



10<sup>a</sup> ESCOLA REGIONAL DE REDES DE COMPUTADORES  
1–3 de outubro de 2012  
Pelotas – RS

# ANAIIS

**Editora**  
Sociedade Brasileira de Computação – SBC

**Organizadores**  
Maurício Lima Pilla  
Gerson Geraldo Homrich Cavalheiro  
Cristiano Bonato Both  
André Rauber Du Bois

**Realização**  
Universidade Federal de Pelotas

**Promoção**  
Sociedade Brasileira de Computação – SBC

Copyright © 2012 Sociedade Brasileira de Computação

Capa: Rodolfo Favaretto

Supervisão Gráfica: Maurício Lima Pilla

Impressão: Gráfica UFPel

### CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Escola Regional de Redes de Computadores (10.: 1–3 out 2012: Pelotas)

Anais / Organizadores: Maurício Lima Pilla, Gerson Geraldo Homrich Cavalheiro, Cristiano Bonato Both, André Rauber Du Bois.  
— Pelotas: Gráfica UFPel, 2012.

130 f.: il.

ISSN 2237-3748

Conhecido também como ERRC 2012.

1. Redes de Computadores. 2. Sistemas Distribuídos. I. ERRC (10.: 1–3 out 2012: Pelotas). II. **UFPel**. III. Pilla, Maurício Lima. IV. Cavalheiro, Gerson Geraldo Homrich. V. Both, Cristiano Bonato. VI. Du Bois, André Rauber. VII. Título.

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

# **ERRC 2012**

<http://lups.inf.ufpel.edu.br/errc2012>

## **Comitê de Programa**

Adenauer Yamin (UCPEL/UFPEL)  
André Peres (IFRS - Campus Porto Alegre)  
André Rauber Du Bois (UFPel)  
Andrea Charao (UFSM)  
Antônio Rodrigo Delepiane de Vit (UFSM)  
Atila Vasconcelos (UNIRITTER)  
Carlos Raniery Paula dos Santos (UFRGS)  
Clarissa Marquezan (Paluno, University of Duisburg-Essen)  
Cristian Koliver (UCS)  
Cristiano Bonato Both (UFRGS)  
Cristiano Cachapuz e Lima (URCAMP)  
Cristiano Costa (UNISINOS)  
Cristina Nunes (PUC-RS)  
Diego Kreutz (UNIPAMPA)  
Eduardo Monks (UCPEL)  
Elder Rodrigues (PUCRS)  
Erico Amaral (IFRS/UFRGS)  
Flávio Roberto Santos (UFRGS)  
Gerson Battisti (UNIJUI)  
Gerson G. H. Cavalheiro (UFPEL)  
Iara Augustin (UFSM)  
Jéferson Nobre (UFRGS)  
João Ladislau Lopes (IFSUL)  
Juergen Rochol (UFRGS)  
Juliano Wickboldt (UFRGS)  
Leonardo Pinho (UNIPAMPA)  
Lisandro Zambenedetti Granville (UFRGS)  
Luciano Paschoal Gaspary (UFRGS)  
Marco Trentin (UPF)  
Marinho Barcellos (UFRGS)  
Mateus Santin (UCPEL/UFPEL)  
Maurício Lima Pilla (UFPEL)  
Odorico Machado Mendizabal (FURG)  
Osmar Marchi dos Santos (UFSM)  
Rafael Avila (UNISINOS)  
Rafael Esteves (UFRGS)  
Rafael Kunst (UFRGS)  
Renata Reiser (UFPEL)  
Ricardo Neisse (Fraunhofer IESE)  
Ricardo Schmidt (University of Twente)  
Roben Lunardi (IFRS)

Rodrigo Calheiros (The University of Melbourne)  
Rodrigo Righi (UNISINOS)  
Rogério Turchetti (UFSM)  
Taisy Weber (UFRGS)  
Tiago Ferreto (PUCRS)  
Vinicius Ribeiro (UNIRITTER)  
Walter Priesnitz Filho (UFSM)

**Revisores Adicionais**

Andrea Krob (UNILASALLE)  
Angel Galvão (UFRGS)  
Daniel Marcon (UFRGS)  
Erico Rocha (UNISINOS)  
Felipe Carbone (UFRGS)  
Janaina Lemos (UNISINOS)  
José Santanna (UFRGS)  
Leonardo Faganello (UFRGS)  
Lucas Bondan (UFRGS)  
Luis Jersak (PUCRS)  
Maicon Kist (UFRGS)  
Marcelo Marotta (UFRGS)  
Matheus Cadori Nogueira (UFRGS)  
Oscar Caicedo (UFRGS)  
Paolo Cemim (PUCRS)  
Ricardo Luis dos Santos (UFRGS)  
Rodolfo Antunes (UFRGS)  
Rodrigo Mansilha (UFRGS)  
Rodrigo Ruas Oliveira (UFRGS)  
Wanderson Jesus (UFRGS)

## **Comitê Organizador**

### **Coordenação Geral**

Prof. Dr. Maurício Lima Pilla (UFPEL)

### **Organização Local**

Prof. Dr. Gerson Geraldo Homrich Cavalheiro (UFPEL)

### **Organização do Comitê de Programa**

Prof. Dr. Cristiano Bonato Both (UNISC)

Prof. Dr. André Rauber Du Bois (UFPEL)

### **Organização de Divulgação e Patrocínios**

Prof. Dr. Adenauer Correa Yamin (UFPEL)

### **Organização de Minicursos e Palestras**

Prof. Dr. Rodrigo da Rosa Righi (UNISINOS)

Prof. Dr. Raul Ceretta Nunes (UFSM)

### **Organização de Atividades Sociais**

Prof. Dr. Cristiano André da Costa (UNISINOS)

### **Organização de Inscrições e Certificados**

Profa. Dra. Renata Hax Sander Reiser (UFPEL)

### **Divulgação e Patrocínios Locais**

Prof. M.Sc. Eduardo Maroñas Monks (SENAC Pelotas)

Prof. Dr. Leonardo Bidese de Pinho (UNIPAMPA Bagé)

Prof. M.Sc. Érico Hoff do Amaral (UNIPAMPA Bagé)

Prof. M.Sc. Odorico Machado Mendizabal (FURG)

Prof. M.Sc. Sérgio Luis Rodrigues (IFSul Pelotas)

Profa. Dra. Fabiane Marroni (UCPEL)

### **Comissão de Organização**

Profa. Dra. Ana Marilza Pernas (UFPEL)

Alan Schlindvein (Bacharelando CIC - UFPEL)

Deives M. Kist (PPGC - UFPEL)

Felipe L. Teixeira (Bacharelando CIC - UFPEL)

Giovane Oliveira Torres (Bacharelando CIC - UFPEL)

Ibero Benitez (PPGC - UFPEL)

Lidiane Visintin (PPGC - UFPEL)

Matheus Nachtigall (PPGC - UFPEL)

Rodrigo M. Duarte (Bacharelando Eng. C. - UFPEL)

Rodolfo M. Favaretto (PPGC - UFPEL)

Timóteo M. Rico (PPGC - UFPEL)

Tymon J. Douglas (Bacharelando CIC - UFPEL)

Vilnei M. F. Neves (PPGC - UFPEL)



# Apresentação

Com imensa alegria apresentamos a 10<sup>a</sup> Escola Regional de Redes de Computadores (ERRC 2012). Neste ano o evento é organizado pela Universidade Federal de Pelotas (UFPel) e promovido pela Sociedade Brasileira de Computação (SBC), ocorrendo de 1 a 3 de outubro, em Pelotas, Rio Grande do Sul. A ERRC é um evento já tradicional que tem por objetivo reunir pesquisadores, estudantes e membros da indústria, ligados à área de redes de computadores e afins no Rio Grande do Sul. O evento conta com palestras, minicursos e painéis discutindo temas atuais e relevantes da área.

Um dos principais pontos do evento está relacionado às sessões onde são apresentados trabalhos de Iniciação Científica e de Pós-Graduação. Os artigos são inicialmente revisados em um processo onde pelo menos três revisores avaliaram cada artigo, a fim de garantir a qualidade e, ao mesmo tempo, apresentar aos autores sugestões relevantes para seus trabalhos. Desta forma, agradecemos ao Comitê de Programa e revisores pelo excelente trabalho na seleção dos textos que compõem este livro. Nesta Escola, 15 artigos de Iniciação Científica de um total de 26 foram aceitos para publicação e apresentação. Dentre os artigos de Pós-Graduação, foram aceitos 11 de um total de 15 submetidos.

Para tornar a programação interessante para os alunos e profissionais da área, um conjunto de palestrantes foi convidado e aceitou o desafio de trazer o estado-da-arte em redes de computadores para discussão. A equipe do Curso de Tecnólogo em Redes de Computadores do SENAC/Pelotas, coordenada pelo Prof. Eduardo Monks, gentilmente promoveu quatro mini-tutoriais em aspectos práticos também relacionados a redes de computadores.

Finalmente, sem as pessoas das diversas comissões que nos ajudaram a organizar a ERRC, este evento não ocorreria. Também agradecemos o patrocínio da Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS), da SAP e da Eckert-Caine, os quais acreditaram no evento e o viabilizaram. Estendemos nossa saudação a UNISINOS, UFSM, SENAC Pelotas, UNIPAMPA Bagé, FURG, IFSul Pelotas e UCPEL pela contribuição de seus professores. Por fim, agradecemos à UFPel e sua Gráfica pelo apoio incondicional pra realização deste evento.

Desejamos a todos os participantes que aproveitem bem a estadia em Pelotas e que tenham uma excelente Escola.

Maurício Lima Pilla  
Gerson Geraldo Homrich Cavalheiro  
Cristiano Bonato Both  
André Rauber Du Bois

Pelotas, outubro de 2012.



# Sumário

<b>I Gerenciamento e Desempenho</b>	1
<b>OFNMS: Gerenciamento de Redes com OpenFlow</b>	
J. Nickel, C. Nunes .....	3
<b>Análise de Desempenho do Protocolo M-DART</b>	
B. Nieto, A. Krob .....	7
<b>Monitoramento e Análise do Impacto no Desempenho em Ambientes Virtualizados</b>	
P. Popolek, O. Mendizabal .....	11
<b>NMS Anywhere: Uma Aplicação Voltada a Apoiar o Gerenciamento de Redes Através de Plataforma Móvel</b>	
F. Pandolfo, C. Nunes .....	15
<b>II Aplicações, Medições e Monitoramento</b>	19
<b>Backup distribuído: uma implementação funcional</b>	
C. Sobral, A. Coêlho .....	21
<b>Influência do Cenário em Rede de Sensores Sem Fio para Rastreamento Animal</b>	
M. Deangelo, P. Domingues, L. Pinho .....	25
<b>Acesso Gratuito A Internet - Uma proposta de cadastro e autenticação para acesso à Internet em locais públicos</b>	
M. Borba, R. Ávila .....	29
<b>III Segurança e Redes sem Fio</b>	33
<b>Proposta e Implementação de um Firewall para Aplicações Web Denominado UniscanWAF</b>	
A. Del Fabro Neto, R. Turchetti, C. Trois, W. Priesnitz Filho, D. Rocha, D. Kreutz .....	35
<b>Uma arquitetura para desenvolvimento de dispositivos de autenticação e acesso a espaços físicos</b>	
J. Raguzzoni, L. Heinsch, T. A. Rizzetti .....	39
<b>Monitoramento Automático de Falha em Transformadores de Redes de Distribuição de Energia Elétrica Utilizando Tecnologia ZigBee</b>	
T. Saidelles, C. Colvero .....	43
<b>Análise de desempenho de redes sem fio com diferentes protocolos de criptografia</b>	
D. Stangarlin, W. Priesnitz Filho .....	47

<b>IV Computação em Nuvem</b>	51
<b>PyCloud: Compartilhamento em nuvem local</b>	
J. Rosa, E. Monks .....	53
<b>Computação em Nuvem com Google Apps for Education: o Caso do Núcleo de Ciência da Computação da Universidade Federal de Santa Maria</b>	
E. Scheid, L. Minato, B. Stein, A. Charão .....	57
<b>Serviço de Presença sobre uma Estrutura Gossip em Cloud</b>	
P. Wilges, H. Lovison, S. Cechin, T. Weber, R. Moraes .....	61
<b>Virtualizando com Xen Cloud Platform (XCP)</b>	
M. Alves, A. Moraes .....	65
<b>V Fórum de Pós-Graduação I</b>	69
<b>Alta disponibilidade em redes IPv6 críticas utilizando o protocolo CARP</b>	
C. Kitahara, L. Soeiro, A. Souza .....	71
<b>Modelagem de uma Base de Conhecimento para o Monitoramento de Ataques</b>	
G. Petri, T. Ceolin Junior, R. Nunes, O. Santos .....	75
<b>Uso de OSPF para convergência de túneis IPSec</b>	
M. Camocardi, V. Muniz, A. Sousa .....	79
<b>Service Desk Móvel com Retenção de Conhecimento e Sensível ao Contexto</b>	
T. Oliveira, R. Medina .....	83
<b>VI Fórum de Pós-Graduação II</b>	87
<b>CEP: Uma proposta de gerenciamento de identidades em Cloud Computing utilizando OpenAM e Captive Portal</b>	
A. Mühlbeier, F. Nunes, G. Voss, S. Stieler, R. Medina, E. Amaral .....	89
<b>Ambiente de acesso seguro a nuvem privada: uma proposta voltada à rede da UNIPAMPA</b>	
D. Borges, M. Sulzbach, A. Charão, B. Stein, R. Medina .....	93
<b>Utilização de técnicas de paralelismo para desenvolvimento de uma ferramenta com alto desempenho para varreduras de dispositivos de rede, escrita em linguagem C utilizando as bibliotecas Socket e OpenMP</b>	
C. Machado .....	97
<b>VII Fórum de Pós-Graduação III</b>	101
<b>Avaliação de Desempenho em Canal de Retorno de TV Digital baseado em Redes Mesh IEEE 802.11</b>	
S. Neves, E. Silva, R. Viégas Jr .....	103
<b>Multiflow: Multicast clean-slate com cálculo antecipado das rotas em redes programáveis OpenFlow</b>	
L. Bondan, L. Muller, M. Kist .....	107

---

<b>Balanceamento de Carga em Sistema de Transações Eletrônicas Financeiras com RMI</b>	
A. Andrade, C. Costa, T. Jost, R. Righi .....	111
<b>Estudo da viabilidade do ROS como plataforma para IoT</b>	
V. Hax, N. Duarte Filho, S. Botelho, O. Mendizabal .....	115



---

I

# **Gerenciamento e Desempenho**

---



# OFNMS: Gerenciamento de Redes com OpenFlow

Jones F. R. Nickel

Faculdade de Informática – PUCRS

jones.nickel@acad.pucrs.br

Cristina M. Nunes

Faculdade de Informática – PUCRS

cristina.nunes@pucrs.br

**Resumo** — Este trabalho apresenta uma ferramenta para o monitoramento de *switches* em redes OpenFlow, visando ajudar o administrador de redes nas tarefas do dia a dia. A ferramenta possibilita a coleta de dados estatísticos de interfaces e fluxos nos *switches* da rede, apresentando estes dados graficamente.

## I. INTRODUÇÃO

O protocolo OpenFlow [2][4] vem prover-nos uma forma centralizada de controle da lógica de encaminhamento de pacotes da rede. A partir deste, um controlador se conecta nos *switches* de forma centralizada e manipula suas tabelas de encaminhamento de mensagens [2][4]. Ao chegar um pacote a um determinado *switch*, este verifica se existe alguma regra para o pacote, se existir aplica-a e o encaminha, caso contrário envia o pacote para o controlador decidir.

Este trabalho visa explorar as novas tecnologias providas pelo OpenFlow. Com base nas informações apresentadas a seguir, este trabalho descreve a implementação de uma ferramenta de monitoramento para redes OpenFlow, chamada de OFNMS (*OpenFlow Network Monitor System*). Tal ferramenta é baseada no controlador NOX [5] [6], com o objetivo de persistir dados dos switches da rede e expor estes dados na forma de informações úteis para a tomada de decisão na gerência dos recursos da mesma. A ferramenta possui uma interface web e apresenta os dados por meio de gráficos.

Este documento está organizado da seguinte forma: na Seção II são apresentados conceitos do protocolo OpenFlow, e a seguir o controlador NOX que serviu de base para este trabalho. A Seção III apresenta as principais ferramentas e trabalhos de pesquisa em destaque, baseados no protocolo OpenFlow e no controlador NOX. A Seção IV apresenta detalhes do desenvolvimento da ferramenta descrita neste trabalho. A Seção V descreve os objetivos alcançados, mostrando brevemente a ferramenta. Por fim, na Seção VI são apresentadas as considerações finais, apresentando a conclusão sobre este trabalho e trabalhos futuros que poderão complementá-lo.

## II. FUNDAMENTAÇÃO TEÓRICA

A seguir serão apresentados alguns conceitos fundamentais para que se possa haver um melhor entendimento sobre a arquitetura e o funcionamento de uma rede OpenFlow. Também será apresentado o NOX, que serviu de base para a implementação do OFNMS (*OpenFlow Network Monitor System*).

### A. OpenFlow

OpenFlow [2][3][4] é um protocolo aberto que permite estabelecer conexões entre um “Controlador Remoto” e os *switches* e roteadores que fazem parte da topologia da rede. O OpenFlow permite que o controlador manipule

remotamente as regras de encaminhamento de mensagens dos equipamentos, obtendo informações dos mesmos e possibilitando a alteração dessas regras.

A arquitetura OpenFlow possui os componentes apresentados na Figura 1 e descritos a seguir:

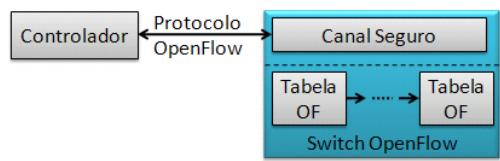


Figura 1 – Arquitetura OpenFlow.

1) *Canal OpenFlow*: canal seguro que conecta cada *switch* OpenFlow a um controlador. Através desse canal, o controlador configura e gerencia o *switch*, recebe eventos e envia pacotes através do mesmo.

2) *Tabelas OpenFlow*: formam um *pipeline* de processamento de fluxo. Cada tabela de fluxo contém múltiplas entradas de fluxo. O *pipeline* OpenFlow define como os pacotes recebidos pelo *switch* interagem com as tabelas de fluxo [3].

3) *Protocolo OpenFlow*: as mensagens suportadas pelo protocolo incluem: “controlador para *switch*”, que tem início no controlador e são usadas para gerenciar ou inspecionar diretamente o estado do *switch*; “mensagens assíncronas”, que têm início no *switch* e são usadas para notificar o controlador sobre os eventos da rede e mudanças no estado do *switch*; e “mensagens simétricas”, que são iniciadas tanto pelo *switch* quanto pelo controlador e são enviadas sem solicitação.

### B. NOX: Um Sistema Operacional de Rede

O NOX é um controlador OpenFlow de código aberto [5][6]. Foi concebido para fornecer uma plataforma simples de desenvolvimento de aplicativos que controlem ou monitorem redes OpenFlow. O NOX é o que pode ser considerado como um “Sistema Operacional de Rede” [6]. Ele fornece uma API de alto nível que possibilita conectar-se aos *switches* OpenFlow, podendo alterar suas tabelas de fluxo, buscar estatísticas dos *switches* e consultar o estado da rede.

O NOX mantém uma visão da rede [6], ou seja, uma base de dados com a observação do estado da rede (*Network View*). As aplicações usam essa visão para tomar decisões de administração. Para o NOX controlar o tráfego da rede, ele deve manipular os *switches* que compõem a topologia.

Quando um pacote recebido pelo *switch* encontra uma entrada de fluxo em suas tabelas, o *switch* aplica as ações correspondentes a esse pacote. Se o pacote não encontrar uma entrada de fluxo, ele é encaminhado para o

controlador, no caso o NOX. A aplicação de controle que roda sobre a API do NOX decide a regra de controle para o pacote e adiciona essa regra na tabela de fluxo do *switch*. Após essa ação todos os pacotes com mesmo cabeçalho utilizarão esta mesma regra [5][6].

### III. TRABALHOS RELACIONADOS

A seguir estão apresentadas as principais ferramentas e trabalhos de pesquisa em destaque hoje baseados no protocolo OpenFlow e no controlador NOX.

#### A. OMNI (*Openflow MaNagement Infrastructure*)

O OMNI é uma ferramenta de código aberto que provê uma interface remota de administração para redes OpenFlow, possibilitando gerenciar facilmente essas redes [1]. A ferramenta é baseada em uma arquitetura orientada a serviços, permitindo o monitoramento e a configuração dinâmica dos encaminhamentos da rede. O OMNI dispõe de uma interface web que permite a configuração das rotas na rede. A arquitetura da ferramenta utiliza como base o controlador NOX modificado para seu propósito, onde foram desenvolvidos componentes que dispõem informações necessárias para a sua interface web.

As principais funcionalidades que a ferramenta dispõe incluem: visualizar as estatísticas da rede OpenFlow, adicionar novos fluxo, remover fluxos, migrar fluxos de um caminho para outro, monitorar os fluxos, tomar ações em um cenário com perda de pacotes e administrar uma rede OpenFlow.

#### B. RouteFlow

O RouteFlow é uma proposta de plataforma de roteamento remoto e centralizado [3]. Esta tecnologia visa o desacoplamento do plano de encaminhamento e o plano de controle, flexibilizando as redes IP quanto à facilidade de adição, remoção e especialização de protocolos e algoritmos. O RouteFlow utiliza máquinas virtuais (MVs) para representar a rede física através de uma topologia lógica de roteamento. Dessa forma, cada MV roda uma engine de roteamento padrão, onde suas interfaces representam as portas dos roteadores. Assim, a lógica de encaminhamento é executada na topologia virtual pelas engines de roteamento, separando o plano de controle do plano de dados. Após as decisões serem tomadas no plano virtual, estas são enviadas para o controlador que as instala nos respectivos elementos físicos da rede [9].

#### C. QuagFlow

O QuagFlow [10] é uma proposta de união transparente, não modificada, da suíte de roteamento de código aberto Quagga [10] com a rede OpenFlow. O QuagFlow explora a viabilidade de mover completamente a pilha de protocolos legados para controladores centralizados utilizando o protocolo OpenFlow [10]. Considera um passo intermediário rumo às redes programáveis para garantir a interoperabilidade com as redes legadas, possibilitando um benefício imediato em uma parceria entre o controle remoto e Quagga.

O QuagFlow replica a topologia física configurando as MVs em uma topologia virtual rodando o plano de

controle do Quagga. A arquitetura do QuagFlow é composta por um controlador QuagFlow (QF-C), desenvolvido como um componente do NOX, e uma série de serviços QuagFlow (QF-S) rodando transparente nas MVs [10], onde estas hospedam o Quagga.

Durante a pesquisa na literatura, foram encontradas ferramentas que possibilitam a administração da rede OpenFlow, como por exemplo o OMNI. Entretanto, não foi identificada, até o momento, nenhuma que possibilite ao administrador de redes fazer uma análise do comportamento e saturação dos recursos da rede OpenFlow.

### IV. OFNMS: OPENFLOW NETWORK MONITOR SYSTEM

A ferramenta desenvolvida neste trabalho tem como objetivo coletar periodicamente e de forma automática, os dados estatísticos dos *switches* em uma rede OpenFlow, persistindo esses dados em uma base de dados e possibilitando a consulta e geração de gráficos a partir desta. Dessa forma, será possível que o administrador da rede realize uma análise do comportamento dos recursos da mesma, como por exemplo, reconhecer o período em que os mesmos ficam sobrecarregados, ou estimar o crescimento futuro no uso da rede OpenFlow.

#### A. Arquitetura

A ferramenta desenvolvida foi dividida em três componentes principais, como ilustrado na Figura 2, sendo descritos a seguir.

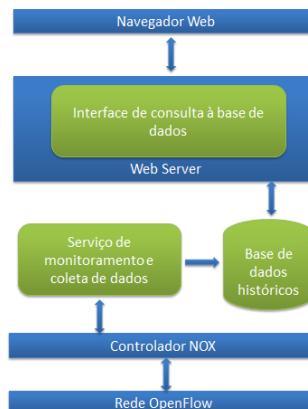


Figura 2 – Arquitetura da Ferramenta OPNMS.

#### 1) Base de dados históricos

Responsável por armazenar os dados coletados pelo serviço de monitoramento e dispor dessas informações para que a interface web possa consultá-los. Para armazenar os dados coletados, considerando a necessidade dos dados serem facilmente recuperados para a geração de gráficos, utilizou-se a ferramenta chamada RRDTool [7] [8]. A ferramenta apresentada neste trabalho se difere de ferramentas tradicionais de visualização de tráfego, baseadas em RRDTool, principalmente na forma de captura das informações, as quais são capturados pelo controlador OpenFlow.

#### 2) Serviço de monitoramento e coleta de dados

Responsável por coletar os dados estatísticos dos *switches* e persistir em banco de dados. O serviço de

monitoramento é iniciado juntamente com o NOX e é executado como um componente do mesmo.

Para a coleta dos dados, o serviço de monitoramento utiliza o componente *switchstats* do NOX, responsável por coletar os dados estatísticos dos *switches*.

A descoberta dos *switches* que compõem a topologia é feita de forma automática, o componente *switchstats* fornece essa informação através da propriedade “*dp\_stats*”, que contém uma coleção de estatísticas e informações gerais de cada *switch* da rede. Esta propriedade é consultada para obter a lista de *switches* e assim poder criar as bases de dados para armazenar suas informações, através do método “*create()*” da ferramenta RRDTool. Em seguida, os dados estatísticos das interfaces e dos fluxos nos *switches* são adquiridos, através das propriedades “*dp\_port\_stats*”, que contém uma coleção de estatísticas e informações das interfaces de cada *switch* da rede, e “*dp\_flow\_stats*”, que contém uma coleção de estatísticas e informações dos fluxos de cada *switch* da rede. Esses dados são armazenados na base de dados através da chamada do método “*update()*”, da ferramenta RRDTool, o qual ocorre em um intervalo de tempo pré-definido.

Os dados retornados dos *switches* estão disponíveis em informações sobre suas interfaces, também chamadas de portas, e informações sobre os fluxos configurados em cada *switch*.

As informações utilizadas para monitorar as interfaces foram:

- ***tx\_bandwidth***: contém a largura de banda de transmissão utilizada;
- ***rx\_bandwidth***: contém largura de banda de recebimento utilizada;
- ***tx\_dropped\_rate***: taxa de pacotes transmitidos que foram eliminados;
- ***rx\_dropped\_rate***: taxa de pacotes recebidos que foram eliminados;
- ***tx\_errors***: quantidade de pacotes com erro transmitidos;
- ***rx\_errors*** : quantidade de pacotes com erro recebidos.

Para monitorar os fluxos foram utilizadas:

- ***packet\_count***: quantidade de pacotes que combinaram com o fluxo e aplicaram sua regra;
- ***byte\_count***: quantidade de bytes que foram transmitidos por um fluxo.

### 3) Interface de consulta à base de dados

Interface utilizada pelo administrador da rede para consultar os dados históricos. Foi desenvolvido sobre plataforma web, podendo ser hospedado no próprio servidor onde roda o serviço de monitoramento e coleta de dados ou em um servidor separado.

A execução da interface web inicia quando seu endereço é requisitado ao serviço HTTP a partir de um navegador web, como apresentado na Figura 3. O serviço HTTP carrega o script correspondente, passando os parâmetros necessários. Esse, por sua vez, consulta a base de dados RRDTool fazendo a chamada ao método

“*graph()*”. O RRDTool gera o gráfico correspondente aos parâmetros que foram passados na chamada do método, retornando-o ao script que monta a página HTML a ser enviada ao navegador web pelo serviço HTTP.

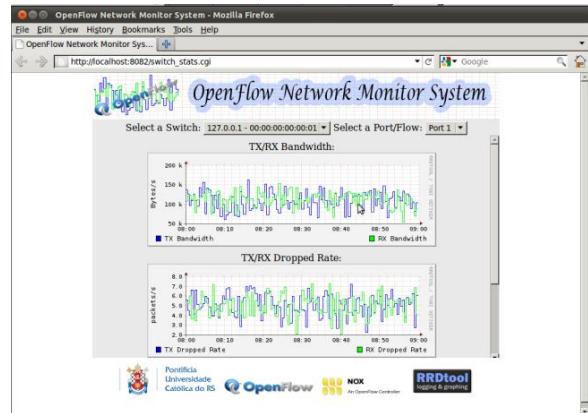


Figura 3 – Tela de Estatísticas dos Switches

### V. OBJETIVOS ALCANÇADOS

O cenário utilizado para executar o OPNMS, apresentado na Figura 4, foi composto por dois *switches* (s1 e s2), cada um com duas interfaces (eth0 e eth1), sendo que os dois *switches* estão interligados por uma das suas interfaces (s1-eth1 conectado a s2-eth1). Conectados a estes *switches* tem-se dois *hosts* (h3 e h4). O *host* h3 conectado ao *switch* s1 (h3-eth0 conectado a s1-eth0) e o *host* h4 conectado ao *switch* s2 (h4-eth0 conectado a s2-eth0). Por fim, tem-se um controlador c0 conectado aos *switches* através de um canal seguro.

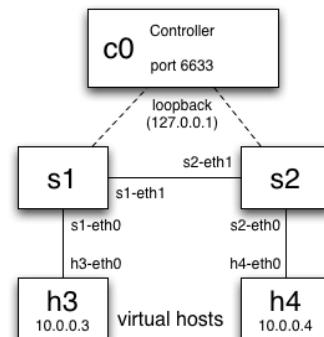


Figura 4 – Topologia utilizada para executar o OFNMS (imagem extraída de [4]).

Iniciando a ferramenta, a página inicial é carregada e disponibiliza um botão chamado “*Switches Stats*”, concedendo acesso à página de estatísticas dos *switches*, apresentada na Figura 3. Nesta tela, os *switches* são mostrados em uma lista suspensa (*ComboBox*) em “*Select a Switch*”. Ao selecionar um *switch*, outra lista suspensa é preenchida com as interfaces e fluxos do mesmo. Selecionando-se uma interface na lista suspensa “*Select a Port/Flow*”, são apresentados os gráficos com informações sobre “*TX/RX BandWidth*”, “*TX/RX Dropped Rate*” e “*TX/RX Errors*”.

O gráfico da Figura 5 mostra a taxa de Largura de Banda, sendo que durante uma hora de tráfego foram transmitidos entre 50 e 150 KiloBytes por segundo. A linha azul indica a taxa de dados transferidos e a linha verde indica a taxa de dados recebidos pela interface.

O gráfico da Figura 6 mostra a taxa de pacotes descartados, sendo que durante uma hora de tráfego foram descartados entre 2 e 7 pacotes por segundo, pacotes estes que foram enviados corrompidos propositalmente para simular uma situação de descartes de pacotes. A linha azul indica a taxa de pacotes transferidos e a linha verde indica a taxa de pacotes recebidos pela interface.

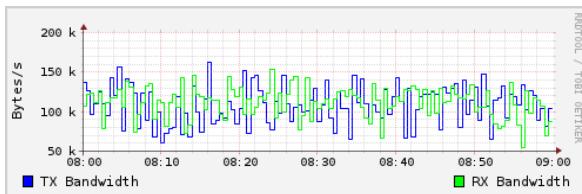


Figura 5 – TX/RX Port1 Bandwidth.

O gráfico da Figura 7 mostra a taxa de pacotes com erro, sendo que durante uma hora de tráfego foram contabilizados entre 2 e 7 pacotes por segundo. A linha azul indica a taxa de pacotes transferidos e a linha verde indica a taxa de pacotes recebidos pela interface.



Figura 6 - TX/RX Port1 Dropped Rate Packets.



Figura 7 - TX/RX Port1 Error Rate Packets.

Selecionando-se um fluxo na lista suspensa “Select a Port/Flow”, os gráficos com informações sobre “Bandwidth” e “Packets Rate” são carregados, apresentados na Figura 8 e Figura 9, respectivamente.

O gráfico da Figura 8 mostra a taxa da Largura de Banda utilizada pelos dados que foram encaminhados pelo fluxo, sendo que durante uma hora de tráfego foram contabilizados entre 60 e 150 KiloBytes por segundo.

O gráfico da Figura 9 mostra a taxa de pacotes encaminhados pelo fluxo, sendo que durante uma hora de tráfego foram contabilizados entre 800 a 1800 pacotes por segundo.

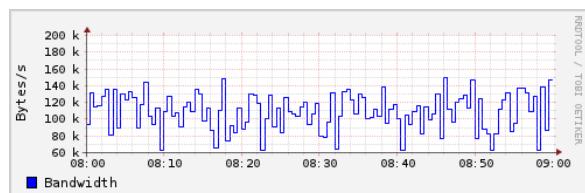


Figura 8 – Flow 1 Badnwidth.



Figura 9 – Flow 1 Packets Rate.

## VI. CONSIDERAÇÕES FINAIS

Este documento apresentou o desenvolvimento de uma ferramenta de monitoramento de *switches* OpenFlow. Tal ferramenta permite coletar dados estatísticos das interfaces e fluxos nos *switches* da rede, persiste esses dados em base de dados, possibilitando a sua visualização graficamente.

Como trabalhos futuros, pode-se desenvolver o monitoramento dos *hosts* e topologia da rede OpenFlow, além de incluir o período de coleta dedados.

## REFERÊNCIAS

- [1] GTA/UFRJ. “OMNI Openflow Management Infrastructure”, Disponível em: <<http://www.gta.ufrj.br/omni/>>. Acesso em: 25 mai. 2012.
- [2] McKeown, Nick. Anderson, Tom. Balakrishnan, Hari. Parulkar, Guru. Peterson, Larry. Rexford, Jennifer. Shenker, Scott. Turner, Jonathan. “OpenFlow: Enabling Innovation in Campus Networks”, ACM SIGCOMM’10, New York, NY, USA, Outubro 2010.
- [3] OpenFlow Consortium. “OpenFlow Switch Specification Version 1.1”, Disponível em <http://www.openflow.org/documents/openflow-specv1.1.0.pdf>. Último Acesso em: Set 2011.
- [4] OpenFlow Website. Disponível em <<http://www.openflow.org/>>. Acesso em: Set 2011.
- [5] NOX Website. Disponível em: <<http://noxrepo.org>>. Acesso em: 05 set 2011.
- [6] Gude, Natasha. Koponen, Teemu. Pettit, Justin. Pfaff, Ben. Casado, Martín. McKeown, Nick. Shenker, Scott. “NOX: Towards an Operating System for Networks”, ACM SIGCOMM’08, New York, NY, USA, Julho 2008.
- [7] RRDTool Website. Disponível em: <<http://oss.oetiker.ch/rrdtool/index.en.html>>. Acesso em: 05 jun 2012.
- [8] RRDTool Documentation. Disponível em <[http://www.luteus.biz/Download/LoriotPro\\_Doc/RRD%20documentation/Introduction\\_RRD\\_EN.htm](http://www.luteus.biz/Download/LoriotPro_Doc/RRD%20documentation/Introduction_RRD_EN.htm)>. Acesso em 25 mai 2012.
- [9] Nascimento, Marcelo R. Rothenberg, Christian E.; Denicol, Rodrigo R., Salvador, Marcos R., Magalhães, Maurício F.. “RouteFlow: Roteamento Commodity Sobre Redes Programáveis” XXIX Simpósio Brasileiro de Redes de Computadores. SBRC 2011, Campo Grande, MS, Brasil, Maio 2011.
- [10] Nascimento, Marcelo R.; Rothenberg, Christian E.; Salvador, Marcos R.; Magalhães, Maurício F.. “QuagFlow: Partnering Quagga with OpenFlow”, ACM SIGCOMM’10, New Delhi, India, Agosto 2010.

# Análise de Desempenho do Protocolo M-DART

Bruno Nieto

Centro Universitário La Salle  
Canoas, Brasil  
[brunoguento@gmail.com](mailto:brunoguento@gmail.com)

Andrea Krob

Centro Universitário La Salle  
Canoas, Brasil  
[andrea.krob@unilasalle.edu.br](mailto:andrea.krob@unilasalle.edu.br)

**Resumo** - Os protocolos de roteamento vêm sendo alvo de pesquisas há muitos anos, principalmente no âmbito das redes *Ad Hoc*, onde os ambientes descentralizados tornam o desafio das redes sem fio ainda maior. Este artigo tem o objetivo de avaliar o desempenho do protocolo de roteamento M-DART através de diferentes *flavors* TCP e identificar o melhor cenário em relação ao desempenho da rede.

**Palavras-Chave:** Flavors TCP, Protocolo M-DART, Protocolos de Roteamento, Redes Ad Hoc, Redes sem fio.

## I. INTRODUÇÃO

Para o bom funcionamento das redes *ad hoc* é indispensável a escolha de um protocolo de roteamento adequado e que o mesmo possua um bom desempenho. Os protocolos de roteamento possuem um papel muito importante, pois eles permitem que cada dispositivo saiba para qual nó ele deve enviar seus pacotes.

O objetivo geral deste artigo consiste em estudar o protocolo M-DART (*Multi-Path Dynamic Address Routing*) [5], avaliando o seu desempenho por meio de simulações com diversos protocolos da camada de transporte. Para que o desempenho do protocolo M-DART seja analisado serão usadas três métricas para avaliação do mesmo: perdas, vazão e atraso. O foco da comparação será abordar quatro tipos de *flavors* do protocolo de transporte TCP, buscando resultados que satisfaçam o foco de cada aplicação, algumas tendo como prioridade a vazão e outras a confiabilidade na entrega, ou seja, menos perdas. Com isso, serão apresentados os resultados obtidos com o intuito de contribuir cientificamente para os estudos futuros e como alternativa de melhoria para as redes *ad hoc*.

Com o desenvolvimento deste projeto será possível conhecer alternativas que melhor se adaptem as redes *ad hoc*, especificamente com relação ao protocolo de roteamento M-DART.

## II. REDES AD HOC

Em muitos casos torna-se difícil ou até mesmo inviável a instalação de redes cabeadas ou redes sem fio infraestruturadas, o que fez com que as redes *ad hoc* fossem cada vez mais exploradas. As redes *ad hoc* [6] são redes que não necessitam de infraestruturas para efetivar o enlace de comunicação, sendo assim, este tipo de rede permite que os dispositivos móveis possam trocar informações entre si ou através de roteamento entre os mesmos sem a necessidade de um ponto de acesso.

Sendo assim, os dispositivos tornam-se capazes de

prover o serviço de roteador e de terminal ou prover os dois serviços concomitantes.

O papel dos protocolos de roteamento é fazer a descoberta e o mapeamento da topologia da rede [1], utilizando tabelas de roteamento como forma de armazenamento das informações coletadas. Com isso eles são capazes de enviar requisições a qualquer dispositivo que estiver dentro da área de cobertura da conexão.

Os protocolos de roteamento para redes *ad hoc* possuem duas grandes categorias: *Unicast* e *Multicast* [2].

Os protocolos *unicast* podem transmitir pacotes de um nó de origem a um único nó de destino, através de nós intermediários. Estes protocolos podem ser classificados conforme a construção de suas rotas nas seguintes modalidades: Reativos, Pró-Ativos e Híbridos. Já os protocolos *multicast* são capazes de transmitir pacotes de um determinado nó de origem para múltiplos nós de destino [3] [4], identificados por um único endereço. Eles são divididos em dois tipos: *Tree-based* e *Mesh-based*.

## III. PROTOCOLO M-DART

O M-DART [5], é um protocolo *unicast* que pertence ao grupo dos pró-ativos. Este protocolo baseia-se em tabelas *hash* distribuídas (*Distributed Hash Table - DHTs*) que consiste em utilizar chaves para o mapeamento, localização e remoção de nós em uma rede. O funcionamento destas tabelas consiste no mapeamento de nós conectados a uma rede através de um código que identifica cada nó.

Com esse código, cada nó tem a capacidade de localizar e identificar seus vizinhos, fazendo com que seja desnecessário o conhecimento de todos os nós da rede. Para o bom entendimento deste protocolo é necessário conhecer um pouco melhor o seu antecessor, o protocolo DART (*Dynamic Address Routing*) [7].

O protocolo DART utiliza um vetor de distância para o descobrimento de rotas, assim como o protocolo AODV. Através de endereçamento dinâmico, é capaz de realizar o encaminhamento hierárquico de pacotes na rede de maneira viável, reduzindo significativamente as informações mantidas por cada nó.

Uma vez que o processo de encaminhamento tem como base os endereços da rede, os nós são distribuídos de maneira eficiente. Os endereços de rede são sequências de bits que podem ser estruturados em forma de árvore, sendo que cada folha da árvore está associada a um destes endereços, contendo internamente um prefixo de endereço binário que representa a

sua localização dentro de um conjunto de folhas, conforme demonstrado na figura 1.

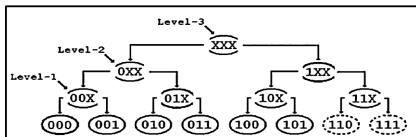


Figura 1: Endereços de rede em formato de árvore

Porém, o maior problema deste protocolo é a possibilidade de haver rotas falsas, conforme mostrado na figura 2, pois se algum dos nós por algum motivo falhar haverá uma quebra na hierarquia da árvore, fazendo com que todos os nós ligados a este com problema tenham de interromper a sua comunicação até que seja concluído o processo de redescoberta de rotas.

Com isso, foi constatado que apenas se baseando no custo das rotas não seria o suficiente para descobrir o melhor caminho.

A principal diferença entre os protocolos está no fato de que o M-DART possui a funcionalidade de obter diversas rotas entre o nó de origem e destino utilizando DHT, tornando o mapeamento e o roteamento mais ágil, assim melhorando o desempenho da rede devido a topologia ser dinâmica.

Uma das grandes vantagens e avanço conquistado com o desenvolvimento do protocolo M-DART é a capacidade de resolver o problema das rotas falsas. Como ele possui diversas rotas disponíveis em sua tabela de roteamento, em caso de falha de um dos nós o protocolo tem a disposição outras rotas pré-calculadas, fazendo com que seja evitada a interrupção de comunicação, até que ao menos exista um caminho disponível, conforme mostrado na figura 3.

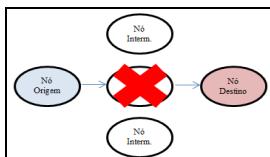


Figura 2: Rotas Falsas

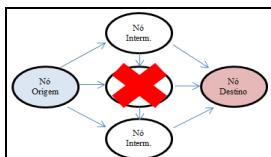


Figura 3: Solução das Rotas Falsas

Para o bom entendimento do funcionamento do protocolo M-DART é necessário conhecer dois conceitos bem importantes sobre endereçamento: o endereço de roteamento e o identificador de nó. O endereço de roteamento de um nó é dinâmico, sendo que o mesmo é alterado conforme o nó se movimenta na rede. O identificador do nó é um número estático, ou seja, permanece o mesmo durante o tempo de vida do nó na rede.

Quando um novo nó começa a fazer parte da rede o mesmo começa a escutar as atualizações periódicas de roteamento dos seus vizinhos, de maneira a identificar um endereço desocupado. Encontrando um endereço vago o nó registra seu identificador e o endereço de roteamento obtido. Devido a mobilidade, o endereço de roteamento pode ser posteriormente alterado, sendo que a tabela de roteamento precisa ser atualizada.

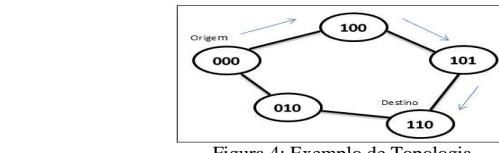


Figura 4: Exemplo de Topologia

Como observado na figura 3, o custo não é um parâmetro suficiente para definir a escolha de um melhor caminho. Sendo assim, o M-DART, levando em conta o recurso hierárquico de endereçamento dinâmico, define como próximo salto o nó que contém o prefixo de endereço mais próximo do destino.

No caso de haver diversos nós com prefixo igual ou próximo do destino, será levado em consideração o de menor custo de rota. Um exemplo seria o demostrado na figura 4, onde o nó com endereço "000" procura o melhor caminho para "110". Como o protocolo M-DART não tem como prioridade o custo da rota e sim o prefixo de endereço, a rota passará pelos nós "100" e "101", pois compartilham o mesmo prefixo "1XX", correspondente ao prefixo do nó destino. Por este motivo principalmente é que o M-DART obteve grande avanço em relação ao protocolo DART.

#### IV. PROTOCOLOS DE TRANSPORTE

Os protocolos de transporte possuem um papel muito importante, tendo como principais metas lidar com controle de erros, com a definição de sequências e com o controle de fluxo [6]. Um dos objetivos deste artigo é analisar o protocolo de transporte TCP e alguns de seus *flavors*, buscando o que possui o melhor desempenho em redes *ad hoc*, identificando características e possíveis melhorias nestes cenários.

Com o intuito de buscar melhorias pesquisadores desenvolveram algumas variações no protocolo TCP [8], abordando questões desde como o protocolo deve tratar uma falha de transmissão ou um congestionamento do canal de comunicação.

Essas variações, conhecidas como *flavors*, buscam corrigir problemas existentes em outras versões. O Tahoe é um exemplo, sendo o primeiro *flavor* a desenvolver o mecanismo de controle de congestionamento [8], assim evitando que sejam enviados mais pacotes para a rede do que a mesma pode suportar [9].

O TCP New Reno foca principalmente a melhoria de múltiplas perdas de pacote que acontecem em uma única janela de transmissão.

Diferentemente das outras variações que utilizam perdas de pacotes para detectar o congestionamento da rede, o TCP Vegas analisa o tráfego nos roteadores [8], entre a origem e o destino, antes de ocorrer alguma perda. Com isso, ele tenta evitar o congestionamento, observando a redução da taxa de envio em relação a taxa esperada.

O TCP Sack (*Selective Acknowledgment*) aumenta o desempenho do TCP em redes de alta velocidade e com maior atraso através do reconhecimento seletivo, fazendo com que o receptor possa informar ao transmissor sobre todos os segmentos que chegarem corretamente, de modo que o transmissor saiba quais os pacotes que necessitam ser retransmitidos devido a perda.

## V. METODOLOGIA

Para que o objetivo deste trabalho fosse atingido optou-se pela utilização da ferramenta de simulação NS-2 (Network Simulator 2) [12] e consequentemente o sistema operacional Linux, devido a alta compatibilidade com o software.

O aspecto mais forte na escolha do NS-2 foi que a ferramenta possui o protocolo de roteamento em pesquisa, o M-DART em sua versão 2.35, sendo que este trabalho visa analisar o comportamento deste protocolo de roteamento. Para isso, será usado o protocolo de transporte TCP e os *flavors*: Tahoe, New Reno, Vegas e Sack.

Para que fossem definidos os *flavors* para este estudo foram analisados alguns critérios: o Tahoe foi escolhido por ser o principal *flavor* utilizado nas transmissões simuladas pelo NS-2. Já o New Reno e o Vegas por apresentarem melhores resultados em outros trabalhos em relação aos demais protocolos [9] [10]. O TCP Sack, comparado com os mesmos *flavors* deste trabalho, só que voltado para redes de satélite obteve melhores resultado sobre os demais [11]. Outro fator relevante foi a questão de não terem sido encontrados, na pesquisa bibliográfica realizada para este trabalho, estudos em relação ao protocolo M-DART, por ser um protocolo ainda recente e pouco estudo em relação as redes *ad hoc*.

Um dos fatores mais importantes dentro de uma simulação é a quantidade de nós utilizados, pois fatores como vazão e perda de pacotes são diretamente afetados. As simulações deste projeto utilizaram três variações na quantidade de nós: 10, 50 e 100 nós em movimento, configurados no script TCL, tanto o caminho de origem ao destino como o tráfego dos nós.

Foi determinado apenas um fluxo de dados fim-a-fim durante as simulações, pois o intuito deste projeto não é estudar engenharia de tráfego, sobrecarregando os nós, mas sim analisar o comportamento do roteamento dinâmico durante a movimentação dos nós na rede, conforme pode ser observado na figura 5, onde é demonstrado o cenário de 10 nós móveis (NM).

Neste projeto não há um intervalo de confiança, pois nas simulações utilizando a ferramenta NS-2 os resultados sempre se mostraram os mesmos, ao menos que haja alguma mudança de parâmetro. Isto se torna um ponto negativo da ferramenta, pois a simulação acaba não tendo uma maior proximidade da realidade.

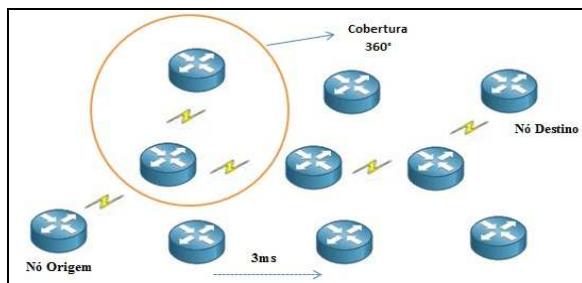


Figura 5: Topologia com 10 NM

O deslocamento dos nós é realizado a uma velocidade de 3 m/s e a cobertura do sinal de cada nó é de 360°. Para a avaliação do protocolo M-DART foram estabelecidas as

seguintes métricas: *delay* (atraso), *throughput* (vazão) e taxa de perdas.

## VI. RESULTADOS

Em relação ao percentual de perda de pacotes, conforme pode ser observado na figura 6, o *flavor* que obteve melhor resultado foi o TCP Vegas em todas as simulações, pois sua característica forte é a de ser pró-ativo em relação ao tráfego da rede, ou seja, o mesmo examina o tráfego nos roteadores, evitando o congestionamento na rede. Com isso, o mesmo reduz o número de pacotes enviados, evitando também a perda de pacotes.

O *flavor* que obteve o pior resultado com relação às perdas no protocolo M-DART foi o New Reno. Isto se justifica devido a sua deficiência nos múltiplos descartes. Este problema, como demonstrado na simulação, fez com que houvesse um grande número de retransmissões de segmento a cada RTT, provocando atrasos no envio dos segmentos seguintes.

O TCP Tahoe, diante do cenário de maior número de nós, apresentou um resultado bastante satisfatório. Isto ocorreu porque o controle de congestionamento deste *flavor* é menos rígido, permitindo que o mesmo envie mais pacotes para a rede. Ao contrário, o TCP Sack obteve melhores resultados com um número menor de nós, pois o mesmo possui um controle muito rigoroso em relação a retransmissão de pacotes que não chegaram corretamente ao destino.

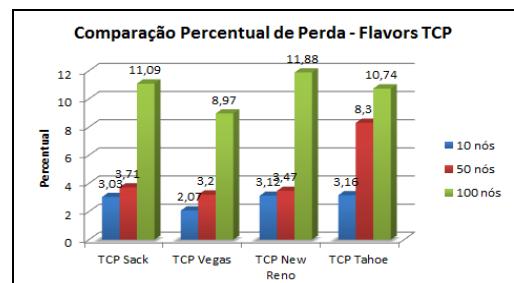


Figura 6: Comparação % de Perdas

Em relação ao *throughput*, conforme pode ser observado na figura 7, o TCP Vegas, em todas as simulações, foi inferior aos demais *flavors*, pois em análise conjunta com a métrica de pacotes perdidos, observa-se que segundo sua característica o mesmo evita o congestionamento da rede reduzindo a vazão de dados.

O TCP Sack obteve excelentes resultados com os cenários de 10 e 50 nós, mas com o aumento para 100 acabou sendo o segundo pior na análise de resultados. Isso ocorreu porque o *flavor*, devido ao alto índice de perdas se preocupou em reenviar os pacotes perdidos segundo suas características, reduzindo significativamente sua vazão.

O TCP New Reno foi o segundo melhor *flavor* em relação a esta métrica, pois assim como o Tahoe, não possui um controle muito rigoroso em relação ao congestionamento da rede, enviando muitos pacotes, não se preocupando tanto com a questão da perda.

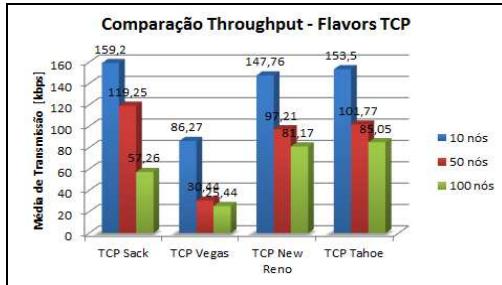


Figura 7: Comparação Throughput

Em relação ao *delay*, conforme pode ser observado na figura 8, o TCP New Reno obteve a média mais alta com o aumento dos nós. Levando em consideração que os *flavors* com maior média de *throughput* foram o Tahoe e o New Reno, observa-se que o Tahoe, por possuir um controle de congestionamento menos rígido, fez com que os pacotes fossem enviados com mais intensidade. O New Reno, com o aumento dos nós, enviou uma taxa alta de pacotes, sendo que o problema de retransmissão de muitos segmentos provocou atrasos no envio dos próximos.

O TCP Sack reduziu significativamente sua média no cenário com maior número de nós, pois com a queda na média de *throughput* na alteração dos NM na rede, fez com que menos pacotes fossem enviados, gerando menos pacotes perdidos e menor atraso na rede.

O TCP Vegas obteve o melhor resultado em todas as simulações, pois conforme sua característica o mesmo evita congestionamentos, evitando a perda de pacotes.

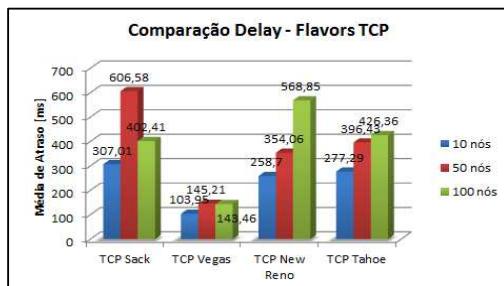


Figura 8: Comparação Delay

## VII. CONSIDERAÇÕES FINAIS

Após este estudo é possível concluir que há dois resultados satisfatórios, sendo que os mesmos deverão ser levados em consideração de acordo com a aplicação desejada, o grau de importância da mesma e o tamanho da rede *ad hoc* que se deseja construir. Conclui-se com isto que o Tahoe obteve melhor desempenho voltado a clientes e aplicações que não levam em consideração como prioridade o atraso na entrega de alguns pacotes, pois este *flavor*, como observado no decorrer do trabalho e nas simulações, possui um controle não muito rigoroso quanto à prevenção ao congestionamento da rede, fazendo com que a vazão deste *flavor* atinja altos índices de pacotes transmitidos.

Aplicações voltadas para fins civis, como redes de táxi, fazendo com que haja comunicação entre os veículos, e salas

de reuniões são exemplos que não haveria necessidade levar tanto em consideração o problema do atraso na entrega dos pacotes, pois são aplicações que não são extremamente críticas. É importante ressaltar que o Tahoe somente é a melhor alternativa se o número de nós for de aproximadamente 100NM, pois conforme mostrado nas simulações, em cenários com números menores de nós a diferença em relação ao Vegas em relação ao percentual de perda se torna bem significativa.

Outro resultado considerado satisfatório neste trabalho é o desempenho do TCP Vegas, voltado a clientes e aplicações que possuem como preocupação obter uma garantia maior na entrega dos pacotes; a final, todas as versões TCP não precisam garantir a entrega dos pacotes, mas sim os *bytes* transmitidos. Exemplos de aplicações seriam para fins militares e operações de emergência. O TCP Vegas é um *flavor* diferentemente de todos os outros, pois se preocupa primeiramente em analisar o tráfego nos roteadores para evitar possíveis congestionamentos, fazendo com que seja reduzido o número de pacotes enviados para a rede.

Este tipo de *flavor* é recomendado, segundo as simulações realizadas, para redes menores que não ultrapassem a quantidade de 50 NM, pois a partir desta quantidade o Vegas começa aumentar a perda de pacotes significativamente.

## VIII. REFERÊNCIAS

- [1] M.D.Ortiz, Incentivando a cooperação em redes *Ad Hoc*, [http://www.teses.ufc.br/tde\\_busca/arquivo.php?cod\\_Arquivo=1365](http://www.teses.ufc.br/tde_busca/arquivo.php?cod_Arquivo=1365) – Acessado em set/2011.
- [2] V.M.Assis et al, Redes Sem Fio em Malha, [http://www.gta.ufrj.br/grad/10\\_1/malha/](http://www.gta.ufrj.br/grad/10_1/malha/) – Acessado em out/2011.
- [3] A.C.Fontoura, Análise de Protocolos de Roteamento em uma Rede Mesh baseada em um Backbone Universitário, [http://www.upf.br/computacao/images/stories/TCs/arquivos\\_20072/antonio\\_fontoura.pdf](http://www.upf.br/computacao/images/stories/TCs/arquivos_20072/antonio_fontoura.pdf) – Acessado em fev/2012.
- [4] C.J.A.Vicentini, Uma nova Métrica de Roteamento para Redes Wireless Mesh com Tráfego Voip, [http://www.ppgia.pucpr.br/lib/exe/fetch.php?me\\_dissertacoes:cleverton\\_juliano.pdf](http://www.ppgia.pucpr.br/lib/exe/fetch.php?me_dissertacoes:cleverton_juliano.pdf) – Acessado em fev/2012.
- [5] M.Caleffi, L.Paura, M-DART: multi-path dynamic address routing, <http://dl.acm.org/citation.cfm?id=1967234> – Acessado em jan/2012.
- [6] A. S. Tanenbaum, Redes de computadores. 5th ed. São Paulo, SP: Pearson Prentice Hall, 2011.
- [7] J. Eriksson, M.Faloutsos, S. Krishnamurthy, DART: Dynamic Address Routing for Scalable *Ad Hoc* and Mesh Networks, [http://nms.csail.mit.edu/~jakob/pubs/dart\\_ton\\_2006.pdf](http://nms.csail.mit.edu/~jakob/pubs/dart_ton_2006.pdf) – Acessado em fev/2012.
- [8] P.R.N.Martins, Análise do Desempenho de diferentes versões do Protocolo TCP: Um experimento através de simulação, [http://tc.online.feevale.br/tc/files/0002\\_1722.pdf](http://tc.online.feevale.br/tc/files/0002_1722.pdf) – Acessado em mar/2012.
- [9] A.J.O.Abdí, Comparative Study on the Performance of different TCP flavors, [http://ctd.uum.edu.my/1618/1/Abdulaziz\\_Jama\\_Omar\\_Abdi.pdf](http://ctd.uum.edu.my/1618/1/Abdulaziz_Jama_Omar_Abdi.pdf) – Acessado em mar/2012.
- [10] L.E.P.Bueno, Estudo do Desempenho do Protocolo TCP em Redes Sem Fio, <http://www.eletrica.ufpr.br/ufpr2/tccs/26.pdf> – Acessado em mar/2012.
- [11] H.Zattar, Avaliação Comparativa dos Mecanismos Congestion Avoidance TCP Sack e Vegas operando sobre uma Rede ATM via Satélite, [sites.uol.com.br/hzattar/ArtigoSBRTcolunadupla.zip](http://sites.uol.com.br/hzattar/ArtigoSBRTcolunadupla.zip) – Acessado em mar/2012.
- [12] R. G. Araujo. A ferramenta de Simulação NS (Network Simulator). Universidade Salvador UNIFACS, Salvador, 2003. Disponível em: <[http://www.harpia.eng.br/pesquisa/tfc\\_EngElet\\_Rafael.pdf](http://www.harpia.eng.br/pesquisa/tfc_EngElet_Rafael.pdf)>. Acesso em: 26out. 2011.

# Monitoramento e Análise do Impacto no Desempenho em Ambientes Virtualizados

Pedro Freire Popolek, Odorico Machado Mendizabal

Centro de Ciências Computacionais

Universidade Federal do Rio Grande – FURG

Campus Carreiros: Av. Itália km 8 Bairro Carreiros, Rio Grande, Brasil

{p.f.popolek,odoricomendizabal}@furg.br

**Resumo**—Embora não seja uma prática inovadora, a utilização de infraestruturas virtualizadas é amplamente empregada em computação em nuvem. Porém, o impacto no desempenho ocasionado por tais infraestruturas torna-se um dos maiores desafios para que rígidos acordos de níveis de serviço (SLA) sejam respeitados. Este trabalho apresenta técnicas de monitoramento e realiza a análise de desempenho em um ambiente virtualizado. Com o uso de cargas sintéticas sobre a infraestrutura monitorada, é possível identificar as principais causas da degradação de desempenho observada.

## I. INTRODUÇÃO

Sistemas computacionais modernos fazem uso de infraestruturas computacionais amplamente distribuídas, possibilitando a implementação de aplicações colaborativas e o compartilhamento de recursos remotos. Tais sistemas têm como principais objetivos melhorar o desempenho, escalabilidade e disponibilidade dos serviços oferecidos, além de proporcionar uma melhor utilização de recursos.

Nesse contexto, computação em nuvem, associada ao uso de virtualização, permite que um conjunto de servidores físicos hospede dezenas ou mesmo centenas de máquinas virtuais. Dessa forma, a escalabilidade de sistemas é ampliada, garantindo a maximização na utilização de recursos. Um grande desafio associado aos serviços oferecidos por esse tipo de infraestrutura é garantir que as máquinas hospedeiras não sejam sobrecarregadas. Uma vez que a carga de trabalho aplicada em uma máquina física excede seu poder computacional, todas as máquinas virtuais executando nesse hardware sofrerão degradação de desempenho, impactando as aplicações em uso.

Este trabalho descreve os primeiros passos no desenvolvimento de mecanismos de monitoramento e detecção proativa de gargalos de desempenho para computação em nuvem. Ao implementar tais mecanismos, será possível prever instabilidade e degradação em máquinas virtuais com razoável antecedência, permitindo a adoção de políticas eficientes para gerenciamento dinâmico de recursos.

Como resultados preliminares desta pesquisa, é apresentado um estudo sobre monitoramento de recursos em sistemas operacionais Windows e Linux. Sem uma análise exaustiva, este artigo identifica um conjunto mínimo de contadores de desempenho a serem observados e, a realização de experimentos indica como as métricas coletadas auxiliam na previsão de degradação de desempenho.

Este artigo está organizado da seguinte maneira. A próxima seção introduz o conceito de máquinas virtuais. A Seção III apresenta ferramentas de monitoramento para plataformas Windows e Linux. Os experimentos realizados são descritos na Seção IV e as conclusões deste trabalho aparecem na Seção V.

## II. MÁQUINAS VIRTUAIS

O uso massivo de máquinas virtuais (VM - *Virtual Machines*) é uma das principais características das infraestruturas de computação em nuvem. A técnica de virtualização permite o particionamento de recursos de servidores físicos entre vários servidores virtuais que executam concomitantemente em uma mesma máquina física. Dentre as vantagens dessa abordagem estão o aumento de escalabilidade da própria infraestrutura e a otimização no uso dos recursos computacionais disponíveis.

Em ambientes virtualizados, além do SO (Sistema Operacional) hospedeiro, que executa sobre o hardware físico, instâncias de VMs e SOs hóspedes, que executam sobre hardwares virtuais, também existe um monitor (VMM - *Virtual Machine Monitor*). O VMM, também conhecido por *hypervisor*, é responsável pelo gerenciamento e controle dos recursos compartilhados pela máquina física (por exemplo, CPU, memória e dispositivos de E/S). Dentre outras tarefas, o VMM é responsável pelo escalonamento e o tratamento de instruções das VMs, fornecendo transparência aos SOs visitantes sobre o hardware onde esses executam [1]. A Figura 1 representa uma possível disposição dos elementos descritos.

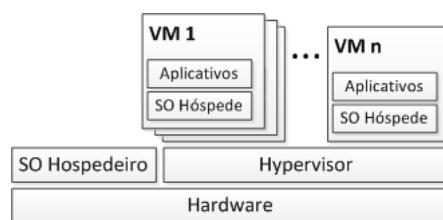


Figura 1. Virtualização híbrida.

No entanto, é comum observar sobrecarga na utilização de recursos em sistemas virtualizados. Supondo uma mesma arquitetura configurada com máquinas virtuais ou apenas com uma máquina física, as VMs não

mantém o mesmo desempenho que a máquina física ao tratar uma carga de trabalho semelhante. O impacto desse comportamento varia de acordo com o tipo de carga de trabalho demandada pelas aplicações. Em suma, esse custo é ocasionado pelo acréscimo de camadas de software que realizam emulações, virtualização de dispositivos, e ao compartilhamento competitivo de recursos entre VMs [2].

Outro aspecto importante no desempenho desses ambientes é o escalonamento feito pelo VMM. Tradicionalmente o escalonamento prioriza a partilha de recursos do processador entre as VMs existentes, deixando em segundo plano o agendamento de operações de E/S. Por esse motivo aplicações *I/O-bound* apresentam perda de desempenho, ou seja, o atraso na execução das operações torna-se um gargalo [3].

Algumas empresas que oferecem serviços em nuvem computacional possuem *hypervisors* próprios, visando melhor aproveitamento do tipo específico de suas arquiteturas computacionais. A empresa Amazon utiliza o Xen Hypervisor com alterações personalizadas no código fonte. O Windows Azure utiliza o Wazure Hypervisor, solução não comercial, baseada em uma solução fornecida pela própria empresa, o Hyper-V.

Com este trabalho e futuros estudos, pretende-se criar um mecanismo de detecção preventiva de degradação de desempenho em ambientes virtualizados. Porém, a atual pesquisa preocupa-se somente em entender e avaliar aspectos de desempenho desses ambientes decorrentes do escalonamento realizado pelo VMM.

### III. TÉCNICAS DE MONITORAMENTO

Esta seção descreve como um sistema operacional permite observar um histórico sobre o seu desempenho. A obtenção desse histórico dá-se através de ferramentas de monitoramento, que coletam dados representando níveis de utilização de recursos de hardware e sistema operacional. A análise desses dados de monitoramento, coletados em intervalos regulares de tempo, permite identificar características do sistema, detectar gargalos de desempenho ou ainda correlacionar dados na busca de causa para um comportamento observado.

Monitorar recursos do sistema para detecção de perfis de carga requer conhecimento sobre as métricas de monitoramento disponíveis. Embora as métricas variem de acordo com o sistema operacional utilizado, é possível estabelecer um conjunto de métricas comuns a qualquer sistema. As principais métricas identificadas são de memória, disco, rede e processador. A HP apresenta um estudo detalhado sobre métricas de monitoramento para o sistema operacional Windows [4]. Cada métrica possui um conjunto de contadores e cada contador fornece um valor numérico que representa uma informação específica de utilização do recurso (por exemplo, Interrupções/s é um contador do processador).

Sistemas operacionais Windows possuem uma ferramenta nativa, chamada *Performance Monitor*, responsável pelo monitoramento de recursos do sistema. As métricas de monitoramento são representadas por objetos e cada

objeto disponível pode possuir uma ou mais instâncias. As instâncias servem para diferenciar elementos não unitários do sistema.

Em sistemas operacionais Linux, é possível extrair dados para monitoramento de recursos do sistema através de arquivos virtuais existentes no diretório /proc/. Porém, um conjunto de ferramentas presentes no pacote sysstat automatizam essa tarefa, sendo as principais: Sar, Iostat e Pidstat.

A Tabela I contém alguns exemplos de contadores de desempenho presentes tanto em Windows quanto em Linux. Embora alguns contadores sejam específicos de um determinado sistema operacional, é possível, em alguns casos, correlacionar métricas observadas em outros contadores para a obtenção uma métrica equivalente.

Tabela I  
EXEMPLOS DE CONTADORES DE DESEMPENHOS

Windows	Linux	Descrição
Tamanho da Fila de CPU	runq-sz	Quantidade de processos em estado pronto e enfileirados.
% tempo de interrupção	% irq + % soft	Média percentual de uso de CPU atendendo interrupções.
Transferências de disco/s	r/s + w/s	Requisições de disco completadas por segundo.

### IV. EXPERIMENTOS

Serão apresentados nesta seção experimentos que visam expor características de gerenciamento de recursos realizado por *hypervisor*, com enfoque no desempenho de sistemas virtualizados. Foram utilizadas cargas de trabalho sintéticas aplicadas em máquinas virtuais, reproduzindo o comportamento de processos *CPU-bound* e *I/O-bound*.

Para realização dos experimentos, foi utilizada uma máquina hospedeira com processador AMD FX-6100 (seis núcleos), 8GB de memória RAM, 500GB de disco rígido, sistema operacional Ubuntu 12.04 64 bit e como hypervisor o KVM. As máquinas virtuais hospedadas foram configuradas cada uma com processador QEMU 64, 1GB de memória RAM, 20 GB de disco virtual pré-alocado e sistema operacional Windows XP Professional SP3 32 bit.

Foram criados seis cenários diferenciando o número de VMs ativas, sendo seis a quantidade máxima. Ao total foram realizados doze experimentos, seis para cada tipo de carga de trabalho nos diferentes cenários existentes. Durante o ensaio, todas VMs em execução recebem o mesmo tipo e volume de carga de trabalho.

Para execução de processo *CPU-bound* foi utilizado o aplicativo *WPrime Benchmark v2.09* [5]. A opção de processamento “1024M” foi selecionada nos testes, ou seja, é calculada a raiz quadrada dos primeiros 1024 milhões números inteiros. Como ponto de partida para análise da utilização de recursos por ambientes virtualizados, observe os tempos de execução desse *benchmark* na Tabela II.

A partir de 3 VMs executando concorrentemente, conforme aumenta o número de VMs em execução, mais tempo leva para a aplicação finalizar. Dentre os fatores desse atraso na execução das tarefas estão o escalonamento

Tabela II  
DURAÇÃO MÉDIA DE EXPERIMENTOS *CPU-bound*

VMs em Execução	Média do Tempo de Processamento
1	34.98 min
2	34.04 min
3	34.82 min
4	39.70 min
5	43.58 min
6	48.54 min

entre VMs e a competição por recursos da máquina hospedeira. Com base nos monitores de desempenho do sistema observa-se que com um número elevado de VMs o percentual de tempo destinado a cada VM reduz, enquanto o tempo de execução do hospedeiro em modo privilegiado aumenta. A Figura 2 exibe as métricas coletadas para porcentagem de uso de CPU para VMs (modo convidado), porcentagem de uso de CPU no modo privilegiado e porcentagem de CPU ociosa.

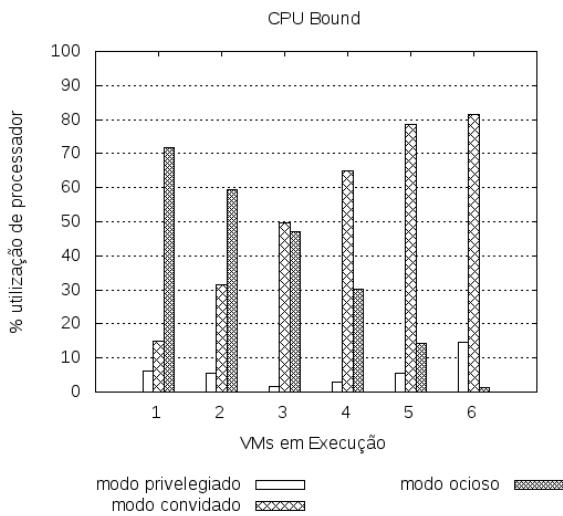


Figura 2. Utilização de CPU da máquina hospedeira durante experimentos *CPU-bound*

Observe que a carga de utilização da máquina hospedeira cresce linearmente (melhor visualizado pelo decrescimento linear de % CPU ociosa). No entanto, com o gerenciamento mais custoso do VMM, o aproveitamento de CPU por parte das VMs não cresce linearmente (veja o crescimento no % de CPU executando no modo convidado).

