

Uma abordagem prática da tecnologia SDN / OpenFlow em redes voltadas para tráfego de voz e dados

Carlos Alfredo Weissheimer Junior¹, Diego Menine²

¹Docente do curso de Redes de Computadores – Faculdade IENH
Rua Frederico Mentz, 526 – Hamburgo Velho – Novo Hamburgo RS

²Dicente do curso de Redes de Computadores – Faculdade IENH
Rua Frederico Mentz, 526 – Hamburgo Velho – Novo Hamburgo RS
{carlosawe, diego.sapi}@gmail.com

Abstract. *This article discusses the implementation of a software-defined network (SDN) to manage voice and data traffic using a switch that supports the OpenFlow protocol. The purpose of this document is to provide a fast alternative to applying QoS -shaped devices that can encourage the reader to think of other alternatives to application service quality using the SDN technology. For collecting and analyzing the results, was used the Model E proposed by the ITU- T G703 where searched tried to get an average for each indicator at the end of the test sections. Concluded that in a chaotic environment without application of QoS, use of voice service is not feasible in need of special treatment in this type of service, in form simply can be obtained by using devices that support SDN technology.*

Resumo. *Este artigo aborda a implementação de uma rede definida por software (SDN) para gerenciar o tráfego de voz e dados utilizando um switch com suporte ao protocolo OpenFlow. O objetivo deste documento é apresentar uma alternativa ágil para aplicação de QoS em dispositivos de forma que, possa incentivar o leitor a pensar em outras alternativas para aplicação de qualidade de serviço utilizando a tecnologia SDN. Para coleta e análise dos resultados, foi utilizado o Modelo E proposto pela ITU-T G703 onde procurou-se obter uma média de cada indicador ao final das seções de teste. Concluiu-se que em um ambiente caótico sem aplicação de QoS, a utilização do serviço de voz se torna inviável necessitando de um tratamento diferenciado neste tipo de serviço, a qual, pode ser obtido de forma simples utilizando dispositivos com suporte à tecnologia SDN.*

1. Introdução

Uma nova abordagem de redes de computadores tem ganhado força nos últimos anos como alternativa para a forma de gerenciar, e também de baratear a implementação e expansão de redes de computadores, trata-se da tecnologia SDN (*Software Defined Network* ou Redes Definidas por Software), com o objetivo de simplificar a rede, segundo [Lara 2013]. A tecnologia de redes definidas por *software* é definida como “[...] uma forma de se implementar políticas de controle de acesso de forma distribuída, a partir de

um mecanismo de supervisão centralizado” [Guedes 2012 apud Casado et al. 2009]. Neste artigo, espera-se contribuir com a comunidade acadêmica, bem como auxiliar administradores, gerentes de projeto e profissionais da área de infraestrutura em T.I. através da quebra de paradigma em relação à utilização de novas tecnologias na área de redes de computadores. E assim, apresentar outra opção de implementação com *software* aberto sem vínculo com fabricantes de *hardware* justifica a pesquisa sobre o assunto.

Como objetivo o presente trabalho implementou e analisou o desempenho de tráfego VoIP utilizando a arquitetura de redes definidas por *software* descritas neste artigo. Neste contexto, este artigo aborda a implementação da tecnologia de redes definidas por *software* e a utilização do protocolo OpenFlow para configuração de um dispositivo físico, buscando comparar o desempenho de uma aplicação de voz sobre IP em um ambiente com grande volume de tráfego sofrendo saturação para análise dos dados obtidos de acordo com os dois cenários propostos neste artigo.

