



11<sup>a</sup> ESCOLA REGIONAL DE REDES DE COMPUTADORES  
6–8 de novembro de 2013  
Porto Alegre – RS

## ANAIIS

**Editora**  
Sociedade Brasileira de Computação – SBC

**Organizadores**  
Tiago Ferreto  
Cristiano Bonato Both  
Juliano Wickboldt

**Realização**  
Pontifícia Universidade Católica do Rio Grande do Sul

**Promoção**  
Sociedade Brasileira de Computação – SBC

Copyright © 2013 Sociedade Brasileira de Computação  
Capa: Endrigo Conte e Israel Campos de Oliveira  
Supervisão Gráfica: Tiago Ferreto  
Impressão: Gráfica Epecê

#### CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Escola Regional de Redes de Computadores (11.: 6–8 nov 2013:  
Porto Alegre)

Anais / Organizadores: Tiago Ferreto, Cristiano Bonato Both, Ju-  
liano Wickboldt. — Porto Alegre: Gráfica Epecê, 2013.

130 f.: il.

ISSN 2237-3748

Conhecido também como ERRC 2013.

1. Redes de Computadores. 2. Sistemas Distribuídos. I. ERRC  
(11.: 6–8 nov 2013: Porto Alegre). II. **PUCRS**. III. Ferreto, Tiago.  
IV. Both, Cristiano Bonato. V. Wickboldt, Juliano. VI. Título.

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

# **ERRC 2013**

<http://www.inf.pucrs.br/errc2013>

## **Comitê de Programa**

Adenauer Yamin (UCPEL/UFPEL)  
Alberto Schaeffer-Filho (UFRGS)  
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)  
César De Rose (PUCRS)  
Clarissa Marquezan (Duisburg-Essen University)  
Cristiano Bonato Both (UFRGS)  
Cristiano Costa (UNISINOS)  
Cristina Nunes (PUCRS)  
Diego Kreutz (Faculty of Science of University of Lisbon)  
Eduardo Monks (FATEC SENAC - Pelotas)  
Erico Amaral (IFRS/UFRGS)  
Erico Rocha (UNISINOS)  
Ewerton Salvador (UFMG)  
Flávio Roberto Santos (UFRGS)  
Iara Augustin (UFSM)  
Jéferson Nobre (UFRGS)  
José Jair Cardoso de Santanna (University of Twente)  
Juliano Wickboldt (UFRGS)  
Leonardo Pinho (UNIPAMPA)  
Lisandro Zambenedetti Granville (UFRGS)  
Luciano Paschoal Gaspary (UFRGS)  
Lucio Prade (Unisinos)  
Marcelo Marotta (UFRGS)  
Marco Trentin (UPF)  
Marinho Barcellos (UFRGS)  
Maurício Lima Pilla (UFPEL)  
Oscar Caicedo (UFRGS)  
Rafael Kunst (UFRGS)  
Rafael Pereira Esteves (UFRGS)  
Raul Ceretta Nunes (UFSM)  
Renata Reiser (UFPEL)  
Ricardo Luis dos Santos (UFRGS)  
Ricardo Neisse (IPSC - Joint Research Center (JRC))  
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)  
Weverton Luis da Costa Cordeiro (UFRGS)

## **Comitê Organizador**

### **Coordenação Geral**

Prof. Dr. Tiago Ferreto (PUCRS)

### **Organização Local**

Prof. Dr. César De Rose (PUCRS)

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

Prof. Dr. Cristiano Bonato Both (UNISC)

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

Prof. MSc. Juliano Wickboldt (UFRGS)

### **Comissão de Organização**

Paolo Cemim (PPGCC - PUCRS)

Luis Jersak (PPGCC - PUCRS)

Marcelo Conterato (PPGCC - PUCRS)

Endrigo Conte (PPGCC - PUCRS)

Fábio Diniz Rossi (PPGCC - PUCRS)

Miguel Gomes Xavier (PPGCC - PUCRS)

Israel Campos de Oliveira (PPGCC - PUCRS)



# Apresentação

Com grande satisfação apresentamos a 11<sup>a</sup> Escola Regional de Redes de Computadores (ERRC 2013). Neste ano o evento é organizado pela Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS) e promovido pela Sociedade Brasileira de Computação (SBC), ocorrendo de 6 a 8 de novembro, em Porto Alegre, 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 são as sessões técnicas, onde são apresentados trabalhos de Iniciação Científica e de Pós-Graduação. Os artigos são inicialmente revisados em um processo onde pelo menos três revisores avaliam 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, um total de 14 artigos de Iniciação Científica e Pós-Graduação foram aceitos para publicação e apresentação, de um total de 26 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. As palestras abordam tópicos bastante atuais e relevantes para área, tais como: Computação em Nuvem, Gerência de Redes, SDN (Software Defined Networks), IPv6, entre outros.

A escola deste ano conta com 3 minicursos, detalhados abaixo:

**Introdução à redes sem fio inteligentes (Prof. Dr. Cristiano Bonato Both - UFRGS)** O minicurso tem os seguintes objetivos: Introdução sobre a regulamentação brasileira em Rádio Frequência; Apresentação sobre os conceitos básicos de Rádio Cognitivo; Demonstrações práticas sobre GNU RADIO no protótipo USRP2; Apresentação sobre o estado da arte em Rádio Cognitivo; Introdução sobre coexistência entre tecnologias nas frequências não licenciadas.

**Introdução ao Protocolo IPv6 (Edwin Cordeiro - NIC.br)** Introdução teórica e prática ao protocolo IPv6, que precisa ser urgentemente implementado, já que os endereços IPv4 se esgotam até a metade de 2014. Serão apresentadas as diferenças entre o IPv6 e o IPv4 e haverá exercícios práticos para uma primeira experiência com o protocolo. Os exercícios práticos necessitam que o interessado leve seu notebook com o software Virtualbox instalado.

**Separação de Planos com OpenFlow: Teoria e Prática (Pedro Duarte e Ricardo dos Santos - DATACOM)**  
Muitos argumentam que as aplicações distribuídas atuais estão sofrendo devido a ossificação do conjunto de protocolos de núcleo de rede. Dentre outros problemas, por hora mostra-se difícil e muitas vezes inviável a implantação de protocolos de escalonamento de requisições quando os nós servidores encontram-se atrás de domínios administrativo distintos (e.g., aplicações instaladas em duas nuvens diferentes). O OpenFlow caracteriza-se como uma abordagem inovadora para transpor essas barreiras. Ele emprega uma estratégia de separação de planos para gestão de

fluxos de dados através de um conjunto padronizado de primitivas que operam sobre esses fluxos. Dessa forma, é possível implementar soluções inteligentes e interoperáveis, onde as decisões tem como base o estado geral da rede. Esse hands-on tem por objetivo apresentar uma discussão introdutória ao OpenFlow, apresentar o controlador FloodLight e realizar práticas laboratoriais com essa última ferramenta. Além disso, serão apresentadas ferramentas auxiliares para gerência de redes OpenFlow.

Gostaríamos de agradecer a todos que contribuíram para a realização desta escola, incluindo colegas professores, funcionários, orientandos e alunos. Também agradecemos as instituições UNISC e UFRGS que apoiaram o evento, e as empresas HP e DATACOM, e o NIC.br que acreditaram na importância da escola através do seu patrocínio. Finalmente, agradecemos à PUCRS e à SBC pelo apoio na concretização do evento.

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

Tiago Ferreto  
Cristiano Bonato Both  
Juliano Wickboldt

Porto Alegre, novembro de 2013.

# Sumário

<b>I Sessão 1 - Trabalhos de IC</b>	1
<b>Computação em Nuvem: Análise Comparativa de Ferramentas Open Source para IaaS</b>	
Eduardo Hentges, Bruna Thomé, Dalvan Griebler .....	3
<b>Elasticidade Automática de Recursos para Aplicações de Alto Desempenho em Ambientes de Computação em Nuvem</b>	
Rodrigo Righi, Cristiano Costa, Vinícius Rodrigues .....	7
<b>Implementação do Protocolo PROFIsafe para o Desenvolvimento de Sistemas Seguros</b>	
William Vidal, Rodrigo Dobler, Sérgio Luis Cechin, Taisy Weber, João Netto .....	11
<b>Engenharia social: explorando o efeito halo para obter acesso físico não autorizado em empresas de cartões de pagamento</b>	
Guilherme Gattino, Jéferson Nobre .....	15
<b>II Sessão 2 - Trabalhos de IC</b>	19
<b>Modulação e codificação adaptativos através de software defined radio</b>	
Matheus Eidt, Roberto Mainardi Schmidt, Lauro Tremea Culau, Rafael Nicolay, Matias Schimuneck, Maicon Kist .....	21
<b>Sensoriamento espectral por detecção de energia utilizando aprendizagem por reforço</b>	
Rafael Nicolay, Matias Schimuneck, Matheus Eidt, Lauro Tremea Culau, Roberto Mainardi Schmidt, Cristiano Both, Maicon Kist .....	25
<b>Algoritmo de decisão espectral usando teoria dos jogos para rádios cognitivos</b>	
Roberto Mainardi Schmidt, Lauro Tremea Culau, Rafael Nicolay, Matheus Eidt, Matias Schimuneck, Cristiano Both, Leonardo Faganello, Maicon Kist .....	29
<b>III Sessão 3 - Trabalhos de IC</b>	33
<b>Gerenciamento da coleta de lixo urbano utilizando Zigbee</b>	
Heitor Neto, Humberto Machry Prado, Douglas Giacomini .....	35
<b>Aplicação de Algoritmos de Escalonamento de Processos para Gerenciamento de Intersecções em VANETs</b>	
Thiago Lopes Trugillo da Silveira, Marcia Pasin .....	39
<b>SNMP Mobile: Uso de Plataforma Móvel para Gerenciamento de Rede Através do Protocolo SNMP</b>	
Caio Lucena, Valter Barbosa .....	43
<b>Utilizando a plataforma Arduino para a comunicação entre dispositivos embarcados e redes TCP/IP</b>	

Alexandre Silva Rodrigues, Tiago Antônio Rizzetti .....	47
---	----

## IV Sessão 4 - Trabalhos de PG 51

<b>Correlação de Alertas Utilizando CBR em um Internet Early Warning System</b> Tarcisio Ceolin Junior, Osmar Marchi dos Santos, Giani Petri, Raul Ceretta Nunes, Luis Silva .....	53
--	----

<b>An Example for Performance Prediction for Map Reduce Applications in Cloud Environments</b> Ivan Carrera, Fabricio Scariot, Pierre Turin, Claudio Geyer .....	57
---	----

<b>Avaliação do suporte à simulação de redes OpenFlow no NS-3</b> Marcelo Conterato, Israel de Oliveira, Tiago Ferreto, César A. F. De Rose .....	61
--	----

---

I

## **Sessão 1 - Trabalhos de IC**

---



# Computação em Nuvem: Análise Comparativa de Ferramentas Open Source para IaaS

Bruna Thomé, Eduardo Hentges

Curso Superior de Tecnologia em Redes de Computadores  
Faculdade Três de Maio (SETREM)  
Três de Maio – RS – Brasil  
{thome.bru, eduhentges}@gmail.com

Dalvan Griebler

Programa de Pós-Graduação em Ciência da Computação  
Pontifícia Universidade Católica do Rio Grande do Sul  
Porto Alegre – RS – Brasil  
dalvan.griebler@acad.pucrs.br

**Abstract**— Este artigo tem por objetivo estudar, apresentar e comparar as principais ferramentas open source de computação em nuvem. O conceito de computação em nuvem está cada vez mais presente nas redes de computadores. A dificuldade não está apenas em implantar uma nuvem, mas também em escolher a ferramenta mais apropriada. Assim, este trabalho buscou estudar as seguintes ferramentas: Eucalyptus, OpenNebula, OpenQRM, OpenStack, CloudStack Ubuntu Enterprise Cloud, Abiquo, Convirt, Apache Virtual Lab e Nimbus. Para estas, foram consideradas as características, funcionalidades e formas de operação, evidenciando o cenário mais indicado para cada uma delas.

**Keywords**— *Computação em Nuvem, Ferramentas Open Source, Modelo IaaS*;

## I. INTRODUÇÃO

A Computação em nuvem (CN) possibilita acessar recursos computacionais (por exemplo, servidores, armazenamento, redes, serviços e aplicações) de maneira prática e sob demanda, rapidamente e que podem ser liberados para o usuário sem qualquer envolvimento gerencial. [1]. Isso pode ser muito importante para agilizar o desenvolvimento do trabalho, reduzir custos, facilitar o emprego de recursos de alto processamento, evitar gastos com manutenção e licenças de software.

As nuvens podem ser caracterizadas em diferentes tipos (pública, privada e híbrida) e diferentes modelos de serviços (IaaS - *Infrastructure as a Service*, PaaS - *Platform as a Service* e SaaS - *Software as a Service*) [2,3]. Neste trabalho, o escopo são as ferramentas open source para administração de nuvem que suportam o modelo IaaS.

A dificuldade não está somente em implantar uma nuvem, mas também em escolher a ferramenta mais apropriada para o projeto de redes. Neste artigo, o objetivo é caracterizar, estudar e comparar as principais ferramentas, evidenciando o cenário mais indicado para cada uma delas. Para isso, o artigo apresenta inicialmente os trabalhos relacionados na Seção II. Na Seção III são estudadas as ferramentas de CN e na Seção IV é efetuada uma análise comparativa das ferramentas.

## II. TRABALHOS RELACIONADOS

Nesta seção o objetivo é expor alguns trabalhos que apresentam uma relação a este e o que há de diferente em comparação ao que já existe na literatura. Alguns deles fazem um comparativo de características. No entanto, outros procuram implantar uma ou duas ferramentas a fim de comparar as funcionalidades em um cenário específico.

No trabalho [4], não é criado nenhum ambiente de teste e apenas são estudadas as características e a arquitetura das seguintes ferramentas: Xen *Cloud Platform*, Nimbus, OpenNebula, Eucalyptus, TPlataform, Apache *Virtual Computing Lab* e Enomaly *Platform Computing Elastic*. A análise comparativa é feita através de uma tabela, onde é descrita a ferramenta, o modelo de serviço, suas principais características e exemplos de quem as utiliza. Os autores concluíram que existe a necessidade de padronização das plataformas atuais, referente à interface, negociação, acesso por meio de *Web Services*. Isso por que as nuvens têm diferentes níveis de abstração.

O trabalho [5] trata da comparação de ferramentas, com o objetivo de descobrir se os usuários necessitavam de mais ferramentas de acesso. Caso necessário, verificar qual delas deveria ser utilizada. Para isso, os autores analisaram a comunidade de usuários da empresa FutureGrid e assim fizeram seus registros. As ferramentas escolhidas como parte do processo de aplicação do projeto foram Nimbus e Eucalyptus. Os autores concluíram que são fornecidas evidências de que existe a oferta de muitas ferramentas e que é necessário o usuário dizer qual é a melhor para ele.

Em [6], são feitos testes utilizando a ferramenta *Open Cirrus*. Para efetuar a avaliação de desempenho, foram utilizados o PlanetLab e o Emulab, para simular a utilização de usuários distribuídos e a utilização de aplicativos em nuvem. O resultado da pesquisa mostrou que o desempenho da transferência de dados em uma nuvem pode variar, dependendo de quantos usuários diferentes estão utilizando o mesmo serviço. Além disso, as variações podem ser atribuídas as características de rede entre a nuvem e usuários. Sendo assim, a distância entre a nuvem e os usuários é de grande determinação para o desempenho.

No trabalho [7], foram utilizadas as ferramentas OpenQRM e Eucalyptus com o intuito de verificar qual delas é a melhor. Para a realização dos testes é utilizado um ambiente isolado de 6 computadores *desktop*, com o sistema operacional Ubuntu 10.04 e uma rede com acesso à *Internet*. Como testes, foram realizadas tarefas em cada um dos componentes das ferramentas individualmente, no qual se executavam tarefas como envio de pacotes ICMP e transferência de arquivos e era realizada alguma falha proposital para ver o resultado. Assim, os melhores resultados foram do OpenQRM.

A pesquisa de [8] afirma que existe uma grande quantidade de características que devem ser levados em consideração para avaliar a CN. Assim, uma série de ferramentas são consideradas como: OpenNebula, Eucalyptus, Ubuntu *Enterprise Cloud*, OpenQRM, Abiquo, Red Hat *Cloud*

*Foundations*, Edition One, OpenStack, Nimbus, mOSAIC. As características são agrupadas em: armazenamento, virtualização, gestão, rede, segurança e apoio. O resultado é que com base nas características pode ser efetuada a escolha da ferramenta mais apropriada, a que se adeque as necessidades da organização.

Os trabalhos [4], [5] e [8] apresentam objetivos em comum a esta pesquisa, o de comparar ferramentas de CN *open source*. No entanto, o presente trabalho compara um conjunto de características maior e traz uma atualização da situação atual das ferramentas que já foram estudadas nos outros trabalhos. Em relação aos trabalhos [4] e [5] são estudadas seis ferramentas diferentes e no trabalho [8], apenas três diferentes são estudadas. Porém o conjunto de características comparadas no presente trabalho é bem mais amplo, contribuindo também com a comparação de: interface, gerenciamento de energia, balanceamento de carga, integração, segurança e monitoramento. Isso deixa claro, que embora exista uma semelhança, várias contribuições podem vir a surgir através de uma comparação mais ampla e detalhada.

As próximas direções desta pesquisa se encaminham no sentido dos trabalhos [6] e [7], pois no futuro, a ideia é que estas ferramentas também sejam avaliadas em um ambiente controlado, sendo possível identificar o comportamento delas e verificar se são coerentes com o que a literatura nos apresenta.

### III. FERRAMENTAS DE COMPUTAÇÃO EM NUVEM

Nesta seção serão apresentadas as ferramentas *Open Source* de computação em nuvem para o modelo de serviço IaaS. Foram selecionadas ferramentas que oferecessem este modelo de serviço.

#### A. Eucalyptus

O Eucalyptus é indicado para computação em nuvem em ambientes de computação empresarial corporativa, pois possibilita oferecer aos usuários acesso as ferramentas utilizadas pela empresa. Tendo como arquitetura cinco componentes (*Cloud Controller*, *Walrus*, *Cluster Controller*, *Storage Controller* e *Node Controller*) básicos, responsáveis pelo seu funcionamento. Além disso, possibilita o uso de diferentes servidores para implantar os componentes e facilitar a configuração [9].

#### B. OpenNebula

O OpenNebula foi desenvolvido para uma gestão mais eficiente e escalável de máquinas virtuais em infraestruturas distribuídas. Suas características são voltadas para atender aos requisitos de empresas que utilizavam a ferramenta em versões anteriores. Sua arquitetura é composta por um *host* responsável pela administração da nuvem e os outros *hosts* são responsáveis pela virtualização das máquinas virtuais [10].

#### C. OpenStack

OpenStack permite criar nuvens públicas e privadas. Através de uma interface, o administrador pode gerenciar a capacidade de computação, armazenamento e recursos de rede presentes no *datacenter*. Entre seus componentes estão o OpenStack *Compute* (*Nova*), OpenStack *Object Storage* (*Swift*), OpenStack *Image Service* (*Glance*), Painel de ferramentas (*Horizon*), Rede (*Quantum*), *Storage Block* (*Cinder*) e Identificação (*Keystone*) [5,11].

#### D. CloudStack

O CloudStack foi desenvolvido para implantar e gerenciar grandes redes de máquinas virtuais, pois possui escalabilidade e alta disponibilidade de infraestrutura. Permite a criação de nuvens privadas, híbridas e públicas que podem fornecer infraestrutura como um serviço para os usuários. A arquitetura é composta pelo armazenamento primário, o *cluster*, Pod (grupo de *clusters*) e armazenamento secundário [12].

#### E. OpenQRM

OpenQRM é uma ferramenta que gerencia virtualização, armazenamento, a rede e toda a infraestrutura de TI a partir de um console. Permite criação de nuvens privadas com alta disponibilidade e também funciona de maneira gerente-agente. Para isso, o controlador da nuvem é o gerente e os recursos que são integrados a ele são os agentes. Neste caso, a estrutura apresenta o gerente, o *storage* e o nós (recursos) [13,7].

#### F. Ubuntu Enterprise Cloud

O Ubuntu *Enterprise Cloud* é baseado na ferramenta Eucalyptus. Devido a isso, apresenta os mesmos componentes (*Cloud Controller*, *Walrus*, *Cluster Controller*, *Storage Controller* e *Node Controller*). Também permite criar um perfil de instalação mínima para gerenciar as máquinas físicas e as virtuais, além de monitorar os componentes da nuvem [14].

#### G. Abiquo

Abiquo visa criar nuvens privadas baseadas em uma infraestrutura já existente ou controlar o uso de serviços em nuvem pública. Fornece *logs* para analisar o que e para que estão sendo utilizados os recursos e possui um mecanismo de preços que atribui um valor a qualquer recurso (CPU, RAM, armazenamento). O Abiquo é constituído pelo Gerenciador de rede, *cluster*, servidor Abiquo, rede de armazenamento, Abiquo serviços remotos e um servidor de armazenamento [15].

#### H. Convirt

O ConVirt possibilita centralizar o gerenciamento através de *datacenters* virtuais. Ele também é capaz de monitorar os recursos do servidor e dos clientes da máquina virtual, possibilitando o controle da carga exercida sobre o servidor. A sua arquitetura é composta pelo *Datacenter-wide*, *Universal web Access* e *Agent-less* [16].

#### I. Apache Virtual Computing Lab (Apache VCL)

Apache VCL oferece como ambiente uma máquina virtual ou até mesmo um *cluster* de servidores físicos. É utilizado para acesso remoto a partir da *internet* de maneira dinâmica, utilizando-se de reservas de recursos computacionais. Ele tem sua arquitetura formada por portal *web*, banco de dados, nós de gestão e nós de computação [17].

#### J. Nimbus

O Nimbus possibilita a construção de nuvens privadas, implantando *clusters* virtuais autoconfiguráveis. Sua arquitetura é composta pelo: *workspace* de serviços (que permite ao cliente implantar e gerenciar grupos definidos de VMs) gerenciador de recursos, (que realiza a implantação de contratos de locação de VM), *workspace pilot* (se estende a gestores de recursos locais), *IaaS gateway* (permite que um cliente utilize outra infraestrutura como serviço) e *workspace client* (fornecendo a funcionalidade total do serviço) [18].

#### IV. ANÁLISE COMPARATIVA

Nesta seção é realizada uma análise comparativa das ferramentas estudadas na Seção III. Esta comparação é feita usando tabelas que elencam as características de: Interface, Gerenciamento de energia, Balanceamento de carga, Rede, Armazenamento, Monitoramento, Integração, Virtualização, Segurança, Escalabilidade e Tolerância a falhas.

Na Tabela I são comparadas as características de Interface, Gerenciamento de energia e Balanceamento. A interface de acesso pode ser de duas formas: através de SSH (*Secure Shell*) que é uma conexão segura entre o cliente e o servidor ou através de uma página web, por HTTP (*Hyper-text Transfer Protocol*). O gerenciamento de energia tem o objetivo de reduzir os custos com energia elétrica. O Eucalyptus realiza isso através da suspensão das máquinas que não estão em uso. A mesma coisa é realizada pelo OpenNebula utilizando o sistema CLUES (*Cluster Energy Saving*) e pelo UEC através do UEC Power Management. Já o OpenStack oferece extensões que somente funcionam junto a processadores Intel Xeon. O CloudStack e Apache VCL colocam os hosts e recursos em modo *standby* quando não estão sendo usados. O OpenQRM busca por recursos não utilizados ou em baixo uso. E o Nimbus move máquinas virtuais para outros servidores. O Abiquo e o Convirt não apresentam esta característica em sua descrição.

TABELA I. COMPARAÇÃO DAS FERRAMENTAS I.

Ferramenta	Interface	Gerenciamento de Energia	Balanceamento de carga
Eucalyptus	SSH e WEB	Possui	<i>Elastic Load Balancer</i>
OpenNebula	SSH e WEB (Solução GUI)	CLUES	Possui
OpenStack	SSH e WEB (Horizon)	Power Management	<i>Quantum Network Load Balancing</i>
Cloud Stack	SSH e WEB	Possui	<i>Citrix NetScaler</i>
OpenQRM	SSH e WEB	Possui	Possui
UEC	SSH e WEB	UEC Power Management	Não
Abiquo	SSH e WEB	Não	Sim
Convirt	SSH e WEB	Não	Não
Nimbus	SSH e WEB	Possui	Não
Apache VCL	SSH e WEB	Possui	Não

O balanceamento de carga é uma maneira eficiente de fazer a divisão das tarefas e melhor aproveitar os recursos computacionais. O Eucalyptus utiliza o *Elastic Load Balancer* que distribui automaticamente o tráfego de entrada das aplicações entre os nós do Cluster. No OpenNebula as máquinas virtuais em execução, são divididas entre os nós operantes na nuvem. O OpenStack, utiliza o *Quantum Network Load Balancing* para dividir a carga de processos entre os nós. O CloudStack utiliza-se do Citrix NetScaler para dividir as tarefas entre os nós e o OpenQRM faz o balanceamento dos recursos do cluster para execução dos processos. Já o Abiquo divide entre os nós as conexões e as demais ferramentas não possuíam informações na literatura sobre esta característica.

TABELA II. COMPARAÇÃO DAS FERRAMENTAS II

Ferramenta	Rede	Armazenamento	Monitoramento
Eucalyptus	Bridge e VLAN	AoE, iSCSI e NFS	Nagios
OpenNebula	Bridge, VLAN e Open Vswitch	NFS, iSCSI, LVM	OpenNebula Sunstone
OpenStack	VLAN e Open Vswitch	AoE, iSCSI e NFS	OpenStack Clamavi
Cloud Stack	VLAN	iSCSI e NFS	<i>Traffic Sentinel</i>
OpenQRM	Bridge e VLAN	NFS, iSCSI, AoE e LVM	openqrm-monitorord
UEC	Bridge e VLAN	iSCSI e AoE	UEC Monitor
Abiquo	VLAN	NFS, iSCSI, LVM	Abiquo Monitor
Convirt	VLAN	NFS, iSCSI e LVM	Convirt Monitor
Nimbus	VLAN	AoE, iSCSI e NFS	Nagios
Apache VCL	VLAN	iSCSI	Não

A Tabela II compara as funcionalidades de Rede, Armazenamento e Monitoramento. Na funcionalidade de rede são demonstrados os métodos de conexão utilizados entre os componentes de uma nuvem. Podem ser citadas a VLAN (permite dividir uma rede física em diversas redes lógicas), o Bridge (permite conectar duas ou mais redes distintas) e o Open vSwitch (cria um switch virtual que encaminha o tráfego de máquinas virtuais dentro de um mesmo host).

O Armazenamento pode ser realizado utilizando de diferentes formas. Como por exemplo, o iSCSI (*Internet Small Computer System Interface*) que é um protocolo de transporte de comandos SCSI, é usado onde dados são armazenados em diversos hosts de uma rede. O AoE (ATA Over Ethernet) é um protocolo de rede para acesso a dispositivos de armazenamento SATA através da rede. O NFS (*Network File System*) é um sistema de arquivos em que diretórios são compartilhados entre os computadores de uma rede. E o LVM (*Logical Volume Management*) é usado para criar um grande disco virtual que pode conter mais de um dispositivo de armazenamento.

Eucalyptus e Nimbus utilizam o Nagios para monitorar recursos como CPU, memória, HD e VMs. No entanto, as demais ferramentas se utilizam de sistema próprio, com exceção o Apache VCL que não possui relato sobre isso.

A Tabela III mostra a comparação da Integração, Virtualização e Segurança. Na Integração são descritas se as ferramentas possuem integração com algum outro serviço. As ferramentas Apache VCL e Abiquo são as únicas que não permitem integração com a Amazon. O CloudStack também permite integração com o CloudBridge, uma plataforma integrada que conecta aplicativos e melhora a utilização da largura de banda em nuvem pública e redes privadas. O OpenQRM permite integração também com as ferramentas UEC e Eucalyptus. O Abiquo possibilita integração com Cisco UCS, o que facilita a mudança para o modelo de serviço IaaS. Já o Nimbus também permite integração com o Cumulus, um sistema de armazenamento em nuvem.

TABELA III. COMPARAÇÃO DAS FERRAMENTAS III

Ferramenta	Integração	Virtualização	Segurança
Eucalyptus	EC2, EBS, AMI, S3, IAM	Xen, KVM e Vmware ESXi	Autenticação, CUG e Active Directory
OpenNebula	EC2	Xen, KVM e Vmware	Autenticação e CUG
OpenStack	EC2 e S3	XenServer, KVM e Hyper-V	Keystone, LDAP, e métodos externos
Cloud Stack	CloudBridge e EC2	Xen, KVM e Vmware ESXi	Autenticação e CUG
OpenQRM	UEC, EC2 e Eucalyptus	Vmware ESX, Xen, KVM e XenServer	Autenticação, CUG e LDAP
UEC	EC2	KVM	Autenticação e CUG
Abiquo	Cisco UCS	VMware ESXi, Hyper-V, XenServer, Xen, KVM	Autenticação, CUG e LDAP
Convirt	EC2	Xen e KVM	Não
Nimbus	EC2, S3, Cumulus	Xen e KVM	Autenticação e CUG
Apache VCL	Não	Vmware, KVM	Autenticação LDAP

As únicas ferramentas que não oferecem suporte a virtualização Xen (tecnologia que pode ser atrelada diretamente ao hardware) são Apache VCL e o UEC. O Eucalyptus e o CloudStack também podem oferecer virtualização com VMware ESXi que não necessita de sistema operacional e pode ser integrado diretamente aos servidores. O OpenNebula e o Apache VCL utilizam o VMware que permite a instalação de um sistema operacional dentro de outro em execução simultânea. Já o OpenStack, OpenQRM e Abiquo utilizam o Xen Server, possibilitando que várias máquinas virtuais rodem em uma máquina física. OpenStack e Abiquo podem utilizar

também o Hyper-V, que fornece infraestrutura de *software* e ferramentas para gerenciar ambientes de virtualização de servidores.

Em relação a segurança, a maioria das ferramentas permitem autenticação e Controle por Usuários e Grupos (CUG). A ferramenta Eucalyptus também possibilita a integração com *Active Directory* (AD). O OpenStack, OpenQRM, Abiquo e o Apache VCL permitem usar autenticação por LDAP, que assim como o AD, realiza autenticação dos usuários. O OpenStack ainda pode utilizar do componente *Keystone*, utilizado para autenticação.

Se uma das necessidades da nuvem é o gerenciamento de energia, as ferramentas Abiquo e Convirt não são indicadas, pois as mesmas não realizam tal ação. Se a prioridade for o balanceamento de carga, dentre as ferramentas avaliadas somente o Eucalyptus, OpenNebula, OpenStack, CloudStack, OpenQRM e Abiquo realizam esta tarefa, logo, estas são mais indicadas. Em relação à rede, se houver a necessidade de conectar diferentes redes, são indicadas as ferramentas Eucalyptus, OpenNebula, OpenQRM e Ubuntu Enterprise Cloud, por possuírem o método de conexão *bridge* e VLAN. Se for preciso uma maior variedade de métodos de armazenamento, são indicados o Eucalyptus, OpenNebula, OpenStack, OpenQRM, Abiquo, Convirt e Nimbus.

Quando é indispensável o monitoramento tanto do *hardware* que compõem a nuvem, quanto das máquinas virtuais que operam nela, não é indicada a utilização do Apache VCL, pois este não apresenta nenhuma alternativa de monitoramento, diferente das outras ferramentas. Embora todas as ferramentas citadas na Tabela III ofereçam integração, exceto o Apache VCL. Ao se separar com um cenário em que se tem como prioridade a integração com diferentes nuvens, as ferramentas Eucalyptus, Nimbus e OpenQRM parecem ser mais apropriadas, por possuírem mais opções de integração.

Ao se separar com um ambiente em que existe grande heterogeneidade arquitetural, ter diversas ferramentas de virtualização torna-se uma vantagem. Nestes casos, é mais indicado a utilização das ferramentas OpenQRM e Abiquo, por possibilitarem a utilização de uma grande quantidade de virtualizadores. Em relação à segurança, se o desejo é utilizar uma base de dados externa de autenticação, são indicados o Eucalyptus, OpenStack, OpenQRM, Abiquo e Apache VCL, por possibilitarem autenticação LDAP ou integração com o *Active Directory*.

## V. CONCLUSÃO

A pesquisa realizada teve como objetivo apresentar, estudar e comparar as principais ferramentas de código aberto utilizadas na administração de nuvens. Após uma breve descrição das qualidades e funcionalidades individuais de cada uma delas, foram analisadas e comparadas com base em um conjunto de característica pré-definidas. Com isso, foi possível constatar que dentre as ferramentas pesquisadas existem várias diferenças e estas se tornaram mais simples de identificar com a comparação efetuada.

Com esta comparação, também foi possível ver que o OpenQRM foi a ferramenta que se mostrou mais completa nos quesitos avaliados, pois oferece uma maior gama de opções e possui todas as características elencadas. As ferramentas Eucalyptus, OpenNebula e OpenStack também se destacaram e ficaram bem próximas das qualidades do OpenQRM. Cabe ressaltar que, a melhor ferramenta é a que atende os requisitos

do projeto de rede. Isso foi evidenciado na análise comparativa (Seção IV), na qual se tentou traçar cenários em que as ferramentas possam oferecer um bom comportamento.

Os desafios desta pesquisa como trabalhos futuros são de avaliar estas ferramentas (que se destacaram) em uma nuvem formada apenas por estações de trabalho. Para os experimentos, pretende-se estudar conjuntos de aplicações (*benchmarks*) que permitirão testar as mesmas características avaliadas qualitativamente nesta pesquisa. Assim, será possível efetuar um comparativo do que a literatura nos apresenta e como estas se comportam em um ambiente limitado e heterogêneo como este. Além disso, a hipótese é que isso pode ser uma boa alternativa para melhor aproveitar os recursos das máquinas que se encontram na maior parte do tempo ociosas nas instituições/orgânicas.

## REFERENCES

- [1] P. Mell e T. Grance. The NIST Definition of Cloud Computing. Gaithersburg: National Institute of Standards and Technology, 2011, p.7..
- [2] E. A. Marks e B. Lozano. Executive's Guide to Cloud Computing. 1ºEd. New Jersey: Published by John Wiley & Sons, Inc., 2010, p.285.
- [3] M. Veras. Virtualização: Componente Central do Datacenter. 1º Ed. Rio de Janeiro: Brasport Livros e Multimídia Ltda, 2011, p. 364.
- [4] P. T. Endo, G. E. Gonçalves, J. Kelner e D. Sadok. A Survey on Open-source Cloud Computing Solutions. In: VIII Workshop em Clouds, Grids e Aplicações. 2010. Gramado-RS. Anais. Gramado: Sociedade Brasileira de Computação, 2010, p.3-16.
- [5] G. V. Laszewski, J. Diaz, F. Wang e G. C. Fox. *Comparison of Multiple Cloud Frameworks*. 2012 IEEE Fifth International Conference on Cloud Computing. Washington, 2012, p.734-741.
- [6] A. Khurshid, A. Al-nayeeem e I. Gupta. Performance Evaluation of the Illinois Cloud Computing Testbed. [S.I.], Urbana-Champaign: Illinois Digital Environment for Access to Learning and Scholarsip, 2009, p.12.
- [7] C. P. Machado. Comparação de ferramentas de software Livre para administração de nuvem privada. Canoas: Ulbra, 2011, p.18.
- [8] I. Voras, B. Mihajevic, M. Orlic, M. Pletikosa, M. Zagar, T. Pavic, K. Zimmer, I. Cavrak, V. Paunovic, I. Bosnic e S. Tomic. Evaluating Open-Source Cloud Computing Solutions. In: MIPRO, 2011 Proceedings of the 34th International Convention. 2011. Opatija-HR. Anais. Washington: IEEE Computer Society, 2011, p.209-214.
- [9] P. Sempolinski e D. Thain. "A Comparison and Critique of Eucalyptus, OpenNebula and Nimbus". In: 2010 IEEE Second International Conference on Cloud Computing Technology and Science. 2010. Indianapolis-USA. Processo. Washington-USA: IEEE Computer Society, 2010, p.417-426.
- [10] Opennebula. *About the OpenNebula.org Project*. Extraído de <<http://opennebula.org/about/about/>>. Acesso em 22 de julho de 2013.
- [11] Openstack. *About OpenStack*. Extraído de <<http://www.openstack.org/software/>>. Acesso em 30 de julho de 2013.
- [12] N. Sabharwal e R. Shankar. *Apache Cloudstack Cloud Computing*. 1º Ed. Reino Unido: Packt Publishing Ltd., 2013, p.294.
- [13] J. F. Ransome e J. W. Rittinghouse. *Cloud Computing: Implementation, Management and Security*. 1º Ed. Boca Raton: CRC Press, 2009, p. 407.
- [14] S. Wardley, E. Goyer e N. Barret. *Ubuntu Enterprise Cloud Architecture*. [S.I.], Man Island: Canonical, 2009, p.18.
- [15] Abiquo. Abiquo. Extraído de: <<http://www.abiquo.com/overview-technical/>>. Acesso em 24 de julho de 2013.
- [16] R. Buyya, J. Broberg e A. M. Goscinski. *Cloud Computing: Principles and Paradigms*. 1º Ed. John Wiley and Sons, 2010, p. 660.
- [17] Apache VCL. Apache VCL. Extraído de <<http://vcl.apache.org/>>. Acesso em 10 de agosto de 2013.
- [18] A. S. Pillai e L. S. Swasthima. A Study on Open Source Cloud Computing Platforms. EXCEL International Journal of Multidisciplinary Management Studies. Zenit, 7 jul 2012. p.31-40.

# Elasticidade Automática de Recursos para Aplicações de Alto Desempenho em Ambientes de Computação em Nuvem

Vinicio Facco Rodrigues

Universidade do Vale do Rio dos Sinos  
CEP: 93.022-000 – São Leopoldo – Brazil  
Email: viniciusfacco@live.com

Cristiano André da Costa, Rodrigo da Rosa Righi

PPG em Computação Aplicada (PIPCA)  
Universidade do Vale do Rio dos Sinos  
CEP: 93.022-000 – São Leopoldo – Brazil  
Email: {cac,rrrighi}@unisinos.br

**Resumo**—A elasticidade é sem dúvida uma das características mais marcantes da computação em nuvem que pode ser usada na execução de aplicações que demandam processamento de alto desempenho (HPC). Normalmente, a elasticidade é oferecida em sistemas Web com o aumento e a diminuição de máquinas virtuais (MVs) que tratam requisições. Para HPC, a maioria dos sistemas necessita alterar a aplicação ou partir de um conhecimento prévio da aplicação para o tratamento dessa característica da nuvem. Nesse contexto, esse artigo apresenta o modelo de elasticidade na camada de Plataforma como Serviço (PaaS) chamado AutoElastic. Seu diferencial está no tratamento automático e sem esforço da elasticidade em aplicações de alto desempenho sem a intervenção do usuário e sem alterações no código da aplicação. A avaliação do protótipo sobre o sistema OpenNebula mostrou um ganho de desempenho de 14% no tempo de execução de uma aplicação e uma baixa intrusividade de AutoElastic na execução da aplicação paralela.