Para processos predominantemente *I/O-bound*, foi utilizado o benchmark *CrystalDiskMark 3.0.1* [6]. Como opção de carga de trabalho foi escolhida 2000MB de leitura/escrita sequencial. A Tabela III apresenta as taxas de leitura/escrita de cada experimento<sup>1</sup>.

Análogo ao experimento anterior, percebe-se que com

<sup>1</sup>Devido aos resultados obtidos em cada VM apresentarem variações consideráveis, optou-se por exibir cada valor obtido, ao invés da média.

Tabela III  
TAXA LEITURA/ESCRITA DE EXPERIMENTOS *I/O-bound*

VMs em Execução	Taxa de Leitura/Escrita de cada VM (MB/s)
1	65.02/14.40
2	69.3/5.205 - 72.97/5.265
3	65.57/3.296 - 69.03/3.320 - 63.65/6.770
4	47.61/2.460 - 62.75/2.358 - 58.86/2.251 - 14.72/3.165
5	9.619/1.793 - 10.89/1.741 - 30.38/1.707 - 59.21/2.809
6	6.506/1.411 - 6.954/1.423 - 6.574/1.396 - 6.799/1.442 - 11.08/1.260 - 45.29/2.284

um número elevado de VMs há um decréscimo considerável na taxa de operações de leitura e escrita. É possível observar na Figura 3 que a oferta de processamento em modo convidado não cresce proporcionalmente ao número de VMs em execução. Esse fato pode ser explicado com base na percentagem crescente de espera de CPU por operações de E/S.

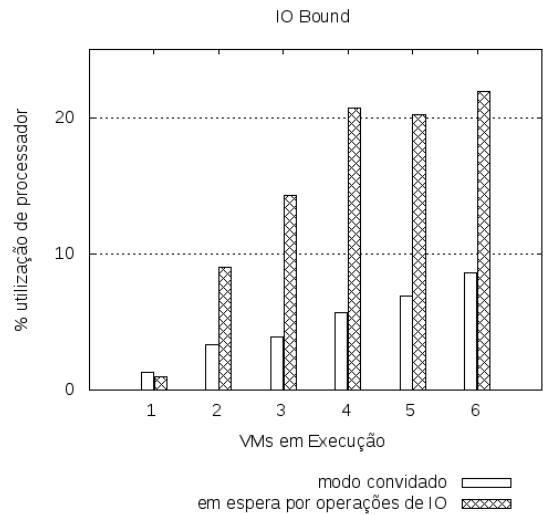


Figura 3. Utilização de CPU da máquina hospedeira durante experimentos *I/O-bound*

Ainda que cada VM possua o seu próprio disco virtual, elas compartilham um mesmo disco rígido físico. O uso intensivo de operações de leitura/escrita nessa unidade de armazenamento acarreta a degradação no desempenho de cada VM, podendo ser observada através da fila de requisições ao disco rígido do hospedeiro (vide Figura 4). Essa competição por recursos da máquina física explica o crescimento não linear na utilização de CPU em espera por operações de E/S, assim como a queda de utilização de processamento destinada às VMs.

Os resultados deste trabalho reafirmam que a virtualização de servidores adiciona um custo ao desempenho do sistema. No entanto, com a observação de métricas obtidas

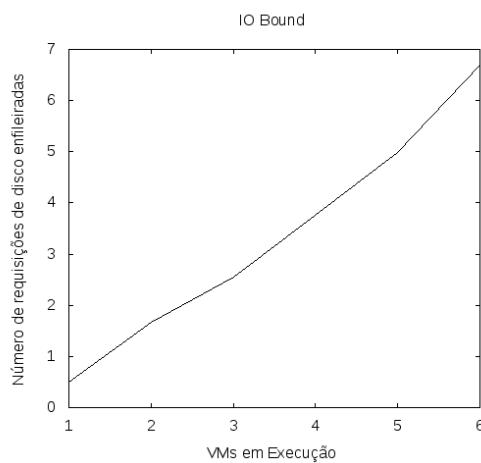


Figura 4. Formação de fila de requisições de disco durante experimentos *IO Bound*

por contadores de desempenho, é possível compreender as causas e relacionar os efeitos que levam à degradação do sistema. Apesar do conjunto de contadores descrito neste artigo ser reduzido, outras métricas também foram observadas e há indícios de que haja correlação entre vários contadores.

Futuramente pretende-se fazer uma análise estatística para comprovar a dependência entre métricas e estabelecer valores limites aos quais o sistema pode operar sem ocasionar perda de desempenho significativa. Além disso, outras métricas serão estudadas, incluindo métricas obtidas pelo monitoramento de interfaces de rede ou considerando infraestruturas com múltiplos discos rígidos.

Estudo similar ao apresentado neste trabalho foi realizado por Menon et al. em [7]. No entanto diferencia-se na utilização de ferramenta de monitoramento própria e trás problemas de desempenho de rede detectados. Gupta et al. consegue resolver alguns problemas de desempenho acrescentando isolamento extra entre as VMs [8]. Como resultado final desta pesquisa que se inicia, pretende-se desenvolver uma solução com abordagem diferente da apresentada no último trabalho citado. Almeja-se detectar problemas de desempenho em tempo de execução e com antecedência, de forma que seja possível realocar VMs para outro servidor presente na infraestrutura com o mínimo de impacto no desempenho dos sistemas.

## V. CONCLUSÃO

Embora o gerenciamento de recursos em computação em nuvem se beneficie da dinamicidade de infraestruturas virtuais, os impactos no desempenho decorrentes da virtualização não são detectados facilmente. Portanto, garantir acordos de níveis de serviço (SLA) nesses ambientes ainda é um grande desafio.

Este trabalho apresenta recursos para monitoramento em plataformas Windows e Linux, a serem utilizados na análise de desempenho de infraestruturas virtuais. Através da realização de experimentos, foi possível destacar efeitos da

degradação de desempenho decorrentes da virtualização, além de identificar a causa raiz com base em métricas coletadas por contadores de desempenho.

Os experimentos utilizaram *benchmarks* para estimular o sistema com cargas de trabalho simulando aplicações *CPU-bound* e *I/O-bound*. Essa abordagem evidencia padrões de uso de CPU, memória e disco em situações extremas. Embora essas cargas de trabalho não representem a grande maioria das aplicações, elas auxiliam na compreensão da degradação de desempenho e possibilitarão a identificação de correlação entre métricas coletadas por diferentes contadores de desempenhos. Futuramente, pretende-se usar essas correlações para desenvolver algoritmos capazes de identificar níveis de desempenho associados às VMs em execução e prever com antecedência a sobrecarga em máquinas hospedeiras. Em posse dessas informações, medidas preventivas permitirão um gerenciamento de recursos mais equilibrado, possibilitando que infraestruturas de computação em nuvem possam atender SLA mais rigorosos.

Em sequência a este trabalho, pretende-se realizar novos experimentos com utilização de padrões de carga mais realistas, como por exemplo padrões estabelecidos pelo consórcio TPC [9]. Além disso, é necessário identificar um maior número de correlações entre métricas de contadores de desempenho.

## REFERÊNCIAS

- [1] J. E. Smith and R. Nair, "An overview of virtual machine architectures," *Elsevier Science*, vol. 26, pp. 1–21, 2003.
- [2] R. McDougall and J. Anderson, "Virtualization performance: perspectives and challenges ahead," *SIGOPS Oper. Syst. Rev.*, vol. 44, 2010.
- [3] Y. Hu, X. Long, J. Zhang, J. He, and L. Xia, "I/O scheduling model of virtual machine based on multi-core dynamic partitioning," in *Proceedings of the 19th ACM HPDC*, 2010.
- [4] *HP Performance Engineering Best Practices*, HP, (Acesso em agosto de 2012). <http://h30499.www3.hp.com/t5/HP-LoadRunner-and-Performance/HP-Performance-Engineering-Best-Practices-Series/ba-p/2407627>.
- [5] wPrime. (Acesso em agosto de 2012). <http://www.wprime.net>.
- [6] CrystalMark, (Acesso em agosto de 2012). <http://crystalmark.info/software/CrystalDiskMark/index-e.html>.
- [7] A. Menon, J. R. Santos, Y. Turner, G. J. Janakiraman, and W. Zwaenepoel, "Diagnosing performance overheads in the xen virtual machine environment," in *Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments*, ser. VEE '05. New York, NY, USA: ACM, 2005, pp. 13–23.
- [8] D. Gupta, L. Cherkasova, R. Gardner, and A. Vahdat, "Enforcing performance isolation across virtual machines in xen," in *Proceedings of the ACM/IFIP/USENIX 2006 International Conference on Middleware*. Springer-Verlag New York, Inc., 2006. <http://dl.acm.org/citation.cfm?id=1515984.1516011>.
- [9] TPC. (Acesso em agosto de 2012). <http://www.tpc.org>.

# **NMS Anywhere: Uma Aplicação Voltada a Apoiar o Gerenciamento de Redes Através de Plataforma Móvel**

Fábio Centeno Pandolfo  
Faculdade de Informática – PUCRS  
fabio.pandolfo@serpro.gov.br

Cristina M. Nunes  
Faculdade de Informática – PUCRS  
cristina.nunes@pucrs.br

**Resumo**—Este trabalho propõe um estudo focado na utilização de plataformas móveis como ferramentas de apoio na atividade de gerência de redes. Para tanto, será apresentada uma aplicação desenvolvida para a plataforma *Android*. A finalidade da aplicação *NMS Anywhere* é monitorar sensores de temperatura de ambiente instalados em salas que comportam ativos de rede críticos e servidores.

## I. INTRODUÇÃO

A área de gerência de redes foi alavancada principalmente pela necessidade de monitoração e controle da vasta gama de equipamentos que compõem as redes de comunicação. Segundo [9], as redes de computadores e seus recursos associados, além das aplicações distribuídas, tem se tornado fundamental e de tal importância para uma organização, que elas basicamente “não podem falhar”. Isto se deve ao crescimento constante de requisitos quanto à disponibilidade.

Atualmente a grande maioria das aplicações de gerência de redes reside em plataformas de gerenciamento, as quais são compostas por computadores estratégicamente localizados dentro das organizações, como por exemplo, em *datacenters* ou em centros de gerência de redes. Dentre os diversos recursos e serviços de alta relevância que necessitam ser gerenciados por tais aplicações, figura a monitoração da temperatura de ambiente em salas que comportam ativos de rede críticos de uma organização.

Paralelamente a isto, há o fato de que cada vez mais está havendo a popularização das plataformas móveis. Estas vêm ganhando mercado rapidamente, e assim passaram a ser alvo do desenvolvimento de novas aplicações nas mais diversas áreas, inclusive na área de gerência de redes.

Dentro deste contexto, o presente trabalho visa propor um estudo sobre a utilização de aplicações desenvolvidas para plataformas móveis como ferramentas de apoio nas atividades de gerenciamento de redes. O trabalho descreve também o desenvolvimento, a implementação e a utilização de uma aplicação de gerência de redes, o *NMS Anywhere*, mais especificamente na área de gerência de falhas. Esta aplicação foi desenvolvida sobre a plataforma móvel *Android*.

O objetivo do *NMS Anywhere* é monitorar sensores de temperatura de ambiente instalados em salas que comportam ativos de rede críticos e servidores da empresa pública SERPRO (Serviço Federal de Processamento de Dados).

Este documento está dividido da forma como segue. A Seção II apresenta a fundamentação teórica referente à área de gerência de redes, e a seguir discorre sobre a plataforma *Android*, que está entre as mais utilizadas nos dispositivos móveis existentes hoje no mercado. A Seção

III descreve as características da arquitetura, da implementação e da utilização da aplicação de gerência de redes através de plataforma móvel, o *NMS Anywhere*. A Seção IV apresenta alguns trabalhos relacionados. Por fim, a Seção V apresenta as considerações finais a respeito do desenvolvimento deste trabalho, limitações encontradas e perspectivas futuras.

## II. FUNDAMENTAÇÃO TEÓRICA

A seguir serão apresentados alguns conceitos fundamentais para que se possa haver um melhor entendimento sobre a infraestrutura necessária para a atividade de gerenciamento de redes. Também serão apresentadas algumas das principais características da plataforma *Android*, a qual é utilizada em dispositivos móveis.

### A. Gerência de Redes

De acordo com [8], gerenciamento de rede é o procedimento que consiste em controlar todos os componentes de hardware e software da rede. As tarefas inerentes à gerência de redes, simplificadamente, são: obter informações da rede, tratar estas informações elaborando e disponibilizando diagnósticos, e com isto fornecer apoio para previsão de possíveis problemas, bem como encaminhamento de soluções para os mesmos.

Segundo [6], a gerência de redes se divide em cinco grandes áreas funcionais, entre elas a gerência de falhas. A função de monitorar os estados dos recursos verificando em qual ponto da rede e quando uma falha ou um erro pode ocorrer, está relacionada à gerência de falhas. A seguir estão listados os elementos que compõem a arquitetura de gerenciamento padrão Internet:

1) *MIB* (*Management Information Base*): é uma estrutura de dados que contém uma descrição de objetos gerenciados [10]. Os dados contidos nesta estrutura serão obtidos a partir dos agentes.

2) *Agente*: são programas que residem nos elementos da rede que devem ser gerenciados [10]. Eles coletam e armazenam diversas informações de gerenciamento.

3) *Gerente*: é um programa executado em uma estação servidora que permite a obtenção e o envio de informações de gerenciamento junto aos dispositivos gerenciados mediante a comunicação com um ou mais agentes [3].

4) *SNMP* (*Simple Network Management Protocol*): é um protocolo da camada de aplicação, utilizado no gerenciamento de redes TCP/IP [2].

### B. Android

O *Android* é a plataforma para dispositivos móveis da *Google*, sendo uma das plataformas mais populares para dispositivos móveis disponíveis no mercado.

De acordo com [7], o *Android* surgiu através da junção da *Google* com outras empresas para a formação da chamada OHA (*Open Handset Alliance*), que foi responsável pelo desenvolvimento da plataforma.

A plataforma *Android* apresenta algumas características atraentes, possuindo licença *Open Source*, abrangendo diversos fabricantes de dispositivos, dando suporte a multitarefa e permitindo aplicativos não proprietários. Desta forma, é possível que outras empresas e até usuários comuns criem aplicativos específicos para utilização pessoal, bem como corporativa.

Em função de todas essas características, a plataforma *Android* vem ganhando cada vez mais espaço no mercado de dispositivos móveis e já se consolidou como uma das mais populares atualmente.

## III. NMS ANYWHERE

A motivação inicial para o desenvolvimento do *NMS Anywhere* surgiu a partir de uma necessidade real de aplicação para a plataforma de gerenciamento de redes da empresa pública SERPRO (Serviço Federal de Processamento de Dados) Regional Porto Alegre.

A referida empresa possui uma aplicação de gerência de redes para a sua rede local, o *Zabbix*<sup>1</sup>, que contempla todo o prédio da regional. A partir do *Zabbix* estão monitorados todos os principais ativos de rede (roteadores, *switches* e servidores) com o maior grau de relevância para a disponibilidade da rede da empresa. Dentro desse ambiente, encontram-se ainda equipamentos que gerenciam sensores de temperatura de ambiente, os quais estão instalados nas salas que comportam os ativos de rede visando o monitoramento das mesmas. Tais equipamentos suportam o protocolo de gerência SNMP.

A aplicação *Zabbix*, desenvolvida em PHP (*PHP Hypertext Preprocessor*), é executada sobre um servidor *Apache* utilizando um banco de dados *MySQL*. Através do *Zabbix* é realizada a gerência de toda a rede da regional, inclusive dos equipamentos que gerenciam os sensores de temperatura instalados nas salas de ativos de rede, através do protocolo SNMP.

### A. Objetivo

A fim de diminuir o tempo de resposta em caso de falhas, proporcionando maior eficiência no gerenciamento da rede, levantou-se a necessidade de transportar o monitoramento dos equipamentos que gerenciam os sensores de temperatura, também para a plataforma móvel *Android*.

Dentro desse contexto surgiu a motivação inicial para o desenvolvimento da aplicação *NMS Anywhere* sobre a plataforma *Android*. O *NMS Anywhere* tem a finalidade de gerenciar os sensores de temperatura instalados nas salas

que comportam os principais ativos de rede do SERPRO - Regional Porto Alegre. Desta forma, será possível prestar suporte nas atividades de gerenciamento da rede em conjunto com a aplicação de gerência *Zabbix*. Para tanto, a aplicação *NMS Anywhere* deve ser integrada à aplicação *Zabbix*, a fim de viabilizar o cenário acima descrito.

### B. Escopo

O escopo da aplicação projetada e desenvolvida neste trabalho se restringe a área de gerência de falhas, que é uma das cinco grandes áreas funcionais no gerenciamento de redes.

O escopo também é restrito exclusivamente ao monitoramento dos equipamentos que gerenciam os sensores de temperatura. Tais equipamentos serão tratados pelo termo ‘elemento’ no ambiente da aplicação.

### C. Arquitetura

A plataforma *Android* é de propriedade da empresa *Google*, e possui licença aberta para desenvolvimento na linguagem de programação *Java*, a qual foi utilizada na implementação e desenvolvimento da aplicação *NMS Anywhere*.

Para possibilitar a integração do *NMS Anywhere* com a aplicação *Zabbix*, optou-se por acessar as informações referentes aos elementos e sensores gerenciados diretamente na base de dados do próprio *Zabbix*, que por sua vez colhe estes dados via SNMP.

Quando o usuário estiver no perímetro da rede Wi-Fi do Serpro, o *smartphone* automaticamente conectará na intranet da empresa. Ao detectar que o *smartphone* possui tal conexão na rede, o *NMS Anywhere* passará a acessar os dados do banco *MySQL* no servidor da aplicação *Zabbix* diretamente através do protocolo JDBC (*Java Database Connectivity*), conforme demonstrado na Figura 1.

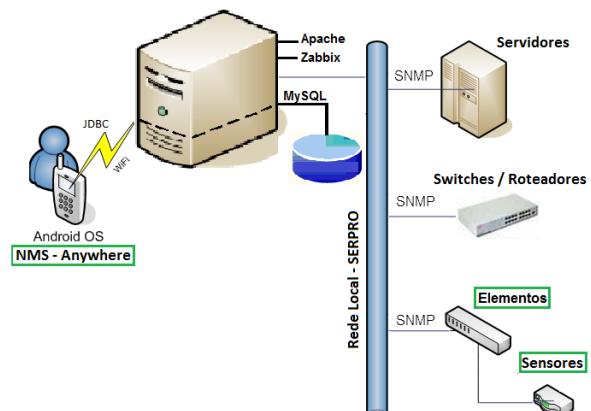


Figura 1. Integração com o *Zabbix* via JDBC.

Já quando o usuário se encontrar fora do perímetro da rede Wi-Fi do Serpro, o *NMS Anywhere* utilizará primariamente uma rede Wi-Fi externa, caso haja uma disponível, para se comunicar com o servidor *Zabbix* e desta forma acessar seus dados. Caso não haja nenhuma rede Wi-Fi disponível, será utilizada conexão 3G do

<sup>1</sup> <http://www.zabbix.com/>

smartphone para realizar a comunicação com o servidor Zabbix. Essas formas de acesso se darão pela internet, através do protocolo HTTP (*HiperText Transfer Protocol*).

Para prover esse tipo de comunicação foi implementado neste projeto um *Web Service*. Através desse *Web Service*, a aplicação *NMS Anywhere* terá a possibilidade de fazer requisições via HTTP ou HTTPS ao servidor do Zabbix. Estas requisições serão recebidas, processadas e respondidas pelo *Web Service* com os dados solicitados pelo *NMS Anywhere*. Para tanto, o *Web Service* estará hospedado no próprio servidor do Zabbix, fazendo uso do servidor *Apache* para trafegar as informações através do protocolo HTTP ou HTTPS, conforme ilustrado na Figura 2.

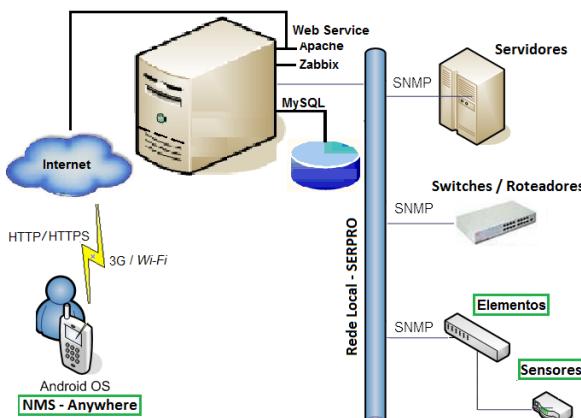


Figura 2. Integração com o Zabbix via Web Service.

De acordo com [4], a tecnologia de Web Services pode ser conceituada, simplificadamente, como uma arquitetura para distribuição de serviços, sendo que os componentes são independentes de plataforma e permitem a interoperabilidade entre aplicações.

Segundo [1] e [5], a independência de plataforma provida pelos *Web Services* se dá em função da adoção de padrões do tipo XML (*Extensible Markup Language*) e JSON (*JavaScript Object Notation*) para codificação das mensagens que trafegam através dos protocolos de aplicação, como o SOAP (*Simple Object Access Protocol*).

#### D. Descrição

Os usuários do *NMS Anywhere* serão os gerentes da rede e farão uso da aplicação através de seus *smartphones* pessoais.

A seguir estão descritas as principais funcionalidades presentes no menu principal do *NMS Anywhere*.

1) *Cadastrar elemento*: Esta funcionalidade permite que o usuário faça o cadastro de novos elementos na base de dados local do *NMS Anywhere*. Após o usuário informar o IP do elemento a ser cadastrado, a aplicação faz uma pesquisa na base de dados remota do Zabbix. Depois de realizada a pesquisa o *NMS Anywhere* persiste em sua base local os dados referentes ao elemento

pesquisado bem como aos seus respectivos sensores gerenciados. A primeira tela da Figura 3 apresenta a listagem de todos os elementos cadastrados no *NMS Anywhere*. Já a segunda tela apresenta todos os sensores gerenciados pelo elemento Poseidon\_OPPAE<sup>2</sup>.

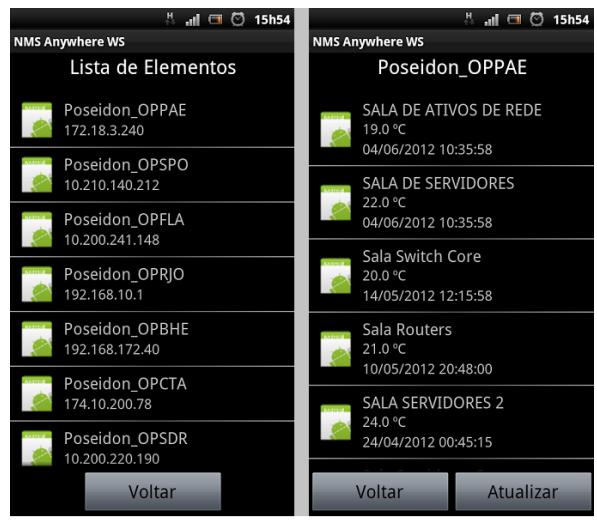


Figura 3. Tela de listagem de elementos seguida da tela de sensores do elemento Poseidon\_OPPAE.

2) *Consultar sensor*: Para realização desta consulta o usuário deve informar nome ou parte do nome do(s) sensor(es) a ser(em) consultado(s). Como resultado é retornado uma lista com todos os sensores que possuem o nome ou parte do nome igual ao informado pelo usuário. A partir dessa lista é possível visualizar as informações detalhadas de um determinado sensor selecionando-o, ou ainda atualizar as informações de todos os sensores presentes na mesma, acessando a base de dados remota do Zabbix.

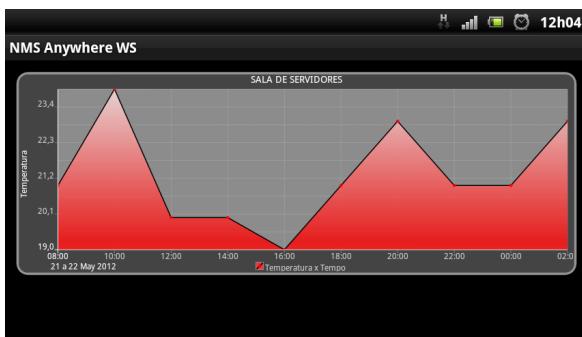
3) *Atualização automática*: Esta funcionalidade permite ao usuário programar períodos de dias, semanas ou meses, em que serão realizadas atualizações automáticas das informações de todos os sensores cadastrados na base de dados local do *NMS Anywhere*. Estas atualizações são feitas através da base de dados remota do Zabbix, e o tempo de intervalo entre cada uma delas também é definido pelo usuário, o qual pode ter sua unidade em segundos, minutos ou horas. Esta funcionalidade tem ainda a capacidade de ser executada em *background*, possibilitando inclusive que a aplicação seja fechada sem afetar o processo de atualização automática.

4) *Gerar gráfico*: Para ter acesso a esta função, o usuário deve selecionar o sensor desejado a partir da tela de listagem de sensores ou de consulta de sensores. Após selecionar o sensor, será exibida a tela com as suas informações detalhadas, e a partir dela estará disponível a função de geração de gráfico. Seguramente esta

<sup>2</sup> Equipamento que suporta o protocolo SNMP e é capaz de gerenciar até cinco sensores de temperatura através da sua MIB.

funcionalidade figura entre as mais úteis da aplicação, pois possibilita que o usuário verifique o comportamento da temperatura de um determinado sensor ao longo do tempo. Para possibilitar isto, a aplicação permite que o usuário determine o período de tempo que deseja para gerar o gráfico, bem como o tempo de intervalo entre cada ponto da função temperatura X tempo, o qual pode ter sua unidade em minutos ou horas.

A Figura 4 demonstra um gráfico da função temperatura X tempo que compreende o período de 21 a 22 de maio de 2012, com tempo de intervalo de duas horas entre cada ponto de função.



**Figura 4. Gráfico da função temperatura X tempo de um sensor.**

#### IV. TRABALHOS RELACIONADOS

As aplicações *Mobbix*<sup>3</sup> e *MoZBX*<sup>4</sup> consistem em clientes do *Zabbix* voltados para plataformas móveis. Depois de instaladas no dispositivo móvel, estas aplicações passam a acessar os dados do *Zabbix* através do endereço URL (*Universal Resource Locator*) do servidor. Dessa forma o *Mobbix* e o *MoZBX* compartilham o banco de dados utilizado pelo *Zabbix*, e dependem diretamente da disponibilidade do servidor *Apache* para realização das suas operações.

A aplicação *Mozaby*<sup>5</sup> foi desenvolvida pela empresa *Kodai Terashima*. Esta aplicação possui uma arquitetura bastante semelhante à arquitetura do *Mobbix* e do *MoZBX*, uma vez que também se trata de um cliente da aplicação *Zabbix* voltado para plataforma móvel. Porém o *Mozaby* foi desenvolvido para a plataforma *iOS* da empresa *Apple*, possuindo licença proprietária.

Em comparação com estas aplicações, o *NMS Anywhere* apresentou uma vantagem arquitetural significativa. Isto se deve ao fato de que o código da aplicação além de ser executado no dispositivo móvel, possui base de dados própria. Caso ocorra indisponibilidade do servidor *Zabbix*, o *NMS Anywhere* continua oferecendo as operações de listagem, consulta e geração de gráficos históricos a partir dos dados persistidos em sua base local até o momento da indisponibilidade no servidor. Já as outras aplicações

pesquisadas apresentariam perda total das suas funcionalidades neste mesmo cenário.

#### V. CONSIDERAÇÕES FINAIS

O presente trabalho apresentou um estudo voltado para uma alternativa de apoio nas atividades de gerenciamento de redes. Tal alternativa se traduz em uma aplicação de gerência de redes, mais especificamente da área de gerência de falhas, projetada sobre plataforma móvel, o *NMS Anywhere*.

O *NMS Anywhere* foi projetado e desenvolvido com a finalidade de gerenciar equipamentos e sensores de temperatura de ambiente, os quais estão instalados em salas que comportam os principais ativos de rede da empresa pública SERPRO (Regional Porto Alegre).

Portanto foram cumpridos os objetivos à que a aplicação se propõe, de maneira que é possível gerar gráficos históricos a partir dos dados armazenados dos sensores gerenciados.

Como trabalhos futuros pretende-se transportar a gerência de outros ativos de rede também já gerenciados no *Zabbix*, para a aplicação *NMS Anywhere*. Além disso, pretende-se implementar *triggers*, possibilitando uma gerência pró-ativa da rede. Com isso o usuário da aplicação passaria a ser alertado automaticamente na eminência de uma falha em um elemento gerenciado, possibilitando assim a sua prevenção.

#### REFERÊNCIAS

- [1] BASCI, D., M. S. **Data complexity metrics for xml web services**. Advances in Electrical and Computer Engineering, v. 9, p. 9 -15, 2009.
- [2] CASE, J.; FEDOR, M.; SCHU\_SATLL, M.; DAVIN, J. **A simple network management protocol (SNMP)**. Request for Comments RFC 1157. July 2002. 45. Disponível em: <<http://www.rfc-editor.org/rfc/rfc1157.txt>>. Acesso em: agosto/2011.
- [3] DIAS, Beethoven Z.; JUNIOR, Nilton A.; **Protocolo de Gerenciamento SNMP**. Centro Brasileiro de Pesquisas Físicas, 2001. Disponível em: <<http://biblioteca.cat.cbpf.br/pub/apub/nt/2001/nt00601.pdf>>. Acesso em: agosto/2011.
- [4] FULLER, J. et al. **Professional PHP Web Services**. Birmingham, UK: Wrox, 2003.
- [5] JUN, Y.; ZHISHU, L.; YANYAN, M. **Json based decentralized sso security architecture in e-commerce**. In: Proceedings of the 2008 International Symposium on Electronic Commerce and Security, ISECS '08, Washington, DC, USA: IEEE Computer Society, 2008, p. 471-475 (ISECS '08). Disponível em: <http://dx.doi.org/10.1109/ISECS.2008.171>. Acesso em: junho/2012.
- [6] LOPES, Raquel V. et al. **Melhores Práticas para Gerência de Redes de Computadores**. Rio de Janeiro: Campus, 2003.
- [7] MARTINS, Rafael J. Werneck de A. **Desenvolvimento de Aplicativo para Smartphone com a Plataforma Android**. Projeto Final de Graduação. Curso de Engenharia da Computação. PUC-RJ. Rio de Janeiro, dezembro 2009.
- [8] RIGNEY, Steve. **Planejamento e gerenciamento de redes**. 1ª edição. Editora Campus, 1996.
- [9] RNP, Rede Nacional de Ensino e Pesquisa. **Boletim Bimestral sobre Tecnologia de Redes: Introdução a Gerenciamento de Redes TCP/IP**, 1997. Disponível em: <<http://www.rnp.br/news/gen/9708/n3-2.html>>. Acesso em: agosto/2011.
- [10] TANENBAUM, Andrew S. **Computer Networks** – 5. ed. Amsterdam: Prentice Hall, 2010.

<sup>3</sup> <http://www.mobix.net>

<sup>4</sup> <http://www.mozbx.net/>

<sup>5</sup> <http://www.mozaby.com/>

---

**II**

## **Aplicações, Medições e Monitoramento**

---



# **Backup distribuído: uma implementação funcional**

Ciro Esteves Lima Sobral  
NCE - UFRJ  
cirosobral@ufrj.br

Álvaro Vinícius de Souza Coêlho  
DCET - UESC  
degas@uesc.br

**Resumo**—O processo de cópia de segurança (backup) responde pela salvaguarda de patrimônios que apresentam valor financeiro e são importantes para a continuidade dos trabalhos das organizações que os possuem: os dados. Enquanto isso, essas mesmas organizações possuem estações de trabalho, que em geral apresentam espaço ocioso em disco. O presente trabalho descreve um sistema de backup de dados (cópia de segurança), que utiliza o espaço em disco disponível de um conjunto de máquinas para armazenar os dados de backup de outras. Este processo permite criar redundância dos dados dos servidores, aumentando a segurança dos mesmos, aproveitando um recurso já existente. Mostramos que o sistema proposto realiza com sucesso as tarefas de backup e restauração dos dados, apresentando um bom desempenho nas duas tarefas.

## I. INTRODUÇÃO

O grande volume de dados e informações produzidas pela sociedade moderna demanda estruturas de reprodução e armazenamento cada vez mais potentes, tornando-as cada vez mais abundantes nas organizações. Com isso, a capacidade de armazenamento dos discos rígidos dos computadores cresce mais do que os usuários comuns necessitam para o seu trabalho diário. Assim, em empresas que possuem estações de trabalho, é comum a existência de espaço de armazenamento de informações ocioso nos discos dessas máquinas.

Por outro lado, a cada dia também cresce a necessidade de se fazer o armazenamento e proteção das informações nas empresas para prevenir os diversos riscos inerentes à perda de dados, muitos dos quais tem grande impacto financeiro. Desse modo, se gasta cada vez mais com estruturas e estratégias de backup (cópia de segurança) para minimizar os riscos, com a contrapartida do aumento de custos. As alternativas de backup existentes, mesmo as mais complexas e onerosas, são passíveis de falha, justificando a necessidade de estudos na busca de formas mais eficazes e de baixo custo para proteger as informações empresariais e mesmo pessoais de grande valor para os usuários.

Neste trabalho apresentamos uma implementação funcional de um sistema de backup distribuído, que torna útil o espaço ocioso nos discos das máquinas de empresas e organizações, utilizando-os para o armazenamento das cópias de segurança das suas informações relevantes, provendo confiabilidade e segurança sem custos adicionais.

## II. DESCRIÇÃO DO PROBLEMA

No mundo corporativo, a realização de backups é uma prática cada vez mais necessária no cotidiano das empresas. Como a perda de dados tem potencial de causar

grandes danos financeiros e operacionais, em algumas organizações a realização de backups cumpre importante papel no gerenciamento da continuidade de negócios [1]. Este é um problema complexo de se equacionar. Por um lado, a ausência de um sistema seguro de backup oferece o risco da perda de informações, com os custos daí decorrentes: custo da recuperação de arquivos, lucros cessantes, custos com possíveis ações judiciais e outros ainda mais difíceis de serem contabilizados, como dano à imagem da corporação [1]; por outro lado, as estratégias de backup, em geral, oneram as empresas com a compra e manutenção de equipamentos, contratação de pessoal e treinamento para proteger seus dados. A escolha do mecanismo de backup utilizado deve buscar minimizar os riscos, mantendo os custos em níveis aceitáveis. Isto é verdade mesmo para usuários domésticos, que também estão sujeitos à necessidade de garantir a persistência dos dados contidos nos computadores, como e-mails, contatos, e arquivos de valor sentimental (vídeos, fotos e músicas). Estudos mostram que 6% dos usuários domésticos perdem alguma informação dos computadores por ano [2].

O processo de backup de dados pode ser efetuado através da simples cópia destes em mídias como CDs, DVDs, pendrives, discos rígidos, fita magnética, etc [2]. De acordo com estudo da Cibecs [1], no mundo corporativo a cópia dos arquivos para um servidor de dados ou a cópia para um disco rígido externo representam 47% dos sistemas de backup utilizados. Este processo gera custos com equipamentos de uso específico para manutenção do backup. Além disso, os equipamentos de backup estão potencialmente à mercé de vários incidentes que causam também a perda dos dados originais, como roubo, incêndios, etc. Isto faz com que as políticas de backup determinem que a mídia em que foi realizada a cópia seja transportada para outro local, a fim de garantir a segurança do equipamento e, consequentemente, dos dados. Por esse motivo, o processo de backup implica em custos cada vez mais significativos para as organizações.

Ao mesmo tempo, enquanto a capacidade instalada em redes de computadores cresce, a utilização da capacidade de armazenamento dos discos rígidos dos computadores é tipicamente baixa. Estudos apontam que se utiliza aproximadamente 50% do espaço disponível nos discos [3]. Considerando que as empresas normalmente trabalham com uma grande quantidade de máquinas, pode-se estimar que há um contingente considerável de recursos de armazenamento ociosos. Estes recursos podem ser aproveitados a fim de suportar o processo de backup, tornando-o mais simples e barato, além de agregar valor aos dispositivos

de armazenamento já disponíveis. Neste sentido, uma estratégia viável e economicamente interessante é fazer o backup usando a transmissão dos dados através da rede de computadores, armazenando-os em máquinas distintas, geograficamente separadas.

### III. O BACKUP DISTRIBUÍDO

No processo de backup que propomos neste trabalho possui três atores principais: o *cliente*, que solicita a operação de backup, o *servidor de backup*, que oferece este serviço e o *servidor de metadados*, que provê um indexador de forma que os dados de cada cliente possam ser localizados satisfatoriamente.

#### A. Funcionamento

Na operação deste sistema, o cliente se conecta ao servidor de metadados e solicita deste a lista de servidores de backup ativos. De posse desta lista, o cliente prepara seus arquivos para serem copiados, empacotando-os junto com suas estruturas de diretórios em um único arquivo e compactando-o a fim de reduzir o espaço necessário para armazenamento e diminuir o tráfego na rede. Este processo irá gerar um único arquivo contendo todos os dados a serem persistidos. Em seguida este arquivo é criptografado, por questão de segurança já que será alojado em máquinas remotas, e particionado em múltiplos arquivos menores, de forma a permitir que cada servidor de backup aloje parte dos dados de um mesmo cliente. Desta forma, pode-se implementar uma política de redundância, em que cada parte dos dados esteja alojada em mais de um servidor. O cliente então escolhe aqueles servidores com espaço em disco suficiente para receber seus arquivos, e envia os arquivos. Após a confirmação da chegada dos dados nos servidores de backup, o cliente informa ao servidor de metadados quais os arquivos foram armazenados em quais servidores de dados.

Caso seja necessária a restauração dos dados, o cliente solicita o início do processo junto ao servidor de metadados, que o atende enviando uma lista com os nomes dos arquivos que compõem seu backup, além dos respectivos servidores em que estes se encontram. De posse desta informação, o cliente se conecta aos servidores, requisitando destes os arquivos que lhe interessam. Após receber todos os fragmentos do backup, procede-se a um processo inverso daquele realizado antes do envio: os arquivos são concatenados, formando um único arquivo que será descriptografado, descompactado e finalmente desempacotado, restaurando a estrutura de pastas e arquivos original.

#### B. Detalhes da Arquitetura

O servidor de metadados é um nó central no sistema, muitas vezes referido na literatura como *tracker* [4]. É o responsável por armazenar as “informações dos dados”, que permitem definir que parte dos dados de cada cliente está com cada servidor. Estas informações são chamadas de metadados. Neste trabalho, estas informações são o nome do arquivo, o *hostname* do cliente e o *hostname* do servidor, havendo uma entrada desta natureza para cada um dos arquivos persistidos. Cabe também ao servidor

de metadados indicar ao cliente quais os *hostname* dos servidores de dados ativos, bem como o espaço em disco livre em cada um deles. Para o pleno funcionamento do sistema é necessário que o servidor de metadados esteja ativo, acessível pela rede tanto pelas máquinas que abrigarão os servidores de dados quanto pelas que utilizaram o sistema como cliente.

O servidor de backup é o agente que irá de fato efetuar a operação de armazenamento dos dados a serem persistidos. Na concepção deste sistema, idealmente deve existir um conjunto de servidores de dados, de forma a se reduzir a carga de trabalho solicitada a cada um deles, bem como para se implementar políticas de redundância de dados. Quando iniciado o serviço, estes servidores se conectam ao servidor de metadados, informando seu endereço de rede e o espaço em disco disponível. A partir deste momento, o servidor de metadados inicia o monitoramento do estado da conexão de cada um dos servidores de dados. Este monitoramento é realizado para que a lista dos servidores disponíveis esteja sempre atualizada, pois é ela que irá indicar ao cliente em que máquinas poderá ser alojado seu backup.

O cliente é a parte do sistema que terá seus dados armazenados. Para operar o sistema, o cliente precisa se conectar ao servidor de metadados, a fim de enviar suas solicitações tanto de armazenamento quanto de restauração de dados.

Caso o cliente envie uma solicitação de backup, o servidor de metadados irá retornar uma lista contendo o *hostname* dos servidores de dados conectados e o espaço em disco disponível em cada um deles. O cliente, então, vai empacotar, compactar, criptografar e particionar seus dados e então se conectar aos servidores disponíveis para proceder ao envio. Após a confirmação do armazenamento, o cliente informa ao servidor de metadados o seu *hostname* e a lista de arquivos armazenados, juntamente com o *hostname* dos servidores onde estes foram alojados.

Se, por outro lado, a solicitação for de restauração, o servidor de metadados irá buscar nos registros dos arquivos enviados através do *hostname* do cliente e enviará uma lista contendo os arquivos e o *hostname* dos servidores de dados. O cliente então tratará de se conectar a cada um dos servidores e solicitar a recuperação de seus arquivos.

#### C. Detalhes da Implementação

O projeto foi desenvolvido utilizando a linguagem de programação Java. Portanto o *bytecode* pode ser executado em qualquer plataforma que rode uma máquina virtual Java capaz de se comunicar através de *sockets*.

Apenas o cliente utiliza ferramentas do sistema operacional Linux como o *tar*, para empacotamento dos arquivos, *gzip* para compactação e *openssl* para criptografia. Todos esses programas são chamados através de um *bash script*. Dessa forma, o cliente deverá ser executado em um plataforma Linux. Ainda assim, caso o usuário deseje utilizar em uma plataforma Windows, basta usar um conjunto de ferramentas GNU como *MinGW* ou *Cygwin*.

#### IV. RESULTADOS

Para verificar o funcionamento do sistema, foi feita uma instalação-piloto para servir o backup dos dados do Colegiado do Curso de Ciência da Computação (COLCIC) da Universidade Estadual de Santa Cruz, em Ilhéus/Ba. Os testes basicamente executaram tarefas de backup e restauração de dados utilizando a arquitetura descrita na seção III-B. Foi feito o backup do diretório de usuários (`/home`) de uma máquina Linux, contendo uma estrutura com um total de 8.883 arquivos, cujo tamanho total foi de 330,7MB. No total foram utilizadas 5 máquinas conectadas numa rede Ethernet, nomeadas como segue.

- 3118-01 (cliente);
- 3118-02 (servidor de metadados); e
- 3118-03, 3118-05 e 3118-06 (servidores de dados).

O servidor de metadados foi instalado na máquina 3118-02, enquanto os servidores de backup executaram nas máquinas 3118-03, 3118-05, 3118-06. O experimento processou toda a operação desde o início. Assim, o cliente, executado na máquina 3118-01, se conectou ao servidor de metadados e recebeu a lista dos servidores de dados ativos. Neste experimento, após o empacotamento, compactação e criptografia, dados de backup acabaram divididos em partes de no máximo 100MB, resultando em um total de 2 partes com tamanho 100MB e uma com tamanho de 11,9MB. Cada uma das três partes do backup foi transferida para um servidor de backup diferente.

Nesse teste, foram medidos os tempos de realização do processo de backup e de restauração com o programa `time` do Linux. No total, o tempo gasto na compactação, encriptação e divisão dos arquivos é mostrado na tabela I.

Tabela I  
TEMPOS GASTOS NOS DIFERENTES PROCESSOS

	Compactação, encriptação e divisão dos arquivos	Processo completo de Backup	Concatenação, desencriptação e descompactação dos arquivos	Processo completo de Restauração
<i>Real</i>	0m43.348s	1m6.755s	0m32.786s	0m57.183s
<i>User</i>	0m48.827s	1m7.656s	0m32.710s	0m52.055s
<i>Sys</i>	0m02.396s	0m3.168s	0m02.252s	0m03.616s

##### A. Considerações sobre Escalabilidade

O ambiente em que este sistema foi testado é pequeno, em função de que os recursos necessários para se montar experimentos maiores, que envolvam mais máquinas e mais dados para backup ainda não estão disponíveis. Isto não permitiu uma análise aprofundada a respeito das características de funcionamento do sistema – notadamente sua escalabilidade. Todavia algumas considerações podem ser feitas.

De modo geral os aspectos que impactam na escalabilidade deste sistema, e consequentemente no tempo necessário para se executar as operações de backup e restauração, são: a velocidade da rede, pois quanto mais rápida forem as conexões mais rapidamente os arquivos poderão ser copiados; a quantidade de servidores de backup, já que quanto mais servidores existirem maior será a

quantidade de redundância possível de ser implementada, aumentando a confiabilidade e possibilitando maior vazão de dados no processo caso a infraestrutura de rede permita; e o servidor de metadados, que pode não atender com presteza a todas as solicitações de backup e restauração em ambientes onde existe muita demanda por estes serviços. Este problema pode ser minimizado com a prática de se fazer as operações de backup em momentos cujo tráfego na rede seja pequeno. Além disso, pode-se estudar estratégias que implementem o servidor de metadados de maneira distribuída, diluindo a carga de trabalho por diferentes máquinas.

#### V. TRABALHOS CORRELATOS

Lilliebridge et al. propõem um sistema de backup cooperativo através da Internet [5]. O sistema funciona utilizando uma rede entre pares (P2P) com a qual dois usuários podem formar parceria, disponibilizando iguais espaços em disco, garantindo uma troca justa. Desta forma, um usuário *A* armazena os dados do usuário *B* e vice-versa. O sistema proposto, no entanto, sofre do inconveniente dos *free-riders*<sup>1</sup>, que podem realizar o backup de seus dados sem garantir, reciprocamente, a integridade dos dados do outro. A cooperação dos pares é assegurada através da realização de testes periódicos, que verificam a presença e a integridade dos arquivos de backup na máquina remota. Este trabalho presume a utilização de servidores de um mesmo domínio administrativo, eliminando o problema dos *free-riders*.

O OurBackup é um sistema de backup que utiliza uma rede social para construção da relação entre os usuários [6]. Nesse sistema, um usuário efetua o backup de uma parcela selecionada de seus dados na máquina de seus amigos. A rede social embutida no sistema tem como finalidade prática diminuir a presença de *free-riders* e aumentar a recuperabilidade do backup. Apresenta um mecanismo de verificação da integridade do backup, da mesma forma que no trabalho de Lillibridge et al.. Entretanto, difere-se daquele sistema pela utilização de um servidor centralizado para armazenar os metadados e localizar os usuários do sistema. Neste trabalho também se utiliza um servidor centralizado para armazenar as informações sobre os arquivos armazenados e os servidores de dados ativos.

O pStore é descrito como um sistema de backup, baseado em uma rede P2P adaptativa, que utiliza espaço em disco de computadores conectados através da Internet para prover proteção de dados aos usuários [7]. Esse sistema é visto como uma junção de um sistema de armazenamento distribuído com um *framework* de versionamento. Alguns dos objetivos desse sistema são garantir especialmente confiabilidade e segurança. A confiabilidade do backup é garantida através da replicação de dados, criando cópias dos mesmos em servidores distintos. A segurança é obtida com a encriptação dos dados pelo cliente antes do envio.

<sup>1</sup>Em sistemas P2P, o termo *Free-rider* é utilizado para denominar usuários que se beneficiam sem colaborar, ou colaborando o mínimo possível.

Essa estratégia também será utilizada neste trabalho, criptografando os dados antes de enviá-los para o servidor.

Foram buscadas implementações dos sistemas citados acima mas, infelizmente, não foi encontrada a implementação ou registros de utilização real de nenhum dos trabalhos relacionados. Seria interessante obter informações sobre esses sistemas para comparação com o sistema proposto neste trabalho. Além disso, os testes realizados em cada trabalho tinham como objetivo verificar variáveis diferentes, não permitindo a comparação entre elas.

Lillibridge et al. simularam uma rede com 10 participantes e mediou a recuperabilidade do backup utilizando o método de correção de Reed-Solomon, com  $k = 6$  e  $n = 2$ . Dessa forma, até 2 das 8 partes do backup poderiam se perder. O processo de backup para 100MB de dados foi realizado em 12 minutos e de 1 GB em 2 horas [5]. No OurBackup foi simulada uma rede com banda, tamanho, redundância do backup, fragmentação dos arquivos e número de pares variáveis. Verificou recuperabilidade e tempo de recuperação tanto no uso da técnica de replicação quanto com erasure code [6]. No pStore os testes verificaram a utilização de rede e o espaço necessários para realização de backup incremental.

## VI. CONCLUSÃO

Neste trabalho foi proposto um sistema de backup que utiliza o espaço em disco ocioso de máquinas de uma organização, alcançando segurança e confiabilidade do processo a baixo custo, e aumentando a utilização dos recursos de armazenamento já existentes, sem que seja necessário nenhum investimento adicional para a operação de backup de dados.

O sistema encontra-se hospedado no Gitorious e seu endereço é <https://gitorious.org/dbbackup>. Pode-se também fazer um clone deste projeto através do repositório remoto. Todos os códigos fontes, **scripts** e um relatório técnico a respeito se encontram neste endereço.

Com este sistema o processo de backup tem maior confiabilidade e menor custo, sem perda da confidencialidade pois apesar de alojados remotamente, os dados são enviados após um processo de criptografia. A operação do sistema é simples, baseada em linhas de comando, e sua versão atual é completamente funcional.

Os metadados são imprescindíveis ao funcionamento do sistema e, por este motivo, o sistema não pode tolerar uma falha neste componente. Neste sentido, um trabalho que se posa é a implementação de soluções mais robustas do que o servidor centralizado utilizado aqui. Uma alternativa é fazer com que os servidores de backup sejam eles mesmos também servidores de metadados. Neste caso, a busca de um cliente pelos servidores de backup que possuem seus arquivos seria baseada em um processo de inundação, conforme acontece em sistemas P2P similares ao Gnutella [8].

Outra alternativa é a implementação de uma DHT para distribuir os metadados, adicionando redundância. Neste sentido, algumas implementações de DHT já foram propostas na literatura, como a CAN, Chord, Pastry e Tapestry [9], [10].

Assim como o servidor de metadados, o computador que hospeda os servidores de backup também podem sofrer falhas e perder os ou corromper os. Para lidar com este problema potencial é necessário criar um módulo de verificação de integridade dos arquivos, permitindo ao cliente periodicamente verificar o estado de seu backup. Uma estratégia que pode ser utilizada é a geração de um código MD5 pelo cliente e seu envio ao servidor de metadados. Desta forma, quando o cliente for verificar a integridade dos arquivos, o servidor de dados deve calcular o MD5 e enviar o resultado ao cliente que efetuará a comparação com o código armazenado no servidor de metadados. Além disso, pode-se employear técnicas de Utilização de *erasure codes*, que permite a recuperação da integridade dos dados mesmo com a perda parcial deles.

## REFERÊNCIAS

- [1] Cibecs, “Business data loss survey,” 2011, acessado em: 15 de novembro de 2011. [Online]. Available: [http://resources.idgenterprise.com/original/AST-0052774\\_NEW\\_Survey-2011\\_Aug.E.pdf](http://resources.idgenterprise.com/original/AST-0052774_NEW_Survey-2011_Aug.E.pdf)
- [2] Seagate, “Importance of backup,” [entre 1998 e 2011], acessado em: 17 de janeiro de 2012. [Online]. Available: <http://www.seagate.com/ww/v/index.jsp?vgnextoid=011ad85104b51210VgnVCM1000001a48090aRCRD>
- [3] J. R. Douceur and W. J. Bolosky, “A large-scale study of file-system contents,” *SIGMETRICS Perform. Eval. Rev.*, vol. 27, pp. 59–70, May 1999. [Online]. Available: <http://doi.acm.org/10.1145/301464.301480>
- [4] L. Toka, M. Dell’Amico, and P. Michiardi, “On scheduling and redundancy for p2p backup,” *CoRR*, vol. abs/1009.1344, p. 9, 2010.
- [5] M. Lillibridge, S. Elnikety, A. Birrell, M. Burrows, and M. Isard, “A cooperative internet backup scheme,” in *In Proceedings of the 2003 USENIX Annual Technical Conference*, 2003, pp. 29–41.
- [6] M. I. d. S. Oliveira, *OurBackup: Uma Solução P2P de Backup Baseada em Redes Sociais*. Campina Grande: UFCG, 2007.
- [7] C. Batten, K. Barr, A. Saraf, and S. Trepelin, “pstore: A secure peer-to-peer backup system,” Massachusetts Institute of Technology Laboratory for Computer Science, Technical Memo MIT-LCS-TM-632, October 2002.
- [8] A. H. Rasti, D. Stutzbach, and R. Rejaie, “On the long-term evolution of the two-tier gnutella overlay.” in *INFOCOM*. IEEE, 2006. [Online]. Available: <http://dblp.uni-trier.de/db/conf/infocom/infocom2006.html#RastiSR06>
- [9] K. Aberer and M. Hauswirth, “An overview on peer-to-peer information systems,” 2002.
- [10] S. Androultsellis-Theotokis and D. Spinellis, “A survey of peer-to-peer content distribution technologies,” *ACM Comput. Surv.*, vol. 36, pp. 335–371, December 2004. [Online]. Available: <http://doi.acm.org/10.1145/1041680.1041681>

# Influência do Cenário em Rede de Sensores Sem Fio para Rastreamento Animal

Moises D. Deangelo, Patrícia S. Domingues e Leonardo B. Pinho  
Engenharia de Computação – Universidade Federal do Pampa (UNIPAMPA) – Campus Bagé  
heco@unipampa.edu.br

**Resumo**— Aumentar a eficiência das práticas agropecuárias é uma necessidade fundamental em tempos de globalização, e faz parte das ações estratégicas governamentais em âmbito nacional e estadual. Para tanto, tecnologias automatizadas de manejo têm sido propostas e inseridas comercialmente nos ambientes de produção. Neste contexto, sistemas para rastreamento animal baseados em RSSF se apresentam como alternativa potencial. Este trabalho visa avaliar a viabilidade do uso de técnicas de localização que usam o LQI (*Link Quality Indicator*) para estimar da posição dos nós considerando a variabilidade deste indicador em função da relação entre as características particulares de diferentes cenários representativos de propriedades rurais e a altura dos animais (bovinos e ovinos). Os resultados preliminares dos testes empíricos sugerem que técnicas complementares para contornar a imprecisão causada em certos cenários serão fundamentais.

## I. INTRODUÇÃO

Rede de sensores sem fio (RSSF) é uma ferramenta para o sensoriamento distribuído de fenômenos ambientais, processamento de dados e disseminação de informações [1]. Muitas RSSF sofrem com o problema conhecido como visada, mais especificamente LOS (*Line-Of-Sight*), pois nem todos os ambientes fornecem esse requisito para um bom funcionamento sem interrupções, como sistemas baseados em LOS temos o GPS, sistemas acústicos e sistemas baseados em luz, porém entretanto na prática existe a necessidade de se ter um sistema robusto o suficiente para operar em NLOS (*Non-Line-Of-Sight*), o que significa a existência de obstáculos interferindo entre os sensores [2]. Diante disso são feitos estudos que objetivam contornar essa dificuldade. Rede de sensores sem fio são sistemas distribuídos para capturar e processar dados, as quais usam *links* de rádio para transmitir dados entre os sensores e um servidor [3], viabilizadas em função do crescente e rápido avanço tecnológico em termos de semicondutores, circuitos integrados e de MEMS (sistemas mecânicos microeletrônicos). Sensores autônomos são dispostos em ambientes externos e se comunicam com um nó sorvedouro. O processamento dos dados coletados pode ser feito em um desktop, com um programa específico para a aplicação sensoriada. Os nós sensores podem ser estáticos ou móveis, como no caso destes estarem presos a elementos a serem monitorados (animais de um rebanho), que tenham capacidade de movimentação própria (um veículo sob controle da administração da rede) ou por ação de terceiros (um sensor em uma roupa) [4].

O chamado rastreamento animal atual é utilizado apenas para garantia da procedência da carne após abate, onde é possível saber todo histórico de vacinas, doenças e manejo. Para tal usa-se equipamentos RFID passivos que devido as suas limitações dificultam o acesso as informações complementares e em tempo real dos animais

no campo. Para se obter informações em tempo real dos animais e coibir, por exemplo, o crime de abigeato (roubo de animais, prática comum em diversas regiões do país e em especial na chamada região da campanha, na fronteira do RS com o Uruguai), uma RSSF com diferentes tipos de sensores móveis poderia ser montada, mostrando a posição aproximada dos animais dentro de uma determinada área de cobertura. A utilização de uma rede de sensores para estimação de campo possibilita a realização das medidas em lugares inacessíveis, pois o observador pode estar a quilômetros de distância recebendo os dados. Esta é tipicamente uma aplicação de envio contínuo de dados, onde a qualidade da estimativa depende diretamente da frequência com que os dados são obtidos. Quanto maior esta frequência, tanto temporal quanto espacial, maior a precisão do sistema. No entanto, maiores frequências implicam um maior tráfego na rede e consequentemente um maior consumo de energia [5].

Para calcular a posição dos sensores em uma RSSF, diferentes técnicas de localização foram propostas. Usualmente, estas técnicas se baseiam no LQI (*Link Quality Indicator*), por meio do qual é possível verificar a qualidade do sinal que chega ao rádio, principalmente em função do RSSI (*Received Signal Strength Indicator*), o qual fornece a intensidade do sinal recebido de um determinado transmissor. A partir da coleta dos dados de LQI de diferentes sensores são necessários algoritmos específicos para estimar as localizações relativas entre estes por meio de triangulação. Por exemplo, o RPE (*Recursive Position Estimation*) e o DPE (*Direct Position Estimation*) são dois algoritmos relevantes para calcular a posição dos nodos [6].

Com base na revisão bibliográfica realizada, foi possível constatar que, embora existam diversos trabalhos abordando com diferentes ênfases diversos fatores que afetam a precisão destas técnicas, não é comum encontrar na literatura resultados empíricos de RSSF testadas em



Figura 1. Locais escolhidos para testes.



Figura 2. Cenários avaliados nos testes: a) Declive; b) Lago; c) Colina; e d) Plano.

cenários reais. Em particular, não foram identificados trabalhos que buscassem avaliar cenários característicos de propriedades rurais – mesclando áreas planas e áreas com declive, existência de açudes e colinas. Soma-se a esta heterogeneidade a altura na qual usualmente estariam os sensores, no caso de serem fixados ao pescoço dos animais.

## II. OBJETIVO

Avaliar empiricamente o comportamento do LQI em diferentes cenários representativos em uma propriedade rural visando poder identificar com precisão a localização dos animais no pasto sem a necessidade de contato visual ou físico com os mesmos.

## III. METODOLOGIA

### A. Nós Sensores

Os sensores utilizados são do *kit Freescale MC1322x* [6], com processador TDMI ARM7™ 32 bits, 26 MHz; alto poder de processamento e integração de sistema, com uma combinação de memória ROM, RAM e *flash*. A *Freescale* disponibiliza um ambiente de desenvolvimento chamado *Freescale BeeKit Wireless Connectivity Toolkit*, o qual inclui a *BeeKit GUI* que possibilita ao desenvolvedor criar, modificar e atualizar várias implementações de redes de sensores sem fio [7]. Segundo o fabricante o padrão ZigBee sem nenhum ganho de antena ou recursos foi criado para operar em um raio de 90 a 120 metros [8].

O referido *kit* possui três tipos de sensores, na Figura 3 observa-se um sensor SRB. Todos possuem interface UART (*Universal Asynchronous Receiver/Transmitter*) para programação e depuração.

Os sensores são compostos de quatro componentes: fonte de energia; unidade de sensoriamento (Sensor e conversor A/D – analógico digital); unidade de

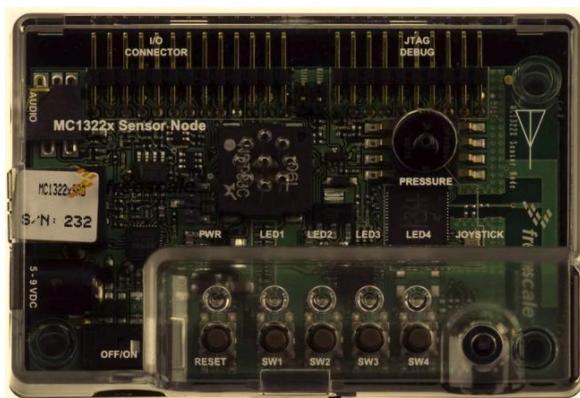


Figura 3. Sensor Freescale 1322x-SRB

processamento e unidade de comunicação (transceptor).

A unidade de sensoriamento caracteriza-se pela medição de grandezas físicas, que são adquiridas em sinais analógicos e são digitalizadas por um conversor A/D, de forma que a unidade de computação possa corretamente processar as informações [9].

### B. Cenários

Foram destacados quatro cenários no entorno do Campus Bagé da UNIPAMPA, apresentados na Figura 1 respectivamente.

Estes corresponderiam aos encontrados em um ambiente rural para estudos, sendo caracterizados como segue:

1) *Declive*: local onde um nó se encontra elevado em relação ao outro podendo haver visada direta ou não, como visto na Figura 2-a.

2) *Lago*: entre os dois pontos há um lago ou um açude, visto na Figura 2-b.

3) *Colina*: caracteriza-se por uma pequena elevação no terreno obstruindo o sinal do rádio, podemos ver um dos pontos da mesma na Figura 2-c.

4) *Plano*: local com pouca ou nenhuma variação de altitude. Contém apenas vegetação rasteira ou terra entre os pontos, como visto na Figura 2-d.

### C. Método de Medição

Nos quatro ambientes diferentes foram demarcados a partir de coordenadas da ferramenta *Google Earth* e com o auxílio do GPS de um *smartphone Samsung Galaxy Y* com o Sistema Operacional Android e a Aplicação GPS Status, locais de medição com distância de 100m entre um ponto e outro. A altura dos pontos (curva de nível) foi escolhida com cuidado para se ter o ambiente desejado nos experimentos práticos. A bateria de testes consistiu no envio de 100 pacotes de um sensor ao outro utilizando antenas *printed F* e estando ambos configurados para transmitir em 3dBm (2mW) que é a potência máxima fornecida por esses aparelhos da Freescale. O rádio receptor gerava um log com informações sobre os pacotes que chegavam, tais como: comprimento, LQI e quantidade de pacotes íntegros. Ambos os rádios foram fixados a hastes na mesma altura e na mesma posição (foram analisadas diferentes posições para verificar o efeito da posição da antena do equipamento). No estudo, foram adotadas três alturas: 1,5 m (simulando uma cabeça de gado no sobreano), 1,2 m (simulando uma cabeça de gado após 120 dias de nascido) e 0,8 m (simulando uma ovelha adulta). Posteriormente fez-se um estudo sobre os pacotes recebidos em relação ao LQI, que é o objeto de pesquisa, foram avaliadas apenas as recepções intactas garantidas por uma técnica de detecção de erros, CRC (*Cyclic*

Tabela I  
DADOS COLHIDOS NO DECLIVE

	Declive c/ Rádio na Horizontal	Declive c/ Rádio na Vertical
<b>LQI Máx. 1,5m</b>	-82	-87
<b>LQI Min. 1,5m</b>	-87	-90
<b>LQI Máx. 1,2m</b>	-86	-90
<b>LQI Min. 1,2m</b>	-94	-97
<b>LQI Máx. 0,8m</b>	-91	-97
<b>LQI Min. 0,8m</b>	-100	-100

Tabela II  
DADOS COLHIDOS NA COLINA

	Colina c/ Rádio na Horizontal	Colina c/ Rádio na Vertical
<b>LQI Máx. 1,5m</b>	-93	-93
<b>LQI Min. 1,5m</b>	-97	-98
<b>LQI Máx. 1,2m</b>	-99	-93
<b>LQI Min. 1,2m</b>	-100	-97
<b>LQI Máx. 0,8m</b>	Sem Sinal	Sem Sinal
<b>LQI Min. 0,8m</b>	Sem Sinal	Sem Sinal

*Redundancy Check), também conhecidos por códigos polinomiais.* No CRC é possível considerar a cadeia de bits a ser enviada como um polinômio cujos coeficientes são os valores 0 e 1, sendo as operações nesta cadeia interpretadas como aritmética polinomial [10].

#### D. Resultados e Discussões

As tabelas a seguir resumem os resultados preliminares obtidos (quanto menos negativo melhor)

Embora tenham ocorrido problemas durante os experimentos, que implicam na necessidade futura de produzir uma bateria mais completa de testes, foi possível coletar e analisar os dados, identificando algumas tendências, dentre as quais cabe destacar:

1) Percebe-se uma melhora no desempenho quando o rádio esta alinhado com o horizonte como se observa de maneira mais destacada na Tabela I, no contexto geral o mesmo também é observado, porém um pouco mais diluído.

2) Conforme esperado, nos locais com visada limitada (NLOS) houve interferência significativa do sinal por vezes inviabilizando sua detecção no receptor como visto na Tabela II;

3) Embora houvesse visada (LOS) entre os rádios no experimento do Lago (Tabela III), percebe-se uma deterioração no sinal em relação aos testes feitos no Plano (Tabela IV).

Diante das especificações do alcance dos sensores [8], considerou-se conveniente saber qual seria o alcance máximo dos mesmos sem nenhum incremento que pudesse aumentar o limite de transmissão, foi escolhida uma área plana, onde os sensores tinham visada a uma altura de 1,5m cada um. O rádio A ficou fixo em um ponto e o rádio B foi sendo deslocado em linha reta e em velocidade constante, mantendo a altura de 1,5m.

Tabela III  
DADOS COLHIDOS NO LAGO

	Lago c/ Rádio na Horizontal	Lago c/ Rádio na Vertical
<b>LQI Máx. 1,5m</b>	-80	-80
<b>LQI Min. 1,5m</b>	-85	-83
<b>LQI Máx. 1,2m</b>	-80	-80
<b>LQI Min. 1,2m</b>	-87	-83
<b>LQI Máx. 0,8m</b>	-81	-91
<b>LQI Min. 0,8m</b>	-86	-100

Tabela IV  
DADOS COLHIDOS NO PLANO

	Plano c/ Rádio na Horizontal	Plano c/ Rádio na Vertical
<b>LQI Máx. 1,5m</b>	Prob. Técnicos	-79
<b>LQI Min. 1,5m</b>	Prob. Técnicos	-88
<b>LQI Máx. 1,2m</b>	-76	-78
<b>LQI Min. 1,2m</b>	-77	-82
<b>LQI Máx. 0,8m</b>	-83	-82
<b>LQI Min. 0,8m</b>	-95	-86

Conforme esperado, a Figura 4 indica que quanto maior a distância menor a qualidade do link (LQI). Em alguns momentos nenhum pacote era recebido, justificando a teoria e a curva do gráfico. Os experimentos provam que a regra do desvanecimento do sinal (*Signal Path Loss*) esta correta, pois em condições não ideais a intensidade do sinal não decai com o quadrado da distância uma vez que existem diversas interferências físicas degradando o mesmo. Quando o LQI atinge -100 a recepção é nula. De 160m até 230m o gráfico mostra que não há nenhuma perda de pacotes, porém neste momento houve um problema com a contagem, portanto nesta faixa de 70m não se possui dados sobre a perda. A porcentagem de perda mostra que durante aquele teste x% dos pacotes não chegaram ao destino, cada barra denota uma sequência de testes. Observando a marcação em roxo pode-se ver que ao final dos 500m a perda de pacotes aumenta significativamente e neste mesmo momento o LQI (em azul) começa a variar menos se aproximando de -100, mostrando que a perda de pacotes está atrelada ao nível do LQI.

#### IV. CONSIDERAÇÕES FINAIS

Neste trabalho foram realizados experimentos práticos em cenários reais representativos de propriedades rurais onde sistemas de localização se apresentam como solução para os problemas de manejo extensivo de animais. Conforme os resultados preliminares apresentados apropriados mostraram, o impacto do cenário é significativo no indicador usado para estimar a distância, o que sugere que técnicas complementares para contornar a imprecisão causada devem ser desenvolvidas para tornar sistemas de localização baseados em RSSF uma alternativa viável e mais interessante do que os sistemas de rastreamento por RFID. Isto permitirá, por exemplo, a identificação de um animal no pasto, evitar o crime de

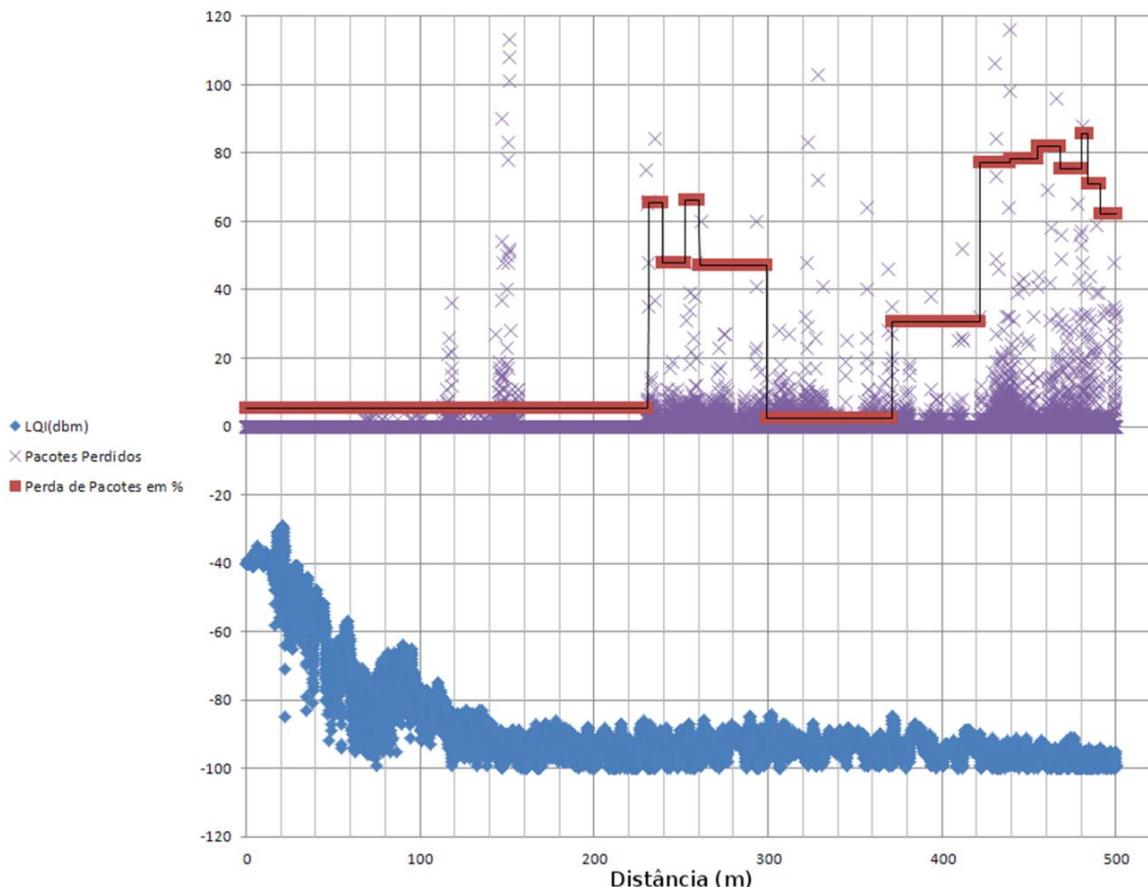


Figura 4. Gráfico mostrando a evolução do LQI (em azul) pela distância, em grená a evolução da perda de pacotes por teste e em roxo a perda de pacotes em dada distância.

abigeato e, com algoritmos apropriados, monitorar o rebanho por meio da verificação de sinais vitais e, em caso de enfermidade, será possível antecipar o tratamento necessário. Nos próximos passos serão aprofundados os testes de campo, aumentando o número de pacotes e criando conexões entre os rádios permitindo agrregar capacidade de alcance entre os dispositivos.

