

Uma Arquitetura para o Aprovisionamento de QoS Interdomínios em Redes Virtuais baseadas no OpenFlow

Diego dos Passos Silva, Allan Borges Pontes, Edson Adriano Maravalho Avelar,
Kelvin Lopes Dias.

Centro de Informática - Universidade Federal de Pernambuco (UFPE)

CEP: 50740-540 - Recife - PE - Brasil

{dps4, abp, eama, kld}@cin.ufpe.br

Abstract. *With the advent of software-defined networks (SDN) and, in particular, the OpenFlow platform, new solutions for QoS provisioning are needed to maintain the applications requirements, as they cross different administrative domains which will compose the new Future Internet ecosystem based on virtual networks. This article presents an architecture based on virtualization of networks with end-to-end QoS support considering two levels of mapping. The first one maps QoS specifications (QSPEC) between OpenFlow and IEEE 802.1p priority scheme. The second level provides mapping and interdomain interoperability through NSIS (Next Steps in Signaling) protocol. The performance results obtained from an OpenFlow testbed demonstrate the effectiveness of the proposal.*

Resumo. *Com o advento das redes definidas por software (SDN Software Defined Networks) e, em particular, da plataforma OpenFlow, novas soluções para o provisionamento de QoS são necessárias para manter os requisitos das aplicações, enquanto estas atravessam diversos domínios administrativos que constituirão o novo ecossistema da Internet do Futuro. Este artigo apresenta uma arquitetura, baseada em virtualização de redes, que fornece suporte à QoS fim-a-fim considerando dois níveis de mapeamento. O primeiro nível mapeia especificações de QoS (QSPEC) entre fluxos OpenFlow e o esquema de prioridades do IEEE 802.1p. O segundo fornece mapeamento e interoperabilidade interdomínios através do protocolo NSIS (Next Steps in Signaling). Os resultados de desempenho obtidos a partir de um testbed OpenFlow demonstram a eficácia da proposta.*

1. Introdução

Prover QoS (*Quality of Service*) fim-a-fim ainda é um dos maiores problemas para o sucesso de determinados serviços nos sistemas heterogêneos de telecomunicações usados ao redor do mundo. Abordagens para o provisionamento de QoS na Internet foram extensivamente discutidas no contexto das arquiteturas de Serviços Integrados (IntServ) e Serviços Diferenciados (DiffServ) [Gozdecki et al. 2003], com diversos mecanismos propostos e avaliados. Entretanto, algumas dessas soluções arquiteturais

* Este trabalho foi realizado com recursos do projeto ReVir: Redes Virtuais para a Internet do Futuro, financiado pelo CTIC/RNP

são complexas e/ou de difícil implementação ou aceitação pelos provedores de acesso e de *backbone*, já que ou apresentam custo muito elevado ou não possuem garantias de QoS fim-a-fim.

Com o advento da virtualização de redes e da plataforma OpenFlow [OpenFlow 2012], novas soluções para o provisionamento de QoS são necessárias para manter os requisitos das aplicações, enquanto estas atravessam diversos domínios administrativos que constituirão o novo ecossistema, baseada em redes virtuais, da Internet do Futuro [Schonwalder et al. 2009]. Dessa forma, surge a seguinte questão: “Como suportar aplicações com requisitos distintos e que atravessam redes que combinam equipamentos que operam na camada 3 (L3 – *Layer 3*) e, exclusivamente, na camada 2 (L2 – *Layer 2*), garantindo QoS fim-a-fim no contexto de redes virtuais?”.

Este artigo propõe e avalia uma arquitetura, denominada RME (*Resource Mediation Engine*), que provê suporte à QoS fim-a-fim em redes virtuais baseadas no OpenFlow. A proposta considera dois níveis de mapeamento. O primeiro nível mapeia especificações de QoS (QSPEC) entre fluxos OpenFlow e o esquema de prioridades do PCP (*Priority Code Point*) [IEEE 2011]. O segundo nível fornece mapeamento e interoperabilidade interdomínios através do protocolo NSIS (*Next Steps in Signaling*) [Fu et al. 2005].

O artigo está organizado da seguinte forma: Na Seção 2, os trabalhos relacionados são discutidos. A Seção 3 apresenta a arquitetura proposta. O *testbed* utilizado para validarmos a solução e os resultados obtidos é abordado na Seção 4. A Seção 5 apresenta as conclusões e trabalhos futuros.

2. Trabalhos Relacionados

Esta seção discute os trabalhos relacionados ao OpenFlow e à QoS em redes virtuais. O OpenFlow é uma iniciativa recente e, por isso, há poucas propostas para provimento de QoS nesta tecnologia. Até a versão 1.0 da especificação do OpenFlow, o provimento de QoS era restrito à reserva de banda para fluxos ou *slices* (fatias de redes virtuais). Por isso, este mecanismo é o mais utilizado ou estendido pelas propostas citadas nessa seção.

No trabalho [Civanlar et al. 2010] os autores propõem um controlador de rede virtual que configura as rotas dos fluxos com base em requisitos de QoS para tráfego de vídeo escalável (SVC - *Scalable Video Coding*) em termos de atraso fim-a-fim e de melhor esforço baseado na perda de pacotes. A lógica do controlador é implementada no NOX [Gude et al. 2008]. Os autores em [Egilmez et al. 2011] também propõem uma arquitetura de roteamento baseado em QoS para SVC através de redes OpenFlow. O artigo divide o tráfego em *SVC base layer* (QoS sem perda de pacotes) e *SVC enhancement layer* (QoS com certa tolerância a perda de pacotes e tráfego de melhor esforço). O algoritmo de roteamento utilizado é o “*Network Simplex*”, o qual é implementado na ferramenta/biblioteca de otimização de rede LEMON [LEMON 2012].

