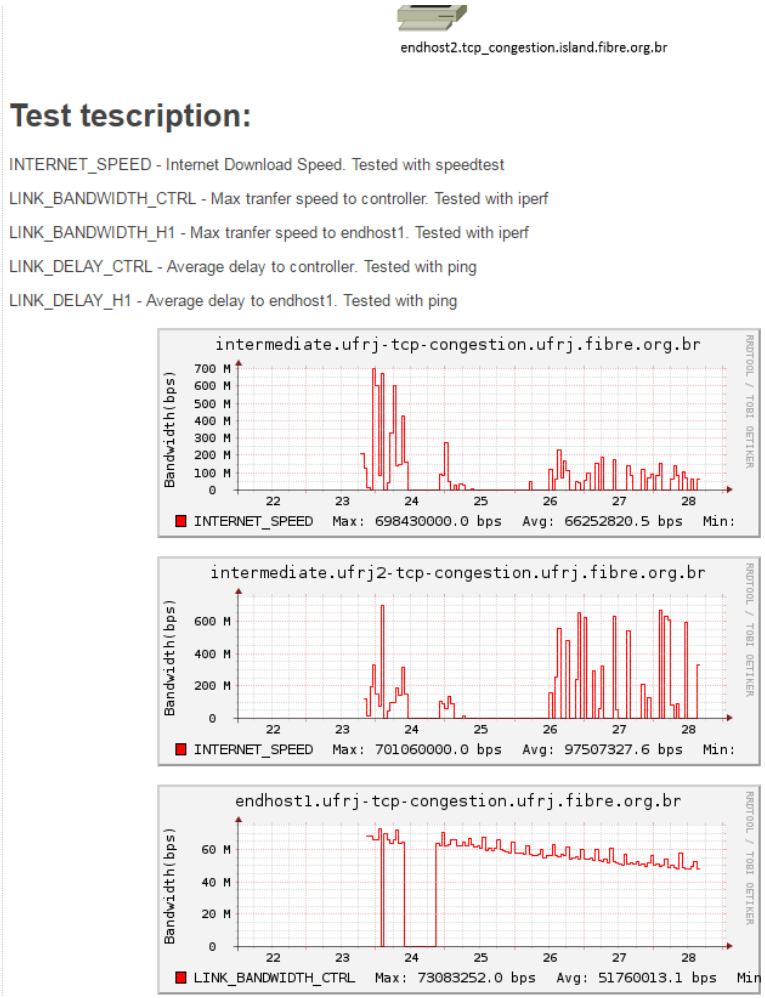


Figura A.8: Tela de monitoração do plano de dados Openflow - Parte2

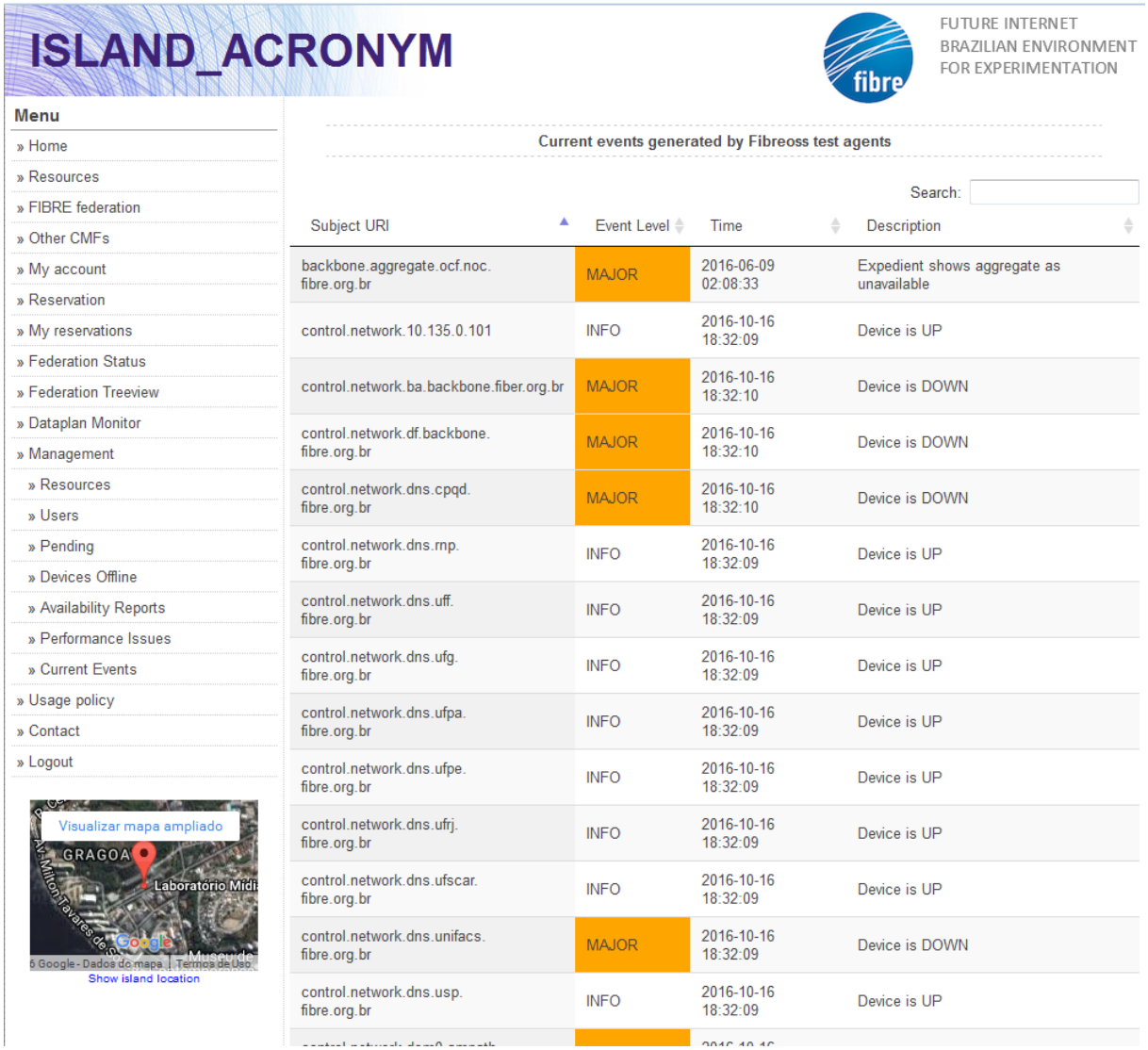


Figura A.9: Tela de eventos correntes

APÊNDICE B – Implementação do serviço web

B.1 Modificações no LS-WEB para visualização dos itens monitorados

Links foram adicionados na página principal. Arquivo: “/var/www/LS-WEB/index.php"

```
%\begin{lstlisting}[language=PHP, caption=Menu,basicstyle=\tiny]

if($_SESSION[''.SECURITY_SESSION.''][’admin’] == ’t’) {
echo ‘‘<a href=’javascript:void()’ id=’menu_management_button’>» Management</a>";
echo ‘‘<div id=’menu_management’>
<a href=’index.php?page=manager_resources’>» Resources</a>
<a href=’index.php?page=manager_user’>» Users</a>
<a href=’index.php?page=pendency’>» Pending</a>
<a href=’index.php?page=devices_down’>» Devices Offline</a>
<a href=’index.php?page=availability’>» Availability Reports</a>
<a href=’index.php?page=perfsonar’>» Performance Issues</a>
  </div>";
}


%\end{lstlisting}
```