## V. AGRADECIMENTOS

Os autores gostariam de agradecer ao Prof. Érico Amaral, colaborador do Grupo Heco (<http://portearas.unipampa.edu.br/heco/>) e ao acadêmico Alessandro Saggiorato pelo apoio nos testes, ao DELET/UFRGS pela disponibilização dos nós sensores, à administradora rural Onilda Santos pelas informações dos cenários, à EMBRAPA Pecuária Sul pelas informações de caracterização da aplicação, à PROPESQ/UNIPAMPA e ao CNPQ que, respectivamente, por meio dos editais de apoio a grupos de pesquisa e pelo INCT NAMITEC, oferecem suporte financeiro a este projeto.

## REFERÊNCIAS

- [1] INCT Namitec. Instituto Nacional de Ciência e Tecnologia de Sistemas Micro e Nanoeletrônicos. Metodologia das Principais Linhas de Pesquisa e Ações. Disponível em: <<http://namitec.cti.gov.br>> Acesso em: 1 out 2011.
- [2] E. C. L. Chan, G. Baciu, S. C. Mak. Using Wi-Fi Signal Strength to Localize in Wireless Sensor Networks. IEEE Computer Society. International Conference on Communications and Mobile Computing 2009.
- [3] W. Wolf. High-Performance Embedded Computing: architectures, applications, and methodologies. Elsevier, San Francisco, 2007.
- [4] E. P. Freitas, I. Müller, C. E. Pereira, N. B. Perez, e L. B. Pinho. Integração Entre Dispositivos Móveis e Estáticos de Redes de Sensores Sem Fio para Monitoramento Animal Através de Algoritmo Bio Inspirado. CBA 2012. Campina Grande, Setembro/2012.
- [5] D. O. Cunha. Redes Sem Fio de Múltiplos Saltos: Protocolos Específicos para Aplicações e Roteamento com Suporte à Diversidade Cooperativa. Tese (Doutorado em Engenharia Elétrica), Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2008.
- [6] E. L. Souza, E. F. Nakamura, and H. A. B. F. Oliveira. On the Performance of Target Tracking Algorithms using Actual Localization Systems for Wireless Sensor Networks. MSWiM'09, Tenerife, October/2009
- [7] FREESCALE Semiconductor Technical Data Document Number: MC12x. Disponível em: <[http://www.freescale.com/webapp/wsps/site/prod\\_summary.jsp?co=MC13224V](http://www.freescale.com/webapp/wsps/site/prod_summary.jsp?co=MC13224V)>. Rev. 1.3 10/2010. Acesso em: 26 nov 2011.
- [8] XBee / RF Family Features Comparison. Disponível em: <[http://www.digi.com/pdf/chart\\_xbee\\_rf\\_features.pdf](http://www.digi.com/pdf/chart_xbee_rf_features.pdf)>. Acesso em: 29 ago 2012.
- [9] P. F. Motter. Sistema de Localização de Objetos Alvo Utilizando Rede de Sensores Sem Fio. 2010. Projeto de Diplomação (Engenharia Elétrica). Universidade Federal do Rio Grande do Sul, Porto Alegre. 2010.
- [10] J. Kurose, e K. Ross. Redes de Computadores e a Internet: uma abordagem top-down. 3.ed. São Paulo: Pearson Addison Wesley, 2006.

# Acesso Gratuito A Internet - Uma proposta de cadastro e autenticação para acesso à Internet em locais públicos

Marcelo de Borba  
UNISINOS  
celoborba@gmail.com

Rafael Bohrer Ávila  
UNISINOS  
rbavila@unisinos.br

**Resumo**—Iniciativas de prover uma estrutura de cidade digital e consequentemente, fornecer Internet gratuita à população vem se tornando uma característica comum de muitos gestores públicos no Brasil. Porém, em muitos projetos que foram ou estão sendo postos em operação, pode-se encontrar diversas falhas e vulnerabilidades, como por exemplo, falta de autenticação e criptografia. Estas características propiciam a ocorrência de graves problemas de segurança além de prover um cenário ideal para a prática de crimes na Internet. Diante deste cenário o presente artigo propõe uma solução segura para o fornecimento de Internet em locais públicos.

## I. INTRODUÇÃO

Atualmente as tecnologias de informação e comunicação estão convergindo para uma solução única, cujo objetivo principal é prover uma infraestrutura dotada dos mais diversos serviços à população. É cada vez mais comum o desenvolvimento de soluções públicas para acesso à Internet e demais serviços eletrônicos, entretanto, os modelos de concessão de acesso a Internet que estão sendo implantados em diversos municípios demonstram um elevado grau de despreparo das empresas, gestores e técnicos em relação à segurança da informação [1], [2].

A utilização de ferramentas e processos devidamente estruturados para estabelecer segurança às soluções de acesso gratuito muitas vezes são deixados de lado em virtude dos custos elevados e a complexidade para implantação e gerenciamento da solução. A falta de mão-de-obra qualificada em conjunto com a ausência de legislação específica para a área de tecnologia e Internet também contribuem para a ausência de segurança.

Em um cenário onde estão envolvidas pessoas das mais diversas classes econômicas, faixas etárias, serviços, além das etapas envolvidas no processo de acesso e utilização da Internet, é importante que a solução desenvolvida possa prover um cenário que abranja os requisitos básicos de segurança como autenticação e criptografia, e ao mesmo tempo, ofereça facilidade e transparência na sua utilização em virtude das características referentes ao conhecimento técnico da maioria dos usuários.

Considerando este contexto, o objetivo principal deste trabalho é propor uma solução segura, além de permitir um processo automatizado para entrada de novos usuários na rede, utilizando bancos de dados comuns a todas as prefeituras brasileiras como por exemplo, o banco de dados dos usuários do SUS - CADSUS e o banco de dados referentes ao cadastro de imóveis e proprietários - IPTU [3], [4].

Tabela I  
CONSOLIDADO GERAL - SOLUÇÕES DE ACESSO PÚBLICO À INTERNET

SOLUÇÕES DE ACESSO				
Cidade	Cripto.	Autent.	Cad.	T. Uso
Garibaldi-RS	NÃO	NÃO	NÃO	SIM
S. J. da Varginha-MG	NÃO	NÃO	NÃO	NÃO
Manoel Vitorino-BA	SIM	NÃO	NÃO	NÃO
Ribeirão Preto-SP	NÃO	SIM	PRÉVIO	NÃO
Alvorada-RS	NÃO	SIM	SIM	SIM
Santa Isabel-SP	NÃO	NÃO	NÃO	NÃO
Porto Alegre-RS	NÃO	NÃO	NÃO	NÃO
Seattle-EUA	NÃO	NÃO	NÃO	SIM
Atenas-Grécia	NÃO	NÃO	NÃO	NÃO

O artigo inicia por uma análise das soluções atualmente empregadas conforme apresentada na seção a seguir.

## II. SOLUÇÕES ATUAIS

Para o levantamento de soluções atualmente implantadas foi efetuado uma pesquisa por cenários semelhantes ao proposto neste artigo levando-se em consideração a utilização de criptografia, cadastramento, autenticação e termo de uso, conforme especificado na Tabela I . Os projetos apresentados a seguir foram selecionados em virtude da quantidade de informações disponíveis, visto que diversas prefeituras não possuem informações técnicas a respeito de seus projetos implementados.

As cidades de Garibaldi [5], São José da Varginha [6], Manoel Vitorino [7], Ribeirão Preto [8], Alvorada [9], Santa Isabel [10], Porto Alegre [11], Seattle [12] e Atenas [1] são utilizadas para comparação com a solução proposta por este artigo. Através dessa pesquisa é possível identificar particularidades que as diferem no quesito de segurança da rede.

A utilização de criptografia presente na solução do município de Manoel Vitorino possibilita uma camada adicional de proteção aos dados trafegados dificultando, por exemplo, a captura de dados através de ataques como o *parking lot* [13], porém, a ausência de cadastro e autenticação individual impossibilita a identificação dos usuários e consequentemente, a responsabilização em caso de incidentes de segurança.

É importante ressaltar que nesse contexto a utilização de criptografia se torna ineficaz à medida que a chave de acesso se torna pública. A utilização de um protocolo criptográfico como WEP e WPA na versão pessoal, implica em compartilhamento das informações de autenticação

para todos os usuários de um determinado ponto de acesso fazendo com que os mesmos sejam divulgados ou compartilhados em diversos meios, como por exemplo, em redes sociais.

A utilização de certificados digitais para prover confidencialidade pode ser implantada de forma satisfatória, porém, como solução de autenticação se torna inviável pois cada cidadão necessita possuir seu próprio certificado digital. A adoção desse modelo torna-se inviável a medida que o custo de implantação e manutenção tornam-se elevados, além de que é inapropriado impor a aquisição de um certificado digital à cada cidadão, ou até mesmo, a Prefeitura Municipal tornar-se uma autoridade certificadora provendo de forma gratuita certificados à toda população. Também se faz necessário ressaltar a dificuldade de utilização desse modelo para cidadãos sem conhecimentos técnicos ou especializados.

Nas soluções analisadas também se pode verificar que a ausência de autenticação é acompanhada da falta de cadastro dos usuários, conforme demonstrado na Tabela I.

Dos nove municípios analisados, somente dois apresentaram mecanismos de autenticação, o que demonstra a necessidade de um modelo adequado para acesso gratuito à Internet. Sem a presença de autenticação e um processo de cadastro confiável, é praticamente inviável efetuar a correta identificação dos usuários da rede e prover a auditoria dos acessos.

Os diferentes processos de cadastro de usuários nas soluções avaliadas possuem algumas características que comprometem a sua eficácia e a autenticidade dos dados informados, entre as quais se pode citar:

- 1) Falta de mecanismos para confirmação e validação dos dados enviados no cadastro;
- 2) Processos estatizados demandando alocação de recursos humanos para validação e análise de documentação;
- 3) Ausência ou geração inadequada de senhas para o primeiro acesso, comprometendo a confidencialidade das mesmas;
- 4) Utilização de email pessoal como meio para confirmação da autenticidade dos dados informados no cadastro.

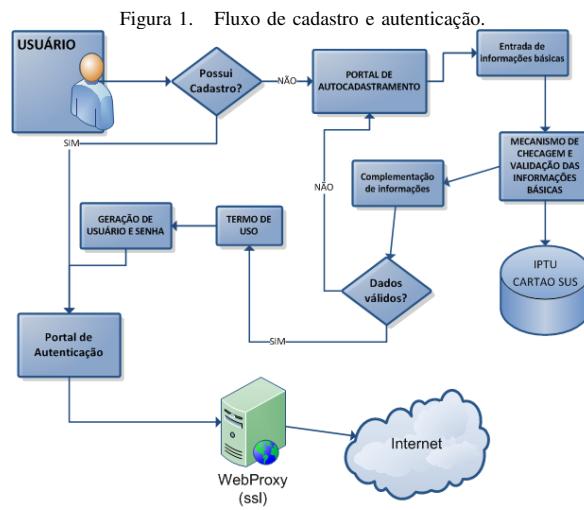
De maneira geral, é possível destacar mecanismos para formulação de uma solução adequada, mecanismos de cadastro, autenticação e criptografia, quando utilizados em conjunto são capazes de propiciar um nível de segurança adequado para acesso público a Internet.

Por fim observa-se que tanto o processo de cadastro e autenticação dos usuários quanto a navegação na rede devem ser protegidos, ou seja, as informações devem ser trafegadas sob canais seguros dotados de criptografia.

Com base no conteúdo exposto, a abordagem proposta apresenta uma solução que atende as demandas elencadas.

### III. PROPOSTA PARA IMPLEMENTAÇÃO DE CONCESSÃO DE ACESSO À INTERNET GRATUITA PARA A POPULAÇÃO

A partir do conjunto de fatores até então apresentados, este trabalho propõe uma solução que visa fornecer um



modelo técnico para implementação de acesso gratuito à Internet de forma segura.

Utilizando recursos adicionais que fazem parte da realidade de órgãos públicos municipais, como por exemplo, banco de dados existentes, é possível prover a sua aplicabilidade na maioria dos municípios brasileiros sem alterações significativas na solução a ser proposta.

Atualmente o governo federal mantém vários serviços que utilizam banco de dados nacionais como fonte de alimentação, porém, o acesso aos dados nem sempre é facilitado ou não possui acesso universalizado às prefeituras.

Assim, as bases de dados escolhidas para a solução são o cadastro de imóveis do município e o cadastro nacional de usuários do Sistema Único de Saúde - SUS, presentes na quase totalidade de municípios do território brasileiro. Tais fontes de dados possuem atualização constante, abrangência em todas as faixas etárias e geralmente possuem um processo de contato direto com o cidadão.

O escopo deste trabalho está delimitado ao processo de concessão de Internet gratuita em locais públicos como praças e escolas, envolvendo apenas as etapas do acesso do usuário à rede pública.

#### A. Fluxo de Cadastro e Autenticação

As bases de dados mencionadas podem ser utilizadas na construção de um mecanismo seguro e automatizado para cadastramento de usuários. Através de conexão segura utilizando o protocolo HTTPS, os usuários efetuam seu cadastramento ou autenticação, conforme fluxo demonstrado na Figura 1.

Ao efetuar a conexão de seu dispositivo ao PAP (Ponto de Acesso Público), o usuário é direcionado ao portal de Acesso Gratuito, este portal possui as opções de autenticação e cadastramento, cabendo ao usuário a escolha apropriada.

Ao se tratar de um novo usuário, o mesmo deve selecionar a opção de autocadastramento, conforme demonstrado na Figura 1, o usuário será então redirecionado para uma

nova página de cadastro onde deverá preencher as informações básicas para início do processo de cadastramento.

As informações básicas solicitadas podem ser personalizadas de acordo com cada projeto ou ponto de acesso, sendo obrigatório o fornecimento de no mínimo, o nome completo e um documento de identificação como RG ou CPF.

A partir do momento do envio destas informações à aplicação, entra em funcionamento o mecanismo de busca e validação dos dados, este mecanismo efetua buscas nas bases de dados cadastrados na solução e seleciona os registros cujos dados informados pelo usuário sejam idênticos aos encontrados pelo mecanismo de busca.

Após efetuar a busca e encontrar ao menos um cadastro válido, a aplicação apresenta ao usuário um questionamento referente aos dados relativos ao cadastro encontrado cabendo ao usuário respondê-las e enviá-las novamente para a aplicação.

Os dados selecionados podem variar conforme a base de dados selecionado, quando utilizado a base de dados do CADSUS pode-se utilizar os campos relativos ao nome da mãe, número do cartão do sus e parte do endereço. Quando utilizada a base de dados referente aos imóveis (IPTU) é possível utilizar dados como o numero de cadastro do imóvel, endereço ou o próprio código de ativação enviado anteriormente na emissão do tributo ao cidadão.

Ao receber as informações complementares informadas pelo usuário o mecanismo efetua a validação das mesmas mediante comparação com os dados armazenados na aplicação. Se o processo de validação ocorrer com êxito é então apresentado os termos de uso da rede, mediante a aceitação do usuário a aplicação gera as credenciais de acesso apresentando-as na tela do usuário. Após confirmar o recebimento das informações o usuário é redirecionado para o portal de autenticação da solução.

Nos casos onde o mecanismo de validação não consegue encontrar dados suficientes, ou quando as perguntas não são respondidas adequadamente o sistema deve abortar as operações seguintes orientando o usuário a reiniciar o processo de cadastramento, ou então, efetuar o cadastro de forma presencial junto ao órgão responsável pela rede de acesso no município.

Após a complementação de informações e a validação dos dados informados, o usuário é redirecionado ao portal de autenticação onde poderá efetuar o seu login para acesso à Internet.

Para a proteção do tráfego dos dados, a solução utiliza uma aplicação de *webproxy* disponível sobre o protocolo SSL. Através de configurações específicas no *firewall* somente o tráfego através deste serviço é permitido, fornecendo assim, uma camada de proteção contra *sniffing* de pacotes ou ataques do tipo *parking lot* [13], [14].

A solução de auto-cadastro e validação deve ser desenvolvida de acordo com cada solução utilizada para o gerenciamento das bases de dados especificadas anteriormente, em virtude das diversas aplicações utilizadas para gerenciamento desse tipo de dados é preciso adaptar os mecanismos de consulta nas bases de dados.

Figura 2. Portal de Autenticação.



#### IV. PROTÓTIPO DA SOLUÇÃO

Para implementação da solução descrita na seção anterior são utilizadas soluções existentes no mercado: a distribuição Pfsense [15], baseada no sistema operacional FreeBSD [16] e o aplicativo Glype [17], responsável pelo serviço de webproxy. A escolha das ferramentas para apresentação do protótipo levam em consideração a facilidade de implementação, bem como, o alto grau de documentação de suporte disponíveis nos sites respectivos de cada solução.

##### A. Portal de Acesso

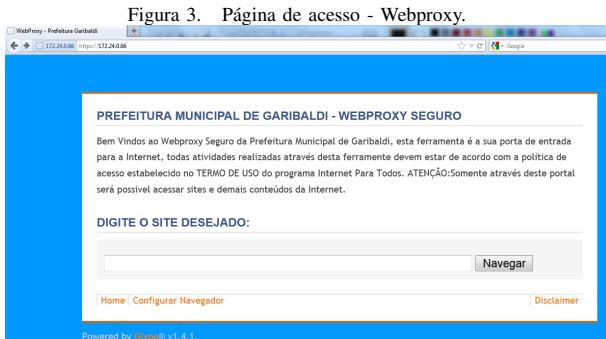
Desenvolvido utilizando-se a solução Pfsense o portal tem por objetivo prover a autenticação dos usuários, a Figura 2 demonstra o portal de autenticação que será utilizado na implementação da solução. Através da utilização da própria ferramenta Pfsense foi desenvolvido um portal para interface do usuário com o mecanismo de autenticação da rede suportando autenticação em base local, *Radius* ou até mesmo, integração com o serviço *Active Directory da Microsoft* [18].

Após a autenticação ser efetuada com sucesso o portal de autenticação redireciona o tráfego do cliente autenticado para o proxy seguro da rede, o qual será o ponto de navegação para a Internet.

##### B. Webproxy

A solução utilizada para prover um sistema de *webproxy* operando exclusivamente pela porta 443 utilizando protocolo SSL [19] foi desenvolvida utilizando um servidor Apache em conjunto com a ferramenta *Glype*. Toda navegação via protocolo HTTP ou HTTPS é redirecionada para o webproxy em questão. Os esforços na construção do solução levaram em conta a transparência e a facilidade de utilização, visto que o usuário não precisa efetuar quaisquer alterações em seu navegador bastante apenas, digitar os endereços desejados no formulário disponível no próprio *webproxy*, conforme demonstrado na Figura 3.

Durante a navegação o usuário pode visualizar, na parte superior do navegador, uma barra de ferramentas disponibilizada para permitir a mudança de site sem ter que retornar à página principal do webproxy, conforme demonstrado na Figura 4.



A aplicação pode ser configurada para efetuar o log de todas as conexões efetuadas, fornecendo uma alternativa para efetuar a auditoria dos acessos. Em conjunto com os dados da autenticação é possível identificar todo o tráfego oriundo de um mesmo usuário, os logs armazenados apresentam as informações referentes ao IP de origem, data e hora do acesso e o endereço acessado na Internet.

## V. CONCLUSÕES FINAIS

Prover um meio alternativo, dotado de requisitos de segurança com baixo custo e com elevado nível de compatibilidade com a infraestrutura de redes públicas existentes é uma das principais premissas da solução apresentada, a próxima etapa deste trabalho consiste na aplicação do mecanismo de auto-cadastramento, os detalhes de funcionamento estão sendo ajustados e a integração com base de dados semelhantes as do Cartão do SUS e IPTU do município de Garibaldi já estão em fase de testes.

A união das ferramentas propostas num único ambiente em conjunto com os requisitos de cadastramento seguro, autenticação e auditoria serão implementadas ao longo do segundo semestre de 2012 como projeto piloto no município de Garibaldi em duas das principais praças da cidade.

## REFERÊNCIAS

- [1] D. Ztoupis, K. Zarifis, I. Stavrakakis, and C. Xenakis, "Towards a security framework for an established autonomous network," in *Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium on*, may 2008, pp. 749 –754.
- [2] G. Camponovo and A. Picco-Schwendener, "Motivations of hybrid wireless community participants: A qualitative analysis of swiss fon members," in *Mobile Business (ICMB), 2011 Tenth International Conference on*, june 2011, pp. 253 –262.
- [3] CARTAONET. (2012) Portal de cadastros nacionais. Disponível em: <http://cartaonet.datasus.gov.br>. Acesso em 15 de abril de 2012.
- [4] Brasil, "Lei 10.257 de 10 de julho de 2001. regulamenta os artigos 182 e 183 da constituição federal, estabelece diretrizes gerais da política urbana e dá outras providencias." Brasília, DF, 2001.
- [5] P. M. de Garibaldi. (2011) Internet para todos. Garibaldi - RS. Disponível em: <http://internetparatodos.garibaldi.rs.gov.br>. Acesso em 23 dezembro 2012.
- [6] P. M. de São José da Varginha. (2012) Internet grátis na praça são josé. São José da Varginha - MG. Disponível em: <http://www.saojosedavarginha.mg.gov.br/destaques/internet-gratis-na-praca-sao-jose>. Acesso em 22 de Abril de 2012.
- [7] P. M. de Manoel Vitorino, "Site institucional," Manoel Vitorino - BA., 2012, disponível em: <http://manoelvitorino.com>. Acesso em 22 de Abril de 2012.
- [8] P. M. de Ribeirão Preto. (2012) Ribeirão digital. Ribeirão Preto - SP. Disponível em: <http://www.ribeirao-preto.sp.gov.br/cidadao/i99rdigital.php>. Acesso em 20 de abril de 2012. Acesso em 22 de Abril de 2012.
- [9] P. M. de Alvorada, "Atendenet," 2012, disponível em: <http://177.43.243.105/atendenet/>. Acesso em 05 de maio de 2012.
- [10] P. M. de Santa Isabel. (2012) Site institucional. Santa Isabel - SP. Disponível em: <http://www.santa-isabel.sp.gov.br/internet>. Acesso em 22 de Abril de 2012.
- [11] PROCEMPA, "Procompa livre e gratuita," 2012, disponível em: [http://www.procompa.com.br/default.php?p\\_secao=76](http://www.procompa.com.br/default.php?p_secao=76). Acesso em 19 de maio de 2012.
- [12] G. Seattle, "Wifi in seattle," Seattle - EUA., 2012, disponível em: <http://www.seattle.gov/html/citizen/wifi.htm>. Acesso em 22 de Abril de 2012.
- [13] H. K. INFOSEC., "Wireless network security," 2010, disponível em: <http://www.infosec.gov.hk/english/technical/files/wireless.pdf>. Acesso em: 1 abril 2012.
- [14] M. Mallick, *Mobile and Wireless Design Essentials*. New York, NY, USA: John Wiley & Sons, Inc., 2003.
- [15] B. P. LLC, "pfSense open source firewall," 2012, disponível em: <http://www.pfsense.org/>. Acesso em: 12 julho 2012.
- [16] FreeBSD, "The power to server," 2012, disponível em: <http://www.freebsd.org/>. Acesso em: 12 junho 2012.
- [17] Glype, "Glype proxy script," 2012, disponível em: <http://www.glype.com/>. Acesso em: 12 junho 2012.
- [18] Microsoft, "Active directory overview," 2012, disponível em: <http://www.microsoft.com/en-us/server-cloud/windows-server/active-directory-overview.aspx>. Acesso em: 20 julho 2012.
- [19] C. Peikari and A. Chuvakin, *Security warrior - know your enemy*. O'Reilly, 2004.

---

III

## **Segurança e Redes sem Fio**

---



# Proposta e Implementação de um Firewall para Aplicações Web Denominado UniscanWAF

Alfredo Del Fabro Neto, Rogério Turchetti,

Celio Trois, Walter Priesnitz Filho

Universidade Federal de Santa Maria - CTISM/UFSM

{alfredodfn, turchetti, celio.trois,

walter}@redes.ufsm.br

Douglas Rocha, Diego Kreutz

Grupo de Pesq. em Sist. de Informação – GPSI

Universidade Federal do Pampa – UNIPAMPA

douglas.poerschke@gmail.com,

kreutz@unipampa.edu.br

**Resumo**—Os sistemas Web predominam na maioria dos segmentos da sociedade. Ao mesmo tempo que cresce o número de aplicações e recursos online, aumentam também as preocupações relacionadas à segurança da informação. Os sistemas Web são os maiores alvos dos atacantes. Buscando contribuir com a segurança das aplicações Web, este trabalho apresenta uma ferramenta, denominada UniscanWAF, que tem por objetivo monitorar em tempo real os acessos a sistemas Web. Sua principal finalidade é detectar, interceptar e bloquear tentativas de ataque às aplicações online. Para isso, a ferramenta utiliza mecanismos e recursos de detecção do Uniscan, tornando possível a interceptação de ataques que visem explorar vulnerabilidades como XSS, LFI, RFI, RCE e SQL-Injection.

## I. INTRODUÇÃO

A maioria das vulnerabilidades encontradas em aplicações Web são derivadas de uma fraca implementação do ciclo de desenvolvimento de software (SDLC - *Software Development Life Cycle*) [1]. Além disso, sistemas Web são alvos constantes de ataques. Um filtro de pacotes convencional, que trabalha nas camadas de redes e transporte, não detecta ou evita ataques que exploram vulnerabilidades em nível de aplicação. Uma das soluções existentes para o nível de aplicação são os *Web Application Firewall* (WAF) [2]. Um exemplo de WAF é o *ModSecurity* [3]. Este foi projetado para atuar com os servidores Web Apache.

No intuito de minimizar o efeito e as ações de ataques a sistemas Web, este trabalho descreve e avalia o UniscanWAF, uma ferramenta capaz de detectar comportamentos anômalos e interceptar, com base em regras pré-definidas, ações maliciosas contra sistemas Web. A ferramenta proposta representa uma extensão ao Uniscan [4], o qual é um scanner convencional de vulnerabilidades em sistemas Web. O Uniscan fornece um diagnóstico do sistema testado, identificando todas as páginas ou URLs problemáticas do sistema. Ele está disponível no SourceForge [5] e na distribuição *Linux BackTrack*.

O UniscanWAF trabalha com as mesmas regras (*plugins* de detecção de vulnerabilidades) do Uniscan. A principal diferença reside no fato de a ferramenta proposta atuar de forma similar a um *proxy* com filtro de pacotes, ou seja, é capaz de analisar e interceptar ações maliciosas em tempo de execução, protegendo a aplicação Web contra ataques conhecidos (regras). Essa característica é especialmente interessante para proteger sistemas Web que estão em

constante desenvolvimento ou evolução. Segundo estudos, muitas aplicações Web entram nessa classificação [6].

A ferramenta proposta encontra-se em fase de desenvolvimento. Neste sentido, o principal objetivo deste trabalho é verificar sua viabilidade de utilização, avaliando as funcionalidades de análise e interceptação de requisições maliciosas destinadas à aplicações Web.

O restante deste trabalho está estruturado da seguinte forma: a seção II apresenta a arquitetura geral do UniscanWAF. Na seção III são apresentadas as vulnerabilidades detectadas pela ferramenta, bem como códigos vulneráveis utilizados no cenário de testes para cada uma das vulnerabilidades. A seção IV apresenta os testes realizados para validar as funcionalidades do UniscanWAF. Por fim, a seção V apresenta as considerações finais.

## II. ARQUITETURA DO UNISCANWAF

A arquitetura do UniscanWAF foi dividida em dois módulos distintos: o módulo de Inicialização e o módulo de Detecção de Vulnerabilidades. O módulo de Inicialização compreende a inicialização da ferramenta de acordo com os parâmetros configurados no arquivo de configuração. Algumas das possibilidades de configuração são: a ação tomada pelo módulo de Detecção de Vulnerabilidades no momento da detecção, que pode ser configurado para encerrar a conexão ou executar um *script*, por exemplo, gerar um relatório técnico, colocar o host de origem em quarentena, ou mesmo obter maiores informações com a execução de outras ferramentas que possibilite um diagnóstico detalhado. Outros exemplos de parâmetros de configuração incluem a localização do arquivo de *log* e a porta que o UniscanWAF utilizará para comunicação.

O módulo de Detecção de Vulnerabilidades é responsável por realizar a análise de todas as requisições dos clientes que tenham como destino o servidor Web a ser protegido. Para cada regra há um arquivo correspondente, que determina as ações a serem executadas pela ferramenta para detectar a existência ou não de uma vulnerabilidade na requisição que está sendo analisada. Com isso, novas regras podem ser adicionadas ao sistema sob-demanda, conferindo flexibilidade e simplicidade ao sistema. Na versão atual, o UniscanWAF é capaz de analisar e processar a maioria das vulnerabilidades implementadas e detectadas pelo Uniscan, incluindo XSS (*Cross-site Scripting*), LFI (*Local File Include*), RFI (*Remote File Include*), RCE

(*Remote Command Execution*) e *SQL-Injection*. A figura 1 ilustra a arquitetura do UniscanWAF.

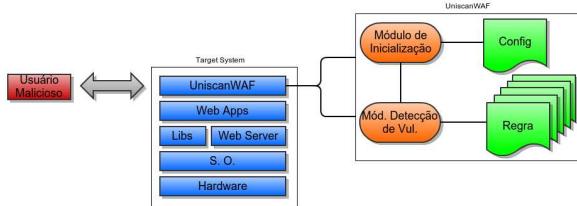


Figura 1. Arquitetura do UniscanWAF

O UniscanWAF atua como um intermediário, similar a um proxy, entre o usuário e o servidor Web. Sendo assim, ele é capaz de analisar e repassar ou não as requisições para a aplicação Web, conforme as regras habilitadas para detecção.

### III. VULNERABILIDADES WEB DETECTADAS PELO UNISCANWAF

Para validar a eficiência do detector de vulnerabilidades proposto, foi criada uma aplicação teste contendo as cinco vulnerabilidades reconhecidas pela ferramenta. Cada vulnerabilidade, acompanhada de um exemplo de código vulnerável, é descrita a seguir.

#### A. Cross-site Scripting

A vulnerabilidade XSS ocorre quando uma aplicação inclui dados fornecidos pelo usuário sem a validação desses dados [7]. O impacto desse tipo de ataque inclui o eventual roubo de sessão. Aplicações que não tratam os dados de entrada adequadamente estão sujeitas a esse tipo de ataque. Esse tipo de vulnerabilidade passa, com uma certa frequência, despercebido pelos desenvolvedores. De maneira a amenizar esse problema, estratégias de segurança presentes no SDLC são recomendadas [8].

```
<?php
print "Pagina: " . urldecode($_SERVER["REQUEST_URI"]);
?>
```

Figura 2. Código-fonte PHP vulnerável a XSS.

Na figura 2 é apresentado um exemplo de código-fonte PHP vulnerável ao ataque XSS. No caso em questão, o trecho de código imprime o recurso que foi requisitado. Consequentemente, ao requisitar uma URL como "<http://sistemaWeb.com.br/index.php?name=teste>", será exibido "Pagina: /index.php?name=teste". Entretanto, há outras possibilidades de ações, como no caso de uma URL similar a "[http://sistemaWeb.com.br/index.php?name=<script>alert\('XSS'\)</script>](http://sistemaWeb.com.br/index.php?name=<script>alert('XSS')</script>)", que, no caso, irá criar uma caixa com o conteúdo "XSS" e um botão "OK". Isso significa que um atacante consegue executar ações e códigos diversos, tornando o ataque potencialmente comprometedor para o sistema Web, ou ambiente alvo, como o sistema hospedeiro do servidor Web.

#### B. Local File Include

A vulnerabilidade LFI torna possível a inclusão de arquivos locais. Este ataque pode expor arquivos e dados dos sistemas hospedeiros da aplicação Web.

```
<?php
$arquivo = $_GET['arg'];
include($arquivo);
?>
```

Figura 3. Código-fonte PHP vulnerável a LFI e RFI.

Na figura 3 é ilustrado um exemplo de código-fonte PHP vulnerável ao ataque LFI. Pode ser observado que o código recebe um parâmetro de nome "arg" via método GET. O valor é atribuído à variável "\$arquivo", que, logo após, é incluída no código sem nenhuma verificação. Desse modo, um usuário poderia requisitar uma URL como "</arquivo.php?arg=/etc/passwd>". Neste caso, o arquivo "</etc/passwd>" será exibido para o usuário.

#### C. Remote File Include

Esta vulnerabilidade é similar a LFI. A principal diferença reside no fato de incluir um arquivo hospedado em outro servidor Web ao invés do sistema local. Isso representa um risco de segurança maior, pois o arquivo remoto pode ter sido intencionalmente criado e preparado pelo atacante, aumentando suas possibilidades de ações maliciosas de forma relativamente simples e prática.

A vulnerabilidade RFI ocorre devido a falta de validação e tratamento correto dos dados de entrada. O código ilustrado na figura 3 permite a inclusão remota de arquivos. Um exemplo de URL maliciosa poderia ser "</arquivo.php?arg=http://servidorWebDoAtacante.com.br/comandos.txt>". Este arquivo pode conter qualquer sequência de comandos e construções reconhecidas pela linguagem PHP e pelo sistema hospedeiro da aplicação alvo. Consequentemente, uma vulnerabilidade RFI é crítica e permite ao atacante realizar diferentes tipos de ações no sistema alvo.

#### D. Remote Command Execution

A vulnerabilidade de RCE permite a um atacante executar remotamente comandos no sistema alvo. Num código PHP, através do comando `system()` sem o devido tratamento dos dados inseridos pelo usuário, é possível executar comandos do sistema operacional.

```
<?php
$user = $_GET['arg'];
system("echo $user >> users_forum.txt");
?>
```

Figura 4. Código-fonte PHP vulnerável a RCE.

A figura 4 apresenta um exemplo de código-fonte PHP vulnerável. O objetivo do código é inserir usuários no final do arquivo "users\_forum.txt", como "<http://rce.php?arg=user4>". Neste caso, apenas o usuário "user4" será acrescentado no arquivo indicado. Porém, um atacante pode ir muito além. A chamada de

sistema `system()` invoca um shell para a execução do comando. Como consequência, qualquer composição de comandos possível em um shell, utilizando o separador " ; ", pode ser invocada pelo atacante. Este, por exemplo, poderia requisitar a URL "/rce.php?arg=user4;cat /etc/passwd; ", o que provocaria a inclusão do "user4" no arquivo "users\_forum.txt", bem como a leitura do conteúdo do arquivo "/etc/passwd".

O risco de segurança aumenta em situações onde os servidores Web são executados em modo *root*. Nesses casos, os sistemas hospedeiros podem ser facilmente comprometidos por um atacante, extendendo as implicações de segurança para além da aplicação Web vulnerável.

#### E. SQL Injection

*SQL Injection* pode ser definida como a inserção de código SQL (*Structured Query Language*) malicioso através de dados de entrada de uma aplicação. Se bem sucedido, esse tipo de ataque pode obter acesso aos dados do banco de dados, modificar esses dados, eventualmente executar operações de administrador de banco de dados, e, em alguns casos, executar comandos do sistema operacional [9]. O tratamento desses dados, por parte de programadores, pode evitar esse tipo de ataque.

```
<?php
$par1 = $_GET['username'];
$par2 = $_GET['password'];
$query = "SELECT * FROM usuario WHERE username= '$par1' AND password= '$par2'";
?>
```

Figura 5. Código-fonte PHP vulnerável a *SQL Injection*.

A figura 3 apresenta um exemplo de código PHP vulnerável a *SQL Injection*. No exemplo, caso a URL de requisição for "/show.php?username=user&password=12345", serão apresentadas as informações do usuário user, com senha 12345, cadastradas no banco de dados. Entretanto, se a URL apresentar o formato "/show.php?username=user&password='1' OR '1'", serão listadas informações de todos os usuários presentes no banco de dados da aplicação Web. Cabe observar que os parâmetros `username` e `password` não sofrem nenhum tipo de tratamento e são inseridos diretamente na pesquisa (`query`) SQL. Deste forma, a operação "1' OR '1" é sempre verdadeira, retornando os dados de todos os usuários registrados.

## IV. TESTES EFETUADOS

O objetivo é apresentar os primeiros resultados de avaliação da funcionalidade de detecção de vulnerabilidades do UniscanWAF. E, também, comparar os resultados com o *ModSecurity* [3], um WAF *open source* projetado para trabalhar junto a servidores Web Apache.

Os testes foram divididos em duas baterias. Na primeira etapa (bateria I) foram realizados testes individuais, ou seja, com recursos de detecção, das ferramentas *ModSecurity* e UniscanWAF, especificamente habilitados para cada uma das vulnerabilidades. Na segunda etapa (bateria II),

foram realizados testes de detecção com todos os recursos habilitados em ambas as ferramentas.

As vulnerabilidades *XSS*, *LFI*, *RFI*, *RCE* e *SQL Injection* foram implementadas em PHP. Foram utilizados os exemplos apresentados na seção III. A aplicação com os códigos vulneráveis foi hospedada em uma máquina virtual com sistema operacional *Linux Ubuntu 10.04.4 LTS*, servidor Web *Apache* versão 2.2.14, PHP versão 5.3.2-1ubuntu4.17, *Perl* versão 5.10.1, *ModSecurity Stable* versão 2.2.6 (com as regras de detecção *Core Rules* versão 2.2.5) e o UniscanWAF.

Para permitir a vulnerabilidade *RFI*, foram ativadas as seguintes variáveis de configuração do PHP: `register_globals`, `allow_url_fopen` e `allow_url_include`. Uma vez que a comunicação e integração entre sistemas é algo necessário e cada vez mais explorada por diferentes organizações, a ativação desses parâmetros é cada vez mais comum [10]. Questões de escalabilidade, como desempenho e tempo de resposta ao usuário, não foram levadas em consideração neste trabalho.

Em um primeiro momento, realizou-se testes no *ModSecurity*, e em seguida, no UniscanWAF. Os detalhes desse processo são descritos a seguir.

#### A. ModSecurity

Os resultados da primeira bateria de testes estão sumarizados na tabela I. A coluna Regra *ModSecurity* refere-se ao arquivo que habilita a regra, a coluna Vulnerabilidade refere-se a vulnerabilidade relacionada com o arquivo e a coluna Status indica se a vulnerabilidade foi ou não encontrada. Nota-se que o *ModSecurity* conseguiu detectar apenas duas das cinco vulnerabilidades, indicando que as regras referentes às vulnerabilidades não detectadas encontram-se deficientemente implementadas (para os arquivos referenciados pela coluna Regra *ModSecurity* da tabela I), ou não implementadas, no caso da vulnerabilidade *RCE*. A empresa *Trustwave* [11] apresenta regras adicionais que contemplam o ataque *RCE*, mas que devem ser adquiridas comercialmente.

Tabela I TESTES INDIVIDUAIS DE VULNERABILIDADES			
Regra	ModSecurity	Vulnerabilidade	Status
modsecurity_crs_41_xss_attacks.conf		XSS	✓
modsecurity_crs_46_slr_et_lfi_attacks.conf		LFI	
modsecurity_crs_46_slr_et_rfI_attacks.conf		RFI	
inexistente <sup>1</sup>		RCE	
modsecurity_crs_41_sql_injection_attacks.conf		SQL-Injection	✓

A segunda bateria de testes foi realizada com todas as regras habilitadas no *ModSecurity*. O resultado dos testes pode ser visualizado na tabela II. Pode-se observar que a ferramenta detectou mais vulnerabilidades com todas as regras habilitadas. De fato, o arquivo

<sup>1</sup>Arquivo não encontrado para esta vulnerabilidade.

"modsecurity\_crs\_40\_generic\_attacks.conf" possibilita a detecção dos ataques de *LFI* e *RFI*, anteriormente não detectados. Este arquivo concentra vários ataques Web, entretanto, na documentação não fica claro quais tipos de ataques o arquivo em questão trata.

Mesmo assim, observou-se um comportamento diferente para diferentes entradas de parâmetros na URL para a vulnerabilidade *RFI*. Quando o formato da URL é similar a este "http://sistemaWeb.com.br/arquivo.php?a\_rg=http://10.1.1.1/index.html", ou seja, com um endereço IP como parâmetro, o *ModSecurity* identifica a tentativa de exploração da vulnerabilidade. No entanto, quando o formato da URL é similar ao seguinte "http://sistemaWeb.com.br/arquivo.php?a\_rg=http://www.dominiovalido.br/index.html", ou seja, sem um endereço IP como parâmetro, mas com uma URL válida, o *ModSecurity* não detecta a tentativa e exploração *RFI* do atacante. Entramos em contato com o suporte do *ModSecurity* para descobrir o motivo dos resultados, mas não obtivemos resposta até o momento.

### B. UniscanWAF

Para o UniscanWAF foi utilizada a mesma metodologia de testes do (*ModSecurity*). Os resultados obtidos foram os mesmos nas duas baterias de testes e podem ser observados na tabela II.

### C. Resultados

A tabela II apresenta o resultado dos testes efetuados com o UniscanWAF e o *ModSecurity* (com todas as regras habilitadas) no cenário proposto. Como pode ser observado, o UniscanWAF conseguiu detectar todas as vulnerabilidades apresentadas, enquanto que o *ModSecurity* não detectou a vulnerabilidade *RCE* e teve problemas com a vulnerabilidade *RFI*, conforme descrito anteriormente.

Tabela II  
COMPARAÇÃO ENTRE *ModSecurity* E UNISCANWAF

Ferramenta	XSS	LFI	RFI	RCE	SQL-Injection
UniscanWAF	✓	✓	✓	✓	✓
<i>ModSecurity</i>	✓	✓	✓ <sup>2</sup>		✓

Os resultados apresentados permitem concluir que o UniscanWAF é capaz de detectar em tempo real requisições que visam explorar vulnerabilidades das aplicações Web. Além disso, detecta mais vulnerabilidades que ferramentas similares, como é o caso do *ModSecurity* e possui uma arquitetura modular e flexível idêntica a do Uniscan, um scanner de vulnerabilidades *open source* com mais de 7.000 downloads de mais de 130 países [5].

### V. CONSIDERAÇÕES FINAIS

Este artigo apresentou o UniscanWAF, uma ferramenta para detectar e interceptar em tempo real a exploração de vulnerabilidades, por parte dos atacantes, endereçadas à aplicações Web. Os resultados mostram a funcionalidade

do mecanismo mais essencial da ferramenta, a detecção das vulnerabilidades. Além disso, foi estabelecida uma breve comparação com o *ModSecurity*, demonstrando que a solução proposta é capaz de detectar mais vulnerabilidades, dentro do conjunto analisado. Complementarmente, o UniscanWAF foi concebido como uma extensão do Uniscan, dentro dos conceitos de flexibilidade, simplicidade e extensibilidade desta ferramenta, um scanner de vulnerabilidades de sistemas Web largamente utilizado.

Entre os próximos passos podem ser incluídos a criação de *white list*, ou seja, desenvolver uma interface que obrigue o programador a validar as atualizações realizadas na aplicação, para que então as novas funcionalidades sejam aplicadas e utilizadas. Esta tarefa faz com que o ciclo de testes de segurança, mesmo com a aplicação já em uso, seja sempre executado. Por último, testes de desempenho em um ambiente de produção serão realizados para avaliar a ferramenta proposta.

### REFERÊNCIAS

- [1] N. Teodoro and C. Serrao, "Web application security: Improving critical web-based applications quality through in-depth security analysis," in *International Conference on Information Society (i-Society)*, 2011, pp. 457–462.
- [2] H. Takahashi, H. F. Ahmad, and K. Mori, "Application for autonomous decentralized multi layer cache system to web application firewall," in *10th International Symposium on Autonomous Decentralized Systems (ISADS)*, vol. 1, 2011, pp. 113–120.
- [3] (2012, Jul.) Modsecurity: Open source web application firewall. [Online]. Available: <http://www.modsecurity.org/>
- [4] D. Rocha, D. Kreutz, and R. Turchetti, "Uma ferramenta livre e extensível para detecção de vulnerabilidades em sistemas web," in *Actas de la 7a Conferencia Ibérica de Sistemas y Tecnologías de Información*, 2011, pp. 747–752.
- [5] D. Rocha. (2012) Uniscan - um scanner de vulnerabilidades para sistemas web. [Online]. Available: <http://uniscan.sourceforge.net/>
- [6] T. O'reilly, "What is web 2.0: Design patterns and business models for the next generation of software," in *Communications & Strategies*, vol. 1, 2007, pp. 17–37.
- [7] (2012, Jul.) The open web application security project. [Online]. Available: [https://www.owasp.org/index.php/Top\\_10\\_2010-A2](https://www.owasp.org/index.php/Top_10_2010-A2)
- [8] B. Chess and G. McGraw, "Static analysis for security," *IEEE SECURITY & PRIVACY*, vol. 2, no. 6, pp. 76–79, Nov-Dec 2004.
- [9] (2012, Jul.) The open web application security project. [Online]. Available: [https://www.owasp.org/index.php/SQL\\_Injection](https://www.owasp.org/index.php/SQL_Injection)
- [10] D. Rocha, "Uniscan: Um scanner de vulnerabilidades para sistemas web," in *Trabalho de Conclusão de Curso apresentado à Universidade Federal do Pampa (UNIPAMPA)*, 2012.
- [11] (2012, Jul.) Trustwave. [Online]. Available: <https://www.trustwave.com/>

<sup>2</sup>Detectou parcialmente a vulnerabilidade.

# Uma arquitetura para desenvolvimento de dispositivos de autenticação e acesso a espaços físicos

Jeann C. M. Raguzzoni, Lamarck Ribas Heinsch  
UFSM  
jraguzzoni@inf.ufsm.br, lamarck@inf.ufsm.br

Tiago Antônio Rizzetti  
UFSM - CTISM  
rizzetti@ctism.ufsm.br

## II. TRABALHOS E SISTEMAS RELACIONADOS

**Resumo**—Ambientes de trabalho e estudo estão cada vez melhor equipados e sofisticados, sendo necessário um rastreamento temporal sobre acessos realizados a estes ambientes.. Esta arquitetura se propõe a resolver este problema através da implementação de um controle de acesso eletrônico com diretrivas e bases de acesso controladas por um sistema de alto nível, baseado em redes TCP/IP, que facilite aos administradores do local gerenciar o fluxo de pessoas autorizadas.

## I. INTRODUÇÃO

Grandes instituições geram um grande fluxo de pessoas, um fluxo difícil de ser controlado. Na tentativa de conter acessos utiliza-se bloqueios a certas áreas. Em uma faculdade, por exemplo, temos incontáveis salas de aula, laboratórios, salas de professores, etc. Estas salas são, na maioria das vezes, mantidas fechadas por chave, a qual é retirada na portaria, ou outro setor especializado, e depois devolvida, muitas vezes sem passar por uma verificação do usuário ou registro de uso. Este é um meio de controle precário, inefficiente e inseguro.

Como uma proposta para solução deste problema, criou-se a arquitetura do projeto ESC (Environment Security Control). Este, consiste em uma arquitetura de controle de acesso diferenciada, focada em uma abordagem que possibilite a comunicação entre os dispositivos de autenticação, junto ao acesso do espaço físico, e uma aplicação gerente, podendo desta forma ampliar a capacidade de configuração e coerência do sistema. Em função desta abordagem, aumenta-se de forma expressiva os aspectos referentes a escalabilidade, flexibilidade e auditoria. Dividiu-se este sistema sobre dois aspectos principais, um considerando o gerente, o qual é responsável pelo tipo de política empregada para autenticação e todo o ambiente de integração com os demais sistemas utilizados, como o LDAP; o segundo subsistema, no qual se encontra o foco deste trabalho, projeta e implementa uma interface de hardware e software capaz de lidar com os dispositivos físicos de autenticação. Obtendo seus dados e, através de um protocolo criado, comunicando-se com o gerente de forma bidirecional. Desta forma possibilitando além de sensoriamento do ambiente, a capacidade de realizar ações sobre o sistema através de atuadores.

O restante deste trabalho terá suas informações focadas no método e protocolo de comunicação entre o dispositivo de hardware remoto e o sistema gerente de alto nível, os dois subsistemas da arquitetura apresentada a seguir.

Os trabalhos relacionados foram separados em duas categorias, soluções proprietárias e soluções acadêmicas. As soluções proprietárias consistem em produtos desenvolvidos por empresas de médio ou grande porte, geralmente especializadas na área de segurança. Essas empresas investem pesado em pesquisa e desenvolvimento de modo a oferecer soluções robustas. A vantagem de utilizar uma solução dessas é o fato de haver maior compatibilidade entre equipamento provido pela empresa e o sistema gerenciador de permissões de acesso. Porém existem desvantagens, como por exemplo, o produto ser vendido em kits, isto é, preços definidos de acordo com a quantidade de dispositivos tornando o custo de aquisição e manutenção alto. Outro problema comum é o código-fonte ser fechado, impossibilitando a correção de erros no código, implementação de novas funcionalidades e adaptação a necessidades do cliente

Dentre as soluções proprietárias, foram analisadas duas empresas, Nibtec [10] e ID Tech [9]. Estas empresas diferem em tipos de dispositivos de acordo com o método de entrada (login/senha, biometria ou smartcards) e na estrutura utilizada entre o software gerente dos dispositivos físicos. Entretanto, por serem empresas que a grosso modo, têm em seu escopo de negócios, clientes de grande porte, ocultam ao máximo seus códigos-fonte afim de dificultar a busca por falhas eventualmente exploradas em uma entrada furtiva e também para manter a propriedade intelectual devido à grande quantidade de pesquisas para gerar tal produto. Além do custo pago pelos dispositivos físicos de acesso, algumas dessas empresas cobram um valor extra, no caso do cliente desejar possuir o software gerente de permissões e restrições, pois é possível possuir apenas o modo de autenticação no próprio dispositivo.

No que concerne as soluções acadêmicas, foram pesquisados dois trabalhos de origem acadêmica, o projeto Sentinel e um trabalho relativo a controle de acesso ao prontuário eletrônico de pacientes.

O projeto Sentinel [8] é um sistema em Java para controle de acesso baseado em papéis. Existe um módulo de auditoria que registra os acontecimentos, porém a ênfase do trabalho é a autenticação. O Sentinel é baseado no conceito de plug-ins, onde é o método de autenticação é desenvolvido de acordo com a necessidade de cada dispositivo físico, tornando-o assim, versátil. Entretanto, é apresentado apenas trechos de códigos-fonte e diagramas UML, não sendo encontrado o código-fonte do sistema.

Já o outro projeto [11], visa permitir acesso aos prontuários de pacientes evitando grandes burocracias e de modo fácil, contrário ao que existe atualmente, que é baseado em papéis. Para isto, foi desenvolvido um sistema em Java que faz acessos a uma base LDAP. Este sistema tem a vantagem de ser um software livre, baseado em componentes sem custos de licenciamento e visa

apresentar bom desempenho para as demandas de acesso do mesmo.

Tendo em vista a dificuldade de flexibilidade, adaptação, introdução de novas categorias de dispositivos físicos e dependência de fornecedores/fabricantes específicos, incluindo altos custos que as soluções proprietárias apresentam, é proposto neste trabalho a arquitetura ESC, buscando construir uma solução utilizando apenas ferramentas livres.

### III. A ARQUITETURA ESC

A arquitetura ESC (Environment Security Control) foi desenvolvida com o intuito de facilitar o gerenciamento de acesso e controle de ambientes restritos. Seu destaque principal é a facilidade de comunicação entre diversos dispositivos de interação com o meio físico e um gerente centralizado de alto nível. Isto é feito através de uma abstração do tipo de dado de entrada, criando, desta maneira, uma grande transparência entre o meio físico e a administração do sistema. A arquitetura foi dividida em dois subsistemas: ESCHA (ESC Hardware), qualquer dispositivo remoto, e ESCMA (ESC Manager), o gerente centralizado para processamento das informações. A conexão entre os dois ocorre através de interfaces de rede ethernet, usando a pilha de protocolos TCP/IP. Uma ilustração desse sistema pode ser vista na figura 1.

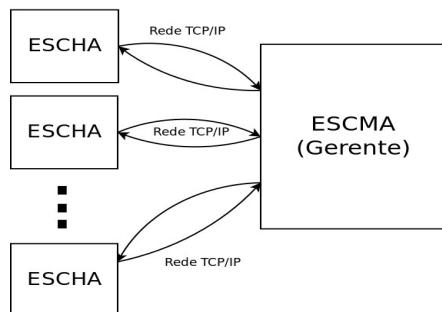


Figura 1. Conexão do hardware remoto com o gerente de alto nível

O dispositivo remoto foi desenvolvido utilizando os microcontroladores AVR de baixo custo da Atmel [1], juntamente com outros periféricos e interfaces que possibilitam a interação com o mesmo. A arquitetura do sistema pode ser dividida em quatro partes, host, interface visual, dispositivo de autenticação (entrada de dados), interface de rede. A figura 2 mostra os componentes envolvidos na comunicação entre um dispositivo de autenticação com a aplicação gerente.

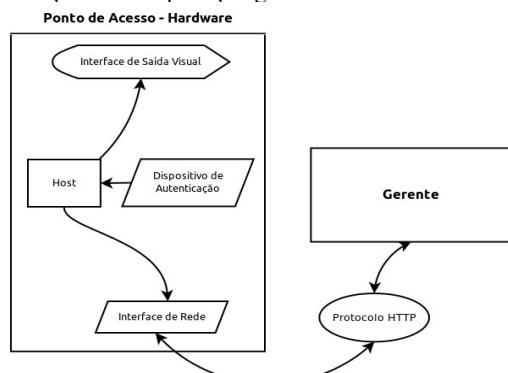


Figura 2. Conexão do hardware remoto com o gerente de alto nível

O dispositivo de hardware desenvolvido consiste em uma interface de coleta de dados e acionamento de outros elementos. Portanto, não é autônomo e depende de um gerenciamento em alto nível para exercer sua função. Sendo esta dependência uma das barreiras que este trabalho se propõe a superar: a dificuldade de comunicação entre um dispositivo de hardware remoto e uma rede de comunicação.

### IV. PROTOCOLO DE COMUNICAÇÃO

Como foi citado anteriormente, o hardware remoto serve apenas como uma plataforma de coleta de dados e acionamento, sem autonomia própria para autenticação. Para que o sistema funcione efetivamente é necessária uma comunicação com o gerente (ESCMA). Esta comunicação foi desenvolvida na camada de aplicação utilizando-se da já consolidada pilha de protocolos TCP/IP, através do protocolo HTTP. Esta, prove as funcionalidades necessárias ao tráfego de dados entre dispositivos e gerente, utilizando-se, portanto, uma infraestrutura largamente utilizada na construção de qualquer rede de comunicação. Portanto, é eliminada a necessidade da criação de uma nova rede cabeada específica para o dispositivo, como uma rede RS232 para conexão de dispositivos [2]. Outra vantagem do sistema trabalhar na camada TCP/IP é a facilidade de desenvolvimento de softwares que se comuniquem com o dispositivo. Evitando, assim, que o desenvolvedor precise aprender linguagens específicas e possa trabalhar em linguagens nas quais já está acostumado.

Para a comunicação entre dois dispositivos, não basta haver uma canal de comunicação entre eles, também é necessário haver um protocolo de comunicação entre esses elementos.

No âmbito do ESC, utilizou-se como parâmetros para a definição deste protocolos itens básicos que devem ser providos em um sistema de segurança, sendo:

a) Simplicidade: o protocolo de comunicação deve ser implementado, basicamente em dois pontos distintos: a) o ESCMA, onde é implementado através de um linguagem de alto nível, como java e, b) no dispositivo de autenticação, construído baseado na plataforma Arduino, e portanto, com diversas limitações sobre a capacidade de processamento deste. Desta forma, em função de tais limitações, protocolos complexos para comunicação e criptografia não são viáveis.

b) Segurança: o sistema deve, impreterivelmente, utilizar mecanismos que garantam a segurança da comunicação entre o dispositivo de autenticação e gerente. Para isso é essencial a utilização de alguma forma de criptografia.

c) Comunicação Assíncrona: eventos de autenticação são gerados de forma assíncrona, portanto é necessário estabelecer formas de comunicação que permitam minimizar o uso da comunicação da rede, ou seja, o dispositivo deve notificar ao gerente que um evento ocorreu.

d) Independência de dispositivo: independente do dispositivo de hardware utilizado para realizar a autenticação, como RFID, teclado para digitação de senha, etc, o protocolo projetado deve suportar, sem alterações na sua estrutura, as informações necessárias a utilização destes.

Baseado em tais critérios, o protocolo construído é apresentado na figura 3. Seu funcionamento baseia-se em envio de pacotes que seguem um padrão de montagem. O protocolo de comunicação abrange, com poucas modificações no hardware, muitos tipos de entrada de dados, seja ela por digitação de senha, leitura de cartões RFID, leitores biométricos, códigos de barra, etc. A reprogramação deste módulo do dispositivo permite que o desenvolvedor adicione mais métodos de entrada de dados com facilidade.

#### A. Análise da Transmissão

O pacote de transmissão é montado a partir de vários fatores e dados. Como é visto na figura 3, ele é dividido em várias partes. É necessário notar que pelo fato dos dados serem transmitidos em texto plano, algumas medidas de proteção são necessárias para evitar que o sistema se torne vulnerável à ataques maliciosos, ou até mesmo a problemas de sincronia. Na figura 3 é apresentada uma ilustração que servirá como exemplo para as explicações. Agora, será feita uma análise detalhada da mesma.

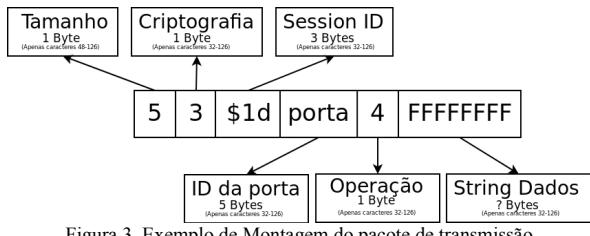


Figura 3. Exemplo de Montagem do pacote de transmissão

1) *Tamanho*: o primeiro byte da transmissão tem como função evitar erros e leituras incorretas. O tamanho é um inteiro em forma de caractere que representa o tamanho total do pacote da camada de aplicação (construído através de uma string), sua leitura é feita a partir da conversão para valor alfanumérico no padrão ASCII. Só pode assumir valores a partir de '0', ou seja, 48 em alfanumérico. É necessário notar que a string possui um tamanho mínimo, ou seja, deve receber os blocos principais: tamanho, criptografia, SID, ID da porta e operação, caso contrário ela deve ser descartada, pois não há informações relevantes a serem aproveitadas.

2) *Criptografia*: este byte irá definir qual o tipo de criptografia a ser usada para descriptografia dos dados relevantes, como os dígitos identificadores de sessão (SID). Os métodos de criptografia devem ser implementados de maneira equivalente entre ESCHA e ESCMA. Métodos de encriptação testados e com bom funcionamento no sistema envolvem embaralhamento e ocultação, algoritmos como o XOR cipher [3] e Caesar Cypher [4]. Devido às restrições de hardware, criptografias unidirecionais (MD5, SHA)[5] não podem ser utilizadas devido ao fato do equipamento não possuir uma base de dados. Esta não é a principal fonte de segurança do sistema, seu principal objetivo é apenas adicionar uma camada de proteção adicional.

3) *Session ID*: o Session ID, ou dígitos identificadores de sessão, são o principal sistema de segurança e proteção do protocolo de transmissão. Seu princípio de funcionamento é semelhante ao tipo de autenticação "One time password"[6], onde senhas aleatórias são geradas e esquecidas periodicamente. Estes dígitos são gerados sempre que uma comunicação é iniciada, chamada de sessão, e acaba automaticamente após receber uma resposta ou após a passagem de um tempo pré-definido. A SID é criptografada através de algoritmos pré-definidos, e age como uma pergunta que deve ser respondida corretamente para uma transmissão ser bem sucedida. Na figura 4 vemos o gerente recebendo os dados com uma SID criptografada e dados, juntamente com o método de descriptografia a ser usado, logo após receber e processar os dados, uma resposta é gerada (permite/nega acesso, por exemplo) e enviada para o dispositivo remoto, juntamente com a SID descriptografada.

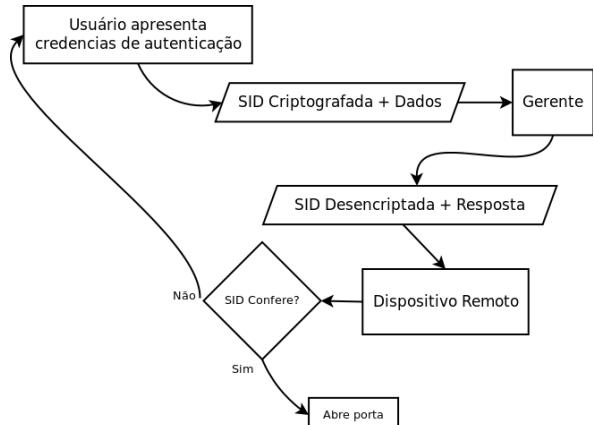


Figura 4. Fluxo de uma comunicação para autenticação

4) *ID da Porta*: dígitos identificadores da porta servem para informar ao gerente o destino da resposta atual sendo tratada.

5) *Dígito de Operação*: este byte irá informar aos sistemas qual operação precisa ser feita com os dados passados, exemplos de operações: Abrir porta, permitir ou negar autenticação de dados, enviar mensagem para o display do dispositivo. O sistema está preparado para receber novas implementações de operações sem alterações em sua estrutura principal.

6) *String de Dados*: este elemento carrega representa informações adicionais a alguma operação. Operações como autenticação podem carregar, por exemplo, dados de um cartão RFID, senhas de um teclado numérico, strings provindas de leitores biométricos entre outros métodos de entrada. Outra função chave do sistema que utiliza os dados é a resposta de autenticação, que contém uma palavra chave informando se a autenticação foi aceita ou negada.

## V. AVALIAÇÃO EXPERIMENTAL

Para análise da solução proposta, montou-se um protótipo integrando o sistema ESCHA ao sistema ESCMA. Para tal, utilizou-se um ambiente de testes alinhado as expectativas de um ambiente real, ou seja, montou-se toda a estrutura de autenticação. Esta, é composta de um dispositivo de autenticação junto aos acessos ao ambiente físico, do hardware necessário para comunicação de rede (ESCHA) interligado, através da rede, ao software para gerenciamento e autenticação centralizada, o ESCMA.

O protótipo do ESCHA foi criado a partir da união de uma placa específica desenvolvida para receber um microcontrolador da Atmel, juntamente com um módulo de interface de rede, que possibilitou a conexão à rede ethernet. Este conjunto está integrado a um módulo de antenas para leitura e escrita em cartões RFID, sendo esta a entrada de dados, e, como a saída de dados, um módulo LCD alfanumérico de 16 colunas e 2 linhas, por final, também foram adicionadas relés para o acionamento de dispositivos externos (fechadura eletromagnética, por exemplo).

O ambiente de teste é ilustrado na figura 5.

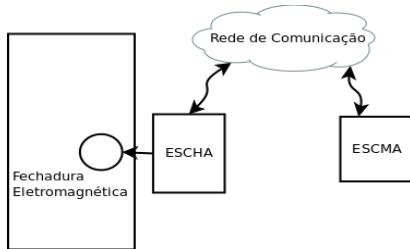


Figura 5. Ambiente de testes ESC

O sistema foi implementado em forma de protótipo para a execução de testes de funcionamento e estresse. Os primeiros experimentos realizados foram estudos para otimizar o funcionamento do protocolo de comunicação, ajustando valores, principalmente de temporização (duração de sessões, tempo entre coleta de dados). O protocolo se mostrou bastante eficiente, dando início aos testes de segurança. Mesmo com a transmissão dos dados ocorrendo sem uma criptografia avançada, em texto plano, a proteção implementada pelo protocolo (criptografia simples e os dígitos identificadores de sessão) dificulta muito a intervenção de programas maliciosos na transmissão. Mesmo com dados analisados através de snifffers, como o Wireshark [7], o processo de quebra da segurança requer níveis exponenciais de estudo e interpretação dos pacotes de pedido e resposta. A primeira vista, os pacotes parecem apenas caracteres aleatórios sem um padrão definido, pois a cada tentativa de acesso uma nova criptografia é usada, juntamente com a criação de uma nova seed que alimenta as mesmas. Apenas um servidor desenvolvido especificamente para aquele hardware será capaz de entender e produzir respostas adequadas para o mesmo.

Com a validação do processo de proteção dos dados, o sistema foi testado sob estresse, forçando requisições de acesso a quantias e intervalos de tempo extremos, quantizando mais de novecentas requisições por hora, em longos períodos de testes. Na tabela 1, é possível observar dados de alguns testes realizados, sendo que no último

teste<sup>1</sup> o tempo entre as requisições foi reduzido pela metade, colocando o sistema em uma situação mais extrema, mesmo assim podemos observar uma taxa de falhas de autenticação relativamente baixa.

Período	Tentativas	Sucesso	Falha	%
8 horas	8228	8196	32	99,6
2,5 horas	2812	2807	5	99,8
5 horas	15631 <sup>1</sup>	14777	854	94,5

Tabela 1. Testes de estresse executados

O sistema atingiu uma taxa de sucesso acima de 99%, mostrando-se robusto para aplicação prática e pronto para solucionar os problemas que deram início ao mesmo, bem como para receber novas atualizações e gerar, a partir dele, novos e avançados equipamentos de monitoria e controle de ambientes.

## VI. CONCLUSÕES E NOVOS SISTEMAS

Esta é uma arquitetura que pede por atualizações frequentes. As possibilidades são muitas, dentre novas funcionalidades propostas, podemos citar: Mapeamento de pessoas baseando-se no último lugar visitado, sensoriamento e controle de ambientes (como temperatura, luminosidade) através de perfis de usuários, reconhecimento facial e sensoriamento de pessoas num ambiente para controlar outros elementos (travar porta, mapeamento melhorado, controlar aparelhos eletrônicos).

## REFERÊNCIAS

- [1] Stanislav Korbel, Vlastimil Janes. Interesting Applications of Atmel AVR Microcontrollers. Department of Computer Science and Engineering of the Czech Technical University. Aug. 2004.
- [2] Strangio, C. E. The RS232 Standard. CAMI Research Inc., Acton, Massachusetts (1993) 15.
- [3] P. Tuyls, H. D. L. Hollmann, J. H. Van Lint and L. Tolhuizen. XOR-based Visual Cryptography Schemes. Volume 37, número 1. 2005.
- [4] Dennis Luciano, Gordon Prichett. "Cryptology" From Caesar Ciphers to Public-Key Cryptosystems. The College Mathematics Journal, Volume 18, Número 1, pp. 2-17. 1987.
- [5] Gary C. Kessler, An Overview of Cryptography, Jun. 2010.
- [6] N. Haller, C. Metz, P. Nesser, M. Straw. A One-Time Password System. IETF RFC2289, Feb. 1998.
- [7] Chris Sanders.Practical Packet Analysis (First ed.). No Starch Press, San Francisco, CA, USA. 2007.
- [8] MATTOS, C. L. A. Sentinel: um engenho Java para controle de acesso RBAC. 2003. Pernambuco. Disponível em: <<http://www.cin.ufpe.br/~tg/2003-1/clam.doc>>. Acesso em: 25/10/2011. Trabalho de Graduação em Segurança da Informação. 50 p.
- [9] ID TECH, Soluções. 2006. Disponível em: <<http://www.idtech.com.br/solucoes.asp>>. Acesso em 08/11/2011.
- [10] NIBTEC, Controle de Acesso. 2011. Minas Gerais. Disponível em: <<http://nibtec.com.br/produtos.html>>. Acesso em 08/11/2011.
- [11] MOTTA, Gustavo H. M. B. Um modelo de autorização contextual para o controle de acesso ao prontuário eletrônico do paciente em ambientes abertos e distribuídos. 2003. São Paulo. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3142/tde-05042004-152226/pt-br.php>>. Acesso em: 30/8/2012. Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Doutor em Engenharia.

<sup>1</sup> Teste do sistema, utilizando o intervalo entre requisições reduzido a metade do tempo usual, dos demais testes.

# Monitoramento Automático de Falha em Transformadores de Redes de Distribuição de Energia Elétrica Utilizando Tecnologia ZigBee

Tiago Saidelles\*

UFSM – Universidade Federal de Santa Maria  
tiago\_saidelles@yahoo.com.br

Claiton Pereira Colvero

UFSM – Universidade Federal de Santa Maria  
claiton@redes.ufsm.br

**Resumo**—Este trabalho foi desenvolvido com o objetivo de implementar um sistema de monitoramento e controle automático de falhas em transformadores de redes de distribuição de energia elétrica através de uma rede industrial de tempo real e adaptativa, montada em topologia *mesh*, utilizando dispositivos baseados na tecnologia ZigBee.

## I. INTRODUÇÃO

As atuais redes de distribuição de energia elétrica tem se modernizado constantemente com o passar dos anos, aumentando a confiabilidade e segurança das operações de forma significativa, porém ainda assim é possível de falhas do sistema, seja por problemas próprios ou agentes externos, principalmente devido a sua grande capilaridade.

Embora os clientes das principais companhias distribuidoras de energia possuam diversos canais de comunicação com as mesmas, até por um motivo mais cultural, costumam fazer suas reclamações através do serviço de *Call Center* via telefone, gerando cerca da metade dos chamados totais somente por esta modalidade [1].

É sempre importante observar que os índices de satisfação dos clientes em geral não muito altos devido ao grande número de ligações para os *Call Centers* das operadoras. Desde as interrupções de energia por falhas do sistema até o atendimento são normalmente relatados pelos usuários grandes períodos de espera. Esta demora é decorrente do tempo desde a percepção da falha, o acionamento do *Call Center*, a localização do cadastro do reclamante, o encaminhamento da ordem de serviço à equipe de manutenção e o conserto da rede propriamente dito.

Diversas destas informações relevantes foram amplamente tratadas no Relatório de Sustentabilidade 2011 da empresa AES Sul [1]. Foi elaborada uma pesquisa referente às ligações recebidas na Central de Atendimento ao Cliente, sendo constatado que a maioria das mesmas eram exclusivamente reclamações referentes a interrupções de energia. A tabela abaixo apresenta estes dados.

Tabela I  
EXCELÊNCIA NO ATENDIMENTO AES SUL [1]

Motivo da Reclamação	2009	2010	2011
Total de ligações atendidas ( <i>call center</i> )	2.955.129	2.655.700	2.630.49
Número de atendimentos nos escritórios regionais	310.722	318.997	299.784
Número de atendimentos por meio da internet	571.270	643.732	1.272.617
Reclamações em relação ao total de ligações atendidas (%)	27,72	34,67	35,08
Tempo médio de espera até o início do atendimento (minutos)	0,58	1,32	0,41
Tempo médio de atendimento (minutos)	3,17	3,25	3,32

\*Apresentador do trabalho no ERRC 2012

Motivados por esse processo descrito anteriormente, definido como ineficiente pela grande maioria dos consumidores que estão sem energia, e também para atender uma demanda crescente das companhias de distribuição de energia, desenvolveu-se um sistema completo de monitoramento, sensoriamento e atuação em tempo real para minimizar os prejuízos causados por falhas em transformadores de distribuição de energia elétrica.

Foi verificada a necessidade de prover uma conexão em tempo real entre os dispositivos de monitoramento da rede e uma Central de Controle e Monitoramento Automática, com capacidade de perceber a falha e imediatamente através de seu tomador de decisão gerar uma ordem de serviço específica para acionar as equipes de manutenção. A eliminação da necessidade do cliente entrar em contato com a Companhia para notificar a falha agiliza o processo e contribui de forma positiva para sua satisfação em relação ao atendimento, diminuindo o impacto da falha.

