Rede SDN-OpenFlow para o Caso de um ISP: Desafios e Oportunidades

Fernando López-Rodríguez[1] e Divanilson R. Campelo[2]

***Resumo*—Este artigo propõe uma arquitetura para um ISP (*Internet Service Provider*) através da utilização de SDN (*Software Defined Networking*) utilizando OpenFlow. Desta forma tenta-se criar uma rede com menor processamento em seus equipamentos e maiores potencialidades de engenharia de tráfego que as redes de hoje, e ao mesmo tempo sem os problemas que podem existir nas redes implementadas com OpenFlow como sobrecarga dos controladores, atrasos adicionais na criação de fluxos e a excessiva dependência da rede ao controlador. Os seguintes atributos foram levados em consideração no projeto da arquitetura: requisitos de robustez, velocidade de resposta de encaminhamento, *jitter*, engenharia de tráfego e carga por processamento nos equipamentos.**

Palavras-Chave*—*Arquitetura, ISP, OpenFlow, Robustez, velocidade de encaminhamento, jitter, engenharia de tráfego, sobrecarga, dependência ao controlador.

***Abstract*—This article proposes an architecture for an ISP (*Internet Service Provider*) utilizing SDN-OpenFlow. With this, we attempt to create a network with less equipment processing and more traffic engineering potentiality than today's one, and at the same time without the problems that might exist in networks implemented with OpenFlow, as controller overload, additional delays in flows creation and excessive controller network dependence. The following attributes have been taken into consideration in the design: network robustness requirements, routing speed response, jitter, delay, traffic engineering and equipment processing load.**

Keywords*—*Architecture, ISP, OpenFlow, Robustness, routing speed, jitter, traffic engineering, overload, controller dependence.

I. INTRODUÇÃO

Provedores de Serviço de Internet (*Internet Service Providers*, ISP) são, por definição, organizações que comercializam serviços de Internet a clientes. Em geral, um ISP compra um serviço denominado de "trânsito" de outros ISPs já conectados à rede; após estabelecida a relação de trânsito, um ISP pode se conectar à "Internet Global" mediante outro ISP. Tipicamente, uma rede de ISP possui dimensão e volume de tráfego significativos, além de uma diversidade de serviços ofertados, tais como VoIP (*Voice over IP*), vídeo, navegação Web, entre outras.

Em termos arquitetônicos, as redes de ISPs possuem mecanismos de controle automáticos e totalmente distribuídos, tais como protocolos de encaminhamento (por ex., *Internal Border Gateway Protocol* -- I-BGP, *Open Shortest Path First* -- OSPF, *Intermediate System to Intermediate System* -- ISIS) e de sinalização (por ex., *Label Distribution Protocol* -- LDP, *Resource Reservation Protocol* -- RSVP), os quais devem estar implementados em cada roteador da rede. Um roteador, que é o elemento de controle responsável por construir tabelas de roteamento e pelo encaminhamento de tráfego, possui tipicamente cargas de processamento elevadas nestas redes. O desempenho da rede de um ISP degrada-se sensivelmente quando há congestionamento de tráfego, sendo necessário o sobreaprovisionamento de recursos para que determinados acordos de níveis de serviço (*Service Level Agreement* -- SLA) e requisitos de qualidade de serviço (*Quality of Service* – QoS) entre a rede IP (*Internet Protocol*) e seus clientes sejam alcançados. Ainda, redes de grande porte, como as de ISPs, são complexas e caras em infraestrutura e operação, pois os ISPs necessitam manter várias redes diferentes com grupos de administradores especializados para o seu gerenciamento. Em geral, para os ISPs é uma tarefa difícil diferenciar serviços oferecidos por outros ISPs, na medida em que as redes são constituídas de equipamentos do mesmo grupo de fabricantes, os quais possuem o mesmo conjunto de funcionalidades, o que limita, por exemplo, as possibilidades de engenharia de tráfego. A "inteligência" da rede está oculta nos equipamentos, tornando as inovacões na rede extremamente lentas e amarradas aos interesses dos fabricantes.

