SimCIT 2016
Simpósio de Ciência, Inovação e Tecnologia

ANAIS

Artigos Completos e Resumos

Organizadores
Cristian Cleder Machado
Guilherme Bontorin

SimCIT

Simpósio de Ciência, Inovação e Tecnologia

ANAIS

Universidade Regional Integrada do Alto Uruguai e das Missões

REITOR
Luiz Mario Silveira Spinelli
PRÓ-REITOR DE ENSINO
Arnaldo Nogaro
PRÓ-REITOR DE PESQUISA, EXTENSÃO E PÓS-GRADUAÇÃO
Giovani Palma Bastos
PRÓ-REITOR DE ADMINISTRAÇÃO
Nestor Henrique de Cesaro

CAMPUS DE FREDERICO WESTPHALEN
Diretora Geral
Silvia Regina Canan
Diretora Acadêmica
Elisabete Cerutti
Diretor Administrativo
Clóvis Quadros Hempel

CAMPUS DE ERECHIM
Diretor Geral
Paulo José Sponchiado
Diretora Acadêmica
Elisabete Maria Zanin
Diretor Administrativo
Paulo Roberto Giollo

CAMPUS DE SANTO ÂNGELO
Diretor Geral
Gilberto Pacheco
Diretor Acadêmico
Marcelo Paulo Stracke
Diretora Administrativa
Berenice Beatriz Rossner Wbatuba

CAMPUS DE SANTIAGO
Diretor Geral
Francisco de Assis Górski
Diretora Acadêmica
Michele Noal Beltrão
Diretor Administrativo
Jorge Padilha Santos

CAMPUS DE SÃO LUIZ GONZAGA
Diretora Geral
Dinara Bortoli Tomasi

CAMPUS DE CERRO LARGO
Diretor Geral
Edson Bolzan

Coordenação Geral
Cristian Cleder Machado
Guilherme Bontorin

Comitê de Organização de Palestras, Minicursos e Oficinas
Marcos Antonio Ritterbuch
Michele Cadore Kern

Comitê de Organização de Pôsteres e Plenárias
Cristian Cleder Machado
Maurício Sulzbach
Michele Cadore Kern

Comitê de Patrocínios
André Luís Stefanello
Leandro Rosniak Tibola
Thiago Roberto Sarturi

Comitê de Divulgação
Catiane Priscila Mazzutti
Cliceres Mack Dal Bianco
Giancarlo Cerutti Panosso

Comitê Técnico do Programa
Adalto Selau Sparremberger
Amyr Borges Fortes Neto
André Luís Stefanello
Andressa Falcade
Bruno Batista Boniati
Camila Cerezer Possobom
Carlos Oberdan Rolim
Carla Lisiane de Oliveira Castanho
Catiane Priscila Mazzutti
Claudio Jose Biazus
Cliceres Mack Dal Bianco
Cristian Cleder Machado
Delfa Huatuco Zuasnábar
Eduardo Ferreira Silva
Eduardo Germano da Silva
Evandro Dalla Vecchia Pereira
Evandro Preuss
Fabrício Herpich
Fahad Kalil
Felipe Becker Nunes
Fernando Pinho Marson
Francisco Duarte Pavin
Giancarlo Cerutti Panosso
Giani Petri
Gleizer Bierhalz Voss
Guilherme Bontorin
Josieli Piovesan
Juliano Araujo Wickboldt
Kassiano José Matteussi
Leandro Lorenzett Dihl
Leandro Rosniak Tibola
Lenin Ernesto Abadie Otero
Marcelo Caggiani Luizelli
Marcos Antonio Moretto
Marcos Antonio Ritterbuch
Marilei Kovatli
Marlon Fernandes Alcantara
Mauricio Barros
Maurício Sulzbach
Michele Cadore Kern
Neilor Avelino Tonin
Oscar Caicedo
Pedro Heleno Isolani
Rafael Padilla
Rafael Pereira Esteves
Rodolfo Favaretto
Rômulo Reis de Oliveira
Solange Solange de L. Pertile
Thiago Roberto Sarturi
Victor Machado Alves
Vinicius Jurinic Cassol

UNIVERSIDADE REGIONAL INTEGRADA DO ALTO URUGUAI E DAS MISSÕES
CAMPUS DE FREDERICO WESTPHALEN
DEPARTAMENTO DE ENGENHARIAS E CIÊNCIA DA COMPUTAÇÃO
CURSO DE CIÊNCIA DA COMPUTAÇÃO

SimCIT

Simpósio de Ciência, Inovação e Tecnologia

ANAIS

Organizadores
Cristian Cleder Machado
Guilherme Bontorin

Frederico Westphalen
2017

Organização: Cristian Cleder Machado e Guilherme Bontorin
Revisão metodológica: Elisângela Bertolotti
Diagramação: Elisângela Bertolotti
Capa/Arte: Cristian Cleder Machado
Revisão Linguística: Wilson Cadoná

O conteúdo de cada resumo bem como sua redação formal são de responsabilidade exclusiva dos (as) autores (as).

Catalogação na Fonte elaborada pela
Biblioteca Central URI/FW

S621a Simpósio de Ciência, Inovação e Tecnologia - SimCIT (1.: 2017 : Frederico Westphalen, RS)

Anais [do] I Simpósio de Ciência, Inovação e Tecnologia - SimCIT [recurso eletrônico] / Organizadores: Cristian Cleder Machado e Guilherme Bontorin . – Frederico Westphalen : URI – Frederico Westph, 2017.
40 p.

Disponível em: <www.fw.uri.br/site/publicacoes>
ISBN: 978-85-7796-213-6

1. Ciência da computação. 2. Mundos virtuais. 3. Linguagem Python. 4. Práticas com o Raspberry Pi e C# com Visual Studio. I. Universidade Regional Integrada do Alto Uruguai e das Missões – Curso de Graduação em Ciência da Computação. II. Machado, Cristian Cleder, Bontorin, Guilherme, org. III. Título.

CDU 004

Bibliotecária: Gabriela de Oliveira Vieira

URI - Universidade Regional Integrada do Alto Uruguai e das Missões
Prédio 9
Campus de Frederico Westphalen
Rua Assis Brasil, 709 - CEP 98400-000
Tel.: 55 3744 9223 - Fax: 55 3744-9265
E-mail: editora@uri.edu.br
Impresso no Brasil
Printed in Brazil

SUMÁRIO

APRESENTAÇÃO

A primeira edição do Simpósio de Ciência, Inovação e Tecnologia (SimCIT), promovido e mantido pelo Curso de Ciência da Computação e Programas de Pós-graduação nas áreas de Ciência da Computação da URI – Câmpus de Frederico Westphalen, destinou-se a pesquisadores, professores, alunos de graduação e pós-graduação e demais profissionais das áreas de Ciência da Computação e afins, tendo como objetivo publicar artigos completos e resumos estendidos que contribuam significativamente para o conhecimento dessas áreas.

Em sua programação de atividades, o evento contou com a realização de palestras seguidas de arguições com o público presente, formado especialmente por alunos e profissionais de áreas afins, juntamente com professores. A fim de atender necessidades da comunidade acadêmica e público em geral, o evento ofereceu uma noite de cursos/oficinas nos temas Mundos Virtuais, linguagem Python, Práticas com o Raspberry Pi e C# com Visual Studio.

O Simpósio contou com a apresentação de seminários de andamento de trabalhos de conclusão de curso, onde os alunos graduandos em ciência da computação fizeram a exposição de fragmentos de seus trabalhos, afim de estimular os participantes do evento a conhecerem ou se envolverem com a área. Por fim, também foram apresentados trabalhos em formato artigo, que compilaram diversos temas tais como Algoritmos e Otimização, Inteligência Artificial, Jogos Digitais, Tecnologias Emergentes, Tolerância a Falhas, Interação Humano-Computador, Processamento de Imagem, Segurança da Informação, Internet das Coisas ou Computação Ubíqua, Aplicações Móveis, Modelagem, Simulação e Arquiteturas, Sistemas Operacionais e Embarcados, e Sistemas Paralelos e Distribuídos. Os trabalhos completos e resumos estão presentes nestes anais.

Cristian Cleder Machado

TOOLFORME: UM TOOLKIT PARA ANÁLISE FORENSE DE MEMÓRIA

TOOLFORME: A MEMORY FORENSICS TOOLKIT

VANESSA DA SILVA FERREIRA[1]*, CRISTIAN CLEDER MACHADO[1]

[1] Departamento de Engenharias e Ciência da Computação, Universidade Regional Integrada do Alto Uruguai e das Missões. URI – Câmpus de Frederico Westphalen – RS;
*E-mail: vanessaferreira1492@gmail.com.

Resumo: O presente artigo tem como objetivo apresentar o desenvolvimento de um toolkit para realização de análise forense de memória chamado ToolForMe. ToolForMe é capaz de realizar análises forenses em dumps de memória gerados a partir de computadores com o Sistema Linux. ToolForMe disponibiliza diversos plugins para análise. Isto faz com que o usuário tenha um vasto conjunto de funções disponíveis, as quais são utilizadas para encontrar vestígios e levantar evidências presentes na memória sobre incidentes ocorridos nos computadores/dispositivos. Para demonstrar o funcionamento do toolkit foi realizado um estudo de caso onde todos os plugins disponíveis foram utilizados. Os resultados mostram que a ferramenta retorna de maneira amigável e consistente as informações obtidas nas análises.

Palavras-chave: Forense. Análise. Dump. Memória.

Abstract: This article aims to present the development of a toolkit for performing forensic analysis of memory called ToolForMe. ToolForMe is able to perform forensic analysis of memory dumps generated from computers with Linux OS. ToolForMe offers several plugins for analysis. This makes the user has a wide range of available functions, which are used to find traces and collect evidence in the memory about incidents on the computers / devices. To demonstrate the operation of the toolkit was conducted a case study where all available plugins were used. The results show that the tool returns friendly and consistent the information obtained in the analysis.

Keywords: Forensic. Analysis. Dump. Memory.

1 INTRODUÇÃO

Atualmente, a tecnologia se faz presente no cotidiano da maior parte da sociedade. Muitas pessoas passam parte do seu dia utilizando dispositivos, tais como, *tablets*, *smartphones*, *notebooks* e computadores, instalando nestes softwares/aplicativos e/ou acessando sites sem saber o que pode estar acontecendo com os seus dados e com as informações armazenadas nestes dispositivos. Muitos desses softwares podem conter vírus ou programas maliciosos, os quais podem estar utilizando dados ou executando ações em seu nome, sem a devida permissão.

Neste contexto, um campo que vem ganhando a cada dia mais espaço para que os autores destes incidentes possam ser identificados e suas ações desvendadas é o da Computação Forense. A Computação Forense tem a importante função de, por meio de aplicações de técnicas e ferramentas, levantar evidências e analisá-las, fazendo com que os dados que foram obtidos nestas análises, possam ser usados contra estes autores e, os mesmos, após serem identificados, sejam devidamente punidos.

Para identificar o que foi feito no sistema invadido, a perícia Forense computacional busca evidências relacionadas às atividades maliciosas que ocorreram no sistema, tradicionalmente, através da coleta e análise dos dados armazenados na memória estática do dispositivo. A Forense de memória permite à perícia Forense computacional a análise dos dados armazenados na RAM, que é a memória volátil do sistema. Estes dados podem ser de suma importância para a investigação, visto que, somente através desta análise, é possível identificar quais processos estão ativos, quais dados estão em uso pelos processos e muitas outras informações que só são armazenadas na memória volátil. Desta forma, a Forense de memória colabora para que a análise do sistema invadido seja mais completa.

Com base na necessidade de analisar os dados contidos na memória, este artigo apresenta um toolkit que detalha os resultados da extração e análise de dados da memória volátil de forma clara e objetiva, com uma interface amigável e de fácil entendimento. Isto faz com que este tipo de análise não seja feita somente em linha de comando, o que pode se tornar, muitas vezes, uma tarefa árdua e não intuitiva devido à quantidade e organização das informações contidas num dump de memória. A partir disso a criação e uso da ferramenta aqui proposta, até mesmo um usuário leigo pode visualizar os resultados obtidos da análise Forense de memória e realizar uma avaliação do que esta acontecendo ou aconteceu em seu sistema. Os resultados mostram que a ferramenta retorna de maneira amigável e consistente as informações obtidas nas análises.

O restante do artigo está organizado da seguinte maneira: na seção 2 são apresentados conceitos gerais sobre forense computacional. Na seção 3 é detalhada em especifico a forense em memória, foco deste trabalho. Na seção 4 o toolkit é apresentado. Na seção 5 um estudo de caso para validação do toolkit é apresentado. Por fim, na seção 6, uma conclusão e propostas para trabalhos futuros são apresentados.

2 FORENSE COMPUTACIONAL

A inovação tecnológica traz muitos benefícios para as pessoas e toda a comunidade, como a agilidade, a praticidade e a mobilidade, entre muitos outros. Porém, em contrapartida, surgem algumas desvantagens, como a possibilidade de realização de novas práticas criminosas e ilícitas. Desta maneira, para que os agentes destes fatos sejam punidos, existe a necessidade de uma investigação, a qual se inicia com a apuração e a análise dos vestígios deixados. A área que investiga esses fatos é conhecida como Computação Forense (ROSA, 2011).

Segundo Eleutério e Machado (2011), a Computação Forense é a ciência que usa técnicas especializadas para coletar, preservar e analisar os dados digitais de um computador ou computadores suspeitos de serem utilizados em um crime virtual. Da mesma forma que a perícia convencional, ela trata de buscar evidências para a solução de um crime.

A Computação Forense cada vez mais vem ganhando importância, tanto para autoridades policiais e judiciarias, como para empresas e organizações. Isto ocorre devido ao fato da necessidade de utilização de conhecimentos em informática juntamente com técnicas de investigação, a fim de obter evidências de ocorrências de incidentes de segurança em sistemas computacionais (FREITAS, 2006).

Um analista, usufruindo de processos Forenses, pode através da reconstituição dos eventos passados descobrir quem invadiu o sistema, como o invasor obteve o acesso, quando isto aconteceu e o que fez enquanto teve domínio do ambiente. Porém, tão importante quanto encontrar os responsáveis e buscar uma punição a eles, é o aprendizado resultante da análise do problema, que deve ser considerado como meta dos trabalhos Forenses aplicados à área de Segurança da Informação (ROSA, 2004)

2.1 Principais tipos de exames Forense Computacional

Os principais tipos de exames Forenses Computacionais realizados atualmente é o Forense *in vivo* e o Forense *post mortem*.

a. Exames e procedimentos no local do crime (Forense *in vivo*): Segundo Sacramento (2012), a Forense *in vivo* é a etapa onde o perito entra em contato com o incidente, onde o computador ainda está ligado, processos rodando e os ativos da rede ainda em funcionamento. Tudo o que é coletado, é armazenado num dispositivo de armazenamento para ser usado na análise pericial.

b. Exames em dispositivos de armazenamento Computacional (Forense *Post Mortem*): Para Eleutério e Machado (2011), exames em dispositivos de armazenamento computacional, são os exames periciais mais solicitados na Computação Forense, e consiste basicamente em analisar arquivos, sistemas e programas instalados em discos rígidos, CDs, DVDs, pen-drives e outros dispositivos de armazenamento digital de dados. As informações coletadas nesta análise ao serem cruzadas com as informações reunidas através da Forense *in vivo* traz uma análise geral do sistema computacional. O objetivo principal é reconstruir todos os fatos possíveis, e obter o máximo de informações para a elaboração do Laudo Pericial.

3 FORENSE DE MEMÓRIA (FORENSE *IN VIVO*)

A análise forense *in vivo* tem o objetivo de recuperar dados antes do desligamento da máquina, como os processos que estavam em execução no momento da apreensão da máquina, quais eram as conexões estabelecidas e até mesmo chaves de acesso a volumes cifrados, entre outros. Nesse contexto, uma abordagem que vem se mostrando bastante promissora é a Forense de Memória, que envolve a captura e análise dos dados armazenados na memória principal do computador/dispositivo (WAITS et al, 2008).

Amari (2009) define a Forense de Memória como sendo, o processo de captura e análise de dados armazenados em memória volátil. Sendo que, por memória volátil, entende-se aquela cujos dados podem se perder no desligamento do sistema, ou podem ser reescritos no funcionamento normal do mesmo.

A Forense de Memória é composta, basicamente, por dois procedimentos. O primeiro trata-se da captura da imagem da memória volátil, e o segundo é a análise desta imagem. Segundo Da Silva e Lorenz (2009), a captura da imagem da memória volátil se dá através de um *dump* de memória, podendo mapear como o sistema estava sendo utilizado no momento da geração da imagem. Já a análise da imagem gerada se dá através da utilização de ferramentas específicas, em um ambiente próprio para análise, pra que possa ser realizada a busca pelas evidências.

a. Extração dos dados da memória: existe uma diversidade de métodos para adquirir os dados presentes na memória. Estes podem ser baseados em hardware ou em software. O perito deve ter o conhecimento de qual se adequa mais ao sistema alvo e ao caso que está investigando. A seguir, serão mostrados os principais métodos usados para a extração dos dados em memória. (ROSA, 2011)

b. Análise dos dados da memória. Existem atualmente diversas ferramentas para realizar análise de dados da memória Volátil. Porém é importante que o perito conheça bem o comportamento das ferramentas que vai utilizar, testando-as em imagens de memória conhecidas antes de adotá-las em um processo de investigação formal, pois, desta forma, é possível conhecer as suas saídas, o formato em que as informações serão apresentadas ou até mesmo o desconhecimento do comportamento do software.

4 DESENVOLVIMENTO DO TOOLFORME

A Linguagem de Programação utilizada no desenvolvimento do *toolkit* ToolForMe, foi o *Python.* O mesmo foi utilizado no desenvolvimento da Interface *Web* do *toolkit*, juntamente com os *Frameworks Django* e *Bootstrap*. Para realização das análises forenses foi utilizado o *framewok volatility*. Este foi escolhido por ter um amplo range de plugins. Apesar desse amplo range, este framework é limitado a execução por linha de comando. Tal abordagem exige do usuário um grau elevado de conhecimento para compreender cada comando e a composição de parâmetros para que as informações sejam analisadas, ao contrário do toolkit proposto que possui uma interface gráfica intuitiva. Por fim, o SGBD utilizado foi o MySQL para armazenamentos dos resultados das análises forense de memória.

Para o desenvolvimento do toolkit utilizou-se o sistema Operacional Linux Ubuntu, por ser uma distribuição que atende aos requisitos para desenvolvimento do mesmo.

Foi criado então o projeto Django, sendo que este cria automaticamente arquivos para o desenvolvimento do projeto. Dentre os arquivos do Django, estão os que mais foram utilizados para a realização do desenvolvimento do projeto ToolForMe, e também para a configuração do mesmo. São eles: views.py, urls.py, models.py, forms.py e settings.py. Nestes foram realizadas todas as configurações necessárias e criadas as funções, url's, formulários e *templates* necessários para a criação do toolkit.

A tela inicial do toolkit pode ser vista na Figura 1.

Fig. 1 - Tela Inicial Do Toolforme

O ToolForMe possui a tela inicial que apresenta uma breve descrição do *toolkit* ToolForMe, como o que ele faz e também através de que ele faz, conforme mostra a figura 1. A partir dai, têm-se as telas de "Home", que volta a tela inicial, "Do Upload" onde é realizado o *upload* do *dump*, "Perform Analysis" onde é realizada a análise forense de memória do *dump* feito upload anteriormente, "Consultations" que é a tela onde é possível visualizar os resultados obtidos das analises realizadas, e a tela "Contact", que contem informações do desenvolvedor.

Na tela "Do Upload", o usuário seleciona o dump que deseja fazer análise, e seleciona de qual sistema operacional o mesmo foi extraído. Esta tela possui a mesma estrutura da tela inicial, porém, ao invés de conter informações sobre o *toolkit*, possui o formulário criado no arquivo *forms.py* e o botão "Add on", onde o usuário fará *upload* do *dump* que deseja analisar. Após realizar o upload do dump que deseja analisar, na tela "Perform Analysis", o usuário poderá analisar forense de memoria deste dump, utilizando os *plugins* disponíveis pelo *toolkit*. A tela "Perform Analysis" pode ser visualizado na figura 2.

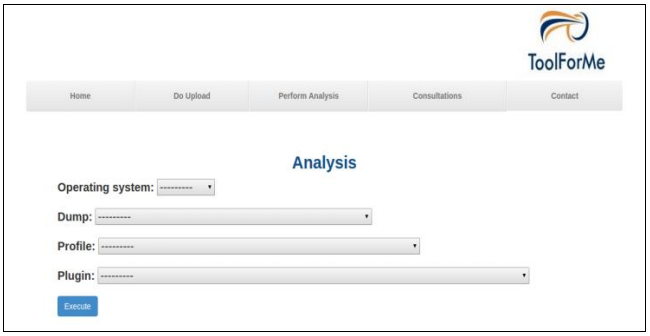


Fig. 2 - Tela "Perform Analysis

Os *plugins* disponíveis para análise de *dumps* gerados a partir de sistemas operacionais Linux, juntamente com os dados que retornam são:

- Linux_bash – Apresenta um histórico dos sistemas;
- Linux_dmesg – Mostra as ultimas ocorrências de buffer do Kernel;
- Linux_ifconfig – Apresenta as interfaces ativas;
- Linux_lsmod – Apresenta os módulos de Kernel carregados;
- Linux_netstat – Apresenta as conexões de rede ativas;
- Linux_pslist – Mostra os processos que estavam ativos;
- Linux_proc_maps – Mostra os mapas dos processos do Linux;
- Linux_pstree – Mostra a relação pai/filho dos processos;
- Linux_cpuinfo - Imprime informações sobre cada processador ativo;
- Linux_lsof - Lista os arquivos abertos;
- Linux_arp – Imprime a tabela ARP.

Quando o usuário selecionar a opção executar na tela de realizar análise, o sistema vai realizar a análise utilizando o *framework volatility*, vai inserir todas as informações coletadas do *dump* na tabela do *plugin* escolhido pelo usuário no Banco de Dados, e logo após mostra na tela os resultados obtidos da análise realizada. Através das informações armazenadas o usuário poderá posteriormente visualizar os resultados obtidos na análise de cada *dump* na tela de consultas.

5 ESTUDO DE CASO E RESULTADOS

Depois de finalizado o desenvolvimento do toolkit, partiu-se então para a realização dos estudos de caso para comprovação das funcionalidades do mesmo. Nesta etapa, foi realizado, até o presente momento, um estudo de caso analisando um *dump* extraído de um computador que possui Sistema Operacional Linux Ubuntu.

A extração do *dump* deu-se através do Lime (https://code.google.com/p/lime-forensics), uma ferramenta para extração de imagens de memória de Sistemas Operacionais baseados em Unix. O comando executado para a criação do *dump* pode ser visualizado na figura 3.

```
root@notebook1:/home/cce# cd lime-forensics
root@notebook1:/home/cce/lime-forensics# ls
doc  src
root@notebook1:/home/cce/lime-forensics# cd src
root@notebook1:/home/cce/lime-forensics/src# ls
disk.c  disk.c~  lime-3.11.0-26-generic.ko  lime.h  main.c  Makefile  Makefile.sample  tcp.c  tcp.c~
root@notebook1:/home/cce/lime-forensics/src# insmod lime-3.11.0-26-generic.ko "path=/root/mem.dump format=lime"
```

Fig. 3 - Comando Lime Para Criação Do Dump Linux

Após ter sido gerado o *dump*, foi realizado o *upload* do arquivo no *toolkit*. Feito isto, partiu-se para a realização da análise utilizando *o toolkit*. Como o *dump* foi extraído de uma máquina contendo o Sistema Operacional Linux, escolheu-se então o Sistema Operacional Linux. Foi escolhido também o *dump* que foi realizado *upload* anteriormente. O *profile* escolhido foi o "LinuxUbuntuServer-amd64-linux-image-3_11_0-12-generic-x64" devido o mesmo ser extraído de um computador que possui *Ubuntu*, com kernel de 64*Bits*.

Neste estudo de caso foram utilizados todos os *plugins* disponíveis no toolkit para realização da análise forense de memória de Sistemas Operacionais Linux. O resultado obtido através do *plugin* "Linux_dmesg", que mostra quais são as últimas ocorrências de *buffer* do *Kernel*, pode ser visualizado na figura 4.

Result Of Analysis

ID	MOMENT	OCURRENCE
1	0.0	Initializing cgroup subsys cpuset
2	0.0	Initializing cgroup subsys cpu
3	0.0	Initializing cgroup subsys cpuacct
4	0.0	Linux version 3.11.0-12-generic (buildd@allspice) (gcc version 4.8.1 (Ubuntu/Linaro 4.8.1-10ubuntu7)) #19-Ubuntu SMP Wed Oct 9 16:20:46 UTC 2013 (Ubuntu 3.11.0-12.19-generic 3.11.3)
5	0.0	Command line: BOOT_IMAGE=/boot/vmlinuz-3.11.0-12-generic root=UUID=b805ad8b-c7c9-4e07-9e32-e4b94d65118b ro quiet splash vt.handoff=7
6	0.0	KERNEL supported cpus:
7	0.0	Intel GenuineIntel
8	0.0	AMD AuthenticAMD
9	0.0	Centaur CentaurHauls
10	0.0	e820: BIOS-provided physical RAM map:

Fig. 4- Resultado da Análise do plugin Linux_dmesg

A Figura 4, mostra que utilizando o *plugin* "Linux-dmesg", o toolkit desenvolvido retorna o MOMENT e a OCURRENCE, ou seja, o momento em que determinada ação aconteceu no sistema, e passou pela memória, deixando ali o seu rastro. Este *plugin* trouxe como resposta 933 linhas de informações.

O resultado da análise utilizando o plugin "Linux_netstat", que verifica quais conexões de rede estavam sendo feitas no momento em que a imagem da memória foi gerada, pode ser visualizado na figura 5.

Result Of Analysis

ID	PROTOCOL	IP	CONNECTION	STATUS	PROCESS
1	TCP	::1:631	:::0	LISTEN	cupsd/694
2	TCP	127.0.0.1:631	0.0.0.0:0	LISTEN	cupsd/694
3	TCP	127.0.0.1:3306	0.0.0.0:0	LISTEN	mysqld/1012
4	TCP	0.0.0.0:0	0.0.0.0:0	CLOSE	apache2/1275
5	TCP	:::80	:::0	LISTEN	apache2/1275
6	TCP	0.0.0.0:0	0.0.0.0:0	CLOSE	apache2/1447
7	TCP	:::80	:::0	LISTEN	apache2/1447
8	TCP	0.0.0.0:0	0.0.0.0:0	CLOSE	apache2/1448
9	TCP	:::80	:::0	LISTEN	apache2/1448
10	TCP	0.0.0.0:0	0.0.0.0:0	CLOSE	apache2/1449
11	TCP	:::80	:::0	LISTEN	apache2/1449
12	TCP	0.0.0.0:0	0.0.0.0:0	CLOSE	apache2/1450

Fig. 5 – Resultado da Análise do plugin Linux_netstat

Conforme pode ser visualizado na Figura 5, este *plugin* lista todas as portas que estão esperando conexão, ou seja, as portas que estão com seu estado *listening* (ouvindo). Lista também as portas no estado *established* (onde a conexão já esta ativa) e as portas no estado *close* e *close_wait* (onde a conexão já foi encerrada).

O "linux_netstat" verifica as conexões TCP, e através dele é possível verificar se algo ou alguém está conectado ao *host* local. Da mesma forma, algumas conexões TCP são desnecessárias, fazendo com que a velocidade e o desempenho do *host* sejam prejudicados, visto que estas conexões consomem boa parte dos recursos de sistema.

No resultado da análise, exibida na Figura 6, o *toolkit* recebeu como resposta 85 linhas de informações. Percebe-se que este *plugin* retorna ao usuário o PROTOCOL, o IP, a CONNECTION e o STATUS.

Outro *plugin* apresentado é o "linux_lsmod". Este *plugin* é utilizado para verificar quais os módulos de *Kernel* estão carregados no momento que o *dump* foi gerado. Como pode ser visualizado na Figura 6, este *plugin* retorna o NAME, ou seja, o nome do módulo, e o NUMBER, que é o número para localização do módulo.

Result Of Analysis

ID	NAME	NUMBER
1	lime	18111
2	michael_mic	12612
3	arc4	12608
4	cfg80211	479757
5	parport_pc	32701
6	ppdev	17671
7	rfcomm	69070
8	bnep	19564
9	bluetooth	371874
10	snd_hda_codec_hdmi	41276
11	snd_hda_codec_realtek	51465
12	joydev	17377

Fig. 6 - Resultado da Análise do plugin Linux_Lsmod

Nesta análise, utilizando o *plugin* "linux_lsmod", o *toolkit* retornou 52 resultados, ou seja, 52 módulos do *Kernel* carregados na memória do computador no momento que foi gerado o *dump*.

O próximo *plugin* utilizado foi o "linux_arp". Este imprime a tabela ARP ativa no computador analisado. Esta análise retornou 9 resultados, como pode ser visualizado na figura 7.

Result Of Analysis

ID	IP	MAC	STATUS	INTERFACE
1	ff02::fb	33:33:00:00:00:fb	on	wlan0
2	ff02::2	33:33:00:00:00:02	on	wlan0
3	ff02::16	33:33:00:00:00:16	on	wlan0
4	ff02::1:ff70:ab12	33:33:ff:70:ab:12	on	wlan0
5	224.0.0.22	01:00:5e:00:00:16	on	wlan0
6	10.0.0.1	00:1a:3f:88:8f:58	on	wlan0
7	127.0.0.1	00:00:00:00:00:00	on	lo
8	224.0.0.251	01:00:5e:00:00:fb	on	wlan0
9	127.0.1.1	00:00:00:00:00:00	on	lo

Fig. 7 – Resultado da Análise do plugin Linux_Arp

Conforme pode-se notar, a análise utilizando o *plugin* "linux_arp" retorna ao usuário os seguintes dados: o IP da máquina na qual foi realizada a comunicação via tabela ARP, o endereço MAC desta máquina, o STATUS da conexão e a INTERFACE de comunicação.

O próximo *plugin* o qual foi realizada a análise para o estudo de caso é o "linux_ifconfig" (Figura 8). Este *plugin* retorna ao usuário às interfaces de rede ativas no momento que foi gerada a imagem de memória.

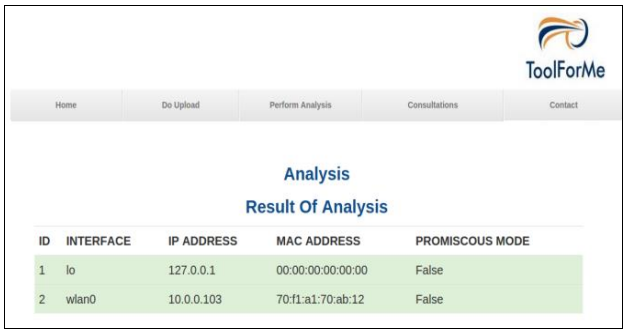


Fig. 8 – Resultado da Análise do plugin Linux_Ifconfig

A análise realizada com o *plugin* "linux_ifconfig", conforme pode ser vista na Figura 9, retorna ao usuário a INTERFACE, o IP ADDRESS, o MAC ADDRESS e o PROMISCOUS MODE.