Em [Min al. 2010] é proposto um mecanismo de roteamento em redes virtuais com base nos requisitos de usuário e estado da rede. A solução foi avaliada utilizando *switches* OpenFlow baseados em NetFPGA (*Hardware/Software* de prototipação). A proposta pode ser dividida em quatro componentes: ENVI, LAVI, componente de monitoramento de rede (*Network monitoring component*) e componente de roteamento

baseado em QoS (*QoS routing component*). Contudo, os autores não detalham os parâmetros e algoritmos utilizados nos dois últimos componentes da solução.

Em [Egilmez et al. 2012] os autores propõem uma solução chamada de OpenQoS. O OpenQoS seleciona as melhores rotas para fluxos OpenFlow baseado em requisitos de QoS. Contudo, apesar de os autores sugerirem um conjunto de parâmetros de QoS para a tomada de decisão, apenas a vazão disponível é utilizada. Em [Mattos e Duarte 2012] os autores propõem uma solução chamada QFlow. Esta proposta baseia-se no controle de recursos de roteadores virtuais OpenFlow usando Xen (XenFlow) como processamento, memória e vazão mínima e máxima de cada roteador. Em [Zhang et al. 2009] o encaminhamento de pacotes é melhorado e a QoS é fornecida através de OSPF, roteadores virtuais (VINI) e do DSCP (Differentiated Services Code Point). O artigo [Bozakov 2011] propõe o roteador virtual como um serviço e um algoritmo de custo mínimo de fluxo para lidar com a alocação dos fluxos.

Em [Puschita et al 2010], os autores propõem um modelo de QoS chamado I-NAME para escolher o melhor caminho fim-a-fim com base em perfis de QoS. No entanto, os autores não abordam soluções específicas para a virtualização de redes. Os autores em [Kassler et al 2012] propõem uma arquitetura baseada em elementos de controle inteligentes e no protocolo SIP (*Session Initiation Protocol*) para o provimento de QoS em redes OpenFlow. Contudo, a proposta não foi testada ou simulada. Com exceção do SIP nenhuma outra tecnologia foi sugerida para a implementação da solução.

Em resumo, encontrou-se apenas a proposta [Kim et al. 2010] que desenvolveu um mecanismo de priorização chamado de SSF (*Shortest Span First*) utilizado para, segundo os autores, maximizar a probabilidade de satisfazer os requisitos de desempenho de um novo fluxo e, ao mesmo tempo, minimizar o número de fluxos rejeitados. O SSF se baseia numa versão modificada do RCSP (*Rate-Controlled Static-Priority Queueing*) [Zhang e Ferrari 1993] para fornecer a prioridade entre os fluxos. Neste caso, ao contrário do RME, a proposta não fornece priorização entre pacotes de um mesmo fluxo. Os autores também propõem um controlador de QoS para redes OpenFlow. O controlador é responsável, de forma dinâmica, por dividir o tráfego entre diferentes *slices* de acordo com os requisitos de QoS (Vazão) de cada fluxo.

Em relação a propostas que utilizam NSIS e PCP podemos citar [Carmo et al. 2006] que sugere a substituição do uso de SBM (*Subnet Bandwidth Manager*) [Yavatkar et al. 2000] juntamente com RSVP por NSIS e PCP. Contudo a proposta não considera a utilização de redes virtuais. O provisionamento de qualidade de serviço fim-a-fim ainda é um desafio, e não existe uma solução consolidada, o que pode ser ratificado pela escassez de trabalhos correlatos apresentados nessa seção. Nenhum dos trabalhos citados propõe uma arquitetura para provimento de QoS fim-a-fim em redes virtuais, levando em consideração domínios administrativos distintos. Outro ponto que difere o RME dos outros trabalhos é a garantia de QoS baseado em fluxos de pacotes e não baseado apenas em portas de comutadores, como é feito na maioria dos trabalhos. O RME é avaliado em um testbed OpenFlow usando-se tanto métricas de QoS quanto métricas de QoE (Qualidade de Experiência).

3. Solução para provisão de QoS fim-a-fim em redes virtuais

Esta seção apresenta a arquitetura para provisão de QoS fim-a-fim e suas entidades. Neste caso: o RME *Switch*, o RME VLAN, o RME *Gateway*. Além disso, a interação desses módulos com o *framework* NSIS.

3.1. Visão Geral da Arquitetura

No cenário mostrado na Figura 1 existe um servidor em um domínio específico (Domínio 1), que transmite o tráfego para um cliente em outro domínio (Domínio 2). Em cada domínio existe um RME *Gateway* que faz a comunicação entre os domínios. Cada *gateway* se comunica com uma instância em execução do NSIS usando comunicação interprocessos através do D-Bus [D-Bus 2012]. Os dois *gateways* OpenFlow estão conectados diretamente a um *switch* Pronto 3290 (RME *Switch*) utilizado no testbed. Os *gateways* e o *switch* são gerenciados pelo controlador RME VLAN, o qual é baseado no controlador NOX [Gude 2008]. A seguir, são detalhados todos os passos necessários para a comunicação entre Clientes e Servidores dentro da arquitetura RME.