2. Componentes utilizados

Foi utilizado um *switch* modelo TL-TW1043ND versão 1.8 onde o *software* do fabricante foi substituído por um *firmware* da OpenWrt backfire versão 10.03.1 [OpenWRT 2013], com suporte ao protocolo OpenFlow versão 1.0, que é uma distribuição Linux com kernel 2.6.32.27 personalizada para ser embarcada nestes dispositivos, [Moraes et al. 2014]. As 4 portas LAN do *switch* foram configuradas para aplicar as regras propostas pelo controlador RYU. Um ambiente virtualizado foi criado através da ferramenta Virtual Box da Oracle versão 4.3.30 disponibilizada em uma máquina customizada utilizando um processador Intel Core i3 com 32GB de memória RAM e um HD de 1Tb rodando o sistema operacional Windows 7 para suportar três servidores virtuais: O controlador RYU versão 3.23, o Sip Server Elastix versão 2.4.0 e o CentOS versão 7.

Quanto aos *softwares*, foi utilizado para tráfego de voz, a distribuição StartTrynity Sip Tester versão 2015-03-04 15:01-UTC utilizando o protocolo SIP no modelo cliente/servidor em conjunto com o *Software* Elastix. Também foi utilizado o *software* Iperf versão 2.0.5 para gerar tráfego TCP também no modelo cliente/servidor com o servidor CentOS 7, compondo o ambiente virtualizado melhor visualizado na figura 1.

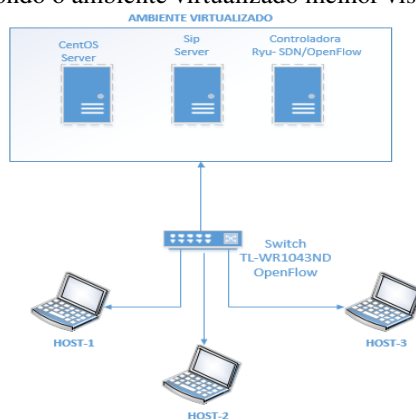


Figura 1 – Arquitetura de rede proposta

Para completar o cenário e esgotar as 4 portas LAN disponibilizadas no *switch*, foram utilizados três máquinas físicas, um *ultrabook* Samsung serie 5 com processador Intel Core i5 e 4 GB de RAM e sistema operacional Windows 8.1 nomeado de *host-1* para gerar tráfego TCP via Iperf com o servidor CentOS, outro *ultrabook* Samsung serie 5 com processador Intel Core i7 com 4 GB de RAM e sistema operacional Windows 8.1 nomeado de *host-2*, utilizado para gerar as chamadas VoIP utilizando o *software* Star Trinity, e um *notebook* Asus com processador Celeron com 2GB de RAM e sistema operacional Windows 7 nomeado de *host-3* para também gerar tráfego TCP executando o *software* Iperf com o servidor CentOS.

3. Referencial bibliográfico

Para que o conceito de SDN (*Software Defined Networks*) ou Redes Definidas por *Software* possa ser compreendido, será introduzida a base teórica quanto aplicação desta tecnologia e os componentes necessários para sua utilização.

Além disso, é apresentado o embasamento quanto ao conceito de aplicação do protocolo OpenFlow, utilizado neste projeto de forma que se torna essencial para o funcionamento da Rede Definida por *Software*, que é o objetivo deste artigo.

3.1. Conceito de SDN (*Software Defined Network*)

O termo SDN surgiu a partir da necessidade de se obter uma melhor forma de gerenciar as redes de computadores conforme [Lópes 2014].

A SDN permite que o administrador de rede gerencie serviços de redes através da abstração das funcionalidades de nível inferior. Isto é conseguido através da dissociação do plano de controle (onde são construídas as decisões de roteamento) e o plano de dados (o nível mais baixo composto pelos dispositivos físicos responsáveis de rotear o tráfego).

Pode ser citado, além da melhora no gerenciamento da rede, o amplo controle de fluxo de dados conforme [Santos et al. 2014]. “*Software Defined Networking (SDN) é um novo paradigma que promete fornecer a capacidade de amplo controle sobre fluxos de tráfego da rede, com o objetivo principal de simplificá-la e de torná-la mais barata e flexível [...]*”.

