

Balanceamento de Tráfego entre Servidores de Vídeo MPEG-DASH em Redes Definidas por Software

Edenilson Jônatas dos Passos¹, Adriano Fiorese¹

¹Universidade do Estado de Santa Catarina (UDESC)
Campus Universitário Prof. Avelino Marcante - Rua Paulo Malschitzki,
200 - Zona Industrial Norte, Joinville - SC, 89219-710

edenilson.passos@yahoo.com, adriano.fiorese@udesc.br

Abstract. *With popularization of video streaming service in recent years, new video distribution technologies have been created. Currently, one of the most promising ones is the Moving Picture Expert Group Dynamic Adaptive Streaming over HTTP or MPEG-DASH. However, with the limitation of the TCP/IP network structure, the end user quality of experience (QoE) may be affected. One issue that can affect user QoE is the workload of content servers. Thus, the imbalancing of server's workload comprising user's attendance can lead to a content server provider non optimized choice. This work presents a load-balancing solution between MPEG-DASH video servers based on Software Defined Networks, using as a balancing workload metric the throughput of the content server.*

Resumo. *Com a popularização do serviço de streaming de vídeo nos últimos anos, novas tecnologias de distribuição de vídeo foram criadas. Uma das mais promissoras atualmente é a chamada de Moving Picture Expert Group Dynamic Adaptive Streaming over HTTP, ou MPEG-DASH. Contudo, com a limitação da estrutura de rede TCP/IP, a Qualidade de Experiência (QoE) do usuário final pode ser afetada. Um dos fatores que podem afetar a QoE do usuário é a carga de trabalho dos servidores de conteúdo. Assim, o desbalanceamento da carga de atendimento dos usuários entre os servidores pode gerar uma escolha não otimizada do provedor de conteúdo. Este trabalho apresenta uma solução de balanceamento de carga entre servidores de vídeo MPEG-DASH baseada em Redes Definidas por Software, utilizando como métrica de balanceamento a taxa de transferência do servidor de conteúdo.*

1. Introdução

A utilização da Internet na transmissão de vídeo possui papel relevante nos modelos de negócio dos provedores de conteúdo atuais. Isso é benéfico para os usuários pois possibilita um grande controle sobre quando e onde o conteúdo pode ser assistido. Uma pesquisa realizada pela empresa *The Nielsen Company* revelou que dentre um total de 30000 participantes de 61 países, 65% afirmam consumir algum tipo de programação de Vídeo Sob Demanda [Nielsen 2016]. Essa mesma pesquisa ainda revela a divisão total da atuação da Internet na distribuição de conteúdo de vídeo baseada em continentes. Na Ásia e Oceania, chega a 32%, na América, 35%, na África 21% e na América Latina 21%. Além disso, segundo [Cisco 2018], até o ano de 2022, é esperado que o tráfego de vídeo na Internet chegue a 82% do total.

Com a popularização de serviços de *streaming*, e conseqüentemente a sobrecarga nos enlaces bem como das restrições impostas a esse tipo de tráfego pelos provedores de serviço de acesso à Internet, empresas começaram a desenvolver formatos e conteúdo de vídeo adaptativo utilizando o protocolo HTTP como método de entrega. Contudo, com plataformas diferentes, existem configurações diferentes, o que gera a necessidade de padronização para que a interoperabilidade entre servidores e clientes de conteúdo de vídeo de diferentes fornecedores se torne possível [Sodagar 2011].

Assim, surgiu o padrão ISO/IEC 23009-1 ou Moving Picture Expert Group Dynamic Adaptive Streaming over HTTP (MPEG-DASH). O MPEG-DASH, foi projetado de maneira a balancear a qualidade de experiência do usuário (QoE) através de uma experiência fluída e com a melhor qualidade de vídeo possível [Duanmu et al. 2017]. Desse modo, o reprodutor de vídeo (*player*) MPEG-DASH no computador cliente se adapta de acordo com diversos fatores. Entre eles estão a resolução do vídeo, o *bitrate* de vídeo e de áudio, além da condição do *buffer* do *player*. Esses fatores variam de acordo com a largura de banda disponível, e a disposição geográfica dos servidores de conteúdo na *Content Distribution Network* (CDN) [Stockhammer 2011].