## I. INTRODUÇÃO

Algumas das características mais marcantes que distinguem a computação em nuvem de outras abordagens de sistemas distribuídos é a elasticidade [1]. A elasticidade de recursos em ambientes de nuvem (*cloud computing*) explora o fato que a alocação de recursos é um procedimento que pode ser efetuado dinamicamente de acordo com a demanda do serviço ou do usuário. Nesse sentido, ela é um princípio essencial para o modelo de nuvem, pois não fornece somente o compartilhamento eficiente de recursos entre usuários, mas também é pertinente para a viabilização de computação no estilo *pay-as-you-go*.

Múltiplos provedores de computação em nuvem estão focando em aplicações que demandam alto desempenho (HPC). Aplicações de HPC na sua maioria possuem dificuldade para usufruir da elasticidade visto que normalmente são construídas com um número fixo de processos. Essa é a situação daqueles programas que seguem a interface 1.0 de MPI (*Message Passing Interface*). MPI 2.0 sobrepassa essa limitação proporcionando a criação de processos em tempo de execução. A elasticidade nesse contexto possui algumas peculiaridades. Um ou mais processos da aplicação deve proativamente ou de forma periódica analisar se há novos recursos para assim utilizá-lo e proceder com o mecanismo de balanceamento de carga. Por exemplo, em MPI 2.0 é necessário um esforço manual para mudar o grupo de processos e redistribuir os dados eficientemente para usar um número diferente de processos [2].

Nesse contexto, esse artigo apresenta um modelo chamado AutoElastic que gerencia a elasticidade em aplicações HPC. Ele provê dinamicidade na alocação dos recursos sem a intervenção do programador. O gerenciador proposto se destaca por operar sem conhecimento prévio da aplicação, desconhecendo por exemplo o tempo esperado de suas fases. Em termos de contribuições científicas, é possível destacar os seguintes pontos: (i) gerenciamento eficiente da elasticidade com o intuito de evitar a alocação e desalocação de MVs de forma desnecessária (*trashing*) baseada na técnica de Envelhecimento (*Aging*) [3]; (ii) assincronismo na criação e destruição de MVs, de modo que a aplicação não espere de forma bloqueada pela conclusão desses procedimentos. Esse artigo descreve em detalhes a implementação de um protótipo de AutoElastic que usa o *middleware* OpenNebula. Os testes mostram resultados encorajadores e ganhos de desempenho de até 14% quando comparadas a solução proposta e a alocação estática de recursos.

## II. AUTOELASTIC: GERENCIADOR DE ELASTICIDADE

Uma das primeiras medidas para o desenvolvimento de AutoElastic é a definição de regras de elasticidade. Tal tarefa não é trivial por vários motivos. Primeiro, ela requer que sejam configuradas regras que usam um ou mais *thresholds* para uso dos recursos (por exemplo, uso de CPU e memória), a quantidade de ocorrência, a janela de monitoramento e eventuais adaptações em circunstância da aplicação. Segundo, regras otimizadas para o comportamento de uma aplicação podem apresentar um desempenho ruim para outra. Por fim, a elasticidade muitas vezes acarreta na alocação de recursos que por sua vez impactam diretamente no custo de uso da nuvem.

A principal ideia de AutoElastic é oferecer um modelo que possa ser usado para criar aplicações elásticas, que estejam aptas para se adaptar automaticamente de acordo com mudanças no número de recursos (em ambos os níveis: nó computacional e máquina virtual). AutoElastic deve tratar com aplicações que não são regidas por *deadlines* e que podem apresentar irregularidades de computação ao longo de sua execução. Em adição, o foco é aplicações com passagem de mensagens e paralelismo explícito, com uso de diretivas de envio e recebimento de dados direto no código fonte. Ainda, AutoElastic deve ser ciente do tempo de lançamento de uma MV e deve trabalhar de modo que essa sobrecarga impacte o mínimo possível na execução da aplicação.

A Figura 1 ilustra a atuação de AutoElastic, onde a elasticidade é feita de maneira reativa e sem a necessidade de preconfiguração de métricas pelo usuário. Abaixo estão listadas as decisões de projeto, bem como as condições para o funcionamento de AutoElastic.

- Usuário não precisa preconfigurar tampouco escrever regras e ações. Em contrapartida, ele deve apresentar acordo de nível de serviço (SLA) que informa o número mínimo e máximo de Nós que podem ser alocadas.
- A aplicação paralela não precisa ser reescrita para ter o caráter elástico.
- O cenário investigado por AutoElastic assume um ambiente que não é compartilhado por outros usuários. Ele contempla um usuário que executa um único programa.
- AutoElastic oferece uma elasticidade reativa, automática, em ambas as modalidades horizontal e vertical [2], e segue o método de replicação.
- O nível de atuação é o PaaS, que contempla uma ferramenta que transforma a aplicação paralela em outra elástica de forma transparente para o usuário.
- Análise de picos de carga e quedas bruscas para não lançar ações de elasticidade de forma desnecessária, minimizando assim um fenômeno conhecido como *thrashing*.

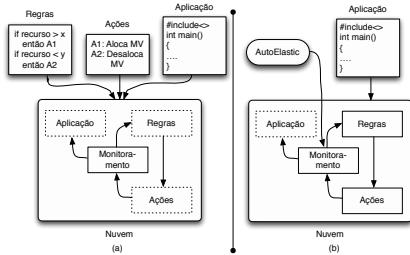


Figura 1. (a) Abordagem usada no Windows Azure e Amazon AWS na qual o usuário pré configura regras e ações; (b) Ideia geral de AutoElastic

#### A. Arquitetura do Modelo

A Figura 2 ilustra a arquitetura de componentes de AutoElastic e o mapeamento de MVs. Primeiramente, o Gerente AutoElastic pode ser mapeado para uma MV dentro da nuvem ou atuar como um programa fora dela. Essa flexibilidade é atingida através do uso da API do middleware de nuvem usado. Uma vez que normalmente aplicações de alto desempenho são intensivas quanto ao uso de CPU, optou-se por criar um processo por MV e n MVs por nó, onde n é o número de núcleos de processamento (*cores*) que o nó possui. Essa abordagem está baseada nos trabalhos de Lee et al [4], onde se busca explorar uma melhor eficiência ( $\frac{Speedup(p)}{p}$ , para p núcleos) para a execução da aplicação. Num primeiro momento, AutoElastic

suporta aplicações desenvolvidas segundo o modelo mestre-escravo. Mesmo tendo uma organização trivial, esse modelo é usado em vários algoritmos genéticos, técnica de Monte Carlo, transformações geométricas na computação gráfica, algoritmos de criptografia e aplicações no estilo SETI-at-home [2].

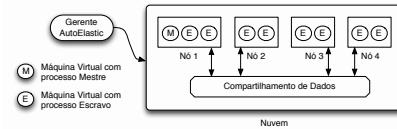


Figura 2. Arquitetura onde os Nós possuem dois núcleos de processamento

O Gerente AutoElastic monitora os Nós e toma as decisões de elasticidade. O usuário pode informar um arquivo de SLA com o mínimo e o máximo de Nós para a execução de sua aplicação. Caso esse arquivo não seja informado, assume-se que o número máximo permitido de Nós é o dobro daquele informado no lançamento. As operações de elasticidade realizadas devem chegar ao conhecimento da aplicação de alguma maneira. Para tal, foi modelada uma comunicação entre as MVs e o Gerente AutoElastic através de uma área de dados compartilhada que pode ser viabilizada, por exemplo, via compartilhamento NFS (Network File System), middleware orientado a mensagens, como JMS (Java Message Service) ou AMQP (Advanced Message Queuing Protocol), ou espaço de tuplas (como JavaSpaces).

Assim como nos trabalhos de Imai et al [5] e Chiu et al [6], o monitoramento do gerente para as ações de elasticidade é dado de forma periódica. Assim, de tempos em tempos analisa-se a métrica, que é comparada a um valor mínimo e outro máximo de *threshold*. A Figura 3 apresenta o modelo reativo de elasticidade de AutoElastic. Quanto às condições, elas trabalham com uma métrica chamada PC (Predição de Carga), que é baseada numa série que leva em conta a carga C em cada um dos pontos de monitoramento numa ordem da mais nova para a mais antiga. A carga C é dada pela média aritmética da carga de CPU de todas as MVs em execução. PC é encontrada pela fórmula de recorrência abaixo.

$$PC(i) = \begin{cases} C(i) & \text{se } i = 0 \\ \frac{1}{2}PC(i-1) + \frac{1}{2}C(i) & \text{se } i > 0 \end{cases}$$

REGRA1: IF CONDIÇÃO1 então AÇÃO1  
REGRA2: IF CONDIÇÃO2 então AÇÃO2  
CONDIÇÃO1: PC(i) < threshold1, onde i é a última observação do sistema de monitoramento.  
CONDIÇÃO2: PC(i) > threshold2, onde i é a última observação do sistema de monitoramento.  
AÇÃO1: Aloca um novo nó e lança n VMs, onde n é o número de núcleos de processamento no nó.  
AÇÃO2: Finaliza as instâncias que estão rodando em um nó e depois o consolida.

Figura 3. Modelo de elasticidade reativa de AutoElastic

#### B. Modelo de Aplicação Paralela

A Figura 4 (a) apresenta uma aplicação mestre-escravo suportada por AutoElastic. Ela se caracteriza pelo caráter iterativo, onde o mestre possui uma série de tarefas, as captura sequencialmente e paralleliza uma a uma para processamento nos processos escravos. Essa captura de trabalhos

é evidenciada no *loop* externo da Figura 4 (a). AutoElastic trabalha com as seguintes diretivas baseadas na Interface de MPI 2.0: (i) publicar e despular uma porta de conexão; (ii) procurar o servidor a partir de uma porta; (iii) aceitar uma conexão; (iv) requisitar uma conexão e; (v) realizar uma desconexão. Diferente da abordagem onde o processo mestre lança processos (usando a diretiva *spawn*), o modelo proposto atua segundo a outra abordagem de MPI 2.0 para o gerenciamento dinâmico de processos: comunicação ponto-a-ponto com conexão e desconexão. O lançamento de uma máquina virtual acarreta automaticamente na execução de um processo escravo, que requisita uma conexão com o mestre.

```

1. Para (j=0; < total_trabalho; j++)
2. {
3.     tamanno = politica_portas(portas);
4.     Para (i=0; < tamanno; i++)
5.     {
6.         aceita_conexao(escravos[i], portas[i]);
7.         calcula_tamanno(trabalho[i], intervalos);
8.         Para (i=0; < tamanno; i++)
9.         {
10.             {
11.                 mestre = procura(endereco_mestre, servico_nomes);
12.                 porta = monta_porta(endereco_IP, id_VM);
13.                 Para (sempre)
14.                 {
15.                     requisiqtaconexao(mestre, porta);
16.                     recebe(mestre, tarefa);
17.                     resultado = computa(tarefa);
18.                     envia_resposta(resultado);
19.                     desconecta(escravos[i]);
20.                 }
21.             }
22.             despulica_portas(portas);
23.         }
24.     }
}

```

(a)

```

1. mestre = procura(endereco_mestre, servico_nomes);
2. porta = monta_porta(endereco_IP, id_VM);
3. Para (sempre)
4. {
5.     requisiqtaconexao(mestre, porta);
6.     recebe(mestre, tarefa);
7.     resultado = computa(tarefa);
8.     envia_resposta(resultado);
9.     desconecta(mestre);
10. }

```

(b)

Figura 4. Modelo de aplicação suportado por AutoElastic: (a) pseudocódigo do mestre e; (b) pseudocódigo do processo escravo

O método da linha 2 do código do mestre verifica um arquivo de configuração ou argumentos passados para o programa que informam identificadores de máquinas virtuais e endereços IP de cada um dos processos. Com base nisso, mestre sabe a quantidade de escravos e cria nomes de porta para receber conexões específicas de cada um deles. Quanto à comunicação, ela acontece de forma assíncrona no processo mestre, onde o envio de dados para os escravos é de forma não bloqueante e a recepção é bloqueante. O fato de assumir programas com um *loop* externo é conveniente para a elasticidade, pois logo no início dele é possível que a quantidade de recursos e processos seja reconfigurada sem alterar a semântica da aplicação. A transformação da aplicação mostrada na Figura 4 em outra elástica pode ser feita em nível PaaS através de uma das seguintes maneiras: (i) numa implementação orientada a objetos, incluir um método cuja funcionalidade seja nula entre as linhas 2 e 3 do código do mestre, o qual é sobreescrito para gerir a elasticidade; (ii) fazer um tradutor fonte-para-fonte que insira um código entre as linhas 2 e 3; (iii) desenvolvimento de um *wrapper* em linguagens procedurais para o método da linha 3.

A região de código de monitoramento verifica no diretório compartilhado se há alguma ação nova de AutoElastic. Se tivermos ocorrência de Ação1, o mestre lê os dados e os adiciona na região de memória de processos escravos. Se ocorrer a Ação2, o mestre retira os processos envolvidos da lista de processos e aciona a Ação3. Embora o foco inicial de AutoElastic sejam aplicações mestre-escravo, a modelagem iterativa e o uso de diretivas de MPI 2.0 (no estilo de Sockets) facilitam a inclusão de processos e o reestabelecimento das conexões para uma nova topologia totalmente arbitrária.

Em nível de implementação, é possível otimizar conexões e desconexões caso o processo persistir na lista de processos ativos. Essa atitude é pertinente principalmente sobre conexões TCP/IP, que usa um protocolo de três vias que sabidamente pode acarretar sobrecarga em aplicações de alto desempenho.

### III. IMPLEMENTAÇÃO DE PROTÓTIPO

Um protótipo foi implementado em Java usando o sistema OpenNebula. Foram criadas imagens de dois modelos de máquinas virtuais: uma para o processo mestre e outra para processos escravos. O Gerente AutoElastic utiliza a própria API Java de OpenNebula para as atividades de monitoramento e emprega a elasticidade nas modalidades horizontal e vertical. Ainda, essa API é usada por ele para lançar a aplicação paralela na nuvem, a qual é associada a um SLA que pode ser fornecido pelo usuário. O SLA segue o padrão XML de WS-Agreement<sup>1</sup> e informa a quantidade mínima e máxima de Nós para teste da aplicação. A seguir são descritas em detalhes algumas decisões técnicas de AutoElastic.

- Compartilhamento de dados - Implementado com NFS. Tecnicamente, AutoElastic usa SSH para se conectar a máquina gerenciadora da nuvem e a partir dali tem acesso ao diretório compartilhado NFS.
- Noção de carga - Em nível de protótipo, a carga  $C$  para a amostra  $i$  de monitoramento, denominada  $C(i)$  na fórmula de recorrência apresentada, é dada pela média aritmética da carga de todas as CPUs em execução num determinado momento.
- Monitoramento periódico - Utilizou-se o valor de 30 segundos para o intervalo de medições de desempenho.
- Thresholds - Com base em trabalhos relacionados, optou-se pela adoção de 80% para o valor máximo e 40% para o mínimo.

### IV. MODELAGEM DA APLICAÇÃO PARALELA E METODOLOGIA DE AVALIAÇÃO

A aplicação usada nos testes realiza o cálculo de soma de dois vetores de mesmo tamanho. A aplicação utiliza sockets TCP para a comunicação entre o processo mestre e os processos escravos. O processo mestre realiza a leitura de  $x$  arquivos, cada um contendo dois vetores que serão somados. Considerando o tamanho do subvetor definido por  $n$  e o tamanho do vetor definido por  $t$ , cada arquivo processado contém  $q$  tarefas para serem distribuídas para os processos escravos, sendo que  $q$  é definido por  $q = \frac{t}{n}$ . O processo mestre envia para cada processo escravo uma tarefa por iteração e aguarda o retorno dos processos para o envio das próximas tarefas. Ao concluir todas as tarefas geradas pela leitura de um arquivo, o arquivo seguinte é processado. Considerando  $i$  a quantidade de iterações para o processamento de cada arquivo e  $p$  a quantidade de processos disponíveis,  $i$  é dado por:

$$i = \begin{cases} \frac{q}{p} & \text{se } mod(\frac{q}{p}) = 0 \\ \frac{q}{p} + 1 & \text{se } mod(\frac{q}{p}) > 0 \end{cases}$$

<sup>1</sup><http://www.ogf.org/documents/GFD.107.pdf>

Em um cenário onde  $\frac{q}{p}$  não retornar um número inteiro, na última iteração do processamento de um arquivo existirão um ou mais processos ociosos.

## V. AVALIAÇÃO E ANÁLISE DE RESULTADOS

Foram realizados testes com um arquivo SLA definindo a quantidade mínima e máxima de Nós em 2 e 4 respectivamente. Os seguintes cenários foram avaliados: (i) Execução em nuvem com quantidade fixa de recursos; (ii) Execução em nuvem utilizando AutoElastic com o SLA informado. Foram utilizados arquivos contendo três tamanhos diferentes de vetores: 500, 1000 e 10000. Em todas as execuções o tamanho dos subvetores foi definido em 500, variando a quantidade de tarefas que cada arquivo de tamanho diferente possui. A aplicação foi executada no ambiente formado por dois Nós e cinco MVs, das quais quatro executando processos escravo e uma executando o processo mestre.

Para o processamento dos arquivos no primeiro cenário, o tempo de execução médio foi de 22 minutos e 34 segundos, com um desvio padrão de 30 segundos. Foram observados três níveis de carga gerados na nuvem dependendo dos arquivos que estavam sendo processados. Isso se deve a quantidade de tarefas que cada arquivo gerou. Neste cenário, nenhuma operação de elasticidade foi executada. No segundo cenário, o tempo de execução médio observado foi de 19 minutos e 23 segundos, com um desvio padrão de 31 segundos. Assim como no primeiro cenário, os arquivos processados geraram três níveis de carga diferentes. Como neste cenário o AutoElastic monitorou estes níveis, foram realizadas operações de elasticidade nos momentos em que cargas ficaram fora da faixa definida pelos *thresholds*, respeitando o SLA. Uma breve comparação demonstra que a utilização do AutoElastic representou um ganho de 14,1% no desempenho da aplicação. Todas as operações de elasticidade ocorreram sem nenhuma ação do usuário.

## VI. TRABALHOS RELACIONADOS

ElasticMPI propõe a elasticidade em aplicações MPI através da parada e relançamento delas no momento da reconfiguração de recursos [2]. A abordagem de ElasticMPI faz uma alteração no código fonte da aplicação de modo a inserir diretivas de monitoramento. Em adição, a abordagem de ElasticMPI faz uma alteração no código fonte da aplicação de modo a inserir diretivas de monitoramento. A elasticidade é mais explorada em nível de plataforma (IaaS) e de forma reativa. Nesse sentido, os trabalhos não são uníssinos quanto ao emprego de um *threshold* único para os testes. Por exemplo, é possível notar os seguintes valores: (i) 70% [7]; (ii) 75% [5]; (iii) 80% [8], [9]; (iv) 90% [10], [11], [12]. Esses valores tratam de limites superiores que quando ultrapassados, acionam ações de elasticidade de forma horizontal e/ou vertical.

## VII. CONCLUSÃO

O presente trabalho descreveu o modelo AutoElastic e um protótipo que trabalha sobre OpenNebula. Apesar de simples, o protótipo possibilitou verificar que os resultados da elasticidade automática são promissores. Foi adotada uma abordagem periódica para o monitoramento, que verifica o estado global dos processos e Nós e aplica regras de elasticidade. Tais regras

estão baseadas em *thresholds* que quando atingidos, disparam ações para redimensionamento da arquitetura. Trabalhos futuros incluem a utilização de uma arquitetura dinâmica de recursos e aplicações irregulares quanto à carga de trabalho em cada processo.

## REFERÊNCIAS

- [1] N. Cook, D. Milojicic, and V. Talwar, "Cloud management," *Journal of Internet Services and Applications*, vol. 3, pp. 67–75, 2012.
- [2] A. Ravendran, T. Bicer, and G. Agrawal, "A framework for elastic execution of existing mpi programs," in *Proceedings of the 2011 IEEE Int. Symposium on Parallel and Distributed Processing Workshops and PhD Forum*, ser. IPDPSW '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 940–947.
- [3] A. Tanenbaum, *Computer Networks*, 4th ed. Upper Saddle River, New Jersey: Prentice Hall PTR, 2003.
- [4] Y. Lee, R. Avizienis, A. Bishara, R. Xia, D. Lockhart, C. Batten, and K. Asanovic, "Exploring the tradeoffs between programmability and efficiency in data-parallel accelerators," in *Computer Architecture (ISCA), 2011 38th Annual International Symposium on*, 2011, pp. 129–140.
- [5] S. Imai, T. Chestna, and C. A. Varela, "Elastic scalable cloud computing using application-level migration," in *Proceedings of the 2012 IEEE/ACM Fifth International Conference on Utility and Cloud Computing*, ser. UCC '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 91–98.
- [6] D. Chiu and G. Agrawal, "Evaluating caching and storage options on the amazon web services cloud," in *Grid Computing (GRID), 2010 11th IEEE/ACM International Conference on*, oct. 2010, pp. 17–24.
- [7] W. Dawoud, I. Takouna, and C. Meinel, "Elastic vm for cloud resources provisioning optimization," in *Advances in Computing and Communications*, ser. Communications in Computer and Information Science, A. Abraham, J. Lloren Mauri, J. Buford, J. Suzuki, and S. Thampi, Eds. Springer Berlin Heidelberg, 2011, vol. 190, pp. 431–445.
- [8] R. Bryant, A. Tumanov, O. Irzak, A. Scannell, K. Joshi, M. Hiltunen, A. Lagar-Cavilla, and E. de Lara, "Kaleidoscope: cloud micro-elasticity via vms state coloring," in *Proceedings of the sixth conference on Computer systems*, ser. CSC '11. New York, NY, USA: ACM, 2011, pp. 273–286. [Online]. Available: <http://doi.acm.org/10.1145/1966445.1966471>
- [9] M. Mihailescu and Y. M. Teo, "The impact of user rationality in federated clouds," *Cluster Computing and the Grid, IEEE International Symposium on*, vol. 0, pp. 620–627, 2012.
- [10] L. Beernaert, M. Matos, R. Vilaça, and R. Oliveira, "Automatic elasticity in openstack," in *Proceedings of the Workshop on Secure and Dependable Middleware for Cloud Monitoring and Management*, ser. SDMCM '12. New York, NY, USA: ACM, 2012, pp. 2:1–2:6. [Online]. Available: <http://doi.acm.org/10.1145/2405186.2405188>
- [11] W. Dawoud, I. Takouna, and C. Meinel, "Elastic virtual machine for fine-grained cloud resource provisioning," in *Global Trends in Computing and Communication Systems*, ser. Communications in Computer and Information Science, P. Krishna, M. Babu, and E. Ariwa, Eds. Springer Berlin Heidelberg, 2012, vol. 269, pp. 11–25.
- [12] B. Suleiman, "Elasticity economics of cloud-based applications," in *Proceedings of the 2012 IEEE Ninth International Conference on Services Computing*, ser. SCC '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 694–695.

# Implementação do Protocolo PROFIsafe para o Desenvolvimento de Sistemas Seguros

William Vidal, Rodrigo Dobler, Sérgio Cechin, Taisy Weber e João Netto

Departamento de Informática Aplicada  
Universidade Federal do Rio Grande do Sul

Porto Alegre, Brasil  
{wrcvidal, rjdobler, cechin, taisy, netto}@inf.ufrrgs.br

**Resumo —** Protocolos de comunicação seguros são essenciais para o desenvolvimento de sistemas seguros a serem utilizados em aplicações críticas na automação industrial, onde falhas não detectadas podem causar danos irreparáveis vida ou ao meio-ambiente. Esses protocolos seguros são desenvolvidos e certificados de acordo com a norma de Segurança Funcional IEC61508. Essa norma especifica os mecanismos para detecção de erros que devem ser implementados para detectar os erros de comunicação que podem ocorrer durante a transmissão de dados, assegurando assim a troca correta de informações entre os dispositivos. Este artigo apresenta a implementação do protocolo seguro de comunicação PROFIsafe, o qual deve ser utilizado para o desenvolvimento de funções seguras que precisam ser certificadas.

**Palavras-chave**—comunicação segura; detecção de erros; sistemas seguros; PROFIsafe; IEC61508;

## I. INTRODUÇÃO

Os sistemas de automação industrial são baseados em barramentos de comunicação [1], os quais permitem a interligação de sistemas cada vez mais descentralizados nas plantas industriais [2]. Essa interligação deve ser realizada de forma rápida, confiável e robusta, características necessárias no caso de aplicações em processos com altos níveis de criticidade [2].

Com o crescente uso dos barramentos e o desenvolvimento de complexos sistemas de automação industrial e de controle de processos, tornou-se essencial que fossem concebidos os sistemas seguros (*safety*). Aquelas em que defeitos são raros e que, quando eventualmente ocorrem, não provocam danos às pessoas ou ao meio ambiente. Ou seja, são aquelas em que os riscos de provocar um acidente são considerados aceitáveis pelas normas reguladoras.

Esses sistemas são caracterizados por incorporarem mecanismos de diagnóstico, detecção e correção de falhas, que permitem implementar sistemas com altos níveis de Integridade de Segurança. Essa é uma medida da eficiência do sistema em evitar ou mitigar um acidente grave.

Entretanto, para que o sistema como um todo possa ser considerado seguro, é necessário que os mecanismos de comunicação entre os seus componentes também apresentem essa propriedade. Assim, foram desenvolvidos os protocolos de comunicação seguros, tais como o FF SIS [3], openSafety [4], PROFIsafe [5], o Safety over EtherCAT [6], entre outros.

Esses protocolos seguros são especificados de maneira a atender os requisitos de segurança da norma de Segurança Funcional IEC 61508 [7]. A norma estabelece requisitos para a determinação do nível de integridade de uma função de segurança. Essas funções se destinam a atingir ou manter um estado seguro em um processo no que diz respeito a um evento perigoso específico, por exemplo, vazamento de gás, ameaça de incêndio.

Ao se desenvolver uma função de segurança para aplicações críticas, como para a área de óleo e gás, é necessário que ela seja certificada. Nesse processo, o hardware e o software são avaliados em relação às especificações da norma. A comunicação de dados precisa ser coberta por mecanismos que assegurem a entrega de pacotes e a detecção dos erros de comunicação previstos na norma.

Dessa forma não se pode utilizar protocolos tradicionais de comunicação, como o TCP, para implementar a comunicação em funções de segurança para aplicações críticas. Isso ocorre porque ele não possui mecanismos de detecção de erros suficientemente robustos para tratar todos os tipos de falhas de comunicação especificados na norma.

Nesse âmbito, este trabalho apresenta a implementação do protocolo seguro PROFIsafe, escolhido para o projeto RIO-SIL, por interesse da empresa parceira. O objetivo desse projeto visa à produção de módulos de entrada e saída digitais para sistemas instrumentados de segurança, os quais devem ser certificados. Além disso, outra justificativa para implementação do protocolo PROFIsafe deve-se ao fato da certificação ser feita em todo o processo de desenvolvimento e não só no resultado final.

O artigo está organizado da seguinte forma: A seção 2 apresenta o protocolo PROFIsafe. A seção 3 apresenta os trabalhos relacionados. A seção 4 mostra alguns equipamentos PROFIsafe. A seção 5 apresenta a implementação do protocolo PROFIsafe. Na seção 6 são comentados os resultados obtidos. A seção 7 apresenta a conclusão e os trabalhos futuros. A seguir são apresentadas as referências utilizadas.

## II. PROFIsafe

É um protocolo de comunicação seguro, no nível de transporte, desenvolvido pela Profibus & Profinet Internacional [8] para ser utilizado com as redes PROFIBUS e PROFINET. O uso em conjunto do protocolo com estas redes, permite que os

equipamentos e sistemas desenvolvidos para elas possam ser utilizados na criação de funções de segurança.

A especificação do PROFIsafe [5] está em conformidade com a norma IEC 61508, atendendo os requisitos para sistemas com até SIL 3 (IEC 61508) e FSCP 3 (*Functional Safety Communication Profile – 3*) da IEC 61784-3 [9]. Os níveis de integridade de segurança vão de SIL 1 a SIL 4, sendo SIL 4 o nível de maior integridade. No setor de óleo e gás os dispositivos precisam ser certificados SIL 3 [7] que pode ser alcançado através do uso do protocolo PROFIsafe em aplicações de segurança, e cuja implementação deve ser certificada pela Agência de Inspeção Técnica alemã TÜV SÜD.

O PROFIsafe reduz a probabilidade de erros nos dados transmitidos entre um *F-Host* (controlador seguro) e um *F-Device* (dispositivos com segurança integrada) para o nível exigido por uma norma. Também é possível a comunicação, de informações seguras e não seguras em um único barramento físico de comunicação. Para isso, é necessário garantir que os dois tipos de processamento ocorram de forma logicamente isolada.

O protocolo deve ser implementado sobre um *Black Channel* [5], o que o torna bastante independente dos canais físicos de transmissão, sejam eles fios de cobre, fibras ópticas, wireless ou *backplanes* (grupo de conectores ligados de maneira a formar um barramento). As taxas de transmissão e os mecanismos de detecção de erros do *Black Channel* não têm qualquer interferência sobre o protocolo seguro. Esse mecanismo torna desnecessária a avaliação de segurança dos elementos que o compõem: *backplanes* individuais, caminhos de transmissão e redes PROFIBUS e PROFINET.

Para detectar erros de comunicação, o protocolo foi especificado com os seguintes mecanismos [5]: número sequencial de mensagens, controle de temporização, identificador único e CRC (Cyclic Redundancy Check). Esses mecanismos são necessários porque quando mensagens são transferidas em redes complexas, vários erros podem ocorrer. Esses erros podem ser ocasionados por falhas de hardware, interferência eletromagnética ou outros tipos de influências e devem ser tratados e corrigidos de alguma forma para garantir a entrega correta de mensagens.

A implementação do protocolo pode ser feita a partir da sua especificação ou obtida através do PROFIsafe StarterKit, o qual contém a implementação da comunicação, para ambos os casos, o código deve ser adaptado para aplicação destino.

### III. TRABALHOS RELACIONADOS

Åkerberg *et. al.* [10] abordam algumas questões que surgem quando o protocolo PROFIsafe é utilizado para implementar a comunicação segura em redes de sensores sem fio. Neste trabalho, foi proposto um método para integração dos dispositivos, que utilizavam o protocolo WirelessHART [11], PROFI-NET IO e o PROFIsafe. Nos testes realizados, foi observado que a taxa de erros de bit da rede de sensores sem fio deve ser considerada quando se deseja conformidade com os padrões de segurança (IEC 61508 e IEC 61784-3), pois, em alguns casos, essa taxa de erros pode ser muito alta. Também foi notado que, em alguns casos, o tempo de resposta do protocolo WirelessHART é muito longo para implementar

algumas funções de segurança que necessitam de um tempo menor de resposta, exigindo uma análise mais detalhada do tempo de resposta do processo a ser controlado. Como principal resultado, apesar das limitações, foi mostrado que é possível estabelecer uma comunicação segura usando-se uma combinação do WirelessHART, PROFINET IO e o PROFIsafe.

No artigo de Malik [12] é apresentada a validação do protocolo seguro PROFIsafe. O objetivo do trabalho era o de garantir a correta especificação do protocolo. Para isso, foi obtido um modelo formal baseado em máquinas de estados finitos a partir da especificação UML do protocolo, a qual define o seu comportamento. Este modelo foi analisado com técnicas de verificação formal para garantir que não existem *loops* e *deadlocks*. Quando uma implementação anterior do protocolo foi verificada quanto aos requisitos para a comunicação segura, foram identificadas falhas. Então, foi realizada uma nova versão da implementação que corrigiu as falhas identificadas. Um conjunto de casos de teste foi derivado, baseados no modelo verificado, para executar testes automáticos de conformidade, com a intenção de verificar se a implementação do protocolo tem o mesmo comportamento do modelo verificado.

### IV. EQUIPAMENTOS PROFISAFE

A segurança na engenharia de automação tem recebido muita atenção devido ao risco de lesões corporais, danos materiais e danos ao meio-ambiente ser inherente aos processos industriais [13].

Assim, muitos fabricantes de componentes seguros como, por exemplo, Siemens [14], WAGO [15], BECKHOFF [16], participaram da criação de padrões abertos sob o framework da Profibus & Profinet International. Isso permitiu o desenvolvimento de um extenso portfólio de produtos seguros [13].

Estes produtos são normalmente utilizados para implementar funções de segurança em processos industriais, no setor de transportes, em equipamentos de guindastes, teleféricos, na indústria automotiva, entre outros [13]. Como exemplo dos equipamentos desenvolvidos com PROFIsafe, pode-se citar: remotas de I/O, sensores ópticos, sistemas de controle de segurança, gateways seguros, sensores de segurança, dispositivos com funções de segurança integrada, válvulas e bloqueios de válvulas, entre outros [5].

Todos esses equipamentos possuem em comum a necessidade de trocar informações através de uma rede segura de comunicação. É nesse cenário que os protocolos de comunicação segura, tais como o PROFIsafe, cuja implementação é motivo desse trabalho, se encaixam.

### V. DESENVOLVIMENTO

O protocolo visa garantir a segurança da comunicação. Então, em qualquer arquitetura de implementação que seja utilizada, o protocolo estará dividido entre uma parte segura e outra não segura. A parte segura é onde está o código da aplicação segura. Esse código deve ser implementado segundo a norma: entre outras limitações pode-se citar o não uso de ponteiros (o que levou ao uso das chamadas de sinalização e passagem de

dados *byte-a-byte*) e o não uso de *threads* (com exceção se houver um sistema de gerência de *threads* que seja certificado). Também é necessária a existência de Hardware (processador) com alto nível de Integridade de Segurança, que garanta a execução correta e que os dados não sejam corrompidos.

Na parte não segura estão os mecanismos padrões de comunicação. A implementação dessa parte pode ser feita de maneira convencional, conforme definido pela técnica de *Black Channel*. Foi necessário controlar o acesso aos *buffers* internos da parte segura, o que levou a implementação de controles através da chamada de uma função específica, usada apenas para leitura.

O modelo de comunicação do PROFIsafe é do tipo um para um, tendo o mestre como o *F-Host* e o escravo como o *F-Device*. Com isso, o *F-Device* só envia mensagens em resposta às mensagens de solicitação de serviços. O resultado final foi uma arquitetura sem *threads*, onde a aplicação determina o passo de execução de todos os mecanismos implementados na parte segura da arquitetura. É importante salientar que as interrupções de recepção da comunicação se encontram na parte não segura da arquitetura.

#### A. Pacote PROFIsafe

O pacote PROFIsafe (*Safety PDU*) é formada por CRC, *Status/Control Byte* e Dados. Há dois modos de operação:

1) I/O binária (processada em alta velocidade): São transmitidos até 12 octetos de dados. É utilizado um CRC de 24 bits (3 octetos).

2) I/O de dados longo (processada mais lentamente): São transmitidos até 123 octetos de dados. É utilizado um CRC de 32 bits (4 octetos).

Em ambas as PDUs o campo *Status/Control Byte* tem 1 octeto. Neste trabalho, levando em consideração a aplicação, decidiu-se pelo modo I/O binária, em que são usados 16 octetos no total para compor a PDU segura.

#### B. Arquitetura da Implementação

A arquitetura é formada por três camadas: a parte segura, onde está a aplicação; a implementação do protocolo seguro; e a parte não segura, onde está a biblioteca do canal de comunicação.

A implementação do PROFIsafe foi feita na linguagem C e definiu-se por utilizar PDUs com tamanho fixo. Entretanto, o programador da aplicação pode decidir o tamanho do *payload* que será transportado, podendo ter no máximo, 12 octetos.

A troca de informações entre camadas de software do *F-Device* é feita através de chamadas de função entre elas e chamadas de sinalização (que não têm parâmetros). O motivo da escolha dessa arquitetura, vista na Fig.1, foi a segurança (*safety*) da implementação de software, que restringe o uso de ponteiros (Norma IEC 61508).

Para recebimento de pacotes, as sinalizações *askRecvMsg()* e *askRecvData()* são implementadas nas camadas superiores, ou seja, para onde o fluxo de dado aponta. Por exemplo, o Protocolo Seguro “pergunta” (*pooling*) para a biblioteca do

canal de comunicação através da *askRecvMsg()* se chegou uma nova mensagem. Caso afirmativo é chamada a função

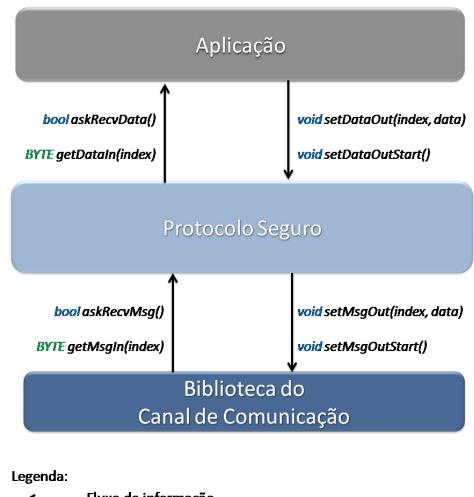


Fig. 1. Arquitetura de implementação em três camadas.

*getMsgIn()* que busca no buffer de saída do Protocolo de Comunicação o octeto correspondente ao *índex* (parâmetro da função). É importante notar que a camada do Protocolo Seguro realiza o ‘desempacotamento’ da PDU, fazendo com que a *getMsgIn()* transfira até 16 octetos enquanto que a *getDataIn()* transfira apenas o *payload* (que tem 12 octetos).

Para envio de pacotes, as chamadas *setDataOut()* e *setMsgOut()* são implementadas na camada de origem do fluxo de dados. Elas servem para enviar o dado ou pacote, *byte-a-byte*, para o buffer de entrada da camada inferior. As sinalizações *setDataOutStart()* e *setMsgOutStart()* informam às camadas inferiores o término, ou seja, que não há mais dados a serem transferidos e que o encapsulamento ou a transmissão podem ser iniciados.

#### C. Mecanismo de detecção por CRC

O uso do CRC por meio de múltiplos níveis, possibilita alto grau de redundância. Primeiramente, ao longo da parametrização inicial do protocolo, é calculado o CRC1 composto de 2 octetos. O CRC que é encapsulado na *Safety PDU* é chamado de CRC2. Ele é formado através de F-Parametros, Dados de saída, *Status/Control Byte* e um Número Consecutivo.

A especificação do protocolo sugere que a implementação do cálculo do CRC seja realizado através do uso de uma tabela. O objetivo dessa técnica é acelerar o cálculo do CRC.

Para trazer mais segurança na técnica de detecção por CRC, esse é recalculado quando o pacote chega ao seu destino e armazenado em uma variável auxiliar que é comparada *byte-a-*

*byte*, com o campo CRC2 presente no pacote. Isso é possível, pois todos os octetos necessários para recalcular o CRC se encontram na própria *Safety PDU* ou já foram estabelecidos na fase inicial de parametrização do PROFIsafe (caso dos F-Parametros).

## VI. RESULTADOS

A tabela I mostra a cobertura de detecção de erros do PROFIsafe tendo como referência os protocolos UDP e TCP. Na tabela pode-se perceber a importância de utilizar um protocolo seguro no desenvolvimento de funções de segurança para aplicações críticas. Isso se deve ao fato da maior gama de cobertura de erros que um protocolo seguro oferece em relação aos protocolos de transporte mais tradicionais.

Tabela I. COMPARATIVO DO PROFISAFE COM PROTOCOLOS TRADICIONAIS

Erros	Protocolo de Comunicação		
	UDP	TCP	PROFIsafe
Repetição sem Intenção	✗	✗	✓
Perda	✗	✓	✓
Inserção	✗	✗	✓
Sequência Incorreta	✗	✓	✓
Corrupção	✗	*	✓
Timeout	✗	✓	✓
Endereçamento	✗	✗	✓
Mascaramento	✗	✗	✓
Falhas Mem.	✗	✗	✓

\* Há detecção de corrupção de segmentos no TCP, porém essa é feita através do mecanismo de Checksum, o qual é mais fraco que o CRC realizado pelo PROFIsafe.