Como caminho alternativo, o protocolo OpenFlow [1][2] proporciona uma interface aberta, através da qual equipamentos clientes podem interagir com um controlador dotado de uma visão global da rede e encarregado de construir as tabelas de fluxo para o encaminhamento de pacotes em todos os clientes. Dessa forma, é possível separar os planos de dados (clientes OpenFlow) e de controle (controlador). O protocolo Openflow proporciona inúmeras possibilidades para engenharia de tráfego [3], unificação do plano de controle para diferentes tipos de redes, como IP e WDM (*Wavelength Division Multiplexing*) [4][5][6], gerenciamento de mobilidade [7], entre outras. Tais possibilidades têm atraído o interesse de diferentes fabricantes de redes [8], alguns dos quais já estão incorporando OpenFlow em seus produtos [9].

Nesta direção, este trabalho tem como objetivo apresentar os desafios e oportunidades da utilização de OpenFlow em uma arquitetura de rede de um ISP. Para um ISP, a arquitetura OpenFlow padrão parece não ser apropriada em razão de: a) sua característica centralizadora, a qual requer que cada novo fluxo seja processado pelo controlador; b) sua dependência excessiva do controlador, o que reduz em grande medida a robustez da rede; c) degradação de desempenho, pois cada fluxo tem que aguardar a resposta do controlador para seu encaminhamento; e d) possíveis sobrecargas no controlador devido à grande quantidade de tráfego que o ISP encaminha. Este artigo propõe uma solução para estes problemas através de uma arquitetura de gerenciamento de tráfego com e sem QoS.

Os autores agradecem ao CNPq o apoio.

[1]Departamento de Engenharia Elétrica, Universidade de Brasília (UnB), Caixa Postal 4386 – 70910-900, Brasília, DF, Brasil, E-mail: fernando.lopez@aluno.unb.br.

[2]Centro de Informática (CIn), Universidade Federal de Pernambuco (UFPE), Av. Jornalista Anibal Fernandes, s/n, Cidade Universitária, 50740-560, Recife, PE, E-mail: dcampelo@cin.ufpe.br

A continuação deste artigo é organizada conforme descrito a seguir. A Seção II aborda a arquitetura proposta no artigo, com destaque para a lógica geral de funcionamento, os mecanismos proativos que permitem alcançar todas as redes destino para encaminhamento interno e MPLS (*Multiprotocol Label Switching*), e a lógica proposta para o gerenciamento para o tráfego QoS, que possibilita o estabelecimento de fluxos individuais sem adição de tempos de espera. A Seção III sugere uma arquitetura alternativa com equipamentos híbridos. A Seção IV realiza uma modelagem que permite quantificar atrasos no caso de OpenFlow padrão. A Seção V apresenta os principais resultados do artigo mediante uma tabela comparativa entre a arquitetura proposta com equipamentos puramente OpenFlow e com equipamentos híbridos, uma rede configurada com OpenFlow padrão e uma rede com controle distribuído clássico. Por fim, a Seção VI apresenta as conclusões do artigo e as perspectivas de trabalhos futuros.

II. ARQUITETURA PROPOSTA

A. *Considerações gerais*

Neste trabalho, propomos uma arquitetura de rede de ISP baseada em OpenFlow com plano de dados MPLS [4]. A tecnologia MPLS já contempla a abstração do plano de dados mediante o conceito de fluxos, adequada, portanto, à utilização com OpenFlow. O próprio protocolo OpenFlow vem sendo desenvolvido nesta direção [2], com a incorporação das funcionalidades necessárias que permitem a utilização do plano de dados MPLS. Dentre as funcionalidades adicionadas, existe a possibilidade de que as regras de classificação e marcação (*matching*) de tráfego comparem, modifiquem, adicionem e apaguem rótulos MPLS.

O fato de OpenFlow permitir definição e controle de fluxos individuais incrementa as potencialidades e precisão de funções de gerenciamento. Contudo, esta manipulação individual também leva a um maior processamento, atraso e sobrecarga na rede. É importante destacar que cada fluxo adicional requer a configuração da regra correspondente nas tabelas de fluxos de todos os equipamentos intervenientes no encaminhamento fim a fim, com várias comunicações com o controlador. Por esta razão, é importante distinguir e definir as características dos diferentes tipos de tráfego na rede, analisando qual tráfego requer um tratamento fluxo a fluxo e qual um tratamento geral, com o objetivo de diminuir a carga no controlador.