O próximo *plugin*, o qual foi utilizado para realização da análise do *dump* é o "linux_bash". Este *plugin* retorna um histórico de comandos utilizados no sistema e que estavam armazenados na memória no momento da geração do *dump*.

Conforme pode ser visto na Figura 9, na análise realizada neste estudo, o *plugin* retornou como resultado 726 linhas de comando que foram executadas, e estavam armazenadas na memória. Como pode-se verificar na Figura 10, o *plugin* utilizado retorna o PID, o NAME, o COMMAND TIME, ou seja, momento que o comando foi executado com a data e hora e o COMMAND, ou seja, o comando executado.

Result Of Analysis

ID	PID	NAME	COMMAND TIME	COMMAND
1	5186	bash	2014-07-18 00:12:09 UTC+0000	cd volatility/tools/linux
2	5186	bash	2014-07-18 00:12:09 UTC+0000	#dd</dev/mem> mem.dump
3	5186	bash	2014-07-18 00:12:09 UTC+0000	sudo apt-get install adobe
4	5186	bash	2014-07-18 00:12:09 UTC+0000	sudo apt-get install flash
5	5186	bash	2014-07-18 00:12:09 UTC+0000	ls

Fig. 9 - Resultado da Análise do plugin Linux_Bash

Dando continuidade ao estudo de caso da imagem do Sistema Operacional Linux, foi realizada a análise utilizando o *plugin* "linux_cpuinfo" (Figura 10). Este *plugin* retorna ao usuário as informações relacionadas à CPU do computador analisado.

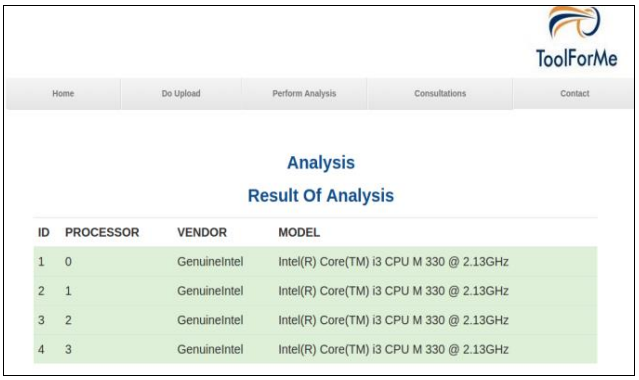


Fig. 10 – Resultado da Análise do plugin Linux_Cpuinfo

Conforme pode ser visualizado na figura 10, o *plugin* "Linux_cpuinfo" retorna ao usuário o PROCESSOR, o VENDOR, ou seja, o vendedor/fabricante, e o MODEL da CPU do *dump* que está sendo analisado.

O próximo *plugin*, utilizado neste estudo de caso para realização da análise do *dump* é o "Linux_lsof". Este *plugin* imprime a lista de arquivos abertos e seus respectivos caminhos, para cada processo em execução. Parte do resultado da análise utilizando o "Linux_lsof" pode ser visualizado na figura 11.

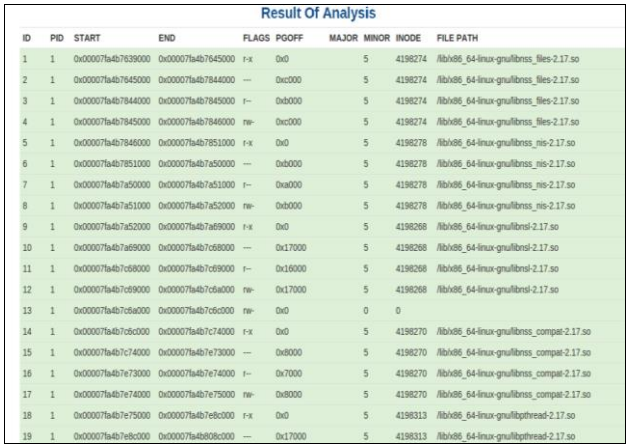
Result Of Analysis

ID	PID	FD	PATH
1	1	0	/dev/null
2	1	1	/dev/null
3	1	2	/dev/null
4	1	3	pipe:[8268]
5	1	4	pipe:[8268]
6	1	5	/anon_inode:/inotify
7	1	6	/anon_inode:/inotify
8	1	7	socket:/UNI:[6666]
9	1	9	socket:/UNI:[1683]
10	1	10	socket:/UNI:[6720]
11	1	11	socket:/UNI:[1569]
12	1	12	/var/log/upstart/modemmanager.log

Fig. 11 - Resultado da Análise do plugin Linux_Lsof

Conforme pode ser visualizado na Figura 11, o *plugin* "Linux_lsof" traz como resultados PID, o FD, e o PATH, ou seja, a lista de arquivos abertos por cada processo e seus respectivos caminhos. Este *plugin* trouxe como resultado 2041 linhas de informações

Dando continuidade as análises do presente estudo de caso, utilizou-se do *plugin* "linux_proc_maps" para realização de mais uma etapa da análise. Este *plugin* lista os mapas, mostrando detalhes de cada processo no Linux, e o resultado desta análise pode ser visualizado na figura 12.



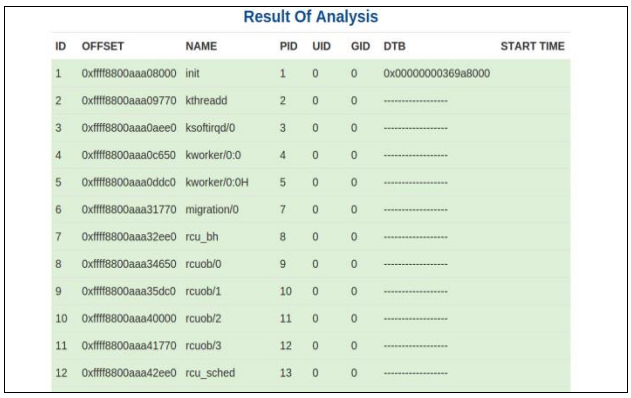
Result Of Analysis

ID	PID	START	END	FLAGS	PGOFF	MAJOR	MINOR	INODE	FILE PATH
1	1	0x00007fa4b7639000	0x00007fa4b7645000	r-x	0x0		5	4198274	/lib/x86_64-linux-gnu/libnss_files-2.17.so
2	1	0x00007fa4b7645000	0x00007fa4b7844000	---	0xc000		5	4198274	/lib/x86_64-linux-gnu/libnss_files-2.17.so
3	1	0x00007fa4b7844000	0x00007fa4b7845000	r--	0xb000		5	4198274	/lib/x86_64-linux-gnu/libnss_files-2.17.so
4	1	0x00007fa4b7845000	0x00007fa4b7846000	rw-	0xc000		5	4198274	/lib/x86_64-linux-gnu/libnss_files-2.17.so
5	1	0x00007fa4b7846000	0x00007fa4b7851000	r-x	0x0		5	4198278	/lib/x86_64-linux-gnu/libnss_nis-2.17.so
6	1	0x00007fa4b7851000	0x00007fa4b7a50000	---	0xb000		5	4198278	/lib/x86_64-linux-gnu/libnss_nis-2.17.so
7	1	0x00007fa4b7a50000	0x00007fa4b7a51000	r--	0xa000		5	4198278	/lib/x86_64-linux-gnu/libnss_nis-2.17.so
8	1	0x00007fa4b7a51000	0x00007fa4b7a52000	rw-	0xb000		5	4198278	/lib/x86_64-linux-gnu/libnss_nis-2.17.so
9	1	0x00007fa4b7a52000	0x00007fa4b7a69000	r-x	0x0		5	4198268	/lib/x86_64-linux-gnu/libnsl-2.17.so
10	1	0x00007fa4b7a69000	0x00007fa4b7c68000	---	0x17000		5	4198268	/lib/x86_64-linux-gnu/libnsl-2.17.so
11	1	0x00007fa4b7c68000	0x00007fa4b7c69000	r--	0x16000		5	4198268	/lib/x86_64-linux-gnu/libnsl-2.17.so
12	1	0x00007fa4b7c69000	0x00007fa4b7c6a000	rw-	0x17000		5	4198268	/lib/x86_64-linux-gnu/libnsl-2.17.so
13	1	0x00007fa4b7c6a000	0x00007fa4b7c6c000	rw-	0x0		0	0	
14	1	0x00007fa4b7c6c000	0x00007fa4b7c74000	r-x	0x0		5	4198270	/lib/x86_64-linux-gnu/libnss_compat-2.17.so
15	1	0x00007fa4b7c74000	0x00007fa4b7e73000	---	0x8000		5	4198270	/lib/x86_64-linux-gnu/libnss_compat-2.17.so
16	1	0x00007fa4b7e73000	0x00007fa4b7e74000	r--	0x7000		5	4198270	/lib/x86_64-linux-gnu/libnss_compat-2.17.so
17	1	0x00007fa4b7e74000	0x00007fa4b7e75000	rw-	0x8000		5	4198270	/lib/x86_64-linux-gnu/libnss_compat-2.17.so
18	1	0x00007fa4b7e75000	0x00007fa4b7e8c000	r-x	0x0		5	4198313	/lib/x86_64-linux-gnu/libpthread-2.17.so
19	1	0x00007fa4b7e8c000	0x00007fa4b808c000	---	0x17000		5	4198313	/lib/x86_64-linux-gnu/libpthread-2.17.so

Fig. 12 – Resultado da Análise do plugin Linux_Proc_Maps

Na Figura 12, pode-se verificar o resultado obtido ao utilizar-se do *plugin* "Linux_proc_maps", apresentando 24.859 linhas de resultados. O mesmo retorna 10 colunas, o PID, o START, o END, as FLAGS, o PGOFF, o MAJOR, o MINOR , o INODE e o FILE PATH.

O "Linux_pslist" também foi utilizado para realização da análise do *dump* extraído do sistema operacional Linux. Este *plugin* imprime a lista de processos ativos no momento em que o *dump* de memória foi gerado. Na Figura 13, tem-se o resultado da análise utilizando-se deste *plugin.*



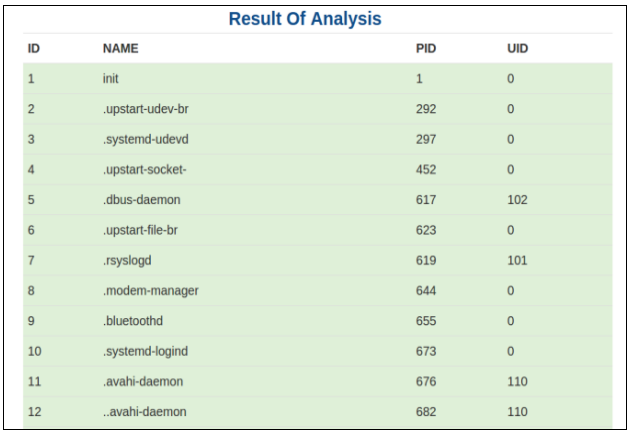
Result Of Analysis

ID	OFFSET	NAME	PID	UID	GID	DTB	START TIME
1	0xffff8800aaa08000	init	1	0	0	0x00000000369a8000	
2	0xffff8800aaa09770	kthreadd	2	0	0	----------	
3	0xffff8800aaa0aee0	ksoftirqd/0	3	0	0	----------	
4	0xffff8800aaa0c650	kworker/0:0	4	0	0	----------	
5	0xffff8800aaa0ddc0	kworker/0:0H	5	0	0	----------	
6	0xffff8800aaa31770	migration/0	7	0	0	----------	
7	0xffff8800aaa32ee0	rcu_bh	8	0	0	----------	
8	0xffff8800aaa34650	rcuob/0	9	0	0	----------	
9	0xffff8800aaa35dc0	rcuob/1	10	0	0	----------	
10	0xffff8800aaa40000	rcuob/2	11	0	0	----------	
11	0xffff8800aaa41770	rcuob/3	12	0	0	----------	
12	0xffff8800aaa42ee0	rcu_sched	13	0	0	----------	

Fig. 13 – Resultado da Análise do plugin Linux_Pslist

Analisando a figura 13, pode-se notar que o *plugin* "Linux_pslist" retorna como resultado da análise diversas informações. São elas: OFFSET, o NAME, o PID, o UID, o GID, o DTB, o START TIME. Neste caso, este *dump* não possui informações de START TIME, ou seja, não possui informações de quando o processo foi iniciado. Nesta análise, o *plugin* retornou 214 linhas de resultados.

O próximo e último *plugin* utilizado no estudo de caso para verificação da funcionalidade do ToolForMe para o Sistema Operacional Linux, é o "Linux_pstree". Este *plugin* mostra as relações pai/filho entre os processos executados na máquina analisada. O resultado obtido na análise utilizando o *plugin* "Linux_pstree" pode ser visualizado na figura 14.



Result Of Analysis

ID	NAME	PID	UID
1	init	1	0
2	.upstart-udev-br	292	0
3	.systemd-udevd	297	0
4	.upstart-socket-	452	0
5	.dbus-daemon	617	102
6	.upstart-file-br	623	0
7	.rsyslogd	619	101
8	.modem-manager	644	0
9	.bluetoothd	655	0
10	.systemd-logind	673	0
11	.avahi-daemon	676	110
12	..avahi-daemon	682	110

Fig. 14 - Resultado da Análise do plugin Linux_Pstree

Conforme pode ser visualizado na figura 14, a análise realizada pelo *plugin* "linux_pstree" traz como resultado ao usuário o NAME, o PID e o UID. Esta análise retornou 214 linhas de informações.

Como pode ser verificado, no estudo de caso realizado, foram utilizados todos os *plugin*s disponíveis para o Sistema Operacional Linux no *toolkit* ToolForMe. Diante dos resultados obtidos, e sabendo que todos os *plugins* funcionaram corretamente e retornaram valores corretos, confirma-se então a veracidade das análises realizadas através do *framework volatility* no *toolkit* ToolForMe, em dumps gerados a partir de Sistemas Operacionais Linux.

6 CONCLUSÃO E TRABALHOS FUTUROS

A ideia principal deste trabalho foi o desenvolvimento de um toolkit para análise forense de memória. Esta ideia surgiu da necessidade de se ter uma interface amigável e de fácil entendimento ao usuário para realizar e verificar os resultados de tais análises. Isto faz com que este tipo de análise não seja feita somente em linha de comando, que pode apresentar ao usuário informações de forma desorganizada e/ou ilegível.

Neste trabalho tanto a criação do toolkit quanto seus experimentos foram voltados para plugins Linux, visto que o estudo de caso realizado foi em um dump gerado a partir de uma máquina que possuía sistema Operacional Linux. Como sugestão para trabalhos futuros, pretende-se estender as funcionalidades da ferramenta para análise de dumps de sistemas operacionais Windows

REFERÊNCIAS

AMARI, Kristine; Techniques and Tools for Recovering and Analyzing Data from Volatile Memory, SANS Institute, 2009.

DA SILVA, Gilson Marques; LORENS, Evandro Mário. *Extração e Análise de Dados em Memória na Perícia* Forense Computacional. Honorary President, p. 21, 2009

ELEUTERIO, Pedro Monteiro da Silva; MACHADO, Marcio Pereira. *Desvendando a Computação Forense*. 1. Ed. São Paulo: Novatec, 2011.

FREITAS, Andrey Rodrigues de; Perícia Forense *Aplicada à Informática*: Ambiente Microsoft; Rio de Janeiro, Brasport, 2006.

LiME, LiME - Linux Memory Extractor, Disponivel em <https://code.google.com/p/lime-forensics/> Acesso em 17 maio 2014.

ROSA, Ana Paula Teixeira. *Forense de Memória: Extração e Análise de Dados Armazenados em Memória Volátil*. Monografia de Especialização – Universidade de Brasília, Faculdade de Tecnologia. Departamento de Engenharia Elétrica. Brasilia, DF, 2011

ROSA, Daniel Accioly; Contextualização da Prática Forense; *Revista Evidência Digital* – Edição 01, Ano I, n. 01, p. 5, Jan., Fev., Mar. 2004. Disponível em: <http://www.guiatecnico.com.br/evidenciadigital>. Acesso em: 30 abr. 2014.

SACRAMENTO, Fabricio Santos; *Estudo sobre as ferramentas de rede para Perícia Forense.* Faculdade Norte Capixaba de São Mateus; Análise e Desenvolvimento de Sistemas. São Mateus. ES, 2012.

SILVA, Ewerton Almeida, and ROCHA, Anderson. *Análise forense de documentos digitais: além da visão humana*. Saúde, Ética & Justiça 16.1 (2011).

SILVA, Rodrigo Segura; *Forense Digital:* Produzindo Provas Legais; Out, 2010. Disponível em <http://www.prevenirperdas.com.br/portal/index.php?option=com_content&view=article&id=177:forensedigital&catid=18:prevencao-a-fraudes&Itemid=7> Acesso em: 11 maio 2014.

WAITS, C., Akinyele, J.A., Nolan, R., Rogers, L.: *Computer Forensics: Results of Live Response Inquiry vs. Memory Image Analysis*, CERT,2008..

SIMULAÇÃO E ANÁLISE DO ALGORITMO DE PERTURBAÇÃO E OBSERVAÇÃO NO RASTREAMENTO DO PONTO DE MÁXIMA POTÊNCIA EM SISTEMAS FOTOVOLTAICOS

SIMULATION AND ANALYSIS OF PERTURB AND OBSERVE ALGORITHM AT MAXIMUM POWER POINT TRACKING OF PHOTOVOLTAIC SYSTEMS

LEONARDO ROMITTI[1]*, FABRÍCIO HOFF DUPONT[1]

[1]Departamento de Engenharias e Ciência da Computação, Universidade Regional Integrada do Alto Uruguai e das Missões, URI - Câmpus de Frederico Westphalen
*E-mail: leonardo.romitti@gmail.com

Resumo: Os valores de corrente e tensão de saída das células fotovoltaicas são sensíveis a variações de irradiação e de temperatura. Assim, é necessário utilizar um algoritmo de rastreamento do ponto de máxima potência (MPPT) para otimizar a geração de energia. Um dos métodos mais utilizados para esta finalidade é o de Perturbação e Observação (P&O). Deste modo, este trabalho tem como objetivo validar o funcionamento do algoritmo de P&O no rastreamento do ponto de máxima potência (MPP) em sistemas fotovoltaicos e analisar a influência da variação dos parâmetros de incremento da razão cíclica e período de amostragem na dinâmica do sistema utilizando um conversor buck-boost.

Palavras-chave: Sistemas Fotovoltaicos. Rastreamento do Ponto de Máxima Potência (MPPT). Algoritmo de Perturbação e Observação (P&O). Otimização;

Abstract: The current and the voltage from photovoltaic cells are sensitive to irradiance and temperature variations. Thus, is necessary to apply a Maximum Power Point Tracking algorithm (MPPT) to optimize the energy generation. One of the most widely used techniques for this purpose is the Perturb and Observe Method (P&O). In this context, this paper aims to validate the P&O algorithm efficiency, analyzing duty cycle's increment and sampling frequency influence in dynamic system behavior using a buck-boost converter.

Keywords: Photovoltaic Systems. Maximum Power Point Tracking (MPPT). Perturb and Observe Algorithm (P&O). Optimization

1 INTRODUÇÃO

Os padrões de vida atuais apresentam uma dependência irreversível e crescente de energia elétrica, sendo que o fornecimento para atender esta demanda tem sido feito de forma insustentável através de uma matriz energética global baseada em combustíveis fósseis.

No entanto, o movimento crescente da sociedade em prol do desenvolvimento de alternativas para compor uma nova matriz energética tem incentivado um grande aumento no desenvolvimento de novas tecnologias para geração de eletricidade. Neste contexto, a energia solar fotovoltaica tem se consolidado como uma fonte renovável e inesgotável na escala de tempo humana.

A potência fornecida por um dispositivo fotovoltaico pode ser representada por uma curva de potência em função da tensão chamada de curva P – V, onde existe apenas uma combinação (localizada no joelho da curvatura) que garante a entrega da máxima potência à carga, o chamado ponto de máxima potência (mpp).
no entanto, os valores de tensão e corrente de saída destes dispositivos são extremamente sensíveis a variações de irradiação e temperatura de modo que potência que é fornecida à carga e, consequentemente, a localização do MPP, sofrem variações ao longo do dia.

Desde modo, é necessário utilizar um método ativo de Rastreamento do Ponto de Máxima Potência (MPPT) aplicado ao sistema de potência para fazer com que o mesmo opere o mais próximo possível do MPP de modo a otimizar a geração de energia.

De acordo com Esram (2007) existem muitos métodos de MPPT desenvolvidos e implementados, entre os quais pode-se citar como exemplo: *Hill Climbing*, Perturbação e Observação (P&O), Condutância Incremental, Método por Lógica *Fuzzy*, Método por Rede Neural, Correlação de *Ripple* (RCC), *Current Sweep*, Fração da Corrente de Curto Circuito e Fração da Tensão de Circuito Aberto. No entanto, segundo Femia *et al* (2005) o algoritmo mais empregado é o de P&O devido a sua simplicidade e facilidade de implementação.

Neste contexto, este trabalho tem os objetivos de validar o funcionamento do método de P&O no rastreamento do MPP em sistemas fotovoltaicos e avaliar o impacto dos diferentes parâmetros que podem ser configurados no algoritmo estudado através de simulações desenvolvidas com o auxílio do software Simulink.

2 DESENVOLVIMENTO

De acordo com Martins *et al* (2011), um sistema de rastreamento pode ser dividido em duas partes. A primeira é um modelo que representa a lógica de funcionamento do algoritmo, fazendo a leitura das informações do módulo, executando os cálculos e definindo o ponto de operação do sistema. A segunda é o estágio de potência que tem como objetivo habilitar o funcionamento do método de MPPT aplicado. A Fig. 1 ilustra o conceito apresentado.

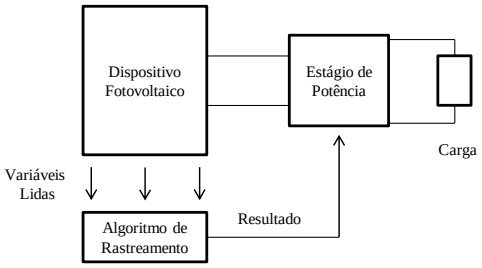


Fig. 1: Sistema típico de rastreamento do Ponto de Máxima Potência em sistemas fotovoltaicos. Fonte: Martins et al 2011.

Para este trabalho, será considerada uma carga com característica resistiva pelo fato de que o estudo conduzido não possui foco em uma aplicação específica, mas sim, a um meio que possibilite a simulação e a validação do método investigado.

2.1 Modelagem matemática e especificações do módulo fotovoltaico

De acordo com Dupont (2014) a célula fotovoltaica pode ser entendida fisicamente como uma junção *p-n* que ao ser exposta à irradiação solar é capaz de gerar uma corrente elétrica. O circuito equivalente que representa este comportamento é ilustrado pela Fig. 2.

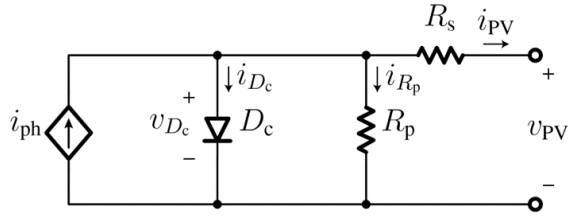


Fig. 2: Circuito equivalente de uma célula fotovoltaica utilizando o modelo de única exponencial. Fonte: Dupont (2014).

No circuito ilustrado, i_{ph} é a corrente gerada pela interação fóton-elétron, D_c é a junção entre os semicondutores *p* e *n*, v_{D_c} é a tensão sobre o diodo, i_{D_c} é a corrente que passa pelo diodo D_c, R_p representa as correntes de fuga que dão origem às perdas internas da célula, i_{R_p} é a corrente na resistência em paralelo, R_S representa as perdas ôhmicas nos contatos metálicos da célula e i_{PV} e v_{PV} representam respectivamente a corrente e a tensão de saída do dispositivo.

Dupont (2014) também apresenta a análise matemática do circuito equivalente da Fig. 2. A corrente de saída da célula é definida através de

$$i_{PV} = i_{ph} - i_{Dc} - i_{Rp} \quad (1)$$

onde i_{ph} pode ser calculada por

$$i_{ph} = \frac{S}{S^{ref}} i_{sc}^{ref} + (T_{op} - T^{ref})\mu_{icc} \quad (2)$$

na qual S é a radiação solar na superfície do módulo, S^{ref} e T^{ref} representam a radiação solar e a temperatura de referência dentro das chamadas de Condições Padrão de Teste (STC - *Standard Test Conditions*), que são respectivamente 1000 W/m² e 25ºC. A grandeza i_{sc}^{ref} é a corrente foto-gerada nas condições de referência e μ_{icc} é o coeficiente de corrente de curto circuito com a temperatura. A variável T_{op} é a temperatura de operação do painel e pode ser obtida através de

$$T_{op} = T_a + (T_{NOCT} - T^{ref})\frac{S}{S^{ref}} \quad (3)$$

sendo T_a a temperatura ambiente e T_{NOCT} a temperatura nominal de operação da célula em K (*Kelvin*).

A corrente que passa pelo diodo é calculada através de

$$i_D = i_0\left(e^{\frac{qV_D}{akT}} - 1\right) \quad (4)$$

onde q é a carga elementar do elétron ($1{,}6 \cdot 10^{-19}$ C), a é o fator de idealidade de D_c, T é a temperatura na superfície do módulo e k é a Constante de Boltzmann ($1{,}38 \cdot 10^{-23}\ J/K$). A corrente i_0, que representa a corrente de saturação do diodo, podendo ser calculada através de

$$i_0 = i_0^{ref}\left(\frac{T}{T_{ref}}\right)^3 e^{\frac{qE_g}{ak}\left(\frac{1}{T^{ref}} - \frac{1}{T}\right)} \quad (5)$$

de modo que E_g representa a energia de banda proibida do material semicondutor que compõe a célula e i_0^{ref} é a corrente de saturação do diodo nas condições de referência, definida por

$$i_o^{ref} = \frac{i_{sc}^{ref}}{e^{\left(\frac{v_{oc}^{ref}}{av_T^{ref}}\right)} - 1} \quad (6)$$

onde v_{oc}^{ref} é a tensão de circuito aberto e v_T^{ref} é a tensão térmica nas condições padrão de teste.

Dupont (2014) complementa a análise matemática do circuito definindo a tensão no diodo através de

$$v_{Dc} = v_{PV} - R_s i_{PV} \quad (7)$$

e a corrente de fuga que flui pela resistência em paralelo R_p através de

$$i_{R_p} = \frac{v_{D_c}}{R_p} \quad (8)$$

Assim, é possível estimar a corrente e a tensão produzidas por determinada célula fotovoltaica em função de características físicas e ambientais.

Tendo em vista que a potência fornecida por estes dispositivos é insuficiente para suprir a demanda da maioria das cargas, as células são agrupadas de modo a formar módulos fotovoltaicos. Por sua vez, os módulos são organizados em séries e/ou paralelos para formar arranjos ou painéis.

A análise apresentada nos parágrafos anteriores pode ser representada em simulações no Simulink através dobloco *PV Array* desenvolvido pelo *National Renewable Energy Laboratory* (2014), onde é possível implementar o comportamento de diversos módulos disponíveis no mercado.

Neste contexto, foi utilizado para o desenvolvimento deste estudo um módulo fotovoltaico KD205GX-LP da Kyocera Solar. Os valores de tensão, corrente e potência de saída deste modelo são, respectivamente, 26,6 V, 7,71 A e 205,08 W.

2.2 Projeto de um conversor buck-boost para habilitar o funcionamento do algoritmo

A partir das características elétricas do módulo escolhido, foi desenvolvido o projeto de um conversor buck-boost para habilitar o funcionamento do algoritmo estudado considerando-se os parâmetros apresentados na Tabela 1.

Optou-se por utilizar o conversor buck-boost devido à sua eficiência no rastreamento do MPPT, como é explicado em Martins *et al* (2011). Maiores detalhes sobre a topologia escolhida podem ser obtidos em Barbi (2000) e Hart (2012).

Grandeza	Símbolo	Valor
Tensão de Entrada	V_s	26,6 V
Corrente de Entrada	I_s	7,71 A
Potência de Entrada	P_s	205,08 W
Razão Cíclica	D	0,5
Frequência de Chaveamento	f	20 kHz
Ondulação de Corrente	Δ_{i_L}	30%
Ondulação de Tensão	ΔV	1%
Tensão de Saída	V_o	26,6 V

Tabela 1: Parâmetros utilizados no projeto do conversor buck-boost.

Segundo Boylestad (2012), é possível definir o valor da resistência do conversor utilizando

$$P_S = \frac{{V_S}^2}{R} = 3{,}45\ \Omega \quad (9)$$

Para Hart (2012), pode-se definir a tensão de saída do circuito através de

$$V_o = -V_s\left(\frac{D}{1-D}\right) = -26{,}6\ \text{V} \quad (10)$$

Ainda, o valor do capacitor do conversor buck-boost é obtido por meio de

$$C = \frac{D}{R\cdot(\Delta V_o / V_o) f} = 725\ \mu F \quad (11)$$

De acordo com Hart (2012) a corrente no indutor é dada por

$$I_L = \frac{V_s \cdot D}{R(1-D)^2} = 15{,}42\ A \quad (12)$$

a partir da qual se calcula a variação de corrente Δ_{i_L}. Assim

$$\Delta_{i_L} = 0{,}3 \cdot I_L = 4{,}62\ A \quad (13)$$

Com base neste valor é possível calcular, ainda de acordo com Hart (2012), a indutância do circuito

$$L = \frac{V_i \cdot D}{\Delta i_L \cdot f} = 144\ \mu H \quad (14)$$

O projeto desenvolvido garante uma potência entregue à carga igual à potência fornecida pela fonte e que pode ser calculada, de acordo com Boylestad (2012), através de

$$P_o = I_o \cdot V_o = 205{,}08\ W \quad (15)$$

2.3 Método de Perturbação e Observação (P&O)

O algoritmo de P&O (Fig. 3) baseia-se na comparação da potência de saída do sistema fotovoltaico em diferentes ciclos para realizar uma perturbação na tensão de operação ou na razão cíclica de um conversor CC-CC e assim estabelecer sua operação próxima ao MPP.

O processo começa com a leitura dos valores de tensão $V(n)$ e de corrente $I(n)$ provenientes do módulo fotovoltaico em determinado instante de tempo (n).

Com base nas medições realizadas é calculada a potência atual $P(n)$ produzida pelo sistema, que em seguida é comparada com a potência fornecida em um momento anterior $P(n-1)$.