Novos parâmetros de configuração foram inseridos em “/var/www/LS-WEB/config.php"

```
    //ADD by vsfarias 2015-06-29
//-------------------------------------
// ZenOSS API control
//-------------------------------------
```

```
define("ZENOSS_INSTANCE", "mon.fibre.org.br");
define("ZENOSS_PORT", "8080");
define("ZENOSS_USERNAME", "**********");
define("ZENOSS_PASSWORD", "**********");


//ADD by vsfarias 2015-07-20
define("DATASOURCES","/var/www/datasources/");

//ADD by vsfarias 2015-08-03
//-------------------------------------
// Perfsonar API control
//-------------------------------------

define("PERFSONAR_INSTANCE", "ps.fibre.org.br");
//define("PERFSONAR_PORT", "8080");
//define("PERFSONAR_USERNAME", "**********");
//define("PERFSONAR_PASSWORD", "**********");
```

No arquivo “/var/www/LS-WEB/pages.php"foram adicionadas as chamadas no trecho de código que processa o parâmetro “page"do GET.

```
case "devices_down" :
validating_authentication ( 'admin' );
devices_down ();
break;
case "availability" :
validating_authentication ( 'admin' );
availability ();
break;
case "perfsonar" :
validating_authentication ( 'admin' );
perfsonar ();
break;
```

E por fim, no arquivo “/var/www/LS-WEB/global_modules"foi modificada a função

"bread_crumbs()"

```
function bread_crumbs($position = NULL, $first = FALSE) {
$page = (isset ( $position ) && $position != NULL) ? $position : (isset ( $_GET ['p

switch ($page) {

[...]

case "devices_down" :
$location = "home";
$bc_title = "Devices Offline";
break;
case "availability" :
$location = "home";
$bc_title = "Availability Reports";
break;
case "perfsonar" :
$location = "home";
$bc_title = "Performance Issues";
break;
```

B.2 Implementação da interface de comunicação com o ZenOSS

No LS-WEB foi necessário adicionar cases na função “api_call"no arquivo “/var/www/LS-WEB/global_modules.php"

```
function api_call($method, $parameters = NULL) {
$url = LABORA_API_BASE_URL . "?method=" . $method . "&key=" . LABORA_API_KEY;

switch ($method) {
```

```
[...]

case "zenoss_get_events" :
$method_parameters = array (
'device_uid'
);
break;
case "zenoss_get_current_events" :
$method_parameters = array (
'eventClass',
'eventState',
'severity',
'prodState'
);
break;
case "zenoss_get_event_comments" :
$method_parameters = array (
'evids'
);
break;
case "zenoss_insert_event_comments" :
$method_parameters = array (
'event'
);
break;
```

E a função “devices_down()"no arquivo “/var/www/LS-WEB/pages.php"

No LS-Sched foram adicionadas as novas funções no arquivo “/var/www/LS-Sched/index.php"

```
[...]

$zenoss_methods = array(
'zenoss_test',
'zenoss_get_events',
```

```
'zenoss_get_current_events',
'zenoss_get_event_comments',
'zenoss_insert_event_comments'
);

[...]

    else if(in_array($api_call, $zenoss_methods))
require("zenoss.php");

[...]

//--------------------------------------------------------------------------------
case "zenoss_get_events":
$obligatory_parameters = array('device_uid' => 'string');
break;
case "zenoss_get_current_events":
$obligatory_parameters = array('eventClass' => 'string');
$optional_parameters = array('eventState' => 'array', 'severity' => 'array', 'prodS
break;
case "zenoss_get_event_comments":
$obligatory_parameters = array('evids' => 'array');
break;
case "zenoss_insert_event_comments":
$obligatory_parameters = array('event' => 'array');
break;


[...]

//--------------------------------------------------------------------------------
case "zenoss_test":
$return['method_result'] = zenoss_test();
break;
case "zenoss_get_events":
```

```
$return['method_result'] = zenoss_get_events($received_parameters['device_uid']);
break;
case "zenoss_get_current_events":
$return['method_result'] = zenoss_get_current_events($received_parameters['eventCla
break;
case "zenoss_get_event_comments":
$return['method_result'] = get_event_comments($received_parameters['evids']);
break;
case "zenoss_insert_event_comments":
$return['method_result'] = insert_event_comments($received_parameters['event']);
break;

[...]
```

E o arquivo que contém as funções chamadas pela API “/var/www/LS-Sched/zenoss.php"em conjunto com o módulo de interface licenciado GPLv3[116] em php que utiliza curl. No diretório “/var/www/LS-Sched/php-zenoss-api/zenoss.php"