B. *Lógica geral de funcionamento*

Na arquitetura proposta, há uma tabela inicial indexada como 0 em cada cliente OpenFlow, a qual encaminha o processamento a uma lógica de gerenciamento especializada (ver Fig. 1). Na tabela 0, são aplicadas regras gerais para classificar os pacotes de entrada com o menor número de regras de coincidência possíveis, as quais podem verificar, por exemplo, presença do rótulo MPLS, portas específicas de VoIP, vídeo ou outras aplicações, grupo de direções IPs internas, IP *precedence* do cabeçalho IP ou EXP de MPLS, entre outras.

As lógicas de gerenciamento específicas escolhidas são:

- Encaminhamento interno: permite alcançar todas as redes internas pertencentes ao ISP, como as redes de interconexão, redes de serviços internos e redes de clientes corporativos ou residenciais. Nesta lógica, o encaminhamento é baseado no cabeçalho IP, mas é possível examinar outros campos e realizar modificações neles (exemplo, adicionar rótulos MPLS). Esta lógica é desenvolvida na Seção II-C.
- Encaminhamento MPLS: caso em que um pacote de entrada é encaminhado em função do seu rótulo MPLS. O resultado do encaminhamento desta lógica é muito similar ao obtido nas redes atuais com protocolos como RSVP ou LDP. Esta lógica é desenvolvida na Seção II-C.
- Encaminhamento de QoS: permite tratar fluxos que requeiram QoS de ponta a ponta em razão dos seus requisitos de serviço. Exemplos são fluxos de comunicação de VoIP, vídeo, TV digital e aplicações de controle em tempo real. Esta lógica, uma das principais contribuições deste artigo, é desenvolvida nas Seções II-D e II-E.
- Outros Encaminhamentos não desenvolvidos neste artigo como: Encaminhamento externo que permite alcançar todos os destinos fora do sistema autônomo; Encaminhamento VPN (Virtual Private Network) que permitem interconectar diferentes pontos clientes mantendo políticas de isolamento; Encaminhamento com regras de segurança: utilizado para o tráfego que necessita ser autorizado antes de ser encaminhado.

Cada uma destas lógicas de gerenciamento é implementada por uma tabela ou grupo de tabelas OpenFlow interligadas, as quais se encontram dentro da linha de processamento de OpenFlow em cada um dos equipamentos (Ver Fig. 1). Estas tabelas variam em função do tipo de cliente OpenFlow e do lugar em que ele se situa na rede. Um exemplo é o caso dos *OpenFlow Label Switch Routers*, que não requerem conhecimento de BGP ou VPN para o encaminhamento, pois encaminham os pacotes considerando somente o rótulo MPLS.

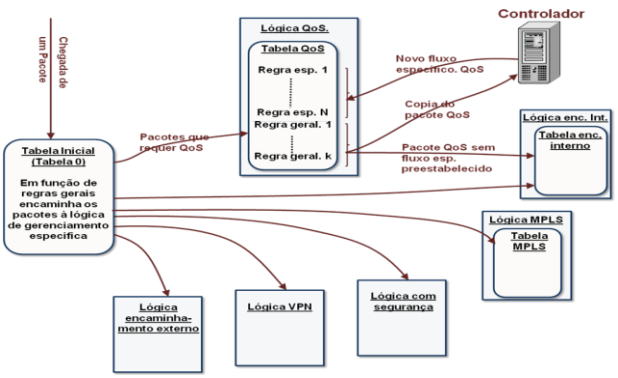

Fig. 1. Arquitetura geral e lógica de gerenciamento QoS.

C. *Lógica de gerenciamento para encaminhamento interno e MPLS*

Estas lógicas são formadas por tabelas com informação de encaminhamento interno para todas as redes destino, similar ao que ocorre nas redes atuais (isto é, não fluxo a fluxo). Estas tabelas são construídas dinamicamente pelo controlador em cada um dos clientes OpenFlow e atualizadas pelo controlador quando uma mudança topológica acontece. Para isso, o controlador deve possuir a informação de todas as redes diretamente conectadas a cada equipamento (topologia) e deve ser informado pelos clientes OpenFlow de qualquer mudança.