Desta forma, o objetivo de uma rede SDN é centralizar e separar o controle do encaminhamento de dados, de forma que proporcione uma visão geral da rede e que facilite o seu gerenciamento como um todo. Conforme [Guedes 2012], os elementos comutadores exportam uma interface de comunicação que permite ao controlador analisar, alterar e definir as entradas da tabela de roteamento do comutador. Corroborando a tudo isso [Lara 2013] apresenta uma motivação que estabelece um paralelo importante com o que este projeto aborda.

Uma das motivações da SDN é executar tarefas de rede que não poderiam ser realizadas sem um *software* adicional para cada um dos elementos de comutação. Aplicativos desenvolvidos podem controlar os *switches* rodando em cima de um sistema operacional de rede, que funciona em uma camada intermediária entre o *switch* e a aplicação.

A Figura 2, ilustra a disposição dos elementos de rede em uma arquitetura SDN.

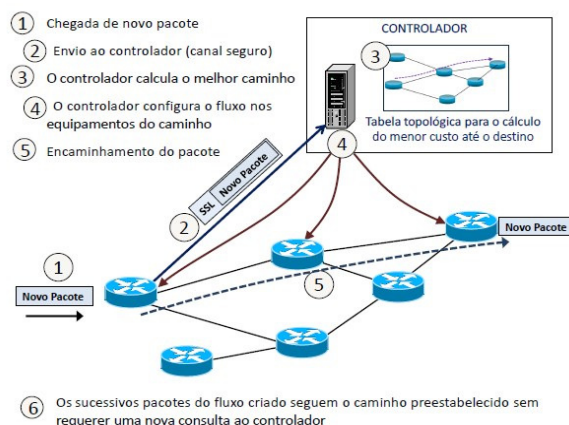


Figura 2 - Elementos de uma rede com arquitetura SDN Fonte: [Lópes 2014]

Conforme [Lópes 2014], quando o primeiro pacote de um fluxo chega ao dispositivo de rede e este não tem uma orientação quanto ao seu tratamento na tabela de fluxos que indique por onde encaminhar este pacote, o dispositivo de rede envia uma consulta ao controlador solicitando como proceder. Desta forma o controlador irá calcular a melhor rota para aquele fluxo específico e através de um protocolo de comunicação irá configurar o fluxo específico no plano de dados dos dispositivos participantes tornando possível o roteamento do pacote.

Na ótica de [Avelar 2013], o OpenFlow é um protocolo de padrão aberto, que implementa o conceito SDN e permite a criação de redes virtuais usando somente recursos L2 (switches Ethernet) com tabelas de fluxo internas e uma interface padrão para adicionar e remover entradas de fluxos. [Astuto et al. 2014], também ressalta a utilização para a indústria que desenvolve dispositivos com suporte a redes definidas por *software*.

4. Metodologia de Pesquisa

A metodologia *Absolute Category Rating* (ACR) padronizada pela ITU-T P.800, foi aplicada para avaliar a qualidade de voz no *Called* a qual recebe uma nota de 1 a 5 de acordo com o índice de *MOS* obtido ao final da seção de teste.

4.1. Técnica de coleta de dados

Para a coleta de dados, foi utilizado o Modelo E normatizado pela ITU-T na recomendação G.107, o qual indica a escala de pontuação definida como *Mean Opinion Score* (MOS) para avaliação da qualidade de voz em uma rede IP. O índice de MOS considera indicadores de perda durante a transmissão de pacotes na rede [Lustosa 2004].

4.2. Simulação e resultados

Conforme orientação da ITU-T P.800, foi definido a quantidade de 10 repetições de teste afim de coletar uma média aproximada entre cada teste, conforme [Gil 2002], “O

procedimento básico adotado na análise estatística nas pesquisas experimentais consiste no teste da diferença entre as médias”.