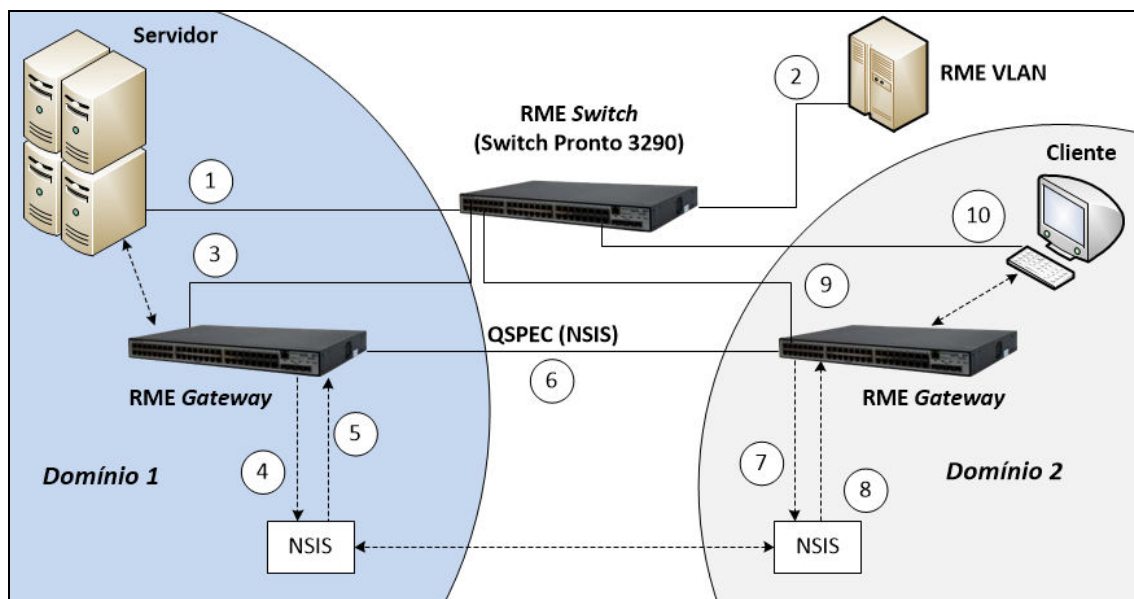


Figura 1. Visão conceitual do funcionamento do gateway Openflow e NSIS.

A comunicação entre servidor e cliente é realizada através dos seguintes passos:

1. O servidor envia o pacote para o cliente que é recebido pelo RME *Switch*.
2. Todo primeiro pacote de cada novo fluxo é enviado para o controlador (RME VLAN) que analisa o pacote e instala regras para esse fluxo nas tabelas de encaminhamento do RME *Switch*.
3. O pacote de entrada é então enviado para o RME *Gateway*.
4. Ao receber o pacote, o RME *Gateway* encaminha-o para o NSIS. Nesse instante o NSIS realiza o mapeamento L2/L3. Esse mapeamento consiste na conversão do

campo PCP (*Priority Code Point*) (Camada 2) para o campo DSCP (*Differentiated Services Code Point*) (camada 3) [Baker 2010].

5. Após o passo anterior, o NSIS envia o pacote mapeado para o cliente (outro domínio administrativo). Além disso, o fluxo é instalado na tabela de encaminhamento do RME *Gateway* para que todos os pacotes com a mesma característica não precisem mais ser enviado para o NSIS.
6. O Pacote passa de um domínio para o outro através dos RME *Gateways*.
7. O RME *Gateway* no domínio 2 recebe o pacote e envia para a instância NSIS de destino.
8. A instância NSIS de destino interpreta o pacote recebido e envia para o RME *Switch* de destino. Um novo fluxo OpenFlow é instalado na tabela de encaminhamento no RME *Gateway* e a reserva de recurso é feita para este fluxo.
9. O pacote é encaminhado ao cliente através do *switch* Pronto 3290 (RME-*Switch*).
10. Por fim, o usuário recebe o pacote enviado pelo servidor com QoS garantida em todo o trajeto. A partir deste momento, todo o pacote priorizado que sair do servidor para o cliente também terá garantias de QoS em ambos os domínios.

3.2. O RME *Switch*

O tráfego em uma rede é originado por uma variedade de aplicações com diferentes requisitos de desempenho, e uma forma de atender às necessidades destas aplicações consiste na utilização de tipos de tráfego pré-definidos ou Classes de Serviço (CoS – *Class of Service*) para permitir a implementação de esquemas de priorização.

O grupo de trabalho IEEE 802.1p dedicou-se ativamente entre 1995 e 1998 para desenvolver um padrão de CoS, e mesmo não existindo um padrão com esse nome publicado pelo IEEE, o termo IEEE 802.1p é utilizado como sinônimo de CoS. O esquema de priorização IEEE 802.1p utiliza um campo de 3 bits, do cabeçalho IEEE 802.1D/Q [IEEE 2004] [IEEE 2011], chamado PCP (*Priority Code Point*). O PCP especifica um valor de prioridade entre 0 e 7 (oito classes) usados para diferenciar o tráfego.

Um das contribuições deste artigo é a implementação de uma solução de priorização de pacotes, baseada no PCP, no *firmware* do *switch* OpenFlow. Para tal, foi necessário utilizar uma distribuição de código aberto com suporte a OpenFlow e compatível com o “*hardware* de prateleira”. Estes fatores motivaram a adoção da Indigo [OpenFlowhub.org 2012]. O Indigo foi criado em Stanford e o projeto disponibiliza uma plataforma de desenvolvimento chamada IODS (*Indigo Open Development System*), que permite modificar e criar imagens do Indigo para os switches compatíveis com OpenFlow. O Indigo é baseado na implementação de referência do OpenFlow e atualmente dispõe de todas as características requeridas pela especificação OpenFlow 1.0. Entretanto, esta escolha também alguns limitações. Uma delas foi a impossibilidade da construção de um escalonador otimizado para o *hardware* adotado, pois os códigos fonte dos *drivers* não estão disponíveis, assim, optou-se por desenvolver este mecanismo na camada de *software*.