Com o conhecimento topológico será possível para o controlador executar um protocolo de roteamento interno conveniente (como OSPF, ou ISIS, ou um novo) para descobrir

como alcançar todas as redes a partir de cada um dos nós, e assim construir as tabelas para a lógica de gerenciamento para o encaminhamento interno em cada um dos clientes OpenFlow. Em seguida, o controlador estabelecerá ligações dos rótulos MPLS com as redes, formando o que é denominado FEC (*Forwarding Equivalent Class*) na terminologia MPLS. Com a configuração adequada de engenharia de tráfego no controlador, é possível construir e preencher as tabelas que compõem a lógica de gerenciamento para o encaminhamento MPLS (como é realizado nas redes atuais com LDP e RSVP).

Estas duas tabelas deverão incluir os campos de coincidência (*matching*), tempos de espera, instruções e, dentro destas, as listas de ações de modificação, adição ou remoção de rótulos MPLS correspondentes. Nos casos em que o protocolo de roteamento encontrar mais de um caminho ótimo, estes poderão ser considerados e adicionados nas tabelas correspondentes. Para isto, é possível configurar no OpenFlow as *Group Tables* [2], que permitem tratar o tráfego com um algoritmo de balanceamento adequado e assim, melhorar a distribuição da carga na rede.

É importante ressaltar que o algoritmo de encaminhamento só é conhecido e executado no controlador, que é o responsável por obter o encaminhamento para todos os destinos para cada um dos nós. Os dispositivos de rede não precisam intercambiar informação de roteamento e mudança topológica, não sendo necessário aguardar tempos de espera que assegurem a certa propagação da informação; é suficiente informar ao controlador as mudanças para se obter a nova convergência. Por fim, não é necessário utilizar complexos algoritmos de propagação e encaminhamento como OSPF-TE (*OSPF Traffic Engineering*) ou ISIS-TE (*ISIS Traffic Engineering*), nem de distribuição de etiquetas como LDP ou RSVP.

D. Lógica de gerenciamento para QoS

Como foi mencionado na Seção II-B, o tráfego que requer QoS será encaminhado pela tabela 0 para a tabela ou grupo de tabelas de QoS (ver Fig. 1). Esta tabela é composta por dois tipos de regras de coincidência para os pacotes:

- Regras gerais: permitem encontrar coincidências gerais com qualquer tipo de tráfego considerado de QoS (portas de VoIP, portas de vídeo, precedência IP, EXP de MPLS, etc.). Estas regras permitem encaminhar os primeiros pacotes de cada fluxo QoS e são utilizadas durante o período de tempo no qual ainda não exista uma regra específica para o fluxo individual.
- Regras específicas: são regras para cada fluxo QoS individual. Inicialmente, a tabela QoS que implementa esta lógica não dispõe de regras específicas. A lógica de preenchimento destas regras para cada fluxo QoS é uma das principais contribuições deste artigo.

Inicialmente, a tabela QoS só possui regras gerais de coincidência e cada novo fluxo QoS será processado por uma destas regras. O pacote será encaminhado por duas linhas diferentes de processamento simultaneamente (ver Fig. 1), descritas a seguir:

1- Primeiramente, a regra geral na tabela QoS envia uma requisição de novo fluxo ao controlador contendo uma cópia total ou parcial do pacote. Para isso, o cliente OpenFlow disponibiliza a **instrução** de execução imediata do tipo ***Apply-Actions*** e dentro desta uma **ação** do tipo ***Output*** à porta reservada ***Controller*** [2]. Com esta informação, o controlador calculará o caminho ótimo que permita satisfazer os requerimentos QoS do fluxo. Para isso, podem ser utilizados algoritmos de Dijkstra com restrições similares aos utilizados por ISIS-TE ou OSPF-TE, ou protocolos de encaminhamento novos específicos para cada tipo de qualidade de serviço, como, por exemplo, tráfego em tempo real sem perdas ou tráfego em tempo real com possibilidade de perda [3]. Finalmente, o controlador configurará uma regra específica para este novo fluxo na tabela de QoS de cada um dos equipamentos intervenientes no caminho ótimo obtido. Este novo fluxo é encaminhado às filas de atendimento preferencial que o OpenFlow disponibiliza, com marcação opcional dos campos EXP de MPLS e/ou DSCP (*Differentiated services code point*) do cabeçalho IP no primeiro dos clientes OpenFlow do caminho, o que facilita a identificação como fluxo QoS para os equipamentos posteriores. É importante mencionar que todos os fluxos de QoS específicos são criados com um *timeout* apropriado, para que sejam apagados automaticamente depois de um período de inatividade. Sempre que isso acontecer, o cliente OpenFlow comunicará ao controlador o fluxo apagado, para manter a coerência com a informação que o controlador possui.