Na depuração da implementação foram realizadas a troca de mensagens entre os dispositivos Mestre e Escravo. Foram detectados erros de timeout e corrupção. A corrupção de mensagens foi feita alterando um byte na *Safety PDU*. Essa alteração no pacote foi percebida pelo *F-Device* por meio de CRC.

## VII. CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi implementado o protocolo de comunicação seguro PROFIsafe conforme a sua especificação. Essa implementação será utilizada no projeto RIO-SIL para o desenvolvimento de uma Remota de I/O, a ser utilizada na implementação de funções de segurança, em aplicações críticas de automação de processos, na área de óleo e gás. Além disso, por exigência do mercado, é necessário que os equipamentos sejam certificados segundo uma norma de segurança. Assim, a comunicação segura entre equipamentos requer o uso de um protocolo seguro de comunicação.

Os testes realizados mostraram que a troca de mensagens entre os dispositivos mestre e escravo ocorreu corretamente. Quando foi inserido um atraso maior do que o valor máximo tolerado pelo protocolo para o recebimento de mensagens, este

foi detectado pelo mecanismo de *timeout*. Ao ser alterado um byte em uma das mensagens trocadas, essa alteração no pacote foi detectada pela verificação do mecanismo de CRC.

Como trabalhos futuros está previsto o teste da implementação através da injecção de falhas com o uso de um injetor de falhas que está em fase de desenvolvimento. O objetivo dos testes é verificar a implementação dos mecanismos de detecção de falhas implementados. Dessa forma, será possível validar corretamente o protocolo implementado e garantir o seu correto funcionamento.

Finalmente, depois de validada a implementação, essa deverá ser integrada ao restante do código da Remota de I/O, onde deverá prover as aplicações de entrada e saída para a implementação de funções de segurança na indústria de petróleo e gás. Durante a realização do trabalho não foram encontrados artigos recentes sobre o PROFIsafe. Isso mostra que se trata de uma área relativamente nova e que academia não tem interesse em implementação de protocolos seguros, denotando que é mais um assunto voltado para indústria.

## REFERÊNCIAS

- [1] Thomesse, J.P. "Fieldbus Technology and Industrial Automation", Emerging Technologies and Factory Automation, pp.651- 653, September 2005.
- [2] Thomesse, J. P. "Fieldbus technology in industrial automation," Proc. IEEE, vol. 93, no. 6, pp. 1073–1101, June 2005.
- [3] FF SYS. Disponível em: < <http://www.fieldbus.org/> > Acessado em: Setembro 2013.
- [4] openSAFETY. Disponível em: < <http://www.open-safety.org/> > Acessado em: Setembro 2013.
- [5] PROFIsafe System Description (2013). Disponível em: < [http://www.profibus.com/nc/download/technical-descriptions-books/downloads/profisaf/af-e-technology-and-application-system-description/down\\_load/9594/](http://www.profibus.com/nc/download/technical-descriptions-books/downloads/profisaf/af-e-technology-and-application-system-description/down_load/9594/) > Acessado em: Setembro 2013.
- [6] EtherCAT FSofE – Safety over EtherCAT Implementation Guide (2010). Disponível em: < [http://www.ethercat.org/pdf/english/ETG5101\\_G\\_D\\_V1111\\_FSoEImplementationGuide.pdf](http://www.ethercat.org/pdf/english/ETG5101_G_D_V1111_FSoEImplementationGuide.pdf) > Acessado em: Setembro 2013.
- [7] International Electrotechnical Commission (2010) "IEC 61508 - Functional Safety Of Electrical/Electronic/Programmable Electronic Safety-Related Systems".
- [8] PI Profibus e Profinet International. Disponível em: < <http://www.profibus.com/home/> >. Acessado em: Setembro 2013.
- [9] International Electrotechnical Commission (2010) "IEC 61784-3 - Functional Safety Fieldbuses - General rules and profile definitions".
- [10] Åkerberg, J.; Reichenbach, F.; Björkman, M. "Enabling safety-critical wireless communication using WirelessHART and PROFIsafe," IEEE Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1 - 8, September 2010.
- [11] Protocolo WirelessHART. Disponível em < [www.hartcomm.org/protocol/wireless\\_technology.html](http://www.hartcomm.org/protocol/wireless_technology.html) >. Acessado em: Setembro 2013.
- [12] Malik, R. and Mühlfeld, R. "A case study in verification of uml statecharts: the profisafe protocol," Universal Computer Science, vol. 9, no. 2, pp. 138-151, February 2003.
- [13] Equipamentos PROFIsafe. Disponível em: < <http://www.profibus.com/technology/profisafe/overview/> >. Acessado em: Setembro 2013.
- [14] Siemens. Disponível em: < <http://www.siemens.com/entry/cc/en/#> >. Acessado em: Setembro 2013.
- [15] WAGO. Disponível em: < <http://global.wago.com/en/overview/index.jsp> >. Acessado em: Setembro 2013.
- [16] BECKHOFF. Disponível em: < <http://www.beckhoff.com/> >. Acessado em: Setembro 2013.

# Engenharia social: explorando o efeito halo para obter acesso físico não autorizado em empresas de cartões de pagamento

Guilherme Gattino\*, Jéferson Campos Nobre<sup>†</sup>

Centro de Informática  
Segurança da Informação  
UNISINOS

São Leopoldo, Rio Grande do Sul

Email: \*ggattino@gmail.com, <sup>†</sup>jcnobre@unisinos.br

**Resumo**—Diversas organizações investem muito dinheiro em controles tecnológicos para proteger as suas informações e controlar os acessos às suas dependências. Porém, conhecer os fatores humanos e saber como induzir falhas nesse aspecto, aprimora as chances de sucesso de um ataque de engenharia social e permite que esses controles sejam facilmente burlados. Este artigo apresenta o uso do viés cognitivo efeito halo como uma maneira de aprimorar ataques de engenharia social que visam trespassar a autorização de visitantes em organizações como os controles de segurança do Padrão de Segurança de Dados do Setor de Cartões de Pagamento implementados.

## I. INTRODUÇÃO

A informação tem uma importante função para as mais diversas corporações, pois é através da informação que elas detêm os seus planos, projetos, valores e demais dados que podem definir o seu sucesso ou o seu fracasso. Proteger a informação contra acesso indevido e evitar que ela seja divulgada e/ou obtida de forma não autorizada, exige a inserção de controles e procedimentos que visam garantir os atributos básicos de segurança da informação, seja através de recursos tecnológicos e/ou processos de negócio.

Esses controles perdem a sua eficiência quando os aspectos comportamentais e psicológicos humanos não fazem parte do escopo da segurança da informação, o que normalmente acontece na maioria das organizações [1]. A negligência desses aspectos pode facilitar que uma pessoa mal intencionada consiga explorar as suas falhas e obter acesso não autorizado às informações ou até mesmo ao interior de uma organização, trespassando as mais poderosas ferramentas tecnológicas de proteção e os seus processos para controle da segurança da informação [2].

A falta de proteção aos aspectos humanos expõe as organizações a riscos reais, pois uma pessoa que possua conhecimento e sucesso em explorar técnicas de engenharia social pode obter e divulgar informações sensíveis e de alto valor para a corporação, além da possibilidade de obter acesso indevido em ambientes físicos e em redes de computadores. Isso só é possível pela existência de inúmeros aspectos psicológicos que podem ser utilizados para induzir pessoas ao erro, forçando-as a divulgar informações, realizar ações que normalmente não fariam e até mesmo mudar as suas opiniões.

## II. A ENGENHARIA SOCIAL E O ASPECTO HUMANO

A metodologia mais comum utilizada pelas organizações para proteger as suas informações é a aplicação de controles tecnológicos, como *hardwares* e *softwares* específicos, investindo altos valores financeiros. Além dessas medidas tecnológicas, algumas organizações decidem por utilizar processos de negócio que visam garantir essa proteção, buscando aplicar políticas, procedimentos, processos e estruturas organizacionais com o objetivo de manter ainda mais efetivos esses controles tecnológicos de segurança da informação.

Mesmo com um alto valor de investimento em controles de proteção da informação, é possível que esses controles sejam simplesmente burlados quando outro aspecto, muitas vezes negligenciado, é explorado por entidades mal intencionadas: o aspecto humano da segurança da informação. Um indivíduo que tenha conhecimento de como forçar falhas dentro desse contexto e explorá-las, pode simplesmente trespassar todos os controles técnicos e obter acesso às informações consideradas confidenciais ou até mesmo obter acesso físico em ambientes considerados críticos [2].

### A. Engenharia social

A engenharia social é definida como a ciência de manobrar habilmente as pessoas através do uso de erros e fraquezas humanas (*i.e.*, vieses cognitivos), para que realizem determinadas ações ou divulguem informações confidenciais [4]. O seu objetivo é obter acesso físico ou lógico não autorizado, além de obter informações sensíveis através de técnicas de manipulação de pessoas ou através de técnicas que as induzem a erros em seus julgamentos e tomadas de decisão.

A forma como as pessoas pensam é um aspecto humano crucial para realizar um ataque de engenharia social. O cérebro humano, quando recebe qualquer informação, tem de processá-la internamente para que seja possível tomar decisões, realizar previsões e julgamentos. Esses processos funcionam na maioria das circunstâncias, porém em certas ocasiões levam a erros sistemáticos [5] e padrões de desvio no julgamento que levam a percepções distorcidas, julgamento impreciso e interpretações ilógicas, conhecidos como viés cognitivo [6].

Com a existência de falhas no processamento de informação no cérebro humano é possível forçar esses erros

para aumentar a probabilidade de sucesso de um ataque de engenharia social. Assim, um ou mais viéses cognitivos podem ser explorados para que um acesso físico seja permitido, mesmo esse não sendo autorizado. Em muitas organizações o controle de acesso físico é realizado através de recepcionistas e guardas patrimoniais, porém isso não impede que outros controles possam ser utilizados para tentar impedir acessos físicos. Em empresas que trabalham com dados de cartão, por exemplo, para controle de acesso físico devem ser aplicadas regras específicas para visitantes (*e.g.*, autorização prévia do visitante antes de sua entrada na empresa, com o objetivo de reduzir as chances de que um acesso não autorizado seja efetuado) [7].

#### B. Viés cognitivo e a segurança da informação

Muitas vezes as pessoas tomam decisões baseadas na crença de probabilidades de que um determinado evento com conclusão incerta possa ocorrer [8]. Como as pessoas possuem um conjunto limitado de conhecimentos que podem ser aplicados para a resolução de problemas [9], muitas tarefas complexas para calcular probabilidades e previsões de valores são reduzidas para simples julgamentos. Esses julgamentos são baseados em dados que possuem uma eficácia limitada, ao qual são processados de acordo com princípios heurísticos. A confiança nesses julgamentos leva a erros sistemáticos, que ocasionam desvios dos mesmos, em interpretações ilógicas, distorções perceptivas ou ações irracionalais [8]. Esses erros são conhecidos como viés cognitivo.

Dentro do contexto da segurança da informação, o viés cognitivo pode ser explorado em alvos humanos, induzindo-os a abrir brechas de segurança, como a divulgação de informações confidenciais, liberação de acessos indevidamente ou a inserção de discos removíveis em uma unidade USB. Um engenheiro social com o conhecimento de como induzir pessoas a desvios nos padrões de julgamentos, percepções e tomadas de decisão (*i.e.*, erro), e com a capacidade de explorar um ou mais viéses cognitivos, pode estar facilitando todo um processo para obter informação ou acesso físico em um ambiente.

Um viés cognitivo que possibilita induzir pessoas ao erro é o efeito halo, que em uma definição geral é a influência que uma avaliação global possui em uma avaliação dos atributos individuais de uma pessoa [10]. Como o efeito halo distorce a avaliação de atributos distintos com base em uma impressão geral, uma pessoa que tenha maiores cuidados com a sua aparência física, ou seja, que sua aparência física seja agradável para o alvo, estes atributos serão superavaliamos pelo expectador. O mesmo pode ocorrer com características inversas, onde uma aparência desagradável ao alvo pode fazer com que outros atributos sejam subavaliados.

A aparência física de uma pessoa é uma característica mais óbvia e acessível para outras pessoas em uma interação social, ao qual permite que, psicologicamente, um indivíduo crie determinadas expectativas simplesmente por conhecer a aparência de outro. Isto ocorre porque existe uma correlação entre características pessoais e a aparência [11]. Além dessa correlação com características pessoais, os estereótipos culturais sobre os tipos de personalidades apropriadas para caracterizar beleza ou feiura podem moldar as características desses

indivíduos, como em culturas onde determinados atributos (*e.g.*, sinceridade, nobreza e honestidade) são definidos como aceitação ou atratividade social.

Quando há a necessidade de tomar decisões e julgamentos, as pessoas utilizam um conjunto de crenças e atitudes que as influenciam, o que eleva a probabilidade de ocorrerem distorções quando essas pessoas estão presenciando algum evento inconsistente ao seu pensamento [12]. A aparência física de uma pessoa é uma característica mais óbvia e acessível para outras pessoas em uma interação social, ao qual permite que, psicologicamente, um indivíduo crie determinadas expectativas simplesmente por conhecer a aparência do outro.

Portanto, quando o efeito halo ocorre, quando um indivíduo enxerga e determina, seja através de padrões pessoais ou culturais, que outro indivíduo é agradável fisicamente, é bastante provável que esta característica seja extrapolada e estendida para outras características pessoais e independentes [13]. Dessa forma, o efeito halo pode ser explorado para obter acesso físico não autorizado, forçando que uma determinada característica física (*i.e.*, aparência física) seja estendida para outras características independentes (*e.g.*, hierarquia ou confiabilidade) e um possível bloqueio de acesso por autorização seja trespassado.

### III. EXPERIMENTO

Para analisar e comprovar que o efeito halo pode ser utilizado para burlar especificamente o controle 9.3.1 do PCI-DSS, que estabelece um processo para autenticação e autorização de visitantes, foi realizado um experimento. Esse experimento tem uma abordagem mais próxima aos experimentos da área da psicologia e busca analisar, através de tentativa de extrapolar uma avaliação da aparência física para outras características, se o efeito halo pode ou não influenciar na decisão de pessoas para permitir um acesso sem autorização. Como é bastante difícil simular em um ambiente controlado todos os fatores reais de uma organização certificada pelo PCI-DSS e que possua um controle efetivo de autorização de visitas, para a realização do experimento foi utilizado um questionário para obtenção de dados.

#### A. Metodologia

O experimento iniciou-se com a seleção de duas pessoas, com boas expressões faciais, para serem utilizadas como modelos onde duas fotografias de cada uma dessas pessoas foram capturadas. Para a primeira fotografia, os modelos foram instruídos a criar uma expressão de autoridade e arrogância, além de serem vestidos com roupas mais sociais, mais próximas de que um alto executivo utilizaria. Para a segunda fotografia, os modelos foram instruídos a se mostrarem com expressões faciais mais carismáticas e utilizaram roupas mais informais.

Para cada uma das fotografias, um conjunto de questionamentos foi formulado com o objetivo de classificar os atributos da pessoa nessa fotografia. Foram classificadas as características da aparência física, a confiabilidade, o conhecimento, a simpatia e a influência hierárquica, baseando-se apenas na fotografia apresentada. Após esta avaliação foi solicitado ao voluntário do experimento se ele permitiria ou não que a pessoa obtivesse acesso ao interior da organização. Em seguida, o efeito halo foi medido, ao classificar o grau

de influência que cada característica teve para a tomada de decisão. Antes dos questionamentos, o voluntário foi instruído sobre o controle 9.3.1 do PCI-DSS, através da leitura de um texto previamente preparado, e explicado que o acesso de visitantes só é permitido após uma autorização formal de algum funcionário da empresa.

Para responder os questionamentos, foram selecionados 30 voluntários, os quais foram divididos em dois grupos com base em suas profissões. O primeiro grupo, definido como grupo A, foi composto por 15 voluntários especializados em segurança da informação, o que inclui profissionais da área e estudantes do curso de Segurança da Informação da Universidade do Rio do Vale dos Sinos (UNISINOS). O segundo grupo, definido como grupo B, foi composto por profissionais de segurança patrimonial, o que inclui porteiros, guardas patrimoniais e recepcionistas.

### B. Resultados

Os resultados apresentados através da coleta de dados deste experimento são mostrados através de gráficos, com o objetivo de um melhor entendimento de seus significados. Foram realizadas várias séries de comparações buscando identificar a ação do efeito halo em cada um dos modelos fotografados.

A Figura 1 apresenta um gráfico com o total das classificações realizadas pelos voluntários ao analisar as fotografias com expressão autoritária. Este gráfico é a soma de todos os resultados obtidos com as fotografias com a expressão autoritária com os dois modelos. Na análise dos dados dessa soma, percebeu-se que há uma grande quantidade de voluntários que avaliou o autoritarismo com um grau mediano de características. Destacaram-se as características de confiabilidade, simpatia e influência hierárquica que foram avaliadas com esses valores por 60% dos voluntários.

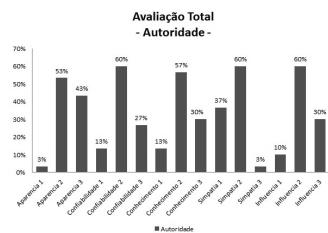


Figura 1. Avaliação total das características dos modelos com expressão autoritária

O próximo gráfico, ilustrado na Figura 2, mostra a soma das classificações realizadas pelos voluntários ao analisar as fotografias com a expressão carismática. Com essa análise, foi possível identificar que a grande maioria das características foram classificadas como medianas (valor equivalente a 2), com exceção da simpatia que foi classificada por 63% dos voluntários com um valor alto.

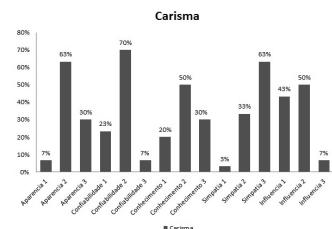


Figura 2. Avaliação total das características dos modelos com expressão carismática

Foi realizada análise para verificar a quantidade de voluntários que permitiu que os modelos nas fotografias acessassem a organização sem uma autorização formal. Mesmo sendo instruídos a solicitar uma autorização prévia antes de permitir o acesso, 50% de todos os voluntários permitiu que o modelo autoritário entrasse na organização. Da mesma maneira, 37% dos voluntários permitiram o acesso do modelo carismático sem uma autorização. É importante notar que, mesmo que em nenhuma das duas expressões as avaliações de suas características tenham sido avaliadas com um valor alto, houve voluntários que permitiram o acesso. A Figura 3 identifica em forma de gráfico a porcentagem de acessos permitidos sem autorização em comparação por tipo de expressão (autoridade versus carisma).

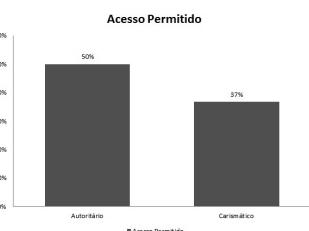


Figura 3. Avaliação total dos acessos permitidos, comparados por expressão (autoritário versus carismático)

Para analisar a eficácia de pessoas especializadas em segurança da informação e que possuam determinado conhecimento dos riscos em que a empresa pode estar exposta quando não há o cumprimento das normas impostas, foi realizada uma comparação entre os dois grupos de voluntários. A comparação foi realizada levando em consideração o total de acessos permitidos ao somarem-se os resultados de acessos permitidos das duas expressões dos modelos. Como o esperado, o gráfico da Figura 4 mostra que 57% dos voluntários especializados em proteção patrimonial (grupo B) permitiram o acesso indevido ao interior da empresa, mesmo que uma instrução fosse ordenada antes do questionário ser iniciado. Mas é surpreendente que 30% dos voluntários especializados em segurança da informação (grupo A) permitiram que o acesso fosse realizado.

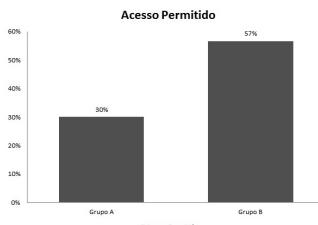


Figura 4. Total de acessos permitidos, comparados por grupo

Finalmente, foi realizada análise da influência que as características avaliadas pelos voluntários tiveram para determinar a sua decisão em permitir ou de não permitir o acesso físico. Para isso, foram consideradas as respostas marcadas com o valor 2 e 3 para cada uma das características, visto que apenas o valor 1 representa nenhuma influência. O gráfico representado na Figura 5 indica que há uma significativa influência da autoridade na tomada de decisão dos voluntários para permitir um acesso. As fotografias de modelos com expressão autoritária passaram confiança para 77% dos voluntários, assim como 67% tiveram suas decisões influenciadas pela aparência e influência julgadas apenas por essas fotografias.

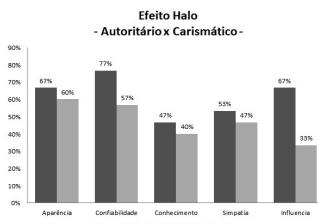


Figura 5. Efeito halo medido por característica avaliada

De acordo com os gráficos apresentados e com os resultados obtidos neste experimento, ficou evidenciado que o efeito halo pode ser explorado para obter acesso físico não autorizado, mesmo que seja exigido um controle de aprovação para liberação deste acesso. Dessa forma, uma pessoa mal intencionada que consiga fazer com que sua aparência seja mais autoritária e que consiga refletir um padrão de vestimentas utilizado pela alta direção da empresa, terá maiores probabilidades de trespassar por esse controle.

#### IV. CONCLUSÃO

Apesar da existência de padrões de segurança das informações e desses padrões apresentarem uma ampla gama de controles, que nem sempre são técnicos, existem chances consideráveis para que pessoas cometam erros comuns na tomada de suas decisões, em suas memórias, em suas crenças e nos seus comportamentos. Estes erros, conhecidos como viés cognitivo, ocorrem em determinadas situações e quando

confrontados com o objetivo de obter informações ou acesso não autorizado trespassam os controles técnicos e os mais tecnológicos sistemas de proteção de acesso. Com isso, controles muito bem implementados podem simplesmente não ter o efeito defensivo esperado, pois a falha explorada foi humana, foi um viés cognitivo.

O experimento realizado obteve êxito em demonstrar que o viés cognitivo efeito halo pode ser explorado para burlar um processo de autorização de visitantes e ficou evidente que pessoas que não possuem uma consciência ou especialidade com a segurança da informação são mais suscetíveis a serem enganadas pelas consequências do efeito halo. A aparência física e as expressões faciais são as principais características que levam à exploração desse viés cognitivo, fazendo com que o alvo estenda o seu julgamento nessas características para outras características distintas. Explorando o efeito halo em conjunto com um pretexto para a entrada na organização (e.g., fazendo-se passar por um técnico de informática ou um consultor) as chances de sucesso são ainda mais aprimoradas.

#### REFERÊNCIAS

- [1] Y. Lafrance. (2004, Fev.) *Psychology: a precious security tool*. [Online]. Disponível em <http://www.sans.org/reading-room/whitepapers/engineering/psychology-precious-security-tool-1409>. [Acesso em 18 de Março de 2012]
- [2] I. Mann. (2011) *Engenharia social*. Série Prevenção a Fraudes. São Paulo: Blucher.
- [3] T. Thornburgh. (2004, Set.) *Social engineering: the dark art*. In *Of the first Annual Conference on Information Security Curriculum Development*. Kenessaw, Georgia. Nova Iorque: ACM. pp.133-135.
- [4] A. Thapar. (2007, Jun) *Social engineering: an attack vector most intricate to tackle*. [Online]. Disponível em [http://www.infosecwriters.com/text\\_resources/pdf/Social\\_Engineering\\_ATThapar.pdf](http://www.infosecwriters.com/text_resources/pdf/Social_Engineering_ATThapar.pdf). [Acesso em 16 de Março de 2012]
- [5] G. Gigerenzer. (1991) "How to make cognitive illusions disappear: beyond heuristics and biases", *European Review of Social Psychology*. Vol. 2. pp.83-115.
- [6] J. Taylor. (2011, Jul). *Understanding cognitive bias*. [Online]. Disponível em [https://www.huffingtonpost.com/dr-jim-taylor/cognitive-biases\\_b\\_896421.html](https://www.huffingtonpost.com/dr-jim-taylor/cognitive-biases_b_896421.html). [Acesso em Agosto de 2012]
- [7] PCI Security Standards Council. (2010, Out) *Navegando pelo PCI DSS: conhecer a intenção dos requisitos — Versão 2.0*.
- [8] A. Tversky e D. Kahneman. (1974) "Judgement under uncertainty: heuristics and biases", *Sciences*. Vol. 185. No. 4157. pp.1124-1131.
- [9] J. Firmino e T. Broto.(2009) "Raciocínio, heurísticas e resolução de problemas: um diálogo teórico-conceitual", *Mosaico: estudos em psicologia*. Vol. 3. No. 1. pp.1-12.
- [10] R. Nisbett e T. Wilson. (1977) "The halo effect: evidence for unconscious alteration of judgments", *Journal of Personality and Social Psychology*. Vol. 35 No. 4. pp.250-256.
- [11] K. Dion, E. Berscheid e E. Walster. (1972) "What is beautiful is good". *Journal of Personality and Social Psychology*. Vol. 24. No. 3. pp.285-290.
- [12] L. Leuthesser, C. Kohli e K. Harich. (1995) "Brand equity: the halo effect measure", *European Journal of Marketing*. Vol. 29. pp.57-66.
- [13] C. Hadnagy. (2011). *Social engineering: the art of human hacking*. Indianapolis: Wiley Publishing.

---

||

## **Sessão 2 - Trabalhos de IC**

---



# Modulação e codificação adaptativos através de Software Defined Radio

Matheus Eidt<sup>1</sup>, Lauro Culau<sup>1</sup>, Matias Schimuneck<sup>1</sup>, Rafael Nicolay<sup>1</sup>,  
Roberto Schmidt<sup>1</sup>, Cristiano Both<sup>1</sup>, Maicon Kist<sup>2</sup>

Universidade de Santa Cruz do Sul<sup>1</sup>, Universidade Federal do Rio Grande do Sul<sup>2</sup>  
{matheus, laurotc, matiass, rafaelrodrigo, rmainardi}@mx2.unisc.br,  
cboth@unisc.br, maicon.kist@inf.ufrgs.br

**Resumo**—Devido a crescente demanda por maiores taxas de transmissão, tem-se buscado otimizar equipamentos de rádio alterando suas características operacionais (como, por exemplo, a modulação e a codificação). A utilização de uma modulação e codificação padrão pode causar, tanto perda de dados, quanto a má utilização da taxa de transferência disponível no canal de transmissão. De modo a resolver este problema, uma das soluções é a utilização da tecnologia de *Software Defined Radio* (SDR). Essa tecnologia possibilita que os equipamentos de rádios tornem-se elementos dinâmicos, capazes de adaptarem suas características operacionais através de *software* e assim selecionar a modulação e codificação que obtenham o melhor desempenho do canal. Para que isso seja possível, é necessário que se defina a melhor configuração a ser utilizada a cada transmissão. Este trabalho propõe uma abordagem para um seletor de modulação e codificação para dispositivos de rádio, através de um *Software Defined Radio* que analisa a qualidade do canal nas tomadas de decisões. Os resultados obtidos demonstram uma melhoria considerável em comparação a um algoritmo que não utiliza modulação e codificação adaptativos.

## I. INTRODUÇÃO

*Software Defined Radio* (SDR) permite que um rádio passe a ser um elemento dinâmico, capaz de alterar suas características operacionais, como largura de banda, modulação e codificação, de acordo com as configurações do *software* [1]. Além disso um SDR, por processar o sinal digitalmente, ao contrário de um rádio convencional, que, usualmente, processa o sinal analogicamente, permite uma melhor manipulação e processamento do sinal. A modulação e codificação adaptativas são técnicas onde parâmetros de transmissão do sinal são dinamicamente adaptados às condições de variação do canal [2]. A combinação entre o tipo de modulação e a taxa de codificação irá gerar um esquema chamado de *Modulation and Coding Scheme* (MCS) [3].

O SDR deve ser capaz de gerenciar esquemas de MCS de forma adaptativa, ou seja, esses dispositivos devem ser capazes de utilizar tanto modulações mais robustas, quanto as menos robustas, de acordo com as informações do canal. Entretanto, para realizar esse gerenciamento, é necessário um módulo, capaz de identificar qual MCS é o mais apropriado para a transmissão em determinada frequência. Além disso, é necessário o desenvolvimento de soluções de baixa complexidade e de custo reduzido.

Diversos trabalhos encontrados na literatura propõem sistemas com modulação e codificação adaptativos em SDRs. O estudo de Xia, Zhou e Giannakis [4], propõe um transmissor adaptativo MIMO-OFDM (*Multi-Input Multi-Output - Orthogonal Frequency Division Multiplexed*), aplicando um processamento no vetor de sinais, capaz de gerar interferências, em cada subportadora OFDM. No trabalho de Souryal [5], é proposta uma estimativa das condições do canal, de modo a melhorar o desempenho do alcance do sinal, sem reduzir a

taxa de transmissão oferecida. Souryal assegura que ao variar a modulação é possível que canais tenham mais bits alocados.

Tendo em vista que a literatura não oferece implementações de SDRs em dispositivos reais, este trabalho propõe-se a definir corretamente o MCS, através da análise da qualidade do canal, com a finalidade de otimizar a transmissão do dispositivo de rádio. A partir disto, foi desenvolvido e avaliado um seletor de modulação e codificação adaptativo para SDRs, que é o responsável por, através da avaliação do canal, definir qual a modulação e codificação deve ser utilizada na transmissão.

O restante desse trabalho está organizado da seguinte forma: Na Seção II, são expostos os conceitos que englobam a proposta de SDR. Na Seção III, são apresentadas as propostas encontradas na literatura que abordam sistemas com modulação e codificação adaptativos em SDRs. A Seção IV é apresentada a proposta e o protótipo desenvolvido do seletor, descrevendo sua arquitetura. Na Seção V é descrita a metodologia de avaliação do módulo seletor adaptativo. 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 Radio

SDR permitem que os equipamentos de rádio sejam reconfigurados via *software*. A capacidade de reconfiguração é útil para aprimorar as funções do rádio, sem que seja necessário alterar o *hardware* desse equipamento [6]. Um SDR realiza diversos processamentos de sinal em um computador de propósito geral ou em uma plataforma de *hardware*. Em um rádio que implementa SDR, muitas das manipulações e processamentos de sinal, são feitas em *software*, ao invés de *hardware*. Assim, o sinal é processado digitalmente, ao contrário de um rádio convencional, que, usualmente, processa o sinal analógicamente.

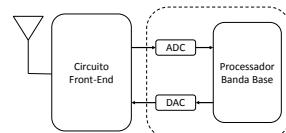


Figura 1. Diagrama de blocos SDR

Para realizar a digitalização do sinal, é necessário um dispositivo conversor, chamado *Analog to Digital Converter* (ADC). O diagrama de blocos básico de um SDR é apresentado na Figura 1. O processo de conversão ADC, é realizado após o circuito *Front End* (FE), que é usado para converter o sinal para frequências mais baixas, chamadas de *Intermediate*

*Frequency* (IF). O ADC tem a função de digitalizar e enviar o sinal para o processador banda base, onde são realizados todos os processos como demodulação, codificação do canal entre outros [7]. Em um rádio convencional, todos esses processos são realizados em *hardware*.

#### A. Modulation and Coding Scheme

O MCS é um termo utilizado para descrever a combinação do esquema de modulação da portadora e o esquema de codificação utilizado quando se transmitem dados [8]. As subportadoras de dados são codificadas utilizando técnicas *Forward Error Correction* (FEC) antes de ser transmitidas. Essas técnicas consistem em adicionar aos dados, informação redundante, que permitem a detecção e correção de erro pelo dispositivo receptor. Após a codificação FEC, os bits de dados são mapeados em uma constelação de modulação, que é formada por símbolos que definem a quantidade de dados que podem ser transmitidos. A combinação entre uma constelação de modulação e uma codificação FEC produz uma configuração MCS. O número de símbolos de uma constelação de modulação, relaciona-se com a quantidade de dados que podem ser transmitidos em cada símbolo da modulação.

Cada tecnologia define suas configurações MCS. As configurações MCS para multiplexação OFDMA (*Orthogonal Frequency Division Multiple Access*), dependem da constelação de modulação e taxas de codificação FEC utilizadas. Quanto mais robusto for o MCS utilizado, há uma maior confiabilidade da transmissão, porque há uma maior capacidade para detectar e corrigir erros. Entretanto, a taxa de dados da rede é reduzida. Desse modo, é importante escolher corretamente o MCS, para que não exista uma transmissão redundante de dados, sem necessidade. Essa transmissão de dados redundantes é conhecida como *coding rate*, que significa a quantia de dados redundantes transmitidos para cada bit de dados. Dessa maneira, um *coding rate* 1/2, utiliza 1 bit de redundância a cada 2 bits de dados.

Nos sistemas de rádio tradicionais, a modulação e a codificação são projetadas para o pior cenário. Assim, na maioria das vezes, são utilizados modulações e codificações mais robustas do que o necessário. Para que isso não ocorra, uma das opções é a utilização de sistemas com modulação e codificação adaptativos, ou *Adaptive Modulation and Coding* (AMC). Nos sistemas AMC, é necessária uma constante monitorização das condições do canal. A ideia central dos sistemas AMC, é adaptar dinamicamente o MCS, com o objetivo de adequar a eficiência espectral global às condições do canal [9].

Para estimar a condição do canal, foi definida a utilização da *Carrier to Noise Ratio* (C/N). C/N é a taxa de energia relativa em relação ao ruído de um sistema, que permite analisar se uma portadora ainda pode ser reconhecida ou se foi destruída por ruídos. O essencial para o presente trabalho é que a C/N provê um valor para a qualidade de um canal de comunicação.

A partir de dados que estimam as condições do canal, como o C/N, o sistema deve ser capaz de reconhecer o nível de modulação e codificação que deverá ser utilizado, bem como o nível de redundância das informações que devem ser enviados. Desse modo, em um canal com alto C/N, não é necessário enviar os dados com muita redundância, visto que, dificilmente os dados são perdidos durante a transmissão. Entretanto, caso um canal tenha um C/N baixo, é necessária uma taxa maior de redundância de dados, para que a perda de dados durante a transmissão seja reduzida. A proposta deste trabalho, é que o rádio seja reconfigurado via *software*, ou seja, as características operacionais do rádio são alterados através de *software*, utilizando SDRs.

### III. TRABALHOS RELACIONADOS

A utilização de sistemas de MCS adaptativo são importantes para adequar a transmissão de acordo com as condições do canal. No trabalho de Souryal [5], é proposta uma estimativa das condições do canal, de modo a melhorar o desempenho do alcance do espalhamento, sem reduzir o *bit rate* oferecido. Com a utilização de modulação adaptativa, os *bits* podem ser alocados de forma a maximizar o número de *bits* transmitidos ou minimizar a probabilidade global de erros. Em canais de frequência seletiva, alguns subcanais apresentam um enfraquecimento maior de sinal, enquanto outros apresentam uma atenuação relativamente insignificante. Ao variar a modulação dos subcanais, Souryal assegura que os subcanais com ganho favorável, tenham mais *bits* alocados do que os canais com atenuação mais profunda.

Através da análise de resultados, Souryal observa que OFDM adaptativo é mais sensível a erros de estimativa do canal do que OFDM com modulação uniforme. Entretanto, a vantagem do OFDM adaptativo continua significante, mesmo na presença de erros de estimativa de canal, para canais com variação relativamente lenta. Por outro lado, o impacto de erros de estimativa da condição de canais com variação mais rápida, pode ser significante.

A utilização de OFDM é necessária neste trabalho, para reduzir erros de desvanecimento ou interferência, além de permitir o uso do espectro de forma paralela. No trabalho de Wang [10], um sistema OFDM com múltiplos usuários é utilizado para transmissão de vídeos através da estação base. Neste trabalho a base conhece as informações da condição do canal (*Channel State Information* (CSI)), bem como a taxa de distorção dos fluxos do vídeo, e busca alocar recursos do espectro para os usuários de acordo com essas informações. Wang utiliza apenas os conceitos de OFDM para transmissão do vídeo, enquanto este trabalho, busca utilizar diversas outras técnicas, além de se preocupar com a implementação do sistema proposto em um dispositivo de rádio. Com base nos conceitos e trabalhos existentes, a próxima seção apresenta a arquitetura do módulo implementado.

### IV. MÓDULO SELETOR ADAPTATIVO

O Módulo Seletor Adaptativo é projetado para SDR, isto é, a abordagem chave está no fato de que a solução apresentada, descreve a implementação deste seletor em um dispositivo de rádio definido por *software*. Além disso, é proposto um receptor para realizar transmissão e recepção completas. A seguir descreve-se a arquitetura do Seletor, e os mecanismos para escolha de canais.

#### A. Módulo Transmissor

As etapas do sistema transmissor são apresentadas na Figura 2. Inicialmente, uma lista de canais pré-definidos é carregada. O transmissor deve verificar se o primeiro canal dessa lista está livre para transmissão. Caso o canal esteja ocupado, a lista é percorrida em sequência, até que um canal livre seja encontrado. Quando um canal livre é encontrado, o transmissor verifica qual MCS deverá ser utilizado. Na primeira leitura, o valor do C/N lido é armazenado na memória do transmissor. Nas leituras seguintes, o valor do C/N é inicialmente comparado com o valor armazenado em memória. Caso esses valores sejam diferentes, o transmissor verifica na Tabela I qual MCS deve ser aplicado aos dados e armazena o novo valor do C/N na memória. Caso não exista uma alteração no valor do C/N, o MCS é mantido.

Indice MCS	Modulação	Coding rate	Bits por símbolo	C/N threshold (dB)
0	QPSK	1/2	2	50
1	QPSK	3/4	2	45
2	16-QAM	1/2	4	40
3	16-QAM	3/4	4	35
4	64-QAM	1/2	6	30
5	64-QAM	2/3	6	25
6	64-QAM	3/4	6	20

Tabela I  
RELAÇÃO DO *threshold* DO C/N COM O VALOR DO MCS

Após a aplicação do MCS nos dados, o transmissor envia, através de um canal de controle, o canal e o MCS que deverão ser utilizados pelo receptor, de forma a receber os dados corretamente. A transmissão na rede sem fio é então realizada. O transmissor deve, a cada período pré-determinado (2 segundos), interromper o envio de dados para verificar se o canal em que transmite ainda está livre. Dessa forma, o sistema retorna à etapa de verificação do canal. Essa interrupção deve ocorrer pois um usuário não pode ocupar um canal em que não possua licença para uso caso o usuário licenciado esteja utilizando. Dessa maneira, a cada período pré-determinado (1 segundo) é necessário sensoriar o canal para verificar se o usuário licenciado não está transmitindo. É importante ressaltar que as transmissões das informações de qual canal e qual MCS deverão ser utilizados pelo receptor, é realizada por um canal de controle. Dessa maneira, essas transmissões não interferem nas transmissões de dados do canal sem fio.