O sistema foi implementado experimentalmente utilizando a tecnologia de rede sem fio conhecida como *ZigBee*, que opera na frequência de 2,4 GHz, dentro da faixa *ISM* que não necessitando de licença especial do Órgão Regulador de Telecomunicações para implantação. As redes *ZigBee* oferecem uma excelente imunidade contra interferências, uma capacidade de hospedar mais de 65 mil dispositivos em uma única rede, e ainda possuem características de interconexão adaptativas (*mesh*), alcançando grandes distâncias de comunicação.

## II. DESENVOLVIMENTO DO PROTÓTIPO

Este trabalho foi desenvolvido em etapas para melhor adequação do sistema com as reais demandas das companhias de distribuição de energia elétrica. Resumidamente ele foi dividido como uma análise de caso e referência bibliográfica, análise das necessidades e problemas de atendimento das companhias, montagem de um protótipo em escala para prova de conceito dos sensores e funcionalidades da rede, e o desenvolvimento do software de controle e monitoramento de falhas, desenvolvido em linguagem de programação *Python* [2].

Utilizando como premissa alguns requisitos básicos para o funcionamento do sistema juntamente com a viabilidade técnica e econômica do projeto, foi iniciada uma pesquisa detalhada sobre as tecnologias atuais disponíveis no mercado de sensoriamento e rede que suprissem as demandas do projeto. Dentre estas necessidades, foi observado o baixo consumo de energia, a possibilidade de comunicação entre dispositivos com alta confiabilidade, o grande alcance, a capacidade de suportar vários dispositivos

conectados na mesma rede e não menos importante, o custo de implementação e manutenção.

Após uma detalhada avaliação, optou-se pela utilização da tecnologia conhecida como *ZigBee*, que foi criada em 2002, quando algumas empresas, (*Honeywell*, *Invensys*, *Philips* e a *Mitsubishi Electric*), uniram seus esforços para criar um consórcio, a *ZigBee Alliance*, com o intuito de desenvolver um padrão que atendesse a diversos requisitos, entre eles [3]:

- Confiabilidade na entrega dos dados;
- Baixo consumo de energia;
- Baixo custo de produção;
- Baixa irradiação de espúrios;
- Padrão global aberto.

Observando as tendências de mercado das redes industriais sem fio, em 2003 a *IEEE (Institute of Electrical and Electronics Engineers)* estabeleceu o novo padrão 802.15.4, que serve como base para as camadas física (*PHY*) e de enlace (*MAC*) do *ZigBee*. Esta tecnologia é dotada ainda das camadas de rede e de aplicação (subdividida), o que faz dela uma categoria singular e com propósitos específicos de *WPAN (Wireless Personal Area Network)*. Diferente das redes de comunicação sem fio dos padrões 802.11 e 802.16, que também compartilham faixas de frequências *ISM*, o padrão 802.15.4 foi especialmente desenvolvido para comunicações de longo alcance com consumo reduzido, [4] utilizando baixa potência de transmissão e sem a necessidade de altas taxas de transmissão de dados no enlace. Para utilização deste padrão de rede industrial de tempo real foi necessária a configuração dos dispositivos conforme descrito a seguir.

#### A. Configuração da Rede ZigBee como Sensor de Falha

Para iniciar a configuração dos dispositivos de comunicação *ZigBee* foi definida a topologia de rede e as funcionalidades específicas de cada um dos módulos de rede configuráveis. Optou-se assim pela implantação de uma rede adaptativa do tipo *mesh*, com a configuração de maior economia de energia para os dispositivos definidos como sensores através da programação da função *RFD* (*Reduced Function Device*) e os demais coordenadores e roteadores como *FFD* (*Full Function Device*).

Na prova de conceito realizada através de um modelo em escala reduzida, foram utilizados dispositivos de rede *ZigBee*, relés de corrente, chaves de operação binárias (*On-Off*), LEDs e demais componentes eletrônicos de interfaceamento. Foi simulada a situação real de uma rede elétrica com dois transformadores operando em regime constante, sendo que cada um possuía um dispositivo *ZigBee* de sensoriamento atuando configurado como *End Device* (dispositivo final) na função *RFD*, para economia de energia. As chaves de operação binária simulavam uma situação de falha por desarme de fusível de um determinado transformador escolhido. Quando o sensor instalado em cada transformador percebia esta queda de energia elétrica da rede de distribuição, o dispositivo final enviava imediatamente uma requisição de verificação e manutenção para a central de monitoramento e controle. Automaticamente também informava os seus endereços

de 16 bits e 64 bits e o nome de rede em *ASCII*, chamado de *NI*, que contém além de uma chave de banco de dados, as informações de endereço do transformador e do ponto de falha. Esta configuração do módulo *ZigBee* operando como sensor pode ser visualizado na Figura 1.

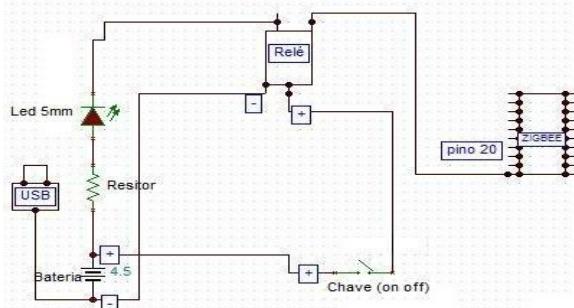


Figura 1. Diagrama do circuito de acionamento do modelo em escala.

Para incrementar a dinâmica da rede, foram adicionados módulos *ZigBee* configuradores como roteadores autônomos entre os dispositivos finais que atuavam como sensores de falhas, atuando somente como nós alternativos da rede *mesh* para garantir a entrega do quadro de dados gerado pelo sensor, aumentando a confiabilidade da rede pela redundância de caminhos de entrega e roteamento.

Na central de monitoramento e controle foi instalado um dispositivo *ZigBee* configurado como coordenador na função *FFD*, com o objetivo de receber todas as informações de falhas enviadas pelos dispositivos finais e passar as mesmas diretamente ao software de gerenciamento. Este coordenador converte o quadro enviado em modo *API* para um formato de melhor visualização no terminal e emissão de uma ordem de serviço automática para as equipes de manutenção.

Os módulos sensores *ZigBee* configurados como dispositivos finais devem ser instalados diretamente nos postes de transformadores de distribuição de energia elétrica, de baixas e médias tensões de corrente alternada, embora eles operem com tensões de corrente contínua de no máximo 3,4V. Como solução mais segura e eficiente do ponto de vista da fonte de alimentação do sensor, foram utilizados nestes dispositivos baterias internas. Para aumentar a vida útil da bateria para pelo menos 5 anos de operação, os dispositivos *ZigBee* atuam configurados como dispositivos finais com funções reduzidas (*RFD*), sendo condicionados a permanecerem no modo *sleep* (dormindo) até que algum problema seja detectado no transformador. Se houver um evento, o sensor *ZigBee* acorda (*wake-up*) imediatamente de forma automática e envia um quadro de dados ao coordenador da rede, que está instalado na central de controle e monitoramento. Este coordenador recebe a informação de qual transformador falhou e gera uma ordem de serviço para a equipe de manutenção mais próxima, conforme pode ser visto no fluxograma da Figura 2.

Caso um destes dispositivos finais configurados como sensores não conseguirem fazer a conexão direta com o coordenador por qualquer motivo, os demais dispositivos configurados como *Router* da rede terão a finalidade de encaminhar este quadro de requisição de

atendimento através de seus nós, caracterizando uma rede *mesh*. Na central de controle e monitoramento o dispositivo configurado como coordenador está encarregado de receber as informações sobre o problema detectado e proceder o encaminhamento da tomada de decisão.

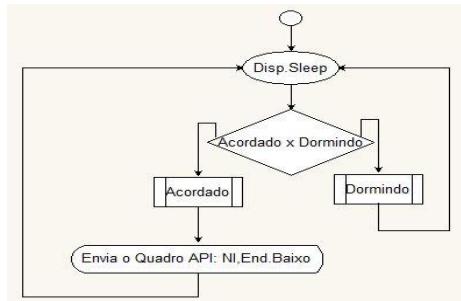


Figura 2. Fluxograma resumido do funcionamento do *ZigBee* configurado como Dispositivo Final da rede – Sensor de falha.

Para a configuração dos dispositivos *ZigBee* foi utilizando o software da *Digi* chamado *X-CTU* [5], permitindo programar os módulos para a formação da rede. Dispositivos finais se comunicam apenas diretamente com o coordenador ou através de seus nós roteadores. Na configuração dos dispositivos finais utilizados como sensores foi atribuído um nome de identificação em *ASCII* no campo *NI*, com o intuito de facilitar a identificação no terminal da central de controle e monitoramento mesmo antes do acesso ao banco de dados.

O coordenador da rede foi programado para atuar em modo *broadcast*, no intuito de escutar todos os dispositivos sensores que estejam no mesmo *channel* e *Pan ID*. Foi também definida a utilização do modo *API* conforme pode ser visto na Figura 3 [6] para todos os dispositivos *ZigBee*. Notadamente era imperativo que o coordenador identificasse o endereço baixo, o endereço alto e o *NI* do dispositivo final sensor quando este fosse acionado, sem a necessidade dele enviar uma requisição de identificação em *broadcast* e aguardar a resposta, poupando energia e principalmente não sobrecarregando a banda da rede, que tem capacidade de operar com até 65.535 sensores.

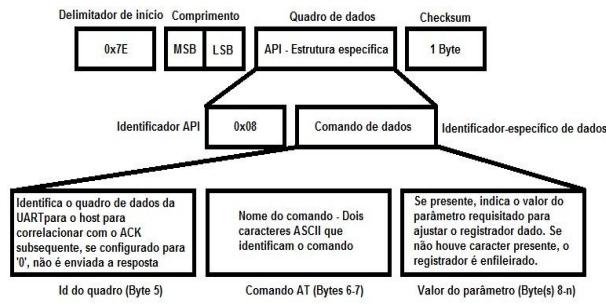


Figura 3. Estrutura do frame (quadro) *API* dos dispositivos *ZigBee*.

O modo *API* (*Application Programming Interface*) dos dispositivos *ZigBee* é um modo mais complexo de realizar a transmissão. Ao invés de enviar comandos diretamente através da interface serial, os comandos são colocados em uma em uma interface estruturada, ou seja, é feita a comunicação de dados em uma ordenação dos quadros pré-definida. O modo *API* permite ao programador definir

como comandos, respostas e status serão enviados e recebidos através da interface de quadros da *UART*.

O acionamento do sensor em caso de falha é sinalizado ao dispositivo *ZigBee* através do desligamento do relé de corrente, e este por sua vez aterra o nível do sinal na porta de entrada do conversor *AD* do dispositivo final (pino 20). Imediatamente após, o *ZigBee* acorda do modo *sleep* e envia sua identificação completa ao coordenador através de um quadro em modo *API*, economizando a bateria do sensor quando o mesmo não está acionado por falha.

Uma vez resolvida a implementação do *hardware* dos sensores, foi desenvolvida a interface de controle e monitoramento, conforme descrito a seguir.

### B. Software de Controle e Monitoramento em Python

O software de controle e monitoramento está sendo desenvolvido em linguagem de programação *Python*. Ele possui basicamente a finalidade de converter o quadro enviado para o coordenador, proveniente de um dispositivo final que atua como sensor, em uma linguagem em que o operador possa identificar facilmente onde ocorreu o problema. Também é responsável por fazer uma interação do software com um banco de dados onde serão armazenadas informações de localização de cada ponto de rede que tiver um dispositivo final acionado, preenchendo uma ordem de serviço de forma automatizada para que uma equipe de manutenção.

Na Figura 4 pode-se observar o fluxograma de tratamento do quadro *API* recebido pelo coordenador quando ocorre uma falha na rede elétrica.

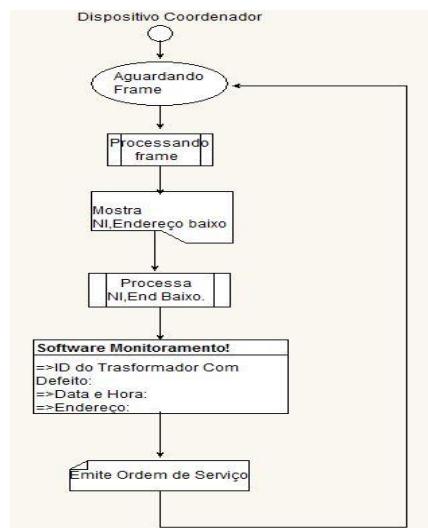


Figura 4. Fluxograma de processamento do quadro recebido pelo dispositivo *ZigBee* coordenador da rede.

## III. RESULTADOS OBTIDOS

Os resultados obtidos neste projeto objetivam avaliar a viabilidade de detecção automática de falhas por problemas técnicos em transformadores de distribuição de energia elétrica. Para os ensaios iniciais foi utilizado um modelo em escala reduzida, onde a mudança de estado de uma chave

binária simula a falha em um sistema real, que aciona imediatamente o sensor incorporado a um módulo *ZigBee* configurado como dispositivo final. Como este sensor permanece em estado normal de *sleep*, ele percebe o aterrramento da entrada *I/O* chamada de *DIO*. A partir deste momento o *ZigBee* acorda e envia um quadro de dados que contém os endereços alto, baixo e sua identificação em forma caracteres *ASCII*, chamado de parâmetro *NI*.

Para monitorar o envio do quadro foi utilizado um terminal com capacidade de verificar o tráfego na interface *UART* do módulo. Foi obtido sucesso total no envio dos pacotes de dados em diferentes tempos e configurações de acionamento das chaves, entregando as informações relevantes em no máximo 10 ms após o momento do acionamento do sensor, característico de uma rede industrial que trabalha em tempo real.

Como segunda etapa dos experimentos, foi realizada uma análise detalhada para desenvolver o software de interfaceamento. Ele deve ter a capacidade de converter as informações enviadas pelo sensor em informações legíveis no terminal para um operador, e também uma base de dados com informações georeferenciadas da localização de cada transformador monitorado. Como opção do projeto, foi escolhido utilizar os módulos *ZigBee* no modo *API*, que transmite um quadro em hexadecimal.

Naturalmente a escolha deste tipo de quadro *API* necessita um maior esforço por parte do programador para desenvolver a interface da central de controle e monitoramento. Por esse motivo foi definido que a linguagem mais adequada para essa função é o *Python*, que é uma linguagem de alto nível, interpretada e orientada a objeto. Com o auxílio desta ferramenta foi possível desenvolver uma interface completa com capacidade de interpretar e separar os endereços alto (*SH*), baixo (*SL*) e o nome do sensor em *ASCII* (*NI*).



Figura 5. Exemplo da tela de interface com o operador na central de controle e monitoramento.

Na Figura 5 pode-se observar uma tela que demonstra a interface preliminar desenvolvida para a central de controle e monitoramento dos sensores instalados, onde um operador pode acompanhar o *status* dos dispositivos da rede e o encaminhamento das ordens de serviços com seus respectivos atendimentos, de forma imediata e simples.

Concluindo os experimentos foi realizado paralelamente um ensaio para estabelecer a máxima distância direta entre um dispositivo final e o seu coordenador, obtendo no máximo 1.110 metros de enlace direto utilizando dispositivos *ZigBee PRO*. Posteriormente foi refeito o mesmo experimento utilizando adicionalmente dispositivos *ZigBee* na função de roteadores e as funcionalidades da rede *mesh*, criando caminhos alternativos através dos novos nós da rede. A distância do enlace foi incrementada n-vezes de acordo com o número de roteadores que foram adicionados linearmente no percurso estabelecido.

#### IV. CONCLUSÃO

Este trabalho teve como objetivo o desenvolvimento de um sistema de monitoramento automático de falhas para redes de distribuição de energia elétrica baseado em redes de comunicação sem fio, utilizando a tecnologia e as facilidades dos dispositivos no padrão *ZigBee*.

Um sistema de monitoramento foi desenvolvido com intuito de aumentar o índice de satisfação dos clientes das Companhias de distribuição de energia elétrica, fornecendo uma importante ferramenta de agilização na detecção e correção de falhas, eliminando a necessidade de o cliente entrar em contato com a Companhia para registrar o problema. De forma automática, ele é capaz de identificar o local e gerar uma ordem de serviço para a equipe de manutenção responsável pelos reparos.

Através dos resultados obtidos nos ensaios avaliamos a viabilidade de implementação do sistema em enlaces longos através do uso de roteadores adicionais aos sensores instalados, como por exemplo, em postes da rede sem transformadores. Se for necessário aumentar a dinâmica do enlace, tanto por distância como por interferências, pode-se instalar um novo dispositivo *ZigBee* com funções de roteador sem agregar novos problemas adicionais na rede.

Como trabalho futuro será implementado um algoritmo para calcular a rota mais eficiente para que as equipes de manutenção possam chegar à área que foi constatada o problema, além de uma integração com interfaces de geoposicionamento, como o *Google Maps*.

#### REFERÊNCIAS

- [1] P. C. V. Penna e M. Magno, "Relatório de Sustentabilidade 2011 – AES Sul", III Ciclo de Diálogos com Públicos de Relacionamento AES: [Aesbrasilssustentabilidade.com.br](http://Aesbrasilssustentabilidade.com.br), pp. 40-43, abril 2012.
- [2] Python Programming Language - Python Software Foundation – Official Website: [www.python.org](http://www.python.org), Copyright 1990 - 2012.
- [3] Campos, F. P. S. C. Estudo e especialização de um sistema de instrumentação para unidades de elevação de petróleo utilizando tecnologia sem fio. 2006. pp. 77, Dissertação de Mestrado na Universidade Federal do Rio Grande do Norte, Natal, 2006.
- [4] Santos, S. A. Sintetizador de frequências de 2.4 GHz em cmos, 0,35µm para aplicações em ZigBee. 2008. Pp. 72, Dissertação de Mestrado – Escola Politécnica da Universidade de São Paulo, São Paulo, 2008.
- [5] XBEE®/XBEE PRO® ZB RF MODULES. ZigBee RF Modules by Digi International. Disponível em: <<http://www.digi.com>>. Acesso em: 2012.
- [6] Rubens's Blog. Exemple of XBee API frames. 12 mar. 2009. Disponível em: <<http://rubenlaguna.com/wp/2009/03/12/example-of-xbee-api-frames/index.html>>. Acesso em: 2012.

# ANÁLISE DE DESEMPENHO DE REDES SEM FIO COM DIFERENTES PROTOCOLOS DE CRIPTOGRAFIA

Douglas Pegoraro Stangarlin, Walter Priesnitz Filho  
UFSM  
{douglas, walter}@redes.ufsm.br

**Resumo**—Este trabalho apresenta o estudo e desenvolvimento de uma análise de desempenho de redes sem fio IEEE 802.11, também conhecida como WiFi, levando em consideração os diferentes padrões de segurança existentes comercialmente para esta tecnologia. O trabalho foi desenvolvido em um ambiente controlado, em uma rede do tipo infraestrutura, considerando a análise dos padrões de segurança em diferentes tempos de testes. Os testes foram executados, sobre o protocolo de transporte UDP, a fim de comparar o desempenho conforme os diferentes padrões de segurança são utilizados.

## I. INTRODUÇÃO

Com o aumento de disponibilidade e serviços das redes sem fio, cada vez mais equipamentos destinados a este fim surgem no mercado. Para que este crescimento continue, o aumento da segurança e do desempenho destas redes tornam-se fundamentais.

As questões de desempenho e segurança são vitais em redes de computadores. Em se tratando de redes sem fio estas questões são ainda mais importantes, pois esta deve ter um alto nível de segurança sem apresentar perdas significativas no seu desempenho.

Redes sem fio são mais fáceis de ser interceptadas do que as com fio. Uma vez que, para ter acesso ao sinal irradiado da rede basta estar no alcance deste sinal com um dispositivo compatível, já em uma rede com fio é necessário ter um ponto de acesso para esta rede.

Para tentar solucionar esta fragilidade é necessário o uso de mecanismos de segurança como cifragem e criptografia. Segundo Stallings [1] a cifragem é a transformação dos dados em um formato que não seja prontamente decifrável através de algoritmos matemáticos, tais procedimentos (transformação e recuperação dos dados) dependem de um algoritmo e zero ou mais chaves criptográficas. Em seu trabalho/estudo Stallings [1] enfatiza que a ferramenta automatizada mais importante em segurança de redes e comunicações é a criptografia.

A criptografia em uma rede de computadores requer utilização da largura de banda da mesma e processamento extra. Como descrito em Stallings [1], o modelo genérico de criptografia consiste em gerar um pacote, criptografar o mesmo e só depois enviá-lo para o destinatário, este por sua vez executa a decriptografia do pacote e só após este processo que o pacote é considerado transmitido com sucesso.

Existem trabalhos relacionados que comprovam que a criptografia interfere no desempenho dos sistemas com-

putacionais e das redes sem fio, podemos citar o trabalho de Suzin [2], o qual traz uma análise de desempenho em redes sem fio com o padrão de segurança OPEN até o WPA (*WiFi Protect Access*). Através dos testes realizados, Suzin [2] constatou que a utilização de um protocolo de criptografia mais robusto, como o WPA, demanda um maior poder de processamento e largura de banda da rede, em função do nível de segurança requerido. Já no trabalho de Barka e Boumal [3] são analisados os impactos do uso de criptografia em uma rede de infraestrutura 802.11g, demonstrando que a utilização do padrão de segurança WEP utilizando a criptografia RC4 interfere na vazão da rede em comparação a não utilização de criptografia.

Neste trabalho é feita uma análise e apresentação de quanto os protocolos de criptografia interferem no desempenho de uma rede sem fio, demonstrando estudos sobre o desempenho com os diferentes protocolos de criptografia disponíveis comercialmente na atualidade.

Este trabalho está estruturado como descrito a seguir: Na seção II são apresentados os padrões de segurança existentes atualmente e suas comparações. A seção III traz a metodologia utilizada para a realização dos testes deste trabalho, trazendo o ambiente de testes utilizado, os tipos de testes efetuados e apresentando as métricas de desempenho utilizadas para medir desempenho da rede. A seção IV traz os resultados dos testes efetuados, já a seção V traz as conclusões referentes ao trabalho.

## II. REVISÃO TEÓRICA

Para solucionar problemas de segurança em redes sem fio o IEEE (*Institute of Electrical and Electronics Engineers*), comitê responsável por padronizar as redes sem fio 802.11, definiu alguns padrões de segurança para serem utilizados nesta tecnologia.

A seguir são apresentados os padrões de segurança existentes e comercialmente disponíveis no Brasil.

### A. Padrão de Segurança WEP

O *Wired Equivalent Privacy* (WEP) é um padrão de segurança disponibilizado juntamente com o padrão 802.11 em 1999. O comitê 802.11 disponibilizou o protocolo sabendo de suas limitações, sendo o WEP a melhor opção disponível para a época. Segundo Thomas [4] o WEP foi desenvolvido com objetivo de tornar os dados trafegados tão seguros como se estivessem em uma rede Ethernet cabeada.

O WEP utiliza chaves fixas de 64 ou 128 bits, com conceito de *Shared Key*, na verdade desses bits 24 são do vetor de inicialização (IV) do WEP, restando, no caso do WEP64, 40 bits para a chave, ou seja, apenas 5 caracteres de chave e para o WEP128, 104 bits, 13 caracteres. Estas chaves devem ser compartilhadas entre os usuários, pois a mesma serve para criptografia e decriptografia dos dados.

O WEP combina o IV com a chave fixa para gerar pseudo-chaves, as quais servem para criptografar os dados. Para cada quadro transmitido é gerada uma nova pseudo-chave, isto torna a criptografia de cada quadro única.

#### B. Padrão de Segurança WPA

O Wi-Fi Protected Access (WPA) foi criado para solucionar os problemas do WEP. Pode ser usado com chaves compartilhadas, como no WEP, ou utilizando o padrão 802.1x, e EAP (*Extensible Authentication Protocol*) que identifica usuários através de certificados digitais.

Segundo Suzin [2] o WPA incorpora um esquema de criptografia denominado TKIP (*Temporal Key Integrity Protocol*), este embaralha os frames utilizando um algoritmo de hash que modifica a chave criptográfica a cada 10 pacotes.

O WPA utiliza o protocolo TKIP para criptografia dos dados através do algoritmo RC4, porém tomando algumas preocupações como não enviar a chave em texto claro e trabalha com uma política de IV mais inteligente. O WPA pode ser utilizado com uma chave secreta entre 32 e 512 bits.

#### C. Padrão de Segurança WPA2

O WPA2 é o padrão IEEE 802.11i na sua forma final, sendo que o WPA é a implementação de parte do padrão. Segundo Caixeta [5] o WPA2 foi desenvolvido para a obtenção de um nível de segurança ainda maior que no padrão WPA.

Caixeta [5] afirma que uma grande inovação do WPA2 é a substituição do método criptográfico do WPA pelo método AES-CCMP (*Advanced Encryption Standard*). O CCMP (*Counter-Mode/CBC-MAC Protocol*) é um modo de operação em cifragens de bloco, ele evita que a mesma chave seja usada para criptografia e autenticação.

#### D. Comparação entre os Padrões de Segurança para Redes Sem Fio

Segundo Amaral e Maestrelli [6] o TKIP foi desenvolvido para solucionar as deficiências do WEP, levando em consideração que a maioria dos equipamentos 802.11b utiliza baixo poder de processamento com limitações para grandes processamentos de segurança.

O AES foi desenvolvido pensando na maior segurança possível para redes sem fio, visto que as principais deficiências do WEP já haviam sido solucionadas pelo TKIP. Segundo Amaral e Maestrelli [6] o TKIP não provê o mesmo nível de segurança do AES, e a especificação da IEEE 802.11 descreve que o TKIP é recomendado para atualizações de equipamentos fabricados antes da publicação do padrão WPA2 (equipamentos pré-RSN - *Robust Security Network*), ou seja, utilizar o WPA como

atualização para o WEP, principalmente por deficiência de equipamentos que não suportam o AES, e que necessita de maior poder de processamento.

O trabalho de Amaral e Maestrelli [6] traz um comparativo entre as diferenças nos padrões de segurança para redes wireless, demonstrando a evolução da segurança, o que se pode observar na Figura 1.

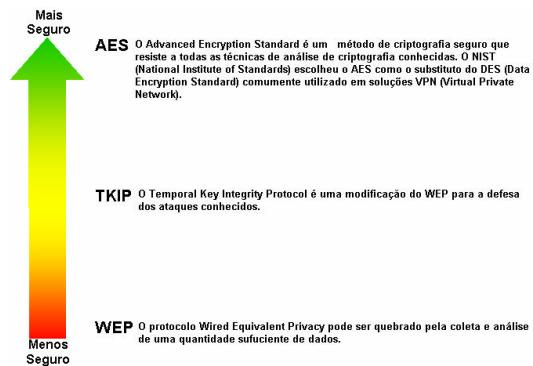


Figura 1. Comparativo entre os padrões de segurança em redes wireless

### III. METODOLOGIA

Para o desenvolvimento deste trabalho foi configurado um ambiente de testes para realização de experimentos que objetivavam a aquisição de informações referentes ao desempenho da rede. Através das informações adquiridas com os testes efetuou-se uma análise para medir o desempenho da rede com os diferentes padrões de segurança existentes na atualidade.

#### A. Ambiente de Testes

Os equipamentos utilizados para a realização dos testes foram:

- 1 Roteador Wireless D-Link modelo DI-524;
- 2 microcomputadores AMD Phenon II X2, 3.00 Ghz, 2 GB RAM, com interface de rede Wireless Encore ENLWI-G2 (RTL8185).

O software utilizado para geração de tráfego de rede para os testes foi o Rude e Crude Versão 0.62 disponível sob a licença GPL versão 2. O Rude serve para geração de pacotes UDP (*User Datagram Protocol*) na rede, o Crude serve para a recepção e coleta das informações dos pacotes na outra extremidade da rede.

Nos microcomputadores realizou-se uma nova instalação do sistema operacional Linux Ubuntu 10.04 LTS 32 bits, ambos configurados igualmente.

Foi utilizada a topologia do tipo infraestrutura para a realização dos testes, a qual necessita de um concentrador central (*Access Point - AP*), sendo que o roteador wireless foi posicionado entre os dois microcomputadores distante 1 metro, conforme Figura 2.

Foram realizados testes para a rede sem segurança (*OPEN*) e para os padrões de segurança WEP, WPA e WPA2 em 1, 2, 4 e 8 minutos, os quais foram realizados três coletas para cada um dos tempos de testes realizados. Sendo que foram utilizadas chaves de maior tamanho



Figura 2. Estrutura da Rede do Ambiente de Testes

possível para obter a maior utilização de recursos de *hardware*, pois quanto maior a chave de criptografia maior será a utilização de *hardware* no processo de criptografia como um todo. Os tempos de testes foram acrescidos desta maneira para indicar possíveis discrepâncias nos testes. Sendo que todos os testes foram realizados dentro da mesma sala para obter as características desejadas sem interferências diferentes.

#### B. Desempenho em Redes sem Fio

Para análise do desempenho na rede experimental foram utilizadas métricas para representar o desempenho da rede.

1) *Throughput*: O *throughput* em uma rede de computadores pode ser definido como a vazão da rede, ou seja, é a capacidade total de um canal de transmissão processar e transmitir em um determinado intervalo de tempo.

2) *Delay*: É a medida de quanto tempo irá demorar para um pacote ir de um computador a outro. É interessante medir o *delay* máximo e o médio para as redes de computadores, através dessas medidas poderá ser conhecido o atraso na propagação dos pacotes na rede.

3) *Jitter*: O *jitter* em uma rede de computadores pode ser definido como o tempo entre a chegada dos pacotes. O *jitter* médio de uma rede de computadores é a variação do tempo entre a chegada de uma série de pacotes.

#### IV. RESULTADOS

A média de atraso está representada pelo gráfico da Figura 3, através desta pode-se observar que a média de atraso aumenta conforme aumenta o padrão de segurança. Pode-se observar também que a maior diferença destas médias, considerando os padrões de segurança disponibilizados, está entre o padrão WEP128 para o padrão WPA-PSK TKIP, com aumento cerca de 25% (24,67%) em média para os quatro tempos de teste.

A Figura 4 mostra o *jitter* médio em função do tempo de coleta para cada padrão de segurança nos diferentes tempos dos testes efetuados. Através da figura pode-se observar que o *jitter* médio nos testes de 1 minuto apresentaram um valor elevado em comparação ao tempo de 2 minutos, em todos os padrões de segurança com exceção da rede sem segurança (*OPEN*), isto ocorre devido a instabilidade inicial da rede, pois a mesma necessita de alguns instantes para estabilizar.

Através da Figura 5 pode-se observar o *jitter* máximo para os padrões de segurança em cada teste. Através dos resultados obtidos pode-se perceber que esta métrica de

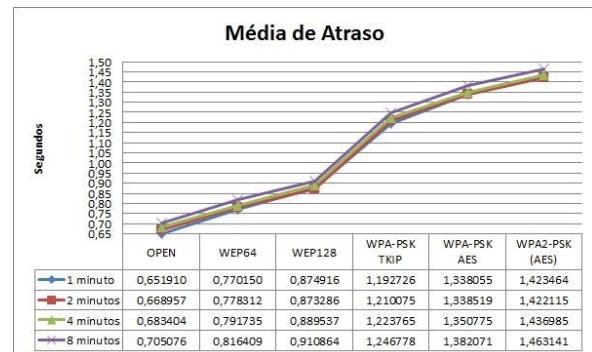


Figura 3. Média de atraso para os padrões de segurança em função do tempo de coleta

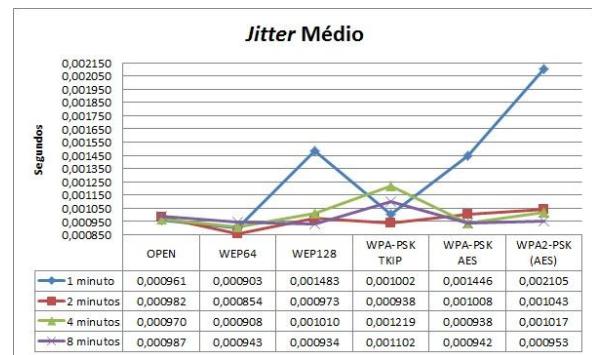


Figura 4. Jitter médio para os padrões de segurança em função do tempo de coleta

desempenho em redes de computadores é bem instável, pois em certos momentos da conexão a rede pode sofrer degradação, uma vez que em se tratando de redes sem fio diversos fatores podem interferir na estabilidade da rede, causando com isso picos no *jitter*, e a métrica do *jitter* máximo traz os maiores valores obtidos em termos de atraso na chegada dos pacotes (*jitter*).

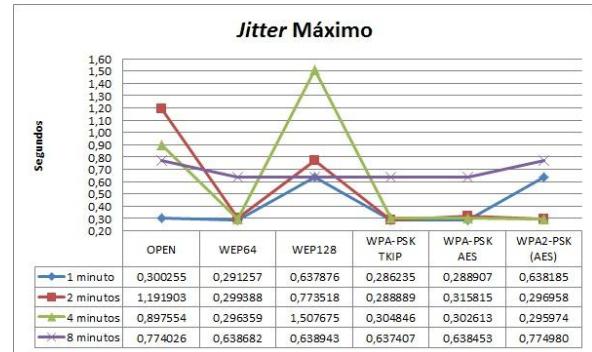


Figura 5. Jitter máximo para os padrões de segurança em função do tempo de coleta

Em relação à vazão da rede, pode-se observar que esta diminui conforme aumenta a segurança, principalmente do padrão WEP128 para o padrão WPA-PSK TKIP, conforme a Figura 6. Através da figura também é possível concluir

que a vazão na rede no tempo de testes de 1 minuto é inferior a tempos de testes maiores. Isto acontece pelo mesmo motivo descrito anteriormente na análise do jitter médio, que a rede é muito instável no começo de sua transmissão.

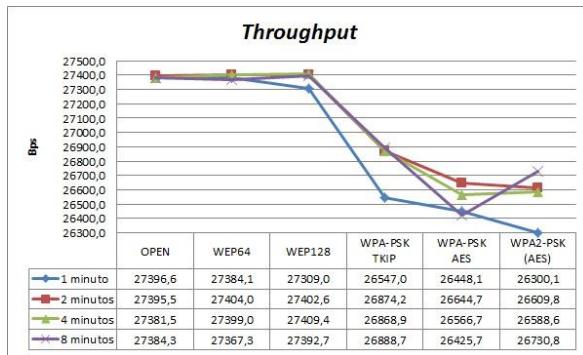


Figura 6. Throughput para os padrões de segurança em função do tempo de coleta

Na Figura 7 pode-se observar a média dos pacotes perdidos por minuto para todos os tempos de testes. Através destes dados pode-se comprovar a instabilidade da rede no inicio da transmissão, pois aumentando o tempo dos testes percebe-se que a quantidade de pacotes perdidos por minuto diminui para praticamente todos os padrões de segurança.

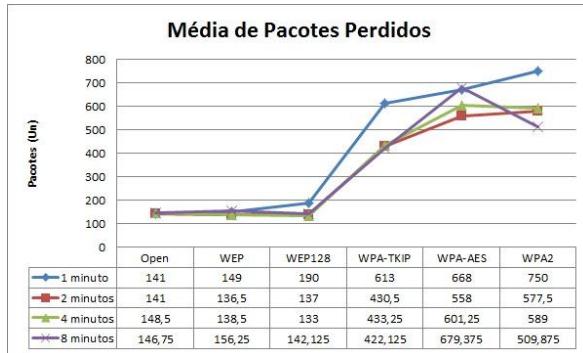


Figura 7. Média de pacotes perdidos em função do tempo de coleta

## V. CONCLUSÕES

Através dos experimentos e análises efetuadas neste trabalho pode-se verificar que a criptografia AES e o padrão WPA2-PSK possuem menor desempenho e maior segurança, conforme Amaral e Maestrelli [6] enfatiza, para a rede. Podem ser utilizados em redes com um alto requisito de confiabilidade, as quais necessitem transferir informações críticas. Pode-se, também, avaliar que os padrões de segurança OPEN, e WEP, com suas derivações, são as melhores opções em redes que necessitem um alto desempenho, sem requisitos significativos quanto a segurança.

Este trabalho demonstrou que as redes sem fio com maior segurança tendem a ter um desempenho inferior

as redes abertas ou com pouca segurança, pois as mesmas necessitam de um maior poder de processamento em função dos algoritmos de criptografia utilizados. Isto deve-se ao fato de o padrão WEP e WPA-PSK TKIP utilizarem como algoritmo de criptografia o RC4, sendo que segundo Amaral e Maestrelli [6] estes padrões de segurança foram desenvolvidos para os equipamentos já existentes. Já no padrão WPA-PSK AES e WPA2 utiliza-se como algoritmo de criptografia o AES, segundo Amaral e Maestrelli [6] a incorporação do AES foi devido a seu alto nível de segurança, sendo que o padrão WPA2 foi desenvolvido pensando na maior segurança possível para as redes sem fio 802.11. Com base nestas informações, é possível determinar a configuração mais adequada aos requisitos no momento da implantação de uma rede.

Também através deste trabalho pode-se comprovar o que foi descrito na subseção D da seção II, sobre a necessidade de maior processamento nos padrões de segurança que utilizam a criptografia AES, para os padrões que utilizam TKIP e o padrão WEP, visto que através da análise desenvolvida na seção dos Resultados (Seção IV), pode-se detectar um aumento da média de atraso e diminuição da vazão da rede.

Para trabalhos futuros, poderá ser analisado o desempenho em redes *ad hoc*, ou fazer variações na rede de infraestrutura, podendo também obter as informações de hardware do Access Point, ou dos equipamentos presentes na rede ou ainda comparar equipamentos com diferentes hardwares para determinar o quanto isto influencia na transmissão. Outro trabalho interessante é analisar o desempenho com diferentes padrões de redes 802.11x. Para estes trabalhos sugere-se a utilização de tempos de testes superiores a dois minutos, uma vez que a rede apresenta bastante instabilidade em intervalos de tempos de até dois minutos.

## REFERÊNCIAS

- [1] W. Stallings, *Criptografia e Segurança de redes: Princípios e Práticas*, 4th ed. São Paulo: Pearson Prentice Hall, 2008.
- [2] C. Suzin, “Análise de desempenho de protocolos de criptografia em redes sem fio,” *Monografia (Graduação em Sistemas de Informação) - Universidade de Caxias do Sul*, p. 70 f., 2007.
- [3] E. Barka and M. Boulmalf, “Impact of encryption on the throughput of infrastructure wlan ieee 802.11g,” *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pp. 2693–2697, 2007, disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4224745&isnumber=4224245>>. Acesso em: 29 Ago. 2012.
- [4] T. Thomas, *Segurança de Redes: Primeiros Passos*. Rio de Janeiro: Editora Moderna Ltda, 2007.
- [5] T. F. G. Caixeta, *Entendendo a Segurança nas Redes sem Fio. Revista Segurança Digital*, 4th ed., 2012, disponível em: <<http://segurancadigital.info/>>. Acesso em: 10 jun. 2012.
- [6] B. M. Amaral and M. Maestrelli, *Segurança em Redes Wireless 802.11*, 2004, disponível em: <<http://www.docstoc.com/docs/27567161/Segurança-em-Redes-Wireless-80211>>. Acesso em: 30 jun. 2012.

---

## IV

# Computação em Nuvem

---



# PyCloud: Compartilhamento em nuvem local

Jerônimo Feijó Noble da Rosa, Msc. Eduardo M. Monks

Faculdade de tecnologia Senac Pelotas

{kastanho,emmonks}@gmail.com

**Resumo**—Com o crescimento da capacidade de armazenamento dos discos rígidos e com o seu custo caindo na mesma proporção, novos computadores já contam com HDs de grande capacidade. A ideia desse projeto é desenvolver uma solução que possa utilizar o espaço que antes estaria ocioso para realizar uma distribuição de arquivos entre diversos computadores criando uma nuvem local para aumentar disponibilidade e a redundância dos arquivos.

## I. INTRODUÇÃO

Nos últimos anos muito tem se falado sobre computação em nuvem e existem várias aplicações de compartilhamento de arquivos, processamento de dados e fornecimento de serviços específicos [5]. Um dos pontos positivos apresentados para o uso de computação em nuvem é a ausência de preocupação com investimento em infraestrutura, manutenção e segurança dos dados. Essas tarefas ficam a cargo do provedor de serviço contratado [3]. Para algumas organizações é um grande problema entregar a terceiros dados sensíveis, pois essas informações podem se perder, vazar ou ficarem indisponíveis por qualquer razão [2] [6].

Muitas organizações adotam outra abordagem que é adquirir a infraestrutura necessária para a própria rede local, mas nesse caso quanto maior a carga de dados mais desempenho será exigido da infraestrutura para manter essas informações disponíveis, confiáveis e seguras. Com isso é necessário à aquisição de servidores de alto desempenho que são extremamente caros e deixam a organização refém de uma tecnologia específica [1].

Se for analisada a infraestrutura existente em uma rede é possível identificar que existem PCs que são utilizados para gerar ou acessar os dados e os dispositivos de rede como servidores, roteadores, *switches* entre outros que fornecem suporte para os usuários terem disponibilidade dos serviços utilizados na organização. Considerando que a capacidade do hardware para computadores desktop muitas vezes é subutilizado e tendo em vista que o custo de aquisição é relativamente baixo. Haja visto que a capacidade de armazenamento vem crescendo nos últimos anos e que exista recurso de armazenamento não aproveitado nos equipamentos, surgiu a ideia de criar uma solução para distribuir arquivos entre computadores abstraindo do usuário para onde os dados estão sendo enviados e armazenados fornecendo alta disponibilidade e tolerância a falhas.

## II. PYCLOUD

Esse sistema foi desenvolvido para permitir que dados sejam replicados entre diversos *hosts* na rede local aproveitando assim o espaço que em outra situação estaria ocioso. Por essa razão, o PyCloud fornece alta disponibilidade e redundância dos dados. Como cada instância do PyCloud mantém o índice completo de arquivos disponíveis, não existe um ponto único de falha. Para a recuperação dos dados bastaria apenas ter 50% dos membros ativo em redes menores (2-10 *hosts*) e no mínimo 20% em rede maiores (200 - 500 *hosts*). No decorrer do artigo será demonstrado quais procedimentos foram realizados para atingir esses valores.

Esse aplicativo foi desenvolvido em Python por isso o nome PyCloud, a tradução seria uma nuvem em python. Foi escolhida essa linguagem por ser multiplataforma, possuir muitas funções já prontas e uma documentação bem completa [4].

O PyCloud foi dividido em dois módulos, o módulo núcleo que controla todas as funcionalidades e a alteração só pode ser realizada através de um arquivo de configuração enquanto o módulo interface do usuário que é o programa que interage o usuário pode ser modificada de acordo com a necessidade do cliente. A interface do usuário se comunica com o núcleo através da comunicação entre processos e da API.

O aplicativo trabalha com comunicação *UDP* e *TCP* sendo que o primeiro utiliza três formas de operação *Multicast* ou *Broadcast* para comunicação com todos os membros e *unicast UDP* para troca individual de informação impedindo tráfego e processamento desnecessários.

O envio de arquivos é realizado através do *TCP*, devido ao comportamento do protocolo em relação a perdas e congestionamento.

Foi implementado também o versionamento de arquivos permitindo enviar arquivos com o mesmo nome, mas com conteúdos diferentes. Outra funcionalidade é a redução de desperdício impedindo que arquivos com mesmo conteúdo sejam enviados mais de uma vez.

O PyCloud fornece certo nível de segurança ao compactar, dividir em pedaços e em seguida distribuir entre diversos *hosts* onde são armazenados com um nome baseado em uma *hash*. Entretanto, isso não impede que um usuário não autorizado possa recuperar o arquivo caso ele tenha acesso à base de dados. Nesta versão do PyCloud, não foi implementado nenhum tipo de criptografia.

### III. DESENVOLVIMENTO

Nas próximas subseções serão explicados os funcionamentos dos módulos que foram propostos para esse projeto.

#### A. Localização de host

Diferentemente da Internet que necessita especificação de um ponto para que os *hosts* se encontrem, as redes locais permitem a localização de forma descentralizada através da utilização de mensagens *multicast* ou *broadcast*.

A função de localização no sistema PyCloud envia mensagens periódicas para informar aos outros membros da nuvem que ele está ativo, além de fornecer as informações de endereço IP, espaço disponível no disco local e outros dados.

Quando o PyCloud é iniciado esse faz um anúncio ao qual todos os membros ativos respondem com um anúncio pessoal, isso é feito com o objetivo de agilizar a convergência. Depois da primeira mensagem todas as demais não recebem resposta. Essas mensagens são utilizadas para manter o registro na tabela dos *hosts* remotos.

#### B. Distribuição

Este módulo exerce uma das funções mais importantes no PyCloud, pois realiza a distribuição dos pedaços de um arquivo e depois indica para qual *host* este será enviado.

Para definir a quantidade de *hosts* que devem receber um determinado pedaço utiliza-se a equação da figura 1.

$$X = 70 - \epsilon * \theta / 100$$

Figura 1. Exemplifica como funciona a taxa de Redução

Onde  $\epsilon$  (primeira variável) é a variável de redução que permite reduzir a quantidade de *hosts* que vão receber um determinado pedaço essa variável é definida através do arquivo de configuração.  $\theta$  (segunda variável) por sua vez representa a quantidade de *hosts* ativos no momento. Utilizando essa equação é possível garantir o crescimento da rede sem afetar a quantidade de espaço ocupado e o desempenho dela. Depois de finalizar a definição de quantos *hosts* devem receber um determinado pedaço, esses são distribuídos de forma aleatória, mas garantindo que cada membro ativo receba a quantidade de pedaços necessários.

#### C. Transferência de arquivos

O processo de envio é responsabilidade de várias *threads*, uma por pedaço. Cada pedaço tem a sua lista de endereços IPs que serão destinos para onde devem ser os dados devem ser enviado. Essa lista deve ser percorrida de forma sequencial e assim que um destino receba o conteúdo ele é marcado como uma origem em potencial e esse *host* já estará apto compartilhar os dados. Realizando isso permite distribuir a carga entre vários *hosts* ao invés de sobrecarregar um único. Quando é necessário baixar um arquivo, uma mensagem é enviada, todos os *hosts* que possuem algum pedaço desse arquivo respondem

com o nome e número de pedaço que está armazenado localmente.

#### D. Controle de versões e redução de desperdício de dados

Um dos problemas que pode-se encontrar no armazenamento de dados é o gasto desnecessário de espaço com arquivos duplicados. Para solucionar esse problema foi implementado no PyCloud uma função para evitar que conteúdos duplicados sejam enviados para a nuvem. A tabela de pedaços utiliza como referência a *hash* do arquivo, permitindo assim a criação de vários registros sem que as informações sejam duplicados no sistema de arquivos.

Outra funcionalidade que pode ser utilizada é o versionamento de arquivos. Este é realizado ao se tentar o envio de um arquivo com o mesmo nome e no mesmo diretório de um já existente. O nome permanecerá o mesmo, mas novos dados serão enviados mantendo os anteriores.

### IV. TESTES REALIZADOS

Nesta seção estão descritos os cenários que foram utilizado para testar as funcionalidades do PyCloud. Inicialmente, foi feita a análise do protocolo de comunicação que envolve a comunicação da aplicação. Neste teste, realizado durante 10 minutos em uma rede com um número crescente de *hosts* reais utilizando o cenário 1 da figura 2. Foi capturado o tráfego do protocolo em transmissões em *multicast* e *unicast*. Com isso foi possível precisar a quantidade de pacotes por segundo, tamanho médio de pacotes e a quantidade de dados trafegados na rede. Para

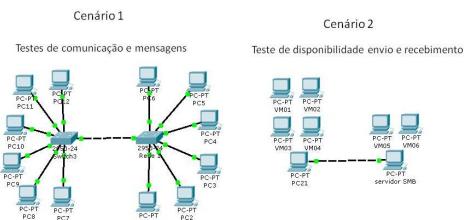


Figura 2. Mostra a estrutura utilizada nos testes.

os testes de operação do PyCloud foram utilizadas as opções de envio e recebimento de dados e depois comparados com o serviço de compartilhamento mais comuns, esses testes foram realizados com dois computadores reais utilizando máquinas virtualizadas (cenário 2 da figura 2). Foi levado em consideração fatores como disponibilidade, redundância e necessidade de espaço em disco.

### V. RESULTADOS

Essa seção inicia apresentando o resultado gerado na análise do protocolo de comunicação na tabela I.

É visível que ao se aumentar a quantidade de *hosts* nessa rede o número de pacotes trocados também crescerá, mas vale lembrar que o PyCloud foi projetado e desenvolvido para ser utilizado em redes locais e que as mensagens estarão contidas em um único domínio de *broadcast*, não sendo propagadas em outras redes. As mensagens

Tabela I  
TIPOS DE MENSAGENS

hosts	Pacotes trocados	Tamanho total (bits)	Média pacotes/sec	Média bytes/sec	Tempo (sec)
3	187	14120	0,309	23,338	605
4	388	29392	0,611	46,312	634
6	600	45600	0,885	67,226	678
8	795	60420	1,175	89,263	676
12	1153	65535	1,773	134,595	650

de anúncio são relativamente pequenas com tamanho de 75 bytes, as demais também tem um tamanho semelhante exceto pela mensagem de informação de arquivo que tem 11 campos e é gerado baseado no registro de um arquivo, então não é possível precisar exatamente o seu tamanho. todas as outras mensagens *UDP* podem ser vistas na tabela II.

Tabela II  
TIPOS DE MENSAGENS

Tipo de mensagem	Tamanho (bytes)	Tipo
Anúncio de <i>host</i>	75	<i>broadcast</i>
Solicitação de arquivo	87	<i>broadcast</i>
Resposta de arquivo	67	<i>unicast</i>
Alteração de registro	Dinâmico	<i>broadcast</i>

Outro teste realizado foi de verificar o tempo de convergência de uma rede utilizando o PyCloud. Esse teste foi feito em um *host* que recém havia se associado a nuvem e depois foi testado quanto tempo seria necessário para que o mesmo fosse removido da tabela de *host* e os resultados são mostrados na tabela III.

Tabela III  
TABELA DE CONVERGÊNCIA

Número de hosts	Convergência (sec)	Remoção da tabela (sec)
3	3	73
4	3	93
6	3	68
8	3	70
12	3	72

Ao analisar as capturas de tráfego constatou-se que o *host* que já estava na rede e recebera o anúncio acrescentou o novo computador em sua tabela quase que imediatamente. No caso do *host* que acabara de entrar na rede, o tempo foi menor ou igual a 3 segundos considerando que a sua tabela estava limpa e tinha que receber a atualização dos demais. O tempo de remoção leva duas vezes e meia mais do que o tempo de anúncio (30 segundos). O resultado apresentado fica perto desse tempo esperado.

Ao capturar as mensagens de resposta dos anúncios ficou constatado que a diferença de tempo entre o recebimento da mensagem número 1 de cada *host* tem em média 0,0008 segundos. A média de tempo entre as 3 mensagens recebidas de um mesmo *host* fica em torno de 2,67 segundos, a figura 3 mostra a variação de tempo entre essas mensagens.

O resultado está dentro do esperado, pois é utilizado um tempo aleatório entre as mensagens para que um *host* não



Figura 3. Tempo entre pacotes enviados por um mesmo *host*.

fique sobrecarregado com inúmeros anúncios recebidos ao mesmo tempo dos demais membros da nuvem.

Para testar o processo de distribuição foi criado um programa que adiciona o número de membros participantes e a quantidade de pedaços que deveriam ser criados. A partir do resultado é testada a disponibilidade dos dados em diferentes situações até que seja encontrado o menor valor possível sem falhas. Na tabela IV mostra o resultado dos testes para a divisão dos arquivos em 10 pedaços.

Tabela IV  
TESTE DE DISPONIBILIDADE DE DADOS

Total de Hosts	Min. de Hosts necessários	% de hosts ativos necessários
3	2	66
10	5	50
20	7	35
50	9	18
100	10	10
500	38	7
1000	70	7

Para chegar a o resultado foram feitos 10 blocos com 100 testes cada, para que o arquivo fosse considerado recuperável todos os blocos deveriam retornar sucesso, caso contrário era aumentada a porcentagem de distribuição e testado novamente até que fosse recebido o retorno esperado.

Antes de fazer a comparação entre com as outras aplicações é necessário apresentar o atraso que é adicionado no momento anterior ao envio e posterior ao recebimento de arquivos que envolve a montagem e descompactação. A tabela V mostra esse tempo com arquivos de diversos tamanhos.

Tabela V  
ATRASO ADICIONADO NO ENVIO E RECEBIMENTOS DE ARQUIVOS

Tamanho (MB)	Envio (seg)	Recebimento (seg)
3	2	1
10	7	4
50	16	7
100	72	40
150	107	55
200	146	75
750	632	432

Mesmo que seja acrescentado um tempo adicional na transmissão dos arquivos as funções que estão sendo executadas são extremamente necessárias para a operação do sistema PyCloud. Como os pedaços estão sendo arma-

zenados em ambientes não seguros a informações podem ser corrompidas ou alteradas, portanto é necessário que se utilize uma função que faça a *hash* do arquivo. Para o armazenamento local e até na transmissão é utilizada a compactação para reduzir o tráfego de rede e o espaço ocupado em disco e, por último e mais importante, a divisão de arquivos é o modo no qual o PyCloud baseia sua funcionalidade.

No teste de disponibilidade e espaço ocupado, foi utilizada uma rede com o tamanho de 4 *hosts* e um arquivo de 10MB. Essa comparação utilizou serviços comuns de compartilhamento de arquivos em redes locais. Os resultados estão disponibilizados na tabela VI.

Tabela VI  
TABELA DE COMPARAÇÃO ENTRE PROTOCOLOS

Protocolo	Disponibilidade	espaço ocupado/ <i>host</i>
PyCloud	Varia com o tamanho da rede	7 MB
Samba	Necessidade de servidores replicados	10 MB
FTP	Necessidade de servidores replicados	10 MB
NFS	Necessidade de servidores replicados	10 MB

O teste mostrou que ao parar o serviço de compartilhamento de arquivos não foi mais possível realizar a recuperação dos dados nas aplicações testadas, como já era esperado. No sistema PyCloud os dados permaneceram disponíveis com apenas dois hosts ativos.

O último teste realizado foi para verificar em redes com tamanhos crescentes a equação de redução da figura 1 e os resultados podem ser vistos na tabela VII.

Tabela VII  
TABELA DE DISTRIBUIÇÃO DE PEDAÇOS

Número de hosts	% pedaços/host	Num pedaço/host	Espaço ocupado (MB)/host
menor que 3	100%	10	10
3	70%	7	7
4	70%	7	7
5	69%	6	6
10	69%	6	6
20	68%	6	6
30	67%	6	6
50	65%	6	6
100	60%	6	6
150	55%	5	5
200	50%	5	5
500	20%	2	2
750	10%	1	1
1000	10%	1	1

É possível verificar que quanto maior a rede menor a distribuição de pedaços entre eles. Isso é feito primeiramente para não haver tráfego excessivo na rede e segundo para não ocupar uma quantidade de espaço desnecessária. Claro que isso implica diretamente na disponibilidade, mas como foi visto anteriormente na tabela de disponibilidade (tabela IV) em quase todos os casos é necessário que tenho no máximo 50% dos *hosts* ativos para recuperação dos dados.

## VI. CONCLUSÕES E PROJETOS FUTUROS

O PyCloud garante a disponibilidade dos dados proporcionalmente a quantidade de hosts ativos, enquanto os outros serviços testados (Samba, FTP, NFS) necessitam de aquisição de hardware e software com o aumento de número de hosts da na rede. Para projetos futuros serão desenvolvidas outras funcionalidades, tais como capacidade de recuperação de falhas, criptografia e fazer o balanceamento e a redistribuição dos dados entre os hosts.

## REFERÊNCIAS

- [1] Chaves, S.(2011) A Questão dos riscos em ambientes de computação em nuvem. *Universidade de São Paulo*.
- [2] Dahbur,K., Mohammad, B., Tarakji A.(2011). *A Survey of Risks, Threats and Vulnerabilities in Cloud Computing*. School of Engineering and Computing Sciences New York Institute of Technology Amman
- [3] Engates J. (2012) Small businesses harness the power of cloud computing. Disponível em: <<http://www.bbc.co.uk/news/business-17222816>>. Acesso em: jun 2012.
- [4] Goerzen, J. (2010) Fundations pf Python Network Programming. 1 Edição. Apress.
- [5] Greengard, S. (2010) Cloud Computing and Developing Nations. *communications of the acm*.
- [6] Weber, T. (2012) Cloud computing after Amazon and Sony: ready for primetime?. Disponível em: <<http://www.bbc.co.uk/news/business-13451990>>. Acesso em: jun 2012.

# **Computação em Nuvem com Google Apps for Education: o Caso do Núcleo de Ciência da Computação da Universidade Federal de Santa Maria**

Eder John Scheid, Leandro Hundertmarck Minato  
Benhur de Oliveira Stein, Andrea Schwertner Charão  
Núcleo de Ciência da Computação  
Universidade Federal de Santa Maria  
{eder, minato, benhur, andrea}@inf.ufsm.br

**Resumo**—Este artigo apresenta um relato de caso de adoção da plataforma Google Apps for Education em uma instituição de ensino. Esta plataforma oferece uma gama de aplicativos em nuvem SaaS (*Software as a Service*), trazendo várias facilidades para membros da instituição, mas também alguns desafios para administradores. Ao longo do artigo, descreve-se o processo de migração para nuvem ocorrido em 2007, o estado atual do domínio migrado e algumas oportunidades a serem exploradas.

## I. INTRODUÇÃO

Observa-se atualmente um aumento na utilização de serviços baseados no paradigma de computação em nuvem, onde os serviços ficam disponíveis em qualquer parte do mundo. Muitas empresas já oferecem serviços na nuvem, sendo o Google um exemplo delas. Sua plataforma de SaaS (*Software as a Service*), conhecida como Google Apps, oferece um conjunto de ferramentas tanto para empresas como para instituições educacionais. Para este último caso, existe o Google Apps for Education, que oferece este conjunto de ferramentas a instituições educacionais sem cobrança alguma e sem anúncios.

No final do ano de 2007, o Núcleo de Ciência da Computação (NCC) da Universidade Federal de Santa Maria (UFSM) passou a adotar a plataforma Google Apps for Education. O NCC responde pelo domínio inf.ufsm.br e oferece vários serviços a alunos, professores e funcionários ligados a alguns cursos de graduação e pós-graduação na área de Informática na UFSM. A principal motivação, na época da migração, foi substituir o serviço de e-mail que era gerenciado localmente e demandava muito esforço da equipe. Desde então, o gerenciamento dos serviços em nuvem vem sendo realizado por uma equipe de alunos e professores, utilizando as interfaces oferecidas pelo Google Apps for Education. O uso dessa plataforma de SaaS trouxe benefícios para usuários e administradores, sendo que, para esses últimos, surgiram também alguns novos desafios e oportunidades.

Este artigo tem por principal objetivo relatar a experiência adquirida pela equipe do NCC com o Google Apps for Education, oferecendo subsídios para outras instituições que planejem trilhar um caminho semelhante. O restante do artigo está organizado como segue. A seção 2 apresenta a plataforma Google Apps for Education, com seus recursos e objetivos. A seção 3, por sua vez, apresenta o Núcleo de Ciência da Computação da UFSM, com

informações sobre seus serviços oferecidos ao domínio inf.ufsm.br. A seção 4 descreve o processo de migração e a experiência adquirida com a administração da plataforma nos últimos anos, incluindo observações sobre problemas enfrentados atualmente. Por fim, a seção 5 apresenta algumas oportunidades a serem exploradas por administradores e a seção 6 apresenta algumas considerações finais.

## II. GOOGLE APPS FOR EDUCATION

A plataforma Google Apps [1] oferece vários aplicativos em nuvem, incluindo correio eletrônico (Gmail), mensagens instantâneas (Google Talk), agenda (Google Calendar), edição colaborativa de documentos (Google Docs) e sites Web (Google Sites). Esta plataforma possui uma versão voltada para empresas (Google Apps for Business), atualmente comercializada no valor mensal de US\$ 5 por usuário. Essa versão permite que a base de usuários no domínio da empresa seja gerenciada via Google Apps, com servidores Google respondendo aos serviços oferecidos a este domínio. O Google Apps for Education [2] é uma edição da plataforma Google Apps com os mesmos recursos da Google Apps for Business, mas oferecida gratuitamente a instituições educacionais.

Com a plataforma Google Apps for Education, os aplicativos em nuvem têm beneficiado muitas instituições em vários níveis de ensino [3]. Além disso, muitos trabalhos têm discutido aplicações da plataforma Google Apps em instituições educacionais [4], [5], [6]. A versão gratuita ajuda entidades que não possuem recursos para instalar e manter tais serviços localmente, mas também constitui uma estratégia vantajosa para equipes de TI, que podem concentrar esforços em outros serviços voltados para atividades-fim das instituições de ensino.

A plataforma Google Apps for Education possui um amplo suporte para atender às dúvidas de usuários e administradores [7]. Para estes últimos, a plataforma oferece instruções detalhadas para inscrição no serviço e para gerenciamento de usuários e aplicativos. Dentre as facilidades para administradores, tem-se APIs administrativas que auxiliam na migração da base de dados LDAP ou Active Directory através da sincronização de diretórios [8]. Para uma gerência eficaz, o Google Apps for Education fornece ainda um painel de controle simplificado (figura 1) e gráficos com estatísticas (figura 2). De maneira geral,

isso ajuda os administradores, pois organiza melhor todas as funcionalidades oferecidas.

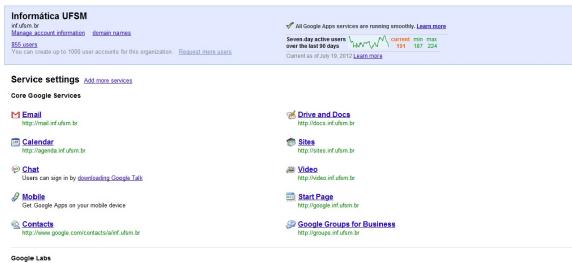


Figura 1. Painel de controle



Figura 2. Gráficos com estatísticas

### III. NÚCLEO DE CIÊNCIA DA COMPUTAÇÃO DA UFSM

O Núcleo de Ciência da Computação (NCC) da Universidade Federal de Santa Maria nasceu a partir de necessidades geradas por alunos e dos professores do primeiro curso de graduação na área de Informática da instituição, iniciado em 1990. Com esse curso, surgiu a demanda por laboratórios preparados com ferramentas essenciais para o ensino, pesquisa e extensão na área da computação. Tais laboratórios compõem e são gerenciados pelo NCC. Com o crescimento da área na instituição, o NCC hoje atende os cursos de graduação em Ciência da Computação (nova denominação do extinto curso de Informática) e Sistemas de Informação, cuja primeira turma ingressou em 2010. O funcionamento do NCC é garantido por um professor coordenador e uma equipe de alunos bolsistas, auxiliados por professores colaboradores. Isso permite que alunos de ambos os cursos participem da administração da rede do NCC e do gerenciamento dos laboratórios, oferecendo uma oportunidade de aprendizado e contato com tecnologias de rede.

Os serviços do NCC têm variado ao longo do tempo e hoje incluem, basicamente, o gerenciamento e o suporte a usuários (incluindo serviços para estes), o gerenciamento de serviços de rede (DNS, firewall, etc.) e servidores virtualizados, além do gerenciamento de hardware e software dos laboratórios didáticos destinados aos dois cursos. O

volume de usuários cresce regularmente no NCC, abrangendo professores e alunos desses cursos e também contas de egressos que são mantidas indefinidamente.

Durante alguns anos, o NCC manteve um serviço de Webmail para seus usuários, usando a ferramenta IMP [9]. Este serviço demandava o gerenciamento de vários servidores (servidor de e-mail, anti-spam, gerenciador de banco de dados, servidor Web, etc.) e era fonte de muitos problemas, pois precisava ser mantido sempre atualizado e disponível. Com a popularização de serviços de e-mail gratuitos, como o Google Mail, muitos usuários passaram a abandonar ou redirecionar o e-mail institucional. Neste cenário, o surgimento do Google Apps for Education mostrou-se uma alternativa viável e vantajosa para garantir a disponibilidade do serviço de Webmail a usuários do NCC, trazendo consigo outros serviços alinhados com as demandas do núcleo. Assim, em 2007, em uma iniciativa pioneira na UFSM, o NCC passou a utilizar esta plataforma, com um processo de migração apresentado na seção a seguir.

### IV. PROCESSO DE MIGRAÇÃO E ESTADO ATUAL

O primeiro passo para utilização do Google Apps for Education é a inscrição no serviço, fornecendo-se um nome de domínio (no caso, inf.ufsm.br) e um e-mail neste domínio. Nesta etapa, o Google verifica a validade da solicitação, para garantir que se trate de uma instituição de ensino e que o solicitante tenha permissões sobre o domínio. Esta etapa, no caso do NCC, foi relativamente rápida (em torno de 24 horas). Para ativar o serviço, e provando a permissão sobre o domínio, foi necessário colocar uma entrada no DNS, pois alguns nomes precisam ser redirecionados para servidores do Google. No total, este processo inicial durou poucos dias, para depois passar-se à próxima etapa, de migração dos usuários.

A criação do lote de usuários no Google Apps for Education, na época, foi feita fornecendo-se uma planilha com dados de cada usuário (nome, sobrenome, usuário e senha). Neste ponto, cada usuário passou a ter uma conta na nuvem, mas manteve sua conta local. As credenciais da conta na nuvem foram enviadas aos usuários. Antes que os serviços locais fossem desativados, desejava-se migrar os dados dos e-mails (mensagens) que encontravam-se armazenados localmente. Para isso, o Google oferece algumas opções [10]. Na época, utilizou-se a opção de transferência do conteúdo das caixas de mensagem do servidor IMAP local para a plataforma Google Apps for Education. Essa opção podia ser utilizada individualmente pelos usuários ou em lote pelo administrador do domínio. Inicialmente, colocou-se essa opção à disposição de cada usuário, para que cada um tivesse controle sobre opções de migração. Para aqueles que não realizaram a migração no prazo previsto, realizou-se a migração automática. A partir deste momento, desativou-se as contas locais. A etapa de migração dos dados (mensagens no servidor de e-mail) foi a que levou mais tempo, em torno de uma semana para cerca de 30 GB.

Após a migração, passou-se a utilizar a interface do

Google Apps for Education para gerenciamento de usuários, grupos e outras configurações do domínio na nuvem. Com o tempo, atingiu-se um limite de número de usuários e foi necessário solicitar mais contas, o que foi atendido prontamente. Atualmente, tem-se 855 usuários cadastrados no Google Apps for Education no domínio inf.ufsm.br. A cada ano, criam-se no mínimo 80 novas contas de usuários para calouros do curso de Ciência da Computação (no primeiro semestre) e do curso de Sistemas de Informação (segundo semestre), juntamente com grupos de discussão para cada nova turma.

A equipe de administração é configurada via painel de controle do Google Apps for Education, que permite definir papéis (Super Admin, Groups Admin, User Management Admin, etc.) e privilégios associados a cada papel (Create, Read, Rename, Move, Delete) (ver figura 3). Também é possível criar novos papéis com permissões personalizadas. Esta gama de opções favorece o trabalho em equipe e se alinha com o perfil do NCC, onde algumas tarefas de gerenciamento são realizadas por alunos.

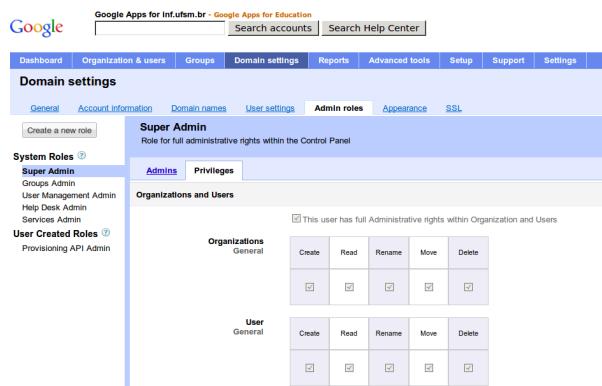


Figura 3. Papéis e privilégios

Como as tarefas de criação de usuários e grupos são recorrentes, o uso do painel de controle baseado na Web tem se mostrado pouco eficiente. Assim, tem-se estudado alternativas para automatizar o processo de criação das contas e também de grupos. Em particular, está sendo analisada a possibilidade de sincronização com a base LDAP local (necessária para acesso a serviços locais). Além disso, está sendo pensada uma nova forma de organização de grupos, identificados por nomes de disciplinas e compostos pelos professores e alunos associados às mesmas, o que tornaria mais eficiente a comunicação entre eles.

## V. OPORTUNIDADES

Uma das dificuldades que os professores encontram todo início de semestre é como criar grupos de discussão para suas disciplinas, visto que não há ainda uma ferramenta que os auxilie nisto. Atualmente, a criação de grupos é autorizada apenas à equipe administrativa, que analisa e atende às solicitações de usuários. No caso de professores responsáveis por disciplinas, alguns acabam por não utilizar este recurso, outros o fazem manualmente

requisitando o e-mail de cada um dos alunos e posteriormente criando grupos de contatos em suas agendas.

Como o Google oferece várias APIs, tanto administrativas quanto voltadas aos usuários, torna-se possível a criação de programas que automatizem rotinas antes executadas pelos próprios administradores de rede, diminuindo assim a demanda de tempo dos mesmos. Dentro do conjunto de APIs pode-se citar a Provisioning API [11] que permite gerenciar grupos e usuários. Implementações desta API estão disponíveis em várias linguagens como Java, Python, .NET e PHP. Com esta API, está sendo desenvolvido um programa que, a cada semestre, cria grupos de alunos matriculados em determinadas disciplinas, tomando como base uma lista de alunos obtida a partir do próprio sistema de informações da universidade. Com este programa, espera-se facilitar a interação entre alunos e professores nos cursos atendidos pelo NCC.

Com relação à criação de contas de usuários, atualmente realiza-se, a cada semestre, dois cadastros para cada nova turma de alunos: um cadastro na base LDAP local, necessária aos outros serviços oferecidos (acesso a máquinas dos laboratórios, acesso remoto por SSH, etc.), e um cadastro na plataforma Google Apps no domínio inf.ufsm.br. Este último é feito usando uma opção de cadastro em lote (*bulk upload users*) na plataforma. Com o serviço de sincronização de diretórios (Google Apps Directory Sync), é possível automaticamente adicionar, modificar ou remover usuários e grupos a partir de um servidor de diretórios LDAP. Esta sincronização pode ser programada para ocorrer de tempos em tempos, mantendo os cadastros equivalentes sem a necessidade de trabalho duplicado. No entanto, o uso deste serviço ainda está em análise, pois algum eventual problema de sincronização pode vir a comprometer o cadastro existente de usuários na plataforma.

## VI. CONSIDERAÇÕES FINAIS

Com migração do domínio inf.ufsm.br para a plataforma em nuvem Google Apps for Education, o Núcleo de Ciência da Computação conseguiu atingir o objetivo de ter vários serviços sendo prestados com uma disponibilidade muito alta, comparando-se com a disponibilidade que se tinha quando os mesmos eram providos pelos próprios servidores locais. O caso em questão corrobora a ideia de que plataformas em nuvem podem auxiliar entidades educacionais atingirem um nível qualidade técnica para comunicação interna sem grande esforço, o qual beneficia tanto alunos envolvidos quanto professores. Adicionalmente, a plataforma Google Apps for Education oferece serviços e APIs que, para alunos e professores em computação, permitem manter atividades de desenvolvimento local de customizações, trazendo benefícios para o ambiente de ensino-aprendizagem. Acredita-se que este relato de caso possa contribuir para que outros núcleos semelhantes ao NCC avaliem a possibilidade de migração de serviços para uma plataforma em nuvem.