2- Simultaneamente ao ponto anterior, e depois de enviada a consulta do novo fluxo ao controlador, o pacote original continua o processamento estabelecido pela regra geral de QoS. Esta regra tem definida uma **instrução** do tipo ***Goto-Table*** [2], que indica que o pacote tem que continuar seu processamento pela tabela de encaminhamento interno (ver Fig. 1). Desta forma, o pacote será processado imediatamente como um fluxo sem qualidade de serviço e não terá que aguardar a resposta do controlador.

É importante destacar que com esta lógica de processamento, o controlador não adiciona tempos de espera para os primeiros pacotes de cada fluxo. Esta afirmação é baseada no fato de que os pacotes do mesmo fluxo sempre são encaminhados inicialmente pelo encaminhamento interno, e são paralelamente processados pelo controlador. Posteriormente, quando a tabela de QoS tiver o fluxo específico preenchido, os pacotes subsequentes do fluxo coincidirão com esta nova regra e encaminharão o fluxo pela via ótima para este tipo de tráfego. Finalmente, nestas escolhas de encaminhamento não é permitido o balanceamento de carga, obtendo-se, assim, um caminho único similar aos VCs (*virtual circuits*) de outras tecnologias, com redução do *jitter* para o tráfego QoS da rede.

Adicionalmente, destaca-se que esta lógica permite reduzir sensivelmente a excessiva dependência do controlador que as implementações com OpenFlow padrão possuem. Quando o controlador cair ou apresentar tempos de resposta altos, os novos fluxos QoS sempre podem continuar sendo encaminhados pela lógica de gerenciamento interna, aspecto de robustez indispensável para um ISP (característica importante da arquitetura proposta).

E. Detalhe da lógica QoS proposta, controle no cliente OpenFlow e modificação do protocolo.

Na arquitetura proposta, como o pacote sem regra de coincidência específica na tabela QoS tem que ser enviado por duas linhas de processamento, é importante evitar que, quando a resposta do controlador chegar, ela não provoque o encaminhamento do pacote que gerou a consulta, pois ele já foi encaminhado pela tabela de encaminhamento interno. É necessário adicionar uma opção no OpenFlow que permita enviar uma consulta ao controlador, e quando a resposta chegar adicionar o novo fluxo à tabela QoS. Contudo, o pacote que

gerou a consulta terá que ser apagado do cliente OpenFlow sem que nenhuma ação seja efetuada.

Além disso, para qualquer implementação de OpenFlow é necessário haver mecanismos nos clientes que evitem que pacotes sucessivos do mesmo fluxo gerem as mesmas consultas ao controlador enquanto se aguarda à resposta do controlador, impedindo, assim, sobrecargas desnecessárias. Para que isso seja possível, o cliente OpenFlow tem que manter os cabeçalhos de todos os pacotes que geram consultas ao controlador e comparar os novos cabeçalho dos pacotes que chegarem ao cliente com os cabeçalhos dos pacotes que aguardam uma resposta do controlador.

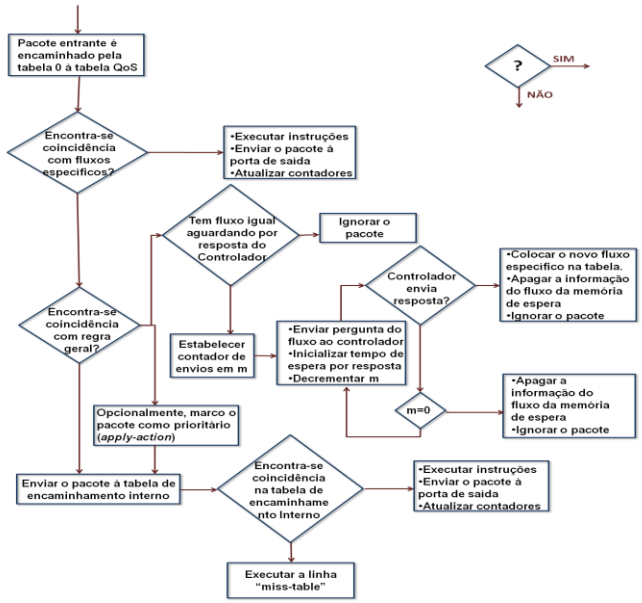

Fig. 2. Arquitetura de controle proposta.