Para exibir o relatório de disponibilidade, foi adicionada a função “availability()"no arquivo “/var/www/LS-WEB/pages.php"

A mesma abre os relatórios previamente gerados guardados no diretório “/var/www/datasources"

Scripts foram adicionados pasta “/var/www/cgi-bin/zenoss-api/"

APÊNDICE C – Relatório de requisitos

C.1 Especificação de funcionalidades e requisitos de monitoração de usuários e de operadores

Para fins de definir os requisitos do sistema, foi necessário, primeiramente, detalhar os requisitos de monitoração de acordo com a visão de usuários e de operadores. A lista definida e discutida no contexto desse projeto é apresentada a seguir. Também são discutidos os requisitos do sistema de correlação de alarmes proposto.

C.1.1 Requisitos dos Operadores

Nessa seção, são listados os requisitos que ajudam na operação do FIBRE e como eles podem ser alcançados.

- Dashboard com nós não disponíveis
  - **Objetivo:** Um painel com o status dos nós sem-fio e switches OpenFlow e histórico de disponibilidade deve estar disponível. Esses dados são obtidos usando a ferramenta \Status\Ping do ZenOSS.
  - **Requisitos:** Acrescentar os nós sem fio na plataforma do ZenOSS em um grupo de hierarquia inferior à ilha.
- Dashboard com servidores não disponíveis
  - **Objetivo:** Mostrar eventos de *device down* usando o (\Status\Ping) do ZenOSS e permitir que os operadores comentem a respeito do problema.
  - **Requisitos:** Embora os eventos possam ser recolhidos com o ZenOSS, é necessário uma interface para a explicação dos operadores sobre os problemas.
- Relatório de disponibilidade

  - **Objetivo:** Esse relatório deve conter informações detalhadas do estado funcionamento dos componentes do FIBRE, incluíndo disponibilidade do servidor e dos seus serviços (ex. Interface Web do portal e suas capacidades).
  - **Requisitos:** Desenvolvimento de interface.

- Uso dos enlaces
  - **Objetivo**: Monitorar todos os enlaces (vazão, perda e atraso).
  - **Requisitos:** Essa tarefa é feita pelo Maddash do perfSONAR. Uma interface dentro do portal da ilha pode ser disponibilizada.

- Uso dos enlaces por experimento
  - **Objetivo:** Correlacionar com o uso do *testbed* com as reservas de recurso do OMF e do OCF.
  - **Requisitos:**
    * OMF - verificar se há reserva de recurso e o access_log do OMF e dos nós.
    * OCF - verificar se o controlador foi acessado ou está em funcionamento. Verificar a base de dados do OCF e procurar a instância da máquina virtual do controlador.

- Número de usuários experimentadores
  - **Objetivo**: Definir o número de usuários por ilha e na federação ao longo do tempo.
  - **Requisitos**: Desenvolvimento de módulo de consulta ao LDAP com interface web.

- Número de usuário ativos no tempo
  - **Objetivo**: Definir usuários ativos em função do tempo (mostrar quantos usuários fazem experimentos ao mesmo tempo).
  - **Requisitos**: Desenvolvimento de módulo de consulta ao logs do OMF e do OCF com interface web.

- Uso de recursos nos servidores
  - **Objetivo**: O ZenOSS faz exatamente o proposto. Monitora em tempo real e disponibiliza histórico.

  - **Requisitos**: Estender o uso do ZenOSS para mostrar todas as variáveis de interesse:
    * Servidor Xen - Monitorar uso da CPU, memória, I/O, número de processos, número de máquinas virtuais ativas e uso de disco
    * Máquinas virtuais do OCF - Monitorar uso da CPU, memória, I/O, número de processos e uso de disco

- Uso da federação
  - **Objetivo**: Monitorar:
    * Enlaces ativos
    * Interligação entre os softwares
    * Portais federados
    * Autenticação federada
  - **Requisitos**: A monitoração de enlaces ativos pode ser feita como uma estensão do uso dos enlaces locais para a rede de controle. Os portais, o funcionamento dos softwares e a autenticação federada podem ser feitos através da simulação de uso do testbed por meio de scripts, a qual deve ser escalonada em testes periódicos. Todo o código e a interface com os resultados precisam ser desenvolvidos.