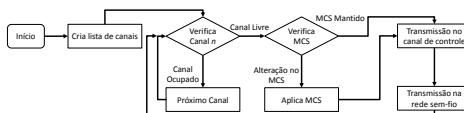


Figura 2. Fluxograma do sistema transmissor

### B. Módulo Receptor

O sistema receptor, opera de maneira diferente do receptor. Inicialmente, o receptor aguarda a recepção de comandos pelo canal de controle. Quando um comando é recebido, o receptor verifica se deve alterar o canal em que busca os dados. Em caso afirmativo, o canal é alterado. Na sequência, o receptor verifica se o MCS que deve utilizar é diferente do que já está sendo utilizado, com base nos dados da Tabela I. Caso a recepção seja a primeira, o MCS deve ser necessariamente alterado. O índice do MCS utilizado é, então, salvo na memória do receptor. Nas transmissões futuras, esse valor é comparado com o índice recebido do transmissor. Caso seja necessário, o MCS é alterado. Por fim, o receptor inicia a recepção dos dados no canal sem fio.

A recepção no canal sem fio é interrompida sempre que o receptor recebe comandos no canal de controle. Desta maneira, quando existe recepção no canal de controle, o sistema volta ao início do fluxo.

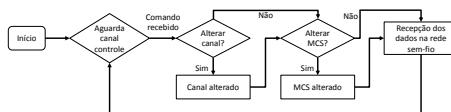


Figura 3. Fluxograma do sistema receptor

## V. AVALIAÇÃO DO MÓDULO

Esta seção apresenta detalhes da metodologia utilizada para avaliação da implementação do Seletor de MCS. O foco dessa metodologia é identificar o impacto do seletor adaptativo no desempenho da transmissão referente, tanto a perda de dados, quanto a má utilização da taxa de transferência disponível no canal de transmissão. A seguir, apresenta-se o ambiente de testes utilizado para realização dos experimentos e o cenário avaliado.

### A. Ambiente de Experimentação

Para realizar os experimentos em um cenário real, foi utilizado o USRP2[11], que permite o desenvolvimento e implementação de sistemas flexíveis de SDR. O USRP2 é, idealmente, projetado para o desenvolvimento de aplicações como protótipos de camada física, acesso dinâmico de espectro e rádios cognitivos, que necessitam alto desempenho de rádiofrequência e grande largura de banda.

Também utilizou-se a ferramenta GNU Radio, que auxilia no desenvolvimento de *software*, possibilitando a implementação de SDRs. É uma ferramenta de software livre e código aberto e fornece diversos blocos de processamento de sinais. Pode ser utilizado com *hardware* de rádiofrequência externo de baixo custo, possibilitando a criação de SDRs, ou pode ser utilizado sem *hardware*, através de um ambiente de simulação.

### B. Cenário

Para testar o comportamento do seletor, os módulos de transmissão e recepção foram executados em dois USRP, cada um com uma rede semi-fio operando em 2,4 GHz, permitindo a comunicação completa entre os dispositivos. O USRP responsável pela transmissão, verifica a qualidade do canal e atribui um MCS aos dados. O transmissor informa ao receptor qual canal e qual MCS deve ser utilizado e, em seguida, envia os dados. O receptor utiliza o MCS correto e qualifica os dados como corretos ou não.

Os módulos de transmissão e recepção estão distantes entre si 20 cm, em ambiente interno. Considerando esse ambiente, os módulos estarão sujeitos a todos os tipos de interferência. Entretanto, devido à pequena distância entre os dois módulos, a atenuação do sinal (*path loss*) e a propagação por diferentes caminhos (*multi-path*), devido à distância podem ser desconsideradas. A transmissão entre os dois módulos tem duração de 2 segundos, sendo interrompida durante 1 segundo para que o canal seja sensoriado. Esse processo (transmissão + sensoriamento), é realizado durante 60 segundos. Os pacotes enviados possuem tamanho de 64 KiloBytes (KB).

No primeiro cenário, é realizado o envio de dados sem MCS, e, no segundo cenário, utiliza-se os diferentes MCS's propostos neste artigo. A partir desse cenário de teste, é possível identificar o correto funcionamento do sistema proposto. Em seguida, são realizados dois cenários de teste para comparar a quantidade de pacotes recebidos corretamente na implementação sem nenhum MCS com a implementação com MCS adaptativo deste artigo. Além disso, outro cenário leva em consideração o *throughput*, que é a taxa de vazão de dados da implementação. Na seção seguinte, são apresentados os resultados e os testes realizados no cenário proposto.

## VI. RESULTADOS

Nesta seção são apresentados os resultados obtidos após uma série de repetições dos experimentos, para garantir 95% de confiança. Os resultados refletem o cenário definido na Seção V. Foram realizadas dez medições do *throughput* para

cada cenário. A Figura 4 ilustra, no eixo vertical, o *throughput*, em Kilobytes por segundo (KB/s), e, no eixo horizontal, o número de medições realizadas. A partir da análise da Figura 4, verifica-se uma grande diferença entre uma implementação que não utiliza MCS adaptativo. A vazão média dos dados na implementação realizada neste artigo ficou, em média, 3 KB/s acima da implementação sem MCS adaptativo.

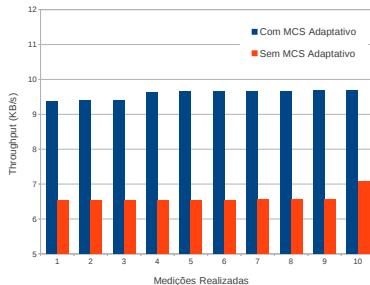


Figura 4. Resultados de *throughput*

Já na Figura 5, é realizada a análise dos cenários de teste para comparar a quantidade de pacotes recebidos corretamente. O eixo vertical representa a porcentagem de pacotes corretos recebidos. Já o eixo horizontal apresenta o número de medições realizadas. Desta maneira, é possível comparar os resultados obtidos nesta implementação com uma transmissão sem MCS. Comparando os dados da Figura 5, a porcentagem de dados recebidos corretamente é superior à implementação sem MCS adaptativo. Isso ocorre pelo fato de a implementação adaptativa avaliar a condição do canal e enviar um MCS apropriado a essa condição, enviando mais quadros e, consequentemente, possuindo um *throughput* mais alto. Além disso, a porcentagem de dados corretos também é um fator relevante, uma vez que, com o MCS apropriado, uma quantidade menor de dados é perdida.

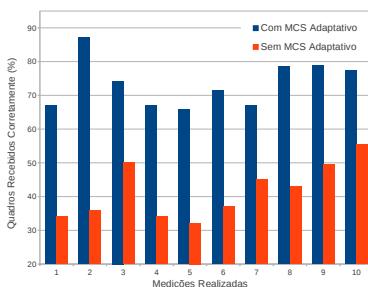


Figura 5. Porcentagem de quadros recebidos corretamente  
VII. CONCLUSÃO E TRABALHOS FUTUROS

O seletor de MCS adaptativo para SDR é uma técnica que permite alterar, conforme a necessidade, a codificação e a modulação utilizada para a transmissão de dados em um dispositivo de rádio. Essa adaptação permite, que com a análise de C/N do canal, diferentes tipos de MCS sejam utilizadas na

transmissão de dados. A proposta deste trabalho permite um melhor aproveitamento do canal de transmissão, reduzindo a má utilização da taxa de transferência, devido a utilização de uma modulação e codificação mais adequada ao canal diminuindo, assim a perda de dados durante a transmissão.

Definida a abordagem, implementou-se o módulo de seleção adaptativa para MCS para um sistema de transmissão utilizando rádio. Realizando experimentos em um ambiente real, foi possível analisar a qualidade dos canais e, com base nessa informação, foram tomadas as decisões sobre qual seria o melhor tipo de modulação e codificação a ser utilizada no canal. Os resultados mostraram que a utilização de um seletor para o MCS que se adapta ao canal ao invés de um padrão fixo, permitiu um melhor desempenho nas transmissões dos equipamentos testados. Com isso, conclui-se que a proposta pode trazer benefícios para a comunicação via rádio, uma vez que a vazão de dados é maior que em uma proposta de MCS fixo. Além disso, através dessa implementação, a taxa de dados recebidos corretamente é superior à implementação de modulação e codificação fixos, mesmo que mais dados sejam transmitidos. Dessa forma, verificou-se que a solução via SDR se mostrou adequada ao propósito.

Um trabalho futuro consiste em transmitir informações do receptor para o transmissor. Dessa forma, seria possível resolver um dos principais problemas enfrentado por esta implementação: quando receptor e transmissor estiverem em distâncias maiores, diferentes níveis de ruído serão encontrados no transmissor e no receptor. Dessa maneira, caso o receptor informe ao transmissor as características do canal, o MCS poderá ser atribuído levando em consideração essa informação, melhorando a recepção dos dados.

## REFERÊNCIAS

- [1] A. L. G. Reis, A. Barros, K. Gusso Lenzi, L. Pedroso Meloni, and S. Barbin, "Introduction to the software-defined radio approach," *IEEE (Revista IEEE America Latina) Latin America Transactions*, vol. 10, no. 1, pp. 1156–1161, 2012.
- [2] R. Kunst, C. B. Both, L. Z. Granville, and J. Rochol, "On the impact of hybrid errors on mobile wiimax networks," *Computer Networks*, vol. 55, no. 16, pp. 3659–3671, Nov. 2011. [Online]. Available: <http://dx.doi.org/10.1016/j.comnet.2011.04.018>
- [3] Motorola, "Adaptive modulation and coding (amc)," 2000, disponível em: <[http://www.3gpp.org/ftp/tsg\\_ran/WG1\\_RL1/TSGR1\\_17/docs/PDFs/R1-00-1395.pdf](http://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_17/docs/PDFs/R1-00-1395.pdf)>. Acesso em: setembro 2013.
- [4] P. Xia, S. Zhou, and G. Giannakis, "Adaptive mimo-ofdm based on partial channel state information," *IEEE Transactions on Signal Processing*, vol. 52, no. 1, pp. 202–213, 2003.
- [5] M. Souryal and R. Pickholtz, "Adaptive modulation with imperfect channel information in OFDM," *IEEE International Conference on Communications*, vol. 6, pp. 1861–1865 vol.6, 2001.
- [6] M. Islam, M. Hannan, S. Samad, and A. Hussain, "Bit-error-rate (BER) for modulation techniques using software defined radio," *International Conference on Electrical Engineering and Informatics*, vol. 02, pp. 445–447, 2009.
- [7] A. Marwanto, M. Sarjari, N. Fisal, S. K. S. Yusof, and R. Rashid, "Experimental study of ofdm implementation utilizing GNU Radio and USRP - SDR," *IEEE 9th Malaysia International Conference on Communications (MICC)*, pp. 132–135, 2009.
- [8] Lever, "MCS," 2013, disponível em: <[http://telecoms-training.co.uk/dict\\_term\\_01.php?term=MCS](http://telecoms-training.co.uk/dict_term_01.php?term=MCS)>. Acesso em: setembro 2013.
- [9] J. Yang, N. Tin, and A. Khandani, "Adaptive modulation and coding in 3g wireless systems," *IEEE 56th Vehicular Technology Conference*, vol. 1, pp. 544–548 vol.1, 2002.
- [10] D. Wang, L. Toni, P. Cosman, and L. Milstein, "Uplink resource management for multiuser ofdm video transmission systems: Analysis and algorithm design," pp. 1–14, 2013.
- [11] ETTUS RESEARCH, "USRP2," 2013, disponível em: <<https://www.ettus.com>>. Acesso em: setembro 2013.

# Sensoriamento espectral por detecção de energia utilizando aprendizagem por reforço

Rafael Rodrigo Nicolay<sup>1</sup>, Lauro Tremea Culau<sup>1</sup>, Matheus Eidt<sup>1</sup>,  
Matias Schimuneck<sup>1</sup>, Roberto Schmidt<sup>1</sup>, Maicon Kist<sup>2</sup>, Cristiano Both<sup>1</sup>

<sup>1</sup>Universidade de Santa Cruz do Sul

<sup>2</sup>Universidade Federal do Rio Grande do Sul

{rafaelrodrigo, laurotc, matiass, matheus, rmainardi}@mx2.unisc.br,  
maicon.kist@inf.ufrgs.br, cbboth@unisc.br

**Resumo**—A demanda por recursos de rede sem fio tem aumentado consideravelmente nos últimos anos. A alocação espectral de forma estática, usada atualmente, não conseguirá atender essa alta demanda nos próximos anos. Uma alternativa para o problema causado pela alocação espectral de forma estática é a adoção de Rádios Cognitivos. Estes dispositivos são capazes de analisar e se adaptar ao estado atual de utilização do espectro de radiofrequências, utilizando temporariamente canais livres de forma oportunista e dinâmica em suas comunicações. Para que o acesso oportunitista ocorra, é necessário garantir a ociosidade do canal de comunicação que será usado pelo dispositivo, de forma a evitar quaisquer interferências danosas a outros usuários. A descoberta de canais ociosos no espectro de radiofrequências é feita através do sensoriamento espectral. Este processo consiste na obtenção de conhecimento sobre o uso do espectro de radiofrequência em uma determinada localização geográfica, com o objetivo de encontrar oportunidades de acesso para realizar a comunicação. Este trabalho propõe uma melhoria na técnica de Sensoriamento por Detecção de Energia. Esta técnica utiliza um limiar para diferenciar um canal de radiofrequência livre de um ocupado. Para realizar essa melhoria foi utilizado o SARSA, um algoritmo de Aprendizado por Reforço. Os resultados obtidos nos experimentos mostram um ganho de desempenho em relação ao uso do algoritmo Bayesiano.

## I. INTRODUÇÃO

O espectro de Radiofrequências (RF) é um recurso natural que tem seu uso licenciado por órgãos reguladores dos governos, como a *Federal Communications Commission* (FCC), nos Estados Unidos e a Agência Nacional de Telecomunicações (Anatel), no Brasil. No modelo atual de alocação do espectro de RF, os órgãos reguladores reservam e alocam canais do espectro para concessionárias, garantindo a elas total direito sobre um determinado canal. As concessionárias que possuem direito de uso sobre um canal são chamadas de Usuários Primários (UPs), pois possuem garantias de que nenhum outro usuário irá interferir na comunicação [1]. Como exemplos de UPs pode-se citar emissoras de rádio e televisão.

Um estudo realizado pela FCC expõe a ineficiência do modelo atual de alocação [2]. Segundo o estudo, os canais de RF reservados apresentam baixa taxa de utilização. Um exemplo típico é a transmissão de canais de televisão, que possuem uma ampla faixa do espectro de RF licenciada. Entretanto, na maioria dos municípios, poucas emissoras de TV atuam, levando a uma subutilização dos canais reservados. Desta forma, a FCC propôs um novo modelo de alocação, em que Usuários Secundários (USs), que não possuem direito de acesso sobre o canal, possam utilizar os canais que estão ociosos de forma oportunitista, isto é, temporariamente e sem causar interferência aos UPs. Esta técnica é comumente denominada *Dynamic Spectrum Access* (DSA).

Dante deste cenário, o Rádio Cognitivo (RC) surge como uma das principais tecnologias que possibilitam o acesso oportunitista ao espectro de forma eficiente [3]. RC é um

dispositivo capaz de analisar a utilização do espectro de RF e adaptar seus parâmetros de transmissão de forma dinâmica e autônoma, para maximizar a transferência de dados, mitigar a interferência ou facilitar a interoperabilidade. O principal requisito para a utilização de RCs é o sensoriamento espectral, responsável por analisar o espectro de RF em busca de canais não utilizados ou ociosos [4]. Dentre as técnicas de sensoriamento espectral dispostas na literatura, a Detecção de Energia (ED) é a que apresenta menor complexidade. Esta técnica utiliza de uma comparação de um limiar com a energia do canal de comunicação para classificá-lo quanto a sua ociosidade. Entretanto, o uso de um limiar estático prejudica sua precisão, ou seja, a taxa de acertos na classificação de canais de RF.

Este trabalho propõe uma melhoria à técnica de ED, visando aumentar a eficiência na detecção, mantendo baixo o tempo necessário na classificação. Para realizar esta melhoria foi utilizado o algoritmo SARSA (*State-Action-Reward-State-Action*), que tem por objetivo ajustar fatores que maximizem a eficiência de classificação, de acordo com o histórico de classificações realizadas anteriormente. Para realizar a validação da melhoria proposta será realizada uma comparação com o algoritmo Bayesiano.

O restante desse trabalho está organizado da seguinte maneira: Na Seção II são apresentados os conceitos necessários para a correta compreensão deste trabalho. A Seção III apresenta trabalhos que têm características em comum com este. A arquitetura e a modelagem do algoritmo é abordada na Seção IV. Já na Seção V é descrita a metodologia de avaliação do módulo proposto. A Seção VI apresenta os resultados obtidos a partir dos experimentos realizados. Por fim, a Seção VII apresenta as considerações finais.

## II. Sensoriamento Espectral

O principal requisito para a implementação de RCs é a capacidade de detectar os *white spaces* (espaços ociosos) do espectro de RF. Através do sensoriamento espectral o equipamento analisa, determina as características de utilização e obtém as regiões não utilizadas do espectro de frequências. O sensoriamento espectral deve encontrar canais livres em um curto intervalo de tempo, pois quanto mais rápido é o sensoriamento, maior é a utilização do espectro de RF ocioso. Além disso, o sensoriamento deve ter alta precisão para que aproveite todas oportunidades de espectro sem causar interferências aos UPs.

As técnicas de sensoriamento espectral podem ser divididas em dois grandes grupos. No sensoriamento local são utilizadas apenas informações observadas pelo dispositivo que efetua o sensoriamento. Já no sensoriamento cooperativo há a comuni-

cação entre dispositivos, desta forma é possível atingir maior precisão nos resultados.

A ED é a técnica mais comum dentre os algoritmos de sensoriamento espectral, devido a sua baixa complexidade computacional e de implementação [5]. Essa técnica é mais genérica do que outras técnicas de sensoriamento local, pois os dispositivos de RC não precisam ter nenhum conhecimento sobre o sinal do UP. O sinal é detectado comparando energia de um sinal recebido com um limiar que depende do nível de ruído [6]. Os maiores desafios do ED incluem a seleção do limiar correto para detectar UPs, incapacidade de diferenciar UP de ruído, e desempenho baixo quando a relação sinal/ruído (*Signal-to-Noise Ratio*, ou SNR) é baixa [7].

Outra técnica de sensoriamento local encontrada na literatura é a Detecção por Formato de Onda. Esta técnica consiste em procurar por padrões no sinal de comunicação recebido. Estes padrões podem ser identificados se parâmetros como preâmbulos e códigos de espalhamento são conhecidos. O desempenho desta técnica é superior à ED, porém o custo de tempo por sensoriamento é maior.

#### A. Rádios Cognitivos

O termo “rádio cognitivo” foi definido inicialmente por Mitola e Maguire como o ponto em que os dispositivos sem fio e as redes destes dispositivos são inteligentes sobre os recursos de rádio, de tal forma que detectam as necessidades de comunicação do usuário em função do contexto de uso. E provêm recursos e serviços que atendam essas necessidades apropriadamente [8]. Mais recentemente, a FCC definiu o termo “rádio cognitivo” em um sentido mais estrito, como sendo dispositivos de comunicação capazes de analisar a utilização do espectro de RF e adaptar seus parâmetros de transmissão dinâmica e autonomamente, para maximizar a transferência de dados, mitigar a interferência, facilitar a interoperabilidade ou acessar mercados secundários.

O Rádio Cognitivo surge como uma das principais tecnologias que possibilitam o acesso oportunista ao espectro de forma eficiente [3]. Uma aplicação de RCs é o DSA, que consiste em permitir o uso de forma temporária das regiões reservadas do espectro que encontrem-se ociosas, comumente denominadas de *white spaces*.

O RC possui duas características principais: capacidade cognitiva e reconfigurabilidade. A capacidade cognitiva é a habilidade do dispositivo de observar e capturar o estado de utilização do espectro de RF no ambiente no qual se encontra [9]. Em outras palavras, dispositivos de rádio com capacidade cognitiva são capazes de aprender quais são os melhores canais para realizar a comunicação no espectro de RF. A reconfigurabilidade, por sua vez, é a capacidade do dispositivo mudar dinamicamente os seus parâmetros de configuração de acordo com o ambiente de rádio.

#### B. Aprendizado por Reforço

O Aprendizado de Máquina pode ser definido como um conjunto de algoritmos e técnicas que possibilitam que um computador aprenda algo, isto é, que permitam ao computador aperfeiçoar seu desempenho em alguma tarefa. No contexto dos algoritmos de aprendizagem de máquina, existe a figura do agente. Um agente é uma entidade imersa no ambiente no qual é capaz de agir. O agente dispõe de uma capacidade de percepção e de representação parcial deste ambiente. Além disso, o agente possui um comportamento autônomo, consequência de suas observações sobre o ambiente e de seu conhecimento.

O Aprendizado por Reforço (AR) é uma das técnicas de Aprendizado de Máquina. O AR aprende pelo método de

tentativa e erro (*trial-and-error*), onde sua cognição é atingida através da interação, sem qualquer conhecimento prévio, com o ambiente. Outra característica do AR é a recompensa atrasada (*delayed reward*). Esta recompensa é uma informação que o agente recebe após executar uma ação. A recompensa pode ser positiva, se a ação executada pelo agente foi boa, ou negativa, se a ação executada foi prejudicial [10]. O problema consiste em escolher ações que maximizem o total de recompensas positivas recebidas pelo agente [11].

A ideia principal do algoritmo SARSA, que é utilizado nesse trabalho, é aprender com o retorno associado a cada ação em cada estado. Por exemplo, considerando um estado de execução e a necessidade de executar uma determinada ação, isso irá gerar uma recompensa, positiva ou não, que será utilizada para quantificar o quanto correta foi a ação tomada neste estado. Com o valor obtido, é gerado um histórico, ou então uma política de seleção de ações, que irá ajudar na próxima tomada de decisão deste estado a fim de aperfeiçoar sua tomada de decisões para melhorar seu desempenho.

### III. TRABALHOS RELACIONADOS

No trabalho realizado por Akyildiz [12] um algoritmo de sensoriamento cooperativo baseado em AR foi proposto. Os objetivos dos autores do trabalho se resumem em descobrir o conjunto ótimo de vizinhos para cooperação que gere o mínimo de tráfego de controle na rede, minimizar o *delay* causado pela cooperação e minimizar o consumo de energia decorrente da cooperação.

No trabalho de Oksanen [13], assim como no trabalho apresentado anteriormente, é proposto uma política de sensoriamento espectral cooperativo baseada em Aprendizado por Reforço. A política tem a função de decidir quais canais serão sensores e qual US fará o sensoriamento. Os objetivos do trabalho são maximizar a taxa de transmissão de USs, alcançar eficiência energética, minimizar perdas de detecções e se adaptar às condições do espectro de radiofrequências.

Pode-se perceber que os trabalhos prezam pela diminuição do *delay* e pela redução do consumo de energia ocasionado pela cooperação entre USs. Este trabalho propõe o uso de aprendizagem em sensoriamento local, objetivando alta eficiência e rapidez na detecção de UPs.

### IV. MÓDULO DE AJUSTE DINÂMICO DE LIMIAR

O módulo de ajuste dinâmico de limiar tem a função de ajustar em tempo real o limiar utilizado na classificação de canais de radiofrequência. O limiar deve se adequar rapidamente a mudanças nas propriedades da onda de sinal sendo analisada, para que alta precisão seja alcançada. A seguir descreve-se a arquitetura do módulo e como foi realizada a modelagem do algoritmo SARSA.

#### A. Descrição do Módulo

O módulo proposto executa em um *front-end* de rádio, realizando a análise sobre os sinais de RF capturados. Para o desenvolvimento do módulo foi utilizado o *Universal Software Radio Peripheral 2* (USRP2) que consiste em um *front-end* de rádio, flexível e de baixo custo, desenvolvido pela *Ettus Research*.

A programação ocorreu utilizando a linguagem Python e o software GNU Radio. Esse software apresenta um conjunto de módulos que permitem a realização do processamento de sinais de RF capturados pelo USRP2.

## B. Funcionamento

A arquitetura resultante da integração entre o Detector de Energia e o algoritmo SARSA é ilustrada na Figura 1. O bloco “Vetorização” é responsável por agrupar o fluxo de dados, recebido do USRP2, e armazena-los em um vetor. Esta operação é necessária para facilitar as demais manipulações de sinais. Já o bloco “Definição Janela” tem a função de dividir o vetor de dados repassado pelo bloco citado anteriormente em grupos contendo  $N$  elementos, onde  $N$  é o tamanho da janela de observação. Na experimentação, o valor da janela de observação foi fixado em 1024 amostras de sinal. O bloco “FFT” aplica a Transformada Rápida de Fourier (“Fast Fourier Transform”, ou FFT) em cada conjunto de dados enviado pelo bloco “Definição Janela”. A FFT tem o objetivo de transformar o sinal no domínio tempo para o domínio de frequência ou vice-versa. Tanto a entrada como a saída de dados deste bloco é um vetor de números complexos.

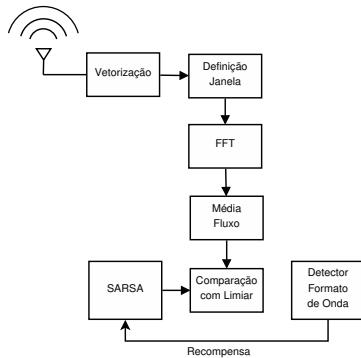


Figura 1. Arquitetura do Detector de Energia

O bloco “Média Fluxo” tem a finalidade de calcular a média de energia presente nos  $N$  valores que representam a janela de observação. Este bloco produz uma única saída para cada janela de observação avaliada. Este valor é comparado com o valor do limiar fornecido pelo algoritmo SARSA no bloco denominado “Comparação com Limiar”. As escolhas possíveis são: ocupado, se o valor de energia for maior que o limiar, ou ocioso, caso contrário. O reforço utilizado pelo algoritmo SARSA é fornecido por um Detector por Formato de Onda, localizado no bloco “Detector Formato de Onda”. Visto que o sensoriamento por Formato de Onda exige mais processamento, o reforço não é enviado ao algoritmo a cada sensoriamento.

Para adaptar o algoritmo SARSA ao Detector de Energia, o problema é modelado como um conjunto finito de estados  $s$  e um conjunto finito de ações  $a$ . O conjunto de estados  $s$  será representado pelos possíveis valores do limiar utilizado na classificação. Supondo que o número de estados seja definido como  $N$ , os estados possíveis seriam ( $S_1, S_2, S_3, \dots, S_n$ ), onde  $S_1$  representa o menor valor possível do limiar, e  $S_N$  representa o maior valor. As ações possíveis são: manter o valor do limiar( $A_1$ ), aumentar o limiar ( $A_2$ ) ou diminuir o limiar( $A_3$ ). Desta forma, se a ação de diminuir o valor do limiar for tomada, o próximo estado será o estado que represente um valor menor do limiar. A Figura 2 ilustra a modelagem utilizada.

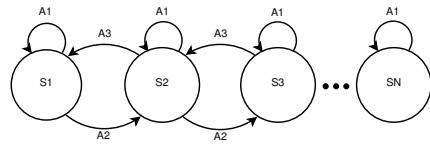


Figura 2. Modelagem Utilizada para Ajuste do Limiar

## V. AVALIAÇÃO DO MÓDULO

Esta seção apresenta detalhes da metodologia utilizada para avaliação do algoritmo de sensoriamento espectral proposto. A seguir, apresenta-se o ambiente de testes utilizado para execução dos experimentos e o cenário de avaliação.

### A. Ambiente de Experimentação

Para testar o desempenho do módulo desenvolvido será realizada a comparação com o algoritmo Bayesiano. Ambos algoritmos tem a função de controlar o limiar a ser utilizado no sensoriamento por ED.

O algoritmo Bayesiano, diferentemente do SARSA, utiliza um modelo probabilístico na escolha do limiar a ser utilizado. Cada limiar está associado a um “risco”, que é atualizado ao decorrer da execução do algoritmo. Os limiares que diminuem a precisão do algoritmo tendem a ser relacionados com riscos maiores. Desta forma, o limiar utilizado é o que apresenta o menor risco.

### B. Cenário de Avaliação

A validação deste trabalho foi realizada através da criação de um cenário de RF controlado. Neste cenário, o UP é representado por um gerador de sinais, que esporadicamente realiza alguma transmissão. Enquanto que o US é representado por um USRP2. Este cenário é ilustrado na Figura 3. O US deve detectar o mais rápido possível quando o gerador de sinais interrompe sua transmissão e sinalizar que o canal está disponível. Da mesma forma, quando o gerador voltar a transmitir, o canal deve ser dado como ocupado.



Figura 3. Cenário de Testes

Para que seja possível realizar uma comparação justa entre o algoritmo SARSA e o Bayesiano, o sinal foi gerado e após armazenado na memória de um computador. O sinal utilizado foi definido segundo uma distribuição de Poisson com média e variância ( $\lambda$ ) igual a 10. Desta forma, não é preciso capturar o sinal de RF a cada execução e pode-se avaliar o desempenho de ambos algoritmos sobre o mesmo sinal de transmissão.

## VI. RESULTADOS

Nesta seção são apresentados os resultados obtidos após uma série de repetições dos experimentos. Os algoritmos foram comparados quanto ao processamento e ao tempo necessário na classificação de canais de RF. Para avaliar o tempo de execução de cada algoritmo, os mesmos foram isolados do restante do projeto. Desta forma é possível conhecer com maior precisão o tempo gasto exclusivamente pelos algoritmos

no processo de sensoriamento. Foram realizadas 20 execuções para cada quantidade de ciclos avaliada.

O tempo necessário para realizar o sensoriamento pelo algoritmo SARSA foi de 68.5% do tempo gasto pelo algoritmo Bayesiano. A Figura 4 ilustra a comparação de tempo entre os algoritmos. Por exemplo, para completar 100000 ciclos, o algoritmo SARSA demora 0.85 segundos, enquanto o Bayesiano precisa de 1.25 segundos para concluir.

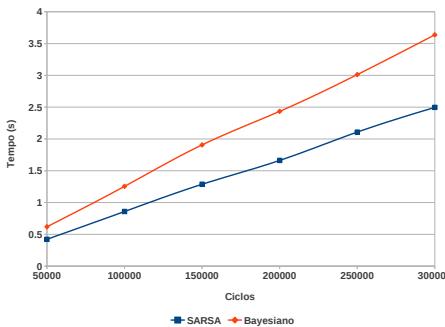


Figura 4. Comparação do Tempo de Execução

Já para medir a precisão dos algoritmos, foram realizados no total de 20 execuções sobre o sinal de radiofrequência armazenado, para cada algoritmo. O sinal de radiofrequência utilizado gera cerca de 80000 sensoriamentos, e o número de reforços recebidos pelos algoritmos é de, aproximadamente, 400 por execução.

Para calcular a precisão dos algoritmos é realizada uma comparação como o resultado ótimo. No que se deve à precisão de acertos, o SARSA supera levemente o Bayesiano. Enquanto que o algoritmo Bayesiano obteve precisão de 99.39%, o algoritmo SARSA obteve 99.46%. Porém o algoritmo Bayesiano é mais estável, ou seja, seus resultados são semelhantes a cada execução. A Figura 5 exibe uma comparação quanto à precisão dos algoritmos.

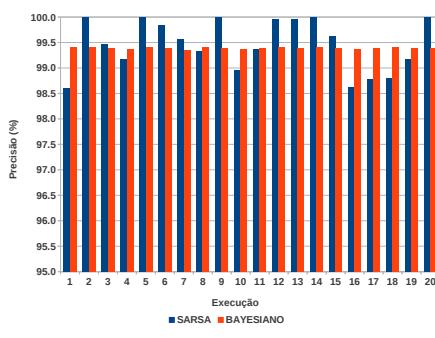


Figura 5. Comparação da Precisão

A Tabela I apresenta as médias de precisão de ambos os algoritmos, bem como seu desvio padrão. Observa-se que o SARSA apresenta maior variabilidade do que o Bayesiano, devido ao fato deste algoritmo utilizar o método de tentativa e erro na escolha das ações a serem executadas.

Algoritmo	Precisão	
	(%)	Desvio Padrão (%)
SARSA	99.46	0.0050
Bayesiano	99.39	0.0001

Tabela I  
COMPARAÇÃO ENTRE ALGORITMO SARSA E BAYESIANO

## VII. CONCLUSÃO E TRABALHOS FUTUROS

O sensoriamento espectral tem a função de analisar o espectro de RF em busca de canais não utilizados ou ociosos. É importante que este processo seja o mais rápido possível para que o espectro de radiofrequência ocioso seja bem aproveitado. É deve também ter alta precisão para que aproveite todas oportunidades de espectro sem causar interferências aos UPs. Este trabalho propõe uma técnica de sensoriamento espectral local e inteligente, objetivando alta eficiência, rapidez na detecção de UPs e menor consumo de energia.

Após a etapa de modelagem do algoritmo SARSA, foi desenvolvido o módulo de controle dinâmico do limiar de um ED. Realizaram-se experimentos para caracterizar o tempo e precisão de sensoriamento. Os resultados mostraram que o uso de AR, mais precisamente do algoritmo SARSA, apresenta ótimos resultados no controle do limiar, superando o algoritmo Bayesiano tanto em precisão quanto em velocidade.

Como trabalhos futuros, estuda-se a construção de um sistema de sensoriamento cooperativo, onde a interação entre dispositivos possa ajudar a corrigir problemas oriundos das transmissões sem fio, como por exemplo, alto ruído e desnecessimento do sinal.

## REFERÉNCIAS

- [1] J. Wang, M. Ghosh, and K. S. Challapali, "Emerging cognitive radio applications: A survey," *IEEE Communications Magazine*, vol. 49, pp. 74–81, 2011.
- [2] "Report of the Spectrum Efficiency Working Group," FCC, Tech. Rep., Nov. 2002.
- [3] A. Ghasemi and E. S. Sousa, "Spectrum sensing in cognitive radio networks: the cooperation-processing tradeoff," *Wireless Communications and Mobile Computing*, vol. 7, no. 9, pp. 1049–1060, 2007.
- [4] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *Communications Surveys & Tutorials, IEEE*, vol. 11, Mar. 2009.
- [5] KNOWS. *Cognitive Radio Networks Over White Spaces*, 2007.
- [6] H. Urkowitz, "Energy detection of unknown deterministic signals," *Proceedings of the IEEE*, vol. 55, no. 4, pp. 523–531, 1967.
- [7] H. Tang, "Some physical layer issues of wide-band cognitive radio systems," in *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005*, IEEE, 2005.
- [8] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *Personal Communications, IEEE*, vol. 6, Aug. 1999.
- [9] T. Jiang, D. Grace, and P. D. Mitchell, "Efficient exploration in reinforcement learning-based cognitive radio spectrum sharing," *Communications, IET*, Jul. 2011.
- [10] M. Bkassiny, Y. Li, and S. Jayaweera, "A survey on machine-learning techniques in cognitive radios," *Communications Surveys Tutorials, IEEE*, vol. PP, no. 99, pp. 1–24, 2012.
- [11] L. Busoni, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *Trans. Sys. Man Cyber Part C*, vol. 38, no. 2, pp. 156–172, Mar. 2008.
- [12] B. Lo and I. Akyildiz, "Reinforcement learning-based cooperative sensing in cognitive radio ad hoc networks," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2010 IEEE 21st International Symposium on*, 2010, pp. 2244–2249.
- [13] J. Oksanen, J. Lundén, and V. Koivunen, "Reinforcement learning based sensing policy optimization for energy efficient cognitive radio networks," *Neurocomput.*, vol. 80, pp. 102–110, Mar. 2012.

# Algoritmo de decisão espectral usando teoria dos jogos para rádios cognitivos

Roberto M. Schmidt<sup>1</sup>, Lauro Tremea Culau<sup>1</sup>, Matias Schimuneck<sup>1</sup>, Matheus Eidt<sup>1</sup>, Rafael Nicolay<sup>1</sup>, Cristiano Both<sup>1</sup>, Leonardo Roveda Faganello<sup>2</sup>, Maicon Kist<sup>2</sup>

Universidade de Santa Cruz do Sul<sup>1</sup>, Universidade Federal do Rio Grande do Sul<sup>2</sup>  
{rmainardi, laurotc, matiass, matheus, rafaelrodrigo}@mx2.unisc.br,  
cboth@unisc.br, {lrfaganello, maicon.kist}@inf.ufrgs.br

**Resumo**—A atual alocação do espectro de frequências, fundamental para realizar acesso à redes sem fio, é ineficiente, uma vez que os canais não são distribuídos conforme sua disponibilidade. Dessa maneira, muitos canais estão alocados, porém ociosos. Nesse contexto, Rádio Cognitivo surgiu como uma ferramenta para aprimorar o uso do espectro de frequências. O princípio por trás deste conceito é detectar canais não utilizados por usuários que detém sua licença de uso e permitir que usuários não licenciados possam utilizá-los. Para isso, é preciso que cumpram alguns requisitos, como por exemplo, interromper sua transmissão assim que um usuário licenciado inicie uma transmissão no mesmo canal. Desta forma é necessário dotar os dispositivos de inteligência, ou seja, fazer com que os dispositivos possuam capacidade cognitiva ou de aprendizado. Esse artigo descreve um algoritmo que utiliza técnicas de Aprendizado de Máquina, que mais especificamente usa Teoria dos Jogos para auxiliar a Seleção Dinâmica de Canal e apresenta os resultados obtidos através da implementação do algoritmo proposto.

## I. INTRODUÇÃO

A crescente demanda por acesso à redes sem fio pode ser observada pelo crescimento exponencial do uso de telefones celulares nas últimas décadas e de dispositivos que disponibilizam acesso à Internet sem fio, como por exemplo, *Access Points*. Devido ao crescimento do número desses dispositivos, ocorre um aumento da utilização do espectro de frequências, que é o recurso essencial para o provimento de serviços sem fio, entre eles, redes WiFi e de celular.

A responsável por regulamentar o acesso ao espectro de frequências no Brasil, é a Agência Nacional de Telecomunicações (ANATEL), vinculada ao Ministério das Comunicações. Atualmente, a regulamentação para acesso ao espectro consiste na alocação de faixas de frequências para entidades licenciadas. Estas entidades, possuem o uso prioritário do recurso e faixas de frequências livres denominadas ISM (*Industrial, Scientific and Medical*) para uso compartilhado entre dispositivos não licenciados [1]. Entretanto, essa alocação é feita de forma estática, por longos períodos de tempo e em grandes regiões geográficas [2]. Essa política dificulta a alocação de novas faixas de frequência, devido à escassez de canais livres no espectro. Além disso, como nem todos os usuários licenciados estão transmitindo durante todo o tempo, ocorre a geração dos denominados "espaços em branco" (*white spaces*) [3], que são regiões do espectro de frequências que encontram-se ociosas.

A baixa utilização efetiva dos canais estimulou a *Federal Communication Commission* (FCC), organização governamental reguladora dos sistemas de Rádio Frequência (RF) nos Estados Unidos<sup>1</sup>, a criar uma nova metodologia de acesso

ao espectro de frequências, onde usuários não-licenciados possam utilizar temporariamente as regiões do espectro que encontram-se ociosas. O acesso em períodos nos quais uma determinada frequência não está em uso é conhecido como *Dynamics Spectrum Access* (DSA). Nesse contexto, Rádios Cognitivos (RC) surgiram como principal alternativa para viabilizar a implementação de técnicas de acesso dinâmico ao espectro de frequências, pois permitem que um maior número de usuários possam acessá-lo simultaneamente [4].