A Fig. 2 contém a arquitetura da lógica de gerenciamento QoS completa que resolve os pontos previamente mencionados. Quando o pacote de QoS chega, a Tabela 0 o encaminha à Tabela QoS. Se existir uma regra específica pré-configurada para este pacote, a tabela QoS o encaminhará utilizando esta regra. Se não existir fluxo específico, a regra geral na tabela QoS encaminhará a consulta do novo fluxo ao controlador para que ele estabeleça um fluxo específico, e ao mesmo tempo enviará o pacote pela tabela de encaminhamento interno (ver Fig. 2). Quando a resposta do controlador chegar, os sucessivos pacotes terão uma regra para o fluxo específico com o caminho ótimo e com as considerações de QoS correspondentes ao fluxo particular. Porém, enquanto isso não acontecer, os pacotes do mesmo fluxo são encaminhados pelo encaminhamento geral sem adição de tempos de espera. É importante notar que, na lógica proposta, os sucessivos pacotes do mesmo fluxo não são enviados ao controlador. Somente o primeiro gera a consulta (como é mostrado na Fig. 2), que é reiterada em função do tempo de espera e o número de repetições configuradas, parâmetros para os quais deve ser permitida a sua adição nas futuras versões de OpenFlow. Além disso, deve ser permitido adicionar a lógica de controle proposta nos clientes OpenFlow IP (roteadores) e tem que ser permitido o apagado da informação do pacote depois de se adicionar o novo fluxo.

III. ARQUITETURA COM EQUIPAMENTOS HÍBRIDOS

Nesta seção, consideramos uma arquitetura alternativa utilizando equipamentos híbridos, os quais mantêm as capacidades do plano de controle distribuído e ao mesmo tempo incorporam OpenFlow. Para isso, pode-se definir nas regras de coincidência do OpenFlow a **instrução *Write-Actions***, a qual permite definir no ***action-set*** a **ação *output*** à porta reservada ***Normal*** [2], o que permite ao OpenFlow encaminhar o pacote ao plano de controle distribuído clássico. A Fig. 1 mostra como a tabela 0 encaminha o pacote às diferentes lógicas de gerenciamento, mas cada uma delas poderia estar implementada por OpenFlow ou por o plano de controle distribuído clássico.

Combinando equipamentos híbridos com a proposta da seção anterior, é possível desenvolver uma arquitetura intermediária, composta de equipamentos híbridos, que programem mediante OpenFlow a Tabela 0 e a lógica de gerenciamento de QoS, mas mantenha as lógicas restantes com o plano de controle distribuído clássico. Neste caso, o funcionamento é descrito a seguir. Se chegar um pacote que requer QoS à Tabela 0, ele é encaminhado à tabela QoS, conforme indicado na Seção II. No caso de não haver uma regra específica, é encaminhado ao controlador e em paralelo ao encaminhamento interno, o qual neste caso é o controle distribuído clássico. Enquanto não existir a regra especifica na tabela QoS, os sucessivos pacotes do fluxo serão encaminhados pelo controle distribuído clássico; quando a regra específica estiver disponível, o fluxo será encaminhado por esta regra de QoS particular.

Esta arquitetura é considerada na comparação apresentada na Seção V, e poderia ter pouca resistência a ser utilizada em um ISP. Esta arquitetura incrementa as possibilidades de engenharia de tráfego e ao mesmo tempo mantém a robustez, pois, no caso de queda do controlador, o serviço pode continuar indefinidamente com o controle distribuído clássico.

IV. CÁLCULO DE ATRASO COM OPENFLOW PADRÃO

Para analisar a contribuição da arquitetura proposta, apresentamos um exemplo de uma rede de um ISP com suas interconexões, seus roteadores (clientes OpenFlow) e o controlador (ver Fig. 3), com o objetivo de quantificar o atraso que a utilização de OpenFlow padrão adiciona.

A arquitetura para a modelagem tem quatro POPs (*Points of Presence*) interconectados pelos roteadores de núcleo da rede. Um cliente no POP 1 é escolhido como o gerador da consulta OpenFlow de novo fluxo ao controlador (ver Fig. 3).