#### REFERÊNCIAS

- [1] Google Inc., “Google Apps.” [Online]. Available: <http://www.google.com/apps/intl/en/index.html>
- [2] ———, “Google Apps for Education.” [Online]. Available: <http://www.google.com/apps/edu>
- [3] ———, “Success stories – Google Apps for Education.” [Online]. Available: <http://www.google.com/apps/intl/en/edu/customers.html>
- [4] K. Barlow and J. Lane, “Like technology from an advanced alien culture: Google apps for education at asu,” in *Proceedings of the 35th annual ACM SIGUCCS fall conference*, ser. SIGUCCS ’07. New York, NY, USA: ACM, 2007, pp. 8–10. [Online]. Available: <http://doi.acm.org/10.1145/1294046.1294049>
- [5] D. R. Herrick, “Google this!: using google apps for collaboration and productivity,” in *Proceedings of the 37th annual ACM SIGUCCS fall conference*, ser. SIGUCCS ’09. New York, NY, USA: ACM, 2009, pp. 55–64. [Online]. Available: <http://doi.acm.org/10.1145/1629501.1629513>
- [6] N. Sultan, “Cloud computing for education: A new dawn?” *International Journal of Information Management*, vol. 30, no. 2, pp. 109 – 116, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0268401209001170>
- [7] Google Inc., “Google Apps help.” [Online]. Available: <http://www.google.com/apps/intl/en/edu/customers.html>
- [8] ———, “Google Apps Directory Sync.” [Online]. Available: <http://support.google.com/a/bin/answer.py?hl=en&answer=106368>
- [9] The Horde Project, “IMP.” [Online]. Available: <http://www.horde.org/apps/imp/>
- [10] Google Inc., “E-mail migration options – Google Apps help.” [Online]. Available: [url{http://support.google.com/a/bin/answer.py?hl=en&answer=57920}](http://support.google.com/a/bin/answer.py?hl=en&answer=57920)
- [11] ———, “Google Apps Provisioning API.” [Online]. Available: <https://developers.google.com/google-apps/provisioning/>

# Serviço de Presença sobre uma Estrutura Gossip em Cloud

Peterson Wilges  
UFRGS  
pwilges@inf.ufrgs.br

Henrique Dalla Costa Lovison  
UFRGS  
henrique.lovison@inf.ufrgs.br

Sérgio Luis Cechin  
UFRGS  
cechin@inf.ufrgs.br

Taisy da Silva Weber  
UFRGS  
taisy@inf.ufrgs.br

Regina Moraes  
UNICAMP  
regina@ft.unicamp.br

**Resumo**— Este artigo traz uma implementação de um modelo de serviço de presença sobre uma estrutura *gossip* em *cloud*. Diferente da maioria dos serviços de presença existentes, este foi desenvolvido de forma totalmente distribuída. Através de disseminação epidêmica e a redundância inherente, o serviço é tolerante a falhas de comunicação e perda de mensagens. No cenário de avaliação desenvolvido, mostramos como acontece a propagação para diferentes números de nodos e como ajustes na estrutura *gossip* podem beneficiar a disseminação.

## I. INTRODUÇÃO

Um serviço de presença tem por objetivo gerenciar a presença de entidades em uma rede. Para ser eficiente e tolerante a falhas em um ambiente dinâmico com um grande número de nodos, é necessário que o serviço supere problemas de escalabilidade. Em nossa abordagem desenvolvemos um serviço de presença de nodos em uma rede através de uma estrutura *gossip*. Desta forma, criamos um modelo que segue uma abordagem diferente da sugerida nas normas da IETF através das RFCs 2778 e 2779[1][2]. Nosso serviço, diferente do modelo sugerido, é totalmente distribuído e facilmente escalável.

Os atuais serviços de presença existentes no mercado possuem uma abordagem centralizada. Em contrapartida, nossa aplicação é totalmente descentralizada, sendo cada nodo responsável por gerenciar quais as entidades presentes em uma determinada rede. Essa característica distribuída pode ser muito mais eficiente em termos de tempo de transmissão e de disponibilidade de serviço, como será demonstrado no cenário em que desenvolvemos.

O nosso modelo foi construído usando um protocolo de comunicação epidêmica entre seus participantes. Assim, cada notificação de presença enviada para o grupo interessado, que chamaremos de *cloud*, é retransmitida para outros membros do grupo, que por sua vez, retransmitem a mensagem para outros membros. O protocolo utilizado, chamado NeEM [3][4], provê mecanismos para a aplicação gerenciar o número ideal de nodos para transmissão (o qual chamaremos de *fanout*).

que o mesmo deve possuir para atingir uma alta probabilidade de entrega de uma mensagem. É fácil notar, desta maneira, que o protocolo é inherentemente redundante: fazendo uma analogia podemos compará-lo a uma fofoca sendo disseminada em um grupo de pessoas. Uma pessoa do grupo conta uma história para outro integrante, que por sua vez repassa a mesma para outros integrantes. Cada pessoa que ouvir a história, contará novamente para pessoas do grupo, e assim observamos que um indivíduo deve, provavelmente, escutar a mesma história mais de uma vez. Entretanto, o NeEM possui mecanismos que visam amenizar essas transmissões desnecessárias [4]. Também precisamos observar que a mensagem não deve ficar eternamente sendo disseminada entre os participantes, sendo essa retransmissão limitada por um parâmetro do protocolo chamado *time-to-live* (*TTL*). O TTL da mensagem determina o número de rodadas para a mensagem ser retransmitida.

Neste trabalho, além de demonstrar como estes mecanismos funcionam, aplicamos alguns testes que demonstram como a disseminação das informações acontece de maneira rápida. Desenvolvemos um cenário utilizando um único computador, já que a sincronização de eventos com base em um único relógio facilita o monitoramento do experimento de teste. Neste cenário, fizemos um nodo lançar uma notificação utilizando o NeEM, que por sua vez, faz a inundação para os outros nodos da rede. Desta maneira, analisamos o tempo de disseminação para um número diferente de nodos. Em uma segunda análise avaliamos o uso de diferentes *fanouts* que junto com o *time-to-live* e o número de nodos determinam a probabilidade de uma mensagem atingir toda a rede [5].

Este artigo está organizado como segue: Na seção II apresentamos uma descrição do protocolo utilizado bem como seu funcionamento. Na seção III descrevemos como funciona nossa aplicação. Na seção IV, descrevemos algumas arquiteturas de serviço de presença existentes. As simulações feitas foram descritas na seção V. E finalmente na seção VI foram apresentadas conclusões sobre os resultados obtidos e possíveis trabalhos futuros.

## II. DISSEMINAÇÃO EPIDÊMICA E PROTOCOLO NEEM

Como já mencionado, o nosso aplicativo faz uso da biblioteca Network-friendly Epidemic Multicast (NeEM), responsável pela disseminação epidêmica. O protocolo, também chamado de *gossip-based* ou protocolo probabilístico, faz uso de conexões TCP para gerenciar o conjunto de nodos com o qual cada ponto está conectado. Esse conjunto chama-se *overlay*, e é gerenciado automaticamente pelo NeEM.

A disseminação epidêmica funciona da seguinte maneira: cada nodo se conecta a um número de nodos  $k$ , formando o *overlay*. A cada mensagem recebida, o nodo retransmite para  $f$  (*fanout*) nodos a mensagem sendo  $f < k$ , ou seja, retransmite para um número menor do que as conexões estabelecidas em *overlay*. O número maior de conexões em *overlay* do que de *fanout* deve-se a uma redundância que permite maior tolerância a falhas bem como melhor gestão dos nodos. Caso um nodo falhar ou cair, outro pode receber a mensagem. Também em toda retransmissão nodos diferentes entre os  $k$  nodos são escolhidos ( $f$ ) e de períodos em períodos conexões novas são estabelecidas mudando a camada de *overlay*. Isso permite que usuários novos, bem como usuários que deixaram o *cloud*, sejam efetivamente ligados ou desligados do grupo.

O protocolo usado também possui mecanismos que visam amenizar altos usos da banda com transmissões desnecessárias, que é inherente a este tipo de protocolo *gossip*. Para tanto, o protocolo funciona sobre dois modos de operação. O primeiro chamado *eager* para quando a mensagem possui um *payload* pequeno, o modo prevê a transmissão da mensagem tão logo se dê seu recebimento. Já em contrapartida, quando houver um grande *payload*, o modo *lazy* entra em operação, mandando uma mensagem pequena aos nodos conectados, e esperando que aqueles que desejarem a mensagem a solicitem [4].

Protocolos *multicast* epidêmicos possuem um alto rendimento independentemente do número de nodos e falhas na rede. Também são escaláveis para um grande número de participantes. Entretanto, esses protocolos geram um grande volume de tráfego na rede devido à redundância.

## III. SERVIÇO DE PRESENÇA PINGCLOUD

Para fazer a gerência de quais nodos estão presentes no *cloud*, criamos um serviço de presença chamado PingCloud. Geralmente, junto a aplicativos de mensagens instantâneas, os serviços de presença têm por objetivo gerenciar entidades presentes da rede, que podem ser humanos, dispositivos ou simplesmente um aplicativo, como em nosso caso.

Para realizar essa gestão, fizemos uso de notificações, onde cada nodo que deseja informar sua presença manda uma mensagem para o *cloud*, informando sua presença. Essa notificação deve ser enviada em intervalos de tempo determinados por um parâmetro. Cada nodo mantém uma lista com os nodos presentes em determinado momento. O período que um nodo pode ficar na lista sem enviar notificação é determinado por um segundo parâmetro. O período que a lista é verificada, no nodo, para remoção de

nodos ausentes é determinado por um terceiro parâmetro. Se um dado nodo presente na lista não enviar uma nova notificação dentro do intervalo de atualização da lista, o mesmo deve ser retirada da lista de presença. O ajuste destes três parâmetros é essencial para o funcionamento do aplicativo. Contudo, para se conectar ao serviço, o nodo precisa conhecer ao menos um nodo presente no grupo.

No PingCloud, não há diferenciação entre tipo de disponibilidade tal como existem nos aplicativos de mensagem instantânea onde o usuário pode estar presente, ausente, ocupado entre outros. Por se tratar da presença ou não de um computador na rede, existem apenas duas possibilidades: estar presente ou não. Para informar a presença, um nodo simplesmente manda uma mensagem notificando a presença. Caso ele não estiver presente ou desejar sair, basta simplesmente não mandar notificações. Junto com a informação também é enviado o tempo em que a mensagem foi gerada para ordenação dos eventos.

É na manutenção da lista de presença que o nosso serviço possui sua grande diferença em relação aos outros. Neste sentido, a arquitetura utilizada no mercado hoje, possui uma abordagem centralizada, sendo um servidor central responsável por gerenciar a rede. Já em nossa arquitetura, cada nodo possui sua lista de presença. Essa lista começa vazia, e a cada mensagem recebida ou a cada período estabelecido, ela é atualizada, retirando os nodos que não mais mandaram atualização e inserindo novos nodos que desejam entrar na rede.

## IV. TRABALHOS RELACIONADOS

Descreveremos alguns exemplos de serviços de presença de código aberto. Como veremos, todos aqui descritos seguem uma abordagem centralizada.

### A. Extensive Message and Presence Protocol (XMPP)

O XMPP [6] é um protocolo aberto para troca de mensagem e informação de presença. Sua arquitetura segue uma abordagem centralizada, já que a lista de presença dos usuários fica em servidores que intermediam toda a comunicação entre os usuários. Todos os nodos (aplicações) são endereçados através de um identificador único. Dois usuários que querem notificar a presença e se comunicar devem cada um fazer uma conexão a um servidor XMPP. Os servidores se comunicam passando as devidas informações de presença e trocando mensagens entre esses dois usuários. Toda a comunicação é feita com trocas de mensagens em formato XML.

### B. SIMPLE

O SIMPLE [7] é um protocolo de serviço de presença e mensagem instantânea baseado no protocolo SIP (*Session Initiation Protocol*), que é um protocolo para iniciar, gerenciar e finalizar sessões multimídia como, por exemplo, aplicações de voz sobre IP. O SIMPLE possui três métodos adicionais ao SIP para troca de mensagem e informação de presença:

- *Subscribe*: método invocado quando um nodo quer receber alguma informação de presença.

- *Notify*: invocado para enviar informação de presença.
- *Message*: invocado para enviar uma mensagem.

A comunicação entre dois usuários pode ser feita por P2P, desde que os nodos se conheçam antecipadamente. Caso os usuários não se conheçam, eles fazem uso de um servidor central que contém a lista de presença e os respectivos endereços.

### C. Wireless Village(WV)

Fundado pela Ericson, Motorola e Nokia o WV [8] foi desenvolvido para prover um conjunto de especificações universais para mensagem instantânea e serviço de presença em dispositivos móveis. Usa uma arquitetura cliente-servidor, onde o cliente é qualquer terminal móvel e o servidor é o Wireless Village Server. O WV Server é responsável pelo gerenciamento de presença bem como outras funcionalidades, como troca de mensagens, gerenciamento de grupos e compartilhamento de arquivos.

## V. ANÁLISE SOBRE O SERVIÇO DE PRESENÇA PINGCLOUD

Para analisar o funcionamento do modelo de serviço de presença desenvolvido, construímos um cenário para demonstrar seu funcionamento. O cenário é construído através de um único computador que simula vários nodos. Um nodo envia uma notificação que se propaga para os outros nodos. Desta maneira, fizemos duas avaliações. Na primeira, desconsideramos o atraso de propagação e simplesmente fizemos a simulação para diferentes números de nodos. Na segunda, simulamos um atraso de propagação de 100ms para todos os pacotes através de uma ferramenta de injeção de falhas [9]. A ferramenta permite introduzir falhas de comunicação como perda e corrupção de mensagens e também atrasos em algumas ou todas as mensagens e tem sido usada para avaliação de protocolos de comunicação [10].

Simulamos uma rede de tamanho fixo de nodos e diferentes configurações de *fanout*.

De acordo com estes cenários foram feitas algumas simulações e seus resultados serão descritos nesta seção.

### A. Avaliação do tempo de disseminação de uma notificação de presença

A primeira avaliação foi feita para verificar em quanto tempo uma dada notificação de presença demora a chegar a todos os nodos presentes no *cloud*, desconsiderando o tempo de propagação. Para esta avaliação, criamos diferentes números de nodos presentes na nuvem para avaliar o serviço. Enviamos uma mensagem de um dos nodos e coletamos, através dos *logs* gerados, o tempo em que a mensagem chegou aos outros nodos. Os parâmetros do NeEM foram ajustados para *fanout* igual a 5 e com *time-to-live* igual a 6.

Para mandar uma única mensagem do nodo, ajustamos o parâmetro de intervalo de envio de notificação de maneira que o nodo não envie duas mensagens ou mais

durante o experimento. Assim determinamos um alto intervalo de envio.

Como nosso aplicativo faz uso de um protocolo de disseminação epidêmica, o tempo para disseminar a informação de presença não cresce linearmente com o aumento da rede. No primeiro momento, quando poucos nodos possuem a mensagem, há pouca redundância de transmissão, assim, a transmissão para  $f$  nodos é total, ou praticamente total. Em um segundo momento, quando uma grande quantidade de nodos já possui a notificação, há muita redundância, sendo os últimos *hops* menos eficientes, já que poucas transmissões serão de fato aproveitadas e a grande maioria, redundante, será descartada. Apesar de neste caso não se fazer necessária, a redundância é uma das características importantes em protocolos epidêmicos, pois o torna tolerante a falhas.

Para demonstrar a disseminação da mensagem no cenário descrito, realizamos a simulação para 15, 25 e 40 nodos, que foi o limite encontrado para um único computador, no qual o resultado não fosse prejudicado pelo grande número de *threads*.

Como observado anteriormente, é possível visualizar no gráfico mostrado na figura 1 que a notificação é disseminada rapidamente em sua fase inicial, e vai perdendo desempenho com o passar do tempo.

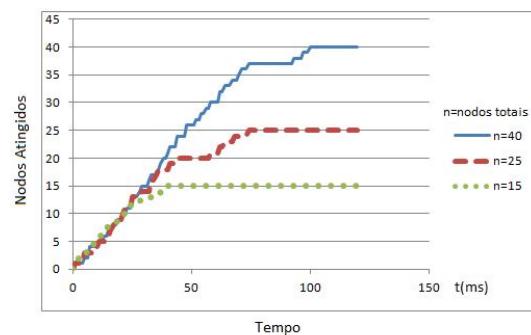


Figura 1 - Análise gráfica para diferentes números de nodos

### B. Avaliação do tempo de disseminação de uma notificação para diferentes fanouts

Nesta avaliação, foram realizados duas simulações, ambas com 30 nodos na rede. Na primeira, foi usado *fanout* igual a 3, e na segunda *fanout* igual a 7. Nesta simulação, utilizamos uma ferramenta chamada FIRMAMENT [9], para colocar um atraso de 100ms em todos os pacotes transmitidos.

Desta maneira, com os atrasos fixos, é possível analisar como acontece cada *round* de transmissão. Primeiro o nodo que deseja transmitir envia uma notificação para  $f$  (*fanout*) outros nodos na rede. Como nenhum destes possui esta notificação todas as transmissões são efetivadas. No segundo *round*, cada um dos  $f$  nodos que receberam a mensagem, retransmitirão a mesma para  $f$  nodos, que desta vez podem ser

redundantes. No terceiro e seguintes *rounds*, acontece o mesmo que no passo dois, com a diferença que quanto maior o *round* maior será a o número de transmissões redundante.

Agora, analisando a simulação para *fanout* 3, é possível observar que no primeiro *round* 3 notificações serão enviadas. No segundo, serão enviadas 9 notificações, havendo uma probabilidade de algumas mensagens serem redundantes. No terceiro, serão transmitidas 3 vezes o número de nodos que receberam a mensagem no *round* anterior, descartando aqueles que só receberam mensagens redundantes, já que estes nodos simplesmente descartarão esta mensagem. Desta maneira, é importante observar que se em uma rodada todas as mensagens transmitidas forem redundantes, todas serão descartadas e a propagação é encerrada.

Para *fanout* igual a 7, acontece analogamente a *fanout* igual a 3 com a diferença que cada nodo deve transmitir a mensagem para 7 outros nodos. É fácil observar assim, que a disseminação é mais rápida e menos rodadas serão necessárias.

De acordo com este entendimento, é possível observar na figura 2, os diferentes *rounds* de transmissão, separados por intervalos de aproximadamente 100ms. Vemos também, que para *fanouts* maiores um número menor de *rounds* é necessários para todos os nodos serem atingidos por uma notificação.

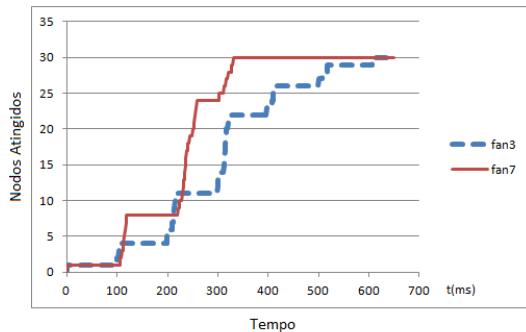


Figura 2 – Análise gráfica para diferentes valores de *fanout*

Convém enfatizar que a ferramenta de injecção de falhas foi usada neste cenário apenas para facilitar a avaliação. Neste caso o atraso inserido apenas emula uma situação de rede com atrasos naturais, não sendo considerada como uma situação de falha.

## VI. CONCLUSÕES E TRABALHOS FUTUROS

Podemos concluir que o modelo distribuído implementado neste trabalho mostra ser possível usar uma abordagem descentralizada para um serviço de presença. Nossa aplicação foi construída como um protótipo para uma futura aplicação mais elaborada deste serviço. Os resultados demonstraram que o uso do protocolo NeEM foi uma escolha oportunista para a disseminação multicast. Também é possível observar que o ajuste dos parâmetros do protocolo é fundamental para seu funcionamento correto e podem ser modificados pela aplicação de acordo

com as necessidades da rede. Por exemplo, numa rede com maior índice de perdas de pacotes podemos aumentar a redundância de mensagens e manter assim uma alta probabilidade de entrega.

Algumas limitações não foram trabalhadas nesta primeira versão da aplicação do serviço de presença, como mecanismos de segurança para um nodo entrar no *cloud*. Outro ponto importante a ser explorado é aprimorar o aplicativo, de modo que ele gerencie os parâmetros do protocolo NeEM em tempo de execução. Parâmetros como *fanout* e *time-to-live*, são relativos ao número de nodos presentes na rede, e desta forma devem ser gerenciados pela aplicação.

Para futuros experimentos, estamos planejando testes com um número maior de computadores para simulação em maior escala. Estamos planejando também que os testes sejam complementados com injecção de falhas de comunicação para permitir, além da eficiência do serviço, extraer medidas como desempenho sob falhas e disponibilidade de serviço.

## AGRADECIMENTOS

Esse trabalho está sendo desenvolvido no escopo do projeto JitCloud com o apoio da RNP.

## REFERÊNCIAS

- [1] J. Rosenberg, M. Day, e others, “A model for Presence and Instant Messaging”, 2000.
- [2] M. Day, S. Aggarwal, G. Mohr, e J. Vincent, “Instant messaging/presence protocol requirements”, Request for Comments, vol. 2779, 2000.
- [3] J. Pereira, L.Rodrigues, M.J. Monteiro, R. Oliveira, A. -M Kermarrec Microsoft Resourch “NEEM: Network-friendly Epidemic Multicast”, 2003.
- [4] J. Leitao, J. Pereira, e L. Rodrigues, “HyParView: A membership protocol for reliable gossip-based broadcast”, in Dependable Systems and Networks, 2007. DSN’07. 37th Annual IEEE/IFIP International Conference on, 2007, p. 419–429.
- [5] P. T. Eugster, R. Guerraoui, A. M. Kermarrec, e L. Massoulié, “Epidemic information dissemination in distributed systems”, *Computer*, vol. 37, nº. 5, p. 60–67, 2004.
- [6] P. Saint-Andre. “Extensive Messaging e Presence Protocol (XMPP)”, 2011.
- [7] A. Niemi, “Session initiation protocol (SIP) extension for event state publication”, 2004.
- [8] Ericsson, Motorola and Nokia. “The Wireless Village initiative: System Architecture Model”, 2001-2002.
- [9] R. Drebes, G. Jacques-Silva, J. da Trindade, e T. Weber, “A kernel-based communication fault injector for dependability testing of distributed systems”, *Hardware and Software, Verification and Testing*, p. 177–190, 2006.
- [10] T. Siqueira, B. Fiss, R. Weber, S. Cechin, e T. Weber. 2009. “Applying FIRMAMENT to test the SCTP communication protocol under network faults”. In Test Workshop, 2009. LATW ’09. 10th Latin American, 1 -6.

# Virtualizando com Xen Cloud Platform (XCP)

Maicon L. R. da Cruz Alves  
Faculdade Senac Pelotas  
admredesmaiconalves@gmail.com

Andre Moraes\*  
Faculdade Senac Pelotas  
chameoandre@gmail.com

**Resumo**—Esse artigo apresenta as vantagens da virtualização, utilizando as ferramentas do Xen Cloud Platform (XCP), facilidade na administração, gerenciamento centralizado, otimização de recursos. Foi simulado um cenário de testes na UFPEL (Universidade Federal de Pelotas) para analisar os recursos do XCP, foram analisados os seguintes recursos, XenMotion, High Availability (Alta Disponibilidade), Bond (Ligaçāo), Switch Virtual, Storage com Iscsi, e realizados testes de performance, falha de energia elétrica, falha de switch, manutenção preventiva, formas de backups.

## I. INTRODUÇÃO

Nos últimos anos as empresas notaram que estão perdendo muito dinheiro com desperdício de hardware, cada software novo necessita de uma configuração diferente dos demais, os empresários tem a difícil escolha de atualizar o software existente que nem sempre é possível ou comprar mais um hardware para instalar o software, este processo gera uma despesa desnecessária, o aumento de máquinas eleva o consumo de energia, redução do espaço físico e necessidade de mais pessoas para assistência técnica. Com a virtualização pode-se consolidar as cargas de trabalho do servidor otimizando o hardware, permitindo a criação de várias instâncias de máquinas virtuais cada uma com a configuração necessária para o software específico, economizando espaço, energia, refrigeração, tempo e dinheiro, contribuindo assim para computação ecologicamente sustentável, maior capacidade de se adaptar as constantes mudanças dos ambientes de TI, e uma melhoria do nível de confiabilidade de TI[1][2].

## II. VIRTUALIZAÇÃO

Com a expansão dos serviços e aplicações empresas procuram soluções que possibilitem acompanhar o avanço da tecnologia com o menor recurso possível.[3].

Atualmente há 5 tipos de virtualização: emulação, completa, paravirtualização, hvm, kvm. Na Seções abaixo, serão descritos cada tipo de virtualização.

### A. O que é a virtualização?

Virtualização é um processo de executar várias instâncias (máquinas virtuais) em um único equipamento físico, todas instâncias são controladas por um VMM (Virtual Machine Monitor) conhecido como hypervisor. Com a virtualização as máquinas virtuais se comportam como se fossem um computador independente, utilizando somente os recursos que foram atribuídos para ela. [3]

\*Orientador do Projeto

### B. Quem utiliza virtualização

Atualmente a cada dez servidores, 4 são virtualizados e cerca de 90 % das empresas utilizam algum tipo de virtualização [4]. Cerca de 4,5 bilhões de dispositivos utilizam JVM (Java Virtual Machine) que é um tipo de virtualização por emulação[5]. Grandes empresas (DELL, IBM, HP) além de utilizar a virtualização também vendem soluções de virtualização para seus clientes.

### C. Tipos de virtualização

Existem 5 tipos de virtualização.

1) **EMULAÇÃO**: Cria o próprio hardware sem depender do hardware onde está sendo processado. Ex: Java Virtual Machine.

2) **COMPLETA**: Roda sobre um Sistema Operacional host (máquina física) e os guest (máquina virtual) tem seu hardware criados e gerenciados pelo VMM (Virtual Machine Manager). Ex:Vmware, Virtual Box.

3) **PARAVIRTUALIZAÇÃO**: Roda diretamente no hardware do host, os guest acessam o hardware através do hypervisor, o S.O do guest administrativo tem seu kernel modificado. OBS: as vms não tem o acesso diretamente ao hardware, I/O de disco e memória são acessados diretamente o hardware real, I/O de cpu e network passam primeiro pelo guest administrativo para depois ir para o hardware real.

4) **HVM**: Roda diretamente no hardware real, os guest acessam o hardware através do hypervisor. S.O guest não tem modificação no kernel, necessita de um hardware específico com suporte a virtualização ex: processador intel(VT) e amd(V).

5) **KVM**: Roda sobre o kernel do seu host, se utiliza de instruções de cpu.

### D. Benefícios da Virtualização

Os principais benefícios da virtualização.

- **Consolidação**: É um agrupamento de vários servidores e aplicações em apenas 2 servidores ou mais e 1 storage [6].

- **Aumento da disponibilidade**: Atualmente sem virtualização quando um serviço fica indisponível pode-se restabelecer em 30 minutos ou mais com ajuda do suporte, com a virtualização conseguimos restabelecer o serviço em poucos minutos sem a necessidade de suporte externo.

- **Redundância a backup**: Com a virtualização há várias formas de backups, snapshot que é um ponto de restauração da máquina virtual, Fast Clone, Full Clone e Export.

- *Espaço físico:* Puma empresa com 7 servidores, cada um com seu serviço e aplicações separados por incompatibilidade de pacotes, com a virtualização pode-se ter apenas 2 servidores com 7 máquinas virtuais, otimizando o espaço físico.
- *Consumo de energia:* No mesmo exemplo dos 7 servidores cada um com fonte redundante ou seja 14 fontes, reduzindo para 2 servidores virtualizados são 4 fontes mais uma fonte da storage total de 5 fontes há uma redução considerável de consumo de energia.
- *Otimização do hardware:* Atualmente empresários compram um servidor com uma excelente configuração para utilizar apenas um serviço que irá usar 10% do hardware do servidor. Com a virtualização pode-se dividir o hardware para várias vms aproveitando da melhor forma o hardware disponível.
- *Balanceamento de carga:* Pode-se migrar uma máquina virtual de um servidor para o outro sem necessidade de parar o serviço, se estiver com Bond configurado no modo ativo-ativo ou quando uma interface estiver sobrecarregada automaticamente será balanceado a carga para a outra interface.
- *Divisão de serviços:* Atualmente o maior problema que o administrador de redes é a incompatibilidade de software (pacotes). Com a virtualização pode-se criar uma máquina virtual separada para determinado serviço, a divisão de serviço é uma forma de ter mais segurança, exemplo um servidor com os serviços dns, apache, banco de dados, pode ser alvo de ataque, em casos de um invasor causar algum dano, provavelmente irá afetar todos os serviços.
- *Gerenciamento centralizado:* Diminuindo o número de servidores físicos, já se torna um gerenciamento centralizado, o XCP pode ser gerenciamento pelo xencenter(software proprietário Citrix) onde pode-se adicionar todos servidores virtualizados e administrar através de um único computador.
- *Diminuição de custo:* Consequentemente com a redução de servidores físicos irá reduzir o consumo de energia, a necessidade de menos pessoas para gerenciar, uma sala menor, uma refrigeração menor, no conjunto será uma redução de custo significativa.
- *Disponibilidade:* Planejando o layout de virtualização conforme a necessidade o tempo de atividade do serviço será de 99,9% ativo [6].
- *Continuidade:* Criando um redundância a falhas o tempo para o serviço voltar a funcionar é de aproximadamente de 5 minutos [6].

### III. TECNOLOGIAS UTILIZADAS

Na elaboração deste projeto foram utilizadas as tecnologias descritas nas seções a seguir, para avaliar as vantagens da virtualização.

#### A. Xen Cloud Platform (XCP)

O Xen Cloud Platform (XCP) é software livre, de virtualização, planejado para plataforma de computação

em nuvem, proporcionando o Xen Hypervisor com suporte para uma variedade de sistemas operacionais convidados incluindo Windows e Linux, rede e suporte de armazenamento. Um dos pontos fortes do XCP é as ferramentas de gerenciamento por web e por software..[7]

#### B. Bond (ligação)

É redundância entre as Nics(Network Interface Controller - Interface física), este recurso necessita-se adicionar duas ou mais NIC para responder como se fosse uma única interface, pode-se utilizar duas opções de Bond, ativo-ativo e ativo-passivo [?].

**Ativo-ativo:** Suporta平衡amento de carga do tráfego, fornece suporte fail-over. A Bond faz balanceamento de tráfego entre múltiplas interfaces virtuais enviando o tráfego através de diferentes NICs físicas, baseado no endereço MAC de origem do pacote.

**Ativo-passivo:** Neste modo apenas uma das NIC irá responder as requisições, quando a NIC máster parar de funcionar, a NIC passiva assume o lugar da principal, não faz balanceamento de carga.

#### C. Switch virtual - layer 2

Em determinados ambientes talvez seja necessário separar as máquinas virtuais da rede real criando assim uma rede privada, por padrão a estrutura de rede do XCP é baseada na *PIF (Physical Interface) - Network0 (Switch virtual padrão) - VIF (Virtual Interface) - VM (Virtual Machine)*, quando é criado mais switchs virtuais eles só irão se comunicar internamente entre vms, a vazão de dados de um switch virtual é 10x mais do que a real, porque todo tráfego passa pelo barramento interno da máquina real.[?]

#### D. XenMotion Live Migration

XenMotion é um recurso que permite a migração de máquinas virtuais. Com XenMotion, pode se mover uma máquina virtual em execução de um host físico para outro host físico, sem qualquer interrupção ou inatividade [?]. Para utilizar o XenMotion necessita de pelo menos dois sistemas hosts físicos em um pool. Os sistemas host precisam ter configurações de processador idênticas, armazenamento compartilhado remoto e conectividade de preferência Gigabit Ethernet entre elas.

O armazenamento da máquina virtual precisa estar compartilhado entre os hosts para permitir a realocação ao vivo com XenMotion. Isso pode incluir compartilhamentos de arquivo baseado em NFS, SAN e iSCSI ou um Fibre Channel SAN.

Na migração ao vivo geralmente perde a conexão de 100 a 150 milissegundos. Esta perturbação é tão pequena que os serviços em execução na máquina virtual não são interrompidos. A maior parte do rompimento é causada pelo comutador de rede deslocando o tráfego para uma nova porta.

### E. HA (Alta disponibilidade)

O HA (High Availability) é uma garantia de continuidade. Para configurar o HA é necessário dois servidores com processadores idênticos, e uma storage compartilhada para armazenar as vms. Quando ocorrer algum problema em um dos servidores físicos, exemplo perda de comunicação com a rede seja, o outro servidor irá verificar que um dos servidores do HA não está respondendo e automaticamente irá iniciar todas as vms que estavam ligadas no servidor que não parou de responder.[8]

### F. XenCenter

É um software desenvolvido para rodar em sistema operacional Windows. Através do XenCenter pode-se gerenciar todos servidores físicos que estão com o XCP e XenServer instalado, ou seja, um gerenciamento centralizado. O xencenter se comunica com o XCP pelas portas 80 e 443, conforme ilustrado na Figura 1, o Xencenter tem uma interface intuitiva de fácil compreensão até usuários comuns conseguem administrar servidores com facilidade.

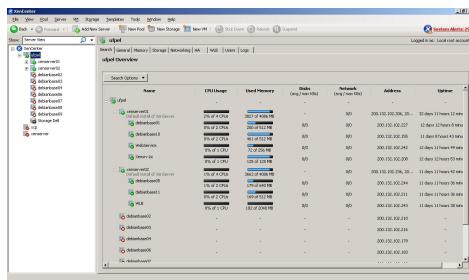


Figura 1. Gerenciamento Centralizado

### G. Iscsi

Um fator principal da virtualização é o Failover Clustering com storages, o iSCSI (Internet Small Computer System Interface) é um protocolo de transporte que encapsula comandos SCSI dentro de um pacote IP entre o anfitrião e o target (dispositivo de destino), permitindo que uma máquina acesse uma storage iSCSI via ethernet. Iscsi é mais um recurso para diminuir custos [9].

## IV. CENÁRIO DE TESTES

Foi simulado um cenário de testes na UFPEL (Universidade Federal de Pelotas) [conforme ilustrado na figura 2]. Para montar o cenário foram utilizados três servidores Dell T310, dois com o XCP instalado e um com Xen (no debian squeeze este terceiro também serve de storage com iSCSI), os servidores conectados a um switch extreme, foi configurado nos dois XCP a storage iscsi remota, para poder utilizar as ferramentas XenMotion e HA, com as principais ferramentas configuradas (XenMotion, HA, Bond) foi criado dez máquinas virtuais para testes.

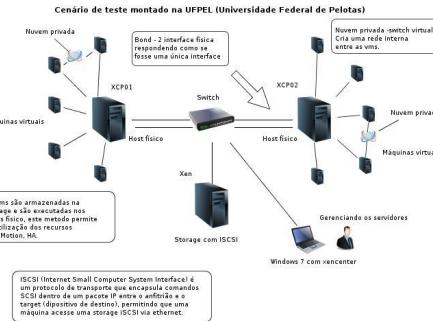


Figura 2. Cenário de Testes - UFPEL

### A. Testes de Performance

Foram realizados testes com: Iperf, Synk4.

- *Iperf*

Foram feitos testes de tráfego de rede para analisar a vazão para rede real e virtual, conforme tabela 1. Foi constatado que o tráfego da rede normal se manteve entre 97 Mb e 98 Mb. No XCP é possível criar switches virtuais e o tráfego de rede entre vms pelo switch virtual foi de 12.4 Gb. Constatou-se que o tráfego da rede virtual criada pelo XCP não sai da máquina real e mantém-se entre o barramento interno das vms, proporcionando muito mais desempenho do que a rede real.

Tabela I  
TESTE DE TRAFEGO DE REDE COM IPERF

Máquinas	Switch Real	Switch Virtual
VM XCP	98.1 mb	12.4 gb
Maq. Real	97.9 mb	X

- *Synk4*

Synk4 envia múltiplos pacotes syn para o alvo, realizou-se dois testes conforme tabela 2, um ataque flood syn de um servidor físico XCP01 para uma vm (debianbase05) no segundo servidor XCP02, sem Bond configurado e com Bond configurado como ativo-ativo.

Tabela II  
TESTE DE TRAFEGO DE REDE COM SYNK4

Método	Resultado
Sem Bond	25 Milhões
Com Bond	110 Milhões

### B. Falha de energia elétrica

Com o uso do recurso High Availability. Foram realizados testes de falhas de energia elétrica e constatou-se que os servidores com o HA configurado, após a falta de energia no servidor X, o servidor Y verificou que o servidor X parou de responder então ligou todas vms do servidor X este processo demorou 5 minutos para começar a iniciar as vms.

### C. Falha de switch ou problema de cabo de rede

Com o uso do recurso Bond. Após configurar as interface como Bond ativo-ativo, com as 2 interfaces com cabo de rede, foi executado uma restauração de banco de dados e teste de ping para uma vm. Após remover um cabo de rede da interface primária, houve perda de um ping o banco de dados parou de restaurar por 3 segundos, foi recolocado o cabo de rede e removido o cabo de rede da interface secundária, não houve perda de ping e restauração foi concluída.

### D. Manutenção preventiva do servidor

Com o uso do recurso XenMotion. Foi realizada uma simulação de um ambiente em produção com a utilização do software "ab" da apache e ping, a intenção da simulação seria o acesso de um sistema feito em web e se haveria grande perda no processo feito para manutenção, com a simulação em andamento foi feito a migração das vms que estavam no servidor X para o servidor Y sem desligar as vms. O processo demorou em torno de 6 minutos, no teste de ping obteve 3 pacotes perdidos, e no teste com o "ab" para 3000 conexões obteve perda e retransmissões da perda de 180 conexões.

### E. Backup e Restauração

Ha 4 formas de backup, Fast Clone, Full Clone, Export, Snapshot, Script de backup.

- *Fast Clone*

Necessita que a vm esteja desligada e um clone idêntico da máquina virtual, o backup sera feito para a mesma storage, o processo do backup fast clone demora em torno de 15 segundos.

- *Full Clone*

Também necessita da vm desligada e um clone idêntico da máquina virtual, pode-se escolher para qual storage será feito o backup.

- *Export*

Da mesma forma que o fast clone e full clone necessita que a vm esteja desligada neste modo de backup pode-se exportar a vm para qualquer lugar nas extensões ovf e xva, após exportar a vm pode ser importada para o xcp ou pode ser importada para o VmWare.

- *Import*

Um dos modos de restauração e a importação de arquivos com a extensão ovf, este processo é um pouco demorado.

- *Script para backup*

Utilizando o snapshot e o export, a vantagem do script é de poder fazer o backup com a vm ligada, o script faz o processo de criar o snapshot, exportar o snapshot com data, exporta o snapshot e excluir o snapshop.

- *Script de restauração*

Este script utiliza o xe vm-import é o mesmo processo de import mas em modo texto, bastante utilizado por administradores de redes que gostam do modo CLI (command line interface).

## V. CONCLUSÃO

A virtualização é uma poderosa técnica que permite ao administrador gerenciar várias instâncias sobre o mesmo hardware físico, com o avanço tecnológico dos hardwares gerou muitos recursos ociosos, com muita disponibilidade de recurso de hardware a virtualização se tornou necessária para aproveitar ao máximo o hardware disponível.

Ao virtualizar um ambiente implicará na mudança da política da empresa tanto na parte de manutenção operacional quanto a compras de novos produtos e sistemas.

Após a realização de vários testes, constatou-se que a queda de desempenho entre máquina virtual e a máquina real é muito pequena considerando que se tem uma gama de recursos disponíveis para o gerenciamento dos serviços.

Se pensar em questão de serviços críticos onde a criação, manutenção e restauração de serviços sejam o mais rápido possível, a virtualização é extremamente necessária eficaz. A virtualização está em constante crescimento, a cada momento surgem ferramentas novas melhorando seu desempenho, se grandes empresas utilizam essa técnica, significa que é extremamente segura e confiável.

## REFERÊNCIAS

- [1] J. N. M. et al., *Executando o Xen - Um Guia Prático para a Arte da Virtualização*, 1st ed. Altabooks, 2009.
- [2] Amaluti, “Conceitos da virtualização.” 2010, disponível em: <<http://amaluli.com/2010/05/12/conceitos-basicos-de-virtualizacao/>>. Acesso em: Mai 2012.
- [3] Wikipédia, “Virtualização,” 2012, disponível em: <<http://pt.wikipedia.org/wiki/Virtualiza%C3%A7%C3%A3o>>. Acesso em: Mar 2012.
- [4] C. Unido, “A cada dez servidores quatro são virtuais, diz estudo,” 2011, disponível em: <<http://computerworld.uol.com.br/tecnologia/2011/07/19/a-cada-dez-servidores-quatro-sao-virtuais-diz-estudo/>>. Acesso em: Abr 2012.
- [5] J. Sun, “Saiba mais sobre a tecnologia java,” 2012, disponível em: <[http://www.java.com/pt\\_BR/about/](http://www.java.com/pt_BR/about/)>. Acesso em: Jun 2012.
- [6] D. Zarpelon, “Ssixen - implementando xenserver 5.5,” 2012, disponível em: <<http://www.sisnema.com.br>>. Acesso em: Mar 2012.
- [7] Xen.org, “Xen cloud platform project,” 2010, disponível em: <<http://www.xen.org/products/cloudxen.html>>. Acesso em: Abr 2012.
- [8] Wikipedia, “High-availability cluster,” 2012, disponível em: <[http://en.wikipedia.org/wiki/High-availability\\_cluster](http://en.wikipedia.org/wiki/High-availability_cluster)>. Acesso em: Mai 2012.
- [9] Wikipédia, “Iscsi,” 2012, disponível em: <<http://pt.wikipedia.org/wiki/ISCSI>>. Acesso em: Jun 2012.

---

V

# Fórum de Pós-Graduação I

---



# Alta disponibilidade em redes IPv6 críticas utilizando o protocolo CARP

Carlos Kenji Kitahara, Lennon Soeiro  
IPT – Instituto de Pesquisas Tecnológicas do Estado de São Paulo  
cakenji@gmail.com e lennonjs@gmail.com

**Resumo**—A crescente dependência dos negócios com a TI aumenta a necessidade das organizações de adotarem soluções de alta disponibilidade em sua infraestrutura de rede. O protocolo Common Address Redundancy Protocol (CARP) é uma solução livre desenvolvida pelo grupo OpenBSD que fornece redundância em nível de gateway. O objetivo deste artigo é avaliar a utilização do protocolo CARP em redes IPv6 e realizar uma análise comparativa com outros protocolos específicos de redundância de gateway como o HSRP, VRRP e GLBP, além do protocolo NDP utilizado em redes IPv6. Na avaliação foi realizado um experimento com o objetivo de medir o tempo de recuperação do protocolo CARP após a interrupção do gateway padrão em uma rede com roteamento OSPFv3 configurado.

## I. INTRODUÇÃO

Para muitas empresas, a informação e a tecnologia que a suporta representa o seu bem mais valioso [6], tornando a área da Tecnologia da Informação (TI) fundamental para a execução dos seus acordos e transações comerciais (negócio) [1]. Consequentemente é essencial que sejam implementadas soluções garantam a disponibilidade das redes de computadores, que são os elementos principais da infraestrutura de TI.

A disponibilidade é a garantia de que um sistema computacional possa ser acessado por seus usuários quando estes necessitarem acessá-lo. O mecanismo de disponibilidade envolve a redundância de hardware, inteligência de software e protocolos para identificar a existência de falha do sistema principal para iniciar e concluir um processo de transferência dos serviços para sistemas alternativos [4].

Uma das técnicas adotadas para se evitar as indisponibilidades nas redes de computadores, prover tolerância a falhas e garantir a continuidade dos serviços críticos de TI é a utilização de gateways redundantes. A figura 1 mostra uma estrutura básica de redundância de rede com dois roteadores.

Este artigo foi desenvolvido com intuito de avaliar o protocolo de redundância de gateways CARP (Common Address Redundancy Protocol) em redes IPv6 e realizar uma análise comparativa com os protocolos HSRP (Hot Standby Router Protocol), VRRP (Virtual Router Redundancy Protocol) e GLBP (Gateway Load Balance Protocol) além do protocolo NDP (Neighbor Discovery Protocol) utilizado em redes IPv6. No processo de avaliação foi realizado um experimento visando medir o tempo de recuperação gasto pelo protocolo após a interrupção do gateway principal em uma rede IPv6.

Dr. Alexandre José Barbieri de Souza  
IPT – Instituto de Pesquisas Tecnológicas do Estado de São Paulo  
abarbieris@hotmail.com

O artigo está organizado da seguinte forma, a seção 2 descreve os protocolos HSRP, VRRP, GLBP, o protocolo NDP para redes IPv6 e o protocolo CARP. A seção 3 apresenta a análise comparativa dos protocolos. A seção 4 apresenta o experimento com o protocolo CARP e seu resultado e a seção 5 a conclusão do artigo. Finalmente na seção 6 apresentam-se sugestões de trabalhos futuros.

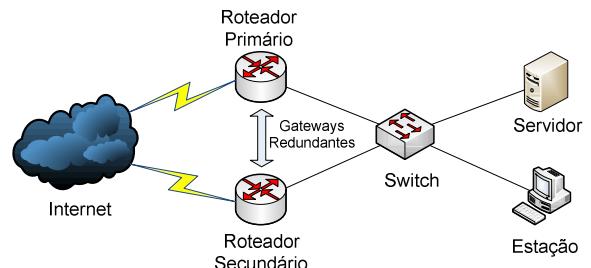


Figura 1. Estrutura Básica de Redundância.

## II. PROTOCOLOS DE REDUNDÂNCIA RELACIONADOS

Além do protocolo CARP existem outras soluções que podem ser utilizadas para prover redundância de gateways. Dentre as soluções destacam-se os protocolos HSRP e GLBP da empresa Cisco Systems e o VRRP criado pela IETF e disponível nos produtos de muitas empresas tais como Juniper Networks, 3Com Corporation e a própria Cisco Systems. O VRRP é implementado também em plataformas livres como Linux e BSD [5].

Outro protocolo que pode fornecer redundância é o *Neighbor Discovery Protocol* (NDP) em redes IPv6. Utilizando este protocolo, os equipamentos recebem informações sobre os roteadores da rede através da mensagem enviada periodicamente chamada *Router Advertisement* e podem detectar uma falha através do mecanismo *Neighbor Unreachability Detection* [13].

A seguir apresentam-se resumidamente os protocolos HSRP, VRRP, GLBP e o protocolo NDP para redes IPv6.

### A. HSRP (Hot Standby Router Protocol)

O HSRP é um protocolo de redundância desenvolvido pela empresa Cisco Systems que fornece tolerância a falhas no contexto de gateway. O protocolo permite que dois ou mais roteadores pertencentes a um grupo chamado *HSRP group* ou *standby group* compartilhem um endereço IP e um endereço MAC denominados endereços virtuais.

Apenas um dos roteadores denominado *active* é responsável pelo encaminhamento dos pacotes. Os demais

roteadores dentro do grupo são chamados *standby* e permanecem em um estado de espera. A definição dos papéis *active* e *standby* é realizada através de um processo de eleição em que o roteador de maior prioridade é designado como ativo. Periodicamente os roteadores trocam mensagens chamadas *Hello* com o objetivo de identificar possíveis falhas.

No caso de uma falha no roteador designado como ativo, o roteador *standby* assume o endereço IP e o endereço MAC virtuais. Se o roteador designado como *standby* falha ou ele torna-se o roteador ativo, uma nova eleição é realizada para designação de um novo roteador *standby*. A identificação de uma falha é realizada através do parâmetro *Hold Time* que é o intervalo de tempo na qual os roteadores aguardam as mensagens do tipo *Hello*. Caso um roteador não receba mensagens *Hello* no período informado pelo parâmetro *Hold Time* considera-se que o outro roteador está indisponível [12].

#### B. VRRP (Virtual Router Redundancy Protocol)

Assim como o HSRP o VRRP é um protocolo que fornece redundância de gateway e permite o compartilhamento de um endereço IP e endereço MAC por vários roteadores. Um dos roteadores denominado de *Master* é o responsável pelo encaminhamento dos pacotes. Os outros roteadores pertencentes ao grupo de redundância são denominados *Backup*.

A definição do papel de *Master* também é definido por um processo de eleição em que o roteador com maior prioridade é eleito o roteador principal.

O intervalo de tempo da troca de mensagens entre os roteadores é configurado pelo parâmetro *Advertisement\_Interval* que por padrão é igual a 1s. Já a identificação da falha é realizada através do parâmetro *Master\_Down\_Interval* calculado pela fórmula [11]:

$$(3 * \text{Master_Adver_Interval}) + \frac{(256 - \text{priority}) * \text{Master_Adver_Interval}}{256}$$

O parâmetro *Master\_Adver\_Interval* possui valor inicial igual ao *Advertisement\_Interval*. Decorrido o tempo configurado no parâmetro *Master\_Down\_Interval*, após a falha do roteador *Master*, inicia-se automaticamente um novo processo de eleição do roteador responsável pelo encaminhamento de pacotes.

#### C. GLBP (Gateway Load Balance Protocol)

Protocolo de redundância desenvolvido pela empresa Cisco Systems que além de fornecer tolerância a falhas oferece também balanceamento de carga no contexto de gateway. O balanceamento de carga é realizado através do compartilhamento pelos roteadores de um único IP virtual e múltiplos endereços MAC.

Tecnicamente, o funcionamento do protocolo GLBP é muito similar ao HSRP, porém o gateway que antes estava em modo *standby* passa a ser utilizado em paralelo como um gateway ativo.

Membros de um grupo do GLBP elegem o roteador de maior prioridade como o *Active Virtual Gateway* (AVG). Este roteador determina um endereço MAC para cada um

dos roteadores do grupo que são denominados *Active Virtual Forwarder* (AVF). Para cada requisição ARP recebida para o endereço virtual, o AVG responde com um dos endereços MAC virtual, transferindo a responsabilidade do encaminhamento dos pacotes ao dono daquele MAC, possibilitando assim o balanceamento [3].

Os roteadores de um grupo GLBP comunicam-se através de mensagens *Hello* enviadas a cada 3 segundos e assim como o HSRP após 10 segundos sem o recebimento da mensagem *Hello* o equipamento é considerado como indisponível e um processo de eleição é iniciado.

#### D. NDP (Neighbor Discovery Protocol)

Uma das funcionalidades nas redes IPv6 é o protocolo NDP que habilita a detecção de roteadores (*Router Discovery*). Apesar de não ser um protocolo específico para redundância de gateways ele pode fornecer alta disponibilidade em uma rede IPv6.

Os processos do NDP utilizam 5 tipos diferentes de pacotes ICMPv6 [13], um par de mensagens *Router Solicitation* e *Router Advertisement*, um par de mensagens *Neighbor Solicitation* e *Neighbor Advertisement* e uma mensagem *Redirect*.

No processo de identificação de um roteador em seu enlace, um nó envia uma mensagem *Router Solicitation* utilizando *multicast*. Roteadores no mesmo enlace respondem com a mensagem *Router Advertisement* ao nó solicitante que configura o endereço do roteador. Periodicamente roteadores enviam mensagens *Router Advertisement* utilizando *multicast* em seu enlace.

A falha de um roteador pode ser detectada por um host através do mecanismo *Neighbor Unreachability Detection*. Para confirmação de que um roteador está ativo nós enviam mensagens *unicast* *Neighbor Solicitation* e aguardam mensagens *Neighbor Advertisement*. Visando evitar tráfego excessivo essas mensagens são enviadas apenas para nós que estão trafegando dados ativamente e após a indicação de que um roteador não está comunicando. Utilizando os parâmetros padrão do protocolo NDP, leva-se aproximadamente 38 segundos para que o nó perceba que um roteador está indisponível e altere o roteador padrão para outro roteador [13].

#### E. CARP (Common Address Redundancy Protocol)

O protocolo CARP, Common Address Redundancy Protocol, desenvolvido pelo projeto OpenBSD também tem por objetivo garantir a redundância através do compartilhamento de um endereço IP virtual por múltiplos computadores. Sua criação pela comunidade Open Source serve como alternativa livre e segura ao protocolo VRRP que possui sua especificação reivindicada pela empresa Cisco Systems [2].

O computador principal denominado *Master* responde a qualquer tráfego ou requisições ARP direcionadas para o IP compartilhado. Os outros membros do grupo são denominados *Backup* assim como no VRRP. Cada computador pode pertencer a mais de um grupo de redundância por vez [9].

O computador *Master* envia anúncios CARP com maior freqüência e é influenciado pelos parâmetros *advbase* e *advskev* pela fórmula  $advbase + (advskev/255)$ . O primeiro parâmetro é a base do intervalo de anúncios já o segundo influencia o intervalo de anúncios CARP. Quanto menor o valor maior a probabilidade de o computador ser considerado *Master* [8]. O valor padrão para *advbase* é 1 segundo e para *advskev* é 0.

Cada membro do grupo verifica se a periodicidade de seu anúncio é menor que os anúncios do nó *Master*. Se por alguma razão o nó *Master* falhar após o valor correspondente a  $3*(advbase + (advskev/255))$  segundos [7] todos os computadores *Backup* enviam seus anúncios baseados em seus próprios parâmetros. Aquele de maior freqüência é eleito o novo mestre.

No quesito segurança o protocolo CARP utiliza o algoritmo de assinatura HMAC SHA-1 para a checagem de integridade e autenticidade dos anúncios [8].

### III. COMPARAÇÃO ENTRE OS PROTOCOLOS

Após a apresentação dos protocolos é possível a realização de uma análise comparativa levando em consideração características como tempo total gasto para a recuperação, ou seja, quando o outro roteador (*backup*) assume o IP do gateway no caso de uma falha no roteador principal, balanceamento de carga e suporte a IPv6. As tabelas I e II apresentam as principais características dos protocolos HSRP, VRRP, GLBP e CARP. Apesar de o NDP fornecer alta disponibilidade ele não é um protocolo específico de redundância de gateways desse modo sua comparação ficou restrita apenas ao tempo de recuperação.

Podem-se visualizar características comuns entre o protocolo CARP e os outros protocolos específicos de redundância de gateways como a possibilidade de configuração de *preempt*, ou seja, tornar o roteador com maior prioridade sempre o mestre, suporte a IPv6 e a utilização do protocolo de transporte UDP para troca de mensagens entre os roteadores. No caso do balanceamento de carga apenas os protocolos CARP e GLBP podem realizar o balanceamento utilizando um único IP virtual. Para os protocolos HSRP e VRRP o balanceamento de carga é realizado utilizando-se vários IPs virtuais.

Conforme tabelas I e II tanto o protocolo CARP quanto os outros protocolos possibilitam a configuração do intervalo de tempo dos anúncios do roteador mestre e do tempo para que os roteadores backup elejam um novo roteador principal após a falha do roteador mestre. O protocolo CARP assim como o VRRP nos padrões de configuração possui tempo de recuperação menor que os dos protocolos proprietários.

Em relação ao protocolo NDP utilizado em redes IPv6 constata-se que dentro ainda das configurações padrões, todos os protocolos de redundância gateways apresentados possuem tempo de recuperação menor que o NDP que é de aproximadamente 38 segundos. Vale destacar que no protocolo NDP é possível reduzir os parâmetros de tempo porém essa configuração causa um aumento significativo no tráfego principalmente em enlaces com muitos nós.

Tabela I

CARACTERÍSTICAS DOS PROTOCOLOS HSRP E VRRP		
Protocolos	HSRP	VRRP
Criado por	Cisco	IETF
Intervalo de anúncios do roteador mestre	Padrão 3 seg	Padrão 1 seg
Tempo de recuperação	Padrão 10 seg	Padrão 3 seg
Ajuste de tempo	Sim	Sim
Preemption	Sim	Sim
Protocolo de Transporte	UDP/1985	UDP/112
Balanceamento de Carga	Cada estação cliente recebe um endereço de gateway diferente	Cada estação cliente recebe um endereço de gateway diferente
Suporte a IPv6	Sim	Sim
Endereço Virtual	00:00:0C:07:AC:{group}	00:00:5E:00:01:{VRID}

Tabela II  
CARACTERÍSTICAS DOS PROTOCOLOS GLBP E CARP

Protocolos	GLBP	CARP
Criado por	Cisco	OpenBSD
Intervalo de anúncios do roteador mestre	Padrão 3 seg	Padrão 1 seg
Tempo de recuperação	Padrão 10 seg	Padrão 3 seg
Ajuste de tempo	Sim	Sim
Preemption	Sim	Sim
Protocolo de Transporte	UDP/3222	UDP/112
Balanceamento de Carga	Sim	Sim
Suporte a IPv6	Sim	Sim
Endereço Virtual	00:07:b4{group, AVF}	00:00:5E:00:01:{VHID}

### IV. EXPERIMENTO

O experimento com o protocolo CARP visa medir o tempo de recuperação após a queda do roteador principal. No experimento também foi configurado roteamento com o protocolo OSPFv3 [10] com o propósito de identificar o impacto do tempo de recuperação do protocolo CARP em uma rede com roteamento configurado. A topologia do experimento encontra-se na figura 2.

Através da estação foram gerados pacotes ICMPv6 para o IP configurado nos gateways redundantes e para o servidor. Durante o envio dos pacotes o roteador mestre foi interrompido e realizada a medição do tempo total para que a comunicação fosse restabelecida. Para o experimento foram definidas 10 amostras.

Na montagem do ambiente da figura 2 foram utilizados os sistemas operacionais Windows XP e Debian 6.0.3 respectivamente para a estação e o servidor. Nos roteadores foi utilizado o sistema operacional FreeBSD 8.2. O software utilizado para roteamento foi o Quagga e o tempo de indisponibilidade foi calculado através da captura de pacotes ICMPv6 utilizando-se o software Wireshark. As figuras 3 e 4 contêm o resultado das informações coletadas.

O protocolo CARP teve como tempo médio de recuperação 2,93 segundos e desvio padrão 0,06. Já o protocolo OSPFv3, conforme figura 3, teve como tempo médio de convergência 37,44 segundos e desvio padrão 0,79.

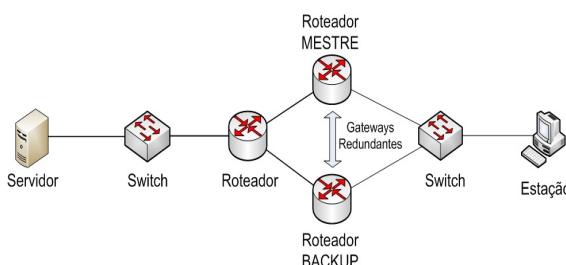


Figura 2: Topologia para avaliação dos protocolos

O resultado mostra que nas configurações padrões, o tempo de recuperação do protocolo CARP é significativamente menor que o tempo de convergência do protocolo OSPFv3 indicando desse modo que em uma rede com roteamento OSPFv3 configurado e que utiliza o protocolo CARP para redundância de gateway a disponibilidade será impactada principalmente pelo tempo de convergência do protocolo de roteamento.

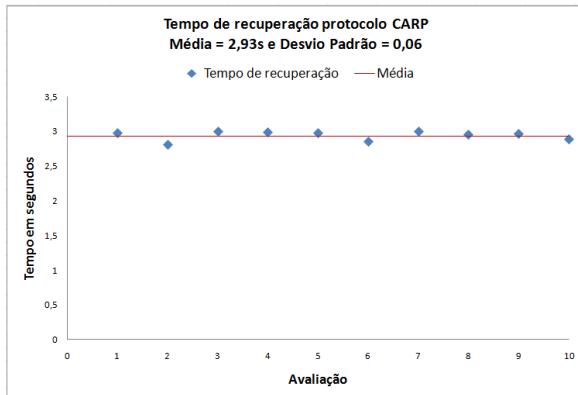


Figura 3: Tempo de recuperação CARP

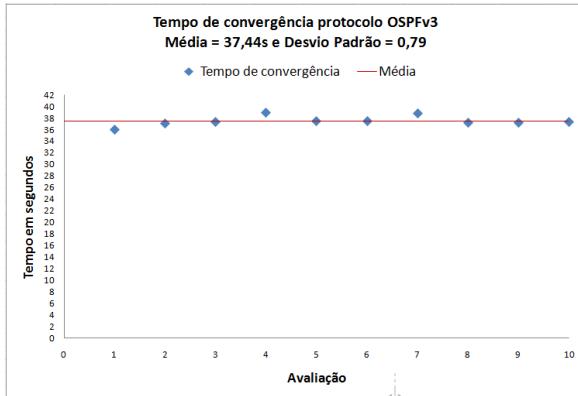


Figura 4: Tempo de convergência OSPFv3

## V. CONCLUSÃO

A disponibilidade da infraestrutura de TI, em especial da rede de computadores, é fundamental para muitas organizações já que o negócio das empresas está cada vez mais dependente da tecnologia. Para garantir a disponibilidade da rede em ambientes críticos é essencial que soluções como a utilização de gateways redundantes sejam adotadas pelas organizações.

Visando avaliar o protocolo CARP foi realizada uma análise comparativa com outros protocolos específicos de redundância de gateways dentre eles o HSRP, VRRP e o GLBP além do protocolo NDP utilizado em redes IPv6. Verificou-se que o protocolo CARP suporta várias características presentes em outros protocolos como a configuração de *preempt*, suporte a IPv6 e balanceamento de carga. Além disso, constatou-se que na configuração padrão, o tempo de recuperação do protocolo CARP é menor que o dos protocolos HSRP, GLBP e NDP.

No experimento realizado com o protocolo CARP em uma rede com roteamento OSPFv3 foi possível identificar que a disponibilidade, na configuração padrão dos protocolos, será afetada pelo tempo de convergência do protocolo OSPFv3.

## VI. TRABALHOS FUTUROS

O escopo deste trabalho limitou-se ao estudo da alta disponibilidade fornecida pelo protocolo CARP. Como pesquisa futura sugere-se estudar o processo de balanceamento de carga do protocolo CARP. Outra sugestão de trabalho seria analisar experimentalmente o desempenho do protocolo CARP utilizando vários níveis de tráfego.

## REFERÊNCIAS

- [1] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS – ABNT. NBR ISO/IEC 38500:2009 – Governança corporativa de tecnologia da informação. Rio de Janeiro: ABNT, 2008.
- [2] CARP. “The Common Address Redundancy Protocol”, Disponível em <http://www.opensbsd.org/faq/faq6.html#CARP>. Acessado em 08/04/2012.
- [3] Cisco, GLBP – Gateway Load Balancing Protocol. Disponível em [http://www.cisco.com/en/US/docs/ios/12\\_2t/12\\_2t15/feature/guide/ft\\_glbp.html](http://www.cisco.com/en/US/docs/ios/12_2t/12_2t15/feature/guide/ft_glbp.html). Acessado em 12/05/2012.
- [4] E. Lopes Filho, “Arquitetura de Alta Disponibilidade para Firewall e IPS baseada em SCTP”, Dissertação de Mestrado em Ciência da Computação, Universidade Federal de Uberlândia, Minas Gerais, 2008.
- [5] G. Attebury and B. Ramamurthy, “Router and Firewall Redundancy with OpenBSD and CARP”, Department of Computer Science and Engineering, University of Nebraska-Lincoln, in IEEE ICC 2006.
- [6] IT GOVERNANCE INSTITUTE. “Control Objectives for Information and Related Technology 4.1 (Cobit 4.1)”. 3701 Algonquin Road, Suite 1010 Rolling Meadows, IL 60008 USA 2007. Tradução e revisão pelo Projeto COBIT-BR.
- [7] OpenBSD Manual Pages, ifconfig (8) Disponível em [http://www.openbsd.org/cgi-bin/man.cgi?query=ifconfig&end=.](http://www.openbsd.org/cgi-bin/man.cgi?query=ifconfig&end=.Acessado em 08/04/2012.)
- [8] P. Danhieux, “CARP The Free Fail-over Protocol”, Global Information Assurance Certification Paper, SANS Institute, 2004
- [9] PF: Firewall Redundancy with CARP and pfsync, Disponível em <http://www.openbsd.org/faq/pf/carp.html>. Acessado em 08/04/2012.
- [10] R. Colton, D. Ferguson, J. Moy and A. Lindem, Ed., “OSPF for IPv6”, RFC 5340, July 2008.
- [11] S. Nadas, “Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6”, IETF, RFC 5798, 2010.
- [12] T. Li, B. Cole, P. Morton and D. Li, “Cisco Hot Standby Router Protocol (HSRP)”, IETF, RFC 2281, 1998.
- [13] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, “Neighbor Discovery for IP version 6 (IPv6)”, RFC 4861, September 2007.

# Modelagem de uma Base de Conhecimento para o Monitoramento de Ataques

Giani Petri, Tarcisio Ceolin Junior, Raul Ceretta Nunes, Osmar Marchi dos Santos

Programa de Pós-Graduação em Informática – PPGI

Universidade Federal de Santa Maria - UFSM

{gpetri, ceolin, ceretta, osmar}@inf.ufsm.br

**Resumo**—A crescente popularização da Internet vem acompanhada do aumento no número de ataques às vulnerabilidades. Diante disto, há uma necessidade de obter um conhecimento sobre a rede para monitorar possíveis ataques. Este trabalho propõe uma base de conhecimento chamada KBAM e apresenta a modelagem dos dados. A base KBAM representa informações de diferentes aspectos da rede. O uso da base KBAM em um estudo de caso permitiu a obtenção de uma consciência situacional do ambiente monitorado.

## I. INTRODUÇÃO

A popularização da Internet vem acompanhada do aumento no número de aplicações *web* que trabalham com informações críticas. Em paralelo a isso, é notório o acréscimo no volume de dados que trafegam pelas redes de computadores, bem como o aumento substancial no número de ataques às vulnerabilidades encontradas [1].

Neste cenário, os sistemas tradicionais de detecção de intrusão (IDS) estão tornando-se limitados. A quantidade de dispositivos conectados à rede, *petabytes* de dados, *gigabytes* de informações transferidas já não estão mais sendo suportadas pelos IDSs tradicionais [2].

Para suprir a necessidade de monitorar a Internet perante este novo cenário, a construção de *Internet Early Warning Systems* tem sido explorada [3][4]. O objetivo destes sistemas é defender e proteger as funcionalidades da Internet, detectando precocemente as ameaças. Além disso, permitem obter uma consciência situacional (percepção da situação de segurança dos recursos de rede) que possibilita uma reação precoce a um evento malicioso, um maior controle e monitoramento dos recursos envolvidos, auxiliando em tomadas de decisões [2].

Para realizar o monitoramento de ataques é preciso uma base de conhecimento que contenha diferentes aspectos sobre a rede monitorada e que dê suporte para as decisões das equipes de segurança. Estes aspectos correspondem aos dados sobre o comportamento normal da rede, informações sobre assinaturas de ameaças, incidentes e medidas de respostas [3].

Porém, os trabalhos existentes na literatura não englobam todos estes aspectos de uma base de conhecimento. Em [5] é proposto uma abordagem para detecção de intrusão utilizando redes neurais artificiais como mecanismo de detecção e uma base de conhecimento contendo assinaturas de ataques conhecidos para a fase de treinamento e aprendizagem, desconsiderando informações sobre medidas de respostas. Em [6] é apresentado um sistema que captura e analisa o tráfego de rede com o objetivo de criar uma base de conhecimento com regras que permita a tomada de decisões, porém esta proposta desconsidera o armazenamento de registros de incidentes.

Em [7] é proposta uma abordagem baseada em conhecimento para a modelagem de detecção de intrusão,

mas essa abordagem também não engloba medidas de respostas. Em síntese, existem diversos trabalhos que envolvem bases de conhecimento na literatura, entretanto, os mesmos não abordam todos os aspectos básicos necessários para uma base de conhecimento conforme citado em [3].

Este trabalho propõe uma base de conhecimento chamada KBAM (*Knowledge Base for Attacks Monitoring*), que engloba os diferentes aspectos de uma base de conhecimento voltada ao monitoramento de ataques. A base KBAM representa os dados de eventos de detecção de intrusão explorando o formato padrão *Intrusion Detection Message Exchange Format* (IDMEF) [8] para as mensagens de detecção de intrusão e o formato *Intrusion Detection Response Exchange Format* (IDREF) [9] para as mensagens de respostas. A representação dos dados contidos na base KBAM contempla os seguintes aspectos: dados de alertas gerados por sistemas de detecção de intrusão, informações sobre as medidas aplicadas em resposta a um alerta, estatísticas do tráfego da rede realizado através da coleta dos contadores dos parâmetros destacados em [10], além das assinaturas dos ataques já conhecidos.

Um estudo de caso realizado na rede da Universidade Federal de Santa Maria, demonstra que o uso da base KBAM permite a obtenção de uma consciência situacional do ambiente monitorado, pois fornece informações que possibilitam um conhecimento da atual situação de segurança da rede.

O restante do trabalho está organizado da seguinte forma. A seção II apresenta uma breve revisão bibliográfica, incluindo os trabalhos relacionados. A seção III apresenta a modelagem dos dados e destaca as principais tabelas e atributos da base KBAM. Na seção IV é realizada uma discussão sobre o uso da base KBAM e é apresentado um estudo de caso. Por fim, a seção V apresenta as conclusões do trabalho.

## II. REVISÃO BIBLIOGRÁFICA

Uma base de conhecimento é um repositório de dados que agrupa informações referentes a uma área específica [11]. Na arquitetura de um *Internet Early Warning System* a base de conhecimento é um dos componentes técnicos mais importantes, por manter informações que possibilitam ações mais efetivas, pois o objetivo é detectar ameaças precocemente, antes que elas possam causar qualquer perigo, ou antes de causar o máximo de perigo. Logo, criar uma consciência situacional, que corresponde a uma imagem da situação atual de segurança [3], depende das informações contidas na base.

De acordo com Bastke, Deml and Schmidt [3], as informações que devem ser armazenadas em uma base de conhecimento de um *Internet Early Warning System* devem corresponder aos seguintes aspectos: dados sobre o

comportamento normal da rede, informações sobre assinaturas de ameaças, incidentes e medidas de respostas.

Os trabalhos existentes na literatura não atendem a todos estes aspectos de uma base de conhecimento. Lima et al. [5] propõem uma abordagem para detecção de intrusão através do uso de redes neurais artificiais. A proposta utiliza, nas fases de treinamento e aprendizagem, uma base de conhecimento com regras de detecção previamente definidas e utiliza as redes neurais como mecanismo para detectar variantes de intrusões.

Flor et al. [6] propõem um sistema que captura e analisa o tráfego de rede. O objetivo é criar uma base de conhecimento com regras a partir da fusão dos dados do comportamento normal e malicioso, coletados por múltiplos sensores. Entretanto, a proposta apresentada em [6] não engloba todos os aspectos básicos de uma base de conhecimento, desconsiderando o armazenamento de informações sobre incidentes e suas medidas de respostas. Em [7], More et al. apresentam um *framework* para uma abordagem ontológica que utiliza diversas fontes para a coleta dos dados de detecção de intrusão e logs. Porém, esta abordagem não armazena informações referentes as respostas aos alertas de detecção.

Na área de detecção de intrusão existem alguns padrões para interoperabilidade de mensagens de detecção e de respostas. Estes padrões, baseados em XML (*eXtensible Markup Language*), definem uma formatação dos dados para serem compartilhados. Um desses padrões é o IDMEF. Criado pelo grupo IDWG (*Intrusion Detection Work Group*), o IDMEF é um formato de dados padrão que sistemas de detecção de intrusão utilizam para reportar e compartilhar alertas sobre eventos considerados suspeitos [8]. O principal objetivo do formato IDMEF é definir uma formatação de dados e procedimentos para a interoperabilidade entre sistemas de detecção de intrusão.