Neste ponto, é realizado o seguinte questionamento: a potência atual $P(n)$ é maior que a potência produzida no

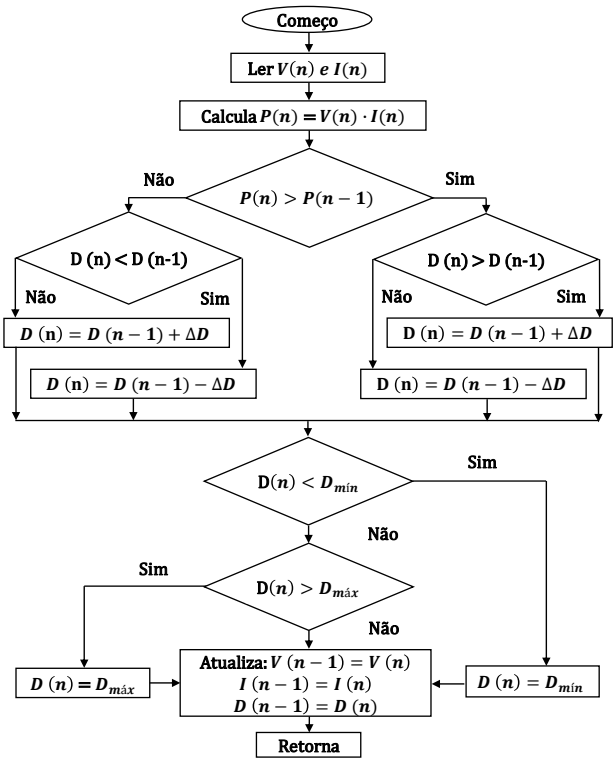


Fig. 3: Algoritmo do Método de Perturbação e Observação para Rastreamento do Ponto de Máxima Potência. Fonte: Martins *et al* (2011).

ciclo anterior $P(n-1)$? Esta comparação possibilita dois resultados. O primeiro deles é que $P(n)$ é sim maior que $P(n-1)$, o que representa uma variação de potência (ΔP) positiva de modo que a potência do sistema está aumentando. O segundo é que $P(n)$ não é maior que $P(n-1)$, o que representa que ΔP é negativa e a potência do sistema está diminuindo.

Na sequência, a razão cíclica atual $D(n)$ correspondente à $P(n)$ é comparada com a razão cíclica do conversor em um momento anterior $D(n-1)$, correspondente à $P(n-1)$. Se ΔP for positiva, é realizado o seguinte questionamento: $D(n)$ é maior que $D(n-1)$? Se a resposta for sim, é acrescentado ao valor de razão cíclica atual uma variação ΔD chamada de passo de incremento. Se a resposta for não o mesmo procedimento é realizado, no entanto, o passo de incremento é subtraído da razão cíclica anterior. Uma lógica semelhante é aplicada se ΔP for negativa, onde é analisado se $D(n)$ é menor que $D(n-1)$.

O resultado parcial é um novo valor para D do conversor, que será analisado para garantir que os valores máximos e mínimos permitidos não sejam desrespeitados. Assim, se a razão cíclica atual $D(n)$ for menor do que $D_{mín}$ o método estabelece como valor de operação o próprio $D_{mín}$ e se o valor de $D(n)$ ultrapassar $D_{máx}$, este último será estabelecido como a razão cíclica atual do conversor.

Quando o MPP é alcançado, é estabelecida uma oscilação em torno do ponto ótimo que ocorre mesmo em regime permanente e permite que a máxima potência produzida naquele instante em função dos níveis de irradiação e temperatura seja fornecida à carga.

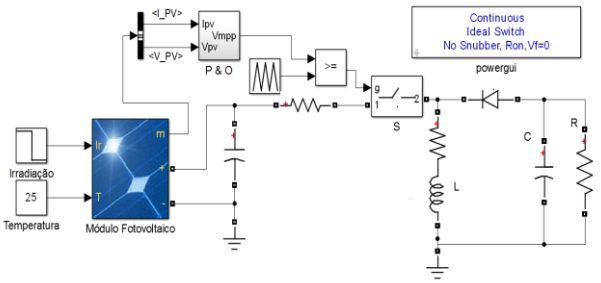


Fig. 4: Simulação do sistema de rastreamento do Ponto de Máxima Potência (MPP) no Simulink utilizando o Método de Perturbação e Observação (P&O).

3 Resultados e Discussões

Com base no desenvolvimento apresentado no item anterior foi realizada a simulação do sistema de potência conforme é ilustrado na Fig. 4. As condições ambientais de operação do painel são simuladas através das entradas T e I_r do bloco *PV Array*, que representam respectivamente uma temperatura de 25°C e a aplicação de um degrau de irradiação de 800 W/m² para 600 W/m² (Fig. 5(a)).

Deste modo, foi possível comparar a potência fornecida pelo sistema sem a aplicação de um método de rastreamento e posteriormente com a sua utilização, como é ilustrado na Fig. 5(b).

Os resultados indicam que sem a utilização do P&O foram obtidos 145,7 W e 83,46 W, respectivamente para 800 W/m² e 600 W/m². Por outro lado, com a aplicação do MPPT a potência fornecida à carga foi de 160,6 W para 800 W/m² e 123,6 W para 600 W/m². Estes valores são muito próximos à potência esperada para as condições simuladas, que são de 165,6 W para o nível mais alto de irradiação e 125 W para o menor.

Ainda, existem dois parâmetros que podem ser configurados no algoritmo de P&O. O primeiro deles é o período de amostragem (t_s), que representa o intervalo de tempo entre a leitura das variáveis provenientes do dispositivo fotovoltaico. O segundo é o passo de incremento da razão cíclica (ΔD), que estipula o tamanho do degrau rumo ao ponto de operação ótimo.

Com o objetivo de avaliar o impacto destes parâmetros no comportamento do sistema foram realizadas simulações considerando-se diferentes valores para t_s e ΔD.

A Fig. 5(c) demonstra a influência do passo de incremento da razão cíclica na dinâmica do sistema para $\Delta D = 0{,}005$ e $\Delta D = 0{,}01$. É possível perceber que quando $\Delta D = 0{,}01$ o sistema atinge o regime permanente com maior velocidade, porém, ao alcançá-lo, é estabelecida uma grande oscilação. Já para o caso em que é imposto um passo de perturbação $\Delta D = 0{,}005$ o sistema torna-se mais lento, mas ao atingir o regime permanente proporciona uma oscilação menor em torno do ponto ótimo.

A Fig. 5(d) ilustra a resposta do sistema para $t_s = 0{,}25\ ms$ e $t_s = 0{,}5\ ms$, considerando $\Delta D = 0{,}005$. Com base nos resultados obtidos percebe-se que o período

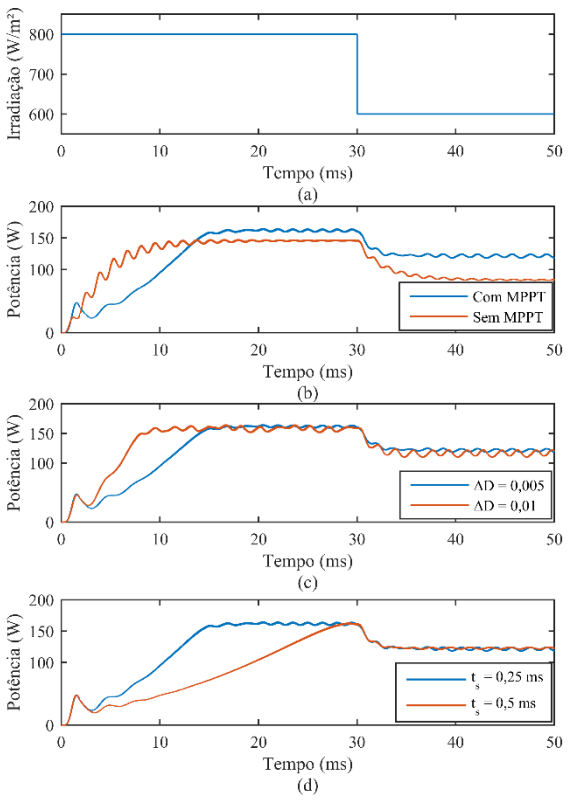


Fig. 5: (a) Degrau de irradiação de 800 W/m² para 600 W/m²; (b) Comparação entre a potência fornecida com e sem MPPT ($\Delta D = 0{,}005$ e $t_s = 0{,}25\ ms$); (c) Comparação entre as respostas obtidas para $\Delta D = 0{,}005$ e $\Delta D = 0{,}01$, considerando $t_s = 0{,}25\ ms$; (d) Comparação entre as respostas obtidas para $t_s = 0{,}25\ ms$ e $t_s = 0{,}5\ ms$, considerando $\Delta D = 0{,}005$;

de amostragem tem influência direta no tempo que o algoritmo leva para atingir o MPP. Quando $t_s = 25\ ms$ o sistema atinge este ponto por volta dos $15\ m$s. Em contrapartida, quando este valor é aumentado para $0{,}5\ m$s isto ocorre somente por volta dos $30\ m$s.

4 CONSIDERAÇÕES FINAIS

A potência fornecida por dispositivos fotovoltaicos é extremamente sensível a variações de irradiação e de temperatura de modo que é impossível que o sistema de potência opere sempre em um ponto fixo. Deste modo, são aplicados algoritmos de MPPT com o objetivo de otimizar a geração de energia, sendo que um dos métodos mais utilizados é o de Perturbação e Observação.

Neste contexto, este trabalho teve como objetivo validar o funcionamento do método de P&O no rastreamento do MPP em sistemas fotovoltaicos e avaliar o impacto de diferentes parâmetros que influenciam na dinâmica do algoritmo através de simulações desenvolvidas com o auxílio do Simulink.

A partir da metodologia utilizada, foi possível comparar a potência fornecida sem o algoritmo de P&O e posteriormente com a utilização do método. Sem a utilização do método de P&O foram obtidos 145,7 W e 83,46 W, respectivamente para 800 W/m² e 600 W/m². Já para o sistema operando com o algoritmo de rastreamento, a potência fornecida à carga foi de 160,6 W para 800 W/m² e 123,6 W para 600 W/m².

Investigou-se também o impacto de diferentes valores de período de amostragem (t_s) e passo de incremento da razão cíclica (ΔD) no comportamento dinâmico do sistema, constatando-se que estes apresentam influência direta no processo de rastreamento de modo a modificar o tempo de resposta para alcançar o MPP e a magnitude da oscilação em torno do ponto quando ele é alcançado.

A investigação realizada ressalta a importância do tema estudado e sua funcionalidade, tendo em vista a otimização da geração de energia em sistemas fotovoltaicos.

AGRADECIMENTOS

Este trabalho tem o apoio da Universidade Regional Integrada do Alto Uruguai e das Missões, Projeto 3657.

REFERÊNCIAS

BARBI, I; MARTINS, D. C; Conversores CC-CC Básicos Não Isolados. Florianópolis: Edição do Autor, 2000.

BOYLESTAD, R. L. *Introdução à Análise de Circuitos*. 12° ed. Rio de Janeiro: Prentice- Hall do Brasil, 2012.

BRITO, M. A. G., et al. *Research on Photovoltaics: Review, Trends and Perspectives.* In: Brazilian Power Electronics Conference (COBEP). p. 531-537, 2011.

DUPONT, F. H. *Estudo, Análise e Implementação de uma Metodologia para Otimização de Rendimento em Sistemas Compostos por Conversores em Paralelo.*Tese (Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Santa Maria - UFSM, Santa Maria, 2014.

ESRAM, T.; CHAPMAN, P. L.; *Comparison of Photovoltaic Array Maximum Power Point Tracking Techniques*. In: IEEE Transactions on Energy Conversion, 2007, v. 22, n. 2, p. 429-449.

FEMIA, N. *et al. Optimization of Perturb and Observe Maximum Power Point Tracking Method*. In: IEEE Transactions on Power Electronics, 2005, v. 20, n.4, p. 963-973.

HART, D. W.; *Eletrônica de Potência*: análise e projetos de circuitos. – Porto Alegre: AMGH, 2012.

MARTINS et al.; Técnicas de Rastreamento de Máxima Potência para Sistemas Fotovoltaicos: Revisão e Novas Propostas (Minicurso). *XI Congresso Brasileiro de Eletrônica de Potência*- COBEP. Natal-RN: 11 a 15 de setembro de 2011.

NATIONAL RENEWABLE ENERGY LABORATORY (NREL); *NREL System Advisor Model*, 2014. Disponível em: <https://sam.nrel.gov.> Acesso em: 14 jun. 2016.

UMA PROPOSTA PARA CRIAÇÃO DE MECANISMOS E ESTRATÉGIAS PARA MANTER RESILIÊNCIA EM CONTROLADORES SDN

A PROPOSAL TO DESIGN MECHANISMS AND STRATEGIES FOR MAINTAINING RESILIENCE IN SDN CONTROLLERS

LUCAS F. CLARO[1]*, GILNEI PELLEGRIN[1], MANUELA TIRLONI[1], CRISTIAN C. MACHADO[1]

[1]Departamento de Engenharias e Ciência da Computação, Universidade Regional Integrada do Alto Uruguai e das Missões, URI - Câmpus de Frederico Westphalen. *E-mail: lucasclaro87@gmail.com

Resumo: Redes Definidas por Software (Software-Defined Networking - SDN) é um paradigma que oferece uma forma mais dinâmica, administrável e adaptável para manipular fluxos de tráfegos. SDN é capaz de identificar e adaptar-se rapidamente às mudanças dos requisitos dos serviços de rede. Apesar dos benefícios na utilização de SDN, centralizar decisões sobre o tráfego no elemento controlador torna SDN um ambiente propicio para ataques. Entre esses ataques está o de negação de serviço que podem levar o controlador a apresentar diversos problemas, tais como atrasos na comunicação ou tornar-se incapaz de atender requisições de fluxos, causando inconsistências ou perda de dados, além de falhas no gerenciamento de novas regras ou, até mesmo, interrupção dos serviços. Neste sentido, este artigo apresenta uma proposta para desenvolver mecanismos e estratégias para detectar e analisar as requisições feitas a controladores com a finalidade de identificar se estas são de tráfego real legitimo ou são provenientes de um ataque e, de acordo com esta análise, decidir ações para mitigar cada ataque.

Palavras-chave: Ataques. Detectar. Mitigar. Redes Definidas por Software.

Abstract: Software-Defined Networking (SDN) is a paradigm that offers a more dynamic, manageable, and adaptable way for handling traffic flows. SDN is capable to identify and adapt quickly for requirement changes of network services. Despite the SDN benefits, the act to centralize decisions about traffic flows in controller devices makes SDN a suitable environment for different types of attacks. Among these attacks are the denials of services (DoS) that can lead SDN controllers to present several problems, such as delays in communication or become unable to meet traffic flow requests, by causing inconsistencies or packet loss, as well as failures in the management of new rules or even service disruptions. In this context, we present in this paper a proposal to develop mechanisms and strategies to detect and analyze the controller requests in order to identify whether these requests are legitimate or illegitimate (an attack), and by according to this analysis, deciding actions to mitigate each attack.

Keywords: Attacks. Detect. Mitigate. *Software-Defined Networking.*

1 INTRODUÇÃO

O paradigma de redes definidas por *software* (*Software-Defined Networking* - SDN) surgiu com o objetivo de fornecer uma arquitetura aprimorada para o gerenciamento e monitoramento de tráfego de rede.

Para alcançar isso, SDN apresenta o desacoplamento entre o plano de controle e plano de dados, que em redes IP tradicionais encontram-se centralizados nos dispositivos de encaminhamento, tais como *switches* e roteadores. Essa nova abordagem proporciona a possibilidade de programar esses dispositivos de maneira centralizada, num elemento denominado controlador de rede. O controlador passa a tomar a decisão lógica sobre os fluxos de dados que trafegam na rede, instalando regras nos *switches* e roteadores, indicando o comportamento esperado de cada fluxo. Como resultado, o controlador apresenta diversos benefícios como, por exemplo, a capacidade de ter uma visão geral da infraestrutura de rede, tais como, *switches*, links, rotas e o conhecimento de características da mesma, tais como, largura de banda disponível, quantidade de saltos de uma rota, *delay* de uma rota.

O uso de SDN tem apresentado diversos benefícios para alcançar resiliência de redes e de seus serviços. Entretanto, poucos esforços têm sido direcionados para explorar quão resilientes controladores de rede são e, principalmente, quais ações devem ser realizadas ao identificar que o controlador esta enfrentando problemas de resiliência.

Neste contexto, este artigo apresenta um projeto que propõe um conjunto de estratégias e mecanismos que visam melhorar a resiliência de controladores. Uma das estratégias que este projeto almeja é o uso de *honeypots* com o intuito de enganar o atacante. Através de um *honeypot*, uma cópia semelhante do controlador poderá assumir as solicitações/ações do atacante para que o mesmo considere o resultado das mesmas como sucesso. Assim, o controlador poderá continuar atendendo às solicitações reais de tráfego sem sofrer degradações. Além da contribuição técnica, esse projeto tende a agregar conhecimento aos pesquisadores envolvidos e produção científica ao grupo de pesquisa.

O restante deste artigo está dividido da seguinte maneira: Na seção 2 será detalhada a contextualização geral deste projeto. Na seção 3 uma visão geral da proposta será apresentada. Por fim, os resultados esperados e uma conclusão serão apresentados na seção 4.

2 CONTEXTUALIZAÇÃO

Redes Definidas por Software (*Software-Defined Networking* – SDN) é uma arquitetura de rede dinâmica, adaptável, controlável, e flexível para a entrega de serviços de rede, capaz de responder rapidamente às mudanças de requisitos de serviço (Open Networking Foundation et al. 2014; MONSANTO et al. 2013). Uma arquitetura SDN compreende quatro planos: controle, dados, aplicação e gerenciamento (Open Networking Foundation et al. 2014). O plano de controle é responsável pelos protocolos e pela tomada de decisões que resultam na atualização das tabelas de encaminhamento dos *switches* e roteadores. O plano de dados, conhecido como o plano de encaminhamento, administra a comutação e roteamento de pacotes de fluxo. O plano de aplicação inclui aplicações SDN (por exemplo, firewalls e balanceadores de carga), aplicações de negócios (por exemplo, portais *e-commerce* e sistemas de gestão empresarial) ou sistemas de Orquestração de Nuvens (por exemplo, OpenStack e CloudStack). Cada aplicativo tem controle exclusivo de um conjunto de recursos fornecidos pelos controladores SDN. O plano de gerenciamento inclui sistemas de gestão que exercem as funções e operações de apoio à infraestrutura, por exemplo, acordos de nível de serviço (*Service Level Agreements* – SLAs) e as políticas de baixo nível para conduzir aplicações e controladores SDN.

Em redes IP tradicionais, o plano de controle é executado em cada dispositivo de rede. Cada dispositivo tem seus protocolos proprietários, o que torna difícil sua programação. Muitas vezes não é possível realizar o processo de tomada de decisão sobre eventos que não tenham sido previstos. Diferentemente, SDN é caracterizado por um plano de controle logicamente centralizado, o que permite que parte da lógica de tomada de decisão realizada pelos dispositivos de rede seja movida para controladores externos. Essa abordagem fornece aos controladores a capacidade de ter uma visão global da rede e seus recursos, tornando-se cientes de todos os elementos da rede e suas características. Com base nesta centralização, dispositivos de rede tornam-se simples elementos de encaminhamento de pacotes, podendo ser programados através de uma interface aberta, como o protocolo OpenFlow (MACHADO et al. 2015; NUNES et al. 2014).

O OpenFlow é um protocolo aberto que permite o desenvolvimento de mecanismos programáveis com base em uma tabela de fluxo padrão localizada nos diferentes dispositivos de encaminhamento. O protocolo OpenFlow estabelece um canal de comunicação seguro entre os switches e o controlador, usando este canal para controlar e estabelecer fluxos de acordo com programas customizáveis (MCKEOWN *et al.* 2008). Resumidamente, os principais elementos de uma arquitetura SDN baseada em OpenFlow são: (i) uma tabela de fluxo em cada switch contendo entradas para cada fluxo, (ii) um controlador que executa programas personalizados para decidir quais regras e ações vão ser instaladas para controlar o encaminhamento de pacotes em cada elemento de comutação, e (iii) uma camada de abstração que se comunica de forma segura com um controlador relatando sobre novos fluxos de entrada que não estão presentes na tabela de fluxo. Cada entrada na tabela de fluxo consiste em: (a) uma máscara de campos encontrados no cabeçalho do pacote, a qual é utilizada para combinar os pacotes que chegam, (b) contadores que armazenam estatísticas para cada fluxo específico, tal como o número de *bytes*, o número de pacotes recebidos, e duração de um fluxo, e (c) uma série de ações a serem executadas quando um pacote corresponde a uma máscara registrada (Open Networking Foundation et al. 2014).

2.1 Resiliência em redes de computadores

Resiliência em redes é uma propriedade que diz respeito à capacidade da rede e dos seus serviços em manter níveis aceitáveis de funcionamento frente a anomalias (SMITH et al. 2011). Tais anomalias podem surgir devido a problemas de configuração de equipamentos e serviços, ataques maliciosos, sobrecarga operacional, falhas de comunicação e de equipamentos. Como resultado, esses problemas podem gerar atrasos na comunicação, perda de dados, indisponibilidade de serviços, apresentando desconforto aos usuários que dependem ou buscam usufruir de tais serviços. Resiliência pode ser alcançada quando utilizadas estratégias e mecanismos, tais como, detecção de ataques e anomalias, como, por exemplo, sistemas de detecção de intrusão e sistemas de monitoramento de largura de banda; correção dessas questões, por exemplo, modelagem de tráfego e balanceamento de carga (IZADDOOST et al. 2014).

Sterbenz et al. (2010) apresentam uma taxonomia para resiliência em redes compostas por dois principais grupos de disciplinas, *Challenge Tolerance* e *Trustworthiness*. A Figura 1 apresenta as disciplinas relacionadas por grupo.

A relação entre os dois principais grupos é a robustez, ou seja, o desempenho e a confiabilidade de um sistema quando perturbado. A seguir, os principais grupos e suas disciplinas são resumidamente apresentados.

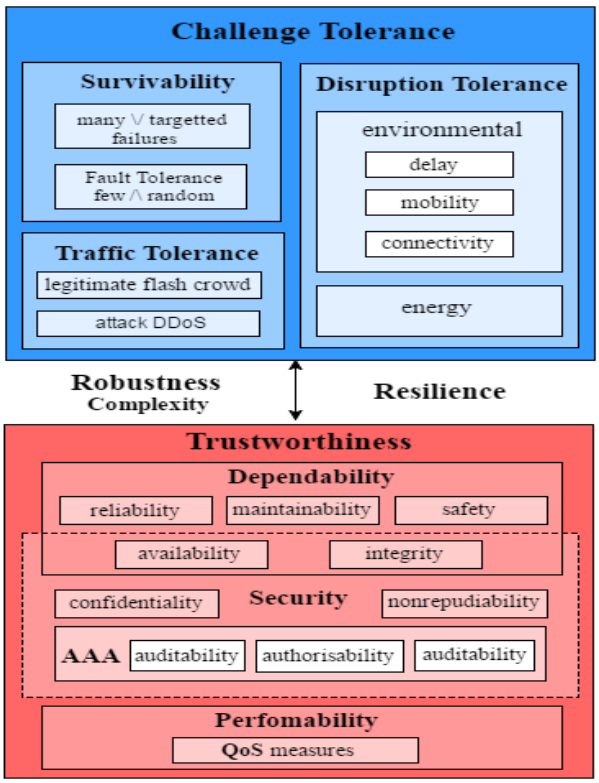


Fig. 1. Taxonomia de resiliência (Adaptada de STERBENZ *et al*. 2010).

2.1.1 Challenge tolerance

No grupo *Challenge Tolerance* encontram-se as disciplinas que tratam do design e da engenharia de sistemas que se mantem prestando seus serviços frente a desafios e mudanças de rede. Esse grupo está subdividido em outros 3 grupos: *Survivability, Disruption Tolerance e Traffic Tolerance.*

2.1.1.1 Survivability

Survivability está relacionada com a capacidade que um sistema possui para cumprir seu objetivo em tempo hábil, mesmo com a presença de anomalias ou na ocorrência de desastres naturais. Uma característica importante relacionada à *Survivability* é a redundância na elaboração de um sistema. A capacidade de *Survivability* requer redundâncias em diversos níveis, tanto para lógico quanto para físico, de modo que o mesmo recurso não seja partilhado por partes do sistema que podem sofrer falhas correlacionadas (IZADDOOST et al. 2014).

2.1.1.2 Disruption tolerance

Um desafio encontrado em redes de comunicação está relacionado à capacidade de manter conexões estáveis. *Disruption Tolerance* é a capacidade de um sistema (ou determinado recurso) de tolerar interrupções de comunicação entre seus componentes. Tais interrupções podem ser em nível ambiental (disposições físicas que os componentes estão submissos) ou energético (questões referentes a alimentação dos componentes).

Existem três principais contribuintes para o campo de tolerância a interrupções. O primeiro é motivado pelo comportamento de redes dinâmicas em que a conectividade entre os membros está mudando continuamente. A segunda contribuição foi motivada por grandes atrasos que protocolos de rede tradicionais não podem tolerar. A terceira contribuição é relacionada às redes com restrições de energia onde os nós que fazem parte da topologia acabam ficando inativos devido ao término de energia de suas baterias, não podendo contribuir para a conectividade de rede (IZADDOOST et al. 2014; STERBENZ et al. 2010).

2.1.1.3 Traffic tolerance

Traffic Tolerance é a capacidade que um recurso tem para suportar determinado tipo ou volume de tráfego, mesmo os não previstos, bem como a habilidade de isolamento de fluxos, a comunicação entre os nós e componentes da estrutura. Um dos desafios referentes a *Traffic Tolerance* está relacionado com identificação de tráfego legítimo e tráfego malicioso, como por exemplo, rajadas de solicitações para determinado serviço e ataques de negação de serviço distribuído (*Distributed Denial of Service* - DDoS). Indiferente desta identificação, o que deve ser levado em conta é que existe uma necessidade clara de implementação de mecanismos de monitoramento e detecção de tráfego para que recursos da rede não sejam afetados a ponto de indispor seus serviços (STERBENZ et al. 2010).

2.1.2 Trustworthiness

No grupo *Trustworthiness* encontram-se as disciplinas que descrevem propriedades mensuráveis de serviços e sistemas resilientes. *Trustworthiness* é definido como a garantia de que um sistema executará suas funções como esperado. Esse grupo está subdividido em outros 3 grupos: *Dependability*, *Security* e *Performability*.

2.1.2.1 Dependability

Dependability é a disciplina que aponta a dependência que certo componente, recurso ou serviço possui na estrutura, i.e., define como um componente pode afetar outro no sistema (LAPRIE et al. 1992). *Dependability* é dividida em cinco disciplinas: *Reliability, Maintainability, Safety, Availability* e *Integrity*, sendo as duas últimas, disciplinas que compartilham características com o subgrupo de disciplinas relacionadas com *Security*.

Reliability indica como o recurso continuará sendo oferecido de forma correta, frente a anomalias. *Maintainability* representa a possibilidade da rede em sofrer mudanças ou reparos objetivando melhorias. *Safety* representa a habilidade da rede em evitar ou ao menos amenizar os resultados apresentados por uma falha ou erro para que não se propaguem de forma prejudicial para a estrutura, sistema ou usuários. *Availability* relaciona-se com as propriedades que um componente possui em estar pronto quando solicitado. Por fim, *Integrity* define o

estado que a rede ou a informação que nela se encontra possui em não sofrer alterações ou degradações.

2.1.2.2 Security

Nas disciplinas de *Security* são abordadas as questões referentes ao monitoramento, identificação, prevenção de acessos não autorizados, má utilização, alteração e negação dos recursos da rede. *Security* é dividida em sete disciplinas: *Confidentiality*, *Nonrepudiability*, *Auditability*, *Authorisability*, *Authenticity*, *Availability* e *Integrity*, estas duas últimas, já descritas anteriormente.

Confidentiality é a propriedade que indica que o recurso não está disponível ou revelado a indivíduos, entidades ou processo não autorizados. *Nonrepudiability* indica que o recurso não pode negar a participação ou envolvimento em um determinado fato ocorrido na rede envolvendo sua identificação comprovada. *Auditability* apresenta-se com a ideia de que a rede pode (e deve) ter a possibilidade de ser auditada se e quando necessário. *Authorisability* indica os privilégios de acesso concedido a um recurso. *Authenticity* comprova que o recurso, por exemplo, serviço, comunicação é válido/verdadeiro e confiável (STERBENZ et al. 2010).

2.1.2.3. Performability

Performability se refere à capacidade de um recurso de alcançar os requisitos de Qualidade de Serviço *(Quality of Service* - QoS) exigidos. Nessa disciplina são apresentadas medidas, tais como, *delay* e largura de banda disponível, que indicam as necessidades de serviços, links, ou usuários na rede para um funcionamento ideal.

Neste contexto, Sterbenz et al. (2010) apresentam cinco princípios necessários para construir um sistema resiliente:

- Necessidades dos serviços a fim de determinar o nível de resiliência que o sistema deve fornecer para seu funcionamento.
- Comportamento normal da rede através da combinação de especificações de parâmetros, juntamente com o monitoramento dos mesmos para aprender sobre as operações normais da rede.
- Modelos de ameaças e desafios que servem como bases essenciais para compreender, definir e implementar mecanismos de resiliência afim de se defender, detectar e corrigir anomalias na rede.
- Métricas analisando as necessidades de serviço e o estado operacional para medir detectar, corrigir e quantificar a resiliência a fim de refinar comportamentos futuros.
- Heterogeneidade no mecanismo, confiança e política visando atender todos os cenários à medida que novos comportamentos e questões não previstas ocorram.

3 PROPOSTA

Este artigo apresenta a proposta de desenvolver um conjunto de mecanismos e estratégias cujo objetivo principal é aliviar problemas de resiliência em controladores SDN utilizando os recursos e benefícios oferecidos por redes definidas por *software*.

Primeiramente será feito o planejamento, definindo o escopo dos mecanismos e estratégias e seus requisitos e a modelagem do banco de dados de acordo com as características do problema.

Em um segundo momento, será desenvolvido um sistema de coleta de informações da rede para melhorar o processo de análise de resiliência em controladores. Dentre as informações coletas encontram-se as máscaras geradas por cada anomalia, que possuem informações tais como portas de origem e destino, protocolos, prioridades, quantidade de pacotes, tempo de duração do fluxo, entre outras.

Em seguida será feita a análise de mecanismos e estratégias existentes para a resiliência em dispositivos que fazem/faziam tomadas de decisão em redes IP tradicionais, tais como, *switches* e roteadores, e desenvolver melhorias e adaptações para utilização em controladores baseadas nas características oferecidas por redes definidas por *software*.

Logo após será desenvolvido um sistema de decisão e aplicação de estratégias que se baseará no contexto da anomalia. O sistema terá uma interface amigável para monitorar anomalias e para apresentar cada decisão tomada, além da flexibilidade para apresentar uma possível alteração de estratégia.