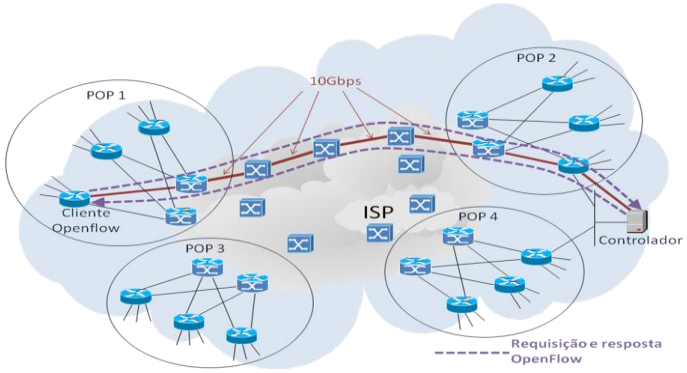

Fig. 3. Arquitetura escolhida para a modelagem.

Para a modelagem foram considerados modelos de enfileiramento M/M/1 com utilização inicial de 50%, cumprimento máximo da rede de 1000 Km, pacotes OpenFlow de tamanho meio de 200 Bytes, pacotes de tráfego na rede de tamanho médio de 500 Bytes e um tempo de serviço no

controlador de 0.24 ms de acordo com [10]. Com estes dados foi possível calcular o atraso transcorrido entre o envio de uma consulta do cliente OpenFlow ao controlador solicitando encaminhamento a um novo fluxo, e a adição do novo fluxo na tabela do cliente OpenFlow mais distante. A formulação analítica do cálculo do atraso pode ser encontrada em [11]. Substituindo-se os valores numéricos na formulação analítica, obteve-se um atraso total de 10,9 ms.

Foi analisado o efeito de variações na carga dos roteadores com enlaces de 10Gbps (ρRou10G), com enlaces de 1Gbps (ρRou1G) e do controlador (ρControlador) no atraso total. Para os três casos, variamos as cargas de um valor inicial de 50% a 100%, obtendo o resultado mostrado na Fig. 4.

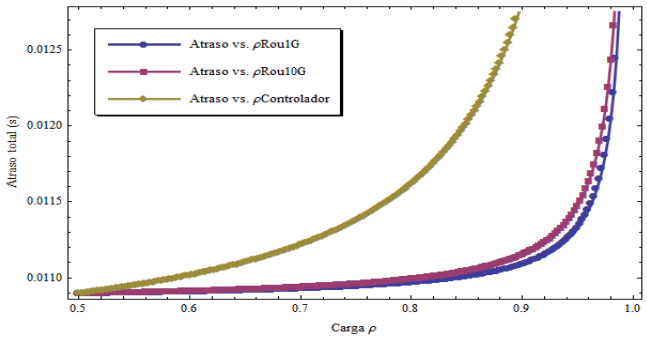

Fig. 4. Atraso total (s) VS. carga.

No gráfico anterior, pode-se observar um comportamento exponencial do atraso total com a variação dos parâmetros de carga da rede. Isso indica que no caso de se utilizar OpenFlow padrão, não só são adicionados tempos no estabelecimento de novos fluxos, como também estes tempos variam, tornando-se importantes nos casos de sobrecargas (ρ perto de 1), afetando, assim, a robustez e desempenho da rede.

Adicionalmente, segundo o resultado obtido em [12] na caracterização do tráfego, a probabilidade de se encontrar um pacote pertencente a um novo fluxo em um *cliente OpenFlow* é de 4%, o que implica que, para o caso de OpenFlow padrão, 4% do tráfego total é encaminhado ao controlador (muito excessivo para os ISPs). Na arquitetura proposta, o tráfego sem QoS não é encaminhado ao controlador e só o primeiro pacote de cada fluxo QoS gera consulta.

V. COMPARATIVO DAS ARQUITETURAS

TABELA I. COMPARATIVO DAS ARQUITETURAS

	Rede atual	OpenFlow padrão	Rede Hibrida proposta	Rede OpenFlow proposta
Resposta ao encaminhamento	Imediata	Intermedia	Imediata	Imediata
Possibilidades de engenharia de tráfego	Boa	Muito Boa	Muito Boa	Muito Boa
Jitter para tráfego QoS	Meio	Baixo	Baixo	Baixo
Comportamento no caso de quedas do controlador	N/A	Ruim	Muito Bom	Bom
Comportamento ante sobrecargas no controlador	N/A	Ruim	Ótimo	Muito Bom
Complexidade nos protocolos de encaminhamento	Muito alta	Intermédia	Muito alta	Intermédia
Carga por processamento nos equipamentos de rede	Alta	Muito Baixa	Alta	Muito Baixa