Umas das principais aplicações do formato IDMEF é para a comunicação de alertas entre o componente de análise e o gerenciador de um IDS. Além disso, o formato IDMEF também pode ser usado para a troca de informações e correlação de alertas, além da possibilidade de padronização de informações em um banco de dados.

Outro formato de dados que objetiva dar continuidade nos modelos desenvolvidos pelo grupo IDWG, criando mecanismos de envio de respostas aos alertas identificados, é o formato IDREF [9]. O IDREF é compatível com o modelo de alertas IDMEF, possibilitando assim, a integração dos dois modelos. O modelo IDREF, assim como o modelo IDMEF, também é orientado a objetos e suas classes foram projetadas com base nas informações contidas nos alertas formatados pelo modelo IDMEF.

### III. MODELAGEM DE UMA BASE DE CONHECIMENTO PARA O MONITORAMENTO DE ATAQUES

Esta seção apresenta o levantamento e a modelagem dos dados que estão representados na base KBAM.

#### A. Levantamento dos Dados

Os dados contidos na base KBAM representam os aspectos básicos de uma base de conhecimento, focando em incidentes de segurança específicos da área de detecção de intrusão.

##### 1) Informações sobre alertas de detecção de intrusão

O registro de informações históricas sobre os incidentes de segurança é importante para o auxílio na confirmação de eventos futuros. Com enfoque em eventos de detecção de intrusão a base KBAM representa os dados dos alertas disparados por IDSs em acordo com os atributos do formato IDMEF. Por ser um formato padronizado e orientado a objetos, o IDMEF permite uma flexibilidade na extensão de informações de ataques, através dos mecanismos de herança e agregação.

##### 2) Informações sobre medidas de respostas

A representação das respostas aplicadas aos eventos detectados está em acordo com as classes e atributos contidos no modelo de dados para troca de respostas de detecção de intrusão IDREF. Por ser um formato padrão similar ao IDMEF, os modelos possuem um forte relacionamento. Além disso, o formato IDREF também permite a representação de diversos tipos de respostas através de sua arquitetura orientada a objetos.

##### 3) Informações sobre assinaturas de ameaças

As assinaturas de eventos maliciosos já conhecidos e aceitos pela comunidade científica estão representadas em arquivos de regras padronizadas, conforme utiliza o IDS Snort [12]. A utilização de regras definidas possibilita um aumento na precisão da confirmação de atividades maliciosas já consolidadas.

Neste trabalho, assume-se que a representação das assinaturas das ameaças é integralmente realizada pela ferramenta Snort. Desta forma, o ambiente monitorado em que é utilizada a base KBAM, também deve estar equipado com a ferramenta Snort, que fica responsável pela detecção de eventos maliciosos a partir das regras previamente definidas. Assim, não há uma representação das assinaturas de ameaças na base KBAM, ficando esta tarefa com responsabilidade irrestrita da ferramenta Snort.

##### 4) Informações sobre o comportamento normal

Monitorar o tráfego de uma rede é essencial para obter uma visão de seu comportamento. O monitoramento dá-se através da quantificação dos dados sobre o tráfego que está ocorrendo em uma determinada rede. A quantificação é realizada através de um coletor, também conhecido como *sniffer*, que é responsável pela escuta e captura do que está acontecendo na rede e também pelo armazenamento desses dados.

Para criar uma visão real do comportamento, além de coletar estatísticas do tráfego é necessária a distribuição dos coletores em locais estratégicos do ambiente monitorado, para capturar informações em diferentes pontos da rede.

A seleção dos parâmetros quantificados a partir do tráfego da rede fundamenta-se nos descritores usados pela sonda do sistema IAS (*Internet Analysis System*) apresentado em [4] e detalhados em [10]. A sonda de um IAS trabalha de forma similar a um *sniffer*, realizando a captura de dados do tráfego de uma rede. Os parâmetros utilizados neste trabalho correspondem aos contadores dos protocolos TCP, ICMP, UDP, HTTP, SMTP, dentre outros.

#### B. Modelagem dos Dados

As informações armazenadas na base KBAM são representadas em um modelo relacional de dados. O modelo de dados é composto por 51 entidades que representam os dados dos alertas de detecção de intrusão,

as respostas a um alerta e as estatísticas sobre o tráfego da rede.

Os alertas de detecção de intrusão são representados através dos atributos do modelo IDMEF. A Figura 1 apresenta as principais entidades da base KBAM que armazenam informações sobre os alertas.

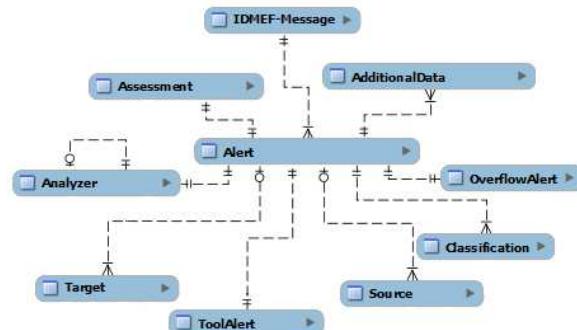


Figura 1. Principais entidades que representam os alertas de detecção.

Conforme mostra a Figura 1, a entidade que registra as informações referentes aos alertas disparados pelos detectores é a entidade *Alert*. O atributo *ident* armazena um identificador para o alerta, o instante da criação do alerta é armazenado no atributo *create-time*, o atributo *analyzer-time* armazena o momento em que o alerta foi disparado, já o instante em que o evento foi detectado está no atributo *detect-time*. A entidade *Alert* relaciona-se com as entidades *Assessment*, *Analyzer*, *Target*, *Source*, *ToolAlert*, *Classification*, *OverflowAlert* e *AdditionalData*.

A entidade *AdditionalData* armazena as informações que não se encaixam no modelo IDMEF. Já a entidade *OverflowAlert* representa informações específicas de alertas do tipo *overflow*. Por sua vez, a entidade *Assessment*, armazena as informações que permitem uma avaliação do evento causador do alerta. A entidade não possui atributos, porém relaciona-se com outras três entidades, *Impact*, *Action* e *Confidence*. Já a entidade *Analyzer* armazena informações referentes a identificação do analisador que originou o alerta. Os dados sobre o nome, a versão, classificação, modelo e fabricante do analisador ficam registrados nesta tabela, além de informações do tipo e versão do sistema operacional que o analisador atua. As entidades *Source* e *Target* correspondem, respectivamente, à possível origem e alvo do evento. Por sua vez, a entidade *Classification* registra uma possível classificação do tipo de alerta. A partir da classificação, o alerta pode ter alguma documentação externa que possua maiores informações referentes ao alerta gerado, essas informações ou *links* para os documentos ficam armazenadas na entidade *Reference*.

As principais entidades que representam os dados referentes as respostas aos alertas gerados estão modelados conforme apresenta a Figura 2.

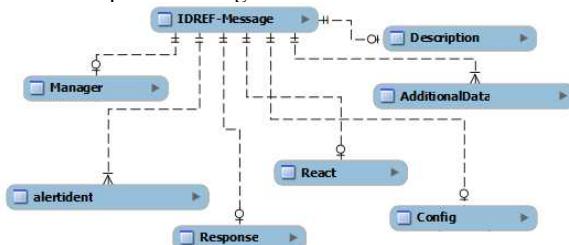


Figura 2. Entidades que representam as respostas aos alertas.

A principal entidade que representa uma resposta a um evento é a *IDREF-Message*. Esta entidade relaciona-se com *Response*, *React* e *Config*, que representam os tipos de respostas suportados pelo modelo IDREF.

O primeiro tipo de resposta é representado pela entidade *Response*, que representa o envio de informações cujo objetivo é avisar ou controlar um ataque. Uma resposta pode ser o envio de pacotes TCP e mensagem ICMP a um alerta ocorrido.

Outro tipo de resposta é através da alteração de configurações de um recurso do ambiente para conter um ataque, representada na entidade *Config*. Esta entidade relaciona-se com *Command* e *Resource*, que representam o(s) comando(s) a ser(em) executado(s) pelo recurso a ser configurado.

Além de permitir o envio de informações e a alteração de configurações, o modelo IDREF também permite a reação do ambiente contra um ataque. Este tipo de resposta é representado pela entidade *React* que possui dois relacionamentos, *Block* e *Shutdown*. As entidades *Block* e *Shutdown* representam respectivamente, o bloqueio e o fechamento de algum recurso.

A entidade *Resource* também está relacionada com as entidades *Block* e *Shutdown* e representa um recurso ao qual a resposta será aplicada. Um recurso pode ser um nó ou um serviço da rede, uma lista de usuários, uma lista de arquivos ou um processo do sistema operacional. Estes recursos são representados, respectivamente, pelas entidades: *Node*, *Service*, *UserList*, *FileList* e *Process*.

A quantificação dos dados capturados pelo *sniffer* para o captura dos contadores dos pacotes que trafegam na rede são armazenadas nas entidades apresentadas na Figura 3.

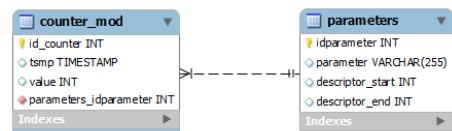


Figura 3. Entidades que representam a quantificação do tráfego da rede.

Conforme mostra a Figura 3, a entidade *parameters* contém todos os parâmetros que são capturados do tráfego da rede monitorada. O atributo *parameter* armazena a descrição do parâmetro utilizado e os atributos *descriptor\_start* e *descriptor\_end* contêm o intervalo dos descritores de cada parâmetro. Por sua vez, a entidade *counter\_mod* é responsável por armazenar todos os contadores dos pacotes capturados na rede. O momento da captura dos dados é armazenado no atributo *tsmp*, a quantificação dos pacotes está no campo *value* e a identificação do parâmetro é realizada através do relacionamento com a entidade *parameters*.

#### IV. DISCUSSÃO SOBRE O USO DA BASE KBAM

Obter uma consciência situacional do ambiente monitorado requer um conhecimento dos dados que trafegam na rede, bem como uma ciência da atual situação de segurança dos recursos envolvidos [4]. As bases de conhecimento dos trabalhos existentes na literatura [5][6][7] não permitem a construção de uma consciência situacional, pois focam somente em aspectos específicos.

Em contrapartida, a base KBAM atende os diferentes aspectos para a construção de uma base de conhecimento. Desta forma, os dados dos aspectos armazenados na base KBAM fornecem informações para a obtenção de um

conhecimento da atual situação de segurança do ambiente monitorado, permitindo a criação de uma consciência situacional.

A construção de uma consciência situacional é realizada a partir de dados coletados em pontos estratégicos da rede monitorada. Mas há diversas maneiras de popular a base KBAM para criar uma consciência situacional. Uma das formas é apresentada no estudo de caso realizado na rede da Universidade Federal de Santa Maria, descrito a seguir.

O estudo de caso envolveu dois pontos de monitoramento para a coleta de dados na UFSM: a rede do setor responsável pelo Vestibular (Coperves) e a rede do Centro de Processamento de Dados (CPD).

A coleta dos alertas de detecção de intrusão foi realizada através do uso do *framework* Prelude. O Prelude destaca-se como uma ferramenta que integra vários sensores distribuídos e possui como principal componente em sua arquitetura o *Prelude-Manager*, que trabalha como um servidor que aceita conexões de sensores distribuídos e armazena os eventos recebidos em um banco de dados [13].

Os sensores utilizados no estudo de caso são os sistemas de detecção de intrusão baseados em assinaturas Snort, em sua versão 2.8.5.2-2 e o Suricata [14], na versão 1.2.1. Os sensores utilizam o formato IDMEF para enviar ao *Prelude-Manager* os eventos detectados. Os eventos são armazenados em um banco de dados modelado de acordo com os dados do formato IDMEF.

No estudo de caso foram utilizadas três máquinas virtuais (VMs) com o sistema operacional Ubuntu Server 10.04.4 LTS x86-32 e o VMware Workstation 8 como monitor das máquinas virtuais. Na infraestrutura implementada, uma das VMs faz o papel do gerenciador, ou seja, nela estão instalados o *Prelude-Manager* e o banco de dados. As outras duas VMs realizam o processo de coleta de dados, utilizando para isso, os sensores Snort e Suricata.

Os dados coletados através da ferramenta Prelude são exportados automaticamente por um *script* que realiza a inserção dos eventos coletados na base KBAM. Além disso, as duas VMs que hospedam os sensores, também hospedam um *sniffer* que realiza a captura do tráfego da rede e armazena nas entidades da base KBAM.

O *sniffer* utilizado no estudo de caso é uma adaptação ao coletor SniffStat2DB, apresentado em [15]. A adaptação realizada no SniffStat2DB refere-se a implementação para a captura de dados de parâmetros utilizados no escopo deste trabalho.

Para armazenar os dados referentes a respostas, foi implementado um componente que gera respostas a um alerta no formato IDREF. O componente atende aos requisitos descritos em [8], permitindo a seleção de um alerta em uma lista de alertas, gerando as respostas ao alerta selecionado e as armazena na base KBAM.

Como resultado do caso de estudo, observa-se que a base KBAM contém informações que permitem a construção de uma consciência situacional sobre a segurança não só nos setores monitorados da instituição.

## V. CONCLUSÕES

O presente trabalho apresentou a modelagem dos dados da base de conhecimento KBAM. Armazenando dados

sobre alertas de detecção de intrusão, informações sobre as medidas aplicadas em resposta a um alerta e estatísticas do tráfego da rede, a base KBAM atende os diferentes aspectos necessários para a construção de uma base de conhecimento.

A realização de um estudo de caso na rede da Universidade Federal de Santa Maria permitiu coletar uma série de dados para inserir na base KBAM. Os dados coletados permitiram a obtenção de uma consciência situacional da rede da instituição.

Em um trabalho futuro, os dados coletados no estudo de caso serão minerados e utilizados para potencializar uma futura tomada de decisão das equipes de segurança.

## REFERÊNCIAS

- [1] Symantec Internet Security Threat Report Trends for 2011. April 2012. Available: [http://www.symantec.com/content/en/us/enterprise/other\\_resources/b-istr\\_main\\_report\\_2011\\_212393\\_64.en-us.pdf](http://www.symantec.com/content/en/us/enterprise/other_resources/b-istr_main_report_2011_212393_64.en-us.pdf)
- [2] M. Golling and B. Stelte. "Requirements for a future EWS - Cyber Defence in the internet of the future," in 3rd International Conference on Cyber Conflict (ICCC), Tallinn, Estonia, pp.1-16, June 7-10, 2011.
- [3] S. Bastke, M. Deml and S. Schmidt. "Internet Early Warning Systems – overview and architecture," European Workshop on Internet Early Warning and Network Intelligence, Hamburg, Germany, January 27, 2010.
- [4] M. Hesse and N. Pohlmann. "Internet Situation Awareness," in eCrime Researchers Summit, Atlanta, GA, pages 1-9, Oct. 2008.
- [5] I. Lima, J. Degaspari and J. Sobral. "Intrusion detection through artificial neural networks," in Network Operations and Management Symposium (NOMS), Salvador, Bahia, pages 867-870, April 7-11, 2008.
- [6] E. Flory, T. Anaya, C. Moody, M. Beheshti, J. Han and K. Kowalski. "A knowledge-based system implementation of intrusion detection rules," in Information Technology: New Generations (ITNG), 2010 Seventh International Conference on, Las Vegas, NV, pages 738–742, April 12-14, 2010.
- [7] S. More, M. Matthews, A. Joshi, T. Finin. "A Knowledge-Based Approach to Intrusion Detection Modeling," in *Security and Privacy Workshops (SPW), 2012 IEEE Symposium on*, pp.75-81, 24-25 May 2012.
- [8] H. Debar, D. Curry and B. Feinstein. "The Intrusion Detection Message Exchange Format (IDMEF)". RFC 4765. March, 2007.
- [9] P. F. Silva and C. B. Westphall. "An Intrusion Answer Model Compatible with the Alerts IDWG Model," in Network Operations and Management Symposium (NOMS), Vancouver, BC, pages 1-4, April 3-7, 2006.
- [10] G. Ricci. "Betrachtung der vom ias gesammelten Kommunikationsparameter auf Relevanz zur Anomalie und Angriffserkennung (Evaluation of the relevance for the detection of abnormalities and attacks of the communication parameters collected by the internet analysis system). Master's thesis, University of Applied Sciences, Gelsenkirchen, Germany, 2008.
- [11] S. Russel, P. Norving. "Artificial Intelligence: A modern approach," Prentice Hall, New York, 1<sup>st</sup> Edition, 1995.
- [12] M. Roesch and S. Telecommunications. "Snort - lightweight intrusion detection for networks," in 13TH USENIX CONFERENCE ON SYSTEM ADMINISTRATION. Proceedings, Seattle, Washington: USENIX Association, pages 229–238, 1999.
- [13] PRELUDE. Disponível em: <<http://www.prelude-technologies.com/en/welcome/index.html>>. Acesso em: jun. 2012.
- [14] SURICATA. Open Information Security Fundation. Disponível em: <<http://96.43.130.5/index.php/downloads>> Acesso em: jun. 2012.
- [15] A. H. Schmidt. "Sniffstat2DB: Uma Ferramenta para Coleta e Armazenamento dos Contadores do Tráfego de uma Rede" Trabalho de Graduação. Universidade Federal de Santa Maria, Santa Maria, 2009.

# Uso de OSPF para convergência de túneis IPSec

Marcelo Tschanz Camocardi

IPT – Instituto de Pesquisas Tecnológicas  
grupo.zender@gmail.com

Vanessa Pádua Muniz

IPT – Instituto de Pesquisas Tecnológicas  
vanessa.padua@gmail.com

Dr. Alexandre Barbieri Sousa

IPT – Instituto de Pesquisas Tecnológicas  
abarbieris@hotmail.com

**Resumo**— A busca pela alta disponibilidade é uma meta para diferentes tipos de empresas, principalmente para aquelas que possuem aplicações críticas para a continuidade de seus negócios. Porém, dispor de ambientes com altos índices de disponibilidade pode envolver grandes custos de aquisição e manutenção. Diante deste panorama, utilizar links não dedicados tende a ser uma realidade viável. Este artigo traz um estudo para avaliar disponibilidade e tempo de convergência utilizando-se de túneis IPSec sob links não dedicados com uso do protocolo de roteamento OSPF.

## I. INTRODUÇÃO

A suíte de protocolos *Internet Protocol Security* (IPSec) [2] se tornou um padrão em implementações que visam interligar localidades geograficamente distantes. Um dos grandes fatores que apoiam este padrão, é a interoperabilidade do IPSec permitindo, desta maneira, que o tunelamento através de *Virtual Private Network* (VPN) possa ocorrer entre diferentes fabricantes e soluções.

As VPNs são amplamente utilizadas em ambiente corporativo [9] possibilitando que o tráfego transite de maneira confiável entre diferentes unidades de negócio. Adicionalmente, as VPNs têm sido implantadas sob links *packet-switched* – *Asymmetric Digital Subscriber Line* (ADSL) - para prover alta disponibilidade de links dedicados [6], como por exemplo, *Multiprotocol Label Switching* (MPLS), aumentando desta maneira a resiliência da rede e do ambiente.

Para a convergência entre diferentes links, o roteamento se torna o fator chave [6]. Porém, atuar com o roteamento estático não seria a melhor abordagem em virtude da gerência das rotas. Para este artigo, foi escolhido o protocolo de roteamento dinâmico *Open Shortest Path First* (OSPF) por sua flexibilidade de configuração e demais características que serão descritas no decorrer deste artigo.

Com a convergência de tecnologias (voz, vídeo e dados), as operações e processos críticos das empresas, passam a ser realizadas através das conexões baseadas em IP, tornando imprescindível a disponibilidade das aplicações para a continuidade da empresa [6].

Solução de alta disponibilidade de links utilizando-se de conexões dedicadas tem dois fatores que podem leva a não adoção da mesma, sendo estas: 1. Custo elevado de links dedicados e 2. Dificuldade em ter 2 links dedicados de operadoras e circuitos diferentes.

A alternativa abordada neste artigo é a utilização de conexões compartilhadas do tipo ADSL ou Cable para atuar como *backup* das conexões dedicadas. Essa solução, mais econômica, torna possível a utilização de recursos de rede como protocolos de roteamento dinâmicos [9] em conjunto com a alta disponibilidade, além de trazer o fechamento dos três pilares de segurança: Confidencialidade, Integridade e Disponibilidade.

Outro ponto de atenção, é o tempo de convergência da rede que precisa ser aceitável, independente do tamanho da tabela de roteamento. Sabe-se que o tempo de convergência entre roteadores vizinhos, desde o inicio até sua finalização, pode ser rápido ou não, dependendo de fatores como protocolo utilizado, capacidade da conexão, mecanismos de autenticação, hardware entre outros.

Desta maneira, este artigo visa analisar o impacto na convergência de túneis IPSec sob links ADSL, utilizando-se do protocolo OSPF.

## II. METODOLOGIA

Para demonstrar o tempo de convergência de redes utilizando-se de OSPF, foi realizado um experimento composto de duas etapas. A primeira etapa é a análise da funcionalidade da solução através do programa GNS3 (<http://www.gns3.net>), que emula roteadores, e outros ativos de rede, com o uso da imagem do próprio equipamento físico, ou seja, o comportamento dos roteadores (exceto questões de desempenho) poderá ser analisado de forma real. A segunda etapa consiste em realizar a medição dos tempos de convergência com equipamentos reais.

A análise do tempo de convergência será medido conforme as fases demonstradas na Figura 1.



Figura 1. Diagrama de fases.

O cenário do experimento é composto da seguinte estrutura:

- Uma localidade denominada “Matriz” (Area 0) com 1 roteador central (*Core*) que possui seis rotas dinâmicas, aprendidas através de OSPF e 1 roteador de borda (ABR – Area Border Router), que faz a adjacência e troca de rotas entre roteadores vizinhos em localidades remotas (que serão descritas a seguir). Ambos vizinhos contam com conexões redundantes mas, apenas uma delas com tunelamento IPSec ativo.
  - Duas Localidades Remotas denominadas “Filial 1” (Área 1) e “Filial 2” (Área 2). Ambas contam com os mesmos recursos de *hardware*, *software* e conexões redundantes até a “Matriz”

### III. EXPERIMENTOS

Para minimizar o impacto de desempenho previsto nos roteadores de borda (ABR), o monitoramento ativo da conexão é realizado nos roteadores das Filiais “Filial 1” e “Filial 2”, no qual cada localidade realizará a gerência de suas próprias conexões. Como resultado, o consumo de recursos de *hardware* no ABR será menor, proporcionando maior escalabilidade ao ambiente. Vale ressaltar, que este monitoramento pode ser feito no ABR porém, em virtude da sobrecarga, é recomendável que esta tarefa seja feita nos roteadores das filiais.

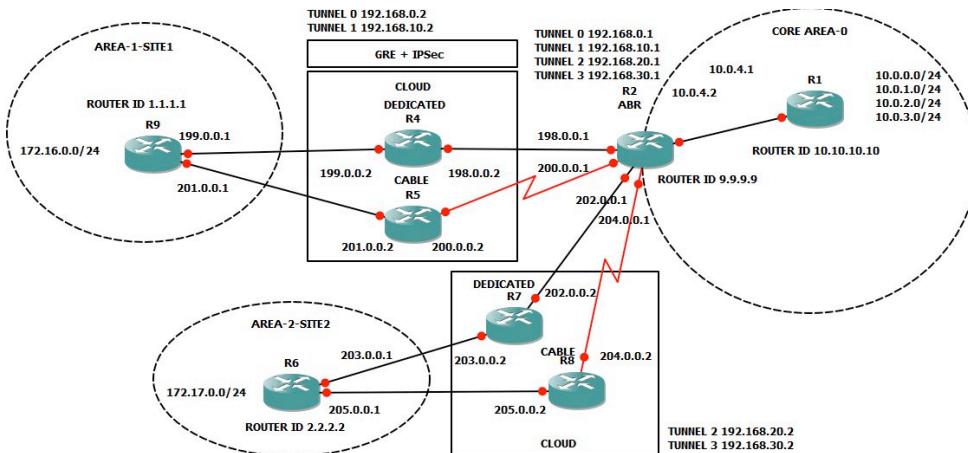


Figura 2. Topologia de Rede.

A seguir, são citados problemas que podem ocorrer e deverão ser mitigados antes da implantação deste ambiente.

- Se no mesmo roteador existirem duas conexões com dois túneis IPsec ativos simultaneamente, a vizinhança será fechada duas vezes com o mesmo vizinho, e pelo fato das conexões de túnel possuírem menor custo, o caminho que os pacotes tomarão não serão adequados e possíveis loops poderão ocorrer entre as interfaces virtuais.

- Em redes com conexões intermitentes poderá ocorrer o *flapping*, ou seja, a conexão se alterna entre o estado ativo e inativo. O mesmo acontece com os túneis e a vizinhança.

entre os roteadores, e como consequência ocorrerá aumento de processamento e alterações de topologia.

Em virtude do baixo custo e por suportar todas as funcionalidades requeridas para este ambiente, foram selecionadas as imagens de roteadores Cisco 2621.

As principais tarefas que os roteadores desempenham nesta topologia podem ser observadas na Figura 3.



Figura 3. Tarefas desempenhadas pelos roteadores das Filiais.

- Os eventos se desencadeiam quando o acesso a matriz se torna indisponível, seja por falha física ou lógica dos equipamentos.

- O monitoramento da disponibilidade da matriz é realizado através das ferramentas *IPSLA* e *tracking*. [4].

- As ações são tomadas com auxílio do recurso *event manager* [3].

```
ip sla monitor 1
type echo protocol ipIcmpEcho 198.0.0.2
timeout 1000
frequency 3
ip sla monitor schedule 1 life forever start-time now
!
track 1 rtr 1
```

Desta maneira, o endereço IP do ABR da matriz (198.0.0.2) é monitorado com frequência de 3 segundos e “timeout” de 1 segundo. O monitoramento começa a ser

realizado a partir do momento em que o equipamento é ligado. O comando “track” é responsável pelo alerta em caso de indisponibilidade.

```

event manager applet TUNNEL-1-LIGA
event syslog pattern "%TRACKING-5-STATE: 1 rtr 1
state Up->Down"
action 1.0 cli command "enable"
action 2.0 cli command "configure terminal"
action 3.0 cli command "interface tunnel 1"
action 4.0 cli command "no shutdown"
action 5.0 syslog msg "Tunnel 1 LIGADO"
action 6.0 cli command "end"
action 7.0 cli command "exit"
event manager applet TUNNEL-1-DESLIGA
event syslog pattern "%TRACKING-5-STATE: 1 rtr 1
state Down->Up"
action 1.0 cli command "enable"
action 2.0 cli command "configure terminal"
action 3.0 cli command "interface tunnel 1"
action 4.0 cli command "shutdown"
action 5.0 syslog msg "Tunnel 1 DESLIGADO"
action 6.0 cli command "end"
action 7.0 cli command "exit"

```

Quando o status se torna “down” é gerada uma mensagem de console no roteador. Este evento é capturado pelo *event manager* e a operação de ativação do túnel IPSec redundante é realizada de forma automatizada.

Para o processo de transferência do tráfego de uma interface para outra, é utilizado o comando “Track” em conjunto com o “Event Manager” e o “Action”. Desta maneira, o “track” gera mensagens no terminal, em seguida o “Event Manager” captura essas mensagens e caso estas possuam “1 rtr 1 state Up->Down” os comandos informados nas linhas “action” para habilitar e desabilitar as interfaces de túnel do roteador serão realizados.

Na Figura 4 é demonstrado o fluxo das atividades quanto ao acesso dedicado até a matriz:

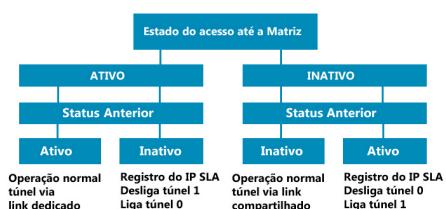


Figura 4. Fluxo de ações dinâmicas realizadas por evento nos roteadores das Filiais.

Para a análise e medição dos tempos de convergência, foram escolhidos roteadores de pequeno porte, sendo estes: Cisco 1841 para as Filiais e Cisco 2801 para a Matriz (ABR).

Através de um script em PHP (*Personal Home Page*) foram criadas rotas estáticas direcionadas para a interface *NULL 0* e redistribuídas para o processo OSPF, gerando assim 1500 rotas e 20.000 rotas consecutivamente. Para alterar as velocidades dos links foi utilizado o comando “clock rate” na interface serial, variando entre 512.000 bits por segundo a 2.000.000 bits por segundo.

Para simular tráfego, foi utilizado o programa Jperf. O tráfego foi gerado entre dois servidores, alcançando 70 conexões simultâneas em redes diferentes e roteadas através da técnica de *Inter Vlan Routing* (Roteamento entre redes virtuais) no ABR.

Os tempos de convergência foram medidos através do programa WinPing com frequência de mensagens a cada 1 segundo e timeout de 1 segundo. Através do *timestamp* foi possível identificar exatamente o tempo de indisponibilidade.

Na Figura 5 é ilustrado no WinPing, o comando ping sendo executado a partir da Filial 1 para a Matriz com o intuito de monitorar o tempo de indisponibilidade de acesso ao ambiente da Matriz.

```

1 Reply from 10.0.3.1 in 6 ms : Bytes: 32 ; Time: 00:00:00.982
2 Reply from 10.0.3.1 in 6 ms : Bytes: 32 ; Time: 00:00:01.980
3 Reply from 10.0.3.1 in 6 ms : Bytes: 32 ; Time: 00:00:02.980
4 Reply from 10.0.3.1 in 6 ms : Bytes: 32 ; Time: 00:00:04.039
5 Reply from 10.0.3.1 in 6 ms : Bytes: 32 ; Time: 00:00:05.047
6 Reply from 10.0.3.1 in 6 ms : Bytes: 32 ; Time: 00:00:06.036
7 Reply from 10.0.3.1 in 6 ms : Bytes: 32 ; Time: 00:00:07.036
8 Reply from 10.0.3.1 in 6 ms : Bytes: 32 ; Time: 00:00:08.095
9 Reply from 10.0.3.1 in 6 ms : Bytes: 32 ; Time: 00:00:09.095
10 Reply from 10.0.3.1 in 6 ms : Bytes: 32 ; Time: 00:00:10.094
11 Reply from 10.0.3.1 in 6 ms : Bytes: 32 ; Time: 00:00:11.154
12 Reply from 10.0.3.1 in 7 ms : Bytes: 32 ; Time: 00:00:12.152
13 TimedOut
14 TimedOut
15 DestinationHostUnreachable
16 DestinationHostUnreachable
17 DestinationHostUnreachable
18 DestinationHostUnreachable
19 DestinationHostUnreachable
20 DestinationHostUnreachable
21 DestinationHostUnreachable
22 DestinationHostUnreachable
23 DestinationHostUnreachable
24 DestinationHostUnreachable
25 DestinationHostUnreachable
26 Reply from 10.0.3.1 in 2 ms : Bytes: 32 ; Time: 00:00:28.003
27 Reply from 10.0.3.1 in 2 ms : Bytes: 32 ; Time: 00:00:29.063
28 Reply from 10.0.3.1 in 2 ms : Bytes: 32 ; Time: 00:00:30.062
29 Reply from 10.0.3.1 in 2 ms : Bytes: 32 ; Time: 00:00:31.123

```

Average of the response times = 5,06 ms

Figura 5. Retorno do Winping.

Como coleta de resultados do experimento, os seguintes dados foram obtidos:

Banda	Tabela de Roteamento	Carga ABR	Convergência Up → Down	Convergência Down → Up	Processamento ABR
512 kbps	6 rotas	70 Conexões 100 mbps	13 segundos	16 segundos	Entre 23% - 32%
512 kbps	1500 rotas	70 Conexões 100 mbps	17 segundos	11 segundos	Entre 25% - 29%
512 kbps	20000 rotas	70 Conexões 100 mbps	36 segundos	16 segundos	Entre 19% - 35 %
512 kbps	6 rotas	sem carga	13 segundos	10 segundos	Entre 0 % - 1 %
512 kbps	1500 rotas	sem carga	15 segundos	12 segundos	Entre 2 % - 4 %
512 kbps	20000 rotas	sem carga	35 segundos	16 segundos	Entre 6 % - 9 %
1 mbps	6 rotas	70 Conexões 100 mbps	11 segundos	14 segundos	Entre 28 % - 35 %
1 mbps	1500 rotas	70 Conexões 100 mbps	14 segundos	13 segundos	Entre 12 % - 27 %
1 mbps	20000 rotas	70 Conexões 100 mbps	32 segundos	14 segundos	Entre 24 % - 37 %
1 mbps	6 rotas	sem carga	12 segundos	11 segundos	Entre 0 % - 1 %
1 mbps	1500 rotas	sem carga	15 segundos	10 segundos	Entre 1 % - 4 %
1 mbps	20000 rotas	sem carga	24 segundos	19 segundos	Entre 2 % - 7 %
2 mbps	6 rotas	70 Conexões 100 mbps	17 segundos	13 segundos	Entre 23% - 31%
2 mbps	1500 rotas	70 Conexões 100 mbps	19 segundos	10 segundos	Entre 17 % - 26 %
2 mbps	20000 rotas	70 Conexões	26 segundos	23 segundos	Entre 27 % - 34 %

Tabela 1. Resultados alcançados

A partir destes dados foi possível observar que em ambientes com grande quantidade de rotas o tempo de convergência tende a aumentar, conforme demonstra o gráfico Figura 6.

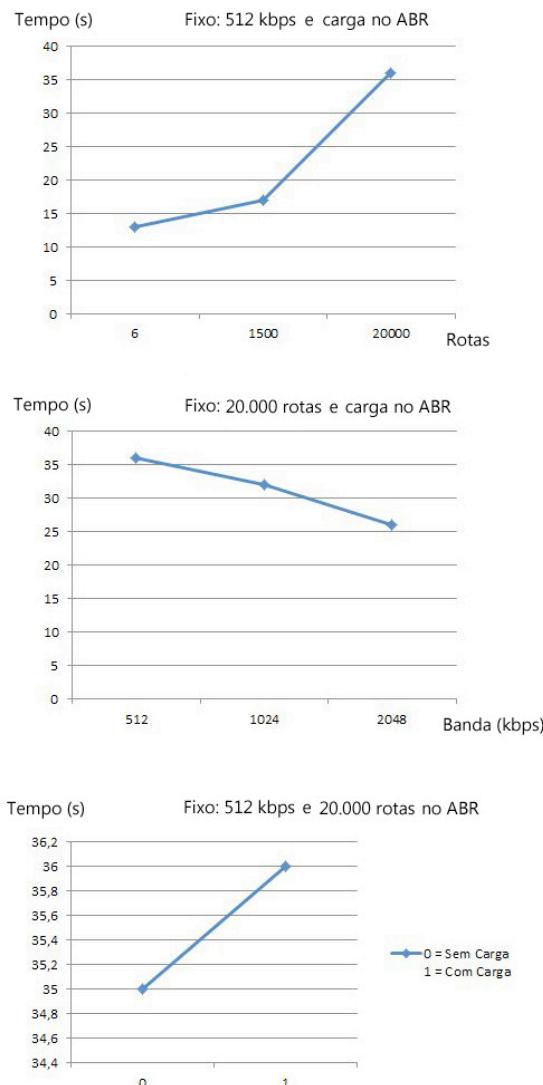


Figura 6. Gráfico com resultados do experimento

#### IV. CONCLUSÃO

Com a análise dos resultados alcançados, foi possível concluir que a solução atende a todos os cenários propostos com tempos de convergência abaixo de 36 segundos. Foram trabalhadas variações nas velocidades das conexões e mesmo com valores baixos, os tempos continuaram com pouca variação.

Adicionalmente, foi possível identificar que mesmo com uma grande quantidade de rotas e carga, o tempo de convergência não sofreu alterações sensíveis tornando a opção viável para ambientes de grande porte.

Desta maneira, o tempo de indisponibilidade de uma empresa que possui 2 links (dedicado e ADSL) para

prover interconectividade e contingência de suas localidades, é mínimo e sendo aceitável para que os serviços corporativos sejam rapidamente reestabelecidos. Como trabalhos futuros, é interessante uma análise deste experimento em ambientes com alto consumo de recursos de hardware dos roteadores envolvidos.

#### REFERÊNCIAS

- [1] Aman Shaikh, *Student Member, IEEE*, Mukul Goyal, Albert Greenberg, *Member, IEEE*, Raju Rajan, and K.K. Ramakrishnan, *Member, IEEE* An OSPF Topology Server: Design and Evaluation IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 20, NO. 4, MAY 2002.
- [2] Barylski, Marcin. On IPSec Performance Testing of IPv4/IPv6 IPSec Gateway. 1st International Conference on Information Technology, IT 2008, IEEE.
- [3] Cisco IOS Embedded Event Manager (EEM). Disponível em <[http://www.cisco.com/en/US/products/ps6815/products\\_ios\\_proto\\_col\\_group\\_home.html](http://www.cisco.com/en/US/products/ps6815/products_ios_proto_col_group_home.html)> Acesso em 8 mar 2012.
- [4] IP Service Level Agreements (IP SLAs). Disponível em <[http://www.cisco.com/en/US/tech/tk920/tsd\\_technology\\_support\\_sub-protocol\\_home.html](http://www.cisco.com/en/US/tech/tk920/tsd_technology_support_sub-protocol_home.html)> Acesso em 8 mar 2012.
- [5] Kini, Shrinivas et al. Fast Recovery from Dual Link Failures in IP Networks. IEEE INFOCOM, 2009, IEEE.
- [6] Kuboniwa, Akiko. IPsec-GW redundancy method with high reliability. Information and Telecommunication Technologies (APSITT), 2010 8th Asia-Pacific Symposium, 2010, IEEE.
- [7] Kyriacos Manousakis, Anthony J. McAuley. Using Stochastic Approximation to Design OSPF Routing Areas that Satisfy Multiple and Diverse End-to-End Performance Requirements. 6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops, 2008, IEEE.
- [8] Mikko Rukanen, Marko Luoma. OSPF Flooding Process Optimization. High Performance Switching and Routing, 2005, IEEE.
- [9] Okhravi, Johnson, Haines, Mayberry, Chan. Dedicated vs. Distributed: A Study of Mission Survivability Metrics. Military Communications Conference, 2011, IEEE.
- [10] Song Wang, Hongbing Lv. A distributed object-based IPSec multi-tunnels concurrent architecture ICCP2011 Proceedings. International Conference on Computational Problem-Solving (ICCP), 2011, IEEE.

# Service Desk Móvel com Retenção de Conhecimento e Sensível ao Contexto

Taciano Balardin de Oliveira  
PPGI – UFSM  
tacianobalardin@gmail.com

Roseclea Duarte Medina  
PPGI – UFSM  
roseclea.medina@gmail.com

**Resumo**—A gerência dos problemas ocorridos em ambientes que fazem uso da Tecnologia da Informação (TI), aliada a necessidade de uma resposta rápida das equipes de suporte, faz com que organizações necessitem de sistemas para gerenciamento desses incidentes. O Service Desk apresenta-se como uma boa solução para centralizar estes registros. Conceitos de computação baseada em contexto, bases de conhecimento, computação móvel podem incrementar estes aplicativos. Portanto, o objetivo deste trabalho é projetar e desenvolver um sistema de *Service Desk Mobile*, que agrupa funcionalidades de sensibilidade ao contexto, tais como localização do usuário, experiência do técnico e tempo. Além disso, a ferramenta possui uma base de conhecimento para que interações passadas fiquem armazenadas e sejam sugeridas como possível solução para novos problemas.

## I. INTRODUÇÃO

O aumento crescente da dependência das organizações na utilização da tecnologia de informação (TI) vem tornado o gerenciamento de serviços de TI dentro desses ambientes uma atividade cada vez mais importante [8]. Na ocorrência de problema em algum equipamento de informática presente nestes locais gerenciados (e.g., computador, impressora, *software* ou qualquer dispositivo que ocasiona o funcionamento anormal dos serviços de TI), a expectativa é que o usuário tenha uma resposta rápida da equipe de suporte para que os prejuízos causados possam ser minimizados [9].

Para que os problemas pudessem ser centralizados e posteriormente solucionados por técnicos responsáveis por estas tarefas, criou-se o conceito de serviço de atendimento aos usuários de TI que, de acordo com [7], foi inicialmente denominado de *Help Desk*. Entretanto, em consonância com [9][12], atualmente esta área absorveu outros serviços e passou a chamar-se de *Service Desk*, tratando-se de uma versão estendida do *Help Desk* e oferecendo uma quantidade maior de serviços.

Um desafio que atinge os responsáveis por estes ambientes de gerenciamento é que em diversas organizações existe uma alta rotatividade dos recursos humanos de TI. Em 2010, a rotatividade do pessoal de TI em todo o mundo era de apenas 3%. Em 2011, de acordo com o instituto de pesquisas Gartner [5], saltou para 5%. Logo, a saída de um funcionário representa uma perda de capital humano e gera um custo de substituição (i.e., recrutamento, seleção, contratação e treinamento) que pode ser alto. Além disso, existe a dificuldade de transferir o conhecimento e experiência entre os colaboradores da área [18].

Outro fator que deve ser considerado em um *Service Desk*, diz respeito à alocação de recurso humano que

tenha o perfil adequado para resolução dos diferentes tipos de problemas. Assim, um *Service Desk* onde os técnicos trabalham sem planejamento algum, atendendo aos chamados desordenadamente, ou sem ter a expertise (i.e., experiência e prática) necessária para resolver o problema relatado, pode ter sérios problemas com a perda de tempo, com deslocamento desnecessário ou com a alocação errada de membros da equipe [8][9].

A computação baseada em contexto usa informações sobre um usuário ou ambiente para melhorar a qualidade de interação com o sistema que, munido destes dados, age de forma proativa, antecipando as necessidades do utilizador e provendo uma maior adaptação às suas atividades em tempo de execução. De acordo com pesquisa de [6], até o ano de 2013 as aplicações sensíveis ao contexto aparecerão em áreas como: serviços baseados em localização, realidade aumentada em dispositivos móveis e comércio móvel. Além disso, os mesmos estudos apontam que existe uma tendência de que, até 2013, os celulares ultrapassem os computadores pessoais como dispositivos mais comuns para acesso à web.

Uma possível solução para o problema de rotatividade de pessoal é a agregação de uma base de conhecimento em um sistema de *Service Desk*, pois, dessa forma, o conhecimento empregado pelos técnicos em situações anteriores pode ficar retido no sistema e ser apresentado como possível solução para problemas com características semelhantes. Por sua vez, a computação baseada em contexto pode auxiliar na alocação de chamados, para que sejam solucionados utilizando informações do ambiente como: localização, à expertise e também o horário de expediente de cada técnico que realiza atendimento.

Este trabalho é continuidade de pesquisas realizadas pelo GRECA (Grupo de Redes de Computadores e Computação Aplicada) da UFSM (Universidade Federal de Santa Maria) e consiste em parte de uma Dissertação para obtenção do título de Mestre em Computação. Tem como principal objetivo projetar e desenvolver uma ferramenta de *Service Desk* móvel, sensível ao contexto e que possua detecção de expertise necessária para alocação do técnico que atenderá os chamados de suporte. Além disso, munir esta ferramenta com uma base de conhecimento, para que o capital intelectual, gerado através dos atendimentos realizados pela equipe de suporte, fique retido no sistema e seja apresentado como possível solução para novos chamados.

O artigo está organizado da seguinte forma, na seção II é apresentada a revisão bibliográfica, enquanto a seção III aborda o desenvolvimento do aplicativo e suas características. Na seção IV constam os resultados obtidos até o momento e, por fim, a seção V traz as conclusões parciais do trabalho.

## II. REVISÃO BIBLIOGRÁFICA

### A. Sistemas de Service Desk

Com o aumento da demanda empresarial e da globalização, cada vez mais as organizações precisam se certificar da qualidade dos serviços executados, para obterem maiores chances no mercado. Com isso, o objetivo de um *Service Desk* é prover aos usuários de TI um ponto único de contato vital para uma comunicação efetiva entre os usuários e as equipes que gerenciam a TI em uma organização. Sua principal missão é o restabelecimento da operação normal dos serviços dos usuários o mais rápido possível, minimizando o impacto nos negócios causados por falhas de TI [15].

O funcionamento de um sistema de *Service Desk* se dá através da abertura de chamados ou *tickets*. A partir deste momento, sempre que houver um chamado em aberto, o mesmo é gerenciado para que seja atendido. Além disso, esses sistemas também podem ser baseados em práticas de alguma metodologia para manutenção de serviços de TI, como, por exemplo, o ITIL (*Information Technology Infrastructure Library*).

### B. Computação Ubíqua e Cliente ao Contexto

O termo Computação Ubíqua foi definido pelo cientista Mark Weiser [14] como um novo paradigma onde a computação deve estar invisível para o usuário. A computação ubíqua prevê a integração e comunicação de diversos dispositivos e recursos (i.e., *software* e *hardware*) em um ambiente real de forma que o usuário possa realizar alguma atividade sem ter a consciência da utilização dos recursos computacionais [9][14].

A computação ciente de contexto é uma área de pesquisa da Computação Ubíqua, que tem o objetivo de prover o acesso de sistemas computacionais às informações de contexto, ampliando a comunicação entre ser humano e sistemas computacionais, assim permitindo o desenvolvimento de serviços computacionais mais úteis e adaptáveis [1].

Um sistema é considerado ciente de contexto se utilizar elementos do contexto para fornecer informações e/ou serviços relevantes para o usuário, sendo a relevância dependente das tarefas a serem realizadas pelo mesmo [1][2]. Assim, espera-se que sistemas computacionais cientes de contexto não somente respondam quanto ao estado social e cognitivo do usuário, mas também antecipem suas necessidades [2][4].

### C. Base de Conhecimento

Uma Base de Conhecimento (BC) é um tipo especial de banco de dados para gestão do conhecimento, uma BC fornece um meio das informações serem coletadas, organizadas, compartilhadas, conhecidas e utilizadas.

Um dos objetivos da implantação de uma arquitetura baseada em conhecimento utilizada como suporte nos ambientes de gerenciamento de TI é encurtar a curva de aprendizado de um membro ingressante na equipe. Esta aceleração da absorção do conhecimento permitirá que o colaborador ingressante assuma de forma plena as tarefas do antigo, minimizando assim o impacto da rotatividade, bem como utilizar o conhecimento aplicado a resolução

de um problema para auxiliar na resolução de outro que tenha características semelhantes [18].

Assim, muitas empresas tentam criar sistemas inteligentes de suporte técnico para melhorar a qualidade de serviço ao cliente. A partir de um novo chamado do usuário, o objetivo de um sistema de *Service Desk* integrado a uma base de conhecimento é descobrir se chamados semelhantes foram processados antes. Sistemas de *Service Desk* geralmente usam bancos de dados para armazenar interações passadas (e.g., descrições de um problema e as soluções recomendadas para sua resolução) [18].

### D. Mineração de Texto e Similaridade entre Strings

Na literatura existem diversas técnicas para calcular similaridade entre *strings*, tais como a remoção de *stopwords*, *stemming*, modelo de índice invertido, algoritmo *Levenshtein Distance*, processamento de linguagem natural.

1) *Remoção de Stopwords*: Um conjunto de *strings* que compõem um documento é formado por algumas palavras (*tokens*) que não possuem valor semântico, sendo útil apenas para que o texto possa ser compreendido de forma geral. Em um sistema de mineração de dados, tais palavras são consideradas as *stopwords* e pertencem a uma *stoplist*. Com uma *stoplist* bem organizada é possível eliminar até 50% do total de palavras de um texto [10].

2) *Stemming*: Seu objetivo é a redução de cada palavra até que seja obtida sua raiz através da eliminação de sufixos que indicam variação na forma da palavra, como plural, tempo verbal, locução adverbial, aumentativo, gênero, acentuação. Por exemplo as palavras “duvido, dúvida, duvidamos, duvidem” se submetidas a uma técnica de *stemming* tornam-se “duvid” [17]. Segundo pesquisa apresentada em [16], a utilização de *stemming* melhorou em 35% a recuperação de informações em alguns conjuntos de dados.

3) *Algoritmo Levenshtein Distance*: Define a distância de diferença entre duas *strings*, é baseado no número mínimo de operações necessárias para transformar uma *string* em outra. Para isso, existem três operações que transformam uma sequência de caracteres em outra: inserção (i.e., insere um novo caractere na *string* “destino”), eliminação (i.e., elimina um caractere na *string* “origem”) e substituição (i.e., substitui um caractere na *string* “origem”, com o objetivo de transformar na *string* “destino”). Um exemplo pode ser os passos necessários para transformar “casas” em “massa”, que define a distância de edição em 3 (substituição de ‘c’ por ‘m’; eliminação do segundo ‘a’; inserção de ‘a’ no final da *string*) [10].

Outras informações sobre métodos para mineração de texto e cálculos de similaridade entre *strings* podem ser encontradas em [3], [10], [11] e [17].

## III. SERVICE DESK MÓVEL COM RETENÇÃO DO CONHECIMENTO E SENSÍVEL AO CONTEXTO

Desenvolvido na linguagem de programação PHP, que é responsável por gerar o conteúdo dinâmico do sistema e

executado em uma arquitetura cliente-servidor. Para que possam ser armazenados todos os dados dos chamados, técnicos, usuários, prédios, salas e também reter o conhecimento gerado pelos atendimentos aos usuários, utiliza o SGBD (Sistema de Gerenciamento de Banco de Dados) MySQL.

O sistema possui uma interface de usuário unificada, ou seja, aplicação única para todos os dispositivos móveis suportados e sistemas operacionais. Para proporcionar essa interface unificada, é utilizado o framework *jQuery Mobile*, que baseia-se no HTML 5 (*HyperText Markup Language*), nas bibliotecas *jQuery* e *jQuery UI* e tem como característica ser otimizado para interações por toque [13].

O HTML 5 também é aplicado no sistema para dar suporte ao contexto de localização geográfica, pois é através dele que a latitude, longitude e altitude dos usuários são requisitadas pelo *Service Desk*.

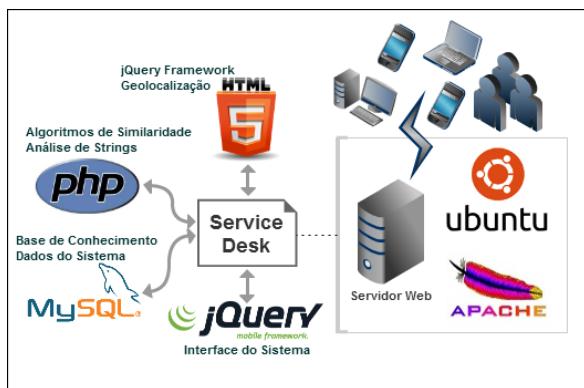


Figura 1. Arquitetura do Sistema de *Service Desk*.

A Figura 1 apresenta a arquitetura do *Service Desk*, dando uma ideia geral sobre as tecnologias empregadas em seu desenvolvimento e a forma como os usuários acessam o ambiente.

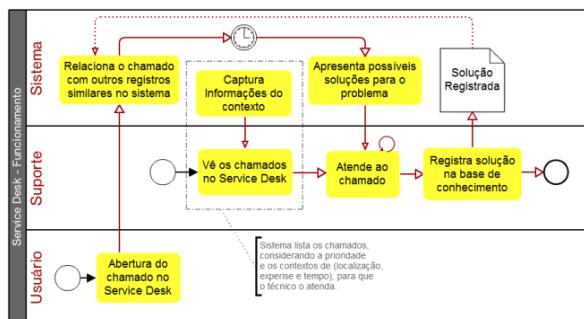


Figura 2. Funcionamento do Sistema de *Service Desk*.

Por sua vez, a Figura 2 mostra como acontece o funcionamento do sistema. A partir da abertura de um chamado pelo usuário, o sistema automaticamente relaciona este com chamados anteriores, com isso, é possível sugerir à equipe de Suporte possíveis soluções relacionadas ao atendimento em aberto. Para relacionar os chamados são utilizados algoritmos de similaridade e também mineração de texto.

A partir do momento em que um usuário de nível Suporte acessa o sistema, são listados os chamados em aberto, por ordem de prioridade, considerando informações do contexto, como por exemplo, localização atual do técnico, seu horário de expediente e também se possui a expertise necessária para atendimento do problema.

Ao iniciar um atendimento, são apresentadas sugestões de problemas relacionados, pois a solução aplicada em algum destes problemas talvez possa ser reutilizada para resolver o novo. Após concluir o atendimento, o técnico vai registrar a solução utilizada na base de conhecimento do sistema, para que a mesma possa servir como solução para outros problemas.

Não apresentado no esquema da Figura 2, mas de vital importância para o funcionamento do sistema, está o Gestor, que é responsável por gerenciar o sistema de *Service Desk*, cadastrando a equipe de suporte, bem como a expertise que cada membro possui. Além disso, controla os usuários que tem acesso ao sistema para abertura de chamados e também os prédios/salas que, ao serem cadastrados, tem o contexto de localização detectado pelo sistema. Com isso, na abertura do chamado para resolver um problema em algum prédio ou sala, é possível identificar o técnico mais próximo do local e que atenda a expertise necessária para solucioná-lo.

#### IV. RESULTADOS PARCIAIS

A aplicação está instalada em um servidor com o sistema operacional Ubuntu, na versão 12.04 LTS, utilizando arquitetura de 32 bits. Neste está instalado e configurado o banco de dados MySQL 5.5 e o servidor web Apache 2.2, com suporte a linguagem de programação PHP 5.3.

Como o *Service Desk* é executado através da *internet* e, atualmente, encontra-se em fase de desenvolvimento e testes, o escopo de abrangência foi definido para o acesso da rede interna da UFSM e externamente através de uma rede privada virtual, utilizando o aplicativo OpenVPN.



Figura 3. Tela de Login.

Figura 4. Abertura de Chamado.

As capturas de telas apresentadas na Figura 3 e Figura 4 foram realizadas a partir de um *smartphone* com sistema operacional Android 2.3.

Na tela apresentada pela Figura 3, o usuário informa seus dados de acesso para entrar no sistema. Caso ainda

não seja cadastrado, tem a possibilidade de cadastrar-se, a partir disso, assim que o Gestor confirmar seu registro, também poderá acessar o ambiente.

A Figura 4 mostra parte do formulário para registrar um novo chamado no *Service Desk*. O campo “Local” é identificado automaticamente de acordo com a posição geográfica do usuário, desde que o dispositivo disponha de tal funcionalidade e o utilizador autorize o acesso do aplicativo a essa informação. Ainda assim, mesmo que identifique a localização, é possível informar outro local onde o problema ocorreu. O campo “Contato” é o nome do responsável pelo chamado, já o campo “Descrição do Problema” serve para escrever sobre o incidente a ser resolvido pela equipe de suporte. Por fim, o campo “Categoria” categoriza os problemas de acordo com algumas características, como por exemplo, se o problema é em um equipamento ou então em um *software*.

Até o momento, o sistema foi executado nos dispositivos apresentados na Tabela 1, que mostra o nome do dispositivo, o Sistema Operacional que o mesmo executa e a aplicação em que o sistema foi testado (S.O / App). Além disso, informa se a interface (Int.) carregou corretamente e também se as funcionalidades (Func.) da ferramenta foram executadas de forma satisfatória, ou seja, sem a ocorrência de erros ou algum problema que impeça a conclusão de alguma atividade no *Service Desk*.

Tabela I  
DISPOSITIVOS TESTADOS

Dispositivo	S.O / App	Int.	Func.
Notebook	Ubuntu - Firefox	Ok	Ok
Notebook	Win7- Chrome	Ok	Ok
Notebook	Win7 - I.E	Ok	Ok
Smartphone	Android 2.3 - Browser Nativo	Ok	Ok
Smartphone	iOs3 - Browser Nativo	Ok	Ok
Smartphone	Symbian - Opera Mini	Ok	Ok
Tablet	Android 2.2 - Browser Nativo	Ok	Ok

De acordo com a Tabela 1, em todos os dispositivos testados a aplicação se comportou de forma correta, executando as funcionalidades e a interface do *Service Desk* de forma plena.

## V. CONCLUSÃO

Sistemas de *Service Desk* centralizam os problemas que acontecem em ambientes gerenciados por equipes de TI. A utilização cada vez maior de dispositivos móveis e o avanço das tecnologias abrem novas possibilidades para expansão deste tipo de serviço, sempre visando à melhoria da qualidade e rapidez nos atendimentos prestados pelos técnicos de suporte.

Até o momento, de acordo com os testes apresentados na Tabela I, o *framework* de desenvolvimento utilizado tem suportado todos os dispositivos testados, além disso, não é necessário criar um aplicativo para cada sistema operacional, seja ele de um dispositivo móvel ou não, todos executam a mesma aplicação.

De acordo com os estudos realizados, as técnicas de mineração de textos apresentam-se como uma possível solução para a identificação de chamados na base de conhecimento, a fim de avaliar a similaridade entre estes e o chamado a ser atendido pela equipe de suporte. Também podem solucionar o problema de identificar o contexto de

expertise necessária para atender certo chamado, baseando-se em experiências anteriores.

Além disso, obter a localização, tanto do técnico quanto do problema a ser solucionado, pode tornar o processo de resolução mais rápido, pois prioriza os técnicos mais próximos do incidente para atendê-lo, desde que estejam em horário de expediente e possuam a expertise necessária para realizar a tarefa.

Como trabalhos futuros, os autores pretendem que o sistema sugira possíveis soluções para problemas simples e recorrentes, sem que seja necessária a intervenção de um técnico, bem como um estudo para melhoria no algoritmo de análise de similaridade entre chamados. Também é possível tornar a ferramenta escalável, para que funcione em um ambiente de *cloud computing*.

## REFERÊNCIAS

- [1] A. Dey, “Understanding and Using Context. Personal and Ubiquitous Computing” In Journal Personal and Ubiquitous Computing 5(1):4-7, 2001.
- [2] C. Jardim “Usando Serviços Web para integrar aplicações clientes de contexto”, Dissertação (Ciências da Computação e Matemática Computacional) – Universidade de São Paulo, São Paulo, SP, 2006.
- [3] C. N. Aranha and E. P. L. Passos “A Tecnologia de Mineração de Textos” RESI. Revista Eletrônica de Sistemas de Informação, v. 2, p. 2, 2006.
- [4] D. Siewiorek “New frontiers of application design” In: Communications of the ACM, 45(12):79–82, 2002.
- [5] Gartner “CIO Alert: U.S. IT Staff Turnover Trends and Analyses” In: <http://www.gartner.com/id=771513>, 2011.
- [6] Gartner “Top 10 Strategic Technologies for 2012 - Analysts Examine Latest Industry Trends During Gartner Symposium” In <http://www.gartner.com/it/page.jsp?id=1826214>, 2011.
- [7] G. Cavalari, H. Costa “Modelagem e Desenvolvimento de um Sistema Help-Desk para a Prefeitura Municipal de Lavras - MG” In: Revista Eletrônica de Sistemas de Informação, 2005.
- [8] I. Magalhães, W. Pinheiro “Gerenciamento de Serviços de TI na Prática: Uma abordagem com base na ITIL”. São Paulo: Novatec, 2007.
- [9] J. Lobo “Contexto de Expertise e Localização Influenciando a Gerência de TI”, Dissertação (Mestrado em Computação) - Universidade Federal de Santa Maria, Santa Maria, RS, 2011.
- [10] J. R. C. Junior “Desenvolvimento de uma Metodologia para Mineração de Textos” Dissertação (Mestrado em Engenharia Elétrica), Rio de Janeiro/RJ PUC-Rio, 2007.
- [11] M. Bendersky and W. B. Croft “Finding Text Reuse on the Web” In: Proceedings of the Second ACM International Conference on Web Search and Data Mining, Barcelona, 2009, pp. 262-271.
- [12] M. Jäntti, J. Kalliokoski “Identifying Knowledge Management Challenges in a Service Desk: A Case Study” In: eKNOW '10 - Second International Conference on Information, Process, and Knowledge Management, p. 100-105, 2010.
- [13] M. S. Silva “jQuery Mobile - Desenvolva aplicações web para dispositivos móveis com HTML5, CSS3, AJAX, jQuery e jQuery UI” Novatec, 2011.
- [14] M. Weiser “The Computer for the Twenty-First Century”. In: Scientific American, pages 94–10, 1991.
- [15] R. Cohen “Gestão de Help Desk e Service Desk” 1 ed. São Paulo: Novatec, 2011. 296 p. ISBN 978-85-7522-276-8.
- [16] R. Krovetz “Viewing morphology as an inference process” In: ACM SIGIR Conference on Research and Development in Information Retrieval, 1993, pp. 191-202.
- [17] V.N. Orenco “A stemming algorithm for the portuguese language” In: String Processing and Information Retrieval, 2001, pp. 186-193
- [18] W. Dingding, T. Li, S. Zhu, Y. Gong “iHelp: An Intelligent Online Helpdesk System”. In: IEEE transactions on systems, man, and cybernetics, vol. 41, issue 1, p. 173-182, 2011.

---

# VI

## Fórum de Pós-Graduação II

---



# CEP – Uma proposta de gerenciamento de identidades em *Cloud Computing* utilizando OpenAM e Captive Portal

Andreia Rosangela Kessler Mühlbeier  
UFSM  
andreiarkmuhlbeier@gmail.com

Felipe Becker Nunes  
UFSM  
nunesfb@gmail.com

Gleizer Bierhalz Voss  
UFSM  
gleizer.voss@gmail.com

Samuel Stieler  
UFSM  
samuel.stieler@gmail.com

Roseclea Duarte Medina  
UFSM  
roseclea.medina@gmail.com

Érico Marcelo Hoff do Amaral  
UNIPAMPA  
ericohoffamaral@gmail.com

**Resumo** - Este artigo apresenta uma proposta de gerenciamento de identidades em um ambiente de *Cloud Computing* utilizando Captive Portal em conjunto com os recursos do OpenAM. O modelo abrange mecanismos de controle de acesso baseado no contexto de papéis provendo uma Camada Extra de Proteção (CEP) a um ambiente de computação em nuvem.

## I. INTRODUÇÃO

A Computação em Nuvem (*Cloud Computing*) vem se tornando um paradigma da área computacional com grande evolução no cenário atual. O aumento de usuários no decorrer da última década propiciou o desenvolvimento de alternativas positivas para a resolução de questões importantes relacionadas às pessoas e empresas.

Esse paradigma aparece com o intuito de disponibilizar serviços de tecnologia da informação sob demanda, no qual o pagamento é baseado conforme a sua utilização. No entanto, há algumas preocupações relacionadas à segurança que precisam ser resolvidas, por exemplo, confidencialidade, autenticidade e integridade das informações e aplicações armazenadas em nuvem [1].

Usuários carecem de garantias rígidas que suas informações estarão bem protegidas pela empresa responsável. Em conformidade com essa situação, diversas pesquisas e propostas têm sido apresentadas [2] [3] [4] na tentativa de quebrar este cenário de desconfiança e assim buscar fornecer maiores níveis e garantias de segurança.

Diante deste contexto, este trabalho apresenta uma proposta de controle de acesso para um ambiente de *Cloud Computing*, com a utilização dos mecanismos OpenAM e Captive Portal. Estes realizam a autenticação e autorização dos usuários no ambiente, fornecendo assim uma Camada Extra de Proteção (CEP).

Este trabalho está organizado da forma que segue: na seção 2 são apresentados os trabalhos relacionados; a seção 3 descreve questões sobre segurança; o gerenciamento de identidades é descrito na seção 4; na seção 5 são descritos os métodos de controle de acesso OpenAM e Captive Portal e dados sobre a implementação desta proposta; por fim na seção 6 é apresentado as considerações finais sobre a pesquisa realizada.

## II. TRABALHOS RELACIONADOS

O trabalho descrito em [2] apresenta uma solução alternativa a um *Identity as a Service* (IDaaS), com o objetivo de fornecer um gerenciamento de identidades baseado no conceito de identidade digital federada. A solução utiliza o Shibboleth, uma ferramenta baseada em *Security Assertion Markup Language* (SAML), que fornece apoio às tarefas de autenticação, autorização e federação de identidades. Desta forma, os autores destacam que é possível oferecer um serviço que permite ao mesmo tempo acesso público, nos casos de acesso apenas para leitura, ou pode exigir credenciais, solicitando ao usuário uma conexão validada para alterar documentos.

A proposta de [3] apresenta uma solução de segurança para o serviço de nuvem baseado em *Usage Control* (UCON), um modelo apenas conceitual, e sem especificação de realização concreta. O modelo UCON é composto por seis partes: Assuntos, Direitos, Objetos, Autorizações, Obrigações e Condições. Como trabalho futuro, os autores pretendem criar o protótipo do sistema de segurança para o serviço de *cloud*.

O trabalho descrito em [4] propõe a criação de identidade centralizada para o gerenciamento de identidade na nuvem. A abordagem é baseada em pacotes ativos e identificação anônima. Destacam-se nessa solução: a independência de terceiros, o fornecimento de informações mínimas ao *Service Provider* (SP) e a capacidade de usar dados de identidade em *hosts* não confiáveis.

Pode-se relacionar ainda o trabalho de [5], que apresenta a proposta de uma arquitetura para uma nova abordagem de "proteção mútua". Ela é baseada no conceito de confiança mútua e na especificação de perfis definidos na forma de um vetor matricial. Por fim tem-se [6] que apresenta um protótipo, utilizando a tecnologia de agentes como uma forma de oferecer privacidade aos dados dos clientes em um ambiente de *cloud computing*. A abordagem utiliza predicados sobre os dados criptografados e computação segura *multi-party computation* (MPC) para a negociação na utilização de um serviço na nuvem.

### III. SEGURANÇA EM CLOUD COMPUTING

Quando o assunto de computação em nuvem é abordado, a segurança é uma das objeções mais frequentemente citadas, apesar da maioria das empresas terceirizarem os pagamentos e utilizarem serviços de *e-mail* externos [8]. Dentre as principais preocupações dos usuários e empresas, estão questões referentes sobre quais usuários terão acesso às informações e quais os riscos que existem em utilizar uma aplicação armazenada em um servidor na nuvem. O fato de que uma nuvem é composta por um aglomerado de informações pode marcá-la como um alvo propício para ataques de potenciais invasores [7].

Deste modo, as organizações precisam verificar os riscos existentes em disponibilizar suas informações e aplicações em servidores na nuvem em relação as suas necessidades, avaliando dessa forma a solução que irá proporcionar um maior número de vantagens para o seu negócio.

### IV. GERENCIAMENTO DE IDENTIDADES

Em ambientes que possuam restrições de acesso, por exemplo, uma nuvem que hospeda aplicações e recursos, o usuário terá que realizar a sua identificação junto da entidade de verificação desse ambiente, conforme as permissões concedidas pela entidade.

Uma identidade pode ser definida como uma representação de um elemento, de forma que seja possível identificá-la dentro de um contexto em particular [9]. Um usuário ou aplicação necessitará de um identificador, como: CPF ou *e-mail*, para ser reconhecido no ambiente.

Conforme este contexto, a gerência de identidades realiza o processo de controle, com o objetivo de permitir que uma entidade possa entrar no ambiente desejado, de acordo com suas permissões de acesso.

#### A. Shibboleth

Shibboleth [10] é a solução de identidade federada mais utilizada mundialmente, sua arquitetura é composta por componentes livres e *open source*, principalmente o *Security Assertion Markup Language* (SAML). O SAML é um padrão aberto no formato *eXtensible Markup Language* (XML) para a troca de dados de autenticação e autorização em um domínio de segurança. O Shibboleth fornece um mecanismo de *Single Sign-On* (SSO) [2], que conforme a definição do *Open group* é um mecanismo que permite um usuário cadastrado acessar por meio de uma única autenticação, todos os sistemas a que tenha permissão de acesso.

#### B. Higgins

Higgins [11] é um *Personal Data Service* (PDS) que permite ao usuário controlar como e com quem os seus dados serão compartilhados, seja com amigos ou organizações em que confia. Esse projeto foi desenvolvido pela Eclipse Foundation, e possui quatro partes principais: o PDS, que é o *Back-end* de serviços de apoio ao cliente *Web*; o *Attribute Data Storage* (ADS), que expõe os dados para o Portal e o *Higgins Browser Extension* (HBX), que utiliza uma interface de mensagens HTTP/Comet; Cliente que é a interface que permite ao usuário ver e editar os atributos; e por fim o HBX que é a extensão que carrega os programas *JavaScript* para o PDS e roda-os no navegador.

### C. OpenAM

O OpenAM [12] foi inicialmente denominado de OpenSSO e desenvolvido pela Oracle. Porém com a aquisição desta pela Sun Microsystems, ele passou a ser desenvolvido pela ForgeRock sob o nome de OpenAM.

Ele é um sistema de código aberto (*open source*), que prove os serviços de autenticação, autorização, verificação de validade de *tokens*, *login* e provisão de identidades. O OpenAM possui três formas de acesso: utilização de uma requisição HTTP, por meio dos serviços *Web* ou com a utilização de um agente. Ele pode ser considerado um IDaaS que realiza a provisão de identidades e controle de acesso [13]. Sua arquitetura é subdividida em 3 camadas: interface do cliente, núcleo e a camada de integração.

### D. Captive Portal

Segundo [14], o Captive Portal funciona como um roteador ou *gateway*, não permitindo que haja tráfego de informações antes da autenticação do usuário. Ele obriga este a visualizar uma página de login ou acesso, onde o usuário deve realizar uma autenticação para obter acesso ao local que deseja, por exemplo, aplicações e informações armazenadas em uma nuvem.

Esta tecnologia faz o monitoramento de pacotes quando um usuário realiza o acesso a uma aplicação ou base de dados, sendo a conexão deste redirecionada para uma página de *login*, na qual ocorre a autenticação/autorização de acesso. A transmissão de informação entre o usuário e a página de acesso é criptografada com a utilização do protocolo *Secure Socket Layer* (SSL) em ambas as direções. [15].

### E. Mecanismos escolhidos

A Tabela I apresenta um comparativo dos mecanismos de controle de acesso mencionados anteriormente. Podemos observar as principais diferenças com relação ao suporte de padrões abertos e também sobre o tipo de paradigma que cada um possui.

Tabela I  
Comparativo dos mecanismos de controle de acesso

Mecanismos Soluções	Shibboleth 2x	Higgins 2.0	OpenAM 10
SSO	X	X	X
SAML 2.0	X	X	X
OpenID 2.0	-	-	Plugin
XACML 2.0	Extenção	-	X
OAuth 1.0	-	-	Plugin
S.Agentes	-	-	X
Paradigma	Federado	C. Usuário	Federado

Fonte - Adaptado de Feliciano 2011.

Baseado na tabela comparativa apresentada anteriormente, foi escolhido o mecanismo OpenAM em relação ao Shibboleth e o Higgins. Os motivos para esta escolha são os seguintes:

- O OpenAM permite a implementação de novos padrões por meio da instalação de *plugins*, por exemplo, o OpenID e OAuth.
- Ele possui uma arquitetura flexível e paradigma centrado no usuário, que possibilita o compartilhamento de informações de identidades.
- Está sendo utilizado de forma crescente pelos usuários e possui bastante tempo de desenvolvimento. Além disso, teve sua última versão lançada no mês de junho de 2012, sendo esta a escolhida para ser utilizada.

Adicionalmente ao mecanismo do OpenAM, será utilizado o Captive Portal com o objetivo de fornecer uma camada extra de proteção ao ambiente da nuvem. Com isso, é realizado o gerenciamento de identidades e o controle de acesso de forma conjunta. O Captive Portal foi escolhido pelo fato de poder ser integrado ao OpenAM, impedindo os usuários de acessar o ambiente caso não tenham realizado a autenticação.