Neste contexto foram realizadas duas etapas de teste, a primeira utilizando uma configuração padrão para conexão e troca de pacotes pela rede utilizando uma programação disponível no controlador Ryu e a segunda repetindo os mesmos testes, mas utilizando uma programação agregada de QoS para tráfego de voz afim de analisar os índices de MOS, Jitter e Packet Loss (pacotes perdidos) de acordo com os dados obtidos pelo tráfego do *Called*.

4.3. Metodologia aplicada

Para viabilizar os dois cenários, foram editados os arquivos *network.conf* e o *openflow.conf* listando as interfaces Eth1 a Eth4 gerenciadas pelo controlador Ryu setado com o IP 192.168.200.254/24. A interface Eth4 recebeu o IP estático 192.168.200.254/24. A configuração *simple_switch.conf* foi iniciada e o *switch* foi identificado e as portas OpenFlow reconhecidas. A captura de pacotes foi realizada utilizando o *software* Wireshark.

Após a certificação de comunicação entre o *switch* e o controlador, os demais servidores virtuais e *hosts* 1, 2 e 3 foram configurados com IPs estáticos da rede 192.168.100.0/24. O servidor CentOS com IP 192.168.100.254 disponibiliza a conexão TCP na porta 445 via Iperf para aguardar a conexão dos clientes. Além disso, o *host* 1 foi configurado com IP 192.168.100.2/24 e o *host* 3 com IP 192.168.100.4/24.

Enquanto os *hosts* 1 e 3 mantinham o tráfego de dados afim de gerar gargalo no *switch*, o *host* 2 no IP 192.168.100.3/24 iniciava a transmissão de tráfego de voz com o servidor Sip Elastix no IP 192.168.100.100 utilizando o *software* Star Trinity Sip Tester. Para viabilizar o tráfego de voz, foram criados dez usuários no *software* Elastix simulando dez ramais e também uma sala de conferências, a qual atendia as chamadas originadas pelo Star Trinity Sip Tester. Cada chamada executada pelos ramais partindo do *host* 2 continha uma gravação de 30 segundos utilizando o Codec G.711 que era executada igualmente pelos dez ramais registrados no Sip Trinity de forma que os pacotes RTP (voz) pudessem partir do *Caller* para o *Called*.

SIP call quality indicators														
Quality indicator name	Ncalls	Min	Average	Max	Percentile: 90%	95%	98%	99%	99.5%	99.8%	99.9%	99.95%	99.98%	99.99%
Caller lost packets (%)	10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Caller G.107 MOS	10	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41
Caller G.107 R-factor	10	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20
Caller max delta (ms)	10	32.34	34.47	46.98	37.08	46.98	46.98	46.98	46.98	46.98	46.98	46.98	46.98	46.98
Caller max RFC3550 jitter (ms)	10	6.94	8.48	9.13	9.11	9.13	9.13	9.13	9.13	9.13	9.13	9.13	9.13	9.13
Caller mean RFC3550 jitter (ms)	10	6.24	6.26	6.27	6.27	6.27	6.27	6.27	6.27	6.27	6.27	6.27	6.27	6.27
Called lost packets (%)	10	7.92	11.59	16.06	15.53	16.06	16.06	16.06	16.06	16.06	16.06	16.06	16.06	16.06
Called G.107 MOS	10	2.63	3.12	3.56	2.67	2.63	2.63	2.63	2.63	2.63	2.63	2.63	2.63	2.63
Called G.107 R-factor	10	50.97	60.52	69.30	51.79	50.97	50.97	50.97	50.97	50.97	50.97	50.97	50.97	50.97
Called max delta (ms)	10	63.96	83.89	140.61	97.56	140.61	140.61	140.61	140.61	140.61	140.61	140.61	140.61	140.61
Called max RFC3550 jitter (ms)	10	7.47	8.26	8.86	8.85	8.86	8.86	8.86	8.86	8.86	8.86	8.86	8.86	8.86
Called mean RFC3550 jitter (ms)	10	4.28	4.38	4.45	4.44	4.45	4.45	4.45	4.45	4.45	4.45	4.45	4.45	4.45
100 response delay (ms)	10	90.00	280.40	442.00	416.00	442.00	442.00	442.00	442.00	442.00	442.00	442.00	442.00	442.00
Answer delay (ms)	10	260.00	408.59	635.00	601.93	635.00	635.00	635.00	635.00	635.00	635.00	635.00	635.00	635.00
-24dB delay (ms)	10	1722.65	1830.61	1926.88	1924.79	1926.88	1926.88	1926.88	1926.88	1926.88	1926.88	1926.88	1926.88	1926.88
RTCP RTT (ms)	10	24.38	34.09	42.18	37.86	42.18	42.18	42.18	42.18	42.18	42.18	42.18	42.18	42.18