Neste artigo, propõe-se um algoritmo de alocação de usuários não-licenciados no espectro de frequências utilizando técnicas de aprendizado de máquina. O aprendizado de máquina é uma área da inteligência artificial que utiliza raciocínio indutivo e procura prever eventos futuros baseados em acontecimentos prévios. No contexto de decisão espectral, essa característica é particularmente útil, porque as condições do canal podem mudar rapidamente, sendo necessário um algoritmo inteligente, capaz de prever qual canal ficará disponível por mais tempo. Quando é iniciada uma transmissão de um usuário primário no canal escolhido, é necessário efetuar *handoff* de espectro, ou seja, alterar o canal de transmissão. A implementação do algoritmo de aprendizado de máquina foi realizada utilizando-se o software *GNU Radio*<sup>2</sup> e prototipado em um *hardware* da Ettus, modelo USRP N210<sup>3</sup>. Com a utilização desses equipamentos é possível realizar transmissões e fazer a troca dos canais durante a transmissão. Os resultados obtidos através do algoritmo proposto apresentam uma transmissão com maior qualidade, pelo fato de o transmissor escolher sempre o melhor canal disponível para efetuar a transmissão.

O restante deste trabalho está organizado como segue: Na Seção II são expostos os conceitos que englobam a proposta de Rádios Cognitivos bem como a técnica de Teoria dos Jogos. Na Seção III são apresentados trabalhos relacionados ao assunto de Aprendizado de Máquina. Na Seção IV é descrita a metodologia de implementação do Algoritmo de Teoria dos Jogos. Na Seção V são expostos o cenário e o ambiente utilizado nos testes. A Seção VI apresenta os resultados obtidos a partir dos experimentos realizados. Por fim, a Seção VII apresenta as considerações finais e trabalhos futuros que poderão ser implementados.

## II. FUNDAMENTAÇÃO TEÓRICA

As próximas subseções apresentam com mais detalhes a fundamentação teórica referente ao trabalho proposto.

<sup>1</sup> <http://www.fcc.gov/>; <sup>2</sup> <http://www.gnuradio.org>; <sup>3</sup> <http://www.ettus.com>

### A. Rádio Cognitivos

A tecnologia de RCs consiste na capacidade de dispositivos de comunicações sem-fio inteligentes modificarem seus parâmetros de transmissão, tais como: frequência de operação, tipo de modulação, potência de transmissão, protocolos de comunicação entre outros, baseados em iterações com o ambiente em que operam. Com isso, um dispositivo de RC é capaz de conhecer o meio em que transmite e tomar decisões baseado nesse conhecimento, visando oferecer uma utilização mais eficiente do espectro de frequências [5]. Duas características principais de RC são:

- Capacidade Cognitiva: Refere-se à capacidade de extrair informações para obter um aprendizado sobre o meio a partir de sua observação, visto que, é possível identificar porções não utilizadas do espectro (*white spaces*) em determinado tempo, como mostrado na Figura 1.

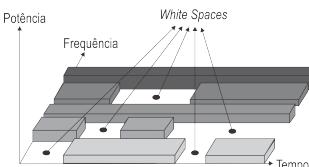


Figura 1. Uso do espectro por usuários licenciados e os *white spaces* criados

- Reconfigurabilidade: Através das informações obtidas por meio da capacidade cognitiva, os parâmetros de transmissão do rádio passam por um processo de adaptação às novas condições do meio. Adicionalmente, RC pode ser reprogramado para transmitir e receber dados em diferentes faixas de frequências.

Os requisitos que devem ser atendidos por um RC incluem: I) determinar quais frequências do espectro estão disponíveis, II) selecionar a melhor frequência disponível, III) compartilhar essa frequência com outros usuários e IV) liberar a frequência quando um usuário primário (PU) for detectado [6].

### B. Teoria dos Jogos

A Teoria dos Jogos é uma área da matemática aplicada, que estuda as situações estratégicas onde agentes escolhem diferentes ações na tentativa de melhorar seu retorno. Essa teoria visualiza qualquer ambiente multiagentes como um jogo, onde, para qualquer agente, será necessário considerar as ações de outros agentes, bem como o modo que essas ações afetam o próprio agente.

A técnica de Teoria dos Jogos estuda decisões que são tomadas em um ambiente onde vários jogadores interagem entre si ou onde não possui interação entre os jogadores, chamadas de Jogo Cooperativo e Jogo Não-Cooperativo, respectivamente. No primeiro, as escolhas de comportamentos ótimos são tomados a partir das escolhas dos outros jogadores. Já no segundo, cada jogador escolhe uma estratégia que é ótima para si, dado que outro jogador escolherá a estratégia ótima para ele.

### C. Software Defined Radio

O *Software Defined Radio* (SDR), é um dispositivo de comunicação sem fio que possui parte de suas funções implementadas por *software* em um computador, ao invés de

usar componentes fixos de *hardware*. A grande vantagem desse paradigma está na possibilidade de integrar e configurar diferentes parâmetros e mecanismos, que possam ser utilizados apenas alterando a programação de seu *software*.

Um SDR pode ser facilmente reconfigurado para desempenhar diferentes funções de acordo com a necessidade. Para suportar diferentes padrões de comunicação, e.g. WiFi, como também ajustar-se as variações do canal de comunicação, basta carregar diferentes tipos de *software* em memória, sem precisar substituir todo o equipamento de rádio [5].

### III. TRABALHOS RELACIONADOS

No contexto de Aprendizado de Máquina, o trabalho de Faganello et al. [7] propõe algumas melhorias no Q-Learning para utilização no contexto de decisão espectral. Uma das modificações propostas, foi analisar uma quantidade finita de épocas anteriores, diminuindo o peso de épocas passadas, uma vez que essa informação envelhece. A segunda modificação foi considerar a qualidade do canal como critério para contabilizar a recompensa do Q-Learning. Adicionalmente, os autores propuseram um modelo de comportamento de tráfego no espectro de frequências, com base no trabalho de Ghosh [8].

No mesmo contexto, porém utilizando a técnica de Teoria dos Jogos, Nie et al. [9] propõe um algoritmo que utiliza aprendizado não supervisionado para analisar o comportamento dos RCs para alocação do canal. Para isso, os autores definem duas funções com objetivos diferentes para o *spectrum sharing*, usuários egocêntricos e usuários cooperativos. Os autores assumem que os rádios podem medir a temperatura local, a interferência em diferentes frequências e podem ajustar-se, otimizando a taxa de transmissão de informação para uma dada qualidade do canal (utilizando codificação adaptativa do canal) e alterando para uma frequência diferente do canal.

Nenhum dos estudos acima leva em conta a abordagem proposta neste trabalho, que se baseia em um algoritmo de Decisão Espectral para decidir qual o melhor canal de frequência para efetuar a transmissão, entre os canais disponíveis no ambiente de Radio Frequência.

### IV. ALGORITMO DE DECISÃO ESPECTRAL

O algoritmo de Decisão Espectral se baseia no modo de Jogo Não-Cooperativo, onde existem dois jogadores, e esses jogadores não interagem entre si. Os *Universal Software Radio Peripherals* (USRPs) são os participantes do jogo. No decorrer desse jogo, cada um dos USRPs possuirá uma pontuação, baseada no valor total do tráfego enviado. A lista com os canais é dividida em quatro partes iguais: canais de muito alta, alta, média e baixa prioridade.

Cada USRP possuirá uma tabela equivalente à Tabela I, obtida a partir dos valores máximos e mínimos de transmissão que esse USRP observou desde o início de sua operação. Os índices dessa tabela são atualizados com o passar do tempo. Com base nessa tabela, cada USRP define qual parte da lista de canais deve utilizar. Dessa forma, dependendo da sua pontuação ele definirá sua prioridade. Baseando-se nessa prioridade, o USRP verifica a lista de canais, já dividida em quatro partes, e, assume que deseja utilizar o primeiro canal de sua prioridade da lista. O USRP, então, informa o canal que deseja utilizar para o módulo sensoriamento espectral.

Para a Tabela I, os valores são obtidos da seguinte forma: para cada USRP, MIN é o valor mínimo do tráfego observado, MAX é o valor máximo do tráfego observado e  $X = \frac{MAX}{4}$ .

Média de Tráfego ( $\mu$ )	Prioridade
$0 \leq \mu \leq MIN$	Baixa
$(MIN + 1) < \mu \leq 2X$	Média
$(2X + 1) < \mu \leq 3X$	Alta
$(3X + 1) < \mu \leq MAX$	Muito Alta

Tabela I  
ATRIBUIÇÃO DA PRIORIDADE

O módulo de sensoriamento espectral então realiza um rápidoo sensoriamento somente no canal escolhido, certificandose de que ainda está livre, ou seja, não foi selecionado por nenhum outro jogador. Caso este canal esteja ocupado, uma mensagem é retornada ao USRP informando-o que este canal está temporariamente indisponível. Dessa forma, o USRP deverá excluir esse canal de sua lista, dividi-la novamente em quatro partes iguais e refazer o processo de seleção do canal. Entretanto, caso o canal não esteja ocupado, o USRP ganha o direito de realizar sua transmissão por este canal.

#### V. METODOLOGIA DE AVALIAÇÃO

Esta seção apresenta detalhes da metodologia utilizada para avaliação da implementação do algoritmo de Decisão Espectral. O foco do trabalho é em um algoritmo para decidir qual o melhor canal de radiofrequência para realizar uma transmissão. 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 em um cenário real, foi utilizado o software GNU Radio. Ele é um conjunto de ferramentas de código aberto que provê um ambiente de desenvolvimento de componentes de processamento de sinais e blocos de processamento, que possibilita a implementação dos SDRs. O GNU Radio é desenvolvido sob a licença GNU General Public License (GPL), versão 3 e todos os direitos autorais do seu código-fonte pertencem à Free Software Foundation (FSF).

Outra ferramenta utilizada foi o *Universal Software Radio Peripheral* (USRP), que é uma plataforma de SDR flexível e de baixo custo desenvolvido pela Ettus Research. Seus circuitos são formados por dois componentes principais: a placa-mãe, responsável pelas funções programáveis mais complexas, e duas placas-filhas, que possuem os módulos de RF.

O USRP foi o equipamento utilizado na implementação da decisão do espectro, junto com as ferramentas do GNU Radio, visto que, possuem compatibilidade entre si.

##### B. Cenário

Nesta seção é apresentado um cenário (Figura 2) para comprovar o funcionamento da modelagem proposta neste trabalho. De modo a testar a implementação, foi utilizado um ambiente que contenha dois USRPs, cada um com uma rede sem-fio operando em 2.4 GHz, com seus respectivos dispositivos disputando acesso ao meio.

Dessa forma, para cada USRP, foi realizada uma varredura no espectro de frequências pelo módulo Sensoriamento, extraíndo as informações do ambiente. Em seguida, o módulo ChiMaS é responsável pelo armazenamento do histórico de ocupação e das condições de cada canal. Caso o canal seja considerado ocupado, ele é retirado da lista de canais candidatos. Em seguida, de acordo com as informações armazenadas anteriormente, estipula-se uma pontuação para os canais, ordenando-os da maior pontuação (menos ocupado, melhores

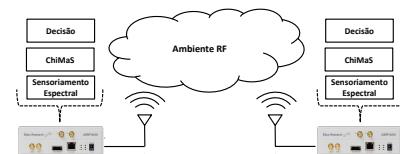


Figura 2. Cenário proposto

condições) para a menor (mais ocupado, piores condições). Por fim, o módulo Decisão é responsável por realizar a decisão de qual canal utilizar. A partir desse cenário de teste, é possível identificar o correto funcionamento do sistema proposto.

Este cenário foi modelado utilizando o software GNU Radio<sup>2</sup>. Para a prototipação desse sistema, foi utilizado um hardware de RF desenvolvido pela Ettus Research chamado de USRP-N210<sup>3</sup>. Estes equipamentos proporcionam a implementação de aplicações robustas de sistemas de SDR, ou seja, sistemas de rádio que podem ter suas características de operação alteradas por um software, como por exemplo, os dispositivos de RC.

Esse capítulo apresentou a solução proposta para este trabalho, bem como o cenário para os testes dessa solução. O próximo capítulo apresentará os resultados obtidos neste trabalho.

#### VI. RESULTADOS

Nesta seção são apresentados os resultados obtidos após uma série de execuções dos experimentos. Os resultados refletem o cenário definido na Seção V.

Para realizar os testes e gerar os resultados, o algoritmo foi executado utilizando 10 medições (ciclos) de tomada de decisão de cada USRP. Dessa maneira, é possível ter uma análise mais detalhada dos valores, pois há um conjunto significativo de amostras para se analisar.

A Figura 3 ilustra, no eixo vertical, a taxa de transmissão (Tx) em Kilobytes/segundo (Kbps) e, no horizontal, o número de medições realizadas pelo algoritmo. Isso ilustra o que cada USRP transmitiu ao longo da execução do algoritmo. Esses dados foram coletados pelo algoritmo de decisão espectral, e utilizados no cálculo da escolha da prioridade de cada aparelho.

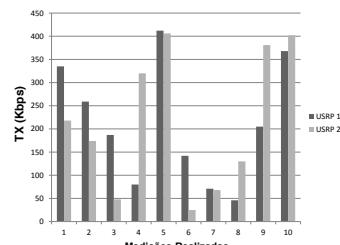


Figura 3. Taxa de Transmissão por medição, para cada USRP

A partir dos valores coletados e observados na Figura 3, o algoritmo definiu as prioridades para cada USRP em cada ciclo de execução (medição), como pode-se analisar na Tabela

II. A escolha da prioridade de cada aparelho é feito segundo a Tabela I. É possível observar que há casos em que ambos os aparelhos tem mesma prioridade, o que faz com que seja necessária uma disputa entre ambos, para determinar quem irá transmitir no melhor canal da prioridade selecionada. Essa escolha também se baseia na quantidade de dados que será transmitida, ou seja, os USRPs com mais dados a serem transmitidos, tendem a ter maior prioridade na escolha do melhor canal.

Medições	Prioridade USRP 1	Prioridade USRP 2
1	Baixa	Baixa
2	Muito Alta	Muito Alta
3	Muito Alta	Muito Alta
4	Alta	Alta
5	Alta	Alta
6	Média	Alta
7	Média	Alta
8	Média	Média
9	Média	Média
10	Alta	Alta

Tabela II  
PRIORIDADES DE CADA USRP POR MEDIÇÃO

O algoritmo de decisão espectral, então, utiliza os dados da Figura 3 e da Tabela II para realizar a escolha do canal de transmissão a ser utilizado, que, como mencionado anteriormente, leva em conta a quantidade de dados a serem transmitidos pelo equipamento em determinado ciclo de execução.

Na Figura 4 observa-se que em cada medição do sistema, os USRPs transmitiam em diferentes canais. Vê-se também, que, mesmo quando as prioridades de cada USRP, na Tabela II, são iguais, o algoritmo de decisão espectral consegue escolher um canal diferente para cada um dos equipamentos, devido às técnicas de Teoria dos Jogos utilizadas na implementação.

Como podemos ver na figura 4, os canais de frequência 1 e 2 possuem prioridade Muito Alta, o que torna-os canais com melhor qualidade para transmissão, já os canais 7 e 8 são os canais que possuem prioridade Baixa, pois são os canais com menor qualidade para a transmissão.

Também é possível analisar que, segundo o cálculo de prioridade, o primeiro ciclo de execução, sempre terá prioridade Baixa, que acarretará a escolha de canais de pior qualidade. Isso se dá pelo fato do cálculo utilizar uma média de transmissão. Durante a primeira execução, apenas um valor será observado, tornando-se, assim, tanto o valor mínimo quanto máximo. Todavia, no decorrer da execução, quando mais medições são executados, novos valores aparecem e é realizado um cálculo de média que tende a alterar a prioridade.

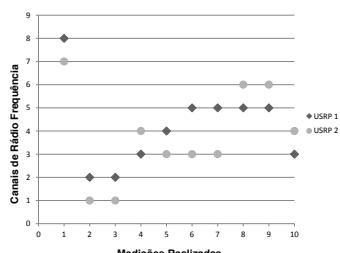


Figura 4. Canais usados por cada USRP para transmissão por medição

## VII. CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho propõe um algoritmo de decisão espectral utilizando uma técnica de Aprendizado de Máquina em cenários com mais de um dispositivo de rede (USRP), mais especificamente a de Teoria dos Jogos.

O algoritmo de decisão se baseia na quantidade de dados a ser transmitida por cada aparelho e a prioridade dos mesmos. O algoritmo, então, aplica a Teoria dos Jogos que escolhe o melhor canal disponível da prioridade selecionada para efetuar a transmissão dos dados.

Os resultados mostraram que o algoritmo efetua a escolha de canais diferentes para cada aparelho, levando em conta a taxa a ser transmitida e a prioridade dos aparelhos. Essa escolha faz com que a transmissão a ser realizada tenha mais eficiência e qualidade, visto que o algoritmo busca escolher o melhor canal para realizar a transmissão. Conclui-se que o algoritmo de decisão espectral, utilizando a técnica de Teoria dos Jogos, pode trazer benefícios para as transmissões de dados em cenários com mais de um dispositivo de rede.

Através da análise de trabalhos relacionados verificou-se que a tecnologia de RCs é um campo com ampla possibilidade de estudos. No que diz respeito à decisão espectral, observou-se que, até o momento, não foram encontrados trabalhos que abordam todos os tópicos que constam neste artigo. Desta forma, espera-se que a contribuição deste trabalho seja de grande importância para este amplo campo de estudo.

Como trabalhos futuros espera-se desenvolver um algoritmo de decisão espectral baseado no modo de Jogo Cooperativo, onde os USRPs compartilham as informações entre si através de um *token*. O *token* define qual dos dispositivos de rede possuirá o privilégio de transmitir no melhor canal. Dessa forma, cada USRP possuirá uma lista de canais disponíveis para transmissão, e o dispositivo que possui o *token* define em qual canal deseja transmitir. Dessa maneira, o dispositivo sempre escolherá o melhor canal da lista. Em seguida, o *token* é repassado para o próximo USRP e o processo se repete.

## REFERÊNCIAS

- [1] A. Ghasemi and E. Sousa, "Spectrum sensing in cognitive radio networks: requirements, challenges and design trade-offs," *IEEE Communications Magazine*, vol. 46, no. 4, pp. 32–39, 2008.
- [2] I. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "A survey on spectrum management in cognitive radio networks," *IEEE Communications Magazine*, vol. 46, no. 4, pp. 40–48, 2013.
- [3] D. Cabric, S. Mishra, and R. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on*, vol. 1, pp. 772–776 Vol.1, 2004.
- [4] D. Piazza, P. C. Cosman, L. B. Milstein, and G. Tartara, "A resource allocation algorithm for real-time streaming in cognitive networks," *IEEE Wireless Communications and Networking Conference*, pp. 1398–1402, 2009.
- [5] P. S. Coutinho, "Detecção de energia para rádios cognitivos usando gnu radio e usrp2," 2011.
- [6] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey," *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, 2006.
- [7] L. R. Faganello, C. B. Both, R. Kunst, and J. Granville, Z. L. ; Rochol, "Improving reinforcement learning algorithms for dynamic spectrum allocation in cognitive sensor networks," *IEEE Wireless Communications and Networking Conference*, vol. 1, pp. 41–46, 2013.
- [8] C. Ghosh, S. Pagadarai, D. Agrawal, and A. M. Wyglinski, "A framework for statistical wireless spectrum occupancy modeling," *IEEE Wireless Communications, Transactions on*, vol. 9, no. 1, pp. 38–44, 2010.
- [9] N. Nie and C. Comanicu, "Adaptive channel allocation spectrum etiquette for cognitive radio networks," *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, pp. 269–278, 2005.

---

III

## Sessão 3 - Trabalhos de IC

---



# Gerenciamento da Coleta de Lixo Urbano utilizando uma Rede Wireless com Tecnologia ZigBee

Heitor Scalco Neto, Douglas Giacomini, Humberto Machry Prado, Claiton Colvero  
Colégio Técnico Industrial de Santa Maria – CTISM – Curso Superior de Redes de Computadores  
Universidade Federal de Santa Maria - UFSM  
Av. Roraima, 100 - Santa Maria - RS, Brasil  
heitorscalco@hotmail.com, giacomini@redes.ufsm.br, humberto87nicosmrs@gmail.com, claiton@redes.ufsm.br

**Resumo -** Este artigo tem como objetivo proporcionar uma maior eficiência no sistema de coleta de lixo da cidade de Santa Maria - RS, colaborando assim para o total controle dos procedimentos de coleta por parte da administração pública e, adicionalmente, a maior facilidade no dia-a-dia dos moradores. O foco deste artigo é disponibilizar uma maior comodidade a população e organizar os contêineres de lixo, de forma adequada, pois muitas vezes encontram-se superlotados e com mau cheiro. O modelo atual de coleta utilizado na cidade atenta contra o bem estar das pessoas que circulam nas ruas e que são obrigadas a conviver, em algum momento, com esses contêineres. Através da utilização de uma tecnologia com padrão 802.15.4 (ZigBee) de redes sem-fio, é possível implementar este sistema de gestão de resíduos, monitorando a coleta, transporte e descarte destes materiais. Essa tecnologia é capaz de receber informações e retransmíti-las para todos os dispositivos ativos ao seu alcance. A utilização de uma tecnologia de baixo custo e de alta confiabilidade possibilita pôr em prática este sistema.

**Keywords —** *Rede mesh; ZigBee; coleta de lixo; automação.*

## I. INTRODUÇÃO

Atualmente ao realizar simples atividades corriqueiras, como deslocar-se de um ponto ao outro de uma grande cidade, utiliza-se diversas tecnologias em soluções de automação e controle inseridas e mascaradas nos serviços e produtos oferecidos. Muitas vezes as pessoas não possuem o conhecimento da complexidade destes sistemas. Através do nível avançado de automação de processos e serviços em todas as áreas de conhecimento, as falhas que esses sistemas autônomos apresentam passam despercebidos, porém em geral nos proporcionam mais conforto, mais facilidades adicionais e principalmente um grau elevado de segurança. Para o desenvolvimento deste trabalho de iniciação científica foi realizada uma criteriosa pesquisa no intuito de identificar algum processo ou serviço corriqueiro oferecido dentro do estado do Rio Grande do Sul que necessitasse urgentemente de um novo sistema de automação baseado em aplicações de redes industriais sem fio. Como critérios de seleção foram observados principalmente viabilidade técnica, a viabilidade econômica, a apresentação de falhas recorrentes e o impacto de cada falha na sociedade.

Como resultado desta pesquisa, o sistema de gerenciamento da coleta do lixo urbano em Santa Maria – RS foi selecionado para ser tratado devido ao alto fator de impacto na sociedade,

pois a correta disposição e tratamento dos contêineres de lixo utilizados nos proporciona a organização devida dos dejetos sólidos produzidos diariamente, evitando assim a exposição direta ao meio ambiente. O presente trabalho tem como objetivo aumentar consideravelmente a eficiência deste sistema atual de gestão de resíduos e consequentemente a comodidade da população local, tendo como foco principal a coordenação da coleta e disponibilização destas informações aos usuários, através de uma consulta online e adicionalmente o aviso de proximidade, via SMS, informando inicialmente aos clientes cadastrados qual o tempo médio restante para a coleta dos resíduos em determinado dia e local de interesse. Isso proporciona aos usuários do sistema, a capacidade de poder programar-se para levar o lixo até o contêiner, evitando assim, o acúmulo desnecessário de lixo por longos períodos, que geralmente causa o mau cheiro característico e evita a ação de catadores e animais que reviram e espalham o mesmo.

Para adicionar estas facilidades propostas neste artigo, um robusto sistema de rede de comunicações precisa ser implementado, com especial atenção aos recursos que deverão ser disponibilizados. Desta forma, foi decidido por utilizar na implementação deste sistema de automação, um padrão de rede industrial de alta confiabilidade e baixo custo, que possua a capacidade de operar em tempo real e que também opere através da tecnologia de comunicação sem fio (*wireless IEEE 802.15.4*) [1][8], tornando mais eficiente a coleta automatizada de lixo na cidade, onde através de um acesso remoto, os moradores serão informados sobre o horário correto do recolhimento do lixo nos contêineres, tendo como principal objetivo evitar que o lixo não fique exposto e acumulado, causando mal cheiro nas ruas e nas residências da cidade.

Complementando a ferramenta de controle, também é necessário implementar um sistema de cadastramento de moradores, onde os mesmos poderão criar logins de acesso baseados no Código de Endereçamento Postal (CEP) e adicionalmente cadastrar os números de seus telefones celulares em uma central da companhia de coleta do lixo. Como cada contêiner e caminhão de coleta será instrumentado com um dispositivo de rede, o sistema apresentará a funcionalidade de perceber a aproximação dos mesmos e de pontos pré-definidos no caminho, despertando o sensor do contêiner para o aviso de coleta próxima. Com o objetivo de preservar a bateria dos nós de rede dos contêineres, os

dispositivo ZigBee instalados estarão configurados em modo “sleep”, e serão acordados pela presença do caminhão de coleta, assim retornando um *frame* de dados para a central de controle, que em seguida realizará a disponibilização no website da companhia e adicionalmente enviará uma mensagem SMS aos celulares dos moradores cadastrados para informar a proximidade da hora de coleta.

Este trabalho foi inicialmente dividido em 3 etapas para a melhor utilização dos recursos disponíveis, onde destacam-se a pesquisa bibliográfica, utilizada para selecionar e definir o problema de maior demanda, a escolha de tecnologia de comunicação, que visa otimizar o sistema de rede industrial wireless a ser utilizado, mantendo a alta confiabilidade baixo custo, e finalmente o ensaio em escala realizado em laboratório, que teve como objetivo testar os conceitos e tecnologias propostas para essa solução de gerenciamento da coleta de lixo na cidade de Santa Maria.

## II. DESENVOLVIMENTO

A cidade de Santa Maria – RS optou pela coleta de lixo em áreas urbanas, para usuários residenciais, através de contêineres desde os meados de 2008, enfrentando até os dias de hoje diversos problemas de rejeição e reclamações dos moradores por problemas de utilização, controle e educação. Na Fig. 1 podemos observar a automação mecânica da coleta de lixo através deste sistema de contêiner, onde não é possível controlar os horários e procedimentos de coleta sem a intervenção local de um fiscal ou da própria população.



Fig. 1. Sistema de coleta de lixo com contêineres em Santa Maria [2].

A correta instrumentação deste processo através da instalação de dispositivos sensores e atuadores de comunicação wireless com propriedades de formar redes adaptativas (*mesh*), proporciona a conexão de forma totalmente autônoma entre os contêineres de lixo, os caminhões de coleta dos mesmos e locais estratégicos fixos de passagem e descarte dos resíduos coletados, que permitem que seja realizado um controle remoto de toda a operação, inclusive disponibilizando aos usuários diversas modalidades de serviços agregados, como consulta de dias e horários aproximados de coleta através do respectivo CEP, sendo ainda possível o cadastro e envio automático de

mensagens SMS de proximidade de horário de coleta de lixo para usuários previamente cadastrados.

### A. Tecnologia de Comunicação Wireless:

Com o crescente desenvolvimento das tecnologias de comunicação sem fio em todo o mundo, na maioria das vezes inclusive substituindo tarefas antes realizadas por pessoas de forma satisfatória, foi desenvolvido este trabalho utilizando como base uma tecnologia que foi estudada em sala de aula, como forma de oferecer um sistema de gerenciamento de resíduos inovador e automatizado para o controle da coleta do lixo urbano, através de um processo que oferece um crescimento social considerável baseado nas facilidades de utilização de um modelo de baixo custo de rede industrial.

Como a rede de comunicação deste projeto possui alguns requisitos específicos para ter viabilidade, foi selecionada a tecnologia ZigBee como a melhor opção de comunicação sem fio entre os diferentes pontos de instrumentação. A tecnologia ZigBee surgiu no começo desta década na tentativa de suprir a necessidade de organização e controle de redes sem fio (padrão IEEE 802.15.4), com a missão de oferecer uma solução completa, de plataforma aberta para a padronização dos sistemas de comunicação, com padrão global confiável, baixo custo, alta eficiência, longo alcance com suporte a topologia de redes adaptativas *mesh*, estrela e árvore, criptografia, baixa potência de transmissão e radiação de espúrios e sem a necessidade de interligação por meio físico [3].

Para os primeiros ensaios em laboratório em escala reduzida foram utilizadas placas comerciais de interfaceamento chamadas de HomeBee, que tem como função original automatizar ambientes residenciais de forma simplificada, onde se pode conectar diretamente diferentes dispositivos, com ou sem fio. A placa dispõe duas saídas a relés pré-definidas, que podem ser usadas para ligar ou desligar dispositivos com tensão de até 220 V e corrente de 10 A [4].

### B. Desenvolvimento do projeto:

Para os desenvolvimentos deste projeto foi definido que os seguintes recursos utilizados na coleta do lixo deverão ser instrumentados com pelo menos um dispositivo ZigBee:

- Contêiner de coleta de lixo urbano: Cada contêiner deverá ser instrumentado com um dispositivo ZigBee configurado como dispositivo final em modo sleep [5];
- Caminhão de coleta de lixo: Cada caminhão de coleta de lixo deverá possuir um gateway [6] com interfaces ZigBee e 3G, para conexão com a rede *mesh* e com a central de controle e gerenciamento de rotas de coleta;
- Caminhão de limpeza de contêineres: Cada caminhão destes deverá ter um gateway semelhante ao de coleta instalado, para que seja realizado o sincronismo entre este e os contêineres recém esvaziados para a limpeza.
- Pontos estratégicos da rota: Diversos pontos da rota de coleta podem ser instrumentados para monitorar a correta passagem do caminhão, registrando a identificação do caminhão, o horário e a data de

passagem por aquele ponto, sincronizando a coleta. Estes dados são enviados pelo *gateway* do caminhão.

- Garagem e ponto de descarte do lixo: Nestes dois pontos também será necessária a instalação de sensores ZigBee para perceber a aproximação dos caminhões e tempo dispensado em cada um deles, assim como adicionalmente também indicar a disponibilidade destes recursos para auxiliar em outras tarefas de coleta que possam estar com problemas de atraso ou de falha dos equipamentos utilizados.

Para a integração destes recursos e disponibilização dos eventos para os usuários em tempo real, também é necessário o desenvolvimento de uma central de controle e monitoramento de rotas de coleta de lixo urbano, que ficará responsável por receber os dados enviados por cada um dos caminhões em deslocamento, processando estes *frames* para disponibilizar em uma página *web* a localização aproximada e tempo de chegada no endereço solicitado, com função adicional de enviar uma mensagem *SMS* a usuários cadastrados com informações de tempo médio de chegada na sua residência.

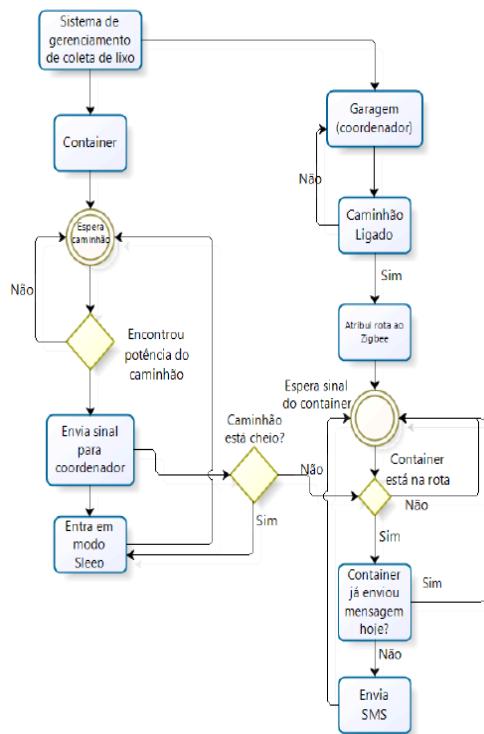


Fig. 2. Fluxograma de processos envolvidos na coleta de lixo.

### III. RESULTADOS OBTIDOS

Em um primeiro momento foram realizados os ensaios preliminares em escala reduzida no laboratório para diminuição de custos e viabilidade do modelo, onde foram utilizado três computadores portáteis conectados com dispositivos *ZigBee*, e uma placa *HomeBee* para conexão. O primeiro computador simulava um contêiner, o segundo simulava o caminhão em deslocamento, o terceiro era responsável por simular um dispositivo instalado em pontos fixos, como na garagem e no centro de descarte, e o último, configurado como coordenador era utilizado com a interface com o software de controle e gerenciamento das informações coletadas.

Todos os dispositivos ZigBee foram configurados de modo que transmitam as informações de movimentação dos caminhões para todos os outros dispositivos na rede (*Broadcast*) [7], assim os dados que forem úteis serão repassados para todos os nós subsequentes da rede, já as informações que não atendem a rota serão descartadas. Desta forma, caso haja uma interferência ou falha durante o processo de comunicações entre os dispositivos de instrumentação, as informações serão replicadas através das funcionalidade da rede *mesh* até chegar no dispositivo desejado, proporcionado o envio destas informações pelo gateway até a central de controle e gerenciamento, que pode estar instalada em qualquer ponto remoto, como na garagem ou na sede da empresa responsável.

O dispositivo configurado como coordenador estará situado na central de controle e monitoramento e terá por finalidade atribuir a rota para o dispositivo presente no caminhão de lixo através da conexão 3G ou ZigBee, dependendo da distância entre eles. Este dispositivo também irá receber a identificação de proximidade do caminhão de coleta ou lavagem dos contêineres através do gateway e ajustar a rota, disponibilizar as informações na página web e requisitar o envio dos SMS aos clientes cadastrados.

Cada contêiner possui um dispositivo ZigBee instalado e que permanece em estado de *sleep* [5] durante a maior parte do tempo, só acordando através da detecção do sinal um outro dispositivo que se aproxima e tenta uma conexão de rede, representado pelo dispositivo ZigBee instalado no caminhão de coleta ou lavagem, que está sempre em operação e interrogando o restante da rede em busca de novas conexões. Quando o sensor do contêiner receber um sinal do dispositivo do caminhão maior ou igual a -80 dbm, o ZigBee do contêiner é acordado (*wake-up*) e imediatamente envia um frame de dados com sua identificação ao coordenador, utilizando o *gateway* do caminhão para esta tarefa. O coordenador (localizado no centro de controle), através de uma pesquisa no banco de dados, cadastrá as informação no servidor *web* para consulta dos usuários e adicionalmente envia um *SMS* para cada morador cadastrado próximo aos contêineres subsequentes da rota em operação. O ZigBee será instalado na coluna superior interna do contêiner, protegido por uma caixa de polímero dielétrico de alta densidade.

Para que os sinais recebidos pela central de controle e gerenciamento das rotas sejam corretamente interpretados e utilizados, é necessária a implementação de um banco de dados com acesso aos frames que o coordenador recebe, com a

finalidade de armazenar os IDs dos contêineres e as rotas (caminhos) que cada caminhão vai seguir para a coleta de lixo, tornando a disponibilização da localização em tempo real e o envio das mensagens aos moradores das ruas mais eficaz, auditando as operações, com a capacidade de otimizar as coletas através de reorganização dinâmica em caso de atrasos ou falhas de equipamentos em quaisquer rotas. As rotas para o recolhimento de lixo serão pré-definidas conforme a sequência que os contêineres estiverem dispostos nas ruas.

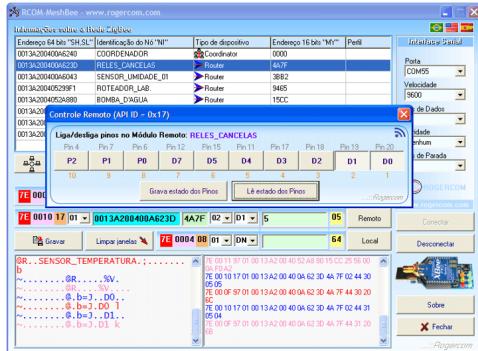


Fig. 3. Software utilizado para monitoramento dos frames enviados.

Através dos ensaios realizados em escala reduzida no laboratório, utilizando 3 dispositivos ZigBee devidamente conectados e configurados em 3 computadores portáteis diferentes e um coordenador conectado em uma placa HomeBee, constatou-se o perfeito funcionamento do sistema. Um dos dispositivos ZigBee atuou como coordenador (localizado na central de controle) e disponibilizou os dados recebidos conforme pode ser visualizado na Fig. 3, e os outros representavam os caminhões se deslocando e os contêineres de lixo em modo *sleep*.

Com as potências de transmissão reduzidas ao máximo para o teste em escala, foi simulada a saída do caminhão de lixo da garagem, após isso foi realizada a aproximação do mesmo ao contêiner instrumentado, que detectou a presença do dispositivo do caminhão e fez com que o ZigBee despertasse do modo *sleep*, enviando um *frame* de dados com sua ID para a placa HomeBee do coordenador através das facilidades da rede *mesh* montada pelo ZigBee do caminhão. Com os dados recebidos na central de controle, ainda não implementada de forma totalmente funcional, foi possível identificar a localização do caminhão de coleta e prever o tempo médio de chegada do mesmo nos próximos contêineres subsequentes. Provou-se desta forma que o conceito do projeto é válido e que os dispositivos interagiram de forma autônoma, mandando mensagens entre si com total confiabilidade.

Constatou-se também que o custo do sistema é relativamente baixo e possui alta confiabilidade [5], já que a velocidade de transmissão é bastante reduzida por ser um dispositivo de rede industrial [1], fornecendo assim, a

possibilidade da implementação do mesmo em meio urbano, desde uma pequena cidade até uma grande metrópole.

#### IV. CONCLUSÃO

Foi desenvolvido um sistema automatizado de coleta de lixo através de contêineres, especialmente baseado nas informações para a cidade de Santa Maria - RS onde, de modo totalmente autônomo, os moradores serão informados sobre o horário correto do recolhimento do lixo nos contêineres, tendo como principal objetivo evitar que o lixo não fique exposto e acumulado, causando mal cheiro nas ruas e nas residências da cidade, além de serem revirados por catadores e animais.

Também é proposto implementar um sistema de cadastramento de moradores, onde os mesmos poderão ter acesso às informações das rotas de coleta de lixo relevantes através do seu CEP, podendo cadastrar seus números de telefones celulares na central para aviso via *SMS*. O sistema funciona com a aproximação do caminhão de lixo com os contêineres, que retira os dispositivos de instrumentação dos mesmos do modo "*sleep*", originalmente configurados desta forma para economia de energia, retornando um *frame* de dados com identificação para a central de controle e gerenciamento, que em seguida cadastrava estas informações em um banco de dados para consulta remota e também envia uma mensagem *SMS* aos celulares dos moradores cadastrados para receber-las.

O sistema apresentado demonstrou possuir viabilidade técnica, econômica e comercial, sendo possível ser implementado em centros urbanos pequenos, médios ou grandes com a mesma facilidade e simplicidade de operação, oferecendo uma importante ferramenta de desenvolvimento social e educação da população, aumentando o grau de satisfação dos mesmos através da disponibilização de serviços associados. A redução de custos a longo prazo pelas administradoras de coleta de lixo também é apresentado como um grande diferencial deste projeto.