A arquitetura proposta gera mudanças importantes com respeito às redes OpenFlow padrão, e ao mesmo tempo mantém as potencialidades adicionais que OpenFlow proporciona. Com a análise realizada nas seções anteriores, foi construída a Tabela 1, que mostra uma comparativa entre as redes atuais, as redes implementadas com OpenFlow padrão, redes com a arquitetura proposta e equipamentos híbridos, e redes com a arquitetura proposta e equipamentos puramente OpenFlow.

VI. CONCLUSÕES E TRABALHOS FUTUROS

A arquitetura apresentada permite melhorar aspectos críticos da arquitetura OpenFlow padrão, tais como a grande dependência de toda a rede nos controladores, os tempos adicionados pelo controlador em cada estabelecimento de novo fluxo, e a excessiva informação que deve ser processada pelos controladores. Ao mesmo tempo, permite solucionar problemas das redes atuais, tais como a impossibilidade de construir caminhos individuais massivamente para fluxos QoS, limitações na engenharia de tráfego, complexos algoritmos de distribuição da informação, *jitter* significativos, e impossibilidade de unificação do plano de controle.

Como desvantagem em relação às redes atuais, a arquitetura proposta requer uso de controladores, adição de uma lógica de controle nos clientes OpenFlow (seção II-E), uma adição no protocolo OpenFlow e o desenvolvimento de um aplicativo de gerenciamento para a gestão do tráfego.

Como trabalhos futuros, pretende-se desenvolver as lógicas de gerenciamento não tratadas neste artigo: VPN, Segurança e encaminhamento externo.

REFERÊNCIAS

[1] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). *OpenFlow: Enabling Innovation in Campus Networks*. ACM SIGCOMM, 38(2):69-74.

[2] Open Networking Foundation. (2012). *OpenFlow Switch Specification Version.1.3.0*. https://www.opennetworking.org/images/stories/ downloads/ specification/openflow-spec-v1.3.0.pdf

[3] Egilmez, H., Gorkemli, B., Tekalp A. and Civanlar, S. (2011). *Scalable Video Streaming Over OpenFlow Networks: An Optimization Framework for QoS Routing. In Image Processing (ICIP)*. 2011 18th IEEE International Conference, p.2241-2244.

[4] Das, S. (2012) PAC.C: Tese de Doutorado, Universidade de Stanford. http://www.openflow.org/wk/index.php/PACC_Thesis.

[5] Shirazipour, M., John, W., Kempf, J., Green, H. and Tatipamula, M. (2012). *Realizing Packet-Optical Integration with SDN and OpenFlow 1.1 Extensions*. Communications (ICC), IEEE International Conference, p.6633-6637.

[6] Das, S., Parulkar, G., McKeown, N., Singh, P., Getachew, D and Ong, L. (2010). *Packet and Circuit Network Convergence with OpenFlow. Optical Fiber Communication (OFC)*, Collocated National Fiber Optic Engineers Conference, p.1-3

[7] Bifulco, R., Brunner, M., Canonico, R., Hasselmeyer, P and Mir Faisal. (2012). *Scalability of a mobile cloud management system*. ACM New York, ISBN: 978-1-4503-1519-7.

[8] Open Networking Foundation (Members). https://www.opennetworking.org/membership/members.

[9] HP Networking-OpenFlow. http://h17007.www1.hp.com/us/en/ networking/solutions/technology/openflow/index.aspx

[10] Jarschel, M., Oechsner, S., Schlosser, D., Pries, R., Goll, S. and Tran-Gia. P. *Modeling and Performance Evaluation of an OpenFlow Architecture*. 23rd International Teletraffic Congress, p 1-7.

[11] López-Rodríguez, F., *Technical Report n. 1/2013*, UnB Brasilia, 2013, https://www.dropbox.com/s/yv5qy2fxixute1e/Technical%20report.pdf

[12] Wamser, F., Pries, R., Staehle, D., Heck, K. and Tran-Gia, P. (2011). *Traffic Characterization of a residential Wireless Internet Access*. Telecommunication Systems, V.48, Issue 1-2, pp 5-17.