Por fim, será realizada a avaliação do desempenho e precisão dos sistemas através da comparação com um conjunto de problemas-testes e modificação de cenários, visando mostrar os resultados propostos neste projeto.

4 CONCLUSÃO E RESULTADOS ESPERADOS

Este projeto tem a finalidade de analisar e desenvolver diversas estratégias e mecanismos para resiliência de controladores em redes definidas por *software* conforme mencionado na primeira seção deste projeto (Introdução). Todos os resultados do projeto poderão servir de base para terem aplicabilidade da solução em ambientes reais como o de operadores de Telecom, provedores de Internet, computação em nuvem, dentro outros. Acredita-se que esta pesquisa será de grande valia, pois com o fato de ser possível aplicar a solução para resolver problemas reais, ela vai gerar economia de recursos, tanto físicos quanto humanos, e vai proporcionar ferramentas e sistemas mais eficientes.

AGRADECIMENTOS

Agradecemos ao Programa Institucional de Bolsas de Iniciação Científica do Edital/PROPEPG/PIIC/URI N° 03/2016 que auxilia essa pesquisa através do projeto #3609 - Mecanismos e estratégias para resiliência de controladores em redes definidas por software.

REFERÊNCIAS

Open Networking Foundation. *SDN architecture*. June 2014. Available at:<https://www.opennetworking.org/sdn-resources/technical-library>. Acesso em: 1º jun. 2016.

MONSANTO, C. *et al*. *Composing software-defined* Conference on Networked Systems Design and Implementation. Berkeley, CA, USA: USENIX Association, 2013. (nsdi'13), p. 1–14. Disponível em: <http://dl.acm.org/citation.cfm?id=2482626.2482629>. Acesso em: 04 ago. 2016.

MACHADO, C. C.; GRANVILLE, L. Z.; SCHAEFFER-FILHO, A.; WICKBOLDT, J. A., *Towards SLA Policy Refinement for QoS Management in Software-Defined Networking*. Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on, vol., no., pp.397-404, 13-16 Maio 2014.

NUNES, B. *et al. A survey of software-defined networking: Past, present, and future of programmable networks.* Communications Surveys Tutorials, IEEE, v. 16, n. 3, p.1617–1634, Fevereiro 2014. ISSN 1553-877X.

MCKEOWN, N. *et al*. Openflow: *Enabling innovation in campus networks. SIGCOMM* Comput. Commun. Rev., ACM, New York, NY, USA, v. 38, n. 2, p. 69–74, março. 2008. ISSN0146-4833.

networks. In: Proceedings of the 10th USENIX

SMITH, P.; SCHAEFFER-FILHO, A.; HUTCHISON, D.; MAUTHE, A. *Management patterns: SDN-enabled network resilience management.* In: Network Operations and Management Symposium (NOMS), 2014 IEEE, 2014. Anais. . . [S.l.: s.n.], 2014. p.1–9.

STERBENZ, J. P.; HUTCHISON, D.; ÇETINKAYA, E. K.; JABBAR, A.; ROHRER, J. P.; SCHÖLLER, M.; SMITH, P. *Resilience and survivability in communication networks: strategies, principles, and survey of disciplines.* Computer Networks, [S.l.], v.54, n.8, p.1245 – 1265, 2010. Resilient and Survivable networks.

IZADDOOST, A.; HEYDARI, S. S. Proactive risk mitigation for communication network resilience in disaster scenarios. In: *A World of Wireless, Mobile and Multimedia Networks* (WOWMOM), 2014 IEEE 15th International Symposium On, 2014. Anais. [S.l.: s.n.], 2014. p.1–4.

LAPRIE, J. C.; AVIZIENIS, A.; KOPETZ, H. (Ed.). *Dependability: basic concepts and terminology.* Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1992

.

MPLS FAST RE-ROUTE: CONTEXTUALIZAÇÃO E VISÃO GERAL

A LITERATURE REVIEW OVER MULTI-PROTOCOL LABEL SWITCH FAST RE-ROUTE (MPLS FRR)

MATEUS VICTORIO ZAGONEL[1]*, JUCIMAR RODRIGUES[1], CASSIANO MÔNEGO[1]

[1]Departamento de Engenharias e Ciência da Computação, Universidade Regional Integrada do Alto Uruguai e das Missões, URI - Câmpus de Frederico Westphalen.
*E-mail: mateuszagonel@hotmail.com.

Resumo: Com a expansão das redes de comunicação e o consequente aumento no uso destas, criou-se uma demanda por disponibilidade e qualidade de serviço. Dentro deste contexto uma forma de diminuir falhas em redes e garantir tais características se dá com o uso de MPLS FRR. Esse protocolo consiste em um serviço de reencaminhamento rápido de pacotes por túneis de proteção que são predefinidos para falhas na rede. O presente artigo tem como objetivo apresentar fundamentos de MPLS FRR, e técnicas de *backup*, bem como um comparativo de uma Rede que usa este protocolo em relação a uma rede IP tradicional, apresentando as vantagens deste protocolo para recuperação de falhas em enlaces ou nós da rede, de forma a garantir qualidade de serviço.

Palavras-chave: MPLS FRR. Protocolo de Encaminhamento. Reencaminhamento de Pacotes. Proteção Local.

Abstract: With the expansion of communication networks and the consequent increase in the use of these it created a demand for availability and service quality. Within this context, a way to reduce failures in networks and secure these characteristics is given to the use of MPLS FRR . This protocol consists of a fast forwarding service packs protection of tunnels that are preset for network failures. The present article aims to present fundamentals of MPLS FRR, and backup techniques, as well as a comparison of a network that uses this protocol in relation to a traditional IP network, presenting the advantages of this protocol for disaster recovery links or network nodes to the network, in order to ensure QoS.

Keywords: MPLS FRR. Protocol. Forwarding Packets. Local Protection.

1 INTRODUÇÃO

Com a popularização da Internet e a crescente utilização das redes de comunicação, criou-se uma demanda por serviços que tenham disponibilidade e baixo tempo de resposta. A busca por qualidade de serviço e diminuição da latência da rede é um dos desafios para as redes de comunicação, pois a Internet não foi uma rede projetada para tantos dispositivos como se tem hoje e sim para atender a uma rede universitária entre dois *câmpus* nos EUA (MENDONÇA et al 2012). Dessa forma têm-se vários estudos acadêmicos com o objetivo de melhorar os algoritmos e protocolos de encaminhamento em dispositivos de rede a fim de tornar o fluxo de dados mais rápido e com maior disponibilidade.

Dentro desse contexto, de encaminhamento de pacotes, inclui-se o MPLS FRR, que consiste em um protocolo de proteção da rede para garantir disponibilidade em tempo real. Com o uso deste protocolo, em caso de queda de um enlace ou de um nó, os pacotes são encaminhados por um caminho alternativo sem interromper a comunicação entre os dispositivos da rede. Tal caminho é definido antes de ocorrer o problema na rede, por essa razão este protocolo é eficiente, uma vez que em alguns milissegundos ele consegue redirecionar os pacotes.

A utilização e criação de tal protocolo se fez necessária pela crescente demanda por *QoS* (*Quality of Service)* que consiste em serviços que necessitam de uma grande quantidade de largura de banda sem interrupção. Um exemplo de um serviço que demanda de *QoS* é o de VoIP, pois este precisa operar em tempo real sem grande latência ou interrupção para que possa funcionar normalmente e os usuários consigam se comunicar.

Diante dessa perspectiva de busca por disponibilidade em redes e qualidade de serviço, o presente artigo tem como objetivo estudar fundamentos de MPLS FRR, de forma a entender seu funcionamento, apresentar seus benefícios e modos de criar rotas de backup dentro de uma rede e realizar um comparativo entre uma rede que utiliza MPLS FRR *versus* uma rede IP Tradicional, em se tratando do tempo de recuperação da comunicação em caso de falhas.

O artigo segue a seguinte organização. Na seção 2 são apresentados conceitos fundamentais de MPLS FRR, exemplos de funcionamento e a problematização que fez com fosse necessária a utilização deste protocolo. Na seção 3 é apresentado um comparativo entre uma rede IP tradicional e uma rede MPLS. Na seção 4 é apresentado um aperfeiçoamento de *Fast Re-Route*, utilizando largura de banda compartilhada. Por fim, na seção 5, são feitas as considerações finais dos autores com relação aos benefícios desse protocolo de encaminhamento.

2 FUNDAMENTOS MPLS FRR

Nesta seção, serão tratados fundamentos de MPLS, definições, funcionamento e a problemática para utilização deste protocolo.

2.1 MPLS FRR (Multi Protocol Label Switch Fast Re-Route)

Consiste em um protocolo de reencaminhamento de pacotes em caso de falhas na rede. As falhas podem ser consideradas um enlace rompido ou um nó inativo na rede. O MPLS Fast Re-Route é uma ferramenta integrante do protocolo MPLS-TE (Engenharia de Tráfego) e tem o objetivo definir rotas alternativas antes de ocorrerem falhas (MENDONÇA et al 2012). Dessa forma os caminhos são definidos previamente e não quando uma falha realmente ocorrer. Existem protocolos que refazem cálculos de rotas, porém estes não conseguem atender a serviços que necessitam da rede em tempo real para funcionar como é o caso de VoIP, ou jogos *online*. Com o uso de MPLS *Fast Re-Router*, em caso de falha este deve reencaminhar o fluxo da rede por um túnel de backup em 10 milissegundos. Com isso atrasos não são perceptíveis garantindo a comunicação da rede (PAN et Al, 2005; WANG et Al 2012; ZHAO et al, 2013).

2.2 Problemática, Funcionamento e Utilização de MPLS FRR.

No roteamento IP tradicional quando um pacote passa por um nó da rede, o roteador deve buscar em sua tabela qual o destino de tal pacote. Para calcular os caminhos e preencher as tabelas de roteamento de cada nó da rede, são usados os protocolos OSPF e IS-IS. Porém o principal problema é que protocolos convencionais não levam em conta o tráfego da rede e sim apenas buscam identificar qual o menor caminho independente deste estar livre ou congestionado. Dentro deste contexto o protocolo MPLS-TE FRR tem objetivo analisar o tráfego da rede, por essa razão "Engenharia de Tráfego", e criar caminhos de *backup* para nós, ou para enlaces da rede, que sejam pontos com potencial de falha (NAVEED et al 2012).

Como citado, MPLS-TE FRR é usado para proteger:
Enlaces – A Fig.1 apresenta uma rede qualquer em que o fluxo normal corresponde a R1, R2, R3, R4. O enlace protegido em questão corresponde a ligação R2-R3

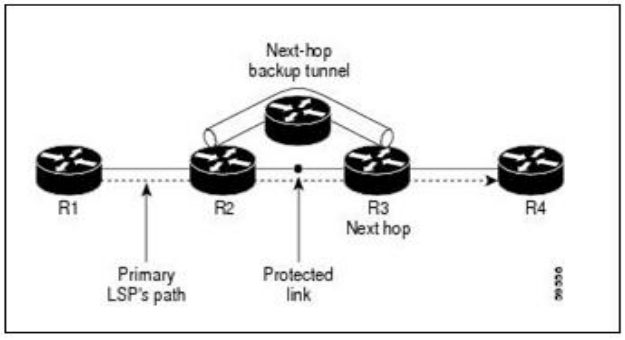


Fig. 1. Exemplo de Proteção de link

Na Fig. caso o enlace R2-R3 seja rompido, é utilizado o *Next-hop backup tunnel* que corresponde à proteção local do enlace. NHOP ou *Next-Hop* corresponde ao túnel de *backup* que ignora uma ligação, em outras palavras que efetua um salto correspondente a um enlace da rede. Com o rompimento do enlace, também é enviada uma mensagem "Path-err" para o último roteador da rede de forma a notificar este para que seja criado um sinal de aviso da utilização do novo caminho (NAVEED et al 2012).

Nós – Para proteção de Nós da rede é usado *Next-Next hop* (NNHOP) *backup tunnel* que corresponde a dois saltos na rede. Conforme se pode visualizar na Fig.2 o enlace R2-R3 está protegido e o nó R3.

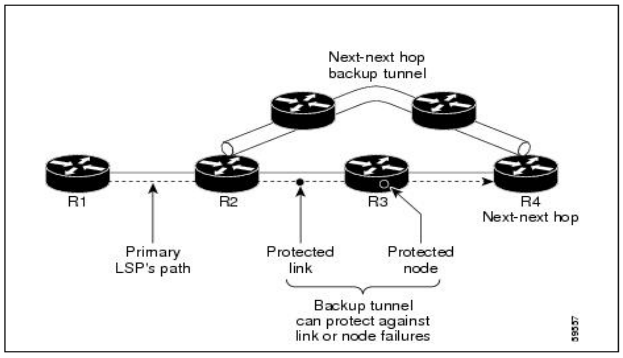


Fig. 2. Exemplo de proteção de Nó

Caso o nó R3 seja rompido, o tráfego da rede deve ser reencaminhado pelo NNHOP *backup tunnel*, sendo redirecionado em R2 o tráfego da rede pelo túnel de *backup* e retornado em R4, conforme é apresentado na Fig.2.

2.3 Técnicas de Reparo Local

Existem dois métodos de *Backup* local: *One-to-One* (1 para 1) ou então *Facility Backup* (Backup Facilidade). Resumidamente o método *One-to-One* apresenta um ponto de reparo Local para cada enlace da rede. Já o método de *Facility Backup* cria um túnel comum de *backup* para os nós da rede. Ambos serão explicados em seguida com mais detalhes.

- *One-to-One Backup* – Neste método de *backup* também chamado de 1 para 1 é estabelecido um túnel de *backup* para cada nó da rede, conforme Fig. 3.

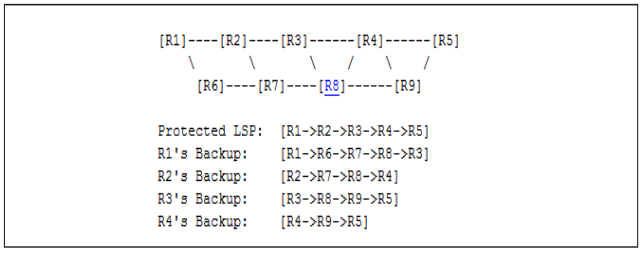


Fig. 3. Exemplo de Backup One-to-One.

Na Fig. 3 tem-se uma rede qualquer em que os nós R1, R2, R3, R4, R5 representam o caminho normal do fluxo da rede. Para entender o funcionamento desse tipo de *backup* em um caso hipotético, se o enlace entre R1 e R2 for rompido ou então se o nó R2 deixar de funcionar,

automaticamente, por definição prévia, o fluxo de pacotes passará a utilizar o túnel de *backup*. Dessa forma este passará de R1, R2, R3, R4, R5 para R1, R6, R7, R8, R3, R4, R5. Outro ponto importante é que todos os *Switchs* da rede em questão apresentam um ponto de *backup*. Ainda em se tratando da referida figura há a descrição de quais os nós estão protegidos, em "*Protected LSP*", no exemplo em questão todos os nós da rede.

- *Facility Backup* – Neste método ao contrário do *One-to-One* não é criado um túnel de *backup* para cada *switch* e sim é criado um túnel de *backup* comum que atende a um conjunto de nós da rede. Esse é denominado de túnel de desvio, na Fig. representado por "*Bypass LSP Tunnel"*. O túnel de *backup* em algum ponto da rede deve "cruzar" o fluxo, no exemplo em questão é feito isso em R2 e R4.

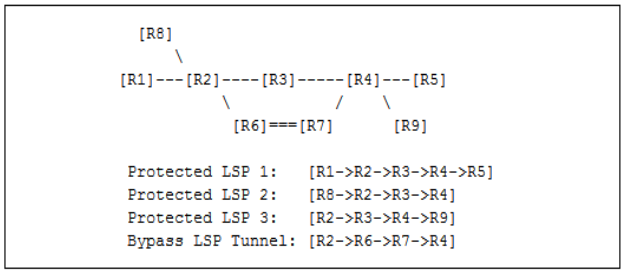


Fig. 4. Exemplo de Facility Backup

Na Fig. tem-se um exemplo de *Facility Backup* em que um túnel protege o fluxo da rede de uma falha no enlace entre R2 e R3, e/ou então de uma falha no nó R3. Esta técnica permite uma maior escalabilidade, pois não importam quantos nós existam entre R2 e R4, no exemplo em questão existe apenas R3, mas se existissem outros não haveria mudanças e o túnel de backup iria garantir o fluxo. Em caso hipotético se ocorresse uma falha no enlace entre R2 e R3, o tráfego seria redirecionado para o enlace R2, R6. Dessa forma o fluxo deixaria de ser de R1, R2, R3, R4, R5 para R1, R2, R6, R7, R4, R5 (PAN et al 2005). O método *Facility Backup* é utilizado para proteger um determinado ponto com potencial de falha da rede (WANG et al 2008; ZHAO et al 2013).

3 COMPARATIVO ENTRE REDE IP E REDE COM MPLS FRR

Nesta seção serão apresentados os resultados obtidos de um comparativo, realizado em Naveed et al (2012), entre uma rede tradicional IP utilizando os protocolos OSPF e IS-IS em relação a uma Rede MPLS *Fast Re-Route*. As simulações foram baseadas em uma topologia comum de rede que pode ser vista na Fig.. A rede é composta por 8 roteadores e 11 enlaces. As comparações são feitas com base em: Perda de Pacotes, Pacotes recebidos e tempo de ida e volta de um pacote (*Round Trip Time – RTT).*

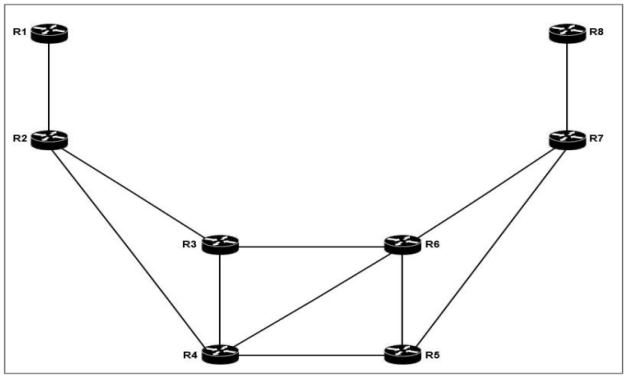


Fig. 5. Topologia da Rede para Comparativo IP versus MPLS.

Na Fig. o Caminho da rede com MPLS a ser considerado como fluxo normal é R2, R3, R4, R5, R7. O túnel de *backup* para o *link* R4-R5 é o caminho R2, R3, R4, R6, R5, R7. O túnel de *backup* para falha do nó R5 consiste no seguinte caminho: R2, R3, R4, R6, R7.

4 RESULTADOS OBTIDOS DA FALHA DO ENLACE R4-R5

A partir da falha no enlace R4-R5, foram enviados 50 pacotes de forma a obter resultados de perda de pacotes com o uso de MPLS em relação a uma rede IP tradicional. A rota considerada foi R1, R2, R3, R4, R6, R5, R7, R8. Dos 50 pacotes não houve nenhuma perda apesar da mudança do enlace, conforme pode-se visualizar na Fig..

No teste em Rede IP convencional, sem o uso de FRR. R1 traçou uma rota até R8. A rota considerada foi R1, R2, R3, R6, R7, R8. Esta rota foi criada levando em conta a falha entre o enlace R4-R5. Com a rede IP teve-se apenas a entrega de 30% dos 50 pacotes enviados, ou seja, apenas 15 pacotes foram entregues ao seu destino, conforme é apresentado na Fig..

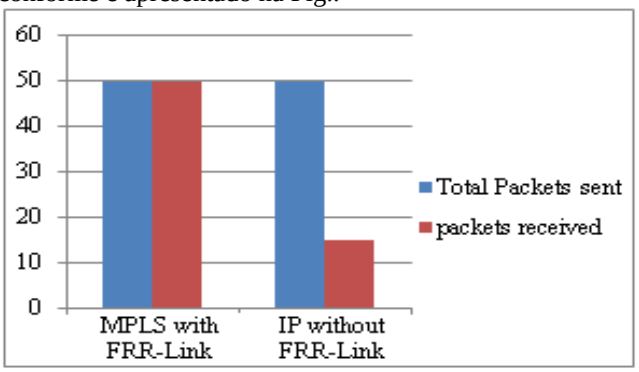


Fig. 6. Gráfico Comparativo Rede IP versus MPLS – Falha de link R4-R5

4.1 Resultados obtidos da Falha do nó R5

Considerado uma falha do nó R5 o caminho da rede com MPLS passa a ser: R1, R2, R3, R4, R6, R7, R8. O ponto de reparo consiste no enlace R4-R6, desviando do nó R5. Com o uso de MPLS FRR, foram enviados 50 pacotes de R1 até R8, como o nó R5 estava inativo, ocorreu também uma falha no enlace R4-R5. Apesar desta falha, foram entregues 49 pacotes ao destino, obtendo índice de 98% de entrega, conforme representado na Fig. .

Em uma rede IP tradicional, com a falha do nó R5, foram enviados 50 pacotes a fim de identificar o comportamento da rede. Como resultado apenas 24% chegaram ao destino, ou seja, 12 pacotes, conforme é apresentado na Fig. .

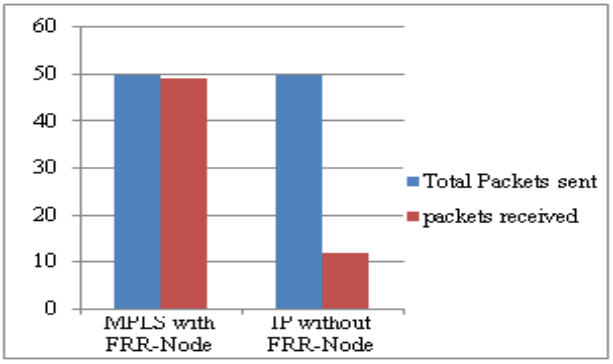


Fig. 7- Gráfico Comparativo Rede IP versus MPLS – Falha de nó R5

4.2 Tempo de Envio e Confirmação de Recebimento (RTT)

Na Fig. é apresentada a comparação do tempo de envio e confirmação da chegada do pacote ao destino em Rede IP tradicional e MPLS FRR, considerando a falha do enlace R4-R5. O tempo mínimo, médio e máximo da confirmação do envio e resposta do recebimento em Rede IP foi, respectivamente: 696ms, 849ms e 996ms. Já com MPLS FRR o tempo mínimo, médio e máximo foi, respectivamente: 468ms, 749ms e 968ms.

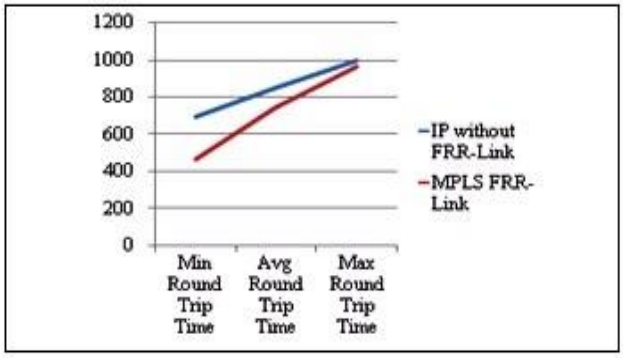


Fig. 8. Comparativo IP versus MPLS – Round Trip Time falha de Enlace.

Em outro teste considerando a falha do nó, R5. O tempo mínimo, médio e máximo da confirmação do envio e resposta do recebimento em Rede IP foi, respectivamente: 360ms, 621ms e 888ms. Já com MPLS FRR o tempo mínimo, médio e máximo foi, respectivamente: 396ms, 498ms e 596ms (NAVEED et al, 2012).

Percebe-se que o *Round-Trip-Time* teve uma taxa maior quando da falha de um nó da rede em relação à falha de um enlace. Isso ocorre, pois o pacote tem de buscar uma nova rota e efetuar dois saltos (NNHop), identificando um novo caminho.

5 APERFEIÇOAMENTO FAST RE-ROUTE RFC 4090

Neste título será apresentada uma solução alternativa à do método apresentado em IETF RFC 4090, com variações das formas de incrementar *backup* local. Em Wang et al (2008) é apresentado uma topologia de forma a ter um melhor aproveitamento da largura de banda da rede.

De acordo com Wang et al (2008) o modelo proposto em IETF RFC 4090, o compartilhamento de banda entre os *links* comuns é ineficiente, pois esta tem de reservar uma largura de banda da rede antes da falha ocorrer, e muitas vezes essa reserva é muito maior do que o necessário para o funcionamento da rede. Dessa forma é proposta uma solução para compartilhar a largura de banda por vários LSP, desde que a rede não esteja sujeita a falhas simultâneas, e tornar o *backup* mais eficiente, conforme Fig. .

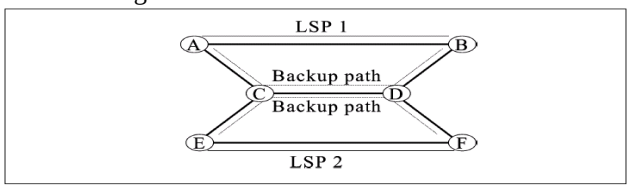


Fig. 9. Exemplo de Backup Compartilhado

Na Fig. têm-se dois enlaces principais LSP1 e LSP2. LSP1 faz a comunição de A e B e LSP2 de E a F. Ambos possuem um enlace de *backup* comum, pois em caso de falha no enlace LSP1, os caminhos de backup seriam A, C, D, B partindo do nó A. Já em caso de falha em LSP2, o caminho de *backup* partindo do nó E, seria E, C, D, F. Dessa forma tem-se o compartilhamento de banda entre dois enlaces em caso de falha.

O exemplo citado por Wang et al (2008) apresenta uma ideia de reservar largura de banda da rede para que o fluxo seja encaminhado por túneis de *backup* de outros enlaces. Com seu funcionamento baseado em trocas de mensagens tem-se um resultado satisfatório, pois a largura de banda em RFC 4090 é reservada para um caminho e pode nunca ser usada por essa razão a ideia de compartilhá-la.

6 CONCLUSÃO

A utilização de MPLS FRR traz garantia de qualidade de serviço por sua rápida recuperação do tráfego da rede. Com *Fast Re-Router* têm-se resultados satisfatórios em relação à utilização de padrões tradicionais de redes IP. Conforme apresentado na seção 3, o tempo de reencaminhamento de pacotes é muito pequeno, em relação a uma rede IP, e ainda o número de perdas de pacotes é reduzido significativamente, independente da falha ter sido em um enlace ou em um nó da rede. Esse rápido redirecionamento é possível devido à criação de rotas de *backup* antes de ocorrerem às falhas.

Entretanto, um aperfeiçoamento ainda se faz necessário a fim de tornar o redirecionamento mais dinâmico, estabelecendo critérios e identificando caminhos menos congestionados - algumas vezes mais longos -, porém mais rápidos da rede. A ideia do compartilhamento, citada na seção 4, apresenta uma alternativa para o aperfeiçoamento do *Fast Re-Route*, de

forma que cada enlace tenha um mesmo túnel de *backup*, com isso economizando-se banda da rede e consequentemente recursos computacionais.

Apesar do *Fast Re-Route* trazer novos conceitos que ajudam a solucionar uma determinada gama de problemas no tráfego de pacotes em redes, ele não consiste em uma solução definitiva para todos os problemas, e sim uma área em que existem inúmeros estudos para aprimorar a utilização dos recursos de *backup* de uma rede, garantido qualidade de serviço e otimização de recursos. Porém, fica a conclusão que mesmo no modo RFC 4090 já é possível obter resultados plenamente satisfatórios em relação a uma rede IP tradicional.

REFERÊNCIAS

MENDONÇA, R.; OLIVEIRA, J. M.; LINS, R. D. *Redes MPLS-fundamentos e aplicações.* Rio de Janeiro: Brasport. 2012.

PAN, P.; SWALLOW, G.; ATLAS, A. *Fast reroute extensions to RSVP-TE for LSP tunnels*. RFC 4090. 2005.

WANG, D.; LI, G.. *Efficient distributed bandwidth management for MPLS fast reroute*. IEEE/ACM Transactions on networking, v. 16, n. 2, p. 486-495, 2008.

NAVEED, S.; KUMAR, S. V. *MPLS Traffic Engineering–Fast Reroute*. International Journal of Science and Research (IJSR), Volume 3. 2012.

ZHAO, K.; LI, R.; JACQUENET C. *Fast Reroute Extensions to Receiver-Driven RSVP-TE for Multicast Tunnels draft-zlj-mpls-mrsvp-te-frr-01.txt* France Telecom Orange, 2013.

UMA FERRAMENTA DE GERENCIAMENTO DE QOS BASEADO EM USUÁRIOS

AN USER-BASED TOOL FOR QOS MANAGEMENT

VITOR UDO JOAO LEAL[1]*, CRISTIAN C. MACHADO[1]

[1]Departamento de Engenharias e Ciência da Computação, Universidade Regional Integrada do Alto Uruguai e das Missões, URI - Câmpus de Frederico Westphalen.
*E-mail: vitorjleal@hotmail.com

Resumo: O surgimento de novos serviços aumentaram em grande escala a quantidade e diversidade de dados que trafegam pelas redes de computadores e pela Internet. Como resultado, a mescla de serviços que podem ser executados em uma única rede apresenta situações onde algumas aplicações necessitam de garantias para seu funcionamento. Para diferenciar e dar garantias a estes tráfegos, mecanismos e técnicas que fornecem qualidade de serviço (QoS - *Quality of Service*) são utilizados. Neste contexto, este trabalho apresenta uma proposta para criação de uma ferramenta que se baseia no usuário para estabelecer regras de QoS. Como resultado, espera-se que, indiferente do dispositivo que o usuário estiver utilizando ou local onde o mesmo esteja, as regras de QoS sejam estabelecidas dinamicamente para atender as prioridades estabelecidas para aquele usuário.

Palavras-chave: Qualidade de serviço. Serviços usuários. Prioridades. Redes de computadores.

Abstract: The emergence of new services increased on a large scale the amount and diversity of data flowing over computer networks and the Internet. As a result, the mix of services that can run on a single network introduces situations where some applications require guarantees for its operation. In order to differentiate and provide guarantees to these services, mechanisms and techniques which provide quality of service (QoS - Quality of Service) are used. In this context, this work presents a proposal to create an user-based tool for establishing and handling QoS rules. As a result, it is expected that regardless the device or place where the user are a set of rules for QoS guarantees must be dynamically established to meet the user.

Keywords: Quality of Service. Services. Users. Prioridades. CBQ. Computer networks.

1 INTRODUÇÃO

Com a evolução das redes de computadores e das tecnologias de comunicação, surgiram novos serviços e funcionalidades que aumentaram em grande escala a quantidade de dados que trafegam pela Internet (FOROUZAN e MOSHARRAF, 2013). Alguns exemplos de serviços comumente usados são a utilização constante de mensageiros web para o compartilhamento de imagens e vídeos, além de um vasto acervo de conteúdo multimídia disponível e acessado em servidores públicos e privados (FOROUZAN e MOSHARRAF, 2013).