#### REFERÊNCIAS

- [1] The ZigBee Advantage. Disponível em: <<http://www.digi.com/technology/rf-articles/wireless-zigbee>>. Acesso em setembro de 2013.
- [2] Sistema de coleta de lixo urbano da cidade de Santa Maria – Empresa Themac. Disponível em: <<http://www.themac.cc/portugues/index.php>>. Acesso em agosto de 2013.
- [3] Configuração de Dispositivos ZigBee – Rede Mesh. Disponível em: <<http://www.rogercom.com/ZigBee/ZigBeePag04.htm>>. Acesso em agosto de 2013.
- [4] Rogercom - Manual Placa HOMEBEE. Disponível em: <<http://www.rogercom.com/ManualHomeBee.pdf>>. Acesso em setembro de 2013.
- [5] Rogercom - Informações e configurações dispositivo ZigBee. Disponível em: <<http://www.rogercom.com/ZigBee/ZigBee.htm>>. Acesso em setembro de 2013.
- [6] Produtos Digi. Disponível em: <<http://www.digi.com/products/zigbee/>>. Acesso em setembro de 2013.
- [7] XBEE®/XBEE PRO® ZB RF MODULES (2012), ZigBee RF Modules by Digi International. Disponível em <<http://www.digi.com>>. Acesso em 2013.
- [8] ZigBee Technology: Wireless Control that Simply Works - Patrick Kinney 2003

# Aplicação de Algoritmos de Escalonamento de Processos para Gerenciamento de Interseções em VANETs

Thiago Lopes Trugillo da Silveira  
Curso de Ciência da Computação - UFSM  
thiago@inf.ufsm.br

Marcia Pasin  
Centro de Tecnologia - UFSM  
marcia@inf.ufsm.br

**Resumo**—Controle de interseções representa um problema interessante no gerenciamento de trânsito. Quando a tecnologia de VANETs for implantada e disponibilizada para usuários em geral, permitindo comunicação entre veículos e com a infraestrutura, novas políticas poderão ser aplicadas para o controle de passagem de veículos em interseções, provendo agilidade a redes de transporte. Neste sentido, este trabalho explora a aplicação de algoritmos de escalonamento de processos em CPUs para gerenciar interseções em vias urbanas, usando como suporte comunicação veículo-infraestrutura, e levando em conta um cenário com demanda de larga escala. Foi realizada uma implementação contemplando diferentes políticas de escalonamento de processos para gerenciar trânsito considerando diferentes interseções.

## I. INTRODUÇÃO

Soluções para gerenciamento de trânsito eficiente e provimento de mobilidade urbana são assuntos que vêm ganhando destaque, tanto no contexto internacional (por exemplo, DARPA *Grand Challenge*), quanto em nosso país (Edital RNP-CTIC Cidades Inteligentes). Com o número crescente de veículos circulando nas vias, e com o consequente aumento da demanda, a aplicação de estratégias para melhorar o gerenciamento do trânsito tornam-se imprescindíveis. Possíveis soluções são o provimento de transporte público de qualidade, a limitação da quantidade de veículos em circulação, e o uso de semáforos adaptativos para controle de interseções viárias.

Os semáforos adaptativos, já implantados em algumas cidades, são calibrados de forma dinâmica, de acordo com o fluxo de veículos, objetivando que o trânsito flua de forma mais eficiente. Essa situação evitaria, por exemplo, a exposição do sinal verde de forma prolongada em uma via com poucos veículos, se no fluxo de contra-partida há uma outra via onde o trânsito é intenso. Os semáforos adaptativos inteligentes seriam um passo mais adiante, estariam baseados nos veículos (e não no fluxo), e usariam algum mecanismo para promover não apenas a eficiência no escoamento do fluxo de veículos, mas também o equilíbrio para o atendimento aos usuários.

Em um cenário futurista, com a implantação de VANETs (*vehicular ad hoc networks* ou redes veiculares) e o conceito de veículos autônomos, semáforos seriam eliminados. O controle de passagem nas interseções e cálculo de rotas (*road-planning*), por exemplo, será realizado pelos próprios veículos (arquitetura descentralizada), desde que os veículos disponham de dispositivos com serviço de GPS e tecnologias de comunicação veicular instalados e operacionais. Novas políticas para o controle de interseções, além da tradicional janela de tempo, adotada pela sinalização,

poderão ser implementadas. Por exemplo, políticas para escalonamento de processos em CPUs, como *first in first out* (FIFO) e *shortest job first* (SJF), poderiam ser implementadas para controle de passagem nas interseções com suporte de comunicação veicular e comunicação veículo-infraestrutura.

A possibilidade de implementar políticas para o controle de interseções baseadas no comportamento e nas interações entre os veículos contrasta com o controle de semáforos tradicional, que usa um modelo matemático para descrever fluxo no trânsito. Uma vez que esta tecnologia ainda não está amplamente disponível, simulação computacional abre caminho para avaliar possibilidades de serem implantadas na prática.

Neste contexto, este artigo propõe a implementação de políticas para controle de passagem de veículos em interseções baseadas em soluções para escalonamento de processos em CPUs, com o suporte de comunicação veículo-infraestrutura. A avaliação das políticas é feita através da execução de experimentos em ambiente controlado.

Para conduzir a avaliação experimental foi usado o simulador Siafu [6], para ambientes ubíquos e implementado com o suporte de MAS (*Multi-Agent System* ou sistema multi-agente). Bazzan [1] enfatiza que o modelo de agentes de *software* é adequado para descrever muitas aplicações envolvendo trânsito de veículos. Cada veículo é representado como um *agente* autônomo que segue um comportamento de forma independente e interage com os demais agentes e/ou com a infraestrutura para a tomada de decisão (isto é, para decidir quem deve passar primeiro em uma interseção).

O texto está organizado como segue. Trabalhos relacionados são descritos na seção II. A contextualização do problema para o modelo de agentes de *software* e políticas de controle de interseções são descritas na seção III. Detalhes de implementação são descritos na seção IV. Avaliação experimental é discutida na seção V. Conclusões e trabalhos futuros são apresentados na seção VI.

## II. TRABALHOS RELACIONADOS

Controle de interseções representa um problema interessante no gerenciamento de trânsito. Dresner & Stone [3] descrevem um esquema de reserva onde o veículo deve alocar, em uma central, espaço e tempo para atravessar a interseção de duas vias. Esta técnica é mais eficiente que o controle semafórico tradicional por janelas de tempo. Entretanto, se o veículo não conseguir reservar um *slot* necessário para a passagem da interseção, pode sofrer

espera indefinida. Outro problema é a existência de uma central para controlar a política de passagem no semáforo. Se a central falhar, o serviço torna-se indisponível. Uma extensão do trabalho de Dresner & Stone para o contexto de múltiplas interseções foi realizado por Vasirani & Ossowski [8]. A ideia é oferecer um serviço adequado ao coletivo, mas sem colaboração entre os agentes. Em Ferreira *et al.* [4], através do suporte de comunicação veicular, um veículo próximo de uma interseção é escolhido para controlar a interseção. Quando o veículo se afasta da interseção, um novo veículo é eleito entre os demais para controlar a interseção. O artigo não descreve especificamente que política é usada para controlar a interseção. Krajzewicz *et al.* [5] compara tamanhos de filas para passagem em interseções e a fila maior tem maior prioridade, ao contrário do que ocorre na política SJF (avaliada neste trabalho). Em Mugnella & Netto [7], o fluxo de trânsito é medido, avaliado e melhorias para a calibração semafórica são propostas pelo uso de algoritmos genéticos. O enfoque do trabalho de Mugnella & Netto é na calibração da janela de tempo adotada na sinaleira. Comunicação veicular não é levada em conta. Em contraste, neste presente trabalho são avaliados algoritmos simples e conhecidos, de fácil implementação e de rápida execução, aproveitando a natureza distribuída inherentemente a VANETs e o suporte de comunicação com a infraestrutura.

### III. MODELAGEM DO PROBLEMA

#### A. Modelo de agentes de software aplicado a gerenciamento de interseções de trânsito

Uma abstração conveniente para modelar veículos em circulação é o conceito de agentes de *software*, que são unidades com comportamento independente. Um grupo de agentes (veículos) obedece a uma determinada política para realizar a travessia de uma interseção. Cada veículo  $c_i$  (para  $i \in \mathbb{N}$ ) é unicamente identificado (na prática, pode ser o RENAVAN ou a sequência alfanumérica que consta na placa do veículo) e pertence a apenas uma fila  $q_m$  ou  $q_n$ . As filas compartilham uma seção crítica (interseção), onde os veículos devem passar para atingir seu objetivo final. Apenas um veículo pode passar a seção crítica por vez. Quem decide qual é este veículo é a **política de controle de interseção**.

#### B. Políticas de controle para interseções de trânsito

Para o controle de interseção, veículos podem obedecer as seguintes políticas:

- **FIFO (first in first out):** utiliza o conceito de uma fila virtual global, formada por elementos das filas  $q_m$  e  $q_n$ ; oferece prioridade ao primeiro veículo que chegar à interseção (topo da fila virtual); é um algoritmo que busca justiça no tempo de atendimento à passagem de interseção (quem chegou primeiro espera menos); no entanto se uma fila for mais rápida (e densa) que a outra, veículos na fila mais lenta podem sofrer longa espera até serem atendidos; consequentemente o tempo médio de espera para os veículos passarem a interseção aumenta.

- **SJF (shortest job first):** calcula o tamanho das filas; a fila menor passa primeiro; esta política pode impor espera indefinida (*starvation*) para uma fila de veículos que for tipicamente maior que a outra, a medida que novos veículos são continuamente adicionados às filas.

- **janela de tempo:** política de controle para interseções tradicional aplicada através de *traffic lights* ou **sinaleira**; fornece fatias de tempo limitadas para cada via (por exemplo 15s ou 20s); dessa forma veículos não sofrem de espera indefinida; entretanto uma fila longa  $q_m$  pode ficar bloqueada, aguardando o esgotamento do tempo de uma fila vazia  $q_n$ , como ocorre frequentemente na realidade.

As três políticas são tipicamente adotadas para escalonamento de processos em CPUs, mas todas poderiam ser usadas para controle de interseção em trânsito, desde que o suporte adequado seja provido (o que já ocorre com a política *janela de tempo*). De forma geral, políticas de controle para interseções podem ser classificadas segundo o tipo de operação, a saber: (i) algoritmos não-adaptativos e (ii) algoritmos adaptativos. A sinaleira convencional é um exemplo de política não-adaptativa, pois normalmente não avalia o tráfego para ajustar a janela de tempo de cada cor apresentada aos motoristas. Algoritmos enfocados neste trabalho, como FIFO e SJF, são exemplos de políticas adaptativas, e são baseadas nos comportamento dos veículos. Neste caso, veículos precisam interagir entre si ou com a infraestrutura para saber quem está na maior (ou menor) fila, ou quem chegou antes na fila de forma a decidir como a sincronização ocorrerá em uma interseção.

### IV. IMPLEMENTAÇÃO

#### A. O simulador Siafu

Como o suporte adequado ainda não está amplamente disponível para avaliar todas as políticas anteriormente descritas, foi construída uma simulação usando o simulador Siafu [6]. Este simulador basicamente é uma API para simulação de ambiente ubíquos de propósito geral, desenvolvido no contexto de MAS.

Para programar através desta API, é necessário descrever o comportamento dos agentes no cenário, mais precisamente descrever (i) o movimento dos veículos nas vias, (ii) os lugares específicos onde os veículos podem circular, ou seja, as vias (ruas) dispostas em um mapa, e (iii) as sobreposições (*overlays*) que são as regiões do mapa que se destinam a originar alterações na simulação. Por exemplo, à noite, o fluxo de veículos é menor. No final da tarde, quando as pessoas estão retornando as suas casas, há maior movimentação de veículos. O recurso de sobreposições não foi usado no escopo deste trabalho.

#### B. Desafios e detalhes de implementação

A simulação em questão destina-se à travessia de uma interseção por duas filas de carros. Uma restrição da implementação é que interseção comporta vias de sentido único, cenário que ocorre na realidade. Como o Siafu é um simulador ubíquo de propósito geral e não apresenta

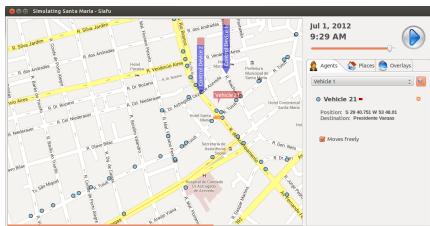


Figura 1. Captura de tela de uma simulação com 2 dispositivos de controle para a cidade de Santa Maria (RS), apresentando informações sobre um veículo (*vehicle 21*) na área lateral ao mapa

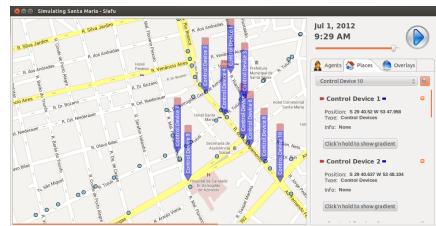


Figura 2. Captura de tela de uma simulação para 10 dispositivos de controle para a cidade de Santa Maria (RS), apresentando informações sobre específicas sobre dispositivos de controle (*Control Device 1* e *Control Device 2*) na área lateral ao mapa

a implementação do conceito de interseção (ou seção crítica), uma extensão através do método *isNear()* foi realizada para detecção de movimento de veículos na seção crítica e, então, o tratamento de conflito entre agentes em interseções (implementado por uma política), pode ser executado. A implementação deste mecanismo foi materializada através do dispositivo de controle ou *Control Device* proposto neste trabalho, que executa uma política de controle de interseção específica.

Basicamente, cada veículo  $c_i$  entra em contato com a infraestrutura (mais precisamente com o dispositivo de controle) através da emissão de uma mensagem  $m_i$  para solicitar sua passagem na interseção, de acordo com política em questão. O dispositivo de controle aplica a política e decide a ordem na qual o veículo deve passar a interseção, e informa ao veículo através de uma mensagem  $m_i - 1$ . Cada dispositivo de controle funciona de forma independente, sem comunicação inter-dispositivo. A comunicação veículo-infraestrutura poderia ser suportada pelos protocolos IEEE 802.11p (WiFi) ou GSM/GPRS e 3G (redes móveis de celular).

Uma captura de tela para uma simulação com 2 dispositivos de controle para cidade de Santa Maria (RS) é apresentada na Figura 1. Na figura, há duas áreas para visualização, (i) o mapa, com um sub-conjunto de ruas da cidade e (ii) informações sobre a simulação (no caso, informações sobre um veículo (*vehicle 21*)).

O suporte do Siafu também permite obter informação sobre os dispositivos de controle *Control Device*, como mostrado na Figura 2, desde que seja selecionada a aba correta na interface da simulação.

## V. AVALIAÇÃO EXPERIMENTAL

Após a implementação de um cenário (para um conjunto de vias da cidade de Santa Maria) com apoio do simulador Siafu, foram realizados experimentos para avaliar o desempenho computacional das políticas de controle para interseções, considerando o tempo médio de espera para iniciar uma travessia. A análise de resultados obtidos nos experimentos, partiu da realização de diferentes testes no ambiente controlado. Foram realizadas simulações com cada uma das três políticas descritas na seção III, variando a quantidade de agentes (veículos) da simulação. Nos experimentos, foram usados até 1.000 veículos, isto é,

1.000 é o número total (máximo) de processos (veículos) que foram gerados na simulação. Veículos se deslocam com a velocidade de 40km/h. Não há aceleração. O período de simulação foi de 10 dias inteiros. O primeiro e o último dia foram desconsiderados. As jornadas foram, então, de 24 horas (o trânsito foi considerado o mesmo durante todo o período).

Mais especificamente, foram realizados dois experimentos, (i) um teste inicial com 20, 40, 60, 80 e 100 veículos e um dispositivo de controle, e (ii) outro experimento de maior escala, com até 1.000 veículos, para simular situações de tráfego mais intenso, como a hora do *rush* ou a saída de algum espetáculo. No segundo experimento foram considerados 10 dispositivos de controle para gerenciamento de interseções em vias de sentido único. O resultado dos experimentos é sumarizado através de valores apresentados nos gráficos das Figuras 3 e 4, respectivamente. Em ambos experimentos, a política *janela de tempo* foi usada como base e aplica janelas de tempo de 20s e 25s, para os dispositivos de controle.

No gráfico apresentado na Figura 3, com 20 agentes simulados, a política por *janela de tempo* obteve a média de quase 1 minuto de espera por veículo para realizar a travessia. Com 50 agentes, foram 2.3 minutos, com 90 agentes foram aproximadamente 3 minutos, respectivamente. Os algoritmos de escalonamento de processos FIFO e SJF conseguiram desempenho equivalente, diferindo pouco nos resultados quando comparados entre si, conforme pode ser observado graficamente. Contudo, de forma geral ambos algoritmos foram mais eficientes que a *janela de tempo*.

O gráfico apresentado na Figura 4 representa situações de tráfego mais intenso. Com até 300 agentes simulados, a política tradicional por *janela de tempo* foi a mais ineficiente quanto ao tempo de espera médio para realizar a travessia. Para 300 até 600 agentes, ocorreu pouca variação no desempenho dos três algoritmos quanto ao tempo de espera, de acordo com o experimento. Para grande quantidade de veículos, SJF obteve os melhores resultados.

De forma geral, os experimentos mostraram que os algoritmos de escalonamento de processos centrados em veículos (SJF e FIFO), quando aplicados para controle de interseções, podem oferecer resultados interessantes,

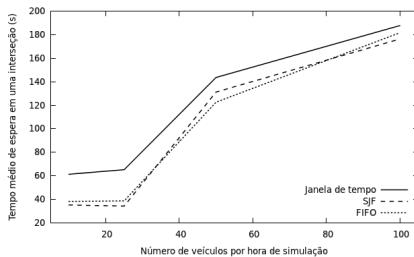


Figura 3. Gráfico indicando tempo médio de espera por interseção na simulação com até 100 carros para um conjunto de vias da cidade de Santa Maria (RS) com um dispositivo de controle

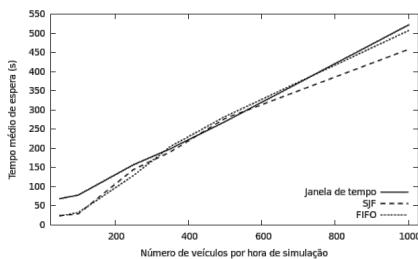


Figura 4. Gráfico indicando tempo médio de espera por interseção na simulação com trânsito intenso para um conjunto de vias da cidade de Santa Maria (RS) com 10 dispositivos de controle

melhorando o fluxo do trânsito, e comprovando nossas hipóteses de trabalho (políticas baseadas em fluxo são mais interessantes do que políticas baseadas em janelas de tempo estáticas). Esta melhoria ocorre porque, diferentemente do controle semafórico aplicado em sistemas reais, as técnicas de escalonamento de processos usadas nesta implementação são associadas ao fluxo de veículos, e podem ser adaptadas de acordo com a demanda.

## VI. CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho demonstrou que o controle de passagem em interseções, tipicamente implementado nas vias brasileiras por sinalizadoras, pode ter uma melhoria significativa com a aplicação de algoritmos com o suporte de VANETs. A proposição de mecanismos, de forma geral, mais adequados para controlar redes de transporte é necessária, dado que a dificuldade de gerenciamento do trânsito é um problema presente e constante no nosso quotidiano. Conforme argumentado anteriormente, soluções existentes são preferencialmente baseadas em janelas de tempo não adaptativas. Porém, percebe-se que diferentes políticas podem ser implementadas, gerando oportunidades para o estudo de soluções com diferentes graus de dificuldade e para o provimento de serviço mais eficiente.

A aplicação-alvo deste artigo envolve redes de transporte e mobilidade urbana, assuntos que têm despertado

muito interesse em toda a sociedade. E, de fato, soluções para melhoria da mobilidade urbana e de processos de transporte tem sido implementadas e estão mais acessíveis, como semáforos adaptativos recentemente instalados Porto Alegre (RS) e sistemas de informação sobre condições de tráfego (*Google Transit*, *Waze*, por exemplo).

Este trabalho representa mais um passo na investigação de políticas para controle semafórico mais eficiente. Trabalhos futuros incluem a implementação de políticas mais sofisticadas, por exemplo, aquelas que penalizam chaveamento entre filas de veículos (quando uma fila pára a passagem de veículos e outra fila começa a passagem de veículos, uma penalidade deve ser adicionada). Políticas descentralizadas (sem dispositivo de controle) também precisam ser avaliadas, antes de serem efetivamente colocadas em prática. Neste caso, a comunicação inter-veicular deve ser considerada, bem como o uso de simuladores mais específicos para redes de transporte [2].

## AGRADECIMENTOS

Este trabalho foi conduzido no escopo dos projetos CTIC/RNP SIMTUR e CNPq/FAPERGS RS-SOC PRO-NEX número 10/0049-7.

## REFERÊNCIAS

- [1] A. L. C. Bazzan. "Opportunities for multiagent systems and multiagent reinforcement learning in traffic control". In *Autonomous Agents and Multiagent Systems*, 18(3):342-375, June 2009.
- [2] M. Behrisch, L. Bieker, J. Erdmann, D. Krajzewicz. "SUMO - Simulation of Urban MOBility: an overview", In *Proc. 3rd International Conference on Advances in System Simulation (SIMUL 2011)*. ThinkMind, Oct. 2011, pp. 63-68.
- [3] K. Dresner, P. Stone. "Multiagent traffic management: a reservation-based intersection control mechanism", *Autonomous Agents and Multiagent Systems*, 2004. AAMAS 2004. In *Proc. 3rd International Joint Conference on*, vol., no., July 2004, pp. 530-537.
- [4] M. Ferreira, R. Fernandes, H. Conceição, W. Viriyasitavat, O. K. Tonguz. "Self-organized traffic control", In *Proc. 7th ACM International Workshop on Vehicular InterNETworking (VANET '10)*. ACM, New York, NY, USA, pp. 85-90, 2010.
- [5] D. Krajzewicz, E. Brockfeld, J. Mikat, J. Ringel, C. Rossel, W. Tutschcheerer, P. Wagner, R. Wosler. "Simulation of modern traffic lights control systems using the open source traffic simulation SUMO", In *Proc. 3rd Industrial Simulation Conference 2005*, EUROSIS-ETI, 2005, pp. 299-302.
- [6] M. Martin, P. Nurmi. "A generic large-scale simulator for ubiquitous computing". In *Mobile and Ubiquitous Systems, Annual Int. Conference on*, 0:1-3, 2006.
- [7] B. Mugnela, M. Netto. GenPolis - Prototipagem e aplicação de uma ferramenta especializada para otimização via algoritmos genéticos de planos fixos de sinalização semafórica em sub-redes urbanas. In *VIII Simpósio Brasileiro de Sistemas de Informação - Trilhas Técnicas*, 2011, pp. 198-209.
- [8] M. Vasirani, S. Ossowski. "A market-inspired approach to reservation-based urban road traffic management", *Autonomous Agents and Multiagent Systems*, 2009. AAMAS 2009. In *Proc. 8th International Conference on*, vol. 1, no., July 2009, pp. 617-624.

# **SNMP Mobile: Uso de Plataforma Móvel para Gerenciamento de Rede Através do Protocolo SNMP**

Caio Modena de Lucena Valter Oliveira Barbosa Cristina M. Nunes

Faculdade de Informática – PUCRS

caio.modena@gmail.com, vobarbosa@gmail.com, cristina.nunes@pucrs.br

**Resumo**—Este trabalho propõe um estudo baseado na junção entre computação móvel e gerência de redes de computadores, com o intuito de apresentar uma aplicação (SNMP Mobile) para auxiliar no gerenciamento de redes. O SNMP Mobile foi desenvolvido para a plataforma Android e seu objetivo é auxiliar o profissional da TI a gerenciar e obter informações dos dispositivos de sua rede, mesmo não estando presencialmente na sua estação de trabalho.

## I. INTRODUÇÃO

No ano de 1992, a IBM criou, em conjunto com a BellSouth, o *IBM Simon Personal Communicator*, conhecido como o primeiro *smartphone* da história. Além das básicas e fundamentais ligações telefônicas, o IBM Simon tinha calendário, agenda, relógio, calculadora, bloco de notas, acesso a e-mails e jogos. Ele não possuía botões, e sim uma tela *touchscreen* [8]. Porém, o uso da palavra *smartphone* para designar conceitos conhecidos até hoje foi usada pela primeira vez em 1997 através da Ericsson com seu protótipo de celular GS88. Seu diferencial consistia na integração das funções de um celular comum com um PDA (*Personal Digital Assistant*), ou seja, como uma agenda pessoal e não mais apenas um serviço de ligações e mensagens.

Após alguns anos veio a popularização do *smartphone* através da Apple, com o lançamento do Iphone em 2007, utilizando-se apenas de uma tela *touchscreen* e criando uma loja virtual de aplicativos. Dessa maneira, o usuário começou a moldar o celular ao seu gosto, fugindo da rigidez anterior. Finalmente, a Google começou a entrar no ramo ao efetuar a compra da Android, uma pequena empresa especializada em sistemas embarcados em 2005 [4]. Ao fim de dois anos de especulações sobre o real significado da compra, a empresa anunciou, no dia sete de novembro de 2007, que estava lançando o Android como uma plataforma baseada em *kernel Linux*, além da criação da OHA (*Open Handset Alliance*), uma junção da Google com outras diversas empresas parceiras [1]. No ano de 2008 foi lançado o primeiro aparelho celular com o sistema operacional da Google, o HTC G1.

Com a constante evolução da tecnologia e também pelo aumento gradativo de usuários, os dispositivos móveis estão dando origem à criação dos mais diversos tipos de aplicativos, sendo que muitos deles oferecem recursos extremamente úteis no auxílio à gestão e monitoramento de serviços utilizados por uma organização. Contudo, conforme uma pesquisa realizada sobre a utilização empresarial de plataforma móvel [5], as empresas de TI (Tecnologia da Informação) estão recém acordando para o desenvolvimento de soluções específicas para dispositivos móveis. Um gerente de redes, por exemplo, possui infinitos aplicativos ou *softwares* que auxiliam sua atividade, ora no projeto de uma rede, ora na análise dos

diferentes tipos de equipamentos e suas configurações. Todavia, a maior parte destes *softwares* é alocada em *desktops* ou servidores fixados em locais estratégicos da organização, não permitindo total agilidade do gerente de rede enquanto o mesmo não estiver no local.

Dado esses fatos, este trabalho tem como objetivo propor um alinhamento entre o uso dessas ferramentas de gerência de rede e recursos oferecidos pelos dispositivos móveis com o gerenciamento de uma rede de computadores. O trabalho descreve o desenvolvimento de uma aplicação voltada ao profissional da TI, com a meta de agilizar e facilitar o processo de gerência e monitoramento da rede. Esse aplicativo implementa um MIB Browser que permite visualizar a hierarquia de uma MIB (*Management Information Base*) na forma de uma árvore, proporcionando informação de cada nodo que retorna dados através do protocolo SNMP (*Simple Network Management Protocol*). Pretende-se, através da aplicação, aumentar a agilidade em obter determinadas informações oferecidas pelo gerenciamento, como por exemplo, o *status* de uma interface de rede. Este *status* é fornecido usando as informações obtidas através de objetos da MIB-II. A aplicação desenvolvida dispõe ao usuário algumas taxas pré-estabelecidas, que podem ser escolhidas pelo mesmo, sem que seu cálculo seja realizado manualmente. O aplicativo obtém as informações, realiza os cálculos e retorna os resultados para o usuário. Este processo significa ganho de tempo e produtividade para o administrador de uma rede.

Este documento está dividido da forma como segue. A Seção II apresenta um detalhamento teórico sobre a área de gerência de redes, o protocolo SNMP com suas versões e a MIB-II. A Seção III descreve as características da arquitetura, da implementação e da utilização da aplicação de gerência de redes através de plataforma móvel, o *SNMP Mobile*. 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.

Conforme [9], o velho modelo de um único computador atendendo a todas as necessidades computacionais da organização foi substituído por outro em que os trabalhos são realizados por um grande número de computadores separados, porém interconectados. Esses sistemas são chamados de redes de computadores.

Em conformidade com o que diz [7], as redes de computadores atuais são compostas por uma grande variedade de dispositivos que devem se comunicar e compartilhar recursos. Para gerenciar esses sistemas e as próprias redes, um conjunto eficiente de ferramentas de gerenciamento torna-se necessário, sendo de extrema importância a utilização de técnicas padronizadas para a correta representação e o intercâmbio das informações obtidas.

O primeiro protocolo para gerência de rede utilizado foi o SGMP (*Simple Gateway Monitoring Protocol*), com seu lançamento no ano de 1987. Entretanto, o SGMP era restrito à monitorização de gateways, e, por esta razão, não era uma ferramenta de uso geral para gerenciamento de redes. A premência de uma ferramenta para gerenciamento de uma forma mais genérica fez emergirem mais algumas abordagens, como HEMS (*High-Level Entity Management Systems*), SNMP e o CMOT (*CMIP Over TCP/IP*) [2].

Em meados de 1988, a IAB (*Internet Architecture Board*) reviu estas propostas e aprovou o desenvolvimento do SNMP para curto prazo e o CMOT para longo prazo. Desde então diversos produtos de gerenciamento já foram desenvolvidos utilizando o SNMP e, de acordo com [7], até hoje é o protocolo que possui o maior número de implementações.

O termo SNMP é atualmente usado para se referir a uma coleção de especificações para o gerenciamento da rede, incluindo o próprio protocolo, a definição das estruturas da informação e conceitos associados. Segundo [10], o protocolo SNMP é uma ferramenta para o gerenciamento de dispositivos conectados a uma rede de computadores. A rede TCP/IP é composta por gerentes e agentes trocando mensagens entre si.

Gerentes são responsáveis pela comunicação e gerenciamento de dispositivos gerenciados, desde que tais dispositivos possuam o *software* do agente. Agentes, por outro lado, residem em objetos da rede como servidores, estações de trabalho e roteadores, provendo informações para os gerentes [10]. A seguir será explicada a definição e também o funcionamento de uma MIB, mostrando como se relaciona com o protocolo SNMP.

MIBs são especificações contendo definições de informações gerenciadas, assim a rede pode ser remotamente monitorada, configurada e controlada [6]. De acordo com [10], são arquivos que descrevem informações de objetos a serem gerenciados e são fundamentais para o protocolo SNMP. Informações contidas no seu interior são a base para a comunicação entre os gerentes e os agentes de uma rede gerenciada. Segundo [3], para o armazenamento de informações na MIB é definida uma estrutura em árvore, composta por nós, onde cada nó tem um OID (*Object Identifier*) e um nome associado. Cada nó da árvore pode ter uma nova subárvore associada.

### III. SNMP MOBILE

A motivação inicial para o desenvolvimento do *SNMP Mobile* foi a de auxiliar o profissional da TI a gerenciar e obter informações dos dispositivos de sua rede local, mesmo não estando presencialmente na sua estação de trabalho. Além disso, foi considerado um caso real em

que um gerente de rede, por não estar em suas dependências – *um datacenter* - não pôde obter um dado significativo de um computador em um determinado momento. O processo de deslocamento entre o setor no qual o gerente se encontrava até o datacenter foi demorado e por consequência houve perda de um tempo considerável e prejuízo para a organização.

A aplicação desenvolvida roda sobre a plataforma Android, de propriedade da Google, ou seja, é voltada à dispositivos que utilizam esse sistema operacional. A plataforma Android foi a escolhida para o desenvolvimento, pois é de código aberto e muito utilizada em uma vasta quantidade de dispositivos disponíveis no mercado.

O aplicativo desenvolvido é chamado de *SNMP Mobile* e foi construído com o objetivo de ser uma ferramenta intuitiva e de fácil navegação, que utiliza os recursos oferecidos pelo SNMP para controle dos ativos da rede. Para a coleta das informações, é necessário que o dispositivo móvel tenha uma conexão à *internet*, seja via rede *wireless* da empresa ou via rede 3G<sup>1</sup>. Também é necessário que o dispositivo que será consultado tenha o agente SNMP instalado e ativado.

A aplicação desenvolvida tem suporte para as versões 1 e 2c do protocolo SNMP, implementa todos os grupos da MIB-II e oferece quatro operações de gerenciamento. Duas de suas operações são da versão 1 (*get* e *getNext*), uma operação é da versão 2c (*getBulk*) e uma operação chamada *walk*, cujo objetivo é apresentar todos os objetos de um determinado grupo da MIB-II. A operação *walk* trabalha em conjunto com a operação *getNext*.

O *software* também realiza a integração de diversas métricas para gerar uma nova informação através de cálculos, muitas vezes transformando valores em porcentagens, a fim de ter uma melhor visualização e entendimento do funcionamento da rede. As métricas utilizadas estão apresentadas a seguir:

- **Status da interface:** comparação entre as métricas *ifAdminStatus* e *ifOperStatus*.
- **Porcentagem de erro de entrada:**  $\text{ifInErrors} / (\text{ifInUcastPkts} + \text{ifInNcastPkts})$ .
- **Porcentagem de erro de saída:**  $\text{ifOutErrors} / (\text{ifOutUcastPkts} + \text{ifOutNcastPkts})$ .
- **Porcentagem de descartes de entrada:**  $\text{ifInDiscards} / (\text{ifInUcastPkts} + \text{ifInNcastPkts})$ .
- **Porcentagem de descartes de saída:**  $\text{ifOutDiscards} / (\text{ifOutUcastPkts} + \text{ifOutNcastPkts})$ .
- **Número de descartes realizados devido ao recebimento de pacotes de protocolo desconhecido:** verificação se *ifInUnknownProtos* e *ifInDiscards* estão crescendo proporcionalmente, caso contrário haverá detecção de problema analisando *ifInUnknownProtos*.
- **Taxa de utilização de uma interface:** (total de bits por segundo) / *ifSpeed*.

<sup>1</sup> Rede móvel oferecida pelas operadoras de telefonia para acesso à *internet*.

- **Congestionamento de rede:** Aumento no valor de *ifOutDiscards* e diminuição no valor de *ifOutOctets*.
- **Taxa de Tráfego IP de entrada de uma entidade:**  $(\text{ifInUcastPkts} + \text{ifInNUcastPkts}) / \text{ipInReceives}$ .
- **Taxa de Tráfego IP de saída de uma entidade:**  $(\text{ifOutUcastPkts} + \text{ifOutNUcastPkts}) / \text{ipOutRequest}$ .
- **Porcentagem de erros de datagramas IP na entrada:**  $(\text{ipInDiscards} + \text{ipInHdrErrors} + \text{ipInAddrErrors}) / \text{ipInReceives}$ .

Na construção do projeto foram utilizadas diversas tecnologias empregadas em ambientes de desenvolvimento de *software*. Entre estas tecnologias pode-se ressaltar a própria plataforma Android, o banco de dados nativo do Android conhecido como *SQLite*, a linguagem de programação *Java* e a biblioteca *snmp4j.jar*, que oferece uma API (*Application Programming Interface*) *JavaSNMP* com suporte a geração de comandos (gerentes), bem como o comando de respostas (agentes). Além disso, o *SNMP Mobile* foi desenvolvido através do *Eclipse*.

Os usuários da aplicação são principalmente os gerentes e/ou administradores de rede, que farão uso da ferramenta através de seus dispositivos móveis. Para o funcionamento do *SNMP Mobile* basta que o usuário inicie o aplicativo a partir do menu de aplicativos de seu aparelho.

A seguir estão descritas as funcionalidades presentes no *SNMP Mobile*.

Ao ser inicializada, a aplicação carrega a tela inicial que apresenta os campos para preenchimento dos dados (endereço de IP, porta) de um determinado ativo da rede, bem como a definição da *Community String*, como demonstra a Figura 1.



**Figura 1 - Tela Inicial do SNMP Mobile.**

Na tela inicial também é possível acessar o menu de dispositivos favoritos, que é composto por duas abas. A aba Cadastro fornece campos para Nome, Endereço de IP, Porta e *Community String*. Finalizando o preenchimento dos campos, o usuário escolhe a opção de salvar o favorito em uma lista que estará disponível na aba Lista, ou caso

seja necessário, pode escolher a opção de limpar o formulário.

Ao salvar o favorito, ele é automaticamente transferido para a aba de lista de favoritos, sendo listados por ordem de inserção. Nesta aba, o nome e o IP são mostrados explicitamente. Ainda existe a opção edição e/ou exclusão de favoritos, possibilitada ao pressionar a tela sobre o favorito desejado por dois segundos. Com um simples clique em qualquer favorito na aba de listagem, o usuário é redirecionado para a página inicial, com campos já preenchidos pela opção escolhida. Através desta funcionalidade há um ganho de agilidade caso a consulta seja repetida muitas vezes para um mesmo dispositivo. Partindo dos dados do dispositivo escolhido, o sistema exibe uma tela de opções, com duas opções de escolha: MIB Browser ou Monitoramento. Escolhida opção MIB Browser, o aplicativo apresenta todos os elementos de uma MIB-II em forma gráfica, conforme mostra a Figura 2.



**Figura 2 – Estrutura gráfica da MIB-II.**

Após escolhido o objeto desejado, o usuário é direcionado à tela que contém as opções para o comando SNMP que será escolhido. Os seguintes campos são descritos abaixo:

- **Operação:** esta opção apresenta uma relação das operações disponíveis para consulta, tais como: *GET*, *GETNEXT*, *GETBULK* e *WALK*.
- **Versão SNMP:** onde é definida a versão do SNMP para a operação escolhida. Na opção *GETBULK* a versão é definida automaticamente para 2c.
- **Instância:** local onde é escolhida a instância do objeto selecionado, por padrão este valor é zero.
- **Timeout:** tempo de limite da consulta ao dispositivo até obter uma resposta.
- **Número de Informações (GETBULK):** campo onde é definido um número de objetos que a operação *GETBULK* irá retornar. Este campo só é habilitado em caso de escolha do *GETBULK*.

Caso a opção escolhida na tela de opções seja Monitoramento, são apresentadas diversas métricas e taxas de monitoramento pré-definidas, como demonstradas na Figura 3. O usuário apenas seleciona uma delas para receber os resultados na tela.

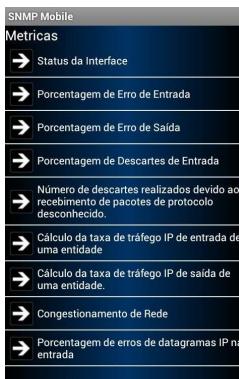


Figura 3 – Tela com métricas pré-definidas.

O resultado, independente do menu escolhido, é exibido em uma nova tela. A tela de resultados exibe as informações retornadas, bem como o endereço de IP do dispositivo consultado, a métrica ou OID do objeto selecionado e o horário/data da consulta.

É possível compartilhar a informação adquirida através de diversos meios de comunicação oferecidos pelo dispositivo móvel, como e-mail e rede social. Caso deseje, o usuário pode salvar o resultado na memória do cartão de memória do smartphone. O arquivo a ser salvo será em formato de texto e seu nome contém o endereço de IP do dispositivo.

#### IV. TRABALHOS RELACIONADOS

A aplicação *SNMP MIB Browser*, desenvolvida pela *Zoho Corporation*, permite ao usuário verificar os dados da MIB dos dispositivos de rede habilitados. A ferramenta é *free* e suporta todas as versões do SNMP.

A aplicação *SNMP Management Service*, desenvolvida por Y. Matsumoto, possui funções como navegador de MIBs, recepção e transmissão de *traps*.

Em comparação com estas aplicações, o *SNMP Mobile* apresentou principalmente vantagens em funcionalidades com fins de praticidade, como a possibilidade de cadastrar dispositivos favoritos, apresentação de *MIB Browser* gráfica e também o compartilhamento das informações obtidas. Além disso, sua maior vantagem consiste no oferecimento de métricas pré-estabelecidas ao usuário, não é necessário que o gerente de rede combine diversos objetos da MIB para obter a informação que realmente deseja.