## V. CONTROLE DE ACESSO UTILIZANDO OPENAM E CAPTIVE PORTAL

É apresentado um controle de acesso para realizar a autenticação/autorização dos usuários em um ambiente de nuvem, na qual está hospedado o ambiente virtual de aprendizagem Moodle, softwares para a gerência de rede e um laboratório virtual. Além destes serviços, é realizado o controle de papéis dos usuários, de forma a restringir o acesso às páginas somente para perfis específicos, impedindo o acesso não autorizado.

O controle de acesso utiliza os serviços do Captive Portal, que adiciona uma camada extra de proteção, trabalhando em conjunto com o OpenAM para atender aos requisitos de segurança citados anteriormente. Simultaneamente realiza o gerenciamento de perfis, permitindo acesso somente aos pontos que lhe são liberados no ambiente.

### A. Arquitetura de Controle

Com o desafio de idealizar um controle de acesso diferenciado, a arquitetura precisa ser também um diferencial. A arquitetura de controle da *Cloud* é simples, porém rica em detalhes demonstrando o seu diferencial quando se trata em segurança, conforme mostra a Figura 1. Ela é definida a partir de três elementos: o Captive Portal, o OpenAM e o servidor de banco de dados e *login*.

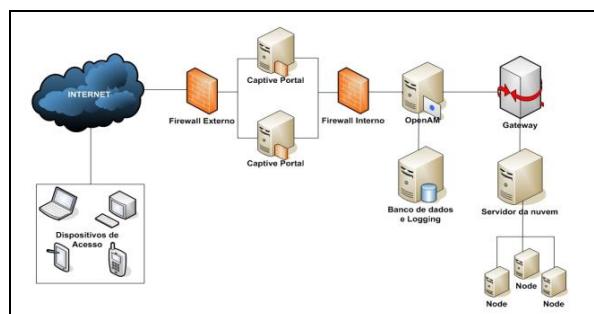


Figura 1. Visão geral da arquitetura de controle do sistema CEP.

O Captive Portal irá atuar como um agente, com o papel de verificar se o usuário que está solicitando o acesso aos recursos disponíveis na *Cloud* já realizou a sua autenticação no ambiente. Caso contrário, será redirecionado para uma página especial de *login*, onde deverá inserir seus dados, por exemplo, CPF e RG. A conexão entre o usuário e a página de acesso é criptografada com a utilização do protocolo SSL em ambas as partes. O desenvolvimento do Captive Portal será feito com a utilização da linguagem de programação PHP e com o sistema gerenciador de banco de dados MySQL.

O OpenAM é responsável pela parte de autenticação e autorização dos usuários no ambiente, além de realizar o gerenciamento dos perfis. Ele funciona com um mecanismo de autenticação *Single Sign-On* (SSO), conforme mencionado anteriormente, que faz a troca de mensagens via XML e certificação via SAML, sendo esses os principais padrões do mecanismo. O OpenAM gerencia o uso de agentes, que serão os responsáveis por enviar a requisição via Captive Portal para o OpenAM, em que é realizada a consulta das identidades de cada usuário.

O servidor de banco de dados e *login* tem a função de armazenar as informações dos usuários cadastrados no ambiente da nuvem, para que a entidade que for realizar a identificação verifique se um usuário é legítimo ou não. Ele também possui uma tabela que deve armazenar os logs gerados por cada acesso ou tentativa mal sucedida de acesso ao ambiente, para quando necessário analisar o comportamento deste. Outro detalhe é a existência de uma flag no banco de dados, com valores de V (Ativo) ou F (Inativo), que indicam se a sessão está ativa ou não.

### B. Funcionamento do controle de acesso

Para que os usuários de uma *Cloud* obtenham permissão de acesso aos serviços hospedados, é necessária a realização de algumas etapas de autenticação e autorização. Na Figura 2 pode ser visualizado o fluxograma da proposta, no qual é apresentado as etapas de funcionamento do controle de acesso, descritas de forma detalhada.

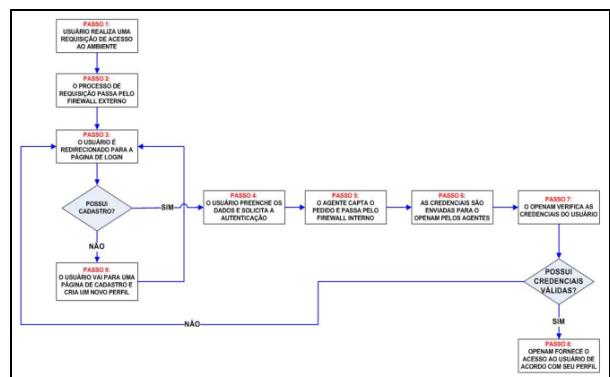


Figura 2. Diagrama de fluxo de dados.

No passo 1, o usuário realiza o acesso ao endereço do serviço na nuvem pelo *browser*, utilizando qualquer equipamento, por exemplo, *notebook*, *desktop* ou dispositivo móvel; No passo 2, o processo de requisição passa primeiramente por um *firewall* externo, que tem como função principal a verificação de segurança por meio das regras de acesso pré-definidas pelo administrador de rede, protegendo a nuvem de ataques maliciosos externos;

No passo 3, o *firewall* externo redireciona o usuário para a tela do Captive Portal, que exibe uma página especial criada para realizar o *login* (passo 4). Caso o usuário não possua cadastro, ele é redirecionado a uma página para a criação de um novo perfil no ambiente (passo 9). No passo 4, o usuário deve preencher os

campos necessários (número de matrícula, senha, instituição/curso) e solicitar a autenticação pelo Captive Portal.

No passo 5, o agente irá interceptar a requisição no momento da autenticação passando por um *firewall* interno, que terá a função de proteger as aplicações instaladas;

No passo 6, as credenciais do usuário são encaminhadas pelo agente para o OpenAM; No passo 7, o OpenAM verifica se o usuário possui as credenciais necessárias para o acesso com uma consulta ao banco de dados;

No passo 8, o OpenAM autoriza o acesso a nuvem, restringindo a navegação somente às páginas que o perfil do usuário tenha permissão. Caso não possua as credenciais de acesso válidas, é redirecionado para a página de *login* do Captive Portal por meio do agente.

Para a criação de um novo perfil (passo 9), o usuário deverá efetuar o cadastro, recebendo as credenciais de acesso e sendo redirecionado para a página inicial.

O novo perfil de usuário é definido de forma padrão (básico), tendo acesso somente aos recursos permitidos para este perfil. Os outros dois tipos são denominados de intermediário e avançado, cada um com recursos específicos, conforme Tabela II. O tipo de perfil poderá ser alterado através de solicitação ao administrador do sistema, que analisará a situação e concederá ou não a alteração.

O acesso as páginas é autorizado conforme o tipo do perfil que o usuário possui, desta forma, este poderá desempenhar determinadas ações, por exemplo, leitura e modificação, caso o seu perfil possibilite realizá-las. Esta forma de controle permite que arquivos e informações confidenciais sejam acessados somente por perfis que tenham permissão, possibilitando assim uma maior segurança em relação ao funcionamento do ambiente.

No momento em que um usuário cadastrado for realizar o acesso ao ambiente, será gerado um *log* de acesso criado por um *script* de rotina do banco de dados, que armazena esses *logs* no mesmo servidor. Ao mesmo tempo, é incrementado o valor de uma *flag* localizada na base de dados criando uma sessão. O valor desta pode ser V (Ativa) ou F (Inativa), indicando se o usuário pode navegar pelo ambiente. Um *time out* controla o tempo ocioso do usuário, obrigando-o a realizar a autenticação novamente, caso tenha expirado o limite estabelecido pelo administrador.

Tabela II  
Tipo de Níveis de Permissões de Acesso

Tipo de Permissões	Básica	Média	Avançada
Ler	X	X	X
Enviar Arquivos	X	X	X
Copiar Arquivos	X	X	X
Modificar/Excluir Arquivos		X	X
Criar/Excluir Tarefas e Fóruns		X	X
Alterar dados perfil	X	X	X
Modificar permissões			X
Acesso Total			X

## VI. CONSIDERAÇÕES FINAIS

A computação em nuvem é um paradigma relativamente recente, cujo objetivo é disponibilizar serviços de tecnologia da informação sob demanda. Desta forma, diversas propostas vêm surgindo com o objetivo de apresentar soluções e melhorias para garantir a segurança nestes ambientes.

Com base na análise descrita na seção 2, foi apresentada a proposta de utilização de uma camada extra de proteção com o uso do Captive Portal associado ao mecanismo OpenAM, que tem como funções: autenticação/autorização e o gerenciamento dos perfis dos usuários no ambiente da nuvem.

Após a descrição da proposta e do funcionamento de sua arquitetura, espera-se que seja possível realizar a inserção da Camada Extra de Proteção (CEP) ao controle de acesso. Com isso, busca-se unir todos os recursos oferecidos pelo mecanismo do OpenAM às funcionalidades que o Captive Portal disponibiliza, agregando um maior nível de segurança ao controle de acesso no ambiente da nuvem.

Como trabalho futuros, propõe-se a implementação dos testes e validação dos resultados obtidos com a aplicação, a fim de verificar a consistência da proposta.

## REFERÊNCIAS

- [1] F. R. C. Souza, L. O. Moreira; J. C. Machado. Computação em Nuvem: Conceitos, Tecnologias, Aplicações e Desafios, ERCEMAPI, 2009.
- [2] M. A. P. Leandro, T. J. Nascimento, D. R. dos Santos,; C. M. Wetphall, C. B. Wetphall. Multi-Tenancy Authorization System with Federated Identity for Cloud-Based Environments Using Shibboleth. ICN 2012: The Eleventh International Conference on Networks.
- [3] C. Danwei, H. Xiuli, R. Xunyi. Access control of cloud service based on UCON. CloudCom, 2009.
- [4] P. Angin, L. B. O. Lilien, M. Linderman. An Entity-centric Approach for Privacy and Identity Management in Cloud Computing. SRDS 2010, 29th IEEE International Symposium on Reliable Distributed Systems.
- [5] A. Albeshri, W. Caelli. Mutual Protection in a Cloud Computing Environment. HPCC 2010, 12th IEEE International Conference on High Performance Computing and Communications.
- [6] R. Ranchal, B. Bhargava, L. B. Othmane, L. Lilien, A. Kim, M. Kang, M. Linderman. Protection of Identity Information in Cloud Computing without Trusted Third Party. SRDS 2010, 29th IEEE International Symposium on Reliable Distributed Systems.
- [7] R. C. C. Castro, V. L. P. Sousa. Segurança em Cloud Computing: governança e gerenciamento de riscos de segurança. Info Brasil, 2010.
- [8] M. A. Armbrust et al. A view of Cloud Computing. Communications of the ACM, v. 53, no. 4, April, 2010.
- [9] T. El Maliki, J. M. Seigneur. A Survey of User-centric Identity Management Technologies. In Proceedings of the The International Conference on Emerging Security Information, Systems, and Technologies, 2007.
- [10] SHIBBOLETH - site oficial. Disponível em:<<http://shibboleth.net/>>. Acesso em: Jun. 2012.
- [11] HIGGINS - Personal Data Service. Disponível em:<<http://www.eclipse.org/higgins/>>. Acesso em: Jun. 2012.
- [12] OpenAM Project. Disponível em: <<http://openam.forgerock.org/>>. Acesso em: Jul. 2012.
- [13] ForgeRock. Disponível em: <<http://openam.forgerock.org/>>. Acesso em: Jul. 2012.
- [14] K. J. Hole, E. Dyrnes, P. Thorsheim. Securing Wi-fi networks - Captive Portals. IEEE Computer Society, v. 38, pages 28-34, July, 2005.
- [15] L. G. Machado. "CPAut" Uma Arquitetura de Controle de Acesso para o CRSPE/INPE - MCT. Trabalho de Graduação apresentado ao Curso de Graduação em Ciência da Computação Bacharelado, da Universidade Federal de Santa Maria (UFSM, RS), 2006.

# Ambiente de acesso seguro a nuvem privada: uma proposta voltada à rede da UNIPAMPA

Douglas Pires  
Borges  
dborges@inf.ufsm.br

Maurício Sulzbach  
sulzbach@uri.edu.br

Andrea Schwertner  
Charão  
andrea@inf.ufsm.br

Benhur de  
Oliveira Stein  
benhur@inf.ufsm.br

Roseclea Duarte  
Medina  
rose@inf.ufsm.br

**Resumo**—A computação em nuvem apresenta-se como um novo paradigma para a oferta de serviços na Web e tem como principal ideia, a transferência da grande maioria do processamento e armazenamento das aplicações dos usuários, para um ambiente remoto de serviços. Apesar de essa abordagem trazer novas oportunidades de negócios, traz consigo também dúvidas relativas à questão da segurança e do gerenciamento das informações. Isso tem feito com que muitas instituições tenham receio em migrar seus serviços para um ambiente em nuvem. Diante disso, esse trabalho tem por objetivo apresentar uma proposta para a migração dos serviços locais, para uma infraestrutura de nuvem privada para a Universidade Federal do PAMPA – UNIPAMPA. Além disso, essa proposta também irá prover um sistema de autenticação através de *Single Sign On* (SSO), visando o acesso aos diferentes serviços da nuvem através de um único usuário.

## I. INTRODUÇÃO

O avanço da tecnologia tem realizado muitas mudanças, apresentando uma nova realidade às pessoas e às empresas, criando novas oportunidades de negócios, formas de comunicação e entretenimento. A cada curto período, novas tecnologias e dispositivos surgem, com a necessidade de aumentar o desempenho, a segurança, a confiabilidade e principalmente, trazer benefícios e satisfação ao usuário. Nesse sentido, a computação em nuvem ou *cloud computing* é um conceito que está emergindo rapidamente, e vem auxiliando muito esse processo de transformação. Diversas aplicações e serviços que eram anteriormente providos por uma estrutura interna em instituições, estão sendo portadas para a nuvem. Essa mudança traz consigo uma série de benefícios como o aumento do poder computacional, redução da infraestrutura e de custos, alocação de recursos conforme a necessidade, escalabilidade e alta capacidade de tolerância a falhas. Porém, migrar serviços para a nuvem também gera preocupações aos usuários com a questão de segurança e de gerenciamento das informações, uma vez que não se sabe onde e como os dados são armazenados e quem realmente tem acesso à informação.

Além disso, a *cloud*, por prover e unificar diferentes recursos e como forma de facilitar o gerenciamento e auditoria (revisão de regras de autenticação, autorização e verificação das atividades do usuário), necessita que os usuários sejam únicos para os diversos serviços, conceito este conhecido por *Single Sign On* (SSO). Nesse ponto, as Identidades Federadas, ou do inglês, *Federated Identity*, criam um suporte importante para que os usuários possam ter *Single Sign On* ao acessar diferentes aplicações na nuvem. Sendo assim, esse trabalho apresenta uma proposta para migração de serviços que estão hospedados internamente na Universidade Federal do PAMPA –

UNIPAMPA, para uma nuvem privada, bem como, um modelo de autenticação e acesso seguro a um ambiente de computação em nuvem, através da utilização de Identidades Federadas. Nesse trabalho, optou-se por um ambiente de nuvem privada, devido ao fato da mesma permitir um nível de segurança maior sobre as informações e possibilitar o gerenciamento dos recursos pelo setor de tecnologia da informação da Universidade.

## II. TRABALHOS CORRELATOS

Chen, Sun e Hu [1] descrevem o processo de autenticação dos serviços e aplicações em uma rede universitária, fazendo o uso de *Single Sing On* (SSO).

Já Bhosale [2] aborda um estudo de caso de uma aplicação bancária, que visa fornecer diversos serviços aos seus clientes através da Internet. Nesse artigo, discute-se uma solução de SSO que forneça uma interface de autenticação única para todos os usuários nos diferentes serviços oferecidos.

## III. PROBLEMAS DE SEGURANÇA EM AMBIENTES DE NUVEM

*Cloud computing* ou computação em nuvem é um modelo de computação distribuída que deriva características da computação em *grids*, no que diz respeito à provisão de informações sobre demanda, para múltiplos usuários concorrentes.

Devido aos evidentes benefícios da computação em nuvem, muitas empresas e instituições estão migrando seus serviços e aplicações para a nuvem. Porém, a segurança e a disponibilidade dos serviços na nuvem ainda é um fator que gera algumas incertezas sobre migrar ou não para nuvem.

Ao pensar em migrar as informações de usuários para um ambiente externo à instituição, deve-se ter certeza de que estas serão armazenadas com segurança e que estejam disponíveis sempre que necessário. Os serviços que são disponibilizados na rede interna, ao serem migrados para a nuvem, devem ser ofertados de forma igual ou superior ao ambiente de rede local, visando reduzir custos com equipamentos e com pessoal técnico.

Antes de efetuar a migração do ambiente interno para a nuvem, uma questão que deve ser discutida, é a segurança da rede interna institucional. Partindo do princípio que os serviços serão acessados a partir da rede interna, se esta não for segura e eficiente, o ambiente de nuvem trará poucos benefícios.

## IV. AMBIENTE PROPOSTO

No modelo proposto neste trabalho, serão centralizadas as credenciais dos usuários, permitindo um gerenciamento integral do ciclo de vida das suas identidades digitais. Espera-se obter um melhor

gerenciamento das credenciais destes, eliminando problemas de utilização dos recursos computacionais, com foco na segurança e na disponibilidade. Com isso, a administração do ciclo do usuário dentro da instituição, desde a sua criação, até o seu encerramento, pode ser facilitada.

Esta proposta engloba desde a reestruturação da rede interna da instituição, até a criação de um ambiente para acesso seguro às informações que serão disponibilizadas por uma nuvem privada. A figura 1 ilustra o cenário que se deseja atingir com a reestruturação da rede interna, aplicando políticas de segurança eficientes. Também demonstra a criação de um meio de acesso seguro ao ambiente externo à instituição (nuvem privada), onde as aplicações e as informações serão hospedadas.

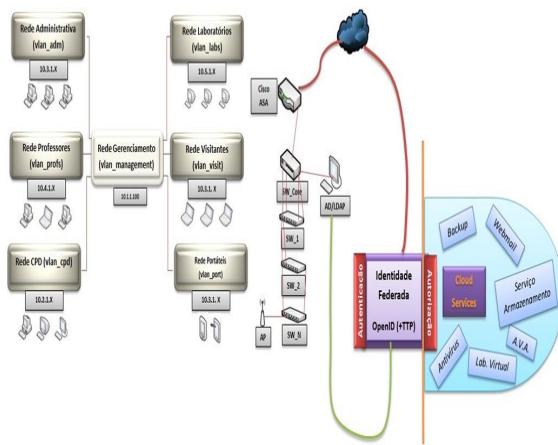


Figura 1 – Cenário de rede esperado com a implantação do modelo proposto (rede interna + ambiente de nuvem).

A seguir, será descrito o modelo de autenticação seguro para a computação em nuvem proposto neste trabalho.

## V. APPLICANDO A SEGURANÇA NO AMBIENTE DA NUVEM

Conforme citado anteriormente, a preocupação com a segurança do ambiente em nuvem é um fator que ainda impede uma grande parcela das instituições de migrar seus dados e aplicações para a nuvem, seja ela privada ou terceirizada.

As informações institucionais exigem um alto nível de sigilo e segurança. Ambientes de nuvem dividem-se em duas categorias principais: nuvem pública e nuvem privada. Em uma nuvem pública, a instituição que pretende utilizar o serviço de *cloud computing*, contrata uma empresa, sendo que, essa empresa fornece toda a infraestrutura de nuvem, incluindo a segurança do ambiente. Já em uma nuvem privada, os recursos computacionais são gerenciados pela instituição, sendo esta a responsável pela manutenção e segurança das informações.

Neste trabalho, serão combinadas tecnologias eficientes para prover o correto gerenciamento de um ambiente de nuvem. Devido ao cenário em questão, aconselha-se a utilização de um ambiente de nuvem privada. Pretende-se criar um ambiente de nuvem gerenciável, com um eficiente nível de autenticação e autorização de usuários. Para isso, pretende-se utilizar os

recursos de *Single Sign On* (SSO), para prover um ambiente de acesso múltiplo, utilizando identidades federadas na nuvem. Serão utilizados os recursos de gerenciamento de identidade e acesso (IAM) e sistemas de gerenciamento de identidade (IMS). Para completar o modelo proposto, serão utilizados ainda, os recursos de *Mutual Protection for Cloud Computing* (MPCC) e *Trusted Third Party* (TTP) respectivamente. Dessa forma, espera-se projetar uma proposta de ambiente de acesso seguro para *cloud computing* utilizando identidades federadas.

#### A. Proposta de Acesso: Autenticação, Autorização e Validação dos Usuários

Uma das principais questões projetadas neste trabalho será a forma de acesso aos recursos providos pela nuvem. A figura 2 ilustra o processo de inclusão de um novo usuário na instituição e todas as ações que esse processo envolve.

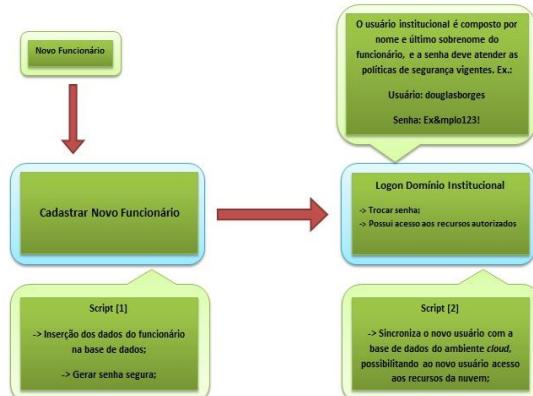


Figura 2 – Processo de inclusão de novo usuário no ambiente institucional e de nuvem.

O processo de criação de usuário local e do ambiente de *cloud* é realizado de forma automatizada, através da comunicação do servidor de autenticação local e do servidor de autenticação da nuvem.

O processo de autenticação no domínio institucional, ou seja, na rede interna, será realizado através da utilização de algum serviço de gerenciamento de usuários e domínio que melhor atenda as necessidades da instituição. Após o primeiro *logon* no domínio local, o usuário será automaticamente replicado na base de dados do ambiente da nuvem, possibilitando o acesso aos serviços e aplicações. Nessa etapa entra o conceito de *Single Sign On* (SSO), que possibilita acesso a diversos serviços e aplicações, utilizando um único conjunto de usuário e senha [3].

Ao projetar-se a utilização da tecnologia SSO, pensou-se nos riscos de segurança, ao fornecer apenas um usuário e senha para várias aplicações. Porém, após uma análise, constatou-se que o nível de aceitação do usuário em relação a várias senhas diferentes, é muito baixo. Sendo assim, devido à necessidade de memorização de várias senhas diferentes, o mesmo pode acabar por realizar anotações, até mesmo em sua mesa, contribuindo para uma possível falha de segurança. Ao implantar um processo de senha unificada, torna-se possível a criação de

uma senha forte, consistente e aceitável seguindo os padrões atuais.

O processo de *login* será concretizado através de um meio de conexão seguro, utilizando esquemas de verificação, validação e liberação de acesso aos sistemas. O esquema de *login* foi projetado utilizando-se os conceitos de Identidades Federadas. Segundo WANGHAM et al. [4], através de uma federação é possível otimizar a troca de informações relacionadas a identidades através das relações de confiança construídas nas federações. Dessa forma, é possível estabelecer acordos entre os provedores de identidades, garantindo que identidades emitidas em um domínio sejam reconhecidas por provedores de serviços de outros domínios, possibilitando assim, uma autenticação única entre os diferentes domínios.

O padrão de gerenciamento de identidades federadas proposto neste trabalho é o *OpenID*. O *OpenID* é um padrão aberto e descentralizado para autenticação de usuários e controle de acesso, permitindo fazer *logon* em muitos serviços com a mesma identidade. É um protocolo simples para acesso único, com suporte a gestão de identidades digitais na web [5].

Além disso, esse modelo é composto pelas tecnologias de gerenciamento de identidades, conhecidas como IAM (*Identity and Access Management*) e IMS (*Identity Management System*). Um *Identity and Access Management* pode ser definido com um método que proporciona um nível adequado de proteção dos recursos e dados de uma organização através de regras e políticas que são impostas aos usuários, tais como senhas, concessão de privilégios e provisionamento de contas de usuário [6]. Já um *Identity Management System*, de forma resumida, refere-se a um sistema de informação ou a um conjunto de tecnologias que podem ser usados para gerenciamento de identidades em empresas ou em aplicações em rede [7].

Para atingir o nível de segurança esperado, propõe-se ainda, juntamente aos servidores de credenciais locais e da nuvem, um método de segurança e criptografia de informações, conhecido como TTP (*Trusted Third Party*). O modelo TTP pode ser caracterizado como uma terceira parte envolvida em uma interação entre outras duas partes garantindo uma relação de confiança entre ambas [8]. Dessa forma, quando um usuário efetua *logon* no domínio local, e está prestes a fazer *logon* no ambiente de nuvem, o TTP atua como um intermediário entre o serviço de domínio local e o serviço de *logon* na nuvem (*OpenID*). Desse modo, os dados são criptografados e as informações continuam seguras. Segundo Leandro [3], *Mutual Protection for Cloud Computing*, é baseado na *Philosophy of Reverse Access Control*, onde os clientes controlam e tentam aplicar meios da nuvem fornecer controle de autenticação e autorização dentro de um ambiente dinâmico e a nuvem garante que o cliente não viole a sua estrutura de segurança.

Durante o processo de *logon*, o usuário, sem perceber, passará por uma série de testes, onde serão verificadas a sua autenticidade e integridade. De forma simples, pode-se resumir o processo acima descrito, analisando a figura 3.



Figura 3 – Modelo de *logon* baseado em três níveis sequenciais: Autenticação, Autorização e Validação.

A cada alteração realizada nas informações dos utilizadores, o serviço de gerenciamento de usuários local enviará uma solicitação de alteração ao servidor de gerenciamento de usuários da nuvem, para que ambos possam iniciar um processo de atualização de suas informações. A cada tentativa de *logon*, o processo de verificação de credenciais dos usuários, entre o ambiente local e da nuvem, é iniciado. A figura 4 demonstra esse processo.



Figura 4 – Modelo de sincronização entre serviço de *logon* local e serviço de *logon* da nuvem.

O processo de *logon* suportará ainda diferentes subdomínios, dentro de um único domínio institucional. A figura 5 ilustra o cenário onde diferentes domínios, com suas redes, poderão ser autenticados e autorizados.

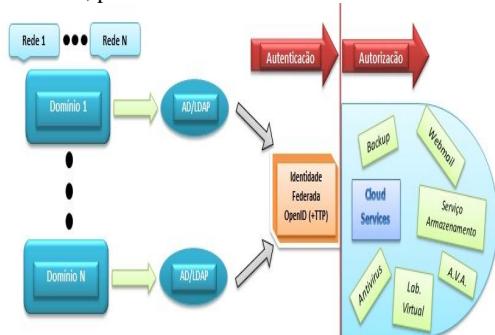


Figura 5 – Vários domínios passando pelo processo de autenticação e autorização do modelo proposto.

## VI. CONTRIBUIÇÕES DO AMBIENTE PROPOSTO

Embora o modelo proposto nesse trabalho não tenha sido implantado, tem-se grande expectativa de bons resultados após sua efetivação. Os benefícios irão englobar desde o setor de tecnologia da informação, até os usuários finais da Instituição. De forma inicial, alguns objetivos tornam-se mais visíveis, sendo descritos a seguir.

### A. Para a Equipe de Tecnologia da Informação

Em algumas universidades, que possuem estrutura multi campus, opta-se pela centralização dos sistemas e serviços, que são oferecidos aos usuários e comunidade em geral. Juntamente com a adoção de uma estrutura centralizada, fatores como o gerenciamento e a disponibilidade das informações são minuciosamente pensados.

Com a adoção de um ambiente de computação em nuvem, os serviços podem ser gerenciados de forma simples e centralizada (visando o aspecto de controle dos serviços) e de forma descentralizada (tendo em vista, que o ambiente institucional local torna-se irrelevante, pois o responsável por determinado serviço, poderá prestar qualquer tipo de suporte, independente de sua localização). Dessa forma, fica claro, que o ambiente de nuvem proposto nesse pode colaborar em vários sentidos para a equipe de tecnologia da informação da UNIPAMPA, bem como, aos demais setores prestadores de serviço à comunidade em geral.

### B. Para a Rede Interna Institucional

Uma das etapas que antecedem a possível implantação do ambiente de nuvem privado aqui proposto envolve uma reestruturação da rede interna da Instituição.

A rede interna deverá passar por uma série de testes e padronizações, onde serão aplicadas normas e métodos, para colaborar com o correto gerenciamento da rede interna da instituição, bem como a adoção de padrões de estruturação de redes de computadores. Dessa forma, problemas de má administração e gerenciamento, tendem a ser solucionados, colaborando para a implantação do projeto de migração dos serviços para um ambiente de nuvem.

### C. Para os Usuários Finais

Uma vez implantado, o modelo proposto proporcionará a comunidade acadêmica a possibilidade de acesso às informações nos mais variados lugares, fazendo com que os usuários não tenham necessidade de estar na instituição para utilizar os serviços oferecidos.

Neste sentido, funcionários, professores e alunos terão uma estrutura de serviço disponível sempre que necessário, possibilitando assim o acesso às informações da instituição, impactando na correta utilização dos serviços oferecidos e colaborando para a integração acadêmica, entre alunos, professores, funcionários e universidade.

## VII. CONCLUSÃO

Esse trabalho apresentou uma proposta de ambiente de acesso seguro para *cloud computing* utilizando conceitos de identidades federadas, *Trusted Third Party* (TTP) e

*OpenID*. Através dessa proposta, acredita-se que caso os serviços sejam migrados para a nuvem privada, o método de autenticação e validação de usuários será eficiente, atingindo o nível de segurança esperado. Utilizando identidades federadas é possível unificar *logins* de acesso para diversos provedores de nuvem (SSO), possibilitando um melhor gerenciamento dos usuários e acesso único a diferentes serviços e aplicações.

Durante o desenvolvimento desta proposta, foram levantadas uma série de premissas e questões fundamentais de segurança envolvendo o ambiente de nuvem e o ambiente da rede local. Através destas informações, tornou-se possível a criação de um modelo de ambiente de nuvem, com um nível de segurança que se enquadra nos padrões atualmente exigidos.

Como sugestão de trabalho futuro, objetiva-se testar o ambiente computacional proposto neste. Espera-se, que com os testes a serem realizados, seja possível verificar a confiabilidade e integridade do modelo apresentado. Juntamente com os testes, espera-se aprimorar o modelo inicial, proporcionando um nível de segurança maior, aumentando a confiabilidade dos ambientes local e de nuvem.

## REFERÊNCIAS

- [1] J. Hu, Q. Sun, H. Chen. Application of Single Sign-on (Sso) in Digital Campus. Broadband Network and Multimedia Technology (IC-BNMT). 3rd IEEE International Conference on, 2010.
- [2] S.K. Bhosale. Architecture of A Single Sign on (Sso) for Internet Banking. Wireless, Mobile and Multimedia Networks. IET International Conference on, 2008.
- [3] M. A. P. Leandro, T. J. Nascimento, D. R. S. Santos, C. M. Westphall, C. B. Westphall. Multi-Tenancy Authorization System with Federated Identity for Cloud-Based Environments Using Shibboleth. ICN 2012 : The Eleventh International Conference on Networks.
- [4] M. S. Wangham, E. R. de Mello, D. da S. Böger, M. Guerios, J. da S. Fraga Gerenciamento de Identidades Federadas. Disponível em: <<http://dainf.ct.utfpr.edu.br/~maziero/lib/exe/fetch.php/ceseg:2010-sbseg-mc1.pdf>>. Acesso em: 14/07/2012.
- [5] Openid. O que é OpenID?. Disponível em: <<http://www.openid.org.pt/2010/04/o-que-e-o-openid/>>. Acesso em: 14/06/2012.
- [6] S. A. Almulla, C. Y. Yeun. Cloud Computing Security Management. IEEE, 2010.
- [7] Researcher's. Federated Identity Management in Cloud Computing. Disponível em: <<http://clean-clouds.com/2012/04/25/federated-identity-management-in-cloud-computing-2/>>. Acesso em: 12/06/2012.
- [8] P. T. Endo, G. E. Gonçalves, J. Kelner, D. Sadok. A Survey on Open-source Cloud Computing Solutions. VIII Workshop em Clouds, Grids e Aplicações, 2010.

# Utilização de técnicas de paralelismo para desenvolvimento de uma ferramenta com alto desempenho para varreduras de dispositivos de rede, escrita em linguagem C utilizando as bibliotecas *Socket* e *OpenMP*

Cristian Cleder Machado  
URI-FW  
cristian@cristian.com.br

**Resumo** — O presente artigo tem como objetivo a implementação e análise de execução de um Port Scan escrito em linguagem C de forma sequencial e outro de forma paralelizada utilizando como comunicação para varredura das portas de destino a biblioteca *Socket*. Os ambientes de simulação variaram entre Arquiteturas de Computadores, Estrutura da Rede e Número de *Threads* executadas em cada teste.

## I. INTRODUÇÃO

Devido ao rápido crescimento na utilização de rede de computadores e o surgimento de várias aplicações voltadas à mesma, percebe-se que a questão segurança tornou-se algo de suma importância.

Ataques para “bisbilhotar” o tráfego na rede se intensificam, visando capturar informações importantes ou muitas vezes, tornar as informações ou suas estruturas de acesso indisponíveis. Cada serviço é disponibilizado utilizando um protocolo – ou seja, um conjunto de regras – que é associado a um número chamado porta que é onde o servidor que provê o serviço aguarda uma conexão.

Para cada serviço existe uma porta padrão. Por exemplo, 21 - FTP (transferência de arquivos), 23 - telnet (terminal virtual remoto), 25 - smtp (envio de e-mails), 80 - http (protocolo www - páginas Internet), 110 - pop3 (recebimento de mensagens). Este padrão foi estabelecido para que empresas pudessem criar suas estruturas e “conversar” com estruturas de terceiros, tornando-se um ponto favorável para toda rede de computadores. Porém, através deste padrão, invasores sabem “teoricamente” o que está sendo executado em determinada porta e podem criar seus ataques de formas bem elaboradas e com buscas muito específicas.

Enfim disso, saber quais as portas que estão abertas ou os serviços que estão sendo executados e que podem sofrer ataques devem ser avaliados. Esse artigo apresenta uma ferramenta para varreduras remotas em dispositivos utilizando a linguagem C[1][2], juntamente com a biblioteca de Sockets[1][2] para realização da comunicação entre a máquina a ser monitorada e a máquina a fazer o monitoramento e à paralelização para otimização do programa, a biblioteca OpenMP[3].

## II. SOBRE O PORT SCAN DESENVOLVIDO

Um Port Scan nada mais é do que um programa que busca as portas abertas em um dispositivo qualquer que tenha uma interface de rede. O Port Scan desenvolvido foi escrito em linguagem C para execução de forma sequencial e também, em forma paralela utilizando a biblioteca OpenMP. O algoritmo foi escrito de forma a varrer o número máximo de portas existentes para a comunicação, ou seja, o valor 65535[2].

Para execução do programa somente foi necessária à passagem do IP ou *hostname* da máquina a ser varrida. No caso da execução com várias *threads*, foi setada a variável de ambiente OMP\_NUM\_THREADS para o valor máximo de *threads* que o programa deveria usar. Um exemplo de execução do programa é apresentado conforme mostra a figura 1.

```
# export OMP_NUM_THREADS=128; ./PortScan x.x.x.x
Escaneando portas abertas no IP\Host x.x.x.x
Porta   Servicos
21     ftp
25     smtp
22     ssh
53     domain
```

Figura 1. Exemplo de execução do Port Scan.

Ao executar o comando conforme ilustra a figura 1, o programa entra num *loop* para varrer as conexões abertas nas portas de 1 a 65535. Como resultado final, a figura 2 mostra a saída do programa com todos os resultados.

```
Escaneando portas abertas no IP\Host x.x.x.x
Porta   Servicos
22     ssh
21     ftp
53     domain
80     www
111    sunrpc
2049   nfs
3306   mysql
10000  webmin
37316  Servico desconhecido.
41089  Servico desconhecido.
42177  Servico desconhecido.
44621  Servico desconhecido.
49599  Servico desconhecido.
49788  Servico desconhecido.

Total de Porta(s) Open 14
Total de Porta(s) Closed 65521
Latencia do host: 0.000113/s
Tempo de execucao do programa: 0.515547/s
```

Figura 2. Amostragem dos resultados após execução.

A figura 2 mostra os resultados após uma execução com sucesso. Durante a execução do programa, para cada ocorrência de situação aberta na porta varrida, esta, é impressa automaticamente na tela. Ao fim da execução, o programa retorna a quantidade de portas abertas e fechadas encontradas, o tempo médio de latência entre todas tentativas de conexão do *Socket*[2] no dispositivo e o tempo de execução do programa. O quadro 1 mostra o trecho de código onde inicia-se o *loop*.

```
/* Trecho que inicia a paralelização utilizando
   a diretiva parallel for e as cláusulas private
   e shared para as variáveis e schedule para o
   tamanho do bloco */
#pragma omp parallel private(dst,
                           sock,
                           portas,
                           InilatTime,
                           FimLatTime,
                           tempoExec,
                           service)
{
    #pragma omp for schedule(dynamic) nowait
    for(portas = 1; portas <= MaxPortas; portas++)
    {
        /* Trecho do executado dentro do laço */
    }
}
```

Quadro 1. Trecho inicial do laço.

O quadro 1 mostra o trecho que inicia a paralelização usando a diretiva *parallel for*[3] e as cláusulas *private*[3] para indicar quais variáveis devem ter uma cópia para cada *thread* e *schedule*[3] que determina o tamanho do bloco de execuções para cada *thread*, juntamente com a cláusula *nowait*[3] que informa que tudo deve ser executado sem aguardar pelo fim de cada execução.

Como o programa busca informações de um dispositivo remoto, este calcula a latência da rede de modo que esta valha como parâmetro para análise dos resultados. O quadro 2 descreve como foi realizado esse cálculo.

```
gettimeofday( &IniLatTime, NULL );
if(connect(sock, (struct sockaddr *)&dst,
           sizeof(struct sockaddr)) == -1)
{
    gettimeofday( &FimLatTime, NULL );
    tempoExec = (float)(FimLatTime.tv_sec
                         - IniLatTime.tv_sec);
    tempoExec += (FimLatTime.tv_usec
                  - IniLatTime.tv_usec)
                 / (float)1000000;
    latencia+=tempoExec;
    close(sock);
}
else
{
    gettimeofday( &FimLatTime, NULL );
    tempoExec = (float)(FimLatTime.tv_sec
                         - IniLatTime.tv_sec);
    tempoExec += (FimLatTime.tv_usec
                  - IniLatTime.tv_usec)
                 / (float)1000000;
    latencia+=tempoExec;
/* ... */
/* trecho ocultado do programa
   por não referenciar o contexto */
/* ... */
printf("Latencia do host: %.6f/s \n",
      (latencia/65535)*100;
```

Quadro 2. Código para cálculo da latência.

No quadro 2, percebe-se que a variável *IniLatTime* recebe o valor inicial de tempo, conhecido através do comando *gettimeofday*[1]. Após, utilizando a biblioteca *Socket*, foi realizado um teste para verificar se o destino estava recebendo conexões naquela porta ou não. Em qualquer um dos casos, há resposta da máquina, e a variável *FimLatTime*, recebe o valor do tempo naquele instante. A latência apresentada no fim da execução do programa foi calculada através da média de todos os tempos de latência para cada teste de conexão de *Socket*.

Para mostrar o serviço que está executando em determinada porta foi utilizada a *struct servente*[2] que

retorna essa informação. O quadro 3 apresenta esse trecho do código.

```
#pragma omp critical
{
    service = getservbyport(htons(portas), NULL);
    if (service>0)
    {
        printf("%d %s\n", ntohs(service->s_port),
               service->s_name);
    }
    else
    {
        printf("%d Serviço desconhecido.\n", portas);
    }
}
```

Quadro 3. Busca de informações da porta.

O quadro 3 apresenta um trecho de código com uma paralelização em uma área crítica, pois a impressão na tela dos valores encontrados retorna fora de ordem, devido a condição de corrida estabelecida pelo *I/O*. Através da função *getservbyport* a variável *service* recebe um *array* de informações do sistema. Caso o retorno for maior do que zero, é impresso na tela a porta e serviço nela estabelecido, caso contrário, é impresso a porta e o texto “Serviço desconhecido”. A maioria das portas não possui um serviço pré-determinado por isso a função não reconhece o que está “rodando”. As portas que geralmente são reconhecidas por qualquer tipo de sistema são as portas até 1024, chamadas “portas padrão”, onde a maioria dos serviços conhecidos estão sendo executados.

Para o cálculo do tempo de execução o método foi semelhante ao da latência conforme explana o quadro 4.

```
main(int argc, char *argv[])
{
    gettimeofday( &IniProgTime, NULL );

    /* CODIGO DO PROGRAMA INTEIRO */

    gettimeofday( &FimProgTime, NULL );
    tempoExec = (float)(FimProgTime.tv_sec
                         - IniProgTime.tv_sec);
    tempoExec += (FimProgTime.tv_usec
                  - IniProgTime.tv_usec)
                 / (float)1000000;
    if(tempoExec>60) {
        printf("Tempo de execução do programa:
               %.2f/m\n\n\n",tempoExec/60);
    } else {
        printf("Tempo de execução do programa:
               %.6f/s\n\n\n",tempoExec);
    }
    return(0);
}
```

Quadro 4. Código do cálculo de execução do programa.

Como mostra o quadro 4, *IniProgTime* e *FimProgTime* são iniciadas com seus valores nos dois extremos do programa para que este seja o mais exato possível.

Como parâmetro para comparativo de desempenho foi utilizado o programa *Nmap*[4] para avaliar se o aplicativo desenvolvido gerava uma concorrência com este e se havia coerência nas informações recolhidas.

### III. ESTRUTURAS, ARQUITETURAS E CONFIGURAÇÕES

Para realização dos testes foram utilizadas diversas estruturas, variando sempre a máquina que realizará a varredura, aqui denominada Monitor, e da máquina que receberá a varredura, denominada Alvo, juntamente com a estrutura de testes.

Nas arquiteturas das máquinas pode-se destacar como pontos importantes para os testes o número de *cores* existentes variando entre 1 e 8 e a velocidade da placa de rede entre 100Mb e 1Gb. É importante ressaltar que as memórias *RAM* e discos rígidos eram diferentes em cada configuração, porém, para o tipo de aplicativo desenvolvido para teste, as mesmas não interferiram nos resultados. Na estrutura da rede, tem-se testes em uma estrutura de fibra ótica, uma de par trançado e uma rede *wireless*. Referente à localização do Alvo, estes tiveram diferenças entre uma conexão ponto-a-ponto, passagem por um roteador wireless e a passagem por vários roteadores na Internet. O link de internet é de 1Mb.

As configurações de cada teste seguem apresentadas nas tabelas 1 e 2. Inicialmente a tabela 1 mostra as configurações relevantes para as análises de cada máquina monitor juntamente à topologia utilizada como conexão para os testes.

Tabela 1. Configuração das máquinas Monitores.

Nome	Conf1	Conf2	Conf3	Conf4
Equipamento	Netbook	Notebook	ML350	ML350
Processadores	1	1	2	2
Cores	2	4	8	8
Clock	1Ghz	2.4 Ghz	2.4Ghz	2.4Ghz
Placa de Rede	100Mb	100Mb	1Gb	1Gb
Topologia	Wireless	Cabo TP	Fibra	Internet
S.O.	Ubuntu 12.04			

A tabela 2 indica as configurações dos dispositivos utilizados como Alvo no experimento.

Tabela 2. Configuração dos Alvos.

Nome	Conf1	Conf2	Conf3	Conf4
Equipamento	Notebook	ML110	ML350	?
Processadores	1	1	2	?
Cores	4	4	8	?
Clock	1Ghz	2.4 Ghz	2.4Ghz	?
Placa de Rede	100Mb	100Mb	1Gb	?
Topologia	Wireless	Cabo TP	Fibra	Internet
S.O.	Ubuntu 12.04			?

As escolhas destas estruturas foram feitas com o intuito de dimensionar a escalabilidade do programa em nível de *hardware* e das interferências que podem ocorrer em cada estrutura de rede.

#### IV. METODOLOGIAS

Os testes foram realizados executando o *Port Scan* 30 vezes em todas as configurações propostas. As variações do *Port Scan* foram feitas com execução do mesmo de forma sequencial, ou seja, sem as diretivas de paralelização; e de forma paralela utilizando 1, 2, 4, 8, 16, 32, 64 e 128 *threads*. Para cada execução foi exportada a variável de ambiente *OMP\_NUM\_THREADS* com o valor do número de *threads* máximo a ser executado no momento. O *Nmap* foi executado mantendo a mesma

quantidade de execuções e para fins de casualidade, as execuções foram alternadas entre a sequência do *Nmap* e o *Port Scan* alternando o número de *threads*.

#### V. RESULTADOS E DISCUSSÕES

Inicialmente foi utilizada a ferramenta *indicator-multiload* do *Ubuntu* para gerar um gráfico de utilização da rede para cada tipo de conexão e acompanhamento para avaliação de como o programa se comporta durante a varredura, analisando, se continuamente, o mesmo fazia solicitações à máquina Alvo. A figura 3 mostra uma tela da ferramenta e o comportamento da rede numa execução.



Figura 3. Histórico dos dados recebidos e enviados na rede.

Ao visualizar a figura 3, tem-se um gráfico com os envios e recebimentos dos pacotes durante a execução do *Port Scan*. Percebe-se que o envio e recebimento formam uma simetria em vários momentos, indicando que as solicitações e as respostas são constantes durante a execução do programa.

Um fator de suma importância que será ressaltado é sobre o tempo de execução realizado nos testes.

Referente ao *Nmap* o tempo de execução é apresentado conforme gráfico 1.

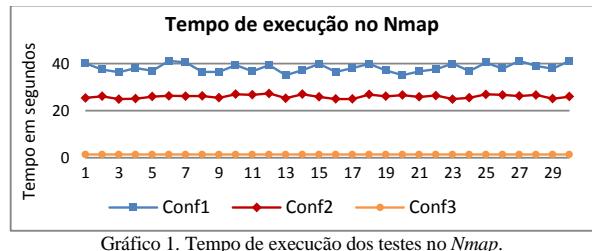


Gráfico 1. Tempo de execução dos testes no *Nmap*.

O gráfico 1 apresenta as informações de tempo de execução em cada configuração exceto a configuração 4, devido ao valor de execução ser discrepante comparado as demais medidas pelo fato da mudança de estrutura, no caso, o acesso a internet. Os valores da Conf4 serão apresentados num gráfico posterior, para fins de melhor entendimento. Quanto à linha de execução da Conf1, pode-se observar que esta, teve uma variável no tempo de execução onde, uma das causas é a estrutura, que não garante a qualidade necessária para envio e recebimento dos pacotes enviados na rede, pois uma rede *wireless* está sujeita a questões de imprevisibilidade do ambiente num todo. Outra questão é a baixa qualidade do *hardware*, que pode ter um poder de processamento inferior à demanda de requisições.

Os gráficos que seguem na sequência apresentam informações variando o programa de forma sequencial e de 1 até 128 *threads*. O gráfico 2 demonstra a média de execução na Conf1.

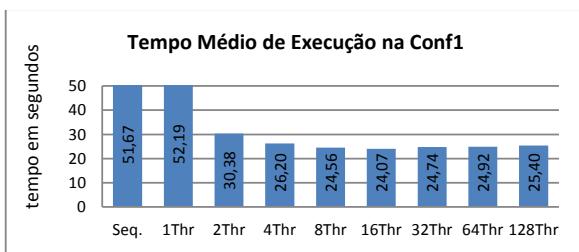


Gráfico 2. Tempo médio de execução da Conf1.

Analizando o gráfico 2, nota-se um *speedup* a partir da utilização de mais de uma *thread*. Na execução dos testes em mais de 16 *threads* percebe-se que não se obtém uma progressão no *speedup*, devido a estrutura da Conf1.

O gráfico 3 apresenta as informações dos testes realizados na Conf2.

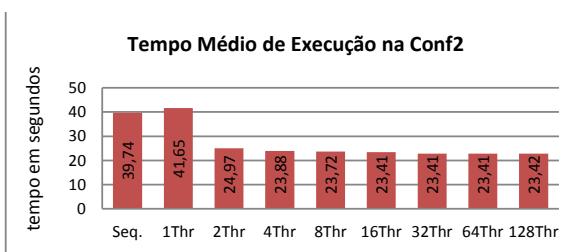


Gráfico 3. Tempo médio de execução da Conf2.

No gráfico 3, é apresentado um *speedup* no aumento do número de *threads* e após o uso de 16 *threads* é mantido o tempo de execução. Assim como na Conf1, a estrutura limita o desempenho.

No gráfico a seguir, temos os testes na Conf3.

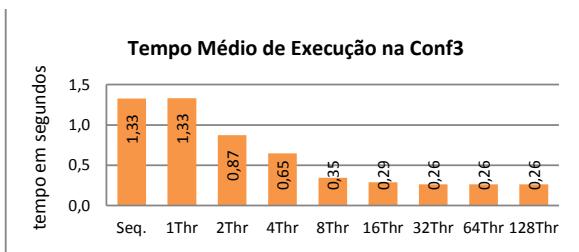


Gráfico 4. Tempo médio de execução da Conf3.

No gráfico 4, com a melhoria da estrutura, é apresentado um *speedup* no aumento do número de *threads*. Diferente das outras configurações, o gráfico se mantém após o uso de 32 *threads*, e posteriormente é mantido o tempo de execução.

Os testes realizados com a Conf4 são analisados no gráfico 5.

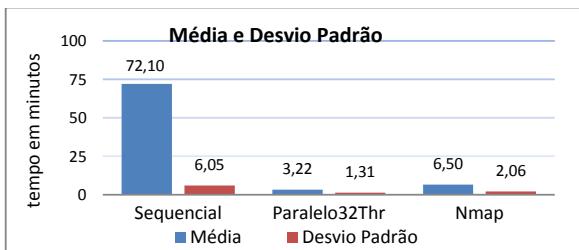


Gráfico 5. Testes acessando domínio na internet.

No gráfico 5 tem-se a comparação entre o *Port Scan* sequencial, o melhor teste de paralelização do mesmo, onde foram utilizadas 32 *threads*, e o *Nmap*. Inicialmente, pode-se notar que o resultado apresenta uma grande diferença na média da execução do *Port Scan* de forma sequencial para a forma paralela, onde a melhor forma paralela chegou a ser 22 vezes mais rápida na média geral.

O programa paralelo também teve melhores resultados quanto à comparação com o *Nmap*, sendo em média 2 vezes mais rápido. É válido salientar os tempos mínimos e máximos de execução de cada teste, onde o programa sequencial teve como tempo mínimo 61,07m e máximo de 83,23m, o paralelo 2,41m e 5,12m e o *Nmap* 4,23m e 9,12m.

É adequado observar que em todas as execuções contra os Alvos, o *Port Scan* identificou a mesma quantidade e as mesmas portas encontradas em aberto, tanto nos *Conf(n)*, quanto nas variações de número de *threads*, ou às comparações com o *Nmap*.

## VI. CONCLUSÕES

Com esta implementação conclui-se que ao empregar paralelização utilizando *OpenMP* para o *Port Scan* desenvolvido, o mesmo obteve um amplo *speedup* devido ao aproveitamento das quantidades de *cores* em cada Monitor testado, juntamente com o aumento do mesmo quando alternadas as quantidades de *threads* por execução.

A quantidade de testes e a diversidades dos ambientes e configurações demonstraram bons resultados no comportamento do mesmo quanto à sua escalabilidade. As comparações com o programa *Nmap* apresentaram que o mesmo foi em média duas vezes superior em tempo de execução. Não é possível afirmar o porquê desta melhoria nesta comparação, pois as variáveis a serem discutidas são muitas, como por exemplo, é desconhecido o número de testes que o *Nmap* faz para cada conexão em determinada porta para definir seu *status*. Talvez, o mesmo faça um maior número, o que pode influenciar gravemente no tempo de execução.

Percebeu-se nos testes da Conf4 que a falta de conhecimento da estrutura Alvo – tais como, link, processamento, demanda/número de conexões, dentre outros – influenciaram muito no tempo de execução de todos os testes.

Como trabalho futuro incentiva-se maior estudo da biblioteca *Sockets* e a biblioteca *OpenMP*, pois, possivelmente, melhores soluções podem surgir juntamente com a busca de mais informações relevantes a implementação e monitoramento para segurança de dispositivos, computadores e servidores numa rede.

## REFERÊNCIAS

- [1] W. R. Stevens, "Unix Network Programming," Interprocess Communications, vol. 2, Prentice Hall, 1998.
- [2] W. R. Stevens. B. Fenner. A. M. Rudoff, "Programação de Rede UNIX," API para Soquetes de Rede, vol. 1, Artmed, 2005.
- [3] OpenMP. (2012, 2 April). Site oficial da API [Online]. Available: <http://openmp.org>.
- [4] Nmap. (2012, 7 January). Site oficial do Aplicativo [Online]. Available: <http://nmap.org/book/man.html>.

---

**VII**

## **Fórum de Pós-Graduação III**

---



# Avaliação de Desempenho em Canal de Retorno de TV Digital baseado em Redes Mesh IEEE 802.11

Samuel C. M. Neves, Edson M. da Silva  
PPGEE – UFPA  
[{samuel.neves, edson.marques}@ifpa.edu.br](mailto:{samuel.neves, edson.marques}@ifpa.edu.br)

Dr. Raimundo Viégas Jr  
PPGCC – UFPA  
[rviegas@ufpa.br](mailto:rviegas@ufpa.br)

**Resumo**— O Sistema Brasileiro de TV Digital faz uso de um canal de retorno para o envio de informações do usuário para a emissora de TV. A escolha desse canal de retorno é muito importante para a implementação de soluções interativas funcionais, devendo adequar-se à região e aproveitar-se da infraestrutura favorável a determinada tecnologia de acesso. Nesse contexto, as redes *Wi-Fi* destacam-se pelo seu baixo custo e fácil escalabilidade. Por isso neste trabalho é feita uma análise do uso da tecnologia IEEE 802.11 em topologia *mesh* como infraestrutura para canal de retorno levando-se em consideração os protocolos de roteamento utilizados e seu desempenho quanto à qualidade de serviço.

## I. INTRODUÇÃO

As facilidades proporcionadas pela informática, principalmente após o advento da Internet, ajudaram a popularizar o acesso à informação. No entanto, uma parcela significativa da população ainda encontra-se à margem desse processo devido à dificuldades de acesso aos meios de comunicação digital. Verifica-se também que uma quantidade expressiva de domicílios brasileiros possui aparelho televisor [9]. Com base nesses indicadores, o Projeto do Sistema Brasileiro de Televisão Digital (SBTVD) vem buscando a democratização da informação por meio do acesso à tecnologia digital através da TV [3].

Nesse contexto, é fundamental o provimento de interatividade entre os usuários do SBTVD e um sistema que permita o fornecimento de serviços e aplicações, ou seja, um sistema em que cada usuário possa interagir individualmente, encaminhando ou recebendo informações e solicitações às emissoras. Assim, surge o conceito de Canal de Interatividade, que é a denominação para o sistema constituído pela conexão entre as redes de telecomunicações da televisão e da Internet [2].

A interatividade pode ocorrer em diferentes níveis. Na interatividade local, o conteúdo é transmitido para usuário final de uma só vez. Nesse ponto o usuário pode interagir livremente com o conteúdo que fica armazenado em seu receptor. Outro nível de interatividade é a interatividade plena, onde o telespectador poderá emitir sua opinião através, por exemplo, de enquetes e votações. Outra possibilidade é o comércio através da TV, bem como aplicações bancárias. O canal de interatividade pode ser dedicado ou não, sendo que o mesmo deve ser definido possibilitando que as informações cheguem com segurança até às emissoras [1].

Esse trabalho propõe o estabelecimento de uma canal de interatividade via redes *mesh* sem fio, baseado no padrão 802.11, como alternativa às regiões onde não há estrutura cabeadas. Para tanto, é utilizado o simulador de redes *Network Simulator* (NS-2) [11], para avaliar os impactos da utilização de protocolos de roteamento na qualidade de

serviço do tráfego, de modo a medir o desempenho desta tecnologia, através do uso de modelos de geração de tráfego que consideram aplicações que enviam dados, voz e vídeo, simulando um nível de interatividade para o usuário da rede *mesh* sem fio como solução de canal de retorno de TV digital.

O trabalho está organizado como descrito a seguir. Na seção II é apresentada a fundamentação teórica necessária para o entendimento do problema contextualizado, onde serão abordados conceitos do canal de interatividade, redes *mesh* e protocolos de roteamento para redes *mesh*. Na seção III é feita a descrição das simulações realizadas neste trabalho, bem como apresentado o cenário onde tais simulações foram realizadas. Na seção IV são apresentados os resultados obtidos a partir dos dados gerados pelas simulações. Na seção V são feitas as considerações finais, bem como direcionamentos a trabalhos futuros.

## II. FUNDAMENTAÇÃO TEÓRICA

### A. O Canal de Interatividade

Conforme ilustrado na Figura 1, a comunicação das emissoras de TV para os usuários é estabelecida pelo chamado canal de descida, constituído pelos canais de radiodifusão, ocorrendo na maioria das vezes por meio de uma comunicação em *broadcast* aberta e disponível a todos os usuários. Ainda conforme a Figura 1, para que o usuário consiga fazer o envio de informações às emissoras, é necessário um canal de retorno, também chamado de canal de interatividade, composto por alguma tecnologia de redes de acesso que permita a comunicação entre usuários do SBTVD e as emissoras ou outro provedor de informações e serviços [7]. O equipamento responsável pela recepção do sinal da emissora pelo canal de difusão e pela interatividade do usuário com a emissora é designado como *set-top box*. Nele as funções de nó de rede devem ser implementadas. Daí a importância da correta modelagem do canal de retorno, pois dependendo das aplicações a serem utilizadas no processo de interatividade, diferentes requisitos devem ser levados em consideração na implementação desse canal, já que as aplicações de rede possuem requisitos variáveis de banda passante, latência e confiabilidade [6].

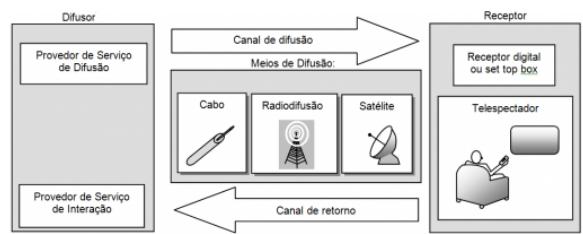


Figura 1. Esquema do canal de retorno [5].

### B. Redes Mesh.

Dentre as diversas tecnologias de rede de acesso que podem ser utilizadas no canal de retorno [7], as redes *mesh*, ou *Wireless Mesh Networks* (WMN), tem ganhado importância, pois disponibilizam uma tecnologia que proporciona longa duração de conectividade de rede em qualquer lugar, a qualquer momento, com simplicidade e baixo custo [1]. Desta maneira, desempenha um papel importante dentro da próxima geração da Internet. As WMN são um caso especial de redes *ad hoc*, onde os nós tem posições fixas e se comunicam com a Internet através de um ou mais *gateways* [2]. A Figura 2 mostra a estrutura da rede *mesh* sem fio utilizada neste trabalho para provimento do canal de retorno, onde cada nó opera não só como um *host*, mas também como um roteador, encaminhando os pacotes em nome de outros nós que podem não estar dentro do alcance direto de transmissão sem fio de seus destinos. É mostrado, ainda, o *gateway*, que tem a função de receber os dados de todos os nós da rede *mesh* e encaminhá-los através da rede pública de telecomunicações.

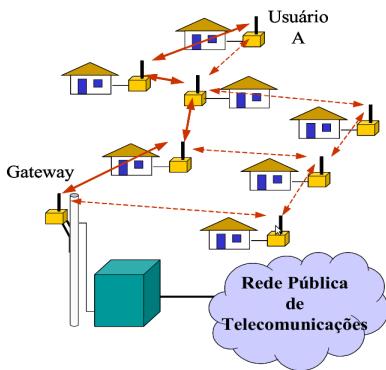


Figura 2. Exemplo de Rede Mesh sem fio [2].

### C. Protocolos de Roteamento em Redes sem fio.

Para prover uma comunicação dentro de uma rede *mesh*, onde os nós constituintes têm características simultâneas de *host* e roteador, o uso de um protocolo de roteamento é requisito indispensável. Esses protocolos são classificados em pró-ativos e reativos.

Nos protocolos pró-ativos, a atualização das informações de roteamento para todos os nós é mantida em cada nó. As principais desvantagens de tais algoritmos é a quantidade de dados para manutenção e a reação lenta em reestruturação e em falhas [2], [6]. Exemplos desse tipo de protocolo são: DSDV (*Destination-sequenced Distance Vector Routing Protocol*), OLSR (*Optimized Link State Routing Protocol*), HSR (*Hierarchical State Routing Protocol*).

Nos protocolos reativos, somente quando uma fonte quer enviar para um destino é que os mecanismos de descoberta de rota são invocados para encontrar a melhor rota. As principais desvantagens de tais algoritmos são o tempo de latência alto na busca de rotas e inundações em excesso, que podem levar ao congestionamento da rede [2], [6]. Os protocolos AODV (*Adhoc On-Demand Distance Vector*), DSR (*Dynamic Source Routing*) e BSR (*Backup Source Routing*) são exemplos desse tipo de roteamento.

Em WMN, onde a mobilidade dos nós é muito pequena ou inexistente, os protocolos reativos tem uma escalabilidade

melhor que os protocolos pró-ativos, sendo esse um dos motivos do uso dos protocolos reativos DSR e DSDV nas simulações realizadas neste trabalho. Considerou-se ainda, um protocolo pró-ativo, o AODV, para comparação e validação dos resultados esperados. Outros fatores para a escolha dos protocolos foram a grande quantidade de literatura existente, que avalia o desempenho desses protocolos, possibilitando a comparação com os resultados obtidos [10], [14], [16], bem como as implementações dos mesmos em versões embarcadas de sistemas operacionais de rede, como os dos projetos *OpenMesh* [12] e *OpenWRT* [13], que favorecem a implementação de *testbeds* e a implantação de redes *mesh* a um baixo custo.

1) *DSDV*: tem como base o algoritmo de roteamento de Bellman-Ford. Os principais compromissos deste protocolo são manter a simplicidade de Bellman-Ford e evitar o problema de *looping*. Cada estação móvel mantém uma tabela de roteamento que lista todos os destinos disponíveis, o número de saltos para alcançá-los e o número de ordem atribuído ao nó de destino. Um sequenciamento é usado para distinguir rotas obsoletas de novas e, desse modo, evitar a formação de *loops*. Periodicamente as estações transmitem suas tabelas de roteamento para seus vizinhos imediatos. A estação também transmite a sua tabela de roteamento se uma mudança significativa ocorrer a partir da última atualização enviada. Assim, a atualização é tanto orientada ao tempo quanto a eventos [15], [16].

2) *AODV*: é um protocolo de roteamento reativo capaz de realizar roteamento *unicast* e *multicast*. Utiliza um algoritmo sob demanda, o que significa que ele constrói rotas entre os nós apenas quando desejado pelos nós de origem. Ele mantém essas rotas, desde que sejam necessárias para as fontes. O protocolo AODV usa números de sequência para garantir a atualização das rotas, é *loop-free*, tem auto-partida e é escalável para um grande número de nós móveis. O AODV descobre rotas ativas através da construção de uma base por meio de um processo que se inicia no nó de origem de dados com o envio de mensagens de requisição de rota em *broadcast* para então retornar uma resposta de rota. São utilizadas mensagens *hello* para notificação de falhas em nós adjacentes. O AODV usa tabelas de roteamento tradicionais, com uma entrada por destino. Desta forma, o protocolo AODV depende das entradas presentes na sua tabela para propagar uma resposta de rota à origem, e, posteriormente, para encaminhar pacotes de dados para o destino [15], [16].

3) *DSR*: é um protocolo de roteamento simples e eficiente, que permite os nós descobrirem dinamicamente uma rota a partir de determinada origem através de múltiplos saltos para qualquer destino através de anúncios na rede. O DSR torna a rede totalmente auto-organizável e auto-configurável, sem necessidade de qualquer infraestrutura ou administração de rede existente. O protocolo é composto de dois mecanismos: descoberta de rota e manutenção de rota, que trabalham juntos para permitir aos nós descobrir e manter rotas para destinos a partir de uma fonte arbitrária na rede *ad hoc*. O uso de roteamento baseado na origem permite que o encaminhamento dos pacotes seja *loop-free*, evitando a necessidade de atualizar informações de roteamento nos nós intermediários através dos pacotes que são encaminhados, uma vez que todos os aspectos do protocolo operam inteiramente sob demanda, permitindo que o esforço de roteamento de pacotes do DSR utilize, automaticamente, apenas o que precisa para reagir a mudanças nas rotas atualmente em uso [15], [16].

### III. DESCRIÇÃO DA SIMULAÇÃO E CENÁRIOS

Utilizou-se simulação para o desenvolvimento do trabalho, pois é uma solução de baixo custo que possibilita a flexibilidade necessária para os testes. Utilizou-se a ferramenta NS-2 para a realização das simulações.

No cenário proposto os dados gerados pelos *set-top boxes* são encapsulados e transmitidos em pacotes via rede sem fio até um nó receptor. Este nó envia esses dados, através da Internet, para a emissora de TV Digital para que uma resposta acerca da situação da qualidade do serviço em uma determinada região seja gerada.

Para isso foi gerado um cenário com 20 *set-top boxes* distribuídos aleatoriamente numa área de 500 x 500 metros. O objetivo da distribuição aleatória dos nós é tornar o cenário mais próximo de uma situação real, em que os domicílios não estejam dispostos regularmente. Os nós são considerados fixos devido ao fato de os *set-top boxes* estarem nas casas dos usuários sem nenhuma mobilidade, o que também garante que não haja problemas de limitação por consumo de energia. Os parâmetros utilizados na simulação são mostrados na Tabela I.

Tabela I  
PARÂMETROS DA SIMULAÇÃO

Parâmetros	Valores
Protocolos de Roteamento	DSR, DSDV, AODV
Camada MAC	802.11
Área de Simulação	500 x 500 m
Nº de nós	20
Tipo de Tráfego	TCP e CBR
Nº Máximo de Saltos	3
Raio de cobertura dos nós	250 m
Tempo de Simulação	50 s
Modelo de Propagação	Two-ray ground reflection

O tratamento dos dados gerados pelo NS-2 foi feito utilizando-se o Awk e o Matlab® [8] para a geração dos gráficos e análise dos resultados.

As métricas utilizadas para a avaliação de desempenho nesta simulação foram: perda de pacotes, vazão média, *overhead* de roteamento e atraso médio fim-a-fim.

### IV. RESULTADOS

Os resultados foram gerados submetendo-se a rede *mesh* sem fio a diversos tráfegos concorrentes que foram incrementados a cada nova simulação até o máximo de 15 fluxos TCP com o intuito de avaliar a qualidade de serviço sob os diferentes protocolos de roteamento testados. Após isso, foram incluídos 2 e depois 4 fluxos CBR respectivamente, objetivando-se avaliar a influência dos fluxos CBR. Não houve necessidade de incluir muitos fluxos CBR concorrentes na simulação, pois o tipo de interatividade utilizada não prevê uma transmissão contínua de informação em UDP, no entanto, percebeu-se que o sistema não é levado ao limite utilizando-se apenas fluxos TCP devido às características intrínsecas deste protocolo de transporte como o controle de fluxo e o de congestionamento [6].

Na Figura 3, é apresentado o gráfico da vazão em relação ao número de tráfegos concorrentes utilizando os protocolos DSR, DSDV e AODV.

Percebeu-se que devido ao fato do TCP implementar um mecanismo de controle de congestionamento, sempre que o volume de carga dos tráfegos CBR é aumentado, atinge-se um ponto de saturação onde a vazão do tráfego TCP diminui bruscamente. Notou-se ainda, que, quando apenas um nó está transmitindo, a quantidade de dados enviada por ele é maior. À medida que outros nós iniciam a transmissão, a quantidade

de dados transmitida por cada nó passa a ser menor. Isso também é devido ao comportamento justo do protocolo TCP [6]. Verifica-se um melhor desempenho do protocolo DSR, principalmente quando o número de fluxos é aumentado, em detrimento do protocolo AODV que obteve um baixo desempenho na vazão.

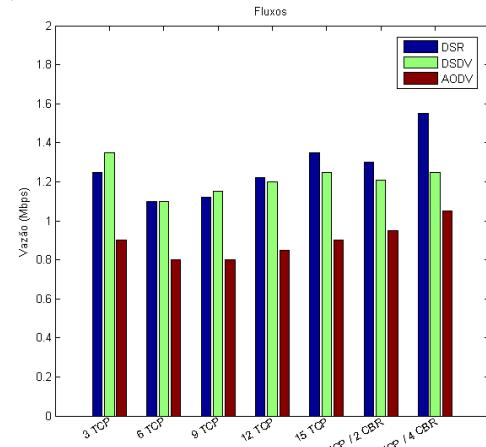


Figura 3. Vazão média da rede

Na figura 4, é mostrado o gráfico correspondente à perda de pacotes, utilizando-se os três protocolos avaliados, em função do número de fluxos concorrentes.

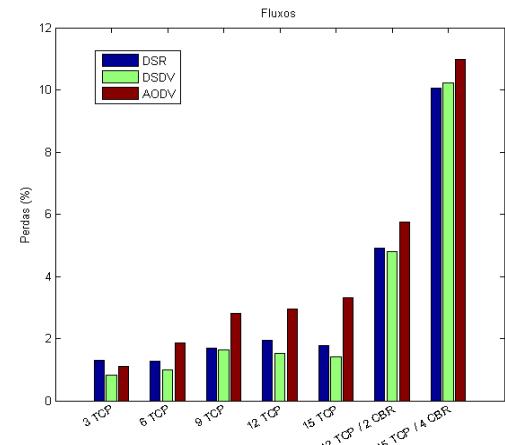


Figura 4. Perda de pacotes na rede

Observou-se, na Figura 4, que a perda de pacotes é relativamente pequena quando se utiliza apenas fluxos TCP, e cresce ao adicionarem-se os fluxos CBR, devido à fila do *buffer* dos roteadores ficar cheia, e grande parte dos pacotes CBR serem descartados. Como o fluxo de dados CBR é contínuo e o protocolo de transporte não é orientado a conexão, esses pacotes são perdidos definitivamente. Notou-se uma pequena vantagem dos protocolos DSR e DSDV sobre o AODV, pois estes apresentaram menores perdas.

Na Figura 5, é mostrado o *overhead* causado pelo envio de pacotes de controle utilizados nos protocolos de roteamento, em função do número de tráfegos concorrentes.

Os índices de *overhead* apresentados foram baixos durante as simulações, fato que se justifica pela ocorrência de poucas quebras de enlace, resultando em menos tráfego de pacotes de controle pela rede, além da falta de mobilidade dos nós, o que faria com que eles tivessem que se reorganizar à medida que se movimentassem.