Tendo em vista a necessidade de uma solução robusta e de alta desempenho, também foi adotada uma fila de prioridade como escalonador entre os diversos fluxos. Essas filas são estruturas de dados que armazenam elementos associados às prioridades, sendo reordenadas a cada novo elemento inserido de forma que a sequência de remoção destes elementos ocorre de acordo com suas respectivas prioridades. Existem vários tipos de filas de prioridades, a mais simples são filas lineares encadeadas em que os elementos estão ordenados por prioridade decrescentes. Também são possíveis filas de prioridades fixas inserindo elementos nas filas que correspondem a uma específica prioridade. Porém, as mais robustas são baseadas em árvores. Por isso, optou-se por utilizar a PQLib [PQLib 2012], uma biblioteca que implementa uma fila de prioridades baseada em árvores escrita em C e adotada por projetos como o servidor Web Apache e o sistema de gerenciamento de bancos de dados PostgreSQL.

3.3. RME VLAN

O uso do PCP em uma rede de *testbed* exigiu a criação de VLANs (*Virtual Local Area Network*) utilizando-se o padrão IEEE 802.1Q entre os *hosts* utilizados. Este padrão define um sistema de *tagging* VLAN em *frames Ethernet*, as *tags* criam um domínio *broadcast* limitado à *tag* VLAN, ou seja, cada *tag* forma uma rede particular e separadas de outras redes.

Contudo, VLANs juntamente com alguns outros recursos da Ethernet como o *Spanning Tree* e o protocolo ARP (*Address Resolution Protocol*) ainda não são completamente suportados por todos os comutadores OpenFlow, como o Pronto 3290, e os diversos controladores, como o NOX. Por isso, tornou-se necessário o desenvolvimento de um componente (aplicação de rede) para o NOX capaz contornar algumas destas limitações da qual chamamos de RME-VLAN.

Um *switch* OpenFlow é dividido em parte de *hardware* e parte de *software*, as tabelas de encaminhamento ficam na parte de *hardware* chamada de Plano de Encaminhamento e o controlador, neste caso o NOX, se comunica com a parte de *software* do *switch* chamada de Plano de Dados através de um canal constituído pelo protocolo OpenFlow que pode ser ou não seguro (criptografado).

O NOX tem a função de manipular (controlar) a tabela de encaminhamento do *switch* OpenFlow. Todos os pacotes que chegam ao *switch* OpenFlow são analisados para determinar se pertencem a algum fluxo da tabela de encaminhamento do *switch*. Caso positivo, o pacote é enviado de acordo com a regra instalada, caso contrário, o pacote é enviado ao NOX para que este analise e instale regras específicas para aquele pacote.

3.4. RME-Gateway

Os dispositivos com suporte ao OpenFlow e ao NSIS atuam como *gateways* realizando o mapeamento entre domínios dos *bits* de prioridade usados na rede interna (PCP). Conforme dito anteriormente, isto é feito via NSIS QSPECs e DSCP. Indiretamente, também ocorre a reserva de recursos via RME *Gateway* através da comunicação entre o *switch* OpenFlow no espaço do usuário e o controlador OpenFlow como mostra a Figura 2.

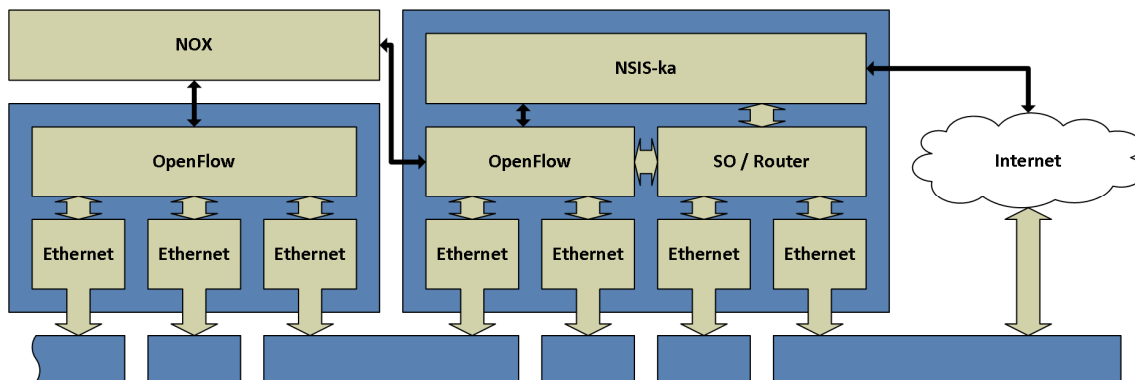


Figura 2. Reserva de recursos no NSIS-ka.

Em resumo, pode-se dizer que a principal atribuição deste módulo dentro da arquitetura proposta é mapear a QoS entre as camadas da pilha de protocolo, mantendo as respectivas atribuições dos seus componentes. O NSIS é responsável pela negociação e reserva de recursos interdomínios, e o conjunto OpenFlow/NOX pelas garantias de qualidade intradomínio. Para fazer a comunicação entre o OpenFlow e o NSIS-ka [NSIS-ka 2012] (implementação aberta do NSIS) utilizou-se o D-Bus, conforme dito na Seção 3.1. Vale ressaltar que esta proposta não se destina a modificar os protocolos de sinalização, mas apenas torná-los complementares. O *testbed* assim como os resultados obtidos é abordado na Seção 4.

4. *Testbed* e Resultados Obtidos

Esta seção aborda o *testbed* desenvolvido para validação da proposta bem como os resultados obtidos. Os *hosts* da rede foram implementados via máquinas virtuais em um *bare-metal hypervisor* para que o cenário pudesse ser heterogêneo em termos de sistemas operacionais, quantidade de interfaces de rede de um *host*, memória RAM etc. Neste caso, foi utilizado o VMware vSphere Hypervisor (ESXi) 5.0 [VMware 2012] em três servidores. Também foram utilizados, um *switch* OpenFlow (Pronto 3290) e quatro PCs, um para acesso central aos sistemas de gerenciamento da rede, e os demais como servidores de streaming e *upload* de dados.