Alguns serviços utilizam grande largura de banda como é o caso de streamings de vídeo, protocolo de transferência de arquivos (FTP - *File Transfer Protocol*) ou redes ponto a ponto (P2P - *Peer-to-peer*), enquanto outros não utilizam grande largura de banda, mas necessitam de agilidade no recebimento e envio de pacotes, como é o caso do voz sobre protocolo de Internet (VoIP - *Voice over Internet Protocol*), jogos online e videoconferências (KOLIVER et al., 2000; HWANG e TSENG, 2005). Essa diversidade tem como resultado a mescla de serviços que podem ser executados em uma única rede, e apresenta situações onde algumas aplicações necessitam de maior qualidade para funcionamento (isto é, devem receber um tratamento diferenciado, por exemplo, enviar por caminhos mais curtos, "furar" filas em roteadores, etc.) compartilham largura de banda com tráfegos convencionais (que não necessitam de tratamentos diferenciados). Mais além, aplicações sensíveis a atrasos na comunicação concorrem com aplicações que consomem grande largura de banda e com aplicações que não suportam atrasos (DE MELO, 2005).

A pesquisa feita por Brunetti *et al.* (2011) aponta que o tráfego em redes de telefonia fixa tem um crescimento de 40% a 50% ao ano, enquanto as redes móveis teriam seu aumento na gama de 60% a 200% ao ano. Este aumento de tráfego se deve principalmente pela aceitação do conteúdo de vídeo entregue na Internet, já que prevê que a soma das formas de vídeo entregue (televisão, vídeo sob demanda, P2P, etc.) corresponde a 91% de todo o tráfego global no ano de 2014.

Entretanto, a rede mundial de computadores não foi projetada para todo este tráfego existente atualmente, pois ela se baseia no modelo "*best-effort*" (melhor esforço), ou seja, todos os pacotes são tratados igualmente, gerando oscilações que podem degradar a qualidade do serviço oferecido até um nível abaixo do aceitável pelo usuário

(KOLIVER et al., 2000). Para diferenciar estes tráfegos são utilizados mecanismos e técnicas que fornecem qualidade de serviço (QoS - *Quality of Service*) visando a organização e descongestionamento das redes computacionais (FOROUZAN e MOSHARRAF, 2013). Esses mecanismos e técnicas foram criados para garantir a qualidade do tráfego de determinados serviços. Resumidamente, eles identificam e organizam ações para cada grupo ou tipo de tráfego, protocolos ou aplicativos, por exemplo, VoIP, FTP e streaming, para que cada serviço usufrua da largura de banda que lhe é disponível/alocada na rede. (FOROUZAN e MOSHARRAF, 2013).

Atualmente, a maioria das ferramentas que fornecem QoS consistem no mapeamento de endereços protocolo de Internet (IP - *Internet Protocol*) ou Controle de acesso de mídia (MAC - *Media Access Control*). Nesta abordagem, quando se quer atribuir QoS para um determinado usuário, tem-se a necessidade de um conhecimento prévio de todos os dispositivos tais como, notebooks, *tablets*, estações de trabalho, entre outros, que o usuário utilizará na rede, se fazendo necessária a configuração das mesmas regras para cada dispositivo que o usuário estiver ocupando, de modo a se manter os mesmos privilégios e prioridades para acesso à rede. Essa abordagem de atribuir QoS diretamente à relação prévia de dispositivos que um usuário pode utilizar torna-se um problema quando os dispositivos são compartilhados e/ou usados por diversos usuários, em momentos diferentes ou não, que devem ter tratamentos diferenciados. Em resumo, essa abordagem é falha nas condições em que existe a necessidade de descobrir quem é o usuário do dispositivo, a fim de lhe conceder a experiência/uso que lhe deve ser fornecido – e garantido - pela rede.

Neste contexto, este trabalho apresenta a proposta de uma estratégia baseada em usuários para mapear QoS. Diferente de abordagens tradicionais, essa ferramenta permite que um mesmo indivíduo utilize as mesmas configurações em diferentes dispositivos, evitando que as mesmas configurações precisem ser feitas várias vezes.

A utilização de QoS por usuários pode apresentar diversos benefícios: i) evitar a necessidade de fazer as mesmas configurações repetidas vezes; ii) controlar usuários com níveis de prioridade diferenciados; iii) garantir que um usuário sempre tenha suas prioridades ativas; iv) garantir consistência nas configurações de um usuário, independentemente do dispositivo; v) definir diferentes níveis de prioridade para um mesmo usuário em diferentes horários.

2 CONTEXTUALIZAÇÃO E ESTADO DA ARTE

A seguir são apresentados conceitos, estudos e ferramentas relacionados com o tema proposto.

2.1 Quality of Service

Quality of servisse (QoS) ou qualidade de serviço se refere ao conjunto de técnicas e mecanismos que garantem o desempenho da rede, com o objetivo de fornecer um serviço de melhor qualidade para a camada de aplicação (FOROUZAN e MOSHARRAF, 2013). Em uma rede com QoS alguns serviços ou fluxos de dados podem ter prioridade sobre outros, ou seja, algumas conexões são mais importantes do que outras, e no caso de algum congestionamento na rede este fluxo será atendido primeiro. Assim, os recursos são reservados para atender à demanda do fluxo prioritário em detrimento de outros, atendendo as necessidades impostas (TANENBAUM, 1997; GORENDER et al., 2010).

Em roteadores com pouco fluxo de dados, os benefícios de QoS podem passar despercebidos pelo usuário final, especialmente quando a quantidade de tráfego é suficientemente baixa, e os atrasos de enfileiramento são pequenos. No pior dos casos, as técnicas de QoS usadas podem diminuir o desempenho, fazendo com que a entrega de pacotes seja atrasada, mesmo quando o roteador não está sobrecarregado. Por isso, QoS é muito mais útil em roteadores que apresentam grande fluxo de dados, com grandes filas e taxas de queda, ou que regularmente lidam com o tráfego em tempo real, que é afetado criticamente pela perda de pacotes ou a alta latência (MCWHERTER et al., 2000).

O trabalho de Mônego (2014) apresenta uma ferramenta *Open Source* para reconfiguração e controle de QoS, onde um administrador de rede pode criar diferentes regras para controle de tráfego sobre endereços IP ou portas, além de limitações de horários para as regras. A ferramenta apresenta uma interface web para gerenciamento de regras e *feedbacks* de desempenho. Este trabalho apresenta alguns problemas se aplicado a redes onde haja constante alternância entre os dispositivos conectados e o grande número de usuários conectados, pois cada dispositivo é tratado individualmente, sendo necessário realizar o cadastro de cada dispositivo no sistema sem nenhum controle sobre quem é o utilizador, diferentemente do projeto proposto neste resumo.

Segundo Forouzan e Mosharraf (2013), para se fornecer qualidade de serviço em qualquer aplicação é necessário definir quatro requisitos: i) Confiabilidade (o requisito que um fluxo precisa para entregar os pacotes integralmente ao seu destino); Atraso (o requisito que define qual o valor máximo e mínimo de atraso na entrega dos pacotes ao destino); *Jitter* (a variação do atraso em pacotes que pertencem a um mesmo fluxo); Largura de banda (a taxa de bits que cada aplicação necessita).

Compreendendo o que são estes requisitos é possível observar na Tabela 1 um resumo de tipos de aplicações e suas respectivas sensibilidades às variações da rede.

Aplicação	Confiabilidade	Atraso	Jitter	Largura de banda
FTP	Alta	Baixa	Baixa	Media
HTTP	Alta	Media	Baixa	Media
Áudio sob Demanda	Baixa	Baixa	Alta	Media
Vídeo sob Demanda	Baixa	Baixa	Alta	Alta
Voz sobre IP	Baixa	Alta	Alta	Baixa

Vídeo sobre IP	Baixa	Alta	Alta	Alta

Tabela 1. Sensibilidade às variações de rede.

2.2 Class Based Queue (CBQ)

O *script* de classe baseado em fila (CBQ - *Class Based Queue*) é um mecanismo utilizado para aplicar QoS com base em atributos encontrados nos tráfegos, tais como IP, Classe, Protocolo entre outros. Ele é baseado em regras de priorização e caracterização de pacotes, permitindo a criação de várias classes com limites de banda distintos (DIAS, 2004).

De acordo com Flores (2016), o CBQ trabalha com arquivos distintos em um mesmo diretório para a configuração de qualidade de serviço. Cada arquivo deve conter obrigatoriamente os alguns parâmetros, por exemplo, "DEVICE" que se refere à identificação do dispositivo que receberá as regras, "RATE" que se refere à velocidade atribuída ao dispositivo, "WEIGHT" que serve como parâmetro de ajuste, sendo proporcional à largura de banda e "PRIO" que define através de um valor numérico o nível de prioridade do dispositivo na rede (quanto maior o valor, menor é a prioridade).

Além destes parâmetros obrigatórios, outros parâmetros úteis para o controle são o TIME, que limita o acesso em horários predeterminados e o RULE, para controles de tráfego avançados, como aplicar controle de tráfego somente a determinada porta (exemplo: porta HTTP = 80).

2.3 Controle de Usuários

Uma questão importante em todas as arquiteturas de QoS é a forma como os serviços e as suas características são gerenciados: da chegada dos pacotes da própria rede interna ou Internet até o usuário final (TSETSEKAS et al., 2003). Para realizar o controle sobre quais usuários estão utilizando determinada rede, pública ou privada, pode-se aplicar um *captive portal* (FLICKENGER, 2002). Este sistema será utilizado para realizar o mapeamento dos usuários que estão conectados à rede. Assim cada usuário terá atrelado ao acesso à rede suas informações referentes às regras de QoS. Além disso, o *captive portal* tem o potencial de aumentar o nível de segurança da rede, pois são criados usuários e senhas particulares, substituindo assim o sistema padrão de acesso dos roteadores *wireless*, que se baseia apenas em uma senha associada a um conjunto de serviços identificadores (SSID - *Service Set Identifier*), comum a todos os utilizadores.

No trabalho aqui proposto, cada dispositivo que se conecta à rede precisa utilizar uma autenticação (composta por usuário e senha) tornando possível identificar e mapear as propriedades de QoS referentes àquele usuário.

3 Descrição Da Estratégia

A ferramenta a ser desenvolvida fará o gerenciamento de QoS em uma rede.

A Fig.1 apresenta os passos da estratégia. Como pode ser observado, o usuário irá se conectar à rede utilizando um usuário e senha. Essa conexão será realizada através de um *captive portal.* Essa abordagem será utilizada, pois o usuário autenticado no computador pode não necessariamente ser o mesmo usuário que estará utilizando a rede no momento. Cada usuário estará alocado em uma classe, e cada classe possuirá suas características e prioridades quanto ao fluxo de dados.

A fig. 1. Apresenta um estudo de caso de classes existentes em uma universidade:

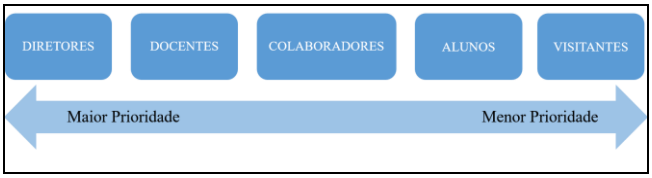


Fig. 1. Diagrama de estudo de caso sobre classes existentes em uma universidade.

Assim, as classes estarão divididas em i) diretores – pessoas responsáveis pela administração da universidade; ii) docentes – pessoas que realizam a atividade docente na universidade; iii) colaboradores – pessoas responsáveis pela parte operacional da universidade; iv) alunos – pessoas que usufruem da estrutura para seus estudos; e v) visitantes – pessoas que não possuem nenhum vínculo com a universidade. As prioridades estão estabelecidas na seguinte ordem, onde o primeiro possui maior prioridade que o segundo, o segundo que o terceiro, e assim sucessivamente: diretores, docentes, colaboradores; alunos e visitantes. Para fins de entendimento, ao observar a sequência, identifica-se que a classe diretores possui a prioridade mais alta de QoS, ou seja, que esta classe passa à frente de todo o tráfego realizado por qualquer outra classe. Por outro extremo, identifica-se que a classe visitantes possui a prioridade mais baixa de QoS, ou seja, que esta classe dá "lugar" para todo o tráfego realizado para qualquer outra classe.

Quando a classe do usuário for identificada conforme o mapeamento das classes contidas no *captive portal* e no *script* CBQ, o CBQ fará a atribuição das regras de QoS para o dispositivo pelo qual foi efetuada a autenticação. Garantindo que um usuário possa manter suas regras de QoS independente do dispositivo que estiver utilizando, conforme ilustrado na Fig. 2.

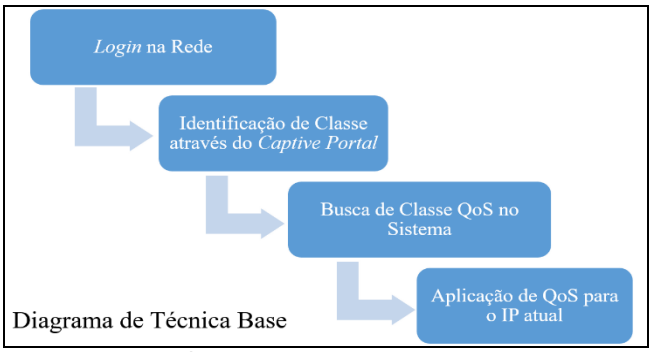


Fig. 2. Diagrama de técnica base

ISBN: 978-85-7796-213-6 <http://ocs.fw.uri.br/index.php/simcit/.

4 Resultados Esperados E Conclusão

Percebe-se, diante dos dados mencionados anteriormente por Brunetti et al. (2011), que o fluxo de dados que circula pelas redes locais e pela Internet ultrapassa em muito a projeção inicial da grande rede de computadores. Fazendo-se necessário um intermediário para fazer o gerenciamento do fluxo de dados em redes empresarias, ou instituições de ensino, onde há grande número de usuários acessando os mais diversos tipos de conteúdo e serviços

Este trabalho é projetado esperando como primeiro resultado a melhora do desempenho da rede para serviços que necessitam de alta disponibilidade de fluxo de dados, como o caso de chamadas VoIP. Também é esperada uma melhora em todo o fluxo de dados da rede uma vez que tráfegos que acabam ”roubando” a largura de banda podem ser controlados. Finalmente espera-se pela divisão de classes que os usuários que necessitem da rede a tenham sempre alta disponibilidade.

Referências

BRUNETTI, J. A.; CHAKRABARTI, K.; IONESCU-GRAFF, A. M.; NAGARAJAN, R.; SUN, D. Open Network Quality of Service and Bandwidth Control: Use Cases, Technical Architecture, and Business Models. *Bell Labs Technical Journal*, v. 16. n. 2. p. 133 – 152, set. 2011.

DE MELO, J. C. *Estudo da Utilização de Mecanismos de QoS em Redes com Enlaces de Banda Estreita*. São Luís, 2005, 136 f. Dissertação de Conclusão de Mestrado em Ciência da Computação – Universidade Federal do Maranhão.

DIAS, L. *Controle sua banda de maneira simples e inteligente com CBQ*. Disponível em: <https://www.vivaolinux.com.br/artigo/Controle-sua-banda-de-maneira-simples-e-inteligente-com-CBQ>. Acesso em: 07 maio 2016.

FLORES, F. *Linux in Brazil* (Controle de banda com CBQ). Disponível em: <http://br-linux.org/artigos/dicas_cbq.htm> Acesso em: 27 maio 2016.

FOROUZAN, B. A.; MOSHARRAF, A. *Redes de Computadores*: uma abordagem top-down. Porto Alegre, AMGH, 2013. p. 896.

GORENDER, S.; MACÊDO, R. J. A.; PACHECO JR, W. L. Controle de admissão para qos em sistemas distribuídos híbridos, tolerantes a falhas. In: Workshop De Testes E Tolerância A Falhas, 11, 2010, Gramado. *Anais do XI Workshop de Testes e Tolerância a Falhas*, 2010. p. 45 – 58.

HWANG, W-S.; TSENG, P-C. A QoS aware Residential Gateway with bandwidth management. *IEEE Transactions on Consumer Electronics*, v. 51. n. 3. p. 840 – 848, ago. 2005.

KOLIVER, C.; FARINES, J-M.; FRAGA, J. S.; DOS REIS, H. L. Um modelo para Adaptação de QoS Orientado ao Usuário Final. In: Simpósio Brasileiro De Redes De Computadores, 18, 2000, Belo Horizonte. *Anais 2000 do 18 Simpósio Brasileiro de Redes de Computadores*. Belo Horizonte, 2000. p. 135 – 149.

KUROSE, J. F.; ROSS, K. W. *Redes de computadores e a internet: Uma abordagem top-down*. São Paulo, Pearson, 2010. p. 572.

MAHADEVAN, I.; SIVALINGAM, K. M. Architecture and experimental results for quality of service in mobile networks using RSVP and CBQ. *Wireless Networks*, v. 6. n. 3. p. 221 – 234, jul. 2000.

MCWHERTER, D. T.; SEVY, J.; REGLI, W. C. Building an IP Network Quality-of-Service Testbed. IEEE Internet Computing, v. 4. n.4. p. 65 – 73, ago. 2000.

MÔNEGO, C. FORCEQOS - *Ferramenta Oppen source para reconfiguração e controle eficiente de QoS*. Frederico Westphalen, 2014, 117 f. Monografia de Graduação do Curso de Ciência da Computação – Câmpus de Frederico Westphalen, Universidade Regional Integrada do Alto Uruguai e das Missões

TANENBAUM, A. S. *Rede de computadores.* Rio de Janeiro, Campus, 1997. p. 993.

TSETSEKAS, C. A.; MANIATIS, S. I.; VENIERIS, I. S. The end-user application toolkit: a QoS portal for the next generation Internet. *International Journal of* Communication Systems, v. 16. n. 7. p. 605 – 626, set. 2003

SISTEMAS DE TERMO-HIGRÔMETRO DIGITAL MICROCONTROLADO COM LEITURA INTERNA E EXTERNA

THERMO-HIGROMETER DIGITAL SYSTEMS MICROCONTROLLED WITH READING INTERNAL AND EXTERNAL

EDEMAR O. PRADO[1], AMAURI F. BALOTIN[1], HAMILTOM C. SARTORI[1]

[I] Departamento de Engenharias e Ciência da Computação, Universidade Regional Integrada do Alto Uruguai e das Missões – URI, Câmpus de Frederico Westphalen, RS, Brasil. *E-mail:eder_iron@hotmail.com

Resumo: O presente trabalho tem por objetivo a elaboração de um termo-higrômetro digital para utilização em ambientes internos e externos simultaneamente, o protótipo desenvolvido não armazena os dados, mas os apresenta em um *display* lcd, com seus valores atualizados a cada 2 segundos. Para o desenvolvimento do protótipo, implementou-se um *firmware* utilizando linguagem C, obedecendo os critérios estabelecidos pelo *datasheet* dos componentes eletrônicos. Foram utilizados sensores DHT11, microcontrolador, *display* lcd e componentes de menor porte, como resistores capacitores e cristal de quartzo, selecionados por apresentarem baixo custo, eficiência aceitável e fácil aquisição. Por fim, observa-se que o equipamento apresentou o resultado esperado, pois foram apresentados no display os valores de temperatura e umidade de cada sensor com os valores atualizados conforme preestabelecido anteriormente. Como sequência do trabalho, pretende-se realizar a implementação de um sistema de armazenamento dos dados, o que irá permitir que este sistema possa ser utilizado para monitoramento de temperatura e umidade, como também, no controle e automação de câmeras de secagem.

Palavras-chave: Termo-higrômetro. Microcontrolador. Linguagem C. DHT11.

Abstract: The present work it aims, the development a digital thermo-hygrometer for use in environments internal and external simultaneously, the developed prototype does not store data, but presents on a lcd display, with their values updated every 2 seconds. For the development of the prototype, implemented a firmware using C language, according to the criteria established by the datasheet of electronic components. DHT11 sensors are used, microcontroller, lcd display and smaller components, as capacitors resistors and quartz crystal, selected for having low cost, acceptable efficiency and easy to purchase. Lastly, it is observed that the machine He presented the expected result, as they were presented on the display temperature values and humidity of each sensor with the updated values as pre-established previously. As a result of the work, it is intended to carry out the implementation a data storage system, which will allow this system it can be used for monitoring temperature and humidity, as also, in control and automation drying cameras.

Keywords: Thermo- hygrometer. Microcontroller. C language. DHT11.

1 INTRODUÇÃO

O microcontrolador é um dos principais componentes quando se aborda o tema eletrônica. Diferentemente de um microprocessador, o microcontrolador não necessita de periféricos externos para seu funcionamento e por sua extrema funcionalidade está presente na maioria dos dispositivos e protótipos eletrônicos.

No mercado, encontram-se os mais diversos modelos de microcontroladores, de acordo com Sena (2008) o microcontrolador PIC16F887 produzido pela empresa *Microchip*®, é um componente de elevada eficiência, fácil aquisição e baixo custo, podendo ser aplicado nos mais diversos processos, envolvendo controle ou, até mesmo, aquisição de valores e leitura de dados.

Chen e Lu (2005) descrevem os sensores de humidade como ferramentas que vêm destacando-se na área de controle de processos industriais e ambientais, sua aplicabilidade compreende desde as mais simples situações, como controle de ambiente em residências, como também, aplicação em circuitos de elevada tecnologia aplicados à indústria.

O sensor de temperatura e umidade DHT11, conforme dados encontrados em seu *datasheet*, apresenta uma leitura complexa; por possuir conversor A/D interno apresenta saída de dados em sinal digital, transmitida por comunicação 1-Wire do sensor até o microcontrolador (DHT11, 2010).

Para que o microcontrolador execute a tarefa desejada utiliza-se um compilador, onde se deve dizer exatamente o que fazer, ou seja, deve-se escrever o programa que o

microcontrolador deve executar. Conforme Silva (2013), para facilitar a programação executada pelo projetista, é vantajosa a utilização de linguagem de alto nível, no caso a linguagem C, onde ignoram-se detalhes internos do chip, preocupando-se somente com a lógica associada ao problema.

Visto isso, o presente trabalho teve como objetivo desenvolver um sistema eletrônico capaz de adquirir informações de temperatura e umidade relativa do ar de dois sensores, um interno e outro externo. Tal sistema poderá ser utilizado no monitoramento e controle de câmaras de secagem ou estufas agrícolas, por exemplo.

Visto isso, o presente trabalho teve como objetivo desenvolver um sistema eletrônico capaz de adquirir informações de temperatura e umidade relativa do ar de dois sensores, um interno e outro externo.

2 MATERIAIS E MÉTODOS

O trabalho subdivide-se em 2 etapas fundamentais, como ilustrado na Figura 1, a qual apresenta um diagrama de blocos do sistema proposto. O sistema é composto por: *firmware* e *hardware*:

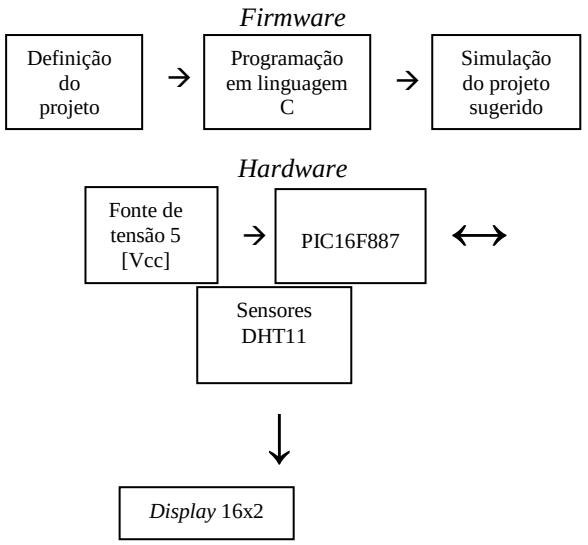


Fig. 1: Diagrama de blocos do circuito

A primeira etapa consiste na elaboração do *firmware* do circuito, que pode ser definido como um sistema de termo-higrômetro digital microcontrolado com leitura interna e externa de ambientes, utilizando sensores de temperatura e umidade dht11. As orientações a serem gravadas no microcontrolador, obedecem aos critérios pré-estabelecidos no *datasheet* do sensor, então interpretados e associados a uma lógica de programação, desenvolvida pelo autor do estudo. Para este estudo utilizou-se uma versão de demonstração do compilador *PCW*, funcional por 30 dias.

Inicialmente, define-se as conexões do *display* lcd com o microcontrolador, neste caso a troca de informação do microcontrolador com o *display* ocorre por meio do PORTB. Seguido pelas diretivas de compilação e as definições de conexão entre *Master* e *slave* (microcontrolador e os sensores), conforme observa-se na Figura 2:

```
 9  //Modo de conexão display lcd
10  #define LCD_RS_PIN        PIN_B0
11  #define LCD_RW_PIN        PIN_B1
12  #define LCD_ENABLE_PIN    PIN_B2
13  #define LCD_DATA4         PIN_B3
14  #define LCD_DATA5         PIN_B4
15  #define LCD_DATA6         PIN_B5
16  #define LCD_DATA7         PIN_B6
17  //Fim das conexões do display
18
19  #include <16F887.h>
20  #fuses HS
21  #fuses NOWDT                  //de
22  #fuses PROTECT                //pr
23  #fuses NOLVP                  //Pr
24  #use delay(clock = 8000000)   //fr
25  #include <lcd.c>              //bi
26  #use fast_io(D)               //co
27
28  // Conexão entre o PIC16F887 e o
29  #BIT nivel = 0x08.0
30  #BIT direcao = 0x88.0
31  #BIT nivel1 = 0x08.1
32  #BIT direcao1 = 0x88.1
33
34  char temperatura[] = "Ti=__ C";
35  char umidade[] =     "Ui=__ %";
36
37  char temperatura1[] = "Te=__ C";
38  char umidade1[] =     "Ue=__ %";
```

Fig. 2: Conexões e diretivas de compilação necessárias

A utilização da rotina de *fast_io(D)*, permite o controle automático do *TRIS* do *PORTD* executado pelo programa, o que gera mais velocidade e simplicidade nas funções de entrada e saída a serem implementadas posteriormente.

Outro fator de grande importância é a declaração de variáveis, que para este caso em especifico, tem suas *strings* principais, declaradas em forma de matriz, permitindo a inserção dos resultados em posições 'x', sem que o display atualize as linhas e colunas por completo a cada laço de atualização dos dados.

Para a obtenção dos dados a partir do sensor, alguns pré-requisitos estabelecidos pelo *datasheet* devem ser seguidos. Diferentemente de outros sensores e protocolos, o DHT11, segue um sistema de comunicação e troca de dados definidos pelo próprio fabricante, que é dado conforme a Figura 3:

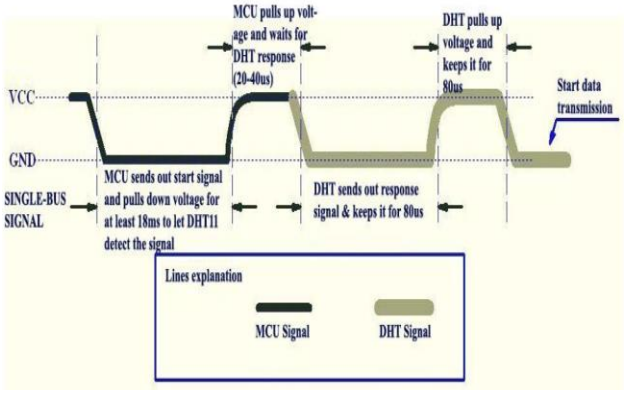


Fig. 3: Troca de dados entre *Master e Slave.* Fonte: DHT11 (2010).

ISBN: 978-85-7796-213-6 <http://ocs.fw.uri.br/index.php/simcit/>

Inicialmente o microcontrolador envia um sinal em nível lógico baixo, por 18 ms, seguidos por um sinal em nível lógico alto, podendo ser de 20 a 40 us, alterando seu *TRIS*, os pinos de conexão com os sensores passam a ser entradas, tento o restante do processo executado automaticamente pelo sensor.

O sensor retorna 40 bits de dados, que devem ser divididos de 8 em 8 bits através do aninhamento de um laço *if*, um laço *for* e um laço *while*. Onde os 8 primeiros bits, correspondem aos valores inteiros da umidade, seguidos por 8 bits referentes às casas decimais da umidade, 8 bits correspondentes aos valores inteiros da temperatura, 8 bits correspondentes as casas decimais da temperatura, e por fim, os últimos 8 bits que devem ser iguais à soma dos 36 bits anteriores.

Se verdadeiro, a rotina de leitura dos sensores e plotagem da *string* no *display* lcd será executada, a rotina de execução está ilustrada abaixo de acordo com a Figura 4:

```
257         if(soma == ((RH_Byte1 + RH_Byte2 + T_Byte1 + T_Byte2) & 0:
258         {
259          temperatura[3]  = T_Byte1/10  + 48;
260          temperatura[4]  = T_Byte1%10  + 48;
261          //temperatura[6] = T_Byte2/10  + 48;
262          umidade[3]  = RH_Byte1/10 + 48;
263          umidade[4]  = RH_Byte1%10 + 48;
264          //umidade[7] = RH_Byte2/10 + 48;
265          temperatura[7] = 223;                  // Simbolo de grai
266
267          lcd_gotoxy(1, 1);                      // Ir para a colui
268          printf(lcd_putc, temperatura);         // Mostra tempera
269          lcd_gotoxy(1, 2);                      // Ir para a colui
270          printf(lcd_putc, temperatura1);        // Mostra tempera
271          lcd_gotoxy(9, 1);                      // Ir para a colui
272          printf(lcd_putc, umidade);             // Mostra umidade
273          lcd_gotoxy(9, 2);                      // Ir para a colui
274          printf(lcd_putc, umidade1);            // Mostra umidade
```

Fig. 4: Leitura e plotagem dos dados

As plotagens dos valores dos dois sensores são impressas para o display dentro do laço de execução de cada sensor, assim, caso um deles esteja desconectado ou com defeito, o display acusará esta falha, sem a necessidade de implementação de uma rotina de erro, reduzindo a utilização de memória RAM do microcontrolador

A montagem do *hardware* foi executada em uma placa, onde foram soldados os componentes, conforme dados contidos nos *datasheets* do sensor, do PIC, e do *display* lcd. Para controle do contraste do display inseriu-se um potenciômetro de 10KΩ, e para os valores de referência do PIC, foi utilizado um circuito oscilador, compreendido por um cristal de quartzo de 8 MHz, e dois capacitores de 22 pF.

A simulação foi efetuada no *software* Proteus, versão demonstração, seguindo as instruções informadas anteriormente e a configuração de conexão a 4 fios entre o display e o microcontrolador, onde o mesmo método adotou-se para a montagem do *hardware*, com a transferência do arquivo inicialmente programado até o microcontrolador, executada em um circuito externo de gravação.