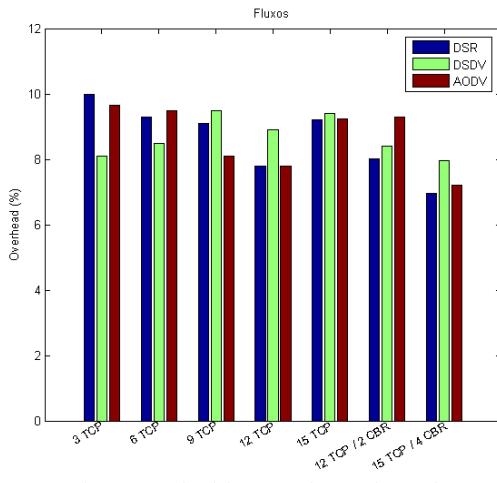


Figura 5. Overhead de pacotes de controle na rede

Notou-se ainda na Figura 5, que quando são incluídos os fluxos CBR, o *overhead* diminui, isso acontece porque o protocolo UDP minimiza a utilização de pacotes de controle em relação ao TCP. O protocolo DSDV manteve a média de *overhead* praticamente estável, isso pelo fato de ser um protocolo pró-ativo.

Na Figura 6 é mostrado o atraso médio fim-a-fim, sofrido pelos pacotes de dados durante a simulação.

Foi observado que os protocolos DSR e AODV sofrem um atraso maior, já que os mesmos são reativos, ou seja, a qualquer tentativa de se atingir um destino, sinalizações devem ser enviadas para determinação da rota, já o DSDV, pró-ativo, mantém uma cópia das tabelas de rotas e consegue quase que instantaneamente atingir o destino requerido, tendo por isso o menor atraso.

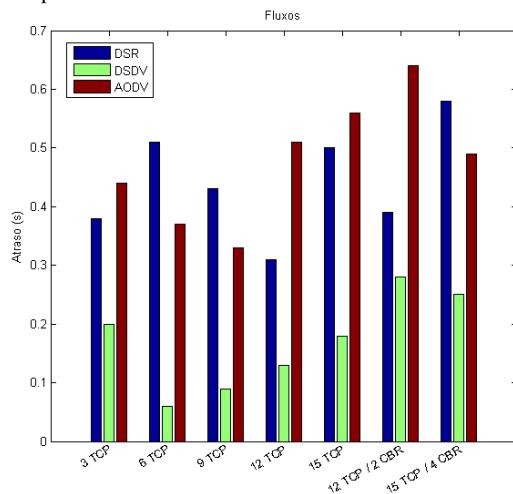


Figura 6: Atraso médio fim-a-fim.

## V. CONCLUSÕES E DIREÇÕES FUTURAS

Os resultados obtidos nas simulações mostraram que os protocolos de roteamento tiveram um desempenho bastante próximo entre eles. Entretanto, o protocolo AODV teve um desempenho inferior aos demais protocolos por apresentar maior perda de pacotes, menor vazão e pelo fato de produzir algum *overhead*, já que é um protocolo reativo. Conforme ilustraram os resultados, os protocolos DSDV e DSR apresentaram um desempenho similar, já que o protocolo

DSR apresenta uma perda praticamente igual ao DSDV, mas possibilita maior vazão em quase todas as situações testadas.

Então, depreende-se que a implementação de canal de retorno utilizando redes *mesh* IEEE 802.11 com os protocolos de roteamento DSR, DSDV e AODV é adequada para a utilização em conjunto com um protocolo de transporte orientado à conexão, tornando o seu uso apropriado para aplicações cujo nível interatividade requeira, por exemplo, preenchimento de formulários, transferência de arquivos, mensagens de texto ou visualização de páginas, aplicações estas que necessitam de confiabilidade em nível de transporte [6].

Em um próximo trabalho será avaliado o desempenho, e dessa forma a viabilidade, em nível de acesso do protocolo 802.11s [14], o padrão das redes *mesh*, que usa como protocolo de roteamento o HWMP (*Hibrid Wireless Mesh Protocol*). Pode-se também considerar as perdas devido a obstáculos, utilizando-se outros modelos de propagação existentes, bem como realizar a inserção de novos modelos, adequados a uma determinada região, na suíte de protocolos do NS-2. Pode-se ainda realizar *testbeds* para comparação com os dados simulados.

## VI. REFERÊNCIAS

- [1] A.V.F. Ribeiro, “Uso de Redes Mesh como solução para canal de retorno da TV Digital Interativa”, Dissertação Mestrado, UFF, 2007.
- [2] Centro de Pesquisas e Desenvolvimento em Telecomunicações. “Canal de Retorno com Redes AdHoc e Tecnologia 802.11b”. Versão PD.30.12.34A.0001A/RT-03-AA. Disponível em: <http://sbtvd.cpqd.com.br/>.
- [3] Decreto Presidencial Número 4.901, 26/11/2003. Disponível em [http://www.planalto.gov.br/ccivil\\_03/decreto/2003/d4901.htm](http://www.planalto.gov.br/ccivil_03/decreto/2003/d4901.htm). Acesso em Mar. 2012.
- [4] G.R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berleemann, B. Walke, “IEEE 802.11s: The WLAN Mesh Standard”, Vol. 17, No. 1, pp. 104-111, Feb. 2010.
- [5] iMasters. Disponível em: <http://www.imasters.com.br/>
- [6] J. F. KUROSE, K. W. ROSS, Redes de Computadores e a Internet: Uma abordagem Top-down. 5<sup>a</sup> ed. Pearson, 2010.
- [7] M. A. R. Manhães, P. J. Shieh, A.C. Lamas, “Canal de Interatividade em TV Digital”, Cad. CPqD Tecnologia, Vol. 1, No. 1, pp. 29-36, Jan/Dez. 2005.
- [8] Matlab, Disponível em: <http://www.mathworks.com>, Acesso em Jun. 2012.
- [9] Mídia Dados Brasil 2012. São Paulo, 2012. Disponível em: <http://midiadados.digitalpages.com.br/home.aspx>. Acesso em: Jun. 2012. pp. 328.
- [10] N. S. M. Usop, A. Abdullah, A. F. A. Abidin, “Performance Evaluation of AODV, DSV & DSR Routing Protocol in Grid Environment”. International Journal of Computer Science and Network Security, Vol. 9, No. 7, pp. 261-268, Jul. 2009.
- [11] NS-2. Disponível em <http://www.isi.edu/nsnam/ns/tutorial/>. Acesso em Abr. 2012.
- [12] OpenMesh, Disponível em: <http://www.open-mesh.com>. Acesso em Abr. 2012.
- [13] OpenWRT, Disponível em: <https://openwrt.org>. Acessado em: Abr. 2012.
- [14] S. Islam, N. Hider, T. Haque, L. Miah, “An Extensive Comparison among DSDV, DSR and AODV Protocols em MANET”. International Journal of Computer Applications, Vol. 15, No. 2, pp. 22-24, Feb. 2011.
- [15] S. S. N. Rao, Y.K.S. Krishna, K. N. Rao, et al. “A Survey: Routing Protocols for Wireless Mesh Networks”. International Journal of Research and Reviews in Wireless Sensor Networks, Vol. 1, No. 3, pp. 43-47, Sep. 2011.
- [16] S. Upadhyay, P. Joshi, N. Gandotra, A. Kumari. “Comparison and Performance analysis os Reactive type DSR, AODV and proactive type DSDV routing protocol for Wireless Mobile Ad-hoc network, using NS-2 Simulator”. Journal of Engineering and Computer Innovations . Vol. 2, No. 10, pp. 36-47. Mar. 2012.

# Multiflow: *Multicast clean-slate* com cálculo antecipado das rotas em redes programáveis *OpenFlow*

Lucas Bondan, Lucas F. Müller, Maicon Kist  
Universidade Federal do Rio Grande do Sul  
{lbondan, lfmuller, mkist}@inf.ufrgs.br

**Resumo**—Cada vez mais populares, aplicações de transmissão de conteúdo multimídia através da *Internet* requerem comunicação *multicast*, a fim de diminuir a taxa de dados trafegando na rede. No entanto, há uma dificuldade da ampla adoção dos protocolos tradicionais de *multicast*, onde a responsabilidade de gerência dos grupos *multicast* é distribuída entre os equipamentos da rede. Pelo fato de utilizarem algoritmos distribuídos, tais protocolos geram atrasos no processamento de eventos de controle dos grupos. Este trabalho propõe uma abordagem *clean-slate* para *multicast*, onde é realizado o cálculo da melhor rota do cliente até a fonte, reduzindo atrasos nos eventos de grupos. O protótipo desenvolvido implementa esta abordagem utilizando a tecnologia *OpenFlow*. Os resultados obtidos nos experimentos mostram um ganho de desempenho em relação aos requisitos de aplicações com comunicação *multicast*.

## I. INTRODUÇÃO

Cada vez mais populares, aplicações de transmissão de conteúdo multimídia através da *Internet* requerem comunicação entre vários *hosts*. Em aplicações como IPTV (*Internet Protocol TV*), o provedor de conteúdo transmite dados, muitas vezes idênticos, para inúmeros assinantes do serviço. Estas aplicações podem utilizar o IP *multicast* para realizar comunicações multiponto, evitando o desperdício de banda ao enviarem dados repetidos através de várias conexões *unicast*.

Considerando a característica de distribuição de responsabilidades da *Internet* atual, tem-se que cada roteador executa parte do algoritmo de roteamento. Desta maneira, protocolos de roteamento *multicast* como *Distance Vector Multicast Routing Protocol* (DVMRP) ou *Multicast Open Shortest Path First* (MOSPF) não são eficientes para realizar mudanças na árvore *multicast*, pois é preciso esperar que os roteadores troquem informações entre si e atualizem suas tabelas de roteamento, o que pode ser um processo demorado. Além disso, o *Internet Group Management Protocol* (IGMP), responsável por controlar a entrada e saída de *hosts* do grupo, precisa enviar mensagens a vários roteadores para notificar o acontecimento de eventos relacionados aos grupos *multicast* (e.g.: entrada e saída de clientes) [1].

Diante deste cenário, uma abordagem logicamente centralizada para realizar o roteamento *multicast* pode trazer diversas vantagens sobre a abordagem distribuída. Entre elas, a possibilidade de criar uma árvore de distribuição ótima para cada ocasião, devido à visão completa da topologia que um algoritmo centralizado possui. Também seria possível processar eventos de controle de grupo mais rapidamente, sem criar inundações de mensagens como ocorre na abordagem distribuída.

Neste trabalho propõe-se uma abordagem de *multicast clean-slate* em que é realizado o cálculo antecipado da rota entre a fonte e o cliente do grupo, com a finalidade de acelerar o processamento de eventos *multicast*. A partir disto, foi implementado o protótipo, utilizando o conceito de *Software Define Networks* (SDN) com uso da tecnologia *OpenFlow* [2]

e realizados experimentos em topologias emuladas no *software Mininet* [3].

O restante deste trabalho está organizado como segue: Na Seção II são expostos os conceitos que englobam a proposta de SDN bem também é apresentada a tecnologia *OpenFlow*. Na Seção III são apresentadas as propostas encontradas na literatura que abordam a gerência de tráfego utilizando protocolos *multicast*. A Seção IV é apresentado o protótipo desenvolvido, descrevendo sua arquitetura. Na Seção V é descrita a metodologia de avaliação do Multiflow. A Seção VI apresenta os resultados obtidos a partir dos experimentos realizados. Por fim, a Seção VII apresenta considerações finais e perspectivas de trabalhos futuros.

## II. Software Defined Networks

Com o grande crescimento do número de equipamentos conectados em rede, cada vez mais aplicações tem sido desenvolvidas para melhorar a comunicação entre eles. Este crescimento traz consigo não somente a necessidade de maior poder computacional nos equipamentos, mas também um aumento na complexidade de desenvolvimento de aplicações que proporcionem melhor desempenho à rede.

O conceito de SDN propõe separar o plano de dados do plano de controle, quebrando completamente os paradigmas atuais no contexto de redes, a fim de retirar dos equipamentos da rede as rotinas de maior complexidade elaboradas para o tratamento dos dados trafegados. A arquitetura de uma SDN possui, geralmente, um sistema operacional de rede centralizado, o qual controla as ações que devem ser tomadas pelos dispositivos de interconexão.

A Figura 1 ilustra o conceito de SDN [4]. Nela pode-se observar que a rede é controlada por um sistema operacional de rede centralizado (2) através da utilização do *OpenFlow* (1), o qual carrega consigo novas propostas para mecanismos de rede já conhecidos, como protocolos de controle e entrega de dados (3). No contexto de SDN, uma tecnologia eminentemente para proporcionar o controle centralizado do plano de dados é o *OpenFlow*, o qual é descrito na subseção a seguir.

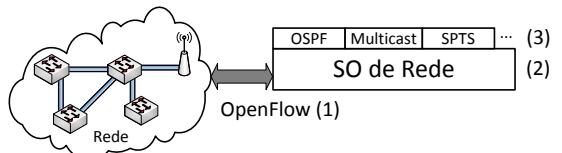


Figura 1. Conceito de funcionamento de SDN

### A. OpenFlow

O OpenFlow é um protocolo onde a tecnologia oferecida possibilita a execução de testes de novos protocolos (experimentais) em redes reais, coexistindo com o tráfego de produção. Esta característica do OpenFlow permite abstração e virtualização de redes, possibilitando o controle de tráfego da rede através de fluxos de dados [5].

O protocolo OpenFlow baseia-se em comutadores programáveis (*switches*) que combinam flexibilidade no desenvolvimento de novas aplicações de rede e facilidade para os fabricantes adaptarem os *switches* legados. Os *switches* OpenFlow são capazes de realizar encaminhamento de pacotes através de regras definidas em suas tabelas de fluxos (*flow tables*). Além disso, existe um elemento controlador (*controller*) conectado aos *switches* OpenFlow. No controlador, aplicações de rede são executadas fazendo uso do protocolo para comandar remotamente os *switches*, gerenciando os fluxos da rede [2]. Dessa forma, o controlador OpenFlow atua como um sistema operacional para gerenciamento e controle das redes, oferecendo uma plataforma com base na reutilização de componentes e diferentes níveis de abstração da rede (comandos da API - Application Programming Interface).

A tecnologia OpenFlow define o padrão de comunicação entre o *switch* e o controlador, possibilitando a adição, remoção e atualização de entradas nas tabelas de fluxos, sendo esta sua principal função [2]. Utilizando essa infraestrutura, no momento em que um *switch* recebe um quadro, ele deve consultar em suas tabelas de fluxos se há alguma ação definida para o fluxo do quadro recebido. Em caso positivo, as ações especificadas são realizadas e o quadro é encaminhado conforme definido. Caso contrário, quando não há ação definida para o quadro, o mesmo é enviado ao controlador, que irá determinar a ação a ser executada para o quadro recebido, possivelmente adicionando uma entrada em uma das tabelas de fluxo do *switch* para quadros desse fluxo [6]. Essa flexibilidade do controlador permite especificar um determinado fluxo e suas respectivas ações com um grande nível de detalhamento.

Com a flexibilidade de programação promovida pelo controlador em redes OpenFlow, pode-se repensar completamente protocolos de roteamento *multicast* sem o uso de algoritmos distribuídos, sendo esta a abordagem inovadora utilizada neste trabalho.

### III. TRABALHOS RELACIONADOS

No contexto da abordagem evolucionária da *Internet* o trabalho de Keshav e Paul [7] apresenta uma proposta centralizada de *multicast*. Utilizando uma estrutura hierárquica de domínios associados com *gateways*, roteadores e controladores raiz dos domínios, adotando para isso o conceito da separação do plano de dados e plano de fluxo, que é semelhante à lógica empregada neste trabalho, mas sem a flexibilidade das redes programáveis. Ainda no mesmo contexto, Ratnasamy *et al.* [8] propuseram a ideia de aproveitar rotas *unicast* existentes para distribuir pacotes *multicast* formando uma rede sobreposta. Entretanto essa solução não recebe suporte da infraestrutura da rede, atuando somente no nível de aplicação.

Recentemente, outras abordagens foram utilizadas, seguindo a linha *clean-slate*, assim como a proposta deste trabalho. A ideia por traz de abordagens *clean-slate* é reformular a maneira tradicional com que determinadas aplicações funcionam, muitas vezes quebrando paradigmas relacionados à soluções amplamente utilizadas. Em Martinez *et al.* [9], foi utilizado o *Multiprotocol Label Switching* (MPLS) sobre *Virtual Private Network* (VPN) para gerenciar tráfego *multicast*, porém essa combinação tem uma alta complexidade em função do modo como a rede é organizada, criando problemas de escalabilidade

para a solução. Por fim, Yap *et al.* [4] sugere primitivas de alto-nível (API) baseadas em OpenFlow para proporcionar um desenvolvimento mais amigável. Dentre as primitivas, há uma implementação simplificada de comunicação multiponto para OpenFlow, mas que não considera questões como mudanças nos grupos e gerência da árvore de escoamento.

Nenhum dos estudos acima leva em conta a abordagem adotada no Multiflow, um protocolo *multicast* escalável, com gerência do grupo, com o conhecimento antecipado da árvore de rotas e a preocupação com o tempo de processamento de eventos.

### IV. MULTIFLOW

Multiflow é uma proposta de abordagem *multicast clean-slate* em redes programáveis, em que os *hosts* podem entrar e sair do grupo *multicast* de forma dinâmica. Onde é fundamental a eficiência no processamento dos eventos de controle de grupo. A seguir descreve-se a arquitetura do Multiflow, e os mecanismos para definição de rotas e processamento dos eventos de grupo.

#### A. Descrição do Protótipo

O Multiflow é a aplicação que executa no controlador OpenFlow. Optou-se por utilizar o controlador NOX [10] para a criação do protótipo, devido a relativa escalabilidade fornecida por esse controlador. O NOX é implementado nas linguagens C e Python e é composto por módulos que descrevem novas funcionalidades. Estes módulos são majoritariamente desenvolvidos na linguagem Python. NOX oferece uma série de recursos, como por exemplo APIs para a manipulação de pacotes de diversos protocolos, como UDP e IP. Para construir a topologia de rede virtual onde os testes do protótipo foram executados utiliza-se a API do Mininet [3].

#### B. Funcionamento

Definida a topologia da rede, a aplicação Multiflow permanece atenta à ocorrência de pacotes do tipo IGMP na rede. Baseado no protocolo IGMP, o Multiflow identifica pacotes cujo destino seja um grupo *multicast*. A Figura 2 ilustra o diagrama de fluxo de operação do Multiflow, mostrando as operações realizadas quando um host anuncia ser o provedor dos dados de um grupo *multicast*, quando um cliente ingressa em um grupo *multicast* e, em seguida, quando um cliente deixa de fazer parte do grupo.

Para iniciar a transmissão para um determinado grupo *multicast*, o servidor de dados envia um pacote do tipo IGMP Query. Uma vez identificado este pacote, o Multiflow reconhece o grupo *multicast* do pacote como um dos grupos ativos na rede, armazenando-o em uma lista de grupos. Os clientes interessados em ingressar no grupo *multicast* devem enviar um pacote do tipo IGMP Join endereçado ao grupo *multicast*.

Ao identificar pacotes IGMP Join, o controlador executando a aplicação Multiflow, que possui conhecimento prévio da topologia da rede, reconhece o interesse do(s) cliente(s) em participar do grupo *multicast* e então, inicia o algoritmo de descoberta da melhor rota entre o servidor e o(s) cliente(s) interessado(s). Para o cálculo de melhor rota, o Multiflow utiliza o algoritmo de Dijkstra [11], amplamente utilizado com esse propósito.

O algoritmo de Dijkstra trabalha com o conceito de peso de cada aresta (conexão). Neste trabalho, o peso de cada aresta do grafo foi definido como o representativo da distância entre os dois nós da rede. A melhor rota é definida como sendo a que apresenta o menor peso agregado. Uma vez calculada a rota, o controlador insere nos equipamentos da rede as regras de fluxo necessárias para o roteamento dos pacotes, iniciando a distribuição do conteúdo aos clientes do grupo.

O outro evento presente no funcionamento do Multiflow é a saída de um cliente de um grupo *multicast* ao qual ele pertence. Esse evento ocorre quando o cliente, não mais interessado, envia um pacote do tipo IGMP Leave. Ao receber este pacote, a aplicação Multiflow busca pela rota que foi previamente traçada para o cliente e a apaga as regras equivalentes nas tabelas de fluxo dos *switches*, cancelando o envio de pacotes destinados ao grupo para o cliente em questão.

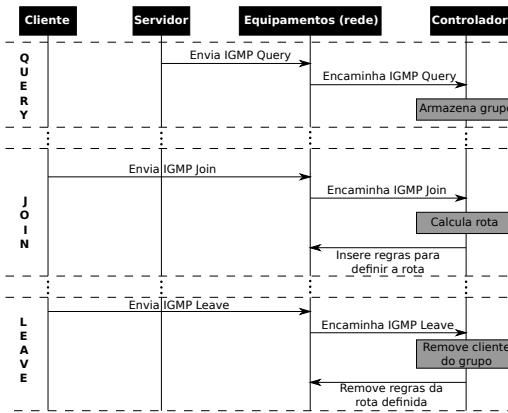


Figura 2. Diagrama de fluxo de operação do Multiflow

## V. AVALIAÇÃO DO PROTÓTIPO

Esta seção apresenta detalhes da metodologia utilizada para avaliação da implementação do Multiflow. O foco dessa metodologia é identificar o impacto do controlador no desempenho do controle de roteamento e na formação dos grupos *multicast*. A seguir, apresenta-se o ambiente de testes utilizado para execução dos experimentos e o cenário avaliado.

### A. Ambiente de Experimentação

Para realizar os experimentos, primeiramente foi criada e emulada uma topologia de rede virtual com *switches* que suportam o protocolo OpenFlow através do *software* Mininet [3]. A partir disto, a definição dos *hosts* ativos e pertencentes aos grupos *multicast*, bem como a escolha das fontes do *multicast* são realizadas a partir da execução ordenada de um conjunto de *bash scripts* desenvolvidos utilizando a API do *software open-source* Mausezahn (MZ versão 0.40) [12], utilizado para geração e avaliação de tráfego de rede. Os *scripts* geram os pacotes IGMP de *query*, *join* e *leave*.

Para testar o comportamento de aplicações *multicast* executando nos clientes e servidores, foi criada na linguagem de programação C, uma aplicação padrão cliente/servidor, que permite efetuar o envio e recebimento de mensagens no formato UDP. A aplicação no modo servidor é iniciada na fonte do *multicast*, enviando pacotes para o endereço IP do grupo *multicast*. Nos demais *hosts* do grupo *multicast* uma instância no modo cliente é executada, recebendo os pacotes enviados para o grupo pela fonte do *multicast*. Esta mesma aplicação é capaz de fornecer ainda estatísticas de desempenho da solução, tratando o tempo decorrido do estabelecimento da rota e a chegada do primeiro pacote de dados.

### B. Cenário

Para a avaliação, foi definida uma topologia em árvore, contendo sete *switches* e nove *hosts*, dos quais dois são responsáveis pela geração de conteúdo para o grupo *multicast*

e os demais representam os clientes interessados ou não nos grupos *multicast*.

Com esse cenários são avaliados dois parâmetros importantes: o impacto do controlador Multiflow no desempenho do controle de roteamento e o impacto no desempenho da formação dos grupos *multicast*. Multiflow foi comparado com outro controlador, o OpenMcast, que confere à rede um comportamento semelhante ao observado em redes normais, que não fazem uso de SDN. Esse controlador é descrito a seguir.

### C. Controlador OpenMcast

O controlador OpenMcast foi desenvolvido para simular em um ambiente de SDN as operações *multicast* realizadas em um ambiente de rede padão, baseado no funcionamento tradicional do protocolo IGMP, onde é realizada a propagação dos pacotes de controle na rede. Uma vez gerado o pacote IGMP Query pelo *host* servidor, o controlador identifica o pacote e o propaga pela rede através da operação de *flood*, onde o pacote é encaminhado para todas as portas dos *switch* em que foi identificado, exceto pela porta de entrada do pacote. A operação de *flood* é realizada por todos os *switches* da rede, a fim de que todos tomem conhecimento da existência do servidor, do grupo *multicast* ao qual o servidor está vinculado e da porta para a qual o *switch* têm acesso ao servidor. Esta operação é realizada para os dois servidores previamente configurados.

Uma vez conhecidos os grupos *multicast* ativos na rede, os *hosts* clientes iniciam as respectivas operações de *join*, gerando pacotes do tipo IGMP Join. Os pacotes IGMP Join são propagados pela rede através da porta que dá acesso ao servidor do *switch* até alcançarem os *switches* que estão diretamente conectados aos servidores. Através desta retro propagação, o caminho entre o *host* servidor e os clientes pertencentes ao grupo *multicast* do servidor é definido, iniciando então a entrega de pacotes UDP destinados ao grupo *multicast* aos respectivos clientes participantes do grupo.

Para a operação de *leave*, o pacote IGMP Leave deve ser retro propagado da mesma maneira que o pacote IGMP Join. Nesta retro propagação, o caminho anteriormente definido é apagado das regras dos *switches* pertencentes ao caminho, a fim de interromper a distribuição dos pacotes ao cliente.

Utilizando o controlador OpenMcast, as trocas de pacotes de controle na rede podem gerar um tráfego consideravelmente alto, uma vez que, em uma rede real com um grande número de *hosts* são frequentes as operações de *join* e *leave* em grupos *multicast*. Além disso, perdas de pacotes podem ocorrer durante a propagação dos pacotes, o que invalida a operação, que deve ser executada novamente.

A sequência de operações realizadas para validação foi planejada baseando-se em uma situação real, onde clientes deixam de pertencer a um grupo *multicast*, passam a pertencer a outro e ainda, clientes sem associação prévia a um grupo, solicitam ingresso em um grupo *multicast*. Na seção seguinte, serão apresentados os resultados envolvendo os testes realizados no cenário proposto.

## VI. RESULTADOS

Nesta seção são apresentados os resultados obtidos após uma série de repetições dos experimentos. Os resultados refletem o cenário definido na Seção V.

Foram realizadas no total mil medições do tempo entre a operação de *join* do cliente e o recebimento do primeiro pacote UDP de dado destinado ao grupo. O tempo gasto para configurar os *switches* apresenta uma variabilidade. O tempo médio observado entre a operação de *join* do cliente e o recebimento

do primeiro pacote UDP na abordagem OpenMcast foi de aproximadamente 715 milissegundos, com um desvio padrão de 357 milissegundos. Na abordagem Multiflow, a mesma operação apresentou tempo médio de aproximadamente 427 milissegundos, com um desvio padrão de 431 milissegundos.

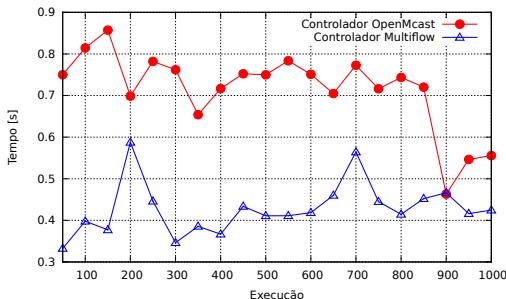


Figura 3. Tempo médio a partir do *join* do cliente e o recebimento do primeiro pacote

A Figura 3 ilustra, no eixo vertical, o tempo total em segundos entre o envio do *join* do cliente e o recebimento do primeiro pacote UDP e, no horizontal, a média das 50 medições das execuções do teste. Este agrupamento foi realizado a fim de tornar o gráfico mais claro e facilitar sua análise. Observa-se que o tempo medido com o controlador Multiflow, que possui o conhecimento da topologia de rede, é menor que o tempo do controlador OpenMcast, que necessita propagar os pacotes de controle pela rede.

Níveis (n)	Queries geradas (k)	Queries propagadas no OpenMcast	Queries propagadas no Multiflow
2	100	300	100
3	100	700	100
4	100	1500	100

Tabela I

PROPAGAÇÃO DE PACOTES NA REDE NAS DUAS ABORDAGENS

Outro resultado interessante pode ser observado na Tabela I, que apresenta como ocorre a propagação de pacotes IGMP Query na topologia de rede avaliada. No controlador OpenMcast, a relação de *queries* geradas por *queries* propagadas na rede ocorre obedecendo a equação  $k \cdot 2^n - k$ , onde  $k$  representa o número de *queries* geradas e  $n$  representa o número de níveis da topologia em árvore. Ou seja, quanto mais níveis a topologia possuir, maior será o número de pacotes de controle sendo propagados na rede. Já no controlador Multiflow, observa-se que o número de *queries* propagadas é o mesmo número de *queries* geradas, já que neste controlador não há a propagação de pacotes de controle da rede.

## VII. CONCLUSÃO E TRABALHOS FUTUROS

O *multicast* é uma técnica de roteamento de pacotes para um grupo específico de *hosts* na rede, onde o principal benefício é a redução do tráfego devido ao apoio dos roteadores em replicar as mensagens. Este trabalho propõe uma abordagem *multicast clean-slate* logicamente centralizado baseado em redes programáveis OpenFlow, onde durante o *setup* do grupo *multicast*, realiza-se o cálculo antecipado da melhor rota possível entre a fonte e o cliente interessado, com o objetivo de reduzir ao máximo o atraso no processamento de eventos de grupo *multicast*, como entrada e saída de *hosts* e o tráfego na rede.

Definida a abordagem, foi implementada a prova de conceito chamada Multiflow. Realizaram-se experimentos para caracterizar o tempo de *setup* e de processamento dos eventos. Os resultados mostraram que a utilização de um controlador, com conhecimento da topologia da rede, traz um ganho de desempenho global satisfatório, em ordem de milissegundos, mais rápido que resultados publicados na literatura de IP *multicast*, uma vez que menos informações de controle são trocadas e a melhor rota pode ser calculada baseada na topologia. Conclui-se que a proposta pode trazer benefícios para aplicações com requisitos de comunicação multiponto em que ocorrem muitas operações de entradas e saídas nos grupos *multicast*, como ocorre em IPTV.

Como trabalhos futuros, espera-se desenvolver experimentos com cenários mais próximos daqueles encontrados na *Internet* direcionados a serviços de *streaming*, principalmente em termos da escala. Questões de segurança quanto ao ingresso de clientes ao grupos e confidencialidade das mensagens transmitidas podem ser consideradas no futuro. Além disso, avaliar heurísticas que possam ser aplicadas ao algoritmo de definição da rota dos fluxos, de modo a reduzir sua complexidade, aproveitando-se dos benefícios oferecidos pela tecnologia OpenFlow.

## REFERÊNCIAS

- [1] P. Paul and S. V. Raghavan, "Survey of multicast routing algorithms and protocols," in *Proceedings of the 15th international conference on Computer communication*, ser. ICCC '02, Washington, DC, USA: International Council for Computer Communication, 2002, pp. 902–926. [Online]. Available: <http://dl.acm.org/citation.cfm?id=838138.838216>
- [2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008. [Online]. Available: <http://doi.acm.org/10.1145/1355734.1355746>
- [3] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," in *Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks*, ser. Hotnets '10, New York, NY, USA: ACM, 2010, pp. 19:1–19:6. [Online]. Available: <http://doi.acm.org/10.1145/1868447.1868466>
- [4] K.-K. Yap, T.-Y. Huang, B. Dodson, M. S. Lam, and N. McKeown, "Towards software-friendly networks," in *Proceedings of the first ACM asia-pacific workshop on Workshop on systems*, ser. APSys '10, New York, NY, USA: ACM, 2010, pp. 49–54. [Online]. Available: <http://doi.acm.org/10.1145/1851276.1851288>
- [5] Y. Kanumi, S. Saito, and E. Kawai, "Toward large-scale programmable networks: Lessons learned through the operation and management of a wide-area openflow-based network," in *Network and Service Management (CNSM), 2010 International Conference on*, oct. 2010, pp. 330–333.
- [6] ONF2011, "Openflow switch specification version 1.2.0," Disponível em <http://www.openflow.org/documents/openflow-spec-v1.2.pdf>. Acesso em Maio de 2012., dec. 2011.
- [7] S. Keshav and S. Paul, "Centralized multicast," in *Proceedings of the Seventh Annual International Conference on Network Protocols*, ser. ICNP '99, Washington, DC, USA: IEEE Computer Society, 1999, pp. 59–68. [Online]. Available: <http://dl.acm.org/citation.cfm?id=850936.852461>
- [8] S. Ratnayam, A. Ermolinskiy, and S. Shenker, "Revisiting ip multicast," in *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '06, New York, NY, USA: ACM, 2006, pp. 15–26. [Online]. Available: <http://doi.acm.org/10.1145/1159913.1159917>
- [9] I. Martinez-Yelmo, D. Larrabeiti, I. Soto, and P. Pacyna, "Multicast traffic aggregation in mpls-based vpn networks," *Communications Magazine, IEEE*, vol. 45, no. 10, pp. 78–85, october 2007.
- [10] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "Nox: towards an operating system for networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 3, pp. 105–110, Jul. 2008. [Online]. Available: <http://doi.acm.org/10.1145/1384609.1384625>
- [11] E. W. Dijkstra, "A note on two problems in connection with graphs," *Numerische Mathematik*, vol. vol. 1, pp. 269 – 271, 1959.
- [12] H. Haas, "Mausezahl fast traffic generator," Disponível em <http://www.perihel.at/sec/mzl/>. Acesso em Maio de 2012., mai. 2012.

## **Balanceamento de Carga em Sistema de Transações Eletrônicas Financeiras com RMI**

Alexandre Luis de Andrade, Cristiano André da Costa, Tiago André Jost, Rodrigo da Rosa Righi  
Programa Interdisciplinar de Pós-Graduação em Computação Aplicada – Unisinos  
{alexandreluisandrade, tiagojost}@gmail.com, {rrrighi, cac}@unisinos.br

### **Resumo**

*Este artigo tem por objetivo apresentar, ainda que de forma preliminar, um arcabouço de balanceamento de carga e tolerância a falhas chamado GetLB. Ele atua para aumentar a eficiência quanto ao tratamento de transações eletrônicas nos centros de processamento. A eficiência, nesse contexto, é entendida pelo atendimento de um maior número de transações por segundo e redução no índice de descarte de transações. Para tal, o comutador que recebe as transações trabalha com informações pertinentes das máquinas processadoras para decidir sobre o despacho. Além disso, ele pode receber notificações para evitar um escalonamento para um determinado alvo. O protótipo implementado usou RMI e testes preliminares mostraram que o arcabouço de interação entre os envolvidos no despacho é viável.*

*keywords:* *balanceamento de carga, sistema de transações eletrônicas financeiras, pagamento eletrônico, RMI, Round-Robin.*

### **1. Introdução**

Sistemas de roteamento e processamento de requisições financeiras são elementos fundamentais de uma rede de transações eletrônicas [1, 11]. Uma transação eletrônica está aliada a uma requisição de compra ou saldo e percorre um caminho de ida e volta desde um terminal até um centro de processamento. Um terminal pode ser representado por um POS (*Point of Sale*), ponto de venda TEF (*Transferência Eletrônica de Fundos*), ATM (*Automatic Teller Machine*), bem como por dispositivos móveis. Nesse contexto, o presente artigo aborda a situação da empresa GetNet, que processa mais de 50 milhões de transações por mês. Naturalmente, épocas de maior movimento financeiro no comércio geram demandas maiores no sistema.

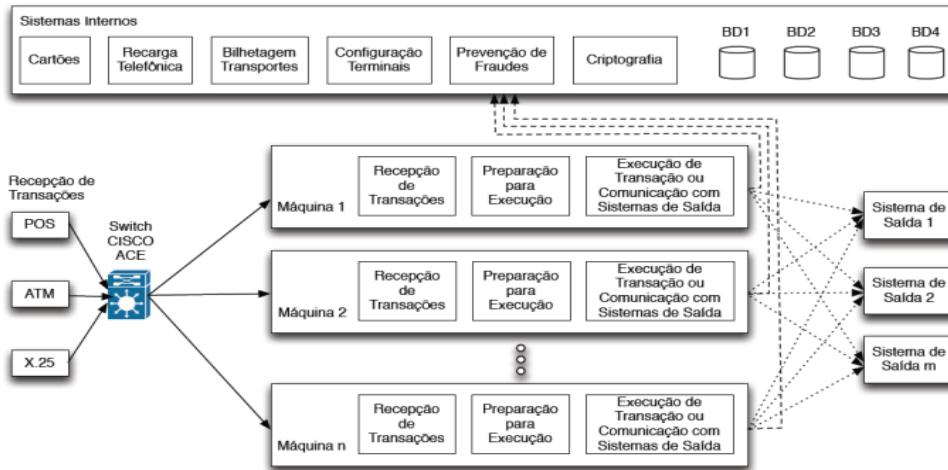
A empresa GetNet suporta diferentes tipos de transações, das quais se pode destacar cartões de crédito e débito, recarga de telefonia pré-paga, transações de saque e depósito para sistema bancário e consultas financeiras, entre outros. Cada um desses tipos possui requisições próprias de CPU e acesso a banco de dados, podendo usar diferentes subsistemas

dentro do centro de processamento. Atualmente, a empresa conta com dois centros que operam independentemente para realizar a computação das transações. Em especial, esse artigo está focado na dinâmica daquele responsável pelos cartões de bandeiras regionais e recarga de telefonia pré-paga. A sua estrutura de recepção e envio de transações está ilustrada na Figura 1. A arquitetura apresenta um comutador que atua como um receptor de transações e máquinas processadoras (denominadas MP ou “Máquina n” na Figura 1) que as recebem e as tratam de acordo com a semântica de cada uma. Os sistemas de saída na Figura 1 dizem respeito a empresas alvo de cada transação.

Cada transação eletrônica engloba as etapas de solicitação, resposta e confirmação [1]. Os objetivos principais no tratamento das requisições são os seguintes: (i) alto desempenho no processamento das transações com menor custo computacional possível; (ii) alta disponibilidade para evitar perda de transação. Esses dois objetivos passam por um eficiente escalonamento de transações feito pelo elemento comutador e a análise da escalabilidade da solução corrente para o processamento de dados.

Atualmente, as máquinas MP são homogêneas e estão alocadas em um mesmo *datacenter*. Uma análise do crescimento da empresa poderia levar ao uso de subsistemas regionais, cada qual com máquinas MP potencialmente heterogêneas. Além disso, alguns países impõem regras de segurança que afirmam que o sistema de processamento de transações deve estar em território nacional. Portanto, a descentralização no tratamento das máquinas processadoras representa uma maneira da GetNet atender tais países. Além da questão da homogeneidade e centralização, a solução corrente faz com que as transações sejam distribuídas segundo o método de escalonamento Round-Robin (RR) [2]. Naturalmente, tal método não é o melhor para sistemas heterogêneos ou dinâmicos, uma vez que atua simplesmente com uma lista circular para efetuar suas ações.

O método RR restringe o uso de recursos computacionais com capacidades especializadas, como o tratamento da criptografia ou decodificação de imagens de forma rápida diretamente em processadores específicos. Outro aspecto a ser considerado na



**Figura 1 - Topologia do sistema de transações eletrônicas financeiras**

distribuição RR é remete ao fato que as transações serão encaminhadas para processamento em máquinas que se presumem estar em plenas condições de atuação. Atualmente, esse problema é contornado com a adição de várias máquinas atuando em paralelo, minimizando a perda de algum recurso.

O presente artigo apresenta preliminarmente uma proposta de arcabouço para balanceamento de carga chamada GetLB. Ele atua como uma alternativa ao uso do método RR no momento do despacho das transações, de modo a considerar o dinamismo das máquinas MP e informações como CPU, disco, rede e memória. Além das vantagens de poder contar com um sistema pró-ativo, também torna possível construir uma arquitetura com alocação de máquinas em diferentes localidades e permite a operação de máquinas heterogêneas no sistema de processamento. Ainda, o presente artigo descreve um protótipo com RMI implementado a partir do arcabouço mencionado.

A Seção 2 do artigo apresenta o arcabouço proposto, bem como o protótipo desenvolvido. Em adição, essa seção descreve o método utilizado para obter uma distribuição mais eficiente das transações. Na seção 3 serão analisados os resultados dos testes. A Seção 4 apresentam alguns trabalhos relacionados e por fim, a conclusão é descrita na Seção 5. Ela aborda questões relativas às contribuições técnicas e científicas do trabalho realizado.

## 2. Trabalhos relacionados

Meios eletrônicos de pagamento são cada vez mais adotados, em detrimento da adoção de dinheiro em

papel moeda e cheque [4, 5]. Além de conveniência para consumidores, o uso de cartões eletrônicos eneficia instituições de comércio e facilita o acesso a aplicações e serviços na Internet. Virnes et al. [4] afirmam que essa transição aparece tanto nos bancos e sistemas de e-commerce, quanto na governança eletrônica, entretenimento, sistemas de saúde e dispositivos móveis. Um dos tópicos mais estudado em sistemas de transações eletrônicas diz respeito à segurança da informação [6, 7, 8]. Vishik et al. [8] apresentam que ambas transmissão segura de dados e relação de confiança devem ser reanalisadas na medida que sistemas embarcados e smartphones recebem espaço para lançar transações. Em especial, Sastre, Bacon e Herrero [6] discutem algoritmos de segurança otimizados para atender diferentes meios de transmissão, como ADSL e GPRS.

Sousa et al. [9] apresentam um modelo estocástico para avaliação de desempenho e planejamento de recursos em sistemas de transferência eletrônica de fundos (EFT). Estes autores fazem um estudo do desempenho levando em consideração características de dependabilidade como disponibilidade, confiabilidade [10], escalabilidade e segurança. Segundo eles uma análise de um sistema EFT sem esses critérios pode levar a resultados imprecisos. Ainda, Sousa et al. relatam que os critérios mostrados anteriormente devem guiar o uso eficiente de recursos para que seja mantido o SLA (acordo de serviço) com os clientes. Araújo et al. [1] afirmam que a análise de desempenho deve observar o pior caso de volume de chegada de transações para ser verossímil com a realidade da empresa de processamento de dados. Para

tal, eles adotaram Redes de Petri e fazem uso de informações de acesso e armazenamento em disco, além do volume transacional.

### 3. Sistema balanceamento de transações

O sistema transacional, conforme descrito na seção anterior, é composto por um comutador que recebe as transações originadas de terminais instalados em estabelecimentos comerciais. O comutador é um equipamento Cisco ACE 65000 [3] que concentra e distribui as transações entre as MP. Para criar o protótipo do sistema GetLB foi criado um ambiente fictício que comprehende uma máquina ACE e outras que realizam o papel de máquina MP. A primeira será utilizada para a entrada das transações e realiza o papel de comutador.

A ideia do arcabouço passa pela criação de objetos remotos tanto na máquina ACE quanto em cada MP. A máquina ACE tem objetos remotos que guardam informações sobre cada uma das máquinas MP. A ideia dessa abordagem está centrada na eficiência no momento do despacho das transações, uma vez que o cálculo do escalonamento não comprehende informações pela rede. Cada máquina MP cria um objeto remoto para tratar o enfileiramento e processamento de transações. Nesse sentido, a máquina ACE possui procurações para os objetos remotos presentes nas máquinas MP e deve decidir qual deles acionar no momento da chegada de uma nova transação.

Na máquina ACE será criado um vetor de objetos remotos, os quais serão instanciados nas máquinas MP. O número de elementos nesse vetor é igual a quantidade de máquinas MP. Cada objeto remoto é responsável por guardar informações como CPU, disco, memória e tempo de rede sobre uma máquina MP em particular. O tempo de rede mencionado se refere ao seguinte custo: tempo para transferir 1 byte multiplicado pela quantidade de bytes que representam uma transação. Na criação de uma máquina MP, dar-se-á a criação de um objeto procurador que atuará sobre o remoto na máquina ACE. Um novo fluxo de execução chamado MachineThread é criado no construtor na classe que define uma máquina MP. Ele tem o papel de periodicamente coletar dados atualizados do estado corrente da máquina e chamar os métodos de atualização do objeto remoto.

No momento da chegada de uma transação, o papel da máquina ACE é decidir para qual máquina MP ela será despachada. Para tal, o objeto que instancia a classe ACE faz uso dos dados locais para a tomada dessa decisão. Nesse ponto está em desenvolvimento uma heurística chamada Potencial de Recebimento

(PR), que irá medir o quanto uma determinada máquina MP está apta a receber uma nova transação. Preliminarmente, a Equação 1 define como o PR será computado.

$$(1) \quad PR = FreeHD + FreeMem + FreeCPU \\ + FreeBandwidth$$

Cada Máquina MP terá seu respectivo PR e será utilizado como critério de distribuição de carga. Além do vetor de objetos remotos na máquina ACE, o arcabouço também compreende a criação de um objeto remoto em cada máquina MP. Cada um deles é uma instância da classe TransactionQueue e é responsável por implementar uma fila para o tratamento de transações. O objeto na máquina ACE têm procurações para cada objeto remoto desse tipo. Logo após a decisão da máquina alvo para receber uma transação, dar-se-á a colocação dela diretamente na fila adequada. A estrutura de objetos remotos e a interação entre máquinas MP e ACE está ilustrada na Figura 2.

O processo tem início com as transações sendo recebidas em ACE, a seguir, com base nos resultados coletados das máquinas MP os respectivos PRs são calculados. As máquinas MP que possuem maiores índices são as que recebem as transações em TransactionQueue.

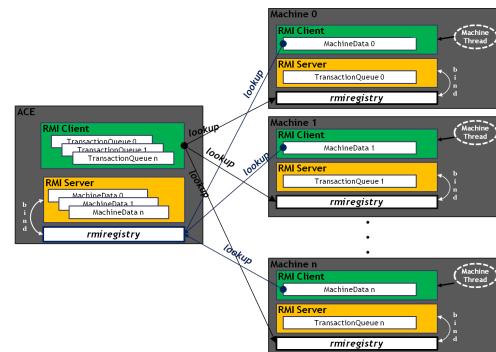


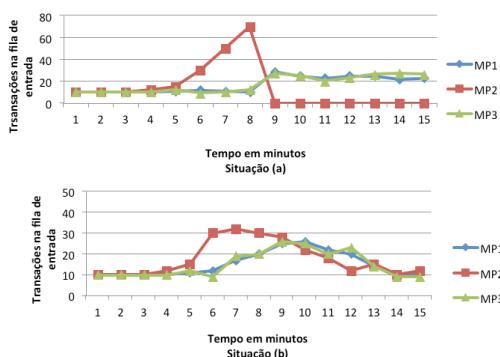
Figura 2 - Arquitetura do sistema GetLB

Com isso se obtém uma distribuição adequada das transações através do sistema, no qual as transações passam a ser distribuídas de acordo com a capacidade das máquinas. Em adição, obtem-se melhor equilíbrio de carga na distribuição e evita-se possibilidade de que alguma máquina sem condições de processamento receba demanda em situações críticas. A arquitetura proposta também possibilita que em ACE os tipos de transações sejam separadas para serem enviadas para máquinas dedicadas, criando novos TransactionQueue e MachineData especializados.

### 3. Análise de resultados

Para uma análise dos resultados do sistema, primeiramente será verificado o comportamento do processamento das transações com balanceamento de carga através do método RR. Conforme se verifica na Figura 3(a), as quantidades de transações que estão sendo atendidas em três máquinas seguem em equilíbrio até um dado instante em que ocorre uma situação de anormalidade em uma delas. Com o método RR, observa-se que há um aumento linear do enfileiramento de transações na máquina afetada, uma vez vista que está apresentando problemas em processar transações. O sistema segue nessa tendência até que a máquina entre em colapso e fique indisponível. A falha em uma das máquinas resulta em aumento de transações nas demais. Esse processo pode levar a um efeito em cascata, levando o sistema a um *crash* e, por consequência, à indisponibilidade total.

Na Figura 3(b), o mesmo problema de equilíbrio das máquinas é identificado em ACE através de MachineData. Ao aplicar o algoritmo de escalonamento PR, o despacho de transações para a máquina com problemas é interrompido e aquelas com melhores condições são usadas de modo a tornar o sistema balanceado. Por fim, verifica-se no gráfico que uma mesma situação de enfileiramento é contornada, possibilitando que seja possível voltar às condições originais de processamento.



**Figura 3 - (a) Situação de balanceamento Round-Robin (b) Enfileiramento de transações contornado com balanceamento através da distribuição de carga com RMI.**

### 5. Conclusão

O sistema proposto apresentou resultados bastante satisfatórios para os critérios de escalonamento adotados. Ou seja, foi possível contornar situações de

problemas que normalmente são ignoradas no método Round-Robin de escalonamento. Naturalmente, uma solução RMI para um sistema que processa grandes volumes de transações pode não ser a mais adequada. RMI foi escolhida dada a sua facilidade de uso para os testes iniciais de viabilidade da arquitetura proposta. A cargo de trabalhos futuros, pretende-se evoluir o cálculo de PR e obter um método que seja adotado no sistema transacional da GetNet. Em adição, serão analisadas alternativas com baixa sobrecarga para comunicação via rede. Em particular, o despacho de transações entre ACE e máquinas MP será testado com uso do protocolo SNMP ou mensagens UDP.

### 6. Bibliografia

- [1] C. Araujo, E. Sousa, P. Maciel, F. Chicout, and E. Andrade. Performance modeling for evaluation and planning of electronic funds transfer systems with bursty arrival traffic. In *Intensive Applications and Services, 2009*.
- [2] R. Rojas-Cessa e Chuan-bi Lin, “Frame occupancybased round-robin matching scheme for input-queued packet switches”, vol. 3 (IEEE, [s.d.]), 1845-1849.
- [3] Cisco. ACE Client and Servers Hitting the Same VIP Disponível em <[www.cisco.com/en/US/products/hw/switches/ps708](http://www.cisco.com/en/US/products/hw/switches/ps708)>. Acesso em: 02/ ago. 2012.
- [4] J. Vines, M. Blythe, P. Dunphy, and A. Monk. Eighty something: banking for the older old. In *Proceedings of the 25th BCS Conference on Human-Computer Interaction*, pages 64–73, UK, 2011. British Computer Society.
- [5] L. Xiaojing, W. Weiqing, and Z. Liwei. The mechanism analysis of the impact of ecommerce to the changing of economic growth mode. In *Robotics and Applications (ISRA), 2012 IEEE Symposium on*, pages 698 –700, 2012.
- [6] R. Sastre, S. Bascon, and F. Herrero. New electronic funds transfer services over ip. In *Electrotechnical Conference, 2006. IEEE*, pages 733 – 736, may 2006.
- [7] P. Seltsikas, G. Marsh, M. Frazier-McElveen, and T. J. Smedinghoff. Secure government in cyberspace? In *Proceedings of the 12th Annual International Digital Government Research Conference, dg.o '11*, pages 359–361, New York, NY, USA, 2011. ACM.
- [8] C. Vishik, A. Rajan, C. Ramming, D. Grawrock, and J. Walker. Defining trust evidence: research directions. In *Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research, CSIIRW '11*, pages 66:1–66:1, New York, NY, USA, 2011. ACM.
- [9] E. Sousa, P. Maciel, C. Araujo, and F. Chicout. Performability evaluation of eft systems for sla assurance. In *Parallel Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1 –8, may 2009.
- [10] A. Avizienis, J.-C. Laprie, B. Randell, e C. Landwehr, “Basic concepts and taxonomy of dependable and secure computing”, *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, p. 11- 33, mar. 2004.
- [11] L. Liu, M. Song, X. Luo, H. Bai, S. Wang, and J. Song. An implementation of the online-payment platform based on saas. In *Web Society (SWS), 2010 IEEE 2nd Symposium on*, pages 658 –662, aug. 2010.

# Estudo da viabilidade do ROS como plataforma para IoT

Vinícius Alves Hax, Nelson Lopes Duarte Filho, Silvia Silva da Costa Botelho, Odorico Machado Mendizabal  
Centro de Ciências Computacionais  
Universidade Federal do Rio Grande – FURG  
Campus Carreiros: Av. Itália km 8 Bairro Carreiros, Rio Grande, Brasil  
viniciushax@furg.br, dmtnldf@yahoo.com.br, {silviacb, odoricomendizabal}@furg.br

**Resumo**—O artigo apresenta um estudo a respeito da viabilidade em utilizar o framework *Robot Operating System* como base para um middleware de *Internet of Things*. São apresentados os conceitos de IoT e do ROS, e logo após, apresentados critérios de avaliação. Por fim, casos de uso hipotéticos de aplicação são analisados com base nas características existentes do ROS e nas premissas de um ambiente de IoT.

## I. INTRODUÇÃO

Esse artigo descreve um estudo de viabilidade da plataforma *Robot Operating System* (ROS) como base para desenvolvimento de uma infraestrutura voltada para o ambiente de *Internet of Things* (IoT). Devido ao crescente interesse em IoT torna-se importante prover ambientes e ferramentas que facilitem o desenvolvimento de novas soluções. Por se tratar de uma área nova, ainda não estão estabelecidas as ferramentas necessárias para implementação dessas soluções. Nesse contexto, surge a possibilidade de utilizar a plataforma ROS como base para uma solução de infraestrutura de IoT. A sua utilização é interessante pois fornece uma abstração que permite acessar hardwares externos como sensores e atuadores de forma facilitada e a sua implementação foi organizada de forma modular, facilitando a adição de novas funcionalidades. Critérios de avaliação de aplicações IoT são analisados no contexto do ROS para saber se o mesmo pode ser utilizado como base para um middleware de IoT. Na seção II é apresentado o conceito de IoT e alguns desafios inerentes desse paradigma. Na seção III é apresentada a plataforma ROS. A seção IV define os critérios utilizados na análise de viabilidade do ROS no contexto de IoT. A seção V analisa o ROS segundo um conjunto de critérios e por fim a seção VI apresenta as conclusões do estudo.

## II. INTERNET OF THINGS

Recentemente surge um novo paradigma tecnológico denominado Internet of Things (IoT), ou Internet das Coisas. O conceito começa a ser explorado cientificamente bem como desperta o interesse de empresas como Verizon, AT&T e Cisco[1] e agências de origem governamental como o Information Society and Media Department da União Europeia [2]. Também a IoT foi considerada como uma das seis tecnologias civis com maior potencial de impacto nos Estados Unidos[3].

A Internet of Things pode ser definida como “uma infraestrutura de rede global e dinâmica com capacidade

de autoconfiguração baseada em padrões e protocolos onde objetos virtuais e reais tem identidades, atributos, personalidade, que usam interfaces inteligentes e são integradas em uma rede de informação” [2]. Esses objetos podem interagir entre si e com o ambiente através de sensores e atuadores diversos. O impacto tecnológico deste novo paradigma computacional pode abranger diferentes áreas de aplicações, tais como setor automotivo, aeroespacial, edifícios inteligentes, sistemas de auxílio à saúde, logística, manufatura, entre outras.

### A. Hardware

O *Internet Engineering Task Force* (IETF)<sup>1</sup>, instituição responsável por diversos padrões para equipamentos de intercomunicação, criou em 2005 um grupo denominado *IPv6 Over Low Power Wireless Personal Network*, com o objetivo de padronizar a utilização dos protocolos de rede disponíveis em redes de baixa potência sem fio de alcance pessoal. Nos documentos produzidos pelo grupo, [4], caracteriza os equipamentos que compõe essas redes como:

- Tendo pequeno tamanho de pacotes de dados, em torno de 81 bytes.
- Suportando 16 bits ou 64 bits padrão IEEE estendido para endereços na camada de acesso ao meio.
- Baixa largura de banda: Taxas de 250, 40 e 20 kbps.
- Topologia em estrela e em malha.
- Baixa potência. Em muitas ocasiões são dispositivos que utilizam bateria.
- Baixo custo. Geralmente estão associados com sensores e possuem baixa capacidade de processamento e pouca memória.
- Alto número de dispositivos instalados.
- Localização dos dispositivos não é pré-definida, pois são instalados à medida que se tornam necessários.

Além disso os dispositivos integrantes dessas redes são de dois tipos:

- Dispositivos de funcionalidade reduzida, que possuem limitações maiores de processamento e memória e geralmente não roteiam pacotes
- Dispositivos de funcionalidade completa, com maior capacidade do que os anteriores, geralmente atuando como roteadores para os nodos de hardware mais limitado

<sup>1</sup><http://www.ietf.org/>

### B. Utilização do protocolo IP em IoT

Na medida em que se faz necessário escolher um protocolo de rede para o funcionamento da IoT, uma opção à ser considerada é utilizar o próprio protocolo IP que se encontra atualmente em utilização. Como vantagens podemos citar que o protocolo IP já está amplamente disseminado e conta com diversos softwares de diagnóstico, configuração e resolução de problemas. A especificação aberta do protocolo também é favorável na sua adoção. Alguns dos problemas do IPv4 são resolvidos no IPv6, como o suporte ao grande número de dispositivos e incorporação de mecanismos de autoconfiguração de rede. Essas características o tornam um protocolo propício para IoT.

### III. ROBOT OPERATING SYSTEM

A plataforma ROS surgiu baseada na dificuldade de integrar soluções disponíveis na área de robótica. Diferentes tipos de robôs possuem hardware muito variado, dificultando o processo de escrita de software, que tradicionalmente precisa ser escrito especificamente para cada hardware. Além disso os hardwares utilizados possuem propósitos e implementações diversas, incluindo atuadores no formato de rodas até sensores visuais [5]. Segundo [6] a necessidade de profundos conhecimentos específicos necessários para cada pesquisador muitas vezes estão além do que é factível. Com o objetivo de facilitar o desenvolvimento na área de robótica surgiu o ROS.

Um dos critérios no desenvolvimento da plataforma foi utilizar uma comunicação *peer-to-peer*, ou seja, uma comunicação não centralizada, distribuída entre os nodos. Em um cenário em que múltiplos nodos registram informações através de sensores e os transmitem para nodos de processamento, torna-se indesejável uma comunicação centralizada pois esta logo sofreria problemas no tocante à escalabilidade e apresentaria degradação de desempenho. Apesar disso, o ROS possui uma centralização no serviço de localização de recursos. Um dos nós, denominado *master*, é responsável por informar quais os recursos oferecidos, e quem é responsável pelo mesmo. Após essa etapa de descoberta de serviços, a comunicação independe do nó *master*.

A Figura 1 mostra como se dá o processo inicial de comunicação entre nós do ROS. O nó, intitulado "hokuyo", oferece um serviço intitulado "scan". Inicialmente esse nó, anuncia ao nó *master* que está oferecendo o serviço, e onde o serviço pode ser acessado. Um segundo nó, intitulado "viewer", pergunta ao *master* quem está oferecendo o serviço e recebe como resposta informações sobre o *host* e a porta onde o serviço pode ser acessado. Após, "viewer" entra em contato diretamente com "hokuyo" e negocia uma conexão direta que poderá ser feita por TCP conforme mostra a figura ou usando UDP. Dessa maneira o *master* só influencia no momento inicial da comunicação, a partir desse momento ele não exerce influência nas comunicações já estabelecidas.

A implementação do ROS busca ser compatível com várias linguagens de programação, permitindo que cada

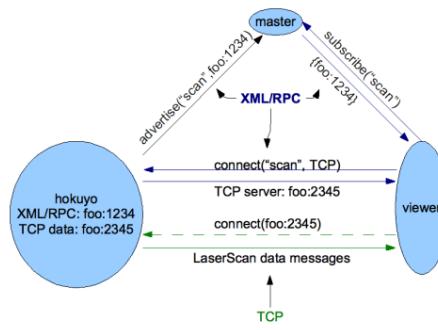


Figura 1. Comunicação no ROS

equipe possa escolher a linguagem mais conveniente. A comunicação inicial da configuração se dá através de XML-RPC, disponível para um grande número de linguagens. Além disso, também existe uma linguagem comum de definição de dados que é simples, porém permite composição através de elementos mais básicos.

Para lidar com a complexidade inerente do ROS, são usados pequenos módulos que se comunicam entre si para realizar tarefas maiores. Somente uma pequena parte da plataforma é implementada de forma centralizada. Além disso, o ROS encoraja o encapsulamento de código, de maneira a reutilizar funcionalidades e algoritmos entre diversos projetos. O próprio ROS reutiliza código de outros projetos, como [7] e [8].

### IV. CRITÉRIOS UTILIZADOS

O IETF sugere alguns critérios para projeto de aplicações no contexto de IoT.

- Instalação: De que maneira se dará a instalação dos dispositivos.
- Tamanho da rede: Principalmente no número de dispositivos do grupo.
- Alimentação dos nós: Difere se os nós vão estar ligados na alimentação tradicional ou através de baterias.
- Conectividade: Informa se os elementos da rede estarão conectados de forma permanente, eventual ou de forma periódica.
- Comunicação: Se a comunicação com o restante da rede será através de um ponto de conexão único ou de múltiplos elementos.
- Padrões de comunicação: Ponto para multiponto, multiponto para ponto, ponto à ponto, entre outras possibilidades.
- Nível de segurança: Qual o nível de segurança necessário na aplicação.
- Mobilidade: Se os nós serão móveis ou se permanecerão fixos.
- Qualidade de Serviço: Além das preocupações tradicionais pertinentes à Qualidade de Serviço, é necessário considerar as necessidades específicas dos grupos de dispositivos, de maneira a minimizar a comunicação e a utilização dos recursos computacionais. Além disso os requisitos podem mudar de acordo com o

modelo de entrega de dados.

Em [9] é apresentado o caso de um hospital onde é necessário manter sob controle a temperatura e a umidade de bolsas de sangue dentro de um hospital. Essa variável precisa ser acompanhada desde o momento da coleta até a sua posterior utilização através de sensores nas bolsas de sangue, no veículo de transporte das mesmas e nas salas onde elas são armazenadas. Referente à instalação, ela é feita de forma manual e pré-planejada. Embora possam haver mudanças no processo, elas não ocorrem com frequência e são planejadas previamente. O tamanho é considerado médio, e varia de acordo com o hospital analisado. A alimentação dos nós se daria na maior parte do tempo através de bateria, e a conectividade precisa ser permanente de forma a permitir acompanhar as informações dos sensores durante todo o processo. A comunicação precisaria se dar através de múltiplos elementos para diminuir a ocorrência de falhas. O padrão de comunicação poderia ser ponto à ponto no envio de sinais de controle dos atuadores de temperatura e umidade e múltiplos pontos à ponto para coletar informações de forma agregada. O nível de segurança é alto: as informações precisam obedecer os critérios de disponibilidade, integridade evitando danos aos pacientes. A confidencialidade pode ser um requisito se estiverem sendo transmitidas informações sobre a bolsa tais como doenças do paciente. Por fim a qualidade de serviço precisa ser levada em consideração, pois os sensores vão transmitir informação que vão garantir que os atuadores sejam ativados rapidamente sempre que necessário ou que alarmes sejam ativados para alertar sobre a necessidade de interferência humana.

## V. AVALIAÇÃO DA PLATAFORMA ROS

A partir dos critérios definidos anteriormente e, utilizando dois cenários hipotéticos, analisamos a viabilidade de utilizar o ROS plataforma base para aplicações de IoT.

### A. Cenário 1 - Proteção à saúde

Supondo o cenário de um paciente de hospital que é liberado para ir para casa, porém necessita ficar sob observação, mas mora sozinho. São utilizados sensores para monitorar a temperatura e frequência cardíaca do mesmo, e a informação obtida dos sensores é transmitida para o hospital onde a situação do paciente é avaliada periodicamente, e em função das variações apresentadas um alarme é disparado.

Com relação à instalação, os sensores teriam uma instalação feita de forma manual e planejada de acordo com a casa do paciente. A rede teria um tamanho pequeno se considerado um único paciente. A alimentação dos sensores se daria por baterias, e com exceção da troca das mesmas os sensores estariam permanentemente conectados. O ROS permitiria que o sensor ficasse desconectado para troca de baterias.

Dada a criticidade da aplicação seria recomendável que algum tipo de procedimento diferenciasse situações de falta de energia e falha na comunicação, mas isso poderia ser programado na aplicação. Possivelmente haverá

um ponto de conexão centralizador da conexão na casa, mas eventualmente poderia ser configurado uma ligação dupla de comunicação na qual os sensores enviariam as informações por múltiplas rotas de saída. Como o ROS permite múltiplas conexões simultâneas isso não seria um problema.

O ROS não oferece intrinsecamente soluções de segurança, especialmente de forma a garantir a integridade e autenticidade dos dados, pois poderia haver um terceiro elemento escutando as comunicações, ou até mesmo forjando dados dos sensores. Seria necessário embutir aspectos de segurança na API do ROS como por exemplo a capacidade de criptografar os dados enviados ou de verificar a identidade dos elementos envolvidos na comunicação.

A mobilidade não é um problema dentro da mesma rede. A partir do momento em que um nó mude de rede, ele perde a comunicação com os elementos da rede anterior. Caso fosse possível ter um ROS *master* em cada sub-rede eles poderiam sincronizar as listagens atualizadas de serviços e tópicos existentes entre os *masters*. Essa sincronização precisaria também ser implementada no ROS pois atualmente ele funciona através de um único *master*.

Por fim, atualmente o ROS não trabalha com aspectos de qualidade de serviço, sendo que cada nó é responsável por gerenciar a sua própria comunicação. Devido a isso, não seria possível oferecer QoS com as primitivas existentes atualmente, porém novas primitivas poderiam ser oferecidas levando em consideração estes aspectos. Dentro do contexto da aplicação poderia ser criada a figura de um nó concentrador, inexistente na arquitetura do ROS, responsável por compactar mensagens semelhantes, reduzindo a largura de banda necessária. Para que essa solução pudesse ser escalável seria necessário que os nós concentradores não fossem fixos, mas que elementos da rede pudessem assumir esse padrão de comportamento quando necessário.

### B. Cenário 2 - Monitoramento de agricultura

Em uma determinada plantação, uma quantidade grande de nós são instalados manualmente, onde cada um possui sensores diversos, como umidade, temperatura, condição do solo, luz do sol. Ao longo da plantação estão instalados nós com mais capacidade que agregam os dados do demais sensores.

Em relação à instalação, ela seria pré-planejada de acordo com a geografia do terreno. O tamanho da rede é grande devido à necessidade de sensores. A alimentação dos sensores seria feita através de bateria. Com relação à conectividade o ROS ofereceria pleno suporte no cenário, pois os sensores poderiam manter comunicação de dados entre eles. Como a comunicação não é feita de forma centralizada poderia haver múltiplos fluxos de informação simultâneos sem possuir um único gargalo de informação. O ROS permite comunicação entre múltiplos elementos, então os sensores poderiam enviar suas informações para múltiplos agregadores de conteúdo. A segurança não seria

tão essencial nessa aplicação. O fluxo de informação poderia ser enviado dos agregadores para os nós e o mesmo no sentido contrário. A mobilidade não seria levada em consideração pois os nós seriam fixos. A qualidade de serviço não seria relevante nessa aplicação para priorizar mensagens pois todas as informações seriam igualmente importantes, porém também nessa situação seria aconselhável na aplicação que houvesse uma compactação dos dados informados, pois provavelmente haveria grande redundância de informação dentro do grupo de sensores.

## VI. CONCLUSÕES E TRABALHOS RELACIONADOS

Existem algumas propostas de *middleware* para IoT. Em [10] e [11] são apresentadas propostas da criação de uma camada de software utilizando uma arquitetura orientada a serviços. O artigo [12] utiliza o ROS como *middleware* na criação de alguns cenários de IoT, porém diferente do presente artigo não são analisados quais aspectos do ROS poderiam ser melhorados para esse fim. Outro trabalho é [13] que elenca os principais desafios no desenvolvimento de um software desse tipo.

Com relação à instalação não existem problemas inerentes de utilização do ROS, o mesmo pode ser dito da alimentação dos nós que independe do ROS. A plataforma permite a comunicação diretamente entre os nós, com isso maiores tamanhos de rede são suportados. O gargalo existe no estabelecimento das comunicações que depende do nó *master*. O ROS suporta conectividade limitada, pois quando um dos nós fica temporariamente desconectado, quando a conexão é recuperada, as aplicações que estavam se comunicando continuam a funcionar. O *framework* em si não facilita a comunicação com múltiplos nós, mas permite que elas sejam estabelecidas em nível de aplicação. Também não são oferecidas primitivas que facilitem a segurança ou a qualidade de serviço. A mobilidade é tratada com sucesso desde que a comunicação entre os nós seja possível.

A qualidade de serviço pode ser implementada em nível de aplicação quando for necessário. A segurança pode ser implementada estendendo a API adicionando principalmente funcionalidades para garantir autenticidade, que poderia se dar através de troca de chaves privadas. No quesito segurança outra alternativa seria integrar o IPsec[14] dentro do ROS. Com isso a criptografia pode ser implementada nas primitivas de envio e recebimento de mensagens. Por fim é importante resolver a pendência do *master*, de forma que possam existir múltiplos *masters* que se comuniquem entre si, evitando que existam pontos que podem tornar o sistema lento, especialmente se tratando do alto número de nós previsto no ambiente de IoT. Adicionando-se as características como abstração de hardware, comunicação *peer-to-peer*, interface com diversas linguagens de programação, organização modular do código tornam o ROS como uma alternativa bastante atrativa no desenvolvimento de um *middleware* para IoT.

## REFERÊNCIAS

- [1] R. MacManus, “AT&T & Cisco Talk Up Internet of Things,” [http://www.readwriteweb.com/archives/verizon\\_att\\_cisco\\_internet\\_of\\_things.php](http://www.readwriteweb.com/archives/verizon_att_cisco_internet_of_things.php), 2010, ultimo acesso em 5 de março de 2012.
- [2] M. e. a. Vermesan, O. e Harrison, “The Internet of Things - Strategic Research Roadmap,” 2009, cluster of European Research Projects on the Internet of Things, CERP-IoT.
- [3] U. N. I. Council, “Disruptive Civil Technologies – Six Technologies with Potential Impacts on US Interests Out to 2025,” [http://www.dni.gov/nic/confreports\\_disruptive\\_tech.html](http://www.dni.gov/nic/confreports_disruptive_tech.html), 2008, conference Report 2008-07. Último acesso em 7 de março de 2012.
- [4] I. over Low power WPAN Group, “IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals RFC 4919,” <http://datatracker.ietf.org/doc/rfc4919/>, 2007, Último acesso em 20 de julho de 2012.
- [5] C. Mello, E. Gonçalves, E. Estrada, G. Oliveira, H. Souto, R. Almeida, S. Botelho, T. Santos, and V. Oliveira, “Tatubot–robotic system for inspection of undergrounded cable system,” in *Robotic Symposium, 2008. LARS'08. IEEE Latin American*. IEEE, 2008, pp. 170–175.
- [6] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, “Ros: an open-source robot operating system,” in *ICRA Workshop on Open Source Software*, vol. 3, 2009.
- [7] O. Project, “Open Source Computer Vision Library,” Último acesso em 20 de julho de 2012.
- [8] R. e. a. Gerkey, B. e Hedges, “The Player Project,” Último acesso em 20 de julho de 2012.
- [9] I. over Low power WPAN Group, “Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs) RFC 6568,” Último acesso em 20 de julho de 2012.
- [10] E. Kosmatos, N. Tsakalis, and A. Boucouvalas, “Integrating rfids and smart objects into a unified internet of things architecture,” *Advances in Internet of Things*, vol. 1, no. 1, pp. 5–12, 2011.
- [11] M. Díaz, D. Garrido, and A. Reyna, “One step closer to the internet of things: Smepp.”
- [12] L. Roalter, M. Kranz, and A. Möller, “A middleware for intelligent environments and the internet of things,” *Ubiquitous Intelligence and Computing*, pp. 267–281, 2010.
- [13] M. Chaqfeh and N. Mohamed, “Challenges in middleware solutions for the internet of things,” in *Collaboration Technologies and Systems (CTS), 2012 International Conference on*. IEEE, 2012, pp. 21–26.
- [14] J. Arkko, V. Devarapalli, and F. Dupont, “Using ipsec to protect mobile ipv6 signaling between mobile nodes and home agents,” 2004.