Foram criados dois *testbeds*. Para verificarmos e validarmos a implementação do PCP na plataforma Indigo, o primeiro *testbed* foi realizado com um servidor de *upload* e dezesseis clientes, todos diretamente conectados através do *RME-Switch* (Pronto 3290). O segundo *testbed* foi realizado com o *RME-Switch* e o *RME-Gateway* conforme mostrado anteriormente na Seção 3.4. O *RME-Switch* foi usado no segundo *testbed* como *switch* central para que houvesse um tráfego ainda maior que o do primeiro *testbed* percorrendo este. Com isso, pudemos verificar a escalabilidade das nossas soluções.

Para termos uma visão da qualidade do ponto de vista da administração da rede e do usuário, esta seção é dividida em duas partes: Uma com os resultados obtidos a partir da avaliação de QoS (Seção 4.1) e a segunda com os resultados da avaliação de QoE (Seção 4.2) conforme a seguir.

4.1 Resultados da Avaliação de QoS

Para testarmos a priorização do tráfego e NSIS provendo QoS interdomínios, foram criados clientes com acesso pelo *switch* Pronto a um servidor de *upload*. Cada cliente envia para este servidor, simultaneamente, um tráfego UDP priorizado e um não priorizado através da ferramenta Iperf [Iperf 2012]. Neste caso, foram realizados três testes:

- **Cenário 1:** Tráfego priorizado total de 32 Mbps e não priorizado de 992 Mbps.
- **Cenário 2:** Tráfego priorizado total de 64 Mbps e não priorizado de 960 Mbps.
- **Cenário 3:** Tráfego priorizado total de 128 Mbps e não priorizado de 896 Mbps.

As relações entre o tráfego priorizado e não priorizado são de 1:31 (Cenário 1), 1:15 (Cenário 2) e 1:7 (Cenário 3), respectivamente. O Iperf também foi utilizado para a medição da perda de pacotes. Em todos os casos o tráfego priorizado não sofreu perda ao contrário do tráfego não priorizado que sofreu uma perda média de pacotes de 4,77% no Cenário 1, de 6,44% no Cenário 2 e de 16,47% no Cenário 3 conforme mostram as Figuras 3 e 4. O comportamento é esperado considerando que a vazão total não atinge o limiar teórico de 1 Gbps por enlace e quanto maior o tráfego priorizado maior tende a ser o descarte de pacotes não priorizados.

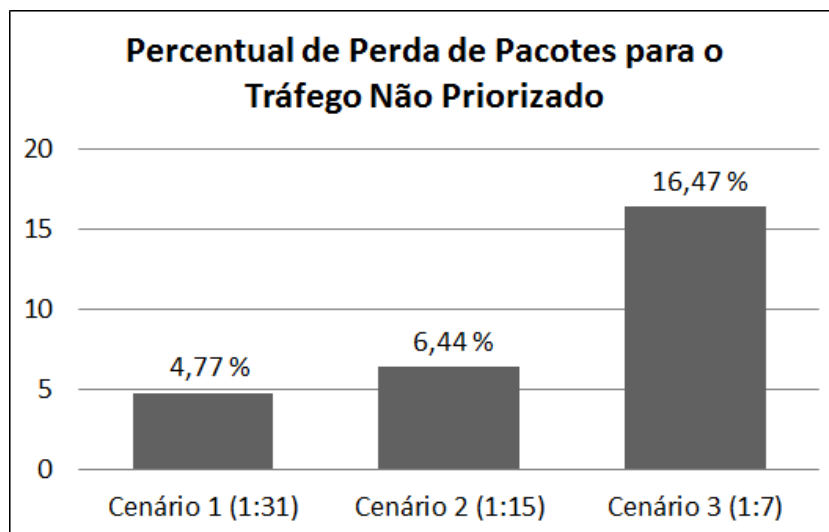


Figura 3. Percentual de Perda de Pacotes versus proporção entre o tráfego priorizado e o não priorizado.

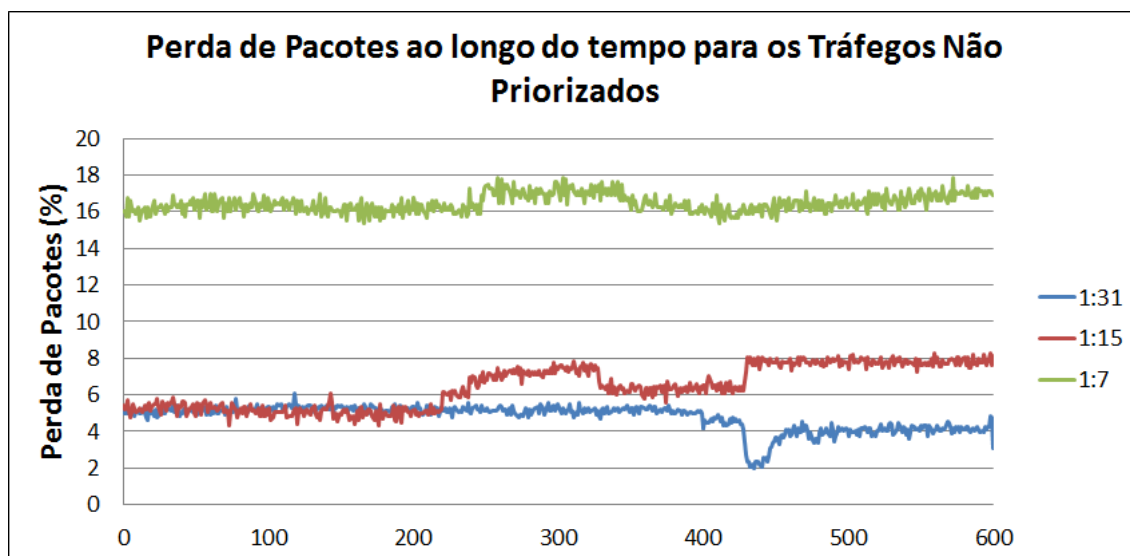


Figura 4. Percentuais de Perda de Pacotes ao longo do tempo proporção para tráfego não priorizado.