Figura 3 - Indicadores de qualidade de voz sem QoS.

A Figura 3 demonstra os indicadores de desempenho de voz degradados devido ao grande fluxo de dados na rede. O indicador de *Called Lost Packets* está elevado, o qual afeta diretamente o índice de MOS que fica abaixo do nível mínimo aceitável para o bom

entendimento da voz. Observa-se na Figura 4, que o servidor Sip Elastix, por padrão, já utiliza o DSCP 46 como DiffServ para priorizar o tráfego RTP, mas sem sucesso em um ambiente com alto fluxo de dados e sem QoS.

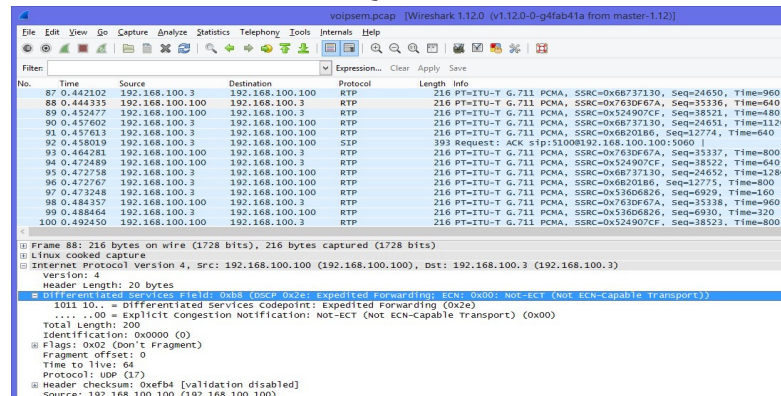


Figura 4 – DSCP 46 (0x2e)

Ao final de dez execuções foram obtidos os seguintes resultados quanto aos indicadores de MOS, atraso entre as entregas de pacotes (Jitter) e pacotes perdidos (pck loss) conforme a Figura 5.

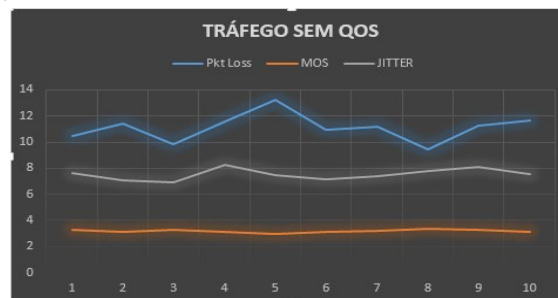


Figura 5 – Gráfico dos indicadores coletados sem QoS

No segundo cenário de testes, mantiveram-se todas as configurações anteriores, com a exceção de uma mudança no controlador, que juntamente com o módulo `simple_switch.py`, também carregou o módulo `rest_qos.py` com a possibilidade de se executar priorização de tráfego de acordo com a programação enviada para o *switch*. Após a mensagem informando a identificação do *switch* e a confirmação de que o módulo de QoS estava ativo e disponível para receber as programações, foram executados dois comandos no controlador para priorizar o tráfego do DSCP 46 já enviado pelo Elastix e para priorizar o range de portas RTP padrão do servidor SIP que vai da porta 10000 a 20000 UDP.