#### V. CONSIDERAÇÕES FINAIS

O principal foco relatado neste trabalho aborda uma integração entre a gerência de redes e dispositivos móveis. Tal integração resume-se a uma aplicação voltada à *smartphones* com o benefício de fornecer uma forma ágil e ativa de gerenciar e controlar os equipamentos presentes

na rede local de computadores da organização, mais especificamente fazendo uso do protocolo SNMP. A escolha por um aplicativo móvel ao invés de simplesmente uma interface web para ser acessada pelo *browser* do *smartphone* deu-se por questões de praticidade e funcionalidade.

Uma aplicação exclusiva para dispositivos móveis oferece funções muito mais específicas para o usuário, aproveitando-se também da rapidez da coleta de resultados através da tela *touchscreen*. Ademais, existem diversidades interfaces web, levando-nos a optar por explorar um mercado ainda ínfimo em opções.

A partir do uso e do sucesso obtido na gerência de alguns equipamentos no setor de TI de uma empresa comercial no ramo automotivo de Porto Alegre, surgiu a ideia de que a aplicação desenvolvida deixe de ser uma aplicação apenas de consulta de informações e obtenha também a operação de escrita para poder alterar determinadas variáveis do dispositivo.

Para tal função, será necessária a implementação da operação *SET*. Todavia, a implementação desta operação no *SNMP Mobile* exigirá um grau elevado de segurança.

Além disso, a aplicação futuramente poderá oferecer gráficos resumidos das métricas, facilitando a visualização do usuário.

#### REFERÊNCIAS

- [1] Android Open Source Project. **Filosofia da Empresa**. Disponível em <<http://source.android.com/about/philosophy.html>>. Acesso em: setembro/2012.
- [2] BLACK, Tomas Lovis. **Comparação de Ferramentas de Gerenciamento de Redes**. Artigo. Instituto de Informática. UFRGS. Porto Alegre.2008.
- [3] CORREIA, Marcelo Fernandes. **Gerência de Redes**. Trabalho de final de curso. União Educacional de Minas Gerais. Disponível em <[www.cect.unimonetres.br/arquivos/dcc/gilmara/1144.pdf](http://www.cect.unimonetres.br/arquivos/dcc/gilmara/1144.pdf)>. Acesso em outubro/2012.
- [4] ELGIN, Ben. **Google Buys Android for Its Mobile Arsenal**. Disponível em <<http://www.businessweek.com/stories/2005-08-16/google-buys-android-for-its-mobile-arsenal>>. Acesso em: agosto/2012.
- [5] MBI. **Pesquisa sobre a utilização empresarial de Plataforma Móveis**. Disponível em <<http://www.mbi.com.br/mbi/biblioteca/relatorios/2011-11-pesquisa-utilizacao-empresarial-plataformas-movveis>>. Acesso em: agosto/2012.
- [6] PERKINS, David; McGINNIS Evan. **Understanding SNMP MIBs**. Prentice Hall PTR, Upper Saddle River, New Jersey, 1997.
- [7] PINHEIRO, José Mauricio Santos. **Gerenciamento de Redes de Computadores: Uma breve introdução**. Artigo disponível em: <[http://www.projetedoderedes.com.br/artigos/artigo\\_gerenciamento\\_d\\_e\\_redes\\_de\\_computadores.php](http://www.projetedoderedes.com.br/artigos/artigo_gerenciamento_d_e_redes_de_computadores.php)>. Acesso em: agosto/2012.
- [8] SAGER. **Before iPhone and Android Came Simon, the First Smartphone**. Disponível em <<http://www.businessinsider.com/mobile-phone-firsts-2011-8#the-first-phone-to-actually-be-called-a-smartphone-ericsson-gs88-1997-10>>. Acesso em: setembro/2012.
- [9] TANENBAUM, Andrew S. **Redes de Computadores** – 5ª edição. Pearson Prentice Hall, São Paulo, 2011.
- [10] WALSH, Larry. **SNMP MIB Handbook. Essential Guide to MIB Development, Use, and Diagnosis**. Wyndham Press, 2008.

# Utilizando a plataforma Arduino para a comunicação entre dispositivos embarcados e redes TCP/IP

Alexandre Silva Rodrigues

Colégio Técnico Industrial de Santa Maria  
Universidade Federal de Santa Maria, UFSM  
Santa Maria, Brasil  
alexandre.rodrigues@redes.ufsm.br

Tiago Antonio Rizzetti

Colégio Técnico Industrial de Santa Maria  
Universidade Federal de Santa Maria, UFSM  
Santa Maria, Brasil  
rizzetti@gmail.com

**Resumo**—A crescente expansão da conectividade entre dispositivos na internet e os recursos da plataforma Arduino possibilitam a comunicação entre dispositivos embarcados e redes TCP/IP. Uma grande aplicação desse contexto é um sistema de autenticação de usuários e controle de acesso em ambientes restritos. A Arquitetura ESC apresenta-se como uma proposta para esses aspectos e utiliza a plataforma Arduino para interligar um conjunto de hardware e software por meio de protocolos de comunicação que possibilitam a transferência de informações entre dispositivos embarcados e a rede tradicional de computadores.

**Palavras-chave**—Internet das coisas; Arduino; controle de acesso; dispositivos embarcados.

## I. INTRODUÇÃO

Conforme a crescente evolução tecnológica, a internet vive um novo momento. Trata-se do conceito de Internet das Coisas (*Internet of Things* ou *IoT*), que proporciona uma vasta gama de recursos a serem explorados e mudanças constantes na forma de comunicação via internet [1].

A Internet das Coisas é formada por uma rede que interconecta objetos (qualquer dispositivo conectado a Internet), onde qualquer objeto pode enviar informações para outros objetos e pessoas [1]. Ou seja, é a interconexão entre objetos físicos e computadores com a internet, transformando tudo que nos cerca em objetos inteligentes (*smart objects*) [2]. A conexão de tais objetos com a internet acontece por meio de endereços IP e URLs, assim como funcionam as páginas Web atuais. Desta forma, os objetos em nosso ambiente tornam-se participantes ativos, ou seja, compartilham informações com outros membros da rede, com capacidade de reconhecer eventos e mudanças no ambiente e de agir de forma autônoma. Com isso, em um mundo onde o real, o digital e o virtual convergem, é possível criar ambientes inteligentes e inúmeras aplicações a serem desenvolvidas [3].

Nesse contexto, a plataforma Arduino desempenha um papel fundamental na interconectividade entre diversos dispositivos. O Arduino é uma plataforma utilizada para programação de microcontroladores responsáveis por processar entradas e saídas entre o dispositivo e os

componentes externos conectados a ele. A principal contribuição da plataforma Arduino é facilitar a integração entre dispositivos eletrônicos e a rede tradicional de computadores. Dessa forma, provendo uma nova gama de oportunidades de softwares e hardwares capazes de automatizar diversas atividades cotidianas, desde o controle de portões eletrônicos, via celular, até sistemas complexos para monitoramento de ambientes. Seu hardware consiste em um projeto simples de hardware livre para o controlador, com um processador Atmel AVR e suporte embutido de entrada/ saída. O software consiste de uma linguagem de programação padrão e do *bootloader* que executa na placa [4] [5].

O Arduino pode utilizar diferentes componentes que permitem a comunicação entre dispositivos ou até mesmo entre plataformas embarcadas e qualquer outro dispositivo na rede IP. Por exemplo, a plataforma Arduino pode ser usada para enviar um conjunto de dados de sensores a ele conectados para um cliente qualquer na Web, através do uso do protocolo HTTP [4] [5].

Relacionado com o contexto de conectividade que a Internet das Coisas oferece e utilizando a plataforma Arduino como base de sua arquitetura, o Projeto ESC (desenvolvido pelos autores deste artigo) desenvolve uma arquitetura (denominada Arquitetura ESC) capaz de realizar o gerenciamento de permissões e controle de acesso a ambientes físicos. Este artigo visa apresentar essa arquitetura (hardware e software), abordando o uso da plataforma Arduino para realizar a comunicação entre dispositivos embarcados e redes TCP/IP. Além disso, será descrito um protocolo utilizado para realizar a comunicação entre duas placas via porta serial e serão apresentados resultados de desempenho da arquitetura em questão.

## II. ARQUITETURA ESC

Em instituições onde existe uma alta rotatividade de pessoas, um gerenciamento de identidades é fundamental para manter um controle de acesso a ambientes restritos. Esse processo envolve as seguintes ações:

- Autenticação: realiza o teste da identificação de determinado usuário. Pode ocorrer por diferentes métodos: biometria, senhas, cartão, entre outros;
- Autorização: é a capacidade de estabelecer se uma identidade tem a permissão de acessar um local;
- Auditoria: cada evento deve ser registrado para eventuais consultas [6].

Para tanto, é preciso um sistema capaz de realizar tais ações e garantir que apenas pessoas autorizadas possam acessar um ambiente. Com o objetivo de apresentar uma solução para isso, a Arquitetura ESC (*Environment Security Control*) apresenta uma proposta para o controle de tais locais, estabelecendo uma comunicação entre dispositivos de interação com o meio físico e um gerente que centraliza as informações. Essa arquitetura realiza autenticação de cada usuário que deseja acessar algum ambiente, permitindo ou não o seu acesso. Além disso, é possível gerar relatórios de acesso a cada local ou por usuários, para fins de auditoria. Tais funcionalidades possibilitam uma forma eficiente e segura de controle de acesso a ambientes restritos em instituições, além de substituir o uso de chaves físicas por chaves digitais que identificam cada usuário.

A Arquitetura ESC divide-se em dois subsistemas: o gerente (ESCPA), que aplica as políticas de permissões e realiza a conexão com o sistema; e um conjunto de hardware e software (ESCHA) baseado na plataforma Arduino, que atua como um dispositivo físico para coletar dados que são enviados para o gerente e realizar ações sobre o sistema através de atuadores [7].

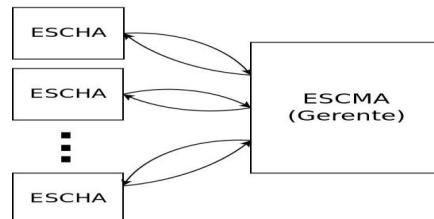
#### A. ESCMA

O gerente atua de forma centralizada sobre os diversos hardwares que compõe o sistema, armazenando informações sobre todo o sistema, o que lhe proporciona flexibilidade e escalabilidade. Um aspecto sobre o gerente é sua arquitetura escalável que permite incluir novos dispositivos físicos e/ou diferentes tipos de dispositivos de entrada de dados (RFID, biometria, teclado, etc.) ao sistema através da inclusão de novos conectores. Cada categoria de dispositivo utiliza uma classe de conectores que recebem os dados do ESCHA e após o processamento, retornam os dados para o ESCHA [8].

Após receber dados do ESCHA, o gerente realiza uma comunicação com sua base de dados, onde ficam cadastrados os usuários do sistema e as respectivas permissões ou restrições, que retorna com o tipo de acesso que determinado usuário possui para o local que está tentando acessar. De acordo com essa resposta, o ESCPA envia para o ESCHA qual ação deve ser realizada. Esta comunicação ocorre por meio de um conector específico.

Diferentes bases de dados podem ser utilizadas, pois, para cada base existe uma classe correspondente [8].

A comunicação entre os dois subsistemas da Arquitetura ESC é desenvolvida na camada de aplicação e utiliza a pilha de protocolos TCP/IP, que oferece as funcionalidades necessárias ao tráfego de dados entre os dispositivos e o gerente [7].



- Placa principal: é uma placa específica, desenvolvida para receber um microcontrolador da Atmel e os seguintes módulos: módulo de interface de rede (*Ethernet Shield*) e um módulo e antenas para leitura e escrita em cartões RFID. Também foram adicionadas relés para o acionamento de dispositivos externos (fechadura eletromagnética, por exemplo);
  - Placa auxiliar: contém um microcontrolador Atmel, um módulo de *display LCD* e um módulo e antenas para leitura e escrita em cartões RFID.

A arquitetura do sistema pode ser dividida em quatro partes:

- **Host:** o ESCHA funciona como um servidor web, tendo um endereço IP que o identifica na rede e podendo ser acessado via interface web;
  - Interface Visual: exibe mensagens no visor LCD, proporcionando uma interação do sistema com o usuário;
  - Dispositivo de autenticação: é o dispositivo que permite a entrada de dados (leitura de um cartão);
  - Interface de rede: permite a comunicação do ESCHA com o gerente [7].

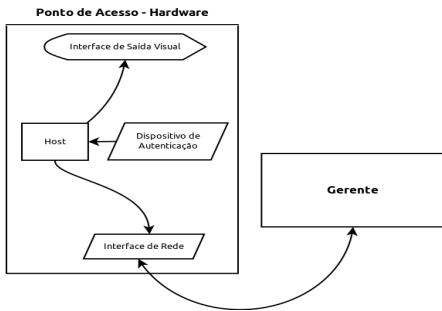


Fig. 2. Conexão do hardware com o gerente

#### *E. Comunicação entre as duas placas do ESCHA*

Para realizar a entrada de dados (leitura de cartões), os módulos e antenas RFID de cada placa são interligados, o que permite que ambas possam realizar esta operação. Sua principal aplicação é possibilitar que um local tenha um dispositivo de controle de acesso de entrada e de saída, sem necessitar que os dois dispositivos acessem o gerente de forma independente.

Existe ainda, uma conexão do pino Tx (transmissor) de uma placa ao pino Rx (receptor) da outra e, vice-versa. Esta conexão possibilita a transferência de dados entre ambas. A comunicação entre elas ocorre por meio de um protocolo de comunicação e utiliza a porta serial para a transferência de dados.

Cada placa atua de forma autônoma, controlando seus próprios sensores e atuadores. Entretanto, existe a necessidade de uma comunicação entre ambas. A placa auxiliar, além de realizar a entrada de dados, atua como um dispositivo de saída, exibindo mensagens em um visor LCD. Estas mensagens são obtidas e sincronizadas por meio da comunicação com a placa principal, utilizando a porta serial para realizar esta operação. Desta forma, existe uma interface de interação entre o usuário e o sistema que exibe informações sobre o processo de autenticação de um usuário, indisponibilidade de acesso a um local, possíveis falhas de comunicação com o gerente e atualizações referentes à data e horário. Todas estas informações são atualizadas por meio da comunicação entre gerente e placa principal. Portanto, novamente evidencia-se a importância de não necessitar uma conexão do gerente com cada placa.

#### *F. Protocolo de comunicação da Arquitetura ESC*

A Arquitetura ESC além de realizar a comunicação entre gerente e ESCHA por meio de redes TCP/IP, realiza uma a comunicação serial entre as placas principal e auxiliar, e a comunicação entre ESCHA e ESCMA. A Fig. 3 demonstra um cenário para aplicação da desta arquitetura e a forma como cada comunicação é realizada.

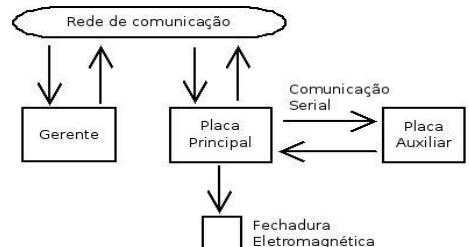


Fig. 3. Cenário de aplicação da Arquitetura ESC

Para estabelecer a transferência de dados entre as placas, por meio da comunicação serial, é necessário um protocolo de comunicação entre elas que diferencie os eventos e as informações que devem ser enviadas. Desta forma, mensagens que serão exibidas são atualizadas instantaneamente.

Cada placa é programada para realizar a operação correspondente ao tipo de dado que está sendo transferido, ou seja, é previsto no código o momento que a placa deve realizar o envio de determinado dado ou a forma que proceder quando receber. Para isso utiliza-se um código para

identificar a ocorrência de determinado evento. Além do código, são enviados os seguintes dados:

- Mensagem: contém a informação a ser exibida no visor LCD. São totalmente flexíveis de acordo com os diferentes eventos que podem ocorrer, por exemplo: autenticação do usuário, travamento de um dispositivo, possível falta de comunicação com o gerente, abertura de uma porta de forma assíncrona, exibição de uma mensagem padrão do sistema;
- Data e hora: mantém a interface de interação com usuário atualizada. Estes dados são enviados pelo gerente na inicialização do sistema e atualizados pela placa principal.

### III. RESULTADOS E CONCLUSÕES

Para verificar o funcionamento da arquitetura ESC e analisar o seu desempenho, foram cadastrados alguns usuários na base de dados com diferentes tipos de permissões de acesso. Além disso, foi construído um protótipo com as duas placas que compõem o ESCHA, integrado ao ESCMA e um cenário de testes, simulando o ambiente real de aplicação para esta arquitetura. O cenário construído ilustra o controle de acesso a um local, utilizando uma placa externa (auxiliar) ao ambiente e uma em seu interior (placa principal), conforme pode ser visualizado na Fig. 3. Para realizar a abertura da porta foi utilizada uma relé, responsável pelo acionamento de uma fechadura eletromagnética.

Os primeiros testes realizados serviram para verificar o comportamento do hardware construído e ajustar os códigos desenvolvidos para cada placa. A comunicação entre as placas através da porta serial também foi submetida a vários testes. Durante estas análises, obteve-se um resultado satisfatório, o que permitiu a simulação envolvendo a comunicação via rede com o gerente.

Utilizando o cenário citado anteriormente, foram realizados diversos experimentos para o controle de acesso a um dispositivo (porta). Os testes foram realizados utilizando uma rede local para conectar o ESCHA ao gerente. Um cabo UTP e conectores RJ 45 foram utilizados para conectar a porta serial de uma placa a outra. Uma adaptação na placa, onde os pinos da porta serial foram disponibilizados em um conector RJ 45 femea, permitiu esta forma de conexão.

Os testes foram realizados em três sessões com duração de uma hora cada. Durante cada sessão foram realizadas quarenta tentativas de autenticação de usuário. Os resultados demonstraram que o intervalo de tempo entre processo de autenticação e a resposta do gerente é de aproximadamente 2,2 segundos. Esse tempo é uma média de todas as tentativas realizadas e é independente de qual placa realiza a leitura do cartão, o que confirma que a comunicação serial não causa nenhuma perda de dados ou atraso ao sistema. Testes de estresse foram realizados, mantendo a Arquitetura

ESC em funcionamento por diversas horas. Nesses testes, foram realizadas tentativas em tempos aleatórios e verificou-se que 90% das tentativas de autenticação obtiveram uma resposta do gerente. As tentativas sem respostas representam o tempo que o gerente pode estar em modo espera (após determinado tempo sem receber uma operação de autenticação) e possível falha de leitura em algum cartão.

Além disso, a exibição de mensagens no visor LCD ocorreu de forma síncrona e flexível, possibilitando que as informações fossem atualizadas conforme a necessidade de cada evento, podendo ainda ser atualizadas pelo envio de mensagens pela a interface web do sistema.

### IV. MELHORIAS E PROJETOS FUTUROS

Conforme as constantes mudanças que ocorrem em relação à tecnologia, esta arquitetura precisa ser atualizada ao decorrer do tempo para poder estar de acordo com novos recursos e/ou dispositivos disponíveis.

Visando fazer um melhor aproveitamento dos recursos oferecidos pela comunicação serial e a via rede, trabalha-se com ideia de utilizar tais comunicações para realizar a atualização do software de cada placa por meio do envio dos códigos para todos os dispositivos que integram o sistema.

Além disso, está sendo analisada a ideia de transformar o gerente em um servidor web, que acesse cada dispositivo (cliente) somente quando for solicitado (usuário está interagindo com o sistema).

### REFERÊNCIAS

- [1] VERMESAN, Ovidiu; FRIESS, Peter. Internet of Things: Global Technological and Societal Trends. Ed. Aalborg River Publishers, 2011.
- [2] ATZORI, Luiz; IERA, Antonio; MORABITO, Giacomo. The Internet of Things: A survey. Computer Networks - Volume 54, Issue 15, 28 October 2010.
- [3] SMITH, Ian G. The Internet of Things. New Horizons. 2012
- [4] MCROBERTS, Michael. Arduino básico; [tradução Rafael Zanolli]. - São Paulo: Novatec Editora, 2011.
- [5] ARDUINO. Arduino Uno (Online). Disponível na internet. URL: <http://www.arduino.cc/en/Main/ArduinoBoardUno>, 2013.
- [6] SANTOS, A. Gerenciamento de Identidades. Rio de Janeiro, Brasports, 2007.
- [7] RAGUZZONI, Jeann C. M.; HEINSCH, Lamarck Ribas; RIZZETTI, Tiago Antonio. Uma arquitetura para desenvolvimento de dispositivos de autenticação e acesso a espaços físicos. 2012.
- [8] HEINSCH, Lamarck Ribas; RAGUZZONI, Jeann C. M.; RIZZETTI, Tiago Antonio; PASIN, Marcia. Introduzindo uma arquitetura de Gerenciamento de Segurança física de ambientes baseada em ferramentas livres.

---

# IV

## Sessão 4 - Trabalhos de PG

---



# Correlação de Alertas Utilizando CBR em um Internet Early Warning System

Tarcisio Ceolin Junior, Osmar Marchi dos Santos, Giani Petri, Raul Ceretta Nunes, Luís Alvaro de Lima Silva

Programa de Pós-Graduação em Informática – PPGI

Universidade Federal de Santa Maria – UFSM

{ceolin,osmar,gpetri,ceretta,luisalvaro}@inf.ufsm.br

**Resumo—**Sistemas de Detecção de Instrução (*Intrusion Detection Systems – IDS*) são projetados para monitorar possíveis ataques à infraestruturas da rede através da geração de alertas. Com a crescente quantidade de componentes conectados, os IDS tradicionais não estão sendo suficientes para a efetiva detecção de ataques maliciosos, tanto pelo volume de dados como pela crescente complexidade de novos ataques. Nesse sentido, a construção de uma arquitetura *Internet Early Warning Systems (IEWS)* possibilita detectar precocemente as ameaças, antes de causar qualquer perigo para os recursos da rede. O IEWS funciona como um coletor de diferentes geradores de alertas (possivelmente IDS), centralizando e correlacionando informações afim de gerar uma visão holística da rede. Nesse contexto, o presente trabalho tem como objetivo descrever uma arquitetura para a correlação de alertas gerados por IDS dispersos geograficamente através da utilização da técnica de Raciocínio Baseado em Casos (*Case-Based Reasoning – CBR*). Além da arquitetura, são apresentados resultados sobre um estudo de caso em um ambiente real.

## I. INTRODUÇÃO

A Internet é amplamente utilizada pela sociedade, indo da concretização de negócios até a realização de tarefas pessoais. Junto com a crescente dependência da nossa sociedade sobre recursos de Tecnologia da Informação (TI), as preocupações em relação à segurança estão cada vez mais urgentes, pois todos os dias são descobertas novas vulnerabilidades em sistemas que oferecem algum tipo de serviço na internet.

Segundo [1], é notório um acréscimo substancial no número de ataques cibernéticos ano após ano. Diante deste cenário [2], Sistemas de Detecção de Intrusão (*Intrusion Detection System – IDS*) tradicionais, por trabalhar de forma isolada, estão tornando-se obsoletos. Com a crescente quantidade de dispositivos conectados à rede, e consequentemente, o acréscimo de informações transferidas, os IDS tradicionais não estão sendo suficientes para efetiva detecção de ataques maliciosos.

Para suprir a necessidade de monitorar a Internet perante este novo cenário, uma nova abordagem, a construção de uma arquitetura *Internet Early Warning Systems (IEWS)* é apresentada por diversos pesquisadores [2], [3] e [4]. O objetivo deste sistema proteger as funcionalidades da Internet, detectando precocemente as ameaças, antes de causar qualquer perigo para os recursos da rede.

Em trabalhos anteriores [5] e [6], foi proposta uma modelagem de dados de uma base de conhecimento para IEWS (Knowledge Base Attacks Monitoring – KBAM). O modelo representa os dados de diferentes aspectos da rede com foco em eventos relacionados a detecção de intrusão, tais como:

dados de alertas gerados por sistemas de detecção de intrusão, informações sobre medidas de respostas, estatísticas do tráfego e assinaturas de ataques já conhecidos. O presente trabalho tem como objetivo trabalhar sobre essa base de conhecimento, provendo uma forma de correlação entre alertas através do uso da técnica de Raciocínio Baseado em Casos (*Case-Based Reasoning – CBR*).

Através da técnica de CBR, possibilita-se a criação de um ciclo que analisa informações de casos de alertas passados, automaticamente criando novos casos a partir de informações atuais. Esse trabalho descreve o desenvolvimento dessa arquitetura focando na correlação de alertas encontrados na base. Por ser um estudo preliminar, este trabalho contempla a etapa de recuperação do ciclo CBR inserida no contexto de detecção de intrusão.

O presente artigo é estruturado da seguinte forma. A próxima seção descreve conceitos fundamentais para o desenvolvimento do trabalho. Na seção III é apresentada uma arquitetura para a correlação de alertas utilizando a técnica CBR. Resultados do uso da arquitetura em um ambiente real são apresentados na Seção IV. A Seção V apresenta considerações finais e trabalhos futuros.

## II. CONCEITOS FUNDAMENTAIS

Essa seção descreve uma revisão sobre *Internet Early Warning Systems*, a base de conhecimento KBAM e a técnica de Raciocínio Baseado em Casos.

### A. Internet Early Warning Systems – IEWS

O cenário atual da Internet juntamente com o acréscimo gradativo no número de informações compartilhadas pelas redes de computadores têm motivado a construção de *Internet Early Warning Systems*. Um IEWS trabalha no monitoramento da Internet e tem como objetivo principal a detecção precoce de eventos que ameaçam as funcionalidades da Internet [3]. Além disso, um IEWS visa construir uma consciência situacional e gerar contramedidas para ameaças atuais com base nas informações adquiridas do ambiente monitorado [6].

Segundo [3], um IEWS é composto por diversos componentes técnicos, dentre eles: sensores, componente de detecção, base de conhecimento, componente de reação e gerenciamento de incidentes, componente de perpetuação de evidências e componente de distribuição das informações.

Sensores são utilizados para a geração da visão da atual situação do ambiente monitorado, criando a consciência situacional. Além disso, são responsáveis pela detecção dos eventos

de segurança e identificação de novas ameaças. O componente de detecção é dividido em duas camadas: a camada de sinal, onde os dados da rede ou os logs são analisados por métodos de detecção por anomalia ou assinaturas, e a camada de eventos, na qual ocorre o relacionamento dos eventos da camada de sinal com eventos reportados por órgãos externos [7].

A base de conhecimento é um dos principais componentes de um IEWS, pois armazena informações de diferentes aspectos da rede. As assinaturas de ameaças, o comportamento da rede, as informações sobre os incidentes e suas medidas de respostas estão armazenadas na base de conhecimento e dão suporte à construção da consciência situacional do ambiente monitorado.

#### B. Knowledge Base Attacks Monitoring – KBAM

De modo a trabalhar como um componente técnico em arquiteturas de IEWS, a Base de Conhecimento KBAM [8], [9], [5] representa os dados de eventos de detecção de intrusão explorando o formato *Intrusion Detection Message Exchange Format* (IDMEF) para mensagens de detecção de intrusão e o formato *Intrusion Detection Response Exchange Format* (IDREF) para mensagens de respostas.

Os dados contidos na KBAM consideram os seguintes aspectos: dados de alertas gerados por sistemas de detecção de intrusão, informações sobre as medidas aplicadas em resposta a um alerta e a quantificação do tráfego da rede [8]. Ao modelar os dados com base nos formatos padrões IDREF e IDMEF, a KBAM pode ser inserida em infraestruturas de rede que possuem IDSs que utilizam esses padrões. Neste caso, a KBAM pode ser utilizada como um componente que armazena dados de diferentes aspectos da rede, que são essenciais para o monitoramento de ataques.

Em trabalhos anteriores foi proposto a integração de diferentes IDS utilizando-se um sistema gerenciador de eventos de segurança denominado Prelude [10]. Este sistema gerenciador de eventos é compatível com o formato IDMEF, permitindo que diferentes tipos de sensores criem alertas utilizando um único padrão de comunicação.

#### C. Raciocínio Baseado em Casos – CBR

CBR é uma técnica que busca soluções para problemas atuais em soluções encontradas no passado, baseando-se em uma das principais características do ser humano, a memória. Segundo [11], Sistemas de Raciocínio Baseado em Casos (*Case-Based Reasoning* – CBR) tem como objetivo resolver novos problemas utilizando e adaptando experiências anteriores contidas em um repositório de experiências concretas de soluções de problemas, denominada base de casos. Na forma mais simplificada, um caso é composto por três elementos: uma descrição do problema, uma solução e uma avaliação da solução. Em geral, o ciclo de CBR consiste em quatro etapas: recuperar (*retrieve*), reutilizar (*reuse*), revisar (*revise*) e reter (*retain*).

O ciclo CBR descrito em [12], considerado um formato completo que permite modelar os principais passos de um sistema CBR, é representado por um ciclo de raciocínio que pode ser contínuo. Este ciclo é composto pelas tarefas de

recuperar, reutilizar, revisar e reter um caso. De acordo com o problema informado, ou novo caso usado como consulta no sistema CBR, a base de casos é pesquisada para buscar problemas anteriormente resolvidos. Este processo de busca é realizado de acordo com o nível de similaridade entre atributos do novo problema e da base de casos [11]. Em resumo, a partir da necessidade de resolver determinado problema, esta etapa de recuperação realiza uma busca na base de casos. Como resultado, a etapa de recuperação seleciona quais casos podem conter soluções relevantes (ou reusáveis) para a solução do novo problema, tornando como referência o nível de similaridade entre o problema atual e os casos da base de casos.

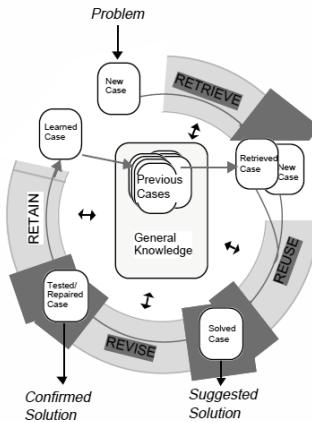


Figura 1. Ciclo CBR [12].

No processo de recuperação de casos, uma métrica de similaridade é uma função que permite avaliar analiticamente os graus de similaridade entre dois casos. Usualmente, são atribuídos pesos diferentes a cada uma das características de um caso. No intuito de combinar as similaridades medidas entre cada um dos atributos representados nos casos, ou similaridades locais, e métodos de agregação como a média ponderada aplicada a valores de similaridades locais são utilizados para gerar um valor global de similaridade entre dois casos. Neste processo, o valor de cada peso é diretamente proporcional à importância de cada atributo definido na estrutura de um caso. A determinação numérica de pesos é geralmente definida como resultado de um processo gradual de ajuste de pesos e consequente avaliação da performance do sistema. Em geral, este processo é caracterizado como um processo de tentativa e erro orientado por resultados de "precision and recall". Este trabalho foca na etapa de recuperação visto que esta etapa é fundamental para a performance de um sistema CBR.

### III. UMA ARQUITETURA DE CORRELACÕES DE ALERTAS UTILIZANDO CBR

Uma das principais características de um sistema IEWS é a capacidade de correlacionar eventos de segurança de forma

que uma atividade maliciosa seja detectada antecipadamente. Nesse sentido, o objetivo deste trabalho é a criação de uma consciência situacional a partir de uma ou diversas redes de computadores, através de agentes distribuídos geograficamente, correlacionando os alertas gerados por estes agentes em um sistema *Internet Early Warning Systems*.

Um sistema de Raciocínio Baseado em Casos tem como objetivo recuperar de sua base os casos mais similares ao problema apresentado. Os casos, que são alertas gerados por IDS distribuídos, não precisam ser necessariamente idênticos a situação atual. Porém, geralmente, quanto maior o nível de similaridade, melhor será a solução encontrada [12]. Um caso recuperado pode ser útil para a solução de determinado problema quando a similaridade entre o problema informado e o caso recuperado da base de casos é alta. Um ou vários casos podem ser recuperados, cabendo ao algoritmo determinar a melhor solução.

Para detectarmos a correlação entre um novo alerta e a extensão da base KBAM, utilizou-se métricas de similaridade local e global entre cada atributo do alerta informado. Similaridade local é utilizada para medir a similaridade entre cada atributo do novo alerta com cada atributo de todos alertas na base de casos (atacantes maliciosos). A partir do resultado do cálculo anterior, realiza-se uma média entre a soma dos resultados obtidos pela quantidade de atributos do alerta, tendo-se assim o cálculo de similaridade global entre dois casos selecionados.

#### A. Correlação de Alertas

A geração de alertas é feita através múltiplas instâncias do IDS Snort [13]. Estes sensores estão distribuídos em pontos distintos dentro da infraestrutura da UFSM e tem seus alertas centralizados diretamente na KBAM.

Inicialmente, para alimentarmos a base de conhecimento KBAM a ser utilizado como parâmetro de comparação, utilizamos todos os alertas gerados por IDS com o tipo de assinatura compatível com varredura de portas (port-scan). Um port-scan tem como objetivo testar as portas lógicas de determinado servidor remoto. Neste teste ele verifica o status das portas, se estão fechadas, escutando ou abertas. Técnicas de port-scan são utilizadas por pessoas mal intencionadas para identificar portas abertas em um computador remoto. Port-scan são facilmente detectados por IDS tradicionais.

Entre todos atributos do formato IDMEF, foram considerados como relevantes na estrutura de um caso, os quais são empregados no cálculo de similaridade: *Analyser ID* (identificador único do sensor), *DetectTime* (Instante de criação do alerta), *Classification* (Classificação do alerta), *Source IP* (Identificação do atacante), *Target IP* (Identificação do alvo), *Source Port* (Porta de origem) e *Target Port* (Porta de destino).

Quando um novo alerta é gerado, o mesmo é correlacionado medindo-se a similaridade entre os atributos do novo alerta e os atributos de cada alerta de toda a base de casos. Assim, ordena-se o resultado encontrado por ordem decrescente por similaridade.

Através de um conjunto de testes randômicos, foram utilizadas diferentes métricas para cálculo de distância ou de similaridade entre os diferentes atributos selecionados. Foram

obtidos melhores resultados utilizando-se o cálculo de distância Euclidiana. Após isso, definiu-se pesos, novamente através de testes, para cada atributo de acordo com a sua relevância para a solução do problema. Por exemplo: o endereço IP do atacante é considerado um atributo de grande importância para a correlação de alertas. Logo, o mesmo recebe um peso 2 pela sua importância. O mesmo critério serve para quantificar o peso do atributo Sensor ID.

Para realizar o cálculo de distância entre atributos que referenciam uma unidade de tempo foi utilizada a seguinte abordagem. Converte-se do formato *timestamp* para *unixtime* e então calcula-se a distância euclidiana entre os valores. A seguir, segundo os testes realizados, percebeu-se que quanto menor a distância entre os atributos, maior era a precisão de acertos nos resultados obtidos. Quanto a classificação do alerta, foi construída uma taxonomia dos diferentes tipos de classificação de ataques e, a partir disto, realizado o cálculo de distância entre este atributo.

Com base em testes realizados, obteve-se melhores resultados (possibilidade de um novo alerta malicioso) quando a similaridade foi superior a 90% (valor de threshold que possibilita obter melhores resultados na performance do sistema), sendo assim, utilizou-se essa métrica em todos experimentos.

#### IV. ESTUDO DE CASO

O ambiente de testes foi utilizado para verificar os resultados da abordagem proposta anteriormente. Conforme descrito na Tabela I, observa-se que a partir da grande quantidade de alertas gerados, foram extraídos cerca de 3000 alertas, estes classificados como *port-scan* e inseridos na base inicial de casos da aplicação CBR. Nesta abordagem, consideramos que todo atacante malicioso, inicialmente, realiza um ataque *port-scan*.

Tabela I. CONTAGEM DE ALERTAS E PORT-SCAN EM DISTINTOS IDS

servidor	alertas	port-scan
coral.ufsm.br	72209	1936
sueci.cpd.ufsm.br	108152	354
coralx.ufsm.br	348263	513
husm.ufsm.br	28500	84
coperves.ufsm.br	72486	71

Um caso pode conter um ou mais atributos, conforme as características do cenário de intrusão ou da atividade suspeita sendo descrita. Um exemplo de port-scan já inserido como caso é apresentado na Tabela II. O caso descreve cada atributo utilizado para calcular a similaridade entre novos casos.

Tabela II. MODELO DE REPRESENTAÇÃO DE UM CASO

Atributo	Caso A
Alert ID	970690
Sensor ID	snort-corala (1)
Detection time	2013-07-12 11:58:47
Source IP Address	113.107.205.57
Source Port	25:80
Target IP Address	200.18.33.52
Target Port	25:80
Service Protocol	TCP
Alert Type	(portscan) TCP Portscan
Classification Type	13

Na Tabela III é descrito o cálculo de similaridade entre um alerta da base de casos e dois novos alertas gerados por

Tabela III. MODELO SIMPLES PARA REPRESENTAÇÃO DO CÁLCULO DE SIMILARIDADE

Atributo	Caso A	Alerta 1	S1	Alerta 2	S2
Sensor ID	snort-corala (1)	snort-sucuri (2)	0	snort-corala (1)	1
Detection UnixTime	1373630327	1378617841	0,2	1373958426	0,6
Source IP	113.107.205.57	218.108.170.169	0	113.107.205.57	1
Source Port	25:80	8080	0	80	1
Target IP	200.18.33.52	200.18.33.57	0	200.18.33.52	1
Target Port	25:80	8080	0	80	1
Service Protocol	TCP	TCP	1	TCP	1
Classification Type	13	9	0	13	1
Alert Type	(portscan) TCP Portscan	WEB-CGI /cgi-bin/ access	*	Cross-Site scripting attempt	*

dois IDS, devidamente identificados no atributo *Sensor ID*. O atributo Alert ID refere-se ao identificador único de cada alerta na base de casos.

No exemplo, observa-se que o mesmo atacante em um primeiro momento, realizou um *port-scan* e foi inserido na base de casos. Em seguida pode-se observar que o mesmo atacante disparou dois novos alertas em servidores distintos. Analisando o Alerta 1, não foi observado praticamente nenhum grau de similaridade (S1) entre os atributos analisados. Já analisando o grau de similaridade (S2) entre os atributos do Alerta 2 e o Caso A, percebe-se que só não ocorreu uma alta similaridade no atributo *Detection Unixtime*.

Com base nessa análise, pode-se observar que o Alerta 1 é um falso-positivo. Já para o Alerta 2, uma resposta no padrão IDREF pode ser gerada e enviada para todos servidores monitorados da rede, com o objetivo de bloquear qualquer tráfego originado do endereço IP deste atacante em questão.

## V. CONCLUSÃO

O conceito de *Internet Early Warning Systems* (IEWS) tem como objetivo possibilitar a detecção precoce de eventos maliciosos sobre a Internet. O presente trabalho apresentou uma forma de correlacionar eventos maliciosos sobre uma base de conhecimento (KBAM), que representa informações de um IEWS, desenvolvida anteriormente em nossos projetos de pesquisa [5], [6], [9]. Nesse trabalho, a correlação de eventos maliciosos foi realizada através da técnica de Raciocínio Baseado em Casos (*Case-Based Reasoning - CBR*), focando-se na fase de recuperação de casos. Uma das grandes dificuldades no desenvolvimento deste trabalho foi o dimensionamento dos pesos dos atributos utilizados nas correlações. Resultados apresentados sobre um estudo de caso real, demonstram a viabilidade da técnica.

Como primeiro trabalho futuro, pretende-se analisar mudanças na arquitetura da base KBAM. Atualmente, um conjunto muito grande de informações é gerado na base. Utilizando a própria solução desenvolvida nesse artigo, a base de conhecimento poderia ser reduzida, eliminando uma grande quantidade de falsos-positivos do sistema.