Os testes mostraram que com a implementação do *switch* no espaço do usuário permite-se que a vazão máxima fica em torno dos 30 Mbps. As médias foram de 0,804% para o tráfego priorizado e de 4,453 % para o tráfego não priorizado. A avaliação da QoE do RME-Gateway e abordado na Seção 4.2.

4.2 Resultados da Avaliação de QoE

Os aspectos e métricas de QoS (*e.g.*: vazão, atraso e *jitter*) são importantes para análises de desempenho de protocolos e arquiteturas do ponto de vista da rede, mas não em termos de percepção humana. Desta forma, as novas arquiteturas não estão sendo mais avaliadas apenas em termos de aspectos de QoS, mas também quanto ao suporte à Qualidade de Experiência (*Quality of Experience*). O PSNR (*Peak Signal to Noise Ratio*) [WINKLER 2005] é uma métrica tradicional de QoE que estima a qualidade do vídeo em decibéis, comparando o vídeo original com o vídeo recebido pelo usuário. Para cada faixa de valores de PSNR, há uma qualificação para o vídeo que foi recebido conforme mostra a Tabela 1.

Tabela 1. Valores de classificação do PSNR.

PSNR(dB)	> 37	31 – 37	25 – 31	20 – 25	< 20
Qualidade	Excelente	Bom	Aceitável	Pobre	Péssimo

Nos servidores foi instalado o Darwin Streaming Server [DSS 2012] para fornecer serviços de *streaming* em uma rede através dos protocolos RTP e RTSP sobre UDP. Para avaliação do vídeo no cenário foi utilizado trailers do filme “Os Vingadores” [Os Vingadores (Trailer) 2012]. Todos possuem a mesma duração, o mesmo codec e o mesmo número de frames, porém com resoluções diferentes conforme mostra a Tabela 2.

Tabela 2. Vídeos utilizados nos testes.

Nome do vídeo	Resolução	FPS	Codec	Bitrate	Duração	Frames
Vídeo 1	640x360	24	H.264/AVC	488 kbps	2m e 3s	2970
Vídeo 2	1280x720	24	H.264/AVC	1732 kbps	2m e 3s	2970
Vídeo 3	1920x1080	24	H.264/AVC	3371 kbps	2m e 3s	2970

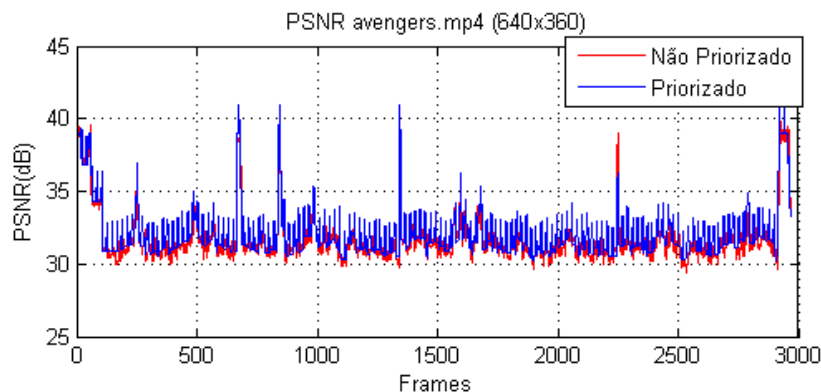
Nas medições, exatamente o mesmo vídeo era transmitido pelas redes Priorizada (PCP=5) e Não Priorizada (PCP=0). Cada medição foi repetida 10 vezes. Os vídeos além de transferidos também foram armazenados nos clientes no formato mpeg-4. Como vimos na avaliação da priorização do tráfego, a rede sem prioridade perde uma grande quantidade de pacotes. No uso de aplicações de vídeo isso implica na perda de *frames* e, conseqüentemente, na degradação da QoE. A Tabela 3 mostra que o vídeo recebido na rede sem prioridade perde uma grande parte das informações, isso se torna mais evidente quando a rede é mais exigida, no caso do vídeo em HD (1920x1080 *pixels*), enquanto que o vídeo na rede com prioridade tem pouco ou nenhuma perda.

Tabela 3. Vídeos utilizados nos testes.

	Original	Recebido Sem Priorização	Recebido Com Priorização
Vídeo 1 (640x360)	9,5 MB	9,5 MB	9,5MB
Vídeo 2 (1280x720)	30,2 MB	25,8 MB±0,2MB	30 MB±0,2MB
Vídeo 3 (1920x1080)	56,7 MB	30,9 MB±0,32MB	55,2MB±1,2MB

O cálculo do PSNR foi feito com a ferramenta MSU VQMT [MSU 2012]. A Figura 5 mostra o PSNR dos vídeos recebidos nas redes Priorizada e Não Priorizada. Percebe-se que os valores neste caso são semelhantes, isso se deve ao fato do Vídeo 1 com resolução de 640x360 não exigir muitos recursos da rede.

A Figura 6 mostra o PSNR do Vídeo 2 (1280x720), neste caso, torna-se mais evidente o efeito da priorização, pois o vídeo com esta resolução precisa de mais recursos para trafegar pela rede. Porém, o PSNR médio ainda está em torno de 25, o que torna o vídeo aceitável segundo os valores de PSNR mostrados anteriormente (Tabela 1).