3 RESULTADOS

A Figura 5 apresenta o circuito resultante obtido na simulação efetuada, onde as casas decimais foram desconsideradas, e o controle do contraste do *display* é executado através de um potenciômetro de 10KΩ:

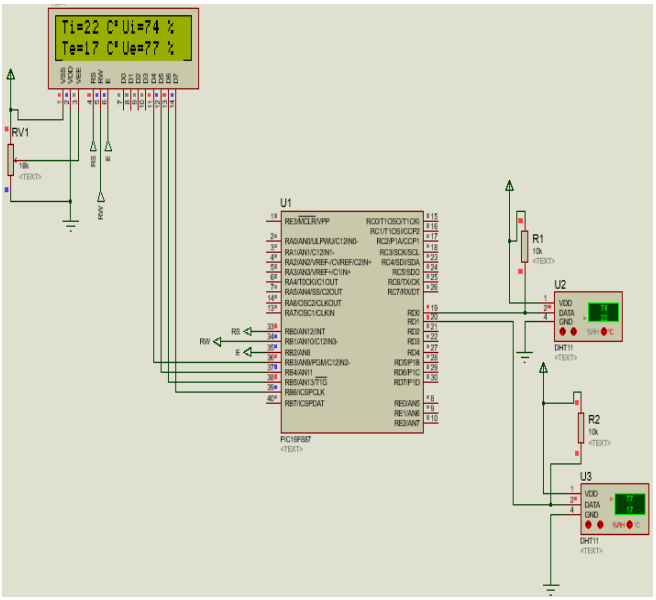


Fig. 5: Simulação do circuito resultante

Os valores apresentados no *display* são valores aleatórios, onde o valor a ser plotado é ajustado manualmente pelo programador, sem influência de ações ambientais.

Já o circuito de *hardware* é de controle automático, a leitura executada pelos sensores é transmitida até o microcontrolador, convertida para digito de impressão, e então plotado no *display*, a representação do circuito resultante do *hardware* executado, apresenta-se conforme Figura 6, onde tem-se o protótipo finalizado, com ambos os componentes e circuitos necessários para seu correto funcionamento:

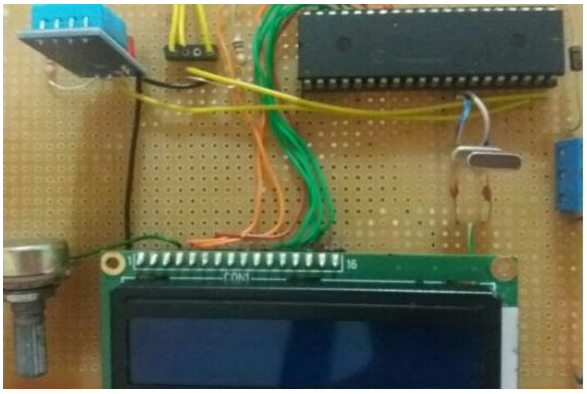


Fig. 6: Circuito de *hardware* resultante

Por fins de melhor visualização, o circuito em operação apresenta-se na Figura 7:

ISBN: 978-85-7796-213-6 <http://ocs.fw.uri.br/index.php/simcit/>

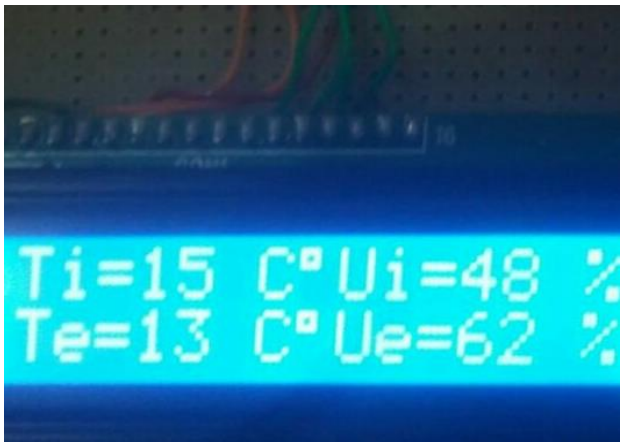


Fig. 7: Circuito em operação

Observa-se claramente na Figura 6 o item anteriormente citado, além do circuito oscilador, presente abaixo do microcontrolador, já a Figura 7 traz o circuito quando, alimentado por uma fonte de tensão de 5 [Vcc], onde o mesmo entra em operação, tendo os dados plotados no *display*.

4 DISCUSSÕES

Com base nos resultados obtidos, observa-se que o circuito proposto funciona de acordo com o esperado.

O arquivo, inicialmente carregado no microcontrolador, executa as tarefas necessárias no *hardware*, conforme executa no simulador, deixando claro a importância da realização de simulação, anteriormente ao processo de montagem de circuitos, sendo uma forma de agilizar o processo.

Com referência ao circuito e sua operação, os resultados atendem às expectativas, apresentando baixo custo benefício, e eficiência aceitável, sua atualização ocorre no tempo previsto, e não apresenta falhas, sendo uma ótima opção em sistemas de controle, onde estejam envolvidas grandezas referentes à temperatura e humidade de dois ambientes simultaneamente.

5 CONCLUSÃO

Conclui-se, com base nos estudos, que o sistema de termo-higrômetro, é um sistema de inúmeras utilidades, não servindo apenas como sistema de leitura, mas também, em sistemas de controle e automação, tanto residencial, no controle de temperatura e umidade de ambientes, como industrial, no controle de estufas e câmaras de secagem, tornando-se uma opção altamente viável, devido a seu baixo custo, simplicidade e robustez, bem como, sua ampla aplicabilidade.

Caso se deseje aumentar a precisão do sistema, a sugestão é a troca do sensor, por um DHT33 do mesmo fabricante, pois apresenta o mesmo sistema de leitura, com menor faixa de erro, porém esta substituição acarretará em um aumento significativo no custo benefício do protótipo.

REFERÊNCIAS

CHEN, Zhi, LU, Chi, 2005. Humidity sensors: a review of materials and mechanisms. *Sensor letters* .v. 3, n. 4, p. 274-295

DHT11 *Humidity & Temperature Sensor*, 2010. Disponível em: <http://www.micropik.com/PDF/dht11.pdf> Acesso em: 22 jul 2016.

SENA, Antônio Sérgio. *Microcontroladores PIC*. Copyright 2008.

SILVA, Vidal Pereira Jr. *Microcontroladores PIC 16F e 18F – Teoria e Prática* 1.ed. NCB São Paulo, Brasil ,2013

O PROTOCOLO DE INICIAÇÃO DE SESSÃO – SIP

SESSION INITIATION PROTOCOL – SIP

RAFAEL POLLON[1]

[1]Departamento de Engenharias e Ciência da Computação, Universidade Regional Integrada do Alto Uruguai e das Missões, URI - Câmpus de Frederico Westphalen.
*E-mail: rafael.pollon@hotmail.com

Resumo: As descobertas de novas formas que possibilitam aprimorar os meios de comunicação existentes impulsionam o surgimento de novos protocolos para suprir as necessidades de cada uma delas. Dentro deste pretexto, com a descoberta da digitalização da voz, surgiu a telefonia digital. E com ela, em meados da década de 1990 foi desenvolvido o protocolo *SIP*. Do Inglês: *Session Initiation Protocol* ou Protocolo De Iniciação De Sessão é um protocolo baseado no modelo requisição-resposta e é utilizado para estabelecer chamadas telefônicas pela rede IP. O presente trabalho ressalta aspectos importantes do protocolo que é o mais utilizado na telefonia digital, para isso, são abordados aspectos que partem desde a conversão analógico-digital de sinais até chegar à sinalização e controle dos pacotes de voz que são transportados por uma rede digital.

Palavras-chave: Digitalização. Telefonia digital. SIP.

Abstract: The discoveries of new ways that allow improve existing media drive the emergence of new protocols to meet the needs of each. Under this pretext, with the voice scan discovery came the digital telephony. And with it, in the mid-1990s it was developed SIP protocol. The SIP is a protocol based on the request-response model and is used to establish telephone calls over the IP network. This study highlights important aspects of the protocol that is the most used in digital telephony, for that address aspects departing from the analog to digital conversion signals to get to the signaling and control of voice packets that are transported by a digital network.

Keywords: Coding. Digital Telephony. SIP.

1 INTRODUÇÃO

Vê-se que os meios de comunicação estão convergindo cada vez mais para que diversos tipos de produtos possam trafegar pelo mesmo meio de transmissão. Um produto que se beneficiou bastante da utilização de um meio de transmissão que já existe para implementar um novo serviço é o sistema de telefonia IP, ou VOIP. O presente trabalho tem por objetivo explorar o funcionamento dessa telefonia com a utilização do protocolo SIP. Para isso, inicia-se uma abordagem pelos aspectos importantes da conversão analógico-digital de sinais, passando por todos os processos necessários para que a comunicação entre usuários seja controlada através da rede IP. O capitulo 2 inicia pelos aspectos fundamentais da conversão analógico-digital.

2 MODULAÇÃO PCM

Com a descoberta da conversão A/D, os sinais analógicos puderam ser codificados por meio da eletrônica digital. Esta técnica possibilita diversos benefícios na transmissão digital, entre eles, o uso de criptografia a fim de adicionar informações de uma forma cifrada para impedir ou dificultar a interpretação da mensagem, bem como, a capacidade de armazenar de maneira mais segura estas informações. (HERSENT; GUIDE; PETIT, 2002).

A maioria das informações que são transportadas em uma rede de telecomunicação são sinais digitais, algumas, que entram e saem da rede de forma analógica. Para que essa transmissão analógica ocorra por um meio digital, é necessário converter estas informações na chegada e desconvertê-las na saída. Uma técnica bastante utilizada para este fim é a modulação por código de pulso ou PCM, do inglês Pulse Code Modulation. (Frenzel, 2013).

A modulação PCM consiste em três estágios: Amostragem, Quantização e Codificação. Estes três estágios se baseiam em: medir o sinal analógico em intervalos de tempo regulares; aproximar os valores para um nível de referência previamente estabelecido; e codificar o valor em uma sequencia de bits. O principio de funcionamento dos três estágios serão explicados a seguir.

O sinal analógico, quando submetido à etapa de amostragem, é entrecortado em intervalos tempo iguais, assim, obtêm-se pulsos de acordo com a amplitude do sinal. Esta operação pode ser realizada pelo modulador PAM do inglês (*Pulse Amplitude Modulation*) ou modulação por amplitude de pulso. (MEDEIROS, 2016). O estágio da amostragem segue o teorema de Nyquist, que diz: “A frequência de amostragem deve ser no mínimo o dobro da largura de banda deste sinal”. A

justificativa deste teorema implica em não conseguir recuperar o sinal. (Frenzel, 2013).

No estágio de quantização, sinal analógico foi convertido para pulsos e agora é necessário medir eletronicamente a altura deles. A técnica chamada de quantização consiste em realizar um arredondamento dos pulsos para que os mesmos se encaixem nos valores de decisão estabelecidos.

No ultimo estágio, chamado de codificação, tem por objetivo transformar o sinal quantizado em um sinal binário, para que possa trafegar por uma rede digital.

3 Transportado Voz Sobre Uma Rede De Pacotes

Se tratando de telefonia fixa, o sistema analógico é o mais comum hoje em dia. Há algumas desvantagens dele em relação à telefonia digital, desde os custos para manter grandes centrais de comutação, os custos com grandes redes analógicas de baixa capacidade de transmissão de dados e até mesmo, desvantagens se tratando da qualidade da voz, já que o ruído parasítico é adicionado em todos os estágios da transmissão, e este sinal não pode ser limpo, pois não há como saber o que é informação e o que é ruído.

Na telefonia digital, o meio de transmissão da informação é a rede IP, assim, fica clara a principal vantagem sobre a rede de transmissão analógica. Utiliza-se um caminho já existente que conecta computadores ao redor do mundo para transportar chamadas telefônicas sem a necessidade direta de pagar mais por isso. Além disso, a quantidade de chamadas simultâneas por esse meio de transmissão impulsiona o avanço de técnicas que permitem a economia do mesmo. Uma delas é a Multiplexação por divisão de tempo, do inglês: *Time Division Multiplexing* – TDM, ela consiste em dividir pequenos intervalos de tempo para cada conversa telefônica que passa no mesmo meio de transmissão. (Frenzel Jr, 2013).

a) Os protocolos TCP e UDP

O Protocolo De Datagrama De Usuário, do inglês *(User Datagram Protocol* - UDP) é o protocolo mais simples da camada de transporte. Seu funcionamento é baseado no envio de *datagramas* encapsulados aos seus destinatários, porém, sem dar a garantias ao remetente que os dados chegaram ao seu destino final. (FOROUZAN E MOUSHRAFF, 2013).

Protocolo de Controle de Transmissão, do inglês (*Transmission Control Protocol* - TCP) é um protocolo orientado à conexão que tem como objetivo prover um fluxo de dados fim a fim confiável. (TANEMBAUM, 2003).

3.2 A telefonia IP

Do Inglês *Voice Over Internet Protocol* (VOIP) ou, Voz Sobre O Protocolo De Internet, é o nome dado à telefonia que funciona com o uso da rede IP. Dessa forma, o VOIP utiliza os protocolos TCP e UDP para estabelecer ligações e mandar mídias de áudio RTP, destinadas a um endereço IP que pode estar em qualquer lugar do mundo. (HERSENT; GUIDE; PETIT, 2002).

Para que uma chamada telefônica seja estabelecida, o servidor da telefonia precisa receber do assinante o número completo a ser chamado, estabelecer o caminho para a chamada e avisar ao outro assinante que existe uma chamada para ele. O sistema que cumpre estas funções em uma rede de telefonia é chamado de sinalização.

4 O Protocolo De Sinalização Sip

Do inglês *Session Initiation Protocol* (SIP), ou Protocolo De Iniciação De Sessão é um padrão da Internet *Engineering Task Force* (IETF) definido na RFC 2543. Este protocolo utiliza o modelo requisição-resposta para estabelecer chamadas telefônicas pela rede IP. (RFC 2543, 1999).

Para estabelecer uma chamada, o SIP necessita primeiramente abrir uma conexão de sinalização entre os pontos de origem e destino da chamada. Para isso, podem ser utilizados os protocolos TCP ou UDP, no caso de se utilizar o protocolo TCP, a mesma conexão pode ser utilizada para todos os pedidos e respostas do SIP, exceto para os dados de mídia. Se o protocolo UDP for utilizado, o endereço IP e a porta de acesso são enviados em cada requisição-resposta. (HERSENT; GUIDE; PETIT, 2002).

4.1 Os pedidos SIP

Como o protocolo é baseado em requisição-resposta, alguns pedidos são utilizados para o funcionamento da sinalização, eles são ilustrados na tabela 1.

Método	Descrição
Invite	Usado para iniciar uma chamada.
ACK	Enviado pelo cliente para confirmar que ele recebeu uma resposta do servidor.
Bye	Enviado pelo agente de origem ou pelo agente de destino para interromper uma chamada
Cancel	Enviado quando se quer interromper um pedido que foi enviado anteriormente, enquanto o servidor ainda não tiver enviado uma resposta final.
Options	Enviado ao servidor pelo cliente para saber as capacidades que o servidor suporta.
Register	Registra a localização atual de um determinado cliente.

Tabela 1. Pedidos SIP.

4.2 As respostas SIP

Um servidor SIP responde a um pedido com uma ou mais respostas SIP. A primeira linha de uma resposta SIP contém um código de status e uma frase inteligível

por pessoas. A tabela 2 ilustra as seis categorias das respostas SIP.

Código	Perfil	Descrição
1xx	Informativo	Pedido recebido, continuando a processar o pedido
2xx	Sucesso	A ação foi recebida, entendida e aceita com sucesso
3xx	Redirecionamento	Uma ação adicional deve ser tomada para completar o pedido
4xx	Erros de cliente	O pedido contém uma sintaxe inválida ou não pode ser efetuado neste servidor
5xx	Erros de servidor	Erro de servidor
6xx	Falha Global	Falha global

Tabela 2. As categorias dos códigos de status.

Os grupos de códigos 2xx, 3xx, 4xx, 5xx e 6xx, são códigos de respostas finais e finalizam a transação SIP. O único código que não pode encerrar uma chamada é o 1xx, pois se trata apenas de um código informativo.

4.3 O modelo de requisição e resposta

Esse modelo de troca de informações opera enviando um pedido e aguardando uma resposta. Um exemplo é quando um cliente que utilizada o protocolo HTTP (*Hypertext Transfer Protocol*) para entrar em um endereço WEB (*World Wide Web*), ele solicita informações ao servidor e fica aguardando uma resposta. (RFC 2616, 1999).

Através deste modelo, uma chamada pode ser sinalizada e estabelecida. A Figura 1 ilustra a ordem dos eventos para o estabelecimento de uma ligação entre dois clientes de telefonia, neste cenário, denominados agentes.

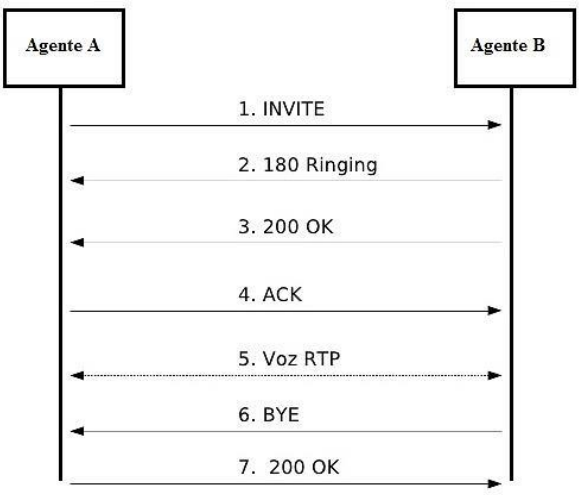


Fig. 1. Conexão ponto a ponto entre dois usuários.

Primeiramente o agente A envia um pedido *Invite* para o agente B. Este responde com *Ringing*, pois pode tocar. Quando a chamada é atendida pelo agente B, ele envia um OK para o agente A que responde com ACK. A partir deste ponto, ocorre o fluxo RTP de áudio entre os dois agentes. Na sexta etapa, o agente B encerra a chamada com *BYE* para o agente A, e o agente A aceita a finalização da chamada com um OK

4.4 A Sintaxe de descrição de sessão, SDP

Do Inglês *Session Description Protocol* (SDP), ou *Sintaxe De Descrição De Sessão*, é um padrão da Internet *Engineering Task Force* (IETF) definido na RFC 4566. O objetivo do protocolo SDP é definir uma sintaxe padrão para alguns tipos de informação, como: qual o endereço *multicast* será usado pela sessão; qual será a porta de destino UDP; quais serão os codificadores de áudio que serão usados; quais as informações adicionais sobre a sessão, como nome e breve descrição; quais as informações para contato; qual é o programa de atividades; (RFC 4566, 2006).

Com essas informações, esse protocolo fica legível para pessoas, facilitando a depuração e a programação.

4.5 O protocolo SIP e a telefonia IP

Nas Figura 2 e 3, é demonstrado um *Invite* SIP. Na linha de início, o IP de origem 192.168.0.9 envia um INVITE para o IP 192.168.0.10 utilizando a porta 15853 para a porta de sinalização 5060, esta que é a porta padrão da sinalização SIP.

Dentro do cabeçalho geral, cabeçalho de pedido e cabeçalho de entidade, são demonstradas informações a respeito do número de origem e do número de destino, neste caso, o usuário do ramal 1000 está realizando uma chamada para o ramal 1001. Este *Invite* é do tipo UDP, então, ambos os lados enviam um pedido e aguardam um tempo até que a outra ponta responda o mesmo, sem a confirmação de que o pedido realmente chegou ao destinatário.

U 192.168.0.9:15853 -> 192.168.0.10:5060 INVITE sip:1000@192.168.0.10 SIP/2.0.	Linha de início.
Via: SIP/2.0/UDP 192.168.0.9:15853;branch=z9hG4bK-aefa5f30. From: CALLER ID <sip:1000@192.168.0.10>;tag=fffaac15a3700ce8o1. To: <sip:1001@192.168.0.10>. Call-ID: f33f959-4bd38074@192.168.0.191. CSeq: 101 INVITE.	Cabeçalho Geral
Max-Forwards: 70. Contact: CALLER ID <sip:1001@192.168.0.9:15853>. Expires: 240. User-Agent: Linksys/PAP2-3.1.22(LS).	Cabeçalho de pedido
Content-Length: 450. Allow: ACK, BYE, CANCEL, INFO, INVITE, NOTIFY, OPTIONS, REFER. Supported: x-sipura, replaces. Content-Type: application/sdp.	Cabeçalho de entidade

Fig. 2. Exemplo de um *Invite* SIP.

As informações a respeito do SDP estão na Figura 3. Nestas informações, é possível verificar que o endereço 192.168.0.9 está pedindo áudio RTP na porta 16396, juntamente com algumas especificações dos codificadores de áudio disponíveis para a negociação da mídia.

ISBN: 978-85-7796-213-6 <http://ocs.fw.uri.br/index.php/simcit/>

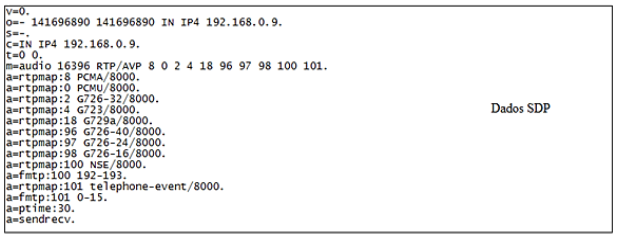


Fig. 3. Dados SDP de um *Invite* SIP.

As respostas ilustradas na figura 4 como *Trying* e *Session Progress*, são separadas por um ponto e uma linha em branco. A resposta *Trying* é do código de status 100, do tipo informativo, que nesse caso o endereço IP 192.168.0.10 fala para o endereço 192.168.0.9 que vai tentar completar a ligação. A resposta *Session Progress* também é uma resposta informativa, do código 183. Ela significa que houve algum progresso na chamada e que o número 1001 respondeu o *Invite* pedindo áudio na porta 16412, juntamente com alguns codificadores de áudio.

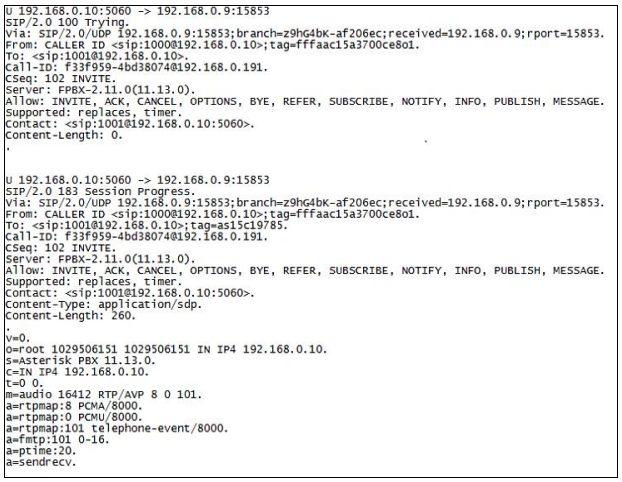


Fig. 4. Sinalização SIP.

A Figura 5 ilustra o ultimo passo da iniciação de uma chamada telefônica pela rede IP, ela demonstra o endereço IP 192.168.10 enviando um evento de código informativo 180 para o endereço IP 192.168.0.9, informando que o telefone está tocando.

Fig. 5. Resposta *Ringing* para um *Invite* SIP.

A partir desse ponto, podem ocorrer diversos códigos de resposta, dependendo do comportamento do agente que está tocando, ele pode atender essa chamada ou cancelar ela. Caso o agente atenda essa chamada, é iniciada a transmissão RTP (*Real Time Transport Protocol*) ou protocolo de transporte em tempo real, descrito na RFC 1889. Este, que permite o transporte UDP do áudio já codificado entre dois endereços IPs, transporte que ocorre no modo *Full-Duplex*. (RFC 1889, 1996).

5 CONSIDERAÇÕES FINAIS

A rede mundial de computadores fez com que muitas portas fossem abertas para novas formas de comunicação na era digital, esse avanço implicou o uso de técnicas de funcionalidade e aprimoramento, para que esse novo meio de transmissão fosse usado da melhor forma possível. A telefonia digital é uma delas. Ela permitiu utilizar essa rede já existente para a transmissão de voz de maneira mais limpa, se comparada aos sistemas de telefonia analógica.

O protocolo SIP utilizou os protocolos já existentes para que as chamadas através da rede IP fossem sinalizadas de uma maneira inteligível por pessoas, facilitando a programação e a resolução de eventuais problemas que podem ocorrer durante a comunicação.

Com esses avanços, a rede de telefonia ficou mais eficiente em diversos aspectos, como na economia de recursos e no melhor aproveitamento dos meios de transmissão existentes

REFERÊNCIAS

HERSENT, O; GUIDE, D; PETIT, J. *Telefonia IP – Comunicação multimídia baseada em pacotes.* p. 62, 2002.

LOUIS E. FRENZEL JR. *Fundamentos de comunicação Eletrônica* – Modulação, Demodulação e Recepção. p. 192, 2013.

MEDEIROS, Julio Cesar de *O. Princípios de telecomunicações* – Teoria e prática. P 68-71, 2016.

LOUIS E. FRENZEL JR. *Fundamentos de comunicação* Eletrônica – Modulação, Demodulação e Recepção. p. 10, 2013.

HERSENT, O; GUIDE, D; PETIT, J. *Telefonia IP – Comunicação multimídia baseada em pacotes*. p. 10, 2002.

FOROUZAN, Behrouz A.; MOSHARRAF, F. *Redes de Computadores:* Uma Abordagem Top-Down. 2013.

TANENBAUM, Andrew S. Redes de Computadores. 2003.

REQUEST FOR COMMENTS. RFC 2543: *SIP*: Session Initiation Protocol. Disponível em: <https://www.ietf.org/rfc/rfc2543.txt> Acesso em: 27 jul. 2016.

REQUEST FOR COMMENTS. RFC 2616: *Hypertext Transfer Protocol* - HTTP/1.1. Disponível em: <https://www.ietf.org/rfc/rfc2616.txt> Acesso em: 27 jul. 2016.

REQUEST FOR COMMENTS. RFC 4566: *SDP*: Session Description Protocol. Disponível em: < https://tools.ietf.org/html/rfc4566> Acesso em: 27 jul. 2016.

REQUEST FOR COMMENTS. RFC 1889: *RTP*: A Transport Protocol for Real-Time Applications. Disponível em: < https://www.ietf.org/rfc/rfc1889.txt> Acesso em: 27 jul. 2016.

UMA PROPOSTA PARA DETECÇÃO E RESOLUÇÃO DE CONFLITOS DE POLÍTICAS EM REDES DEFINIDAS POR *SOFTWARE*

A PROPOSAL FOR DETECTION AND RESOLUTION OF POLICY CONFLICTS IN SOFTWARE-DEFINED NETWORKING

MANUELA TIRLONI[1], FELIPE TOMM[1], LUCAS F. CLARO[1], CRISTIAN C. MACHADO[1]

[1]Departamento de Engenharias e Ciência da Computação, Universidade Regional Integrada do Alto Uruguai e das Missões, URI - Câmpus de Frederico Westphalen. *E-mail: inf23847@uri.edu.br.

Resumo: O paradigma de Redes Definidas por Software (*Software-Defined Networking* - SDN) simplifica o gerenciamento de rede através da remoção de parte da lógica de tomada de decisões sob o processamento de tráfego que é realizada nos elementos de comutação, tais como, *switches* e roteadores, centralizando essa tarefa em um componente chamado controlador de rede. Assim, SDN apresenta um ambiente propício para aplicar abordagens utilizadas para reduzir a complexidade nas tarefas de gerenciamento, tais como Gerenciamento Baseado em Políticas (*Policy Based Management* – PBM). Esse tipo de abordagem proporciona a possibilidade de que um conjunto de regras (políticas) sejam escritas para serem aplicadas em diversos dispositivos, tais como *switches* e roteadores. Entretanto, PBM não garante que tais regras sejam aplicadas sem sobreposição e que garantam que o sistema seja conduzido de maneira consistente e eficiente, conforme esperado. Neste contexto, o presente trabalho tem como objetivo apresentar o projeto de uma ferramenta capaz de detectar e propor soluções para resolver os conflitos entre as políticas de rede.

Palavras-chave: Conflito. Detecção. Políticas. Redes Definidas por Software.,

Abstract: The paradigm of Software-Defined Networking (SDN) simplifies the network management by removing part of decision-making logic on the processing of traffic that is carried out in the network devices, such as switches and routers, centralizing this task in a component called network controller. As a result, SDN offers an environment to deploy approaches to reduce complexity in management tasks, such as Policy-Based Management (PBM) approaches. PBM provides the possibility that a set of rules (policies) are written to be applied in several devices, such as switches and routers. However, PBM does not guarantee that such rules are applied without overlapping and to ensure that the system be conducted in a consistent and efficient manner, as expected. In this context, this paper aims to present the design of a tool capable to detect and propose solutions to solve the conflicts between network policies.

Keywords: Conflict. Detection. Policies. Software-Defined Networking.

1 INTRODUÇÃO

Por muitos anos, as organizações têm desenvolvido estratégias de gestão para lidar com a escalabilidade e complexidade computacional em infraestruturas de Tecnologia da Informação e Comunicação (TIC) (FITO et al. 2012). O surgimento do paradigma de Redes Definidas por *Software* (*Software-Defined Networking* - SDN) (NUNES *et al.* 2014) tem como objetivo fornecer uma arquitetura mais sofisticada e precisa para o gerenciamento e monitoramento de tráfego de rede. Para conseguir isso, SDN simplifica o gerenciamento de rede através da remoção de parte da lógica de tomada de decisões sob o processamento de tráfego que é realizada nos elementos de comutação, tais como, *switches* e roteadores, centralizando essa tarefa em um componente chamado controlador de rede. Assim, os elementos de comutação se tornam simples dispositivos de encaminhamento de pacotes, e os controladores podem ter uma visão global do tráfego de rede e de todos os elementos que a compõe. Como resultado, SDN permite uma forma eficaz para fornecer de forma dinâmica - e em tempo de execução - serviços que suportam, por exemplo, reconfiguração de Qualidade de Serviço (*Quality of Service* - QoS), controle de acesso, e balanceamento de carga (MACHADO *et al*. 2014).

Adicionalmente, pela clara separação dos planos em SDN e pela simplificação e alocação de parte da lógica de decisão no controlador, SDN apresenta um cenário propício para aplicar abordagens utilizadas no passado a fim de reduzir a complexidade nas tarefas de gerenciamento, tais como Gerenciamento Baseado em Políticas (*Policy Based Management* - PBM). Em sistemas PBM um administrador especifica os objetivos/metas e restrições esperadas da infraestrutura na forma de regras para orientar o comportamento dos elementos em um sistema. Esse tipo de abordagem proporciona a possibilidade de que um conjunto de regras sejam escritas para atender diversos elementos. Entretanto, esse agrupamento de regras não garante que as mesmas sejam aplicadas sem sobreposição e a garantia de que o sistema seja conduzido de maneira consistente e eficiente, conforme esperado. Assim, para alcançar ou

pelo menos aliviar esse problema, uma solução é o uso de técnicas para a detecção e resolução destes possíveis conflitos.