Repetiu-se o mesmo modelo de teste executado no cenário anterior com as mesmas configurações de IPs e padrões de geração de tráfego com Iperf e com o Sip Trinity. A Figura 6 retorna o ganho de qualidade obtido no tráfego de voz com a priorização aplicada.

SIP call quality indicators														
Quality indicator name	Ncalls	Min	Average	Max	Percentile: 90%	95%	98%	99%	99.5%	99.8%	99.9%	99.95%	99.98%	99.99%
Caller lost packets (%)	10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Caller G.107 MOS	10	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41	4.41
Caller G.107 R-factor	10	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20	93.20
Caller max delta (ms)	10	32.83	34.27	37.34	37.31	37.34	37.34	37.34	37.34	37.34	37.34	37.34	37.34	37.34
Caller max RFC3550 jitter (ms)	10	7.05	8.99	10.78	10.53	10.78	10.78	10.78	10.78	10.78	10.78	10.78	10.78	10.78
Caller mean RFC3550 jitter (ms)	10	6.23	6.28	6.30	6.30	6.30	6.30	6.30	6.30	6.30	6.30	6.30	6.30	6.30
Called lost packets (%)	10	0.52	1.09	2.00	1.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
Called G.107 MOS	10	4.23	4.32	4.37	4.28	4.23	4.23	4.23	4.23	4.23	4.23	4.23	4.23	4.23
Called G.107 R-factor	10	86.01	89.24	91.28	87.80	86.01	86.01	86.01	86.01	86.01	86.01	86.01	86.01	86.01
Called max delta (ms)	10	42.45	58.71	153.39	60.12	153.39	153.39	153.39	153.39	153.39	153.39	153.39	153.39	153.39
Called max RFC3550 jitter (ms)	10	4.54	4.89	5.63	5.26	5.63	5.63	5.63	5.63	5.63	5.63	5.63	5.63	5.63
Called mean RFC3550 jitter (ms)	10	2.95	3.02	3.06	3.06	3.06	3.06	3.06	3.06	3.06	3.06	3.06	3.06	3.06
100 response delay (ms)	10	29.00	198.50	409.00	295.00	409.00	409.00	409.00	409.00	409.00	409.00	409.00	409.00	409.00
Answer delay (ms)	10	154.00	269.84	422.00	406.45	422.00	422.00	422.00	422.00	422.00	422.00	422.00	422.00	422.00
-24dB delay (ms)	10	1494.56	1595.44	1700.73	1670.22	1700.73	1700.73	1700.73	1700.73	1700.73	1700.73	1700.73	1700.73	1700.73
RTCP RTT (ms)	10	17.24	19.30	23.01	20.49	23.01	23.01	23.01	23.01	23.01	23.01	23.01	23.01	23.01

Figura 6 – Indicadores de qualidade de voz com QoS

Os resultados quanto aos indicadores de qualidade de voz (MOS), atraso entre as entregas de pacotes (Jitter) e pacotes perdidos (pck loss) apresentaram um excelente desempenho, podendo-se destacar o indicador MOS ficando em 4 pontos conforme a Figura 7.