**Figura 5. PSNR dos vídeos 640x360 recebidos na rede priorizada e não priorizada.**

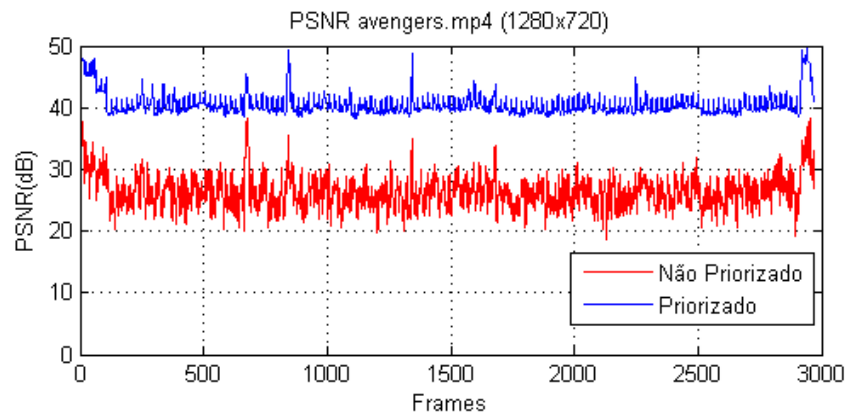


Figura 6. PSNR dos vídeos 1280x720 recebidos na rede priorizada e não priorizada.

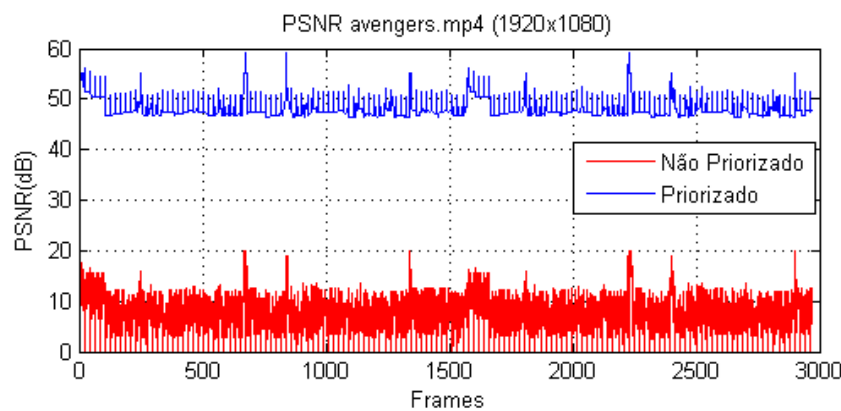


Figura 7. PSNR dos vídeos 1920x1080 recebidos na rede priorizada e não priorizada.

Com o vídeo em alta definição os efeitos da priorização são muito visíveis (Figura 7). Este vídeo exige muito recursos da rede como uma alta vazão (em média 3371 Mbps), por exemplo, e a o esquema de priorização reserva a maior parte desta para a rede priorizada tornando o vídeo não priorizado muito degradado. Mesmo assim, há perda de QoE na Rede Priorizada.

A Tabela 4 mostra os PSNR médios obtidos. Se considerarmos os PSNR médios entre as redes Priorizada e Não Priorizada para cada um dos vídeos transmitidos. Os ganhos entre estes são de 1,4%, 54% e 520% para os Vídeos 1, 2 e 3 respectivamente.

Tabela 4. PSNR médios dos vídeos recebidos pelas Redes Priorizada e Não Priorizada.

Vídeos	PSNR (Média)	
	Rede Priorizada	Rede Não Priorizada
Vídeo 1 (640x360)	32.1412±1.90	31.6751±2.71
Vídeo 2 (1280x720)	40.5352±2.70	26.3133±5.0234
Vídeo 3 (1920x1080)	47.9077±1.78	7.7237±3.65

A perda ou duplicação dos pacotes na rede ocasionam queda da QoE oferecida ao usuário. A figura abaixo mostra um exemplo de *frame* do filme em alta definição. Percebe-se que este *frame* na Rede Priorizada não difere do *frame* original, porém a perda de pacotes faz com que o vídeo recebido na Rede Não Priorizada seja um *frame* degradado.



Figura 11. Comparação do 1412º frame entre o vídeo original e os recebidos nas redes priorizada e não priorizada.

5. Conclusão

O surgimento da virtualização de redes e do conceito de redes definidas por software implementado pela plataforma OpenFlow trouxe grande inovação para as redes tradicionais. Porém, por serem um conceito em desenvolvimento, novas soluções precisam ser propostas visando esses novos conceitos. Neste contexto, garantir qualidade de serviço fim-a-fim é um dos desafios. Este trabalho apresentou uma proposta de solução para a provisão de QoS fim-a-fim na plataforma OpenFlow através do PCP e do *framework* NSIS. A solução proposta tem o objetivo de garantir QoS em redes com tecnologia OpenFlow, além de permitir a extensão dessa garantia em domínios administrativos distintos. As decisões de projeto foram tomadas tendo em mente a adoção de conceitos previamente estabelecidos como o modelo de QoS do IEEE 802.1p e a sinalização do NSIS.

A proposta foi avaliada em um testbed OpenFlow utilizando diversas ferramentas e dispositivos usados nas empresas e universidade, o que coloca a proposta é um cenário bem realista. Na avaliação, mostrou-se que a proposta possui um ganho significativo tanto considerando avaliação de métricas de QoS, que refletem o comportamento da rede, quando avaliação de métricas subjetivas de QoE que estão mais próximas aos critérios dos usuários. Como trabalhos futuros, pretendemos utilizar o IPv6 e o mapeamento e negociação de QoS não apenas nas redes Ethernet, mas também em cenários heterogêneos com tecnologias como por exemplo, o IEEE 802.11 e IEEE 802.16.