Neste contexto, este artigo apresenta um projeto que propõe desenvolver um conjunto de ferramentas para detecção e resolução de conflitos de políticas em SDN. Esta é a primeira vez que conflitos de políticas em redes definidas por *software* é apresentado.

Em suma, esperam-se deste projeto os seguintes resultados:

- Monitoramento de conflitos de políticas no nível de rede que podem interferir no desempenho de sistemas.
- Análise de conflitos de políticas com mínima intervenção humana.
- Diminuição da quantidade de regras de rede codificadas em cada *switch*.
- Diminuição da quantidade de regras de rede gerenciadas pelo controlador.
- Otimização e consistência no encaminhamento de fluxos a partir do estabelecimento de regras que não sobrepõem outras regras.

O restante deste artigo está dividido da seguinte maneira: Na seção 2 será apresentada a contextualização geral deste projeto. A proposta será detalhada na seção 3. Por fim, na seção 5 serão apresentados os resultados esperados e uma conclusão.

2 CONTEXTUALIZAÇÃO

2.1 Redes Definidas por Software (Software-Defined Networking - SDN)

Redes Definidas por *Software* (*Software-Defined Networking* – SDN) é uma arquitetura de rede dinâmica, adaptável, controlável, e flexível para a entrega de serviços de rede, capaz de responder rapidamente às mudanças de requisitos de serviço (OPEN NETWORKING FOUNDATION 2014). Uma arquitetura SDN, conforme representada na Fig. 1, compreende quatro planos: gerenciamento, aplicação, controle e dados (NUNES et al. 2014). O plano de gerenciamento inclui sistemas de gestão que exercem as funções e operações de apoio à infraestrutura, por exemplo, acordos de nível de serviço (*Service Level Agreements* – SLAs) e as políticas de baixo nível para conduzir aplicações e controladores SDN. O plano de aplicação inclui aplicações SDN (por exemplo, *firewalls* e balanceadores de carga), aplicações de negócios (por exemplo, portais *e-commerce* e sistemas de gestão empresarial) ou sistemas de Orquestração de Nuvens (por exemplo, OpenStack e CloudStack). Cada aplicativo tem controle exclusivo de um conjunto de recursos fornecidos pelos controladores SDN. O plano de controle é responsável pelos protocolos e pela tomada de decisões que resultam na atualização das tabelas de encaminhamento dos *switches* e roteadores. Por fim, o plano de dados, conhecido como o plano de encaminhamento, administra a comutação e roteamento de pacotes de fluxo.

Em redes IP tradicionais, o plano de controle é executado em cada dispositivo de rede. Cada dispositivo tem seus protocolos proprietários, o que torna difícil sua programação. Muitas vezes não é possível realizar o processo de tomada de decisão sobre eventos que não tenham sido previstos. Diferentemente, SDN é caracterizado por um plano de controle logicamente centralizado, o que permite que parte da lógica de tomada de decisão realizada pelos dispositivos de rede seja movida para controladores externos. Essa abordagem fornece aos dispositivos controladores a capacidade de ter uma visão global da rede e seus recursos, tornando-se cientes de todos os elementos da rede e suas características. Com base nesta centralização, dispositivos de rede tornam-se simples elementos de encaminhamento de pacotes, podendo ser programados através de uma interface aberta, como o protocolo OpenFlow (NUNES et al. 2014).

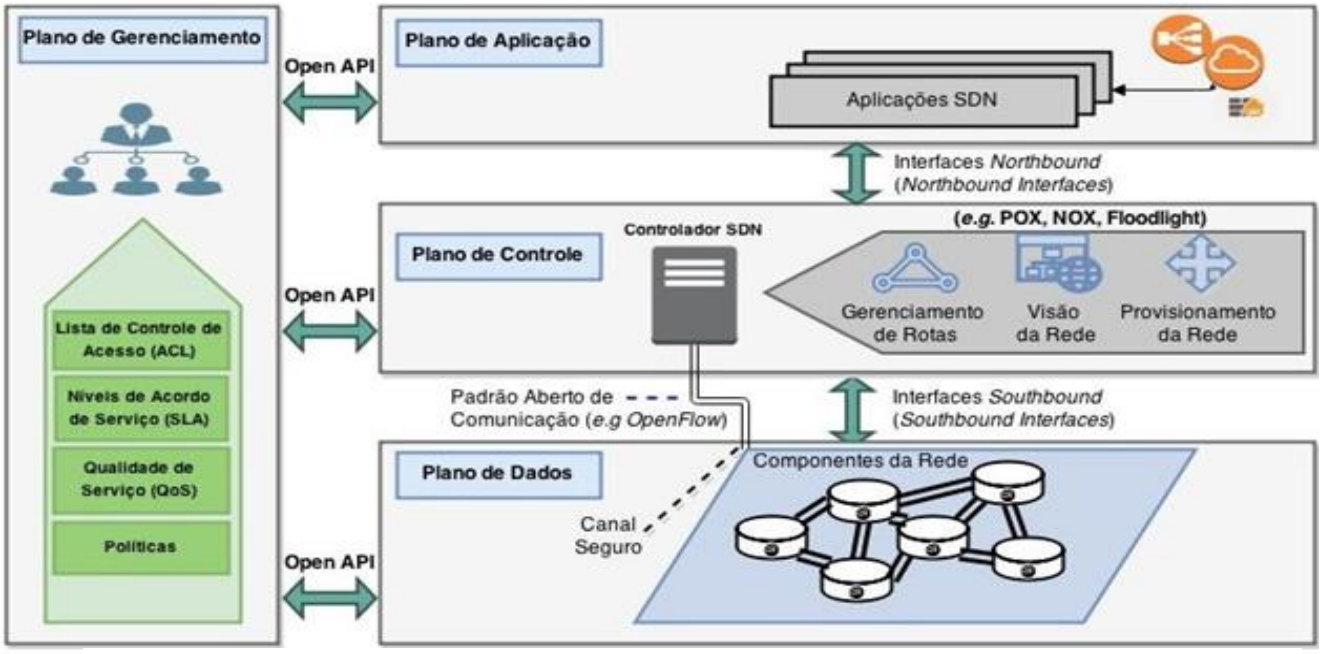


Fig. 1. Arquitetura de uma rede SDN. Fonte: Adaptado de OPEN NETWORK FUNDATION (2014).

ISBN: 978-85-7796-213-6 <http://ocs.fw.uri.br/index.php/simcit/>

O OpenFlow é um protocolo aberto que permite o desenvolvimento de mecanismos programáveis com base em uma tabela de fluxo padrão localizada nos dispositivos de encaminhamento.

2.2 Gerenciamento Baseado em Políticas (Policy-Based Management – PBM)

Políticas são definidas como um conjunto de regras que expressam e reforçam o comportamento exigido de um recurso. A RFC 3198 (WESTERINEN et al. 2001) fornece as seguintes definições para política:

- A meta ou ação definida que determina como as decisões presentes e futuras sejam tomadas. As políticas são estabelecidas ou executadas dentro de um contexto particular;
- Políticas referem-se a um conjunto de regras para gerenciar e monitorar o acesso a recursos de uma infraestrutura de TIC em especial.

Em sistemas para gerenciamento baseado em políticas (*Policy-Based Management* – PBM) um administrador especifica os objetivos/metas e restrições em forma de regras para orientar o comportamento da infraestrutura (VERMA et al. 2002). O uso de PBM apresenta três benefícios principais (HAN *et al.* 2012). Em primeiro lugar, as políticas são pré-definidas pelos administradores e armazenadas em um repositório. Quando ocorre um evento, essas políticas são solicitadas e acessadas automaticamente, sem a necessidade de intervenção manual. Em segundo lugar, a descrição formal de políticas permite a análise automatizada e verificação com o objetivo de garantir a coerência, em certa medida. Em terceiro lugar, por causa da abstração de detalhes técnicos, as políticas podem ser verificadas e alteradas de forma dinâmica em tempo de execução sem modificar a implementação do sistema.

Segundo Moffett et al. (1993), políticas podem ser vistas em dois principais níveis de abstração: políticas de baixo nível, que estão relacionados a um domínio ou um dispositivo e políticas de alto nível que são mais amigáveis ao usuário. Um exemplo simples de uma política de baixo nível é a configuração em roteadores para que os pacotes de tráfego multimídia tenham prioridade sobre pacotes de tráfego web. Um exemplo de uma política de alto nível é um acordo de nível de serviço (*Service Level Agreement* – SLA).

2.1.1 Entidades básicas

Waller *et al.* (2011) introduz quatro entidades básicas para modelar a arquitetura de um sistema baseado em políticas, assim como demonstra a Fig. 2.

- **Ferramenta de Gerenciamento de Políticas (*Policy Management Toolkit* – PMT) –** permite que o administrador gerencie políticas;
- **Repositório de Política (*Policy Repository* – PR) –** armazena informações relacionados com a política;
- **Ponto de Decisão da Política (*Policy Decision Point* – PDP) –** pesquisa, verifica e valida às condições necessárias para as políticas;
- **Ponto de Aplicação da Política (*Policy Enforcement Point* – PEP) –** executa e monitorapolíticas também fornecendo *feedbacks* de informações relevantes durante a execução.

A chave desta arquitetura é o PDP. PDP exerce uma parte importante do processamento de controle do sistema. Nesta entidade, as políticas de alto nível são traduzidas em ações compreendidas pelos elementos do sistema, e permanecem aguardando em algum momento ser executadas nos PEPs. Para a operação correta, o PDP deve diferenciar cada detalhe de cada PEP no sistema para proporcionar maior precisão em cada política.

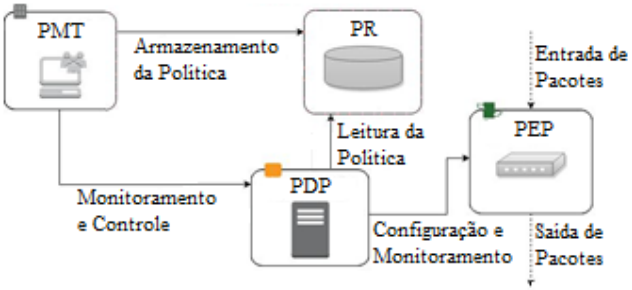


Fig. 2. Arquitetura básica de sistemas PBM. Fonte: Adaptado de STRASSNER (2003).

Neste tipo de arquitetura, os administradores definem políticas de gerenciamento que estão inseridos no PR -, por exemplo, um *Lightweight Directory Access Protocol* (LDAP), - através de um PMT. Depois disso, a PDP realiza o monitoramento de eventos no sistema, de acordo com as configurações estabelecidas pelo administrador. Quando ocorrer um evento específico, o PDP será acionado para recuperar do PR as políticas aplicáveis a cada caso individual. Para cada política recuperada do evento específico, quando estiverem reunidas as condições específicas, as ações correspondentes são aplicadas pelo PEP associado ao elemento monitorado.

2.3 Detecção e Resolução de Conflitos

A detecção e resolução de conflitos entre políticas é fundamental para o projeto de um sistema de gerenciamento baseado em políticas confiável, escalável e implementável. O conceito de detecção de política é bastante simples, no entanto, a implementação de resolução de conflitos entre política pode ser bastante complicado (STRASSNER et al. 2013).

Resumidamente, conflitos são apresentados em três modalidades:

- Obrigação Positiva / Obrigação Negativa – as ações são ambas obrigadas e não obrigadas a serem realizadas sobre os elementos.
- Autorização Permitida / Autorização Proibida – as ações são ambas permitidas e proibidas a serem realizadas sobre os elementos.
- Obrigação Positiva / Autorização Proibida – as ações são necessárias, mas proibidas de serem realizadas sobre os objetos.

Em geral, sempre que várias políticas se aplicam a um elemento, há um potencial para alguma forma de conflito. Porém, é essencial que várias políticas sejam aplicadas de forma a abranger a diversidade de funções de gerenciamento de forma mais abrangente possível para

que o objetivo principal de políticas seja alcançado, o de diminuir ou retirar o trabalho manual dos administradores no gerenciamento dos elementos de uma infraestrutura de TIC.

Quanto à detecção de conflito, o objetivo fundamental é analisar especificações de política, a fim de proporcionar um perfil de um dos tipos de conflito que podem ocorrer dentro de um sistema. Os objetivos de detecção de conflito são caracterizados da seguinte forma:

- Para identificar o conflito real que ocorreu e que poderia ter sido resolvido estaticamente em tempo de compilação.
- Para prever que um conflito poderá ocorrer no futuro, visualizando mais especificamente, quais as circunstâncias irão expor o conflito.
- Para comunicar o conflito real ou potencial a um processo de resolução.

A partir da análise das especificações para detecção de conflitos, o objetivo fundamental da resolução de conflitos é determinar quando é apropriado para resolver o conflito e como o conflito será resolvido. As metas de resolução de conflitos podem ser caracterizadas da seguinte forma:

- Para comunicar com o processo de detecção de conflito, a fim de obter as especificações do conflito efetivo ou potencial que ocorrer em um sistema.
- Para decidir quando é apropriado resolver o conflito.
- Para identificados e monitorar potenciais conflitos que podem ocorrer, necessitando de resolução.
- Para decidir como resolver o conflito real ou potencial, de forma adequada.

3 Proposta

Este artigo apresenta a proposta de desenvolvimento de um conjunto de ferramentas cujo objetivo principal é a detecção e a resolução de conflitos de políticas em redes definidas por *software*, utilizando o paradigma de gerenciamento baseado em políticas e os recursos e benefícios oferecidos por redes definidas por *software*.

Primeiramente, será realizado o planejamento, definindo o escopo dos sistemas e seus requisitos e modelando o banco de dados de acordo com as características do problema.

Em um segundo momento será desenvolvido um sistema de coleta de informações da rede para melhorar o processo de detecção e resolução de conflitos de políticas de baixo nível. Dentre as informações coletas encontram-se as máscaras geradas para cada regra, que possuem informações tais como portas de origem e destino, protocolos, prioridades, entre outras.

Em seguida será feita a análise de técnicas existentes para a detecção de conflitos de políticas de baixo nível e implementação de melhorias baseadas nas características e recursos oferecidos por redes definidas por *software*.

Logo após será desenvolvido um sistema de resolução de conflitos de políticas de baixo nível. O sistema terá uma interface amigável para identificar cada conflito e apresentar ao menos uma ou um conjunto de possíveis soluções.

Por fim, utilizando diferentes cenários e fazendo comparações com um grupo de problemas-testes, será efetuada a avaliação de desempenho dos sistemas, visando
desta forma demonstrar os resultados propostos neste projeto.

4 Resultados Esperados E Conclusão

Com este projeto pretende-se desenvolver um conjunto de ferramentas para detecção e resolução de políticas em redes definidas por *software* conforme mencionados na seção anterior. Os resultados do projeto servirão de base para aplicabilidade da solução em ambientes reais como provedores de Internet e de computação em nuvem, operadoras de Telecom, dentre outros. Assim, acredita-se que a pesquisa será de grande valia, pois será possível aplicar a solução para a resolução de problemas reais gerando economia de recursos, tanto físico quanto humano, e proporcionando ferramentas e sistemas mais eficientes.

Agradecimentos

Agradecemos ao Programa Institucional de Bolsas de Iniciação Científica do Edital/PROPEPG/PIIC/URI N° 03/2016 que auxilia essa pesquisa através do projeto #3608 - Detecção e resolução de conflitos de políticas em redes definidas por software

Referências

FITO, J. O. et al. Business-driven it management for cloud computing providers. In: 4TH IEEE International Conference on Cloud Computing Technology and Science, 2012, Taipei, China. *Proceedings*... Taipei: IEEE, 2012. v. 4, p. 193–200.

NUNES, B. *et al. A survey of software-defined networking: Past, present, and future of programmable networks.* Communications Surveys Tutorials, IEEE, v. 16, n. 3, p.1617–1634, Fevereiro 2014. ISSN 1553-877X.

MACHADO, C. C.; GRANVILLE, L. Z.; SCHAEFFER-FILHO, A.; WICKBOLDT, J. A. *Towards SLA Policy Refinement for QoS Management in Software-Defined Networking, Advanced Information Networking and Applications (AINA)*, 2014 IEEE 28th International Conference on , vol., no., pp.397,404, 13-16 Maio 2014. doi: 10.1109/AINA.2014.148.

OPEN NETWORK FUNDATION *SDN Architecture Overview*. 2014. Disponível em: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN-ARCH-Overview-1.1-11112014.02.pdf>. Acesso em: 4 ago. 2016.

WESTERINEN, A. et al. *Terminology for policy-based management (RFC 3198)*. IETF Request for Comments - Network Working Group, Southborough, USA, v.1, n. 1, p. 1–21, nov. 2001. Disponível em: <https://www.ietf.org/rfc/rfc3198.txt>. Acesso em: 1 ago. 2016.

VERMA, D. *Simplifying network administration using policy-based management.* Network, IEEE, v. 16, n. 2, p. 20–26, Março 2002. ISSN 0890-8044.

HAN, W.; LEI, C. *A survey on policy languages in network and security management.* Computer Networks, v. 56, n. 1, p. 477 – 489, 2012. ISSN 1389-1286.

MOFFETT, J. *et al. Policy hierarchies for distributed systems Management.* Selected Areas in Communications, IEEE Journal on, vol.11, no. 9, pp. 1404–1414, 1993.

WALLER, A. *et al. Policy based management for security in cloud computing.* In: Secure and Trust Computing, Data Management, and Applications. [S.l.]: Springer, 2011. p. 130–137.

STRASSNER, John. *Policy-based network management: solutions for the next generation*. Morgan Kaufmann, 2003.

STRASSNER, J. *et al. Terminology for policy-based management.* RFC Editor, Tech. Rep., 2001.

SIGECA: SISTEMA DE GERENCIAMENTO DO CONSUMO DE ÁGUA DE RESIDÊNCIAS

SIGECA: MANAGEMENT SYSTEM OF RESIDENTIAL WATER CONSUMPTION

MAURICIO FELIPE SOARES[1]*, MAURÍCIO SULZBACH[1], ANDRÉ LUÍS STEFANELLO[1]

[1]Departamento de Engenharias e Ciência da Computação, Universidade Regional Integrada do Alto Uruguai e das Missões, URI - Câmpus de Frederico Westphalen.
*E-mail: inf25913@uri.edu.br.

Resumo: Em meio a tantas campanhas sobre a escassez da água e sobre a sua utilização consciente, o surgimento de uma ferramenta que possibilite o controle do consumo de água de uma residência, bem com o planejamento de sua utilização consciente, pode ser um importante instrumento que gere economia financeira e do recurso utilizado. Nesse sentido, este trabalho apresenta o SiGeCA – Sistema de Gerenciamento de Consumo de Água de Residências, um sistema web que possibilita ao usuário consultar o consumo de água de diversos pontos de uma residência, como torneiras e chuveiros, no momento de sua utilização ou após a sua utilização, sendo uma ferramenta de acompanhamento, conscientização sobre o consumo e de auxilio na redução da manutenção de uma residência. Através do Arduino e de sensores de fluxo de água, o sistema armazena em um banco de dados as informações do consumo de água, possibilitando ao usuário, acompanhar essas informações através do sistema web, desenvolvido utilizando a linguagem Java.

Palavras-chave: SiGeCA. Monitoramento do consumo de água. Utilização consciente. Arduino.

Abstract: In the midst of so many campaigns on water scarcity and on their conscious use, the appearance of a tool that allows control of the water consumption of a residence, as well as planning your conscious use can be an important tool that manages financial economy and of resources used. In this sense, this work presents the SiGeCA - System of Management of the Consumption of Water of Residency, a web system that enables the user check the water consumption from different points of a residence, as taps and shower heads, at the time of use or after its use, as a monitoring tool, awareness of the consumer and aid in reducing maintaining a residence. Through the Arduino and of water flow sensors, the system stores in a database a water consumption information, allowing to user, follow the information through the web system developed using the Java language.

Keywords: SiGeCA. Monitoring of water consumption. Conscious use. Arduino.

1 INTRODUÇÃO

O aumento da população mundial e a falta de um consumo consciente tem gerado uma demanda cada vez maior por água, tanto para o consumo, quanto para a geração de energia elétrica. Segundo dados de pesquisa da ONU (Organização das Nações Unidas), em 2050 serão 10 bilhões de pessoas existentes no mundo, o que torna um motivo preocupante em relação ao consumo consciente (VICTORINO, 2007). Em muitos países há escassez de água até mesmo para o consumo humano, quanto mais para gerar energia elétrica. Atualmente, existem formas alternativas para conseguir água potável e gerar energia elétrica, porém são processos caros se comparados aos recursos naturais disponíveis.

Nos dias atuais, o consumidor não consegue saber o consumo de água de uma residência, a não ser quando a sua conta chega. Esse fato dificulta na diminuição do consumo e também na conscientização da economia financeira e do recurso utilizado. Possibilitar o acompanhamento no momento e após a utilização de uma torneira ou de um chuveiro, por exemplo, pode ser um importante instrumento na redução do consumo de um dos bens mais importantes para os humanos.

Nesse sentido, o presente artigo irá apresentar o SiGeCA – Sistema de Gerenciamento de Consumo de Água de Residências, uma ferramenta de monitoramento do consumo de água, que visa manter os consumidores informados sobre os gastos de cada componente em sua residência, tornando assim uma ferramenta aliada à redução do consumo. Para simular o funcionamento do sistema foi construída uma maquete, onde o Arduino, s*hields*, sensores e demais equipamentos foram dispostos, programados e testados.

O presente artigo está assim estruturado. A seção dois apresenta a fundamentação teórica, importante para a compreensão dos recursos utilizados. A seção três detalha o desenvolvimento do sistema proposto e por fim a seção quatro apresenta as conclusões.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão detalhados os referenciais sobre as tecnologias utilizadas no desenvolvimento do SiGeCA.

Arduino.

O Arduino é uma placa de microcontrolador baseado no ATmega 328P. Ele contém 14 pinos digitais de entrada e saída, 6 entradas analógicas, conexão USB, um cabeçalho ICSP e um botão de *reset*. Pode-se usar a conexão USB para troca de dados e também para alimentação da placa. Além disso, pode ser usada uma bateria como fonte de alimentação para a placa. A programação para o Arduino é baseada na linguagem Wiring (semelhante ao C/C++) e através da IDE Arduino Software. A figura 1 apresenta uma placa Arduino, modelo UNO (ARDUINO, 2015).

Fig. 1. Placa Arduino UNO (adaptado de ARDUINO, 2015).

O modelo UNO é a primeira de uma série de placas Arduino USB, sendo o modelo de referência para a plataforma Arduino. "UNO" em italiano significa “UMA” e foi escolhido para marcar o lançamento do Arduino Software 1.0 (ARDUINO, 2l015).

Em suma, Arduino pode ser considerado como um pequeno computador que pode ser incluída uma programação para gerenciar as entradas e saídas dos componentes ligados a ele. Também pode ser definido como uma plataforma de computação embarcada. Através do Arduino é possível desenvolver qualquer projeto, desde um simples acendimento de uma lâmpada ou *led* por um determinado período, ou soluções complexas, como por exemplo, as automações residenciais e industriais. A usabilidade da arquitetura Arduino é principalmente em projetos com objetos interativos independentes, que podem ou não estarem conectados a um computador, a uma rede ou a uma conexão direta com a internet, possibilitando a recuperação e envio de dados do Arduino (MCROBERTS, 2015).

2.1 Arduino Ethernet Shield

O Arduino *Ethernet Shield*, apresentado pela figura 2, é um acessório que possibilita a conexão da placa Arduino com a internet. Seu código é totalmente aberto, possibilitando assim mudanças em seu funcionamento.

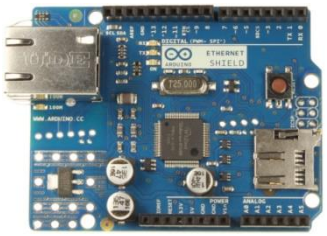


Fig. 2. Arduino Ethernet Shild (Adaptado de ARDUINO SHIELD, 2016).

A Ethernet Shield é baseada no *chip Ethernet Wiznet* W5100. O *Wiznet* W5100 fornece suporte para as redes TCP e UDP. Esse *shield* tem suporte de até quatro conexões de soquete simultâneo. É conectada a uma placa Arduino usando pinos que se estendem através do *shield*. Também possibilita a extensão de mais outros *shields* que podem ser acoplados na parte superior. (ARDUINO SHIELD, 2016).

2.2 Sensor de Fluxo de Água

Para medir o consumo de água, este trabalho utilizou o sensor de fluxo de água SF-201, apresentado pela figura 3. Ele é usado como medidor de fluxo básico, mas pode ser destinado a inúmeros projetos, dependendo da necessidade exigida, sendo necessário estar conectado a uma corrente de líquido. O SF-201 usa um sensor do tipo cata-vento para medir a quantidade de líquido que foi movida pelo sensor. O cata-vento tem um pequeno ímã preso, e há um sensor magnético de efeito *hall* do outro lado do tubo de plástico que pode medir quantos giros o cata-vento executou.

Fig. 3. Sensor de Fluxo de Água (adaptado de FILIPEFLOP, 2016).

O sensor vem com três fios: vermelho (*power*), preto (terra) e amarelo (saída de pulso). Ao contar os pulsos a partir da saída do sensor, pode-se facilmente acompanhar o movimento de fluídos. Cada pulso é de aproximadamente 2,25 mililitros de líquido que passou pelo sensor (ADAFRUIT, 2016).

2.3 Linguagem Wiring

Wiring é definido como um framework de programação, de código aberto, utilizado em microcontroladores. Através de Wiring é possível o controle de dispositivos conectados às placas de microcontroladores, possibilitando a interação entre diferentes objetos. Milhares de estudantes, pesquisadores e amadores utilizam o Wiring para a aprendizagem, criação de protótipos e até mesmo trabalhos profissionais (WIRING, 2016).

ISBN: 978-85-7796-213-6 <http://ocs.fw.uri.br/index.php/simcit/>

2.4 Linguagem Java

Java é uma tecnologia criada pela Sun Microsystems em 1995 com o objetivo de desenvolver uma nova plataforma para a computação interativa. A linguagem de programação Java representa uma linguagem simples, orientada a objetos (MENDES, 2009). A tecnologia Java é utilizada para criar páginas Web com conteúdo interativo e dinâmico, para desenvolver aplicativos corporativos de grande porte, para fornecer aplicativos para dispositivos destinados ao consumidor final, tais como celulares e *tablets*, entre outros (DEITEL, 2003).

3 DESENVOLVIMENTO DO SIGECA

As seções a seguir detalharão o desenvolvimento do sistema proposto neste trabalho.

3.1 Funcionamento da aplicação

A aplicação embarcada no sistema Arduino é simples e de fácil compreensão. Primeiramente, foi realizada a configuração e montagem dos componentes do sistema. Na programação do sistema, a placa periodicamente faz uma verificação se existem pulsos nos sensores. Quando um pulso é detectado no sensor, os dados do mesmo são enviados para a placa Arduino. Na placa são efetuados todos os cálculos de vazão e as informações são armazenadas no banco de dados. A figura 4 apresenta o diagrama de atividade da aplicação embarcada na plataforma Arduino.

Fig.4. Diagrama de atividade da aplicação embarcada na plataforma Arduino.

Já a aplicação web é uma ferramenta que possibilita ao usuário, através de qualquer dispositivo conectado à rede local, visualizar os dados do consumo de água. Através da figura 5, é possível visualizar como o usuário pode administrar a ferramenta. Inicialmente o usuário acessa a aplicação, podendo ser através de qualquer dispositivo conectado a rede local, informa qual o equipamento gostaria de verificar ou consumo (ou de todos, deixando em branco essa opção) e o período que deseja visualizar o consumo. Ao executar a operação pesquisar, o usuário recebe os dados buscados no banco de dados que o sistema embarcado armazenou.

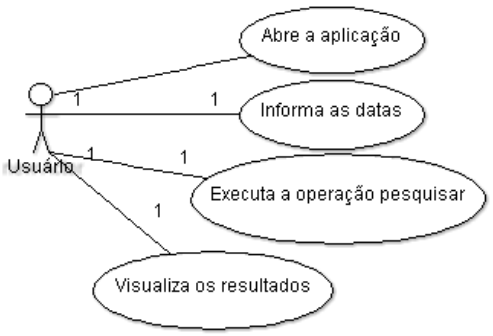


Fig.5. Diagrama de caso de uso.

3.2 Desenvolvimento do Banco de Dados

No desenvolvimento deste sistema foi escolhido o banco de dados MySQL para o armazenamento e consulta das informações do consumo de água. A escolha se deu principalmente por ser uma ferramenta livre e que atende às exigências da proposta. A figura 6 apresenta o *script* utilizado para a criação do banco de dados e a criação da tabela *sensores*.

```
1  CREATE DATABASE PROJETOARDUINO;
2  USE PROJETOARDUINO;
3
4  CREATE TABLE SENSORES(
5      ID INT PRIMARY KEY,
6      TIPO_SENSOR VARCHAR(255) NOT NULL,
7      RECORDED TIMESTAMP NOT NULL,
8      TOTAL_MLS FLOAT NOT NULL,
9      TIPO_EQUIPAMENTO VARCHAR(255) NOT NULL
10 )
```

Fig. 6. Script de criação do banco de dados e tabela sensores.

A tabela *sensores* armazenará um *ID* de cada inserção realizada no banco de dados, além do nome do sensor que originou as informações, a data/hora da inserção, o total de mililitros de água passados pelo sensor e o tipo do equipamento que o sensor está medindo (torneira, chuveiro, etc.).

3.3 Desenvolvimento do sistema embarcado na plataforma Arduino

O sistema embarcado na plataforma Arduino é o responsável pela leitura e comunicação dos dados dos sensores e seu correto armazenamento no banco de dados. Para a implementação do sistema foi unida a placa Arduino UNO com a *Shield* de *Ethernet*, sendo realizada a conexão via *jumpers* de todo o sistema na *protoboard*. Para enviar os dados dos sensores para o banco de dados foi usado um cabo de *Ethernet*, ligado ao computador. Por fim, foi realizado o *upload* do código de configuração, tanto da placa Arduino UNO, como da *Ethernet Shield*.

Esse código tem as funções responsáveis para realizar a comunicação entre os componentes, fazer as leituras do sensor, realizar cálculos de vazão, contagem dos pulsos e posteriormente armazenar as informações no banco de dados, via conexão *Ethernet*. Toda vez que a placa detectar pulsos vindos do sensor, o mesmo realiza os

ISBN: 978-85-7796-213-6 <http://ocs.fw.uri.br/index.php/simcit/>

cálculos e armazena a quantidade de litros no banco de dados.