- Teste periódico dos serviços
  - **Objetivo**: Testar a disponibilidade dos serviços:
    * Flowvisor
    * Orbit Management Framework (OMF)
    * OFELIA Control Framework (OCF)
    * Lightweight Directory Access Protocol (LDAP)
    * Virtual Private Network (VPN)
  - **Requisitos**: Todos esses serviços podem ser automaticamente testados pelo script que simula o uso do *testbed*. A disponibilidade também pode ser observada com menos precisão com testes descritos para o Dashboard com servidores não disponíveis.

C.1.2 Requisitos de usuários

Nessa seção, são listados os requisitos que ajudam no uso do FIBRE com relação à monitoração e como eles podem ser alcançados. Cabe observar que aqui não são apresentadas ferramentas para monitoração do experimento, mas ferramentas que fornecem dados sobre disponibildiade e estabilidade do ambiente de teste que podem ser importantes para o usuário na hora de escolher quais recursos usar e em que momentos usar.

- Dashboard e mapa de nós sem fio ativos
  - **Objetivo**: Mostrar o mapa dos nós sem fio ativos no momento atual.
  - **Requisitos**: Esse *dashboard* terá os eventos gerados pelo ZenOSS e mostrará os resultados usando um mapa[1], para que o usuário tenha uma visão geográfica do experimento.
- Histórico de falhas e comentários
  - **Objetivo**: Permitir que, antes de fazer a reserva dos recursos, o usuário seja capaz de saber se os nós e se a ilha estão estáveis.
  - **Requisitos**: Interface com resumo para o usuários dos dados de monitoração de nós e serviços que tenham implicação direta no experimento do usuário.
- Disponibilizar a monitoração do uso do espectro dos nós sem fio no LS-WEB
  - **Objetivo**: Disponibilizar os resultados dos nós de monitoração podem ser disponibilizados na interface web.
  - **Requisitos**: Interface já está pronta, necessitando ajuste sobre o período que deve ser monitorado.
- Disponibilizar tabelas de fluxo do OpenFlow
  - **Objetivo**: A visão das tabelas de fluxo dos switchs devem estar disponíveis aos usuários para depuração das aplicações de controle desenvolvidas.
  - **Requisitos**: Desenvolver e disponibilizar interface que permita que o usuário veja o estado das tabelas de fluxo a qualquer momento em qualquer switch da rede.
- Visão do banco de dados do OML por experimento

[1] a sugestão é usar o mesmo *plugin* (openmaps) do Viaipe - http://viaipe.rnp.br/)

  - **Objetivo**: Quando um experimento é realizado, os resultados do experimento são armazenados em um banco de dados. Esses dados devem ser disponibilizados no LS-WEB.
  - **Requisitos**: Desenvolver uma interface integrando o banco de dados com o LS-WEB.

- Estatística de uso do enlace do backbone
  - **Objetivo**: Permitir que o usuário possa estimar se alguma sobrecarga no backbone interferiu no seu experimento.
  - **Requisitos**: Desenvolver módulo de monitoração ativa com baixo impacto usando o perfSONAR para estimar o uso dos enlaces da rede de dados no backbone.

C.1.3 Requisitos do sistema de correlação de alarmes

O sistema de correlação de alarmes está começando a ser projetado nesse momento, então apenas os requisitos iniciais foram levantados até o momento, conforme mostrado a seguir.

- Correlacionar experimentos com erros/anormalidades observadas na rede.
- Correlacionar alarmes de problemas no Xen e VMs com paradas no funcionamento do FIBRE.
- Gerar alarmes de ambiente de experimentação inapropriado para uso.
- Gerar alarmes de experimentos que estão prejudicando o testbed.