6. Referências

- Gozdecki, J. et al. (2003) "Quality of service terminology in IP networks," Communications Magazine, IEEE , vol.41, no.3, pp.153,159, March.
- OpenFlow Switch Specification - Version 1.0.0. (2012)
<http://www.OpenFlow.org/documents/OpenFlow-spec-v1.0.0.pdf>, Março.
- Schonwalder, J. et al. (2009) "Future Internet = content + services + management," Communications Magazine, IEEE , vol.47, no.7, pp.27,33, July.
- IEEE (2011) "IEEE Standard for Local and metropolitan area networks--Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks". IEEE Standard 802.1Q-2011.
- Fu, X. et al. (2005) "NSIS: a new extensible ip signaling protocol suite" Communications Magazine, IEEE, v. 43, n. 10, p. 133-141.
- Civanlar, S. et al. (2010) "A QoS-Enabled OpenFlow Environment for Scalable Video Streaming", IEEE Globecom 2010 Workshop on Network of the Future (FutureNet-III), Miami, USA.
- Gude, N. et al. (2008). "NOX: towards an operating system for networks" In: SIGCOMM Comput. Commun. Rev. 38, 3, July, p. 105-110.
- Egilmez, H.E. et al. (2011) "Scalable video streaming over OpenFlow networks: An optimization framework for QoS routing," Image Processing (ICIP), 2011 18th IEEE International Conference on , vol., no., pp.2241,2244, 11-14 September.
- LEMON (2012) "Library for Efficient Modeling and Optimization in Networks." <http://lemon.cs.elte.hu>, March.
- Min, S.H. et al. (2010) "Implementation of a Programmable Service Composition Network using NetFPGA-based OpenFlow Switches", 1st ASIA NetFPGA Developer's workshop, Korea, June.
- Egilmez, H.E. et al. (2012) "OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software-Defined Networks", *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific* , vol., no., pp.1,8, 3-6 December.
- Mattos, D. M. F., and Duarte, O. C. M. B. (2012) "QFlow: Um Sistema com Garantia de Isolamento e Oferta de Qualidade de Serviço para Redes Virtualizadas", in XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC'2012, pp. 536-549, Ouro Preto, MG, Brazil, May.
- Zhang, Y et al. (2009) "A QoS-Oriented Network Architecture Based on Virtualization," Education Technology and Computer Science, 2009. ETCS '09. First International Workshop on , vol.2, no., pp.959-963, 7-8 March.
- Bozakov, Z. (2011) "Architecture and algorithms for virtual routers as a service," Quality of Service (IWQoS), 2011 IEEE 19th International Workshop on , vol., no., pp.1-3, 6-7 June. Civanlar, S. et al. (2010) "A QoS-Enabled OpenFlow Environment for Scalable Video Streaming", IEEE Globecom 2010 Workshop on Network of the Future (FutureNet-III), Miami, USA.

- Puschita, E. et al. (2010) "An Innovative QoS Paradigm Based on Cognitive In-network Management of Resources for a Future Unified Network Architecture: I-NAME QoS Model," *Advances in Future Internet (AFIN)*, 2010 Second International Conference on , vol., no., pp.37-43, 18-25 July.
- Kassler, A. et al. (2012) "Towards QoE-driven Multimedia Service Negotiation and Path Optimization with Software Defined Networking" *International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, September.
- Kim, W. et al. (2010) "Automated and Scalable QoS Control for Network Convergence," *Proceedings of the INM/WREN'10*.
- Zhang, H. and Ferrari, D. (1993) "Rate-controlled static-priority queueing" *In Proceedings of the IEEE INFOCOM*.
- Carmo, M. et al. (2006) NSIS-based quality of service and resource allocation in Ethernet networks. In: Braun T et al. (ed) *WWIC 2006*, LNCS 3970. Springer, Berlin, pp 132–142.
- Yavatkar, R. et al (2000) "SBM (subnet bandwidth manager): A protocol for RSVP-based admission control over IEEE 802-style networks," May, internet RFC 2814.
- D-Bus (2012) <http://www.freedesktop.org/wiki/Software/dbus>, Março.
- Baker, F.; Polk, J.; Dolly, M. (2010) A Differentiated Services Code Point (DSCP) Capacity-Admitted Traffic. IETF RFC 5865.
- IEEE (2004) "IEEE Standards for Local and Metropolitan Area Networks: Media access control (MAC) bridges" IEEE Standard 802.1D-2004.
- OpenFlowhub.org (2012) Indigo, <http://www.OpenFlowhub.org/display/Indigo/Indigo++Open+Source+OpenFlow+Switches>, Março.
- PQLib (2012) "Priority Queue Library", <http://www.ohloh.net/p/pqlib>, Março.
- NSIS-ka (2012) <http://nsis-ka.org/>, Março.
- VMware (2012) "VMware vSphere Hypervisor (ESXi) 5.0," <http://www.vmware.com/br/products/datacenter-virtualization/vsphere/mid-size-and-enterprise-business/overview.html>, Março.
- Iperf (2012) <http://sourceforge.net/projects/iperf/>, Março.
- Winkler, S. (2005) Perceptual video quality metrics – a review, in *Digital Video Image Quality and Perceptual Coding*, cap. 5, CRC Press.
- DSS (2012) "Darwin Streaming Server" <http://dss.macosforge.org/>, Março.
- Os Vingadores (Trailer) (2012) Direção: Joss Whedon. [S.l.]: Walt Disney Home Entertainment, 2012. 1 DVD (2m e 30 s) NTSC, color. Título original: The Avengers.
- MSU (2012) "MSU Video Quality Measurement Tool" http://compression.ru/video/quality_measure/video_measurement_tool_en.html, Março.