A figura 7 demonstra uma parte do código enviado à placa Arduino.

```
1  void setup()
2  {
3      Ethernet.begin(mac, ip, gateway, subnet);
4      delay(500);
5      Serial.begin(38400);
6
7      Serial.println("Conectando...");
8      if (my_conn.mysql_connect(ip_server, mysqlPort, user, password)){
9        sqlconnect=true;
10       Serial.println("Conectado!");
11       delay(1);
```

Fig. 7. Código de teste de conexão com o banco de dados.

Na linha 8 da figura 7 é realizada a verificação da conexão da placa com o banco de dados. Em seguida, se a conexão estiver correta é exibida a mensagem "Conectado!", linha 10. Se a comunicação estiver correta são executadas as instruções restantes.

Já a figura 8 apresenta parte do código que realiza o cálculo da vazão de líquidos do sensor de fluxo de água.

```
1  if((millis() - oldTime) > 1000)
2   {
3     detachInterrupt(sensorInterrupt);
4     flowRate = ((1000/(millis() - oldTime))* pulseCount)/ calibrationFactor;
5     oldTime = millis();
6     flowMilliLitres = (flowRate / 60) * 1000;
7     totalMilliLitres += flowMilliLitres;
8
9     unsigned int frac;
10    Serial.print("Taxa de Fluxo: ");
11    Serial.print(int(flowRate));
12    Serial.print(".");
13
14    frac = (flowRate - int(flowRate)) * 10;
15    Serial.print(frac, DEC) ;
16    Serial.print("L/min");
17    Serial.print("  Liquido Corrente: ");
18    Serial.print(flowMilliLitres);
19    Serial.print("mL/Sec");
20
21    Serial.print(" Qantidade de Liquido Saido: ");
22    Serial.print(totalMilliLitres);
23    Serial.println("mL");
```

Fig. 8. Código do cálculo da vazão.

Entre as linhas 3 e 7, da figura 8, é efetuado o cálculo da taxa de fluxo, que será armazenada na variável *FlowRate*, linha 4. Também se define o cálculo do líquido corrente e o total de líquido consumido, nas linhas 6 e 7.

Na figura 9 tem-se um trecho do código que verifica se existem pulsos no sensor de fluxo de água.

```
1  if(pulseCounter != 0){
2       if (sqlconnect == true){
3         sprintf(INSERT_SENSOR,"INSERT INTO projetoarduino.sensores
4         " VALUES (null,'Sensor de agua',null,%d)",totalMilliLitres
5         my_conn.cmd_query(INSERT_SENSOR);
6
7     pulseCount = 0;
```

Fig. 9. Código do script de inserção no banco de dados.

No momento que é identificado algum pulso no sensor são executadas as instruções dentro do "if" da linha 1. Na linha 2 é verificada se a conexão com o banco está ativa e estando, é efetuada a inserção dos dados no banco, conforme as instruções das linhas 3 a 5.

3.4 Desenvolvimento do sistema web para consulta do consumo de água

Para o cumprimento deste objetivo foi desenvolvida uma ferramenta, através da linguagem Java, para possibilitar ao usuário a visualização dos dados em tempo de consumo, ou posterior ao consumo.

A primeira etapa foi a criação da conexão com o banco de dados MySQL, como ilustra a figura 10.

```
1  Class.forName("com.mysql.jdbc.Driver");
2  Connection con = DriverManager.getConnection("jdbc:mysql:"+
3  "//localhost:3307/projetoarduino","root","usbw");
4  Statement st = con.createStatement(
5      ResultSet.TYPE_SCROLL_SENSITIVE,
6      ResultSet.CONCUR_READ_ONLY
7  );
8  ResultSet rs;
9
```

Fig. 10. Código fonte da conexão da página com o banco de dados.

Na figura 10, pode-se visualizar primeiramente o carregamento do *driver* JDBC, linha 1, responsável pela conexão com o banco de dados. Posteriormente a aplicação efetua a conexão e a partir do *driver* carregado na linha 2 e 3, usando além da porta 3307, o nome do banco de dados, o usuário e a senha para ter acesso aos dados do banco.

Através de uma consulta SQL no banco de dados, o resultado é mostrado na página web, conforme demonstra a figura 11. A aplicação possibilita, via internet, ao usuário o acesso das informações que foram enviadas, pelo sistema embarcado, ao banco de dados. Para realizar a implementação da página web foi usada a ferramenta Netbeans, intercalando entre Linguagem HTML, Java e instruções SQL.

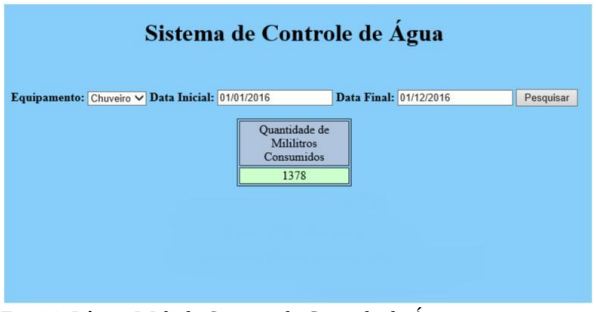


Fig. 11. Página Web do Sistema de Controle de Água.

Na página web, o usuário pode selecionar o tipo de equipamento (torneira, chuveiro, ou outro) e o período inicial e final que se quer saber o consumo. Para simular o funcionamento do sistema foi construída uma maquete, onde o Arduino, s*hields*, sensores e demais equipamentos foram dispostos, programados e testados.

4 CONCLUSÃO

É papel fundamental do cientista da computação saber a importância do uso das novas tecnologias para a criação de uma nova cultura na sociedade, uma cultura de conscientização. Através das tecnologias disponíveis atualmente, é possível criar diversos tipos de aplicações que facilitem a vida das pessoas e da sociedade onde elas vivem. Um exemplo disso é a tecnologia Arduino, que possibilita o desenvolvimento de muitos projetos de automação que podem ser utilizados tanto no meio residencial como no empresarial.

A presente proposta objetivou apresentar o SiGeCA, um sistema para consulta do consumo de água de uma

ISBN: 978-85-7796-213-6 <http://ocs.fw.uri.br/index.php/simcit/>

residência, que objetiva mostrar ao usuário o consumo no momento da utilização ou pós-utilização dos pontos de água (torneiras, chuveiros, etc.) de uma residência. Através desta ferramenta, pode-se ter a informação a qualquer momento, de uma forma simples e prática, tornando-a importante na conscientização e na redução do consumo de água de uma residência, e consequentemente resultando em uma economia financeira.

Como trabalhos futuros, está sendo desenvolvida a segunda etapa deste projeto, que contempla o acompanhamento do consumo de energia elétrica dos eletrodomésticos de uma residência. Para isso, será desenvolvida na aplicação embarcada na plataforma Arduino as instruções vindas dos sensores de corrente elétrica, sendo essas armazenadas no banco de dados. Posteriormente, será aperfeiçoada a aplicação web de consulta das informações, para que os dados do consumo dos eletrodomésticos possam ser obtidos no sistema

REFERÊNCIAS

ARDUINO UNO. *Especificações Placa Arduino Uno*. Disponível em: < http://www.filipeflop.com/pd-6b58d-placa-uno-r3-cabo-usb-para-arduino.html>. Acesso em: 13 out. 2015.

ARDUINO. *Arduino UNO & Genuino* **UNO.** Disponível em: < https://www.arduino.cc/en/Main/ArduinoBoardUno>. Acesso em: 16 out. 2015.

MENDES, D. R. *Programação Java com Ênfase em Orientação a Objetos*. São Paulo: Novatec, 2009.

ARDUINO SHIELD. *Arduino Ethernet Shield*. Disponível em: <https://www.arduino.cc/en/Main/ArduinoEthernetShield >. Acesso em: 02 mar. 2016.

FILIPEFLOP. *Sensor de Fluxo de Água 1/2" YF-S201*. Disponível em: < http://www.filipeflop.com/pd-206c5b-sensor-de-fluxo-de-agua-1-2-yf-s201.html>. Acesso em: 27 maio 2016.

ADAFRUIT. *Liquid Flow Meter*. Disponível em: < https://www.adafruit.com/products/828>. Acesso em: 27 maio 2016.

DEITEL, H. M. *Java, Como Programar*. 4ª ed. Porto Alegre: Bookman, 2003.

FOWLER, M. *UML essencial.* 2ª ed. Porto Alegre. Bookman, 2000.

WIRING. *Wiring. What will you do with the w?*. Disponível em: < http:// http://wiring.org.co/>. Acesso em: 30 jul. 2016.

VICTORINO, C. J. A. *Planeta água morrendo de sede: uma visão analítica na metodologia do uso e abuso dos recursos hídricos.* Porto Alegre: EDIPUCRS, 2007.

APLICABILIDADE DA GAMIFICAÇÃO PARA ENSINO DE QUÍMICA LABORATORIAL

APPLICABILITY OF GAMIFICATION FOR TEACHING LABORATORY CHEMISTRY

RAFAEL BALREIRA DOS SANTOS[1]*, LEANDRO ROSNIAK TIBOLA[1]

[1] Departamento de Engenharias e Ciência da Computação, Universidade Regional Integrada do Alto Uruguai e das Missões. URI – Câmpus de Frederico Westphalen, Brasil.
*E-mail: rafabalreira@gmail.com.

Resumo: O presente artigo abordará o desenvolvimento de um jogo no tema de gamificação, que é o uso de jogos em assuntos do mundo real, para auxiliar no ensino de química, focando nos experimentos feitos em laboratório, com uma breve explicação antes para que o aluno receba diversas informações sobre o experimento. O "porque" por trás deste trabalho é simples, o alto custo de equipamentos e componentes químicos, o risco que diversos experimentos podem acarretar e a falta de estrutura e pessoal especializado em diversas instituições de ensino, este trabalho busca amenizar estes problemas que existem no ensino de química atual. Por fim, busca-se o desenvolvimento e um protótipo totalmente funcional do jogo, com sala aula, vestiário e experimento. Este artigo resulta de um TCC em andamento.

Palavras-chave: Gamificação. Unity3D. Desenvolvimento. Jogo. Laboratório de química. Aprendizagem.

Abstract: This article will address the development of a game on the gamification theme, which is the use of real-world issues in games, to assist in the teaching of chemistry, focusing on experiments done in laboratories with a brief explanation before for the student to receive a variety of information about the experiment. The "why" behind this work is simple, the high cost of equipment and chemicals, the risk that several experiments can lead to and the lack of infrastructure and skilled personnel in various educational institutions, this paper seeks to mitigate and cancel these problems that exist in the current chemistry education. Finally, we seek at the end of this development to have a fully functional prototype of the game, with class room, dressing room and experiment. This article results from a TCC in progress.

Keywords: Gamification. Unity3D. Development. Game. Chemistry lab. Learning.

1 INTRODUÇÃO

Segundo Verga (2005), nos laboratórios químicos são manipulados diversos tipos de compostos químicos que, se manuseados de forma incorreta, podem ferir ou até mesmo matar um ser humano, sendo que estes acidentes podem ocorrer em qualquer instalação laboratorial de empresas, instituições e universidades. Além disso, existe uma variedade de possíveis riscos dentro de um laboratório, tais como substâncias tóxicas, corrosivas, irritantes, inflamáveis e diversas outras, além de equipamentos que podem fornecer riscos mecânicos, térmicos, elétricos, radioativos e alguns outros. A partir destas informações sobre segurança laboratorial, é necessário ter o conhecimento sobre diversas regras básicas para a proteção pessoal e coletiva. Sendo que alguns cuidados básicos devem ser o uso de Equipamento de Proteção Individual (EPI) e Equipamento de Proteção Coletivo (EPC), manusear vidrarias, reagentes e descartes adequadamente, de modo a evitar que acidentes ocorram (AREND et al, 2007).

Adicionalmente, segundo Carvalho (1999), os laboratórios, em geral, são alvos para análises de situações de risco, pois aqueles que estão realizando suas funções no referido laboratório podem não perceber, mesmo que esteja evidente, que existe algo que pode trazer risco à vida e à saúde dos operadores, isto se deve ao nível de atenção que certas atividades necessitam e, logo, pode não se fazer uma verificação das condições seguras de trabalho, e que são necessárias.

Outro aspecto que impede diversas aulas práticas, segundo Altun et al (2009), é a falta de acesso a diversos componentes de laboratório, dentre eles os reagentes, equipamentos e acessórios necessários para o ensino de química. Além de todos os possíveis perigos e custos existentes em um laboratório, segundo Areno (2003), existe a tradicional metodologia de ensino: esta se baseia na teoria, havendo uma exclusão da parte prática de diversos conteúdos. Entretanto, novos métodos de ensino e aprendizagem estão surgindo e uma possível alternativa para minimizar estes problemas é a gamificação dos laboratórios químicos (ARENO, 2003).

Assim, este trabalho tem como objetivos propor uma alternativa para as lacunas citadas anteriormente ao investigar a aplicação da gamificação e desenvolver um jogo para simular as boas práticas de um laboratório de química em um ambiente virtual. Além disto, a gamificação de experimentos químicos pode cortar gastos e riscos, e aumentar o interesse, prática e confiança de

alunos que optarem por utilizar este jogo como ferramenta de estudo.

2 REFERENCIAL TEÓRICO

2.1 Gamificação

De uma forma mais ampla, para Zichermann e Cunningham (2011), a gamificação pode significar o uso de jogos para fazer a propaganda de algum serviço ou produto, ou a criação de algum mundo virtual para realizar o treinamento de funcionários em algum sistema complexo. Para estes autores as descrições estão corretas, já que a gamificação faz a reunião de todas estas descrições de elementos de jogos para dentro do mundo dos não jogos. Ou seja, a gamificação é o uso de jogos e seus elementos, mecânicas e conceitos no mundo real e em seus problemas, serviços e produtos (ZICHERMANN e CUNNINGHAM, 2011).

2.2 Tipos de gamificação

Segundo Werbach e Hunter (2012), a gamificação pode ser aplicada a uma enorme gama de atividades e contextos, em geral existem três principais tipos de gamificação: a Interna, a Externa e a de Mudança de Comportamento. A Interna está relacionada ao estímulo individual de cada funcionário, aumentar a camaradagem e até na busca de resultados positivos feitos pelos mesmos. A Externa representa seus clientes. Normalmente a aplicabilidade desta é voltada para o marketing, aprimoramento da relação vendedor/cliente, incremento do interesse e o impulso dos lucros são algumas das áreas. Porém, o último tipo de gamificação não difere os indivíduos nos tipos interna ou externa, uma vez que almeja toda uma população, por conta disto possui o nome de Mudança de Comportamento.

3 ELEMENTOS DE JOGOS

Segundo Werbach e Hunter (2012), um jogo em si é um quebra cabeças completo, e os elementos que o compõem são as peças que se juntam para formar o total no fim. Por exemplo, em um jogo de xadrez as peças, as regras de movimentação e o ataque também são elementos de jogos que separados não possuem muita lógica, mas juntos formam o jogo de xadrez como conhecemos. Porém, na gamificação não são utilizados todos os elementos de um jogo, pois é necessário ter uma flexibilidade para criar algo gamificado. Voltando ao exemplo do xadrez, ao usar todas as regras deste jogo não existirá gamificação, mas sim uma reprodução do jogo que em nada muda para outras tantas similares, porém, se forem utilizadas apenas algumas regras, uma mescla de ideias ou a criação de todo um novo conceito, tem-se a gamificação do jogo de xadrez. Este é um dos pontos principais para se usar a gamificação: a flexibilidade do uso de quaisquer elementos de jogo.

3.1 Uso sério dos jogos

Uma questão que deve ser levada em consideração ao se pensar na gamificação: por que uma prática baseada em jogos deve ser levada a sério no mundo real? Segundo Werbach e Hunter (2012), existem diversas respostas para estas perguntas, porém existem três pontos principais que se destacam, são eles o Engajamento, a Experimentação e os Resultados.

O Engajamento é à base da gamificação, pois traz motivação às pessoas que o usam. A Experimentação abre uma poderosa ferramenta: a oportunidade. Jogar um jogo é uma experimentação em si, onde se espera vitória e também o fracasso, e no caso deste é sempre possível recomeçar. Os Resultados são a maior prova que a gamificação funciona efetivamente, para isto, pode-se citar que médicos que treinam em simuladores, soldados simulam uma batalha em um computador para tomarem decisões sem causar a perda de vidas humanas, companhias aumentando seus lucros e alunos aprendendo mais. Alguns exemplos da utilização da gamificação no mercado de produtos são os das grandes empresas Nike (Nike+, 2016) e Microsoft (Microsoft, 2016) (WERBACH E HUNTER, 2012).

Outro ponto fundamental na gamificação é a motivação do jogador. Segundo Zichermann e Cunningham (2011), na gamificação o jogador é a raiz de tudo, é a motivação dele que faz com que todo o projeto faça valer a pena e ao entender isto, o sucesso virá para o domínio da aplicação da gamificação. É do conhecimento comum de que jogos são ótimos motivadores por focarem em três pontos: prazer, recompensa e tempo

4 TRABALHOS RELACIONADOS

Inicialmente, Dalgarno et al (2009), disponibilizaram o *software* "*Virtual Chemistry Laboratory*", um laboratório virtual 3D criado para que os alunos sintam uma maior familiaridade com o laboratório, aumentando o aproveitamento do tempo na aula prática, obtendo um ensinamento prévio do uso de EPI, acessórios, equipamentos e outros que podem não estar disponíveis fisicamente para o aluno, além do aluno prestar mais atenção aos conceitos de química.

Por fim, Labster (2016) é uma companhia dedicada a desenvolver laboratórios avançados e interativos, baseados em algoritmos matemáticos, elementos de gamificação, tais como: imersão 3D, sistema de pontos e outros. Os laboratórios desenvolvidos são usados em universidades de renome pelo mundo todo, algumas delas são: Harvard, MIT, Universidade de Hong Kong, Faculdade Técnica Gwinnett entre outras. Seus laboratórios envolvem áreas como a da bioquímica, biotecnologia, ecologia, química, genética e algumas outras. Na Figura 1 encontra-se uma imagem do laboratório virtual usado para o equipamento de *High Performance Liquid Chromatography* (HPLC), em tradução livre, Cromatografia Líquida de Alta Eficiência.

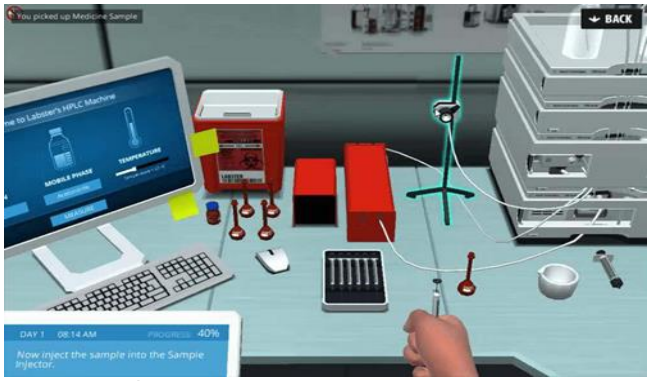


Fig. 1. Laboratório do equipamento HPLC. Fonte: LABSTER (2016).

5 PROTÓTIPO DO JOGO

5.1 Unity3D

Com a *Game Engine* Unity3D é possível criar tantos jogos 2D (Duas Dimensões) quanto 3D (Três Dimensões) com a mesma facilidade e praticidade, além de serem altamente otimizados e visualmente agradáveis. Sendo que a Unity3D é a líder em multiplataforma do mercado, é possível disponibilizar jogos para um total de 23 sistemas operacionais diferentes. Além disso, a *Application Programming Interface* (API) é totalmente expansiva, ou seja, é possível para o usuário criar extensões para a ferramenta ou baixar elas da Asset Store (UNITY3D, 2016).

Devido às vantagens que o Unity3D apresenta, diversos autores utilizam esta ferramenta para desenvolver suas aplicações, sendo utilizada por Silva (2014) para desenvolver um aplicativo para Android que utiliza a RA e é destinado ao ensino dos sistemas do corpo humano. Já para Moreira, Tirabassi e Dogo (2015) a ferramenta foi utilizada para criar um jogo educativo que estimula a aprendizagem ao usar conceitos de administração. E Silveira (2014) utilizou a referida ferramenta para a criação de um jogo para ajudar no entendimento da tabela periódica. Na Figura 2 tem-se a visão de todo o mapa modelado até o momento.


Fig. 2. Mapa principal do jogo em desenvolvimento.

6 DESCRIÇÃO DO JOGO

6.1 Área de Aplicação e Experimento

É de conhecimento geral de que existem três estados da matéria: sólido, líquido e gasoso. Também é de conhecimento geral que sem a água não existiria vida no planeta Terra. Por conta disto existem hoje muitas reações químicas que ocorrem em meio aquoso, portanto deve-se levar em conta alguns conceitos e propriedades de diferentes substâncias em solução com a água. Sendo assim, uma solução é uma mistura homogênea de duas ou mais substâncias, onde o que existe em menor quantidade é o soluto e o que existe em maior é o solvente. Um dos tipos mais comuns de reação ocorre neste meio aquoso, é a reação de precipitação, onde uma reação entre dois componentes gera um precipitado, ou corpo de fundo, que nada mais é que um produto que apresenta baixa solubilidade (CHANG, 2010).

A reação química que este artigo busca realizar é a de precipitação do iodeto de chumbo (PbI2), e que segundo Daniel (2013), ocorre quando se mistura o iodeto de potássio com o nitrato de chumbo, dando origem ao iodeto de chumbo em cor amarelada e ao nitrato de potássio que é incolor.

7 LABORATÓRIO DE QUÍMICA PROPOSTO

O laboratório de Química proposto será constituído de três partes principais, sendo estas: Sala de aula, Vestiários e Laboratório. Na sala de aula serão apresentados os conceitos básicos sobre os EPI's e EPC's; sobre o experimento e sobre boas práticas e normas em um laboratório. Após isto o jogador será direcionado para o vestiário onde poderá equipar-se com o EPI necessário e desejado. Por fim, este deve ir ao laboratório realizar o experimento.

Além das três partes principais, o jogador terá que lavar as mãos em um banheiro adequado ao uso em laboratórios, pegar os materiais no estoque e depois tratar o descarte adequadamente. Adicionalmente, toda regra seguida, norma obedecida e uso correto de todo e qualquer objeto contará pontos para que no final seja demonstrado o resultado, juntamente com um relatório sobre as ações do jogador. Na Figura 3 tem-se o esquema completo de como será o mapa do jogo.

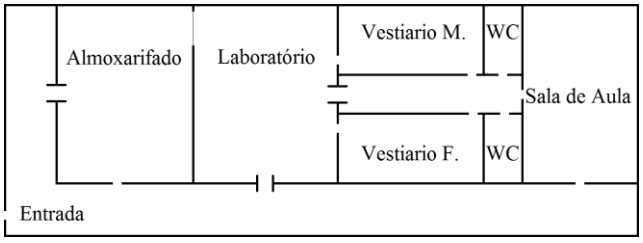


Fig. 3. Esquema do mapa completo.

Além disto, durante a aula, o jogador receberá informações sobre os itens descritos anteriormente e após terminar de ler todos os tópicos, este deverá realizar uma avaliação contendo os assuntos abordados. Esta avaliação terá nota mínima de sete pontos. Caso o jogador alcance nota superior a sete pontos ele poderá assistir qualquer tópico da aula novamente ou seguir adiante no jogo. Caso ele tire uma nota menor deverá repetir os tópicos com os quais ficou com desempenho insuficiente.

Os objetos, ferramentas e detalhes serão importados de sites especializados em modelagem 3D gratuitos, caso não seja encontrado o modelo necessário o mesmo será

modelado na própria Unity3D. Adicionalmente, os scripts necessários serão criados a partir de uma ferramenta interna da própria Unity3D e utilizando os conhecimentos já dominados pelos autores, além disso, serão realizadas consultas à comunidade Unity3D e buscas na Internet.

Após um desenvolvimento preliminar serão feitos os primeiros testes, dentro da própria ferramenta, para que sejam detectados os erros de modelagem e programação. Na modelagem do laboratório serão levados em conta a temperatura, umidade, luminosidade e ventilação do ambiente, fatores que são fundamentais para prática em laboratório, além de variáveis dos próprios componentes e utensílios usados na experiência. Os produtos químicos terão como variáveis: armazenamento e manuseio adequados, além dos itens citados.

Ao finalizar esta fase preliminar do desenvolvimento, será requisitado para que profissionais da área voluntariamente testem o jogo e relatem sua experiência na forma de uma avaliação técnica, a fim de permitir a melhoria do jogo.

Por fim, este jogo se inspira no Labster (2016), porém apresenta menos detalhamento gráfico que ele por ser um protótipo acadêmico. Contudo, o jogo resultante deste trabalho, poderá ser um diferencial no domínio acadêmico, principalmente na área da Química, pois existem diversos jogos semelhantes, mas a grande maioria focada em *smartphones*, *web* ou até mesmo em assuntos mais focados para o ensino fundamental e médio, sendo que não foi encontrado nenhum artigo ou trabalho focando o ensino superior, com a abrangência demonstrada neste trabalho. Levando em conta que o jogador deverá assistir a uma aula, equipar-se com o EPI necessário e realizar o experimento usando todas as normas e procedimentos, isto leva a uma imersão no conteúdo que não se encontra facilmente no mercado de jogos como ferramentas de ensino.

8 Resultados Esperados

Com o desenvolvimento deste protótipo, o primeiro resultado esperado a ser alcançado é a gamificação de um laboratório de Química, ou seja, a realização de uma experiência química dentro deste laboratório e o ensino diferenciado de conceitos necessários para que um aluno entre em laboratório com confiança e com vontade de aprender o que praticou no jogo, levando em conta as boas práticas e as normas de segurança.

9 Perspectivas Futuras

Espera-se que no decorrer do desenvolvimento seja possível programar um sistema de *easter eggs*, *badges*, experimentos perigosos, levando a mais *levels*, além de uma melhora no gráfico do jogo. Também se busca adicionar a figura do professor e a opção de realizar experimentos livres após finalizar todos os experimentos do *game*.

Além disto, pretende-se deixar este trabalho como um projeto de pesquisa futuro, envolvendo alunos da área da Ciência da Computação e da Química, para que sejam apresentados novos experimentos, técnicas de programação e até a possível publicação global do jogo para que este possa ser usado em outros lugares e beneficiando mais alunos que buscam conhecimento de formas diferenciadas.

10 Conclusões

Percebe-se que, diante dos fatos mencionados por Areno (2003), Verga (2005), Arend et al (2007) e Altun (2009), o ensino de química apresenta grandes dificuldades e limitações diante de tantas restrições, porém a gamificação apresentada neste trabalho é se mostra como um possível alternativa para amenizar este quadro.

Além disto, as técnicas de gamificação apresentadas por autores como Zichermann e Cunningham (2011) e Werbach e Hunter (2012) podem apresentar novas luzes para esta ciência, e se tecnologias como o Labster (2016) puderem ser mais disseminadas o ensino de química no geral pode ter um impulso e frear a constate fama de difícil do ensino de química.

REFERÊNCIAS

ALTUN, E. et al. *Developing an interactive virtual chemistry laboratory enriched with constructivist learning activities for secondary schools*. In: World Conference On Educational Sciences: New Trends And Issues In Educational Sciences, 2009, Nicosia: North Cyprus. 1895-1898.

AREND, K. et al. *Manual de Segurança em Laboratórios Didáticos*. Universidade Regional Integrada do Alto Uruguai e das Missões. 2007.

ARENO, H. B. *Simulação Como Ferramenta De Ensino Em Cursos De Engenharia De Produção E Administração.* São Paulo, 2003. 111 f. Engenharia de Produção, Universidade de São Paulo.

CARVALHO, P. R. 1999. *Boas Práticas Químicas em Biossegurança*. Rio de Janeiro: Interciência, 1999.

CHANG, Raymond. 2010. Química Geral: conceitos essenciais. Porto Alegre: AMGH, 2010

DALGARNO, Barney et al. *Effectiveness of a Virtual Laboratory as a preparatory resource for Distance Education chemistry students.* Elsevier ltda, Science Direct, 853-865, 2009. Digital.

DANIEL, Vera. 2013. *Reação química entre o nitrato de chumbo e o iodeto de potássio*. Disponível em: <gqj.spq.pt/chemrus/2013/iodeto.pdf>. Acesso em: 29 jul. 2016.

GONÇALVES, Patrícia C. T. *Desenvolvimento de uma Interface Gráfica para o Programa FastComp*. Portugal: Porto, 2005. 137 f. Dissertação do grau em Mestre – Métodos Computacionais em Ciências e Engenharia, Universidade do Porto.

LABSTER. *Empowering the Next Generation of Scientists to Change the World*. Disponível em: <https://www.labster.com>. Acesso em: 23 jul. 2016.

MICROSOFT. 2016. Microsoft. Disponível em: <https://www.microsoft.com/pt-br/>. Acesso em: 23 jul. 2016.

MOREIRA, André R.; TIRABASSI, Paulo H.; DOGO, Vinicius R. Desenvolvimento de jogo educativo digital para estimular o processo de aprendizagem. In:

Simpósio Interdisciplinar De Tecnologias Na Educação, 1 ago. 2015, Boituva: SP. 244-248.

NIKE+. 2016. Seu personal trainer. *A qualquer momento. Em qualquer lugar*. Disponível em: <http://www.nikeplus.com.br/>. Acesso em: 23 jul. 2016.

SHELDON, Lee. *The Multiplayer Classroom*: Designing Coursework as a Game. Boston: Course Technology, 2012.

SILVA, Rodolpho S. ANATOMIA-RA: *Aplicativo Para Android Destinado Ao Ensino Dos Sistemas Do Corpo Humano Com A Utilização Da Realidade Aumentada*. Campina Grande, 2014. 101 f. Licenciatura em Computação, Universidade Estadual da Paraíba.

SILVEIRA, Aleph C. *Aventuras no mundo da tabela periódica:* Criação de uma aplicação pedagógica para o ensino de Química. Lavras, 2014. 50 f. Departamento de Ciência da Computação, Universidade Federal de Lavras.

UNITY3D. 2016. Create Games, *Connect With Your Audience, and Achieve Success*. Disponível em: <http://unity3d.com/pt/unity>. Acesso em: 25 jul. 2016.

VERGA, A. F. *Artigo alerta sobre causas de acidentes em laboratórios*. 2005. Disponível em: <http://www.crq4.org.br/informativomat_435>. Acesso em: 23 jul. 2016.

WERBACH, Kevin; HUNTER, Dan. *How Game Thinking Can Revolutionize Your Business*. Philadelphia: Wharton Digital Press, 2012

ZICHERMANN, G; CUNNINGHAM, C. *Gamification by Design*. Sebastopol: O'Reilly Media, 2011.

.

A presente edição foi composta pela URI,
em caracteres Times New Roman,
formato e-book, pdf, em junho de 2017