Outro trabalho futuro importante é o uso de arquiteturas de baixo custo, como Raspberry Pi [14] ou BeagleBoard [15], para o desenvolvimento embarcado de sensores dinâmicos de alertas (*probes*), que serviriam de fonte a base de conhecimento KBAM. Pelo baixo custo destes equipamentos embarcados, seria possível colocar em funcionamento centenas de sensores em uma mesma organização (ou até país), obtendo um ótimo índice de monitoramento da rede. Esse trabalho já está em fase de testes, onde foram desenvolvidos softwares de captura

e envio de alertas diretamente para a KBAM a partir de *probes* de baixo custo.

## REFERÊNCIAS

- [1] CERT.BR, "Centro de Estudos, Resposta e Tratamento de Incidentes no Brasil." 2013. [Online]. Available: <http://www.cert.br/>
- [2] M. Golling and B. Stelle, "Requirements for a future EWS-Cyber Defence in the internet of the future," 2011 3rd International Conference on Cyber Conflict (ICCC), pp. 1–16, 2011. [Online]. Available: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=5954706](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5954706)
- [3] S. Baskic and S. Schmidt, "Internet Early Warning Systems - Overview and Architecture: Objectives of an Internet Early Warning System A definition for early warning in the area of natural catastrophe is following :" pp. 1–19, 2009.
- [4] M. Apel, J. Biskup, U. Flegel, and M. Meier, "Towards Early Warning Systems - Challenges, Technologies and Architecture," *Critical Infrastructure Infrastructures* ..., pp. 151–164, 2010. [Online]. Available: [http://link.springer.com/chapter/10.1007/978-3-642-14379-3\\_13](http://link.springer.com/chapter/10.1007/978-3-642-14379-3_13)
- [5] G. Petri, T. Ceolin Junior, R. C. Nunes, and O. M. Santos, "Modelagem de uma Base de Conhecimento para o Monitoramento de Ataques," *Escola Regional de Redes de Computadores*, 2012.
- [6] G. Petri, R. C. Nunes, V. L. O. Lopez, T. Ceolin Junior, and O. M. Santos, "Building Situation Awareness to Monitor Critical Infrastructures," *Latin-American Symposium on Dependable Computing (LADC)* , 2013. [Online]. Available: <http://www.lbd.dcc.ufmg.br/colecoes/ladc/2013/0019.pdf>
- [7] G. Fan, Y. JiHua, and Y. Min, "Design and implementation of a distributed IDS alert aggregation model," *4th International Conference on Computer Science \& Education, 2009, ICCE '09*, pp. 975–980, 2009. [Online]. Available: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=5228172](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5228172)
- [8] G. Petri, "Modelo de Dados de uma Base de Conhecimento para Internet Early Warning Systems," Master's thesis, Universidade Federal de Santa Maria, 2013.
- [9] G. Petri, R. C. Nunes, V. L. O. Lopez, T. Ceolin Junior, and O. M. Santos, "KBAM: Data Model of a Knowledge Base for Monitoring Attacks," *Latin-American Symposium on Dependable Computing (LADC)* , 2013. [Online]. Available: <http://www.lbd.dcc.ufmg.br/colecoes/ladc/2013/0021.pdf>
- [10] Prelude, "Prelude SIEM." [Online]. Available: <http://www.prelude-ids.com/index.php/uk/>
- [11] A. von Wangenheim and C. G. von Wangenheim, *Raciocínio Baseado em Casos*, 2003.
- [12] A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," *AI communications* , vol. 7, pp. 39 — 59, 1994. [Online]. Available: <http://iospress.metapress.com/index/316258107242IP65.pdf>
- [13] Snort, "Snort." [Online]. Available: <http://www.snort.org/>
- [14] R. Pi, "Raspberry Pi." [Online]. Available: <http://www.raspberrypi.org/>
- [15] BeagleBoard, "BeagleBoard.org." [Online]. Available: <http://beagleboard.org/Products/BeagleBone>

# An Example for Performance Prediction for Map Reduce Applications in Cloud Environments

Iván Carrera, Fabricio Scariot, Cláudio Geyer

Institute of Informatics INF

Federal University of Rio Grande do Sul (UFRGS)

Porto Alegre - RS - Brazil

{ivan.carrera, geyer}@inf.ufrgs.br,  
fabricio.scariot@ufrgs.br

Pierre Turin

Ensimag - Grenoble INP

681 rue de la Passerelle - BP 72

F-38402 St Martin d'Hères Cedex, France

Pierre.Turin@ensimag.grenoble-inp.fr

## Abstract

*One of the advantages of cloud computing is lowering costs to the user by charging only for the computational resources used by the application. In data-intensive applications like Map Reduce, it can be done by using a virtual machine (VM) cluster in the cloud. The goal of this paper is to address the challenge of modelling the behavior of a distributed application running in a Cloud environment and determine the characteristics of the VM cluster that can have the desired performance in the least time. After measuring the time taken by the application and varying parameters of the infrastructure like workload input size and numbers of workers of the cluster, the goal is achieved by finding a model of the execution time which was then applied to predict the execution time for different values of the same variables. Two mathematical models are presented, one for a private cluster and another for a Cloud environment.*

## 1. Introduction

In April 2013, a private company, located in Porto Alegre/Brazil, asked the GPPD/UFRGS laboratory for consultancy to develop and test a Map Reduce application to process large amounts of logs. The logs come from their clients hourly, and have to be processed to feed a Database. The application is meant to be run in a Cloud Computing infrastructure. It is important to know the minimum cluster configuration in terms of quantity of nodes that can meet the demand of this application and the time that will take to accomplish said demand, so the cost of running the application can be minimized. The goal for the consultancy from the GPPD/UFRGS laboratory in this research was to study the behavior of the Map Reduce log processing application as a part of a research that is currently being per-

formed with MapReduce applications over the Cloud. We based our work on a procedure described in [4], which explained a simple but general methodology for modelling the performance of a parallel application running in a Cloud infrastructure, containing 4 steps as follows:

- Defining the parameters that affect the performance of a computer system;
- Perform observations of the application varying the parameters defined in the previous step and taking measures of the performance;
- Propose a mathematical model that relates the performance measures taken with the values of the parameters defined previously, and finally;
- Assess the developed model and test its accuracy.

This methodology will be conducted on several environments to see if it is always accurate. Two environments were chosen: a private cluster and Amazon Web Services' Elastic MapReduce. The AWS platform was chosen by the enterprise, since they are commercial partners.

The remainder of this paper is organized as follows: Section 2 presents the Motivation and background of this work; Section 3 describes how the experiments were conducted and its results; and finally, Section 4 states the conclusions and future work of this research.

## 2. Motivation

### 2.1. Map Reduce

Map Reduce is a programming model used to easily implement distributed programs. It was proposed by Google in [5] and there are several implementations. The Map Reduce (MR) model lets the programmer define his Map and Reduce functions, however, it is restrictive enough so that the application can be automatically parallelized, scheduled

and scaled on different machines, which could be physical or virtual. MR applications have a wide range of domains where they can be useful, and that is the reason for research to be done in order to expand it to add more features and correcting some of its features. The Hadoop [2] implementation of Map Reduce was chosen for this case since it can achieve the goal of the application, its ease of use and because it can be easily expanded to add more features, needed by the application.

The Map and Reduce functions are written by the user. For the Map phase, each machine analyses a part of the input, which is divided in pieces called *chunks*. And for the Reduce phase each machine processes a subset of the intermediate results arranged by key.

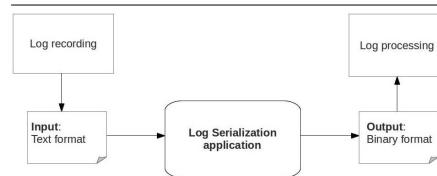
## 2.2. Capacity Planning, Performance evaluation and Performance prediction

Capacity Planning is described in [8] as a process where it can be ensured that adequate computer resources will be available to the users of the system to meet future workload demands, and also [9] refers to Capacity Planning, defining it as properly design and size a computer system for a given load condition. In [1], authors present the importance of having Performance models. The developed Performance Model is an analytical model that capture aspects of the system and relate each one by mathematical formulas and/or computational algorithms.

An interesting work in the topic of Map Reduce and Performance Prediction is [10]. In this paper, authors present a cost function that shows a relationship between some characteristics of the Map Reduce application and the time that takes for the application to execute; and also that they present also cost functions for running MapReduce applications on virtual machines in Cloud environments. In [7], authors develop a way for performing Capacity Planning in Cloud computing environments for Map Reduce applications. Authors deal with the problem of determining the virtual machine cluster resources and the Map Reduce configurations to achieve user-defined requirements on execution time and economic cost for a given workload.

## 2.3. Goal

The goal of this paper is to develop a simple model to predict the performance in terms of execution time of a *log serialization application* running in a private cluster and a cloud computing environment. This model is intended to help the users of the application to perform an accurate capacity planning for the cluster they will use in the cloud computing environment.



**Figure 1. Block diagram of the application**

## 3. Experiments

**3.0.1. Application** The developed Map Reduce application (Fig. 1) receives files containing logs and transforms this input data in a serialized format so the data can then be more easily processed by a Database.

The input data files contain lines of the logs from a CDN server in a text format. Each file contains an hour worth of logs. Later, output files are classified in directories representing the date of the logs. These files contain all the serialized logs for each server, corresponding to the input files.

**3.0.2. Performance evaluation** To make observations to the application and take measures of performance in terms of execution time, two parameters will be taken into account:

- input files sizes ( $W$  - workload) will be tested with values:  $U_W = \{1, 5, 10, 20, 25\}$  GB, and
- number of machines that compose the cluster (*workers*) ( $p$ )
  - for the private cluster, it will be tested with values:  $U_p = \{4, 8, 12, 16\}$ ;
  - for Amazon, it will be tested with values:  $U_p = \{4, 6, 8\}$ .

The goal of these measures is to find a function  $f$  which can gives us an expression of the execution time  $t$  of the application in terms of the number of workers  $p$  and the input size workload  $W$ :  $t = f(W, p)$ . Values for parameters  $W$  and  $p$  were chosen only to have enough data to obtain a linear model of the execution time of the application, since the values of  $W$  for the application in a production environment in the cloud can be among those values.

### 3.1. Experiment description

The set of tested parameters were all the combinations of the elements of  $U_W$  and  $U_p$ .

**3.1.1. Cluster** Every combination of parameters ( $p$  and  $W$ ) was tested 10 times. The Map Reduce application was run on a cluster composed of 18 nodes in total. The cluster is located in INF/UFRGS and accessible at `gradep.inf.ufrgs.br` and each node has an Intel Pentium 4 2.79 GHz CPU and 2 GB in RAM.

The Apache recommendations in [3] were used to select an optimal number of Reduce tasks. The number of reduces ( $N_R$ ) followed this rule:  $N_R = p \times n_{CPU} \times 1.75$ . Where  $p$  is the number of workers,  $n_{CPU}$  is the number of CPU cores in the workers, and 1.75 is a value given by Apache.

**3.1.2. Cloud services** Every combination of parameters ( $p$  and  $W$ ) was tested 2 times. We chose to use AWS' `m1.small` instance type for the master and worker nodes.

## 3.2. Results

Results are shown in Fig. 2, where we can see that for a fixed number of workers, the execution time increases along with the input workload size. The lines in the graphs represent the given model found as explained in section 3.3.

## 3.3. Mathematical model

The results were processed with R [6] to complete a linear regression. We used the model:

$$t_{exec} = \beta_0 + \beta_1 W + \beta_2 p + \beta_3 \cdot \frac{1}{p} + \beta_4 W \cdot p + \beta_5 W \cdot \frac{1}{p}$$

Where  $t_{exec}$  is the execution time we try to predict,  $W$  is the input size in GB,  $p$  is the number of workers and the  $\beta_i$  are the coefficients that the linear regression will calculate.

The linear regression for the cluster measures gave the results:

$$\beta_0 = 465.8, \beta_1 = 16.8, \beta_2 = -11.6$$

$$\beta_3 = -714.9, \beta_4 = 2.1, \beta_5 = 569.8$$

$$R^2 = 0.99$$

And for the Amazon measures, it gave:

$$\beta_0 = -2550.3, \beta_1 = -114.1, \beta_2 = 246.1$$

$$\beta_3 = 8616.4, \beta_4 = 6.2, \beta_5 = 1008.1$$

$$R^2 = 0.95$$

The given coefficients of determination  $R^2$  are high, which gives us a strong confidence in it.

Values	Predicted Value	Exp. Value	Error
$W=1, p=6$	401.6	430.7	7.24%
$W=1, p=10$	373.4	374.8	0.38%
$W=1, p=14$	339.7	387.3	14.0%
$W=5, p=6$	899.3	962.1	6.98%
$W=5, p=10$	752.9	734.7	2.42%
$W=5, p=14$	687.9	653.1	5.05%
$W=10, p=6$	1521.4	1442.0	5.22%
$W=10, p=10$	1227.3	1241.0	1.12%
$W=10, p=14$	1123.0	1250.3	11.33%
$W=20, p=6$	2765.6	2526.2	8.66%
$W=20, p=10$	2176.0	2454.4	12.8%
$W=20, p=14$	1993.3	2021.5	1.41%
$W=25, p=6$	3387.7	3392.4	0.14%
$W=25, p=10$	2560.3	2557.0	3.52%
$W=25, p=14$	2428.5	2296.9	5.42%

**Table 1. Results for model assessing of the cluster measures**

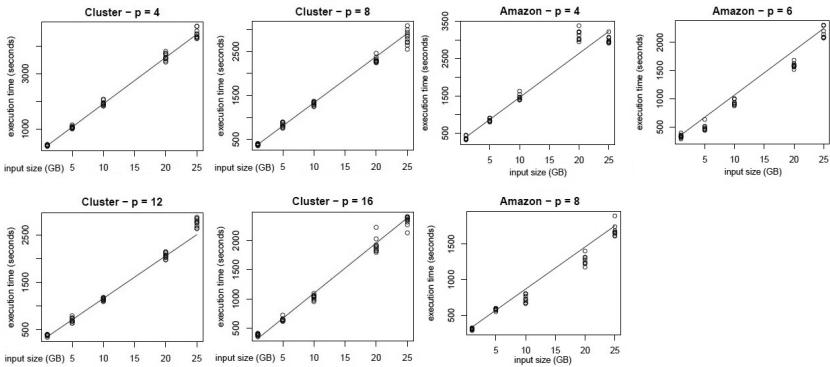
Values	Predicted Value	Exp. Value	Error
$W=1, p=5$	531.0	458.2	13.71%
$W=1, p=7$	488.2	361.5	25.95%
$W=5, p=5$	1061.4	870.2	18.01%
$W=5, p=7$	860.6	780.1	9.26%
$W=10, p=5$	1724.4	1556.9	9.71%
$W=10, p=7$	1326.2	1186.7	10.51%
$W=20, p=5$	3050.3	2826.9	7.33%
$W=20, p=7$	2257.2	2928.5	29.74%
$W=25, p=5$	3713.3	3556.4	4.23%
$W=25, p=7$	2722.7	2414.6	11.32%

**Table 2. Results for model assessing of the Amazon measures**

## 3.4. Assessing the Model

After the mathematical model was proposed, it had to be assessed with experimental results. The same values of  $W$  were used, being  $W=\{1, 5, 10, 20, 25\}$  GB. The new values to assess the model were for  $p$ , being  $p=\{6, 10, 14\}$  nodes for the cluster experiments and  $p=\{5, 7\}$  nodes for the Amazon experiment. Experiments in this part were conducted as explained in section 3.1, and the average values are described in Tables 1 and 2.

From the cluster results, shown in Table 1, we can see that the model was good enough to predict the execution time for the conditions expressed in the Values column, with



**Figure 2. Results of execution time (in seconds) against input size of the workload (in GB) of the experiments, with indicated number of workers running the Map Reduce application**

an error no bigger than 14% of the predicted value. From the Amazon results, shown in Table 2, we can see that most errors are low except the two of them that range near from 35%.

With these results we say that the models are accurate and valid for execution time predictions with this application.

#### 4. Conclusions and Future Work

Based on the results of section 3.4, we see that a mathematical model can be obtained and applied to predict the execution time of a Map Reduce application on different executing environments.

Using the methodology described in [4] helped to model the behaviour of a parallel application in a private cluster environment. So, a future work will be to research about the variance of the results, in a field known as *Performance Debugging*.

#### 5. ACKNOWLEDGEMENTS

This work was made with the support of the *Programa Estudantes-Convênio de Pós-Graduação PEC-PG, of CAPES/CNPq - Brazil*.

#### References

- [1] V. A. Almeida and D. A. Menascé. Capacity planning an essential tool for managing web services. *IT professional*, 4(4):33–38, 2002.
- [2] Apache. 2013. Welcome to Apache Hadoop! <http://hadoop.apache.org/> accessed on 24/07/2013.
- [3] Apache. 2013. HowManyMapsAndReduces - Hadoop Wiki <http://wiki.apache.org/hadoop/HowManyMapsAndReduces> accessed on 24/07/2013.
- [4] I. Carrera and C. Geyer. Impressionism in cloud computing – a position paper on capacity planning in cloud computing environments. In *Proceedings of the 15th International Conference on Enterprise Information Systems ICEIS - Vol. 2*, pages 333 – 338. INSTICC, 2013.
- [5] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.
- [6] R. Foundation. 2013. The R Project for Statistical Computing <http://www.r-project.org/> accessed on 24/07/2013.
- [7] H. Herodotou, F. Dong, and S. Babu. No one (cluster) size fits all: automatic cluster sizing for data-intensive analytics. In *Proceedings of the 2nd ACM Symposium on Cloud Computing*, page 18. ACM, 2011.
- [8] R. Jain. *The art of computer systems performance analysis*, volume 182. John Wiley & Sons Chichester, 1991.
- [9] D. A. Menasc  , V. A. Almeida, L. W. Dowdy, and L. Dowdy. *Performance by design: computer capacity planning by example*. Prentice Hall Professional, 2004.
- [10] F. Tian and K. Chen. Towards optimal resource provisioning for running mapreduce programs in public clouds. In *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, pages 155–162. IEEE, 2011.

# Avaliação do suporte à simulação de redes OpenFlow no NS-3

Marcelo Conterato  
PPGCC, PUCRS

Porto Alegre – Brasil  
marcelo.conterato@acad.pucrs.br

Israel de Oliveira  
PPGCC, PUCRS

Porto Alegre – Brasil  
israel.oliveira@acad.pucrs.br

Tiago Ferreto  
PPGCC, PUCRS

Porto Alegre – Brasil  
tiago.ferreto@pucrs.br

César A. F. De Rose  
PPGCC, PUCRS

Porto Alegre – Brasil  
cesar.derose@pucrs.br

**Resumo**—A avaliação de ambientes simulados usando o protocolo *Openflow* está se tornando uma tendência, em virtude do aumento da demanda por soluções para redes definidas por software (SDN). O protocolo *OpenFlow* pode ser utilizado em diversos cenários como: segurança, roteamento, tolerância a falhas, desempenho,平衡amento de carga, entre outros. Por ser um protocolo experimental, existe uma demanda por ferramentas para avaliação do protocolo nestes cenários. Este artigo tem como objetivo discutir as vantagens e desvantagens do uso do simulador de redes NS-3 em conjunto com o protocolo *OpenFlow*. O artigo faz um levantamento das características das opções disponíveis em ambientes reais, emulados e simulados, levando em consideração o suporte parcial ao protocolo *OpenFlow* pelo NS-3.

## I. INTRODUÇÃO

As redes definidas por software (SDN – *Software Defined Networks*) [8] são caracterizadas pelo uso de um controlador para programar o funcionamento da rede de acordo com diferentes necessidades e propósitos, tendo por objetivo facilitar inovações estruturais na rede. As redes SDN fornecem abstrações basicamente em três áreas da rede: distribuição do fluxo de rede, encaminhamento e configuração. Além disso, as redes SDN facilitam a criação de políticas de encaminhamento de pacotes através do uso de fluxos, permitindo o uso de caminhos com características específicas, como por exemplo: maior largura de banda, menor latência ou menor número de saltos, contribuindo ainda, para uma redução no consumo de energia.

Para a implementação das redes SDN, dois requisitos devem ser plenamente atendidos: (i) deve existir uma arquitetura comum em todos os equipamentos de rede envolvidos na comunicação (e.g., *switches* e roteadores) ou qualquer outro dispositivo gerenciado por um controlador SDN e, (ii) deve existir um protocolo padronizado seguro para a comunicação entre o controlador SDN e os dispositivos de rede. Ambos os requisitos são satisfeitos pelo protocolo *OpenFlow* que tem como objetivos configurar os fluxos nos dispositivos de rede e gerenciar toda a infraestrutura definindo políticas de gestão do tráfego de rede.

Permitir que os usuários definam fluxos e determinarem qual o caminho que esses fluxos devem tomar através da rede, sem interromper o tráfego normal, é um dos objetivos das pesquisas em Internet do Futuro. O termo Internet do Futuro é relativo a uma ampla iniciativa mundial para identificar os rumos tecnológicos que a rede deverá tomar nos próximos anos.

Atualmente, a demanda por ferramentas para avaliação de cenários de redes vem crescendo em virtude da necessidade de testar as soluções para redes antes mesmo da sua utilização no mundo real. Alguns fatores como: investimento, relação custo-benefício, complexidade de gerenciamento e tempo necessário para implementação são algumas das preocupações associadas aos ambientes de avaliação das redes atuais e das tecnologias de redes que estão surgindo. Por este motivo, o estudo dos ambientes possíveis para a execução de experimentos se torna importante no contexto atual das redes. A avaliação do funcionamento de cada ambiente, seja através de experimentos em um ambiente real, emulado ou simulado pode trazer benefícios para os estudos de novas tecnologias SDN.

O NS (*Network Simulator*) é um simulador de redes bastante conhecido e popular na pesquisa de redes de computadores. O NS permite avaliar o funcionamento da rede em termos de tráfego e desempenho. A terceira versão do simulador (NS-3) possui suporte à simulação de redes *OpenFlow*.

Este artigo tem como objetivo avaliar as vantagens e desvantagens do uso do simulador de redes NS-3 em conjunto com o protocolo *OpenFlow*. O artigo apresenta também uma comparação das opções disponíveis para avaliação de redes *OpenFlow* em ambientes reais, emulados e simulados.

O artigo está organizado da seguinte forma: na Seção 2 é apresentado o funcionamento do protocolo *OpenFlow* e o levantamento das opções para avaliação do protocolo *OpenFlow* usando ambientes reais, emulados e simulados; a Seção 3 apresenta uma análise sobre a simulação do protocolo *OpenFlow* com o simulador NS-3; a Seção 4 apresenta uma comparação entre as estratégias de avaliação do protocolo *OpenFlow* usando ambientes reais, emulados ou simulados, e as conclusões do artigo são apresentadas na Seção 5.

## II. SUPORTE PARA EXPERIMENTAÇÃO DO PROTOCOLO OPENFLOW

O protocolo *OpenFlow* [9] foi proposto pela Universidade de Stanford em 2008 e atualmente está na versão 1.3.1 [13]. A arquitetura de uma rede que utiliza o protocolo *OpenFlow* é composta por *switches* e controladores. O objetivo do controlador é, através de um canal seguro, modificar a tabela de fluxos de um *switch OpenFlow*. Quando o controlador recebe uma mensagem de um *switch*, ele examina o cabeçalho desta mensagem e verifica se um novo fluxo precisa ser criado ou qual ação precisa ser realizada. Se uma nova entrada precisa ser criada, o controlador envia uma mensagem ao *switch* para

instalar um novo fluxo. O controlador tem a capacidade de adicionar, atualizar e remover fluxos da tabela dos *switches* OpenFlow.

A Figura 1 apresenta uma arquitetura de rede usando o protocolo OpenFlow.

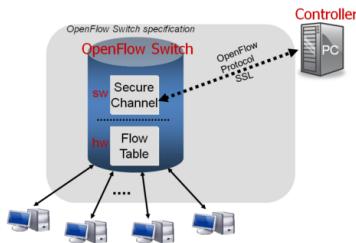


Figura 1. Arquitetura de rede usando o protocolo OpenFlow [9]

Em equipamentos de rede como roteadores e *switches* clássicos, o rápido encaminhamento de pacotes (plano de dados) e as decisões de encaminhamento (plano de controle) ocorrem em um mesmo dispositivo. Um *switch* OpenFlow tem a finalidade de separar estas duas funções. A função de encaminhamento dos dados permanece sendo função do *switch*, enquanto que, as decisões de encaminhamento de alto nível são funções agora de um controlador remoto, normalmente localizado em uma máquina servidora. O controlador e o *switch* OpenFlow realizam a comunicação através do protocolo OpenFlow, que define as mensagens, tais como: pacotes recebidos, encaminhamento dos pacotes de saída, modificação da tabela de encaminhamento, etc. A Figura 2 detalha o protocolo OpenFlow, apresentando a separação do plano de dados e do plano de controle.

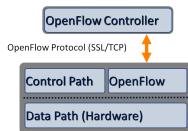


Figura 2. Separação dos planos de dados e controle no OpenFlow [10]

O OpenFlow permite o controle dos fluxos de dados, escolhendo o caminho que cada pacote segue e o processoamento que irá receber. Desta forma, o OpenFlow permite experimentar novos protocolos de roteamento, mecanismos de segurança ou modelos de endereçamento na rede.

Atualmente, o protocolo OpenFlow pode ser avaliado usando um ambiente real, emulado ou simulado.

#### A. Ambientes Reais

Vários fabricantes já disponibilizam *switches* com suporte ao protocolo OpenFlow para criação de ambientes e realização de testes. Atualmente, também é possível alterar o *firmware*

original de alguns *switches* por um *firmware open source* chamado Indigo [6]. O Indigo foi originalmente desenvolvido pela Universidade de Stanford e é uma implementação que suporta a especificação OpenFlow 1.0. A solução suporta até 48 portas 10-gigabit ethernet.

O projeto NetFPGA<sup>1</sup> (*Network Field Programmable Gate Array*) permite que pesquisadores, professores e estudantes realizem a prototipagem de dispositivos de rede (e.g. switches e roteadores) usando *hardware* programável. Em virtude de ter estas características, ele é amplamente utilizado nas pesquisas em Internet do Futuro com redes SDN. Por ser uma plataforma aberta, pesquisadores têm utilizado o NetFPGA para construir sistemas de rede avançadas para processamento de fluxos. O sistema consiste em uma placa programável que pode rotear pacotes de quatro sub-redes. Várias placas NetFPGAs podem ser instaladas em uma mesma máquina. Atualmente, existem duas plataformas: NetFPGA-1G (1G) que fornece 4 portas gigabit ethernet e a NetFPGA-10G (10G) que fornece 4 portas 10-gigabit ethernet.

Podemos citar ainda, como solução para uso em ambientes reais, a utilização do OpenWRT [2]. Esta solução consiste em um sistema operacional Linux utilizado principalmente para dispositivos embarcados, onde é possível personalizar o *firmware* de alguns modelos específicos de dispositivos, habilitando o suporte a utilização do protocolo OpenFlow. Atualmente o OpenWRT encontra-se na versão 12.09 [14].

#### B. Ambientes Emulados

O MiniNet<sup>2</sup> é uma ferramenta para a simulação de redes definidas por software que permite a emulação de uma grande infraestrutura virtual de rede com a utilização de apenas uma máquina. O Mininet possibilita a criação de ambientes escaláveis usando redes virtuais utilizando primitivas de virtualização do sistema operacional. Com essas primitivas, o usuário pode rapidamente criar, interagir, personalizar e compartilhar um protótipo de rede definida por software (SDN) para simular uma topologia de rede que utiliza *switches* OpenFlow.

O MiniNet permite desenvolver topologias personalizadas utilizando *scripts* em Python, com a possibilidade de interação com a rede física existente utilizando o Mininet CLI (juntamente com a API disponível), sendo ainda, possível a sua implantação em *hardware* real. O MiniNet utiliza controladores reais, como NOX [3], POX<sup>3</sup> e o FloodLight<sup>4</sup>, para o gerenciamento das tabelas de fluxos.

#### C. Ambientes Simulados

Atualmente, a simulação tem um papel decisivo no projeto, análise e implementação de sistemas de comunicação, principalmente quando estes sistemas são caros e complexos. A simulação de um sistema real pode ser definida como o processo de avaliação numérica de um modelo de simulação, que deve representar o mais fielmente possível o sistema real a ser simulado. As informações resultantes deste processo são utilizadas para estimar variáveis de interesse deste sistema.

<sup>1</sup><http://www.netfpga.org/>

<sup>2</sup><http://mininet.org/>

<sup>3</sup><http://www.noxrepo.org/pox/about-pox/>

<sup>4</sup><http://floodlight.openflowhub.org/>

A utilização de ambientes de simulação vem aumentando de forma significativa uma vez que estes permitem o estudo e a avaliação de sistemas a custos reduzidos. Os simuladores de rede desempenham um papel importante na tarefa de desenvolver, analisar e aperfeiçoar protocolos de comunicação. Destacam-se três importantes vantagens do uso de simulação [4]:

- 1) Permitem testar o comportamento dos protocolos em diversas redes e ambientes, cuja preparação num laboratório ou em uma empresa poderia ser impraticável, isso por questão de custos, ou em tempo de instalação, ou mesmo do ponto de vista administrativo;
- 2) Facilitam a execução de testes em um ambiente controlado, onde é mais fácil fazer variar parâmetros relevantes, mantendo os restantes parâmetros constantes;
- 3) Facilitam a execução dos protocolos em múltiplos cenários de execução.

Segundo [5], alguns dos principais motivos pelos quais é recomendável que os ambientes de redes sejam previamente simulados antes da sua instalação são:

- A experimentação no mundo real pode causar alguma espécie de dano ao ambiente;
- A modelagem e análise requerem altos níveis de abstração e possíveis falhas podem surgir na especificação de detalhes;
- Alterações em redes existentes podem requerer uma carga de trabalho expressivo;
- Falhas na configuração de determinadas tecnologias podem gerar um custo elevado.

Existem diversos simuladores de rede disponíveis, porém poucos são os simuladores que oferecem suporte ao protocolo *OpenFlow*. Um destes simuladores é o *Network Simulator 3* (NS-3). A Seção 3 apresenta em detalhes o suporte à simulação do protocolo *OpenFlow* no NS-3.

### III. SIMULAÇÃO DO PROTOCOLO OPENFLOW NO NS-3

O NS-3 é um simulador de rede de eventos discretos utilizado principalmente por pesquisadores, principalmente por possuir distribuição gratuita e código aberto [12]. Tal fato o torna adequado a situações em que é necessário desenvolver novas funcionalidades, como em teses e projetos de pesquisa aplicada. A execução de experimentos de simulação no NS-3 exige a implementação de códigos em C++ ou Python.

O NS-3 atualmente está na versão 3.18, tendo seu desenvolvimento sido iniciado em julho de 2008. Ao contrário do que sugere a versão do simulador, o NS-3 é um simulador totalmente novo, e não uma extensão da versão anterior do simulador. O NS-3 não suporta as APIs do NS-2, porém algumas funcionalidades do NS-2 foram portadas para o NS-3.

O simulador de redes NS-3 é considerado uma das ferramentas de simulação que apresenta melhor desempenho [16]. O NS-3 tem sua arquitetura baseada na técnica de rastreamento [1]. O rastreamento facilita a descoberta de eventos significativos na simulação e permite que o pesquisador obtenha métricas

importantes de uma simulação que podem ser utilizadas para comparação entre diferentes cenários.

Atualmente, a implementação do protocolo *OpenFlow* existente no NS-3 disponibiliza dois controladores, que executam funções básicas, permitindo apenas que determinados fluxos de dados ou sejam descartados pelo *switch*, ou ainda, que seja realizado o processo de aprendizagem tradicional dos switches, onde é cada máquina detectada é mapeada para uma porta específica do switch. Os controladores implementados no NS-3 são:

- *DropController*: Quando um *switch* encaminha para o controlador um pacote que ele não sabe o que fazer, o controlador retorna ao *switch* a regra para descartar os pacotes;
- *LearningController*: Quando um *switch* encaminha para o controlador um pacote que ele não sabe o que fazer, o controlador verifica em uma tabela interna se existe alguma porta específica mapeada para o destinatário do pacote. Se existir, ele configura o *switch* para enviar o pacote para a porta específica, caso contrário, ele indica para o *switch* enviar o pacote para todas suas portas (*flooding*). Logo após, ele obtém a identificação da máquina e porta origem do pacote e envia uma regra ao *switch* mapeando o endereço da máquina para a porta específica.

Por vários aspectos o NS-3 não simula de forma fidedigna uma rede *OpenFlow* real. O módulo *OpenFlow* apresenta as seguintes limitações:

- não tem suporte ao tráfego TCP entre o *switch* e o controlador;
- não tem suporte ao protocolo *Spanning Tree* e *Multi Protocol Label Switching* (MPLS);
- não possibilita a utilização de controladores externos;
- a implementação do *switch* não suporta remontagem do pacote IP.

O NS-3 possibilita a captura do tráfego gerado durante a simulação para fins de análise dos dados. O NS-3 utiliza o formato pcap para arquivos de captura, utilizado também pela ferramenta TCPDump [7]. A ferramenta Wireshark [15] também pode ser utilizada para visualização dos fluxos, exigindo previamente a configuração do plugin *OpenFlow Wireshark Dissector*<sup>5</sup>. Através da ferramenta NetAnim [11] é possível visualizar graficamente simulações armazenadas em arquivos de rastreamento especiais armazenados em formato XML.

### IV. COMPARAÇÃO ENTRE AS SOLUÇÕES PARA AVALIAÇÃO DO PROTOCOLO OPENFLOW

Esta seção apresenta uma comparação das principais soluções para avaliação do protocolo *OpenFlow* usando ambientes reais, emulados e simulados. A Tabela I resume as principais funções dos ambientes vistos anteriormente.

Um dos fatores que mais chamam a atenção é a escalabilidade, onde o Mininet foi classificado como de escalabilidade

<sup>5</sup><http://goo.gl/lLrk5>

Tabela I. RECURSOS DISPONÍVEIS

	Switches OF / NetFPGA	Mininet	NS-3
Especificação OpenFlow	1.3.1	1.3	0.8.9
Tipo de Ambiente	Real	Emulação	Simulação
Controlador no mundo real	X	X	-
Escalabilidade	Alta	Média	Alta
Suporte GUI	Configuração Observação	Observação Config C++	Observação Config Phyton

média em virtude de necessitar executar um processo *shell* (por exemplo */bin/bash*) para conseguir emular cada nodo, sendo ainda necessário, executar um processo para cada *vSwitch* no espaço do usuário para simular cada *switch OpenFlow*. Como resultado, o Mininet é menos escalável que NS-3, pois em virtude do nodos, dos switches OpenFlow e dos controladores serem todos simulados no NS-3, eles estão todos implementados como módulos C++ e conectados entre si como um único processo. Os ambientes reais também são classificados como de escalabilidade alta, pois sua única limitação é o custo de aquisição dos equipamentos.

O MiniNet até o momento ainda não fornece desempenho e qualidade fiéis a uma rede real, embora o código utilizado nele sirva para uma rede real baseada em placas NetFPGAs, ou switches comerciais. Isso se deve aos recursos que são tratados pelo *kernel* da máquina simuladora em tempo real, uma vez que a largura de banda total é limitada por restrições de CPU e memória da mesma.

Já o NS-3 apresenta uma limitação relevante que é a inexistência de uma comunicação TCP entre o switch OpenFlow e o controlador. Atualmente somente a comunicação UDP é suportada. Em cenários reais, a falta de controle do congestionamento que é realizado pelo protocolo TCP pode afetar o desempenho da rede ocasionando perda de pacotes, afetando assim a comunicação entre o controlador e o switch. Outra limitação existente no NS-3 é a manipulação da tabela de fluxos pelos usuários finais. Os recursos de adição e modificação de fluxos estão disponíveis no NS-3, porém sua utilização requer o conhecimento da implementação do protocolo OpenFlow através da manipulação da estrutura *ofp\_flow\_mod* onde estão localizados os parâmetros responsáveis pela tomada de decisão dos fluxos.

## V. CONCLUSÃO

Neste artigo foi apresentado um levantamento das vantagens e desvantagens do uso do simulador de redes NS-3 para estudo do protocolo OpenFlow. Foi realizado um levantamento das opções disponíveis em ambientes reais, emulados e simulados. O uso de simuladores, como o NS-3, torna-se mais interessante por sua flexibilidade quanto a programação, permitindo ao usuário a criação de seu próprio controlador OpenFlow e de uma rotina de testes, agilizando a execução das simulações.

Devido a dificuldade de demonstrar experimentalmente seu comportamento em condições próximas das reais, essas novas tecnologias são de alto custo, tornando as soluções de emulação e simulação atrativas. Dessa forma, a simulação do protocolo OpenFlow no NS-3 ainda necessita evoluir em vários aspectos, onde destacamos a conexão com controladores OpenFlow externos, não permitindo assim, quantificar os efeitos de

simular múltiplos switches conectados a um único controlador OpenFlow.

## REFERÊNCIAS

- [1] Gustavo Carneiro, Pedro Fortuna, and Manuel Ricardo. Flowmonitor: a network monitoring framework for the network simulator 3 (ns-3). In *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, page 1. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.
- [2] Florian Fainelli. The openwrt embedded development framework. In *Proceedings of the Free and Open Source Software Developers European Meeting*, 2008.
- [3] Natasha Gude, Teemu Koponen, Justin Pettit, Ben Pfaff, Martín Casado, Nick McKeown, and Scott Shenker. Nox: towards an operating system for networks. *ACM SIGCOMM Computer Communication Review*, 38(3):105–110, 2008.
- [4] Conceição V. Nuno C. Rodrigues L. Guedes, S. Plataforma de desenvolvimento e simulação de protocolos. 2005.
- [5] J. Hughes. Network simulation introduction. Website OpenXtra, 2009. <http://www.openxtra.co.uk/articles/network-simulation>.
- [6] Indigo. Indigo project. Website, 2013. <http://www.openflowhub.org/display/Indigo/Indigo++Open+Source+OpenFlow+Switches++First+Generation>.
- [7] Van Jacobson, Craig Leres, and S McCanne. The tcpdump manual page. *Lawrence Berkeley Laboratory, Berkeley, CA*, 1989.
- [8] Bob Lantz, Brandon Heller, and Nick McKeown. A network in a laptop: rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, page 19. ACM, 2010.
- [9] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.
- [10] Nick McKeown and Guru Parulkar. State of openflow and sdn, 2010.
- [11] Development NS-3-Team. Netanim - offline animator. Website, 2013. <http://www.nsnam.org/wiki/index.php/NetAnim>.
- [12] Development NS-3-Team. Ns-3 tutorial. Website, 2013. <http://www.nsnam.org>.
- [13] OpenFlow. Openflow switch specification. Website, 2011. <http://www.openflow.org/documents/openflow-wp-latest.pdf>.
- [14] OpenWRT. Openwrt project. Website, 2013. <https://openwrt.org/>.
- [15] Angela Orebaugh, Gilbert Ramirez, and Jay Beale. *Wireshark & Ethereal network protocol analyzer toolkit*. Syngress, 2006.
- [16] Elias Weingartner, Hendrik Von Lehn, and Klaus Wehrle. A performance comparison of recent network simulators. In *Communications, 2009. ICC'09. IEEE International Conference on*, pages 1–5. IEEE, 2009.