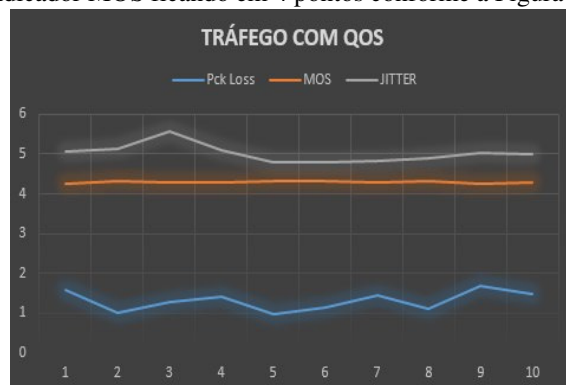


Figura 7 – Gráfico dos indicadores coletados com QoS

Conclusão

Em um ambiente com grande fluxo de dados sem priorização do tráfego de voz, os índices coletados são inaceitáveis inviabilizando a utilização desta modalidade de serviço, neste contexto, foi possível aferir a exigência de regras de controle específicas visando otimizar o desempenho de aplicações com maior sensibilidade a gargalos na rede, as quais podem ser manipuladas de forma centralizada com a implementação de um controlador e do protocolo OpenFlow.

A conclusão apresentada no parágrafo anterior é justificada ao se realizar a análise dos resultados obtidos após a aplicação de QoS na rede de forma rápida através do controlador e do protocolo OpenFlow sem necessitar de acesso ao *switch*, onde o indicador de Packet Loss ficou abaixo dos 2% e o indicador de MOS acima de 4 pontos em uma escala que vai de 1 a 5 onde 5 é excelente.

Referências

Astuto, Bruno Nunes. Mendon, Marc. Ca, Xuan Nam Nguyen, Katia Obraczka, Thierry Turletti. A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks. Communications Surveys and Tutorials, IEEE

- Communications Society, Institute of Electrical and Electronics Engineers (IEEE), 2014, p.1617 – 1634
- Avelar, Edson Adriano Maravalho. Pmipflow: uma proposta para gerenciamento de mobilidade em redes definidas por software. Recife 2013. 167 p. Dissertação (Mestrado em Ciências da computação) – Universidade federal de Pernambuco, Recife, 2013. [Orientador: Prof. Kelvin Lopes Dias]
- Casado, M., Freedman, M. J., Pettit, J., Luo, L., Gude, n., mckeown, n., and shenker, s. Rethinking enterprise network control. IEEE/ACM Transactions on Networking. 2009 p.1270–1283
- Gil, A. C. Como elaborar projetos de pesquisa. 5. ed. São Paulo: Atlas, 2002.
- Guedes, d.; vieira, l. F. M.; vieira, m. M.; rodrigues, h.; nunes, r. V. Redes Definidas por Software: uma abordagem sistêmica para o desenvolvimento de pesquisas em Redes de Computadores. Minicursos - Livro Texto do XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. 1ed.Porto Alegre, p. 161-212, 2012.
- ITU-T 2003a. ITU-T Recommendation G.107. The E-Model, a computational model for use in transmission planning. Genève, mar. 2003.
- ITU-T P.800. ITU-T Series P: Telephone Transmission Quality Methods for objective and subjective assessment of quality
- Lara, a.; Kolasani, a.; Ramamurthy, B. Network Innovation Using OpenFlow: A Survey. IEEE Commun. Surv. Tutor. 2013, 16, 1–20.
- López Rodríguez, Fernando. Arquitetura e Protótipo de uma Rede SDN-OpenFlow para Provedor de Serviço. Brasília 2014. 59 p. Dissertação (Mestrado em engenharia elétrica) – Universidade de Brasília, Distrito Federal, 2014. [Orientador: Prof. Divanilson Rodrigo de Souza Campelo].
- Lustosa, L.C.G., Carvalho, Rodrigues, P.H.A., Mota, S. E., Utilização do Modelo E para avaliação da qualidade da fala em sistemas de comunicação baseados em voz sobre IP, em: Anais do XXII SBRC. Gramado, maio 2004.
- Santos, Marcel. Endo, Patricia. Bezerra, Moisés. Gonçalves, Glauco. Sadok, Djamel. Fernandes, Stênio. Revisitando uma Infraestrutura Autônoma: Uma Perspectiva Baseada em uma Rede Definida por Software. Anais do 4º Workshop de Sistemas Distribuídos Autônomos – WoSiDA. Universidade Federal de Pernambuco (UFPE). 2014