A CDN é responsável pelo armazenamento do conteúdo buscado pelos clientes e também pelo roteamento das requisições de conteúdo para um servidor mais próximo do cliente. Além disso, uma CDN permite o atendimento da demanda crescente dos usuários por conteúdo assumindo as responsabilidades pela manutenção da qualidade do serviço de distribuição dos conteúdos e sua escalabilidade. Entre essas responsabilidades encontra-se o balanceamento de carga [Bourke 2001], [Cardellini et al. 1999]. De acordo com [Doshi et al. 2015], o caso de atraso que conseqüentemente mais afeta a QoE do usuário é o congestionamento do servidor, pois é percebido pelo usuário como interrupção e ruído na imagem e áudio do conteúdo.

Uma abordagem contemporânea para o balanceamento de carga em CDNs para distribuição de conteúdo de vídeo é baseada no atendimento das requisições dos diversos clientes por vários servidores distribuídos na CDN. Para tal, a atribuição de qual servidor atenderá qual cliente é feita através da implantação personalizada de servidores *Domain Name Server* (DNS) por provedor de CDN. As entradas no DNS então, são adaptadas a todo momento de modo a fornecer o balanceamento de carga entre os servidores. Contudo, fluxos de grande volume e de longa duração, como por exemplo o tráfego de vídeo sob demanda (VoD), dificilmente são gerenciáveis através dessa abordagem DNS. A introdução de SDN é promissora para resolver este problema, pois instanciações de SDN como o *OpenFlow* permitem o redirecionamento transparente de fluxos na rede [Wichtlhuber et al. 2015], sem a necessidade de manipulação de serviços de infraestrutura.

Portanto, o problema do balanceamento de carga em CDNs, ou seja, entre os servidores de conteúdo, nesse caso particular de servidores de vídeo MPEG-DASH, sofre de dificuldades de gerenciamento com as abordagens tradicionais atuais. A contribuição deste artigo visa disponibilizar uma abordagem arquitetural baseada em SDN para a solução desse problema.

Sendo assim, o presente artigo trata da proposição de uma arquitetura para o balanceamento de carga em rede de distribuição de vídeo com o auxílio de SDN para realizar

a manipulação de fluxos necessários por meio do protocolo Openflow versão 1.3 e do controlador Ryu. O balanceamento executado tem como objetivo o alívio da carga de trabalho dos servidores origem do conteúdo e utiliza a vazão de dados dos servidores como métrica de balanceamento.

Este trabalho está organizado da seguinte maneira. Na Seção 2, os trabalhos relacionados são discutidos. Em seguida, na Seção 3, a arquitetura da solução proposta é detalhada. A Seção 4, apresenta a avaliação da proposta e a Seção 5 apresenta as considerações finais deste trabalho.

2. Trabalhos Relacionados

Em [Jiang et al. 2015] é apresentado um algoritmo adaptado de Dijkstra estendido para balanceamento de carga em uma CDN utilizando a tecnologia SDN. Segundo a justificativa utilizada, algoritmos que se utilizam de abordagens de seleção aleatória ou ainda a abordagem *Round Robin*, ao selecionar uma rota para o servidor de conteúdo, podem selecionar a rota mais longa (em relação a saltos) e / ou com congestionamentos, diminuindo drasticamente o desempenho do serviço. Com o algoritmo proposto, a intenção é selecionar o caminho mais curto e com menor congestionamento. Assim, são realizados diversos testes referentes a latência do servidor, tempo de resposta e *throughput*. Nesse trabalho, os resultados do algoritmo proposto são comparados com aqueles das abordagens de escolha aleatória e Round Robin.

Semelhante a [Jiang et al. 2015], a ideia central de [Kaysudu et al. 2016], é propor um algoritmo, em uma rede SDN, que encontra uma rota mais apropriada para se alcançar o melhor servidor de distribuição de conteúdo. De acordo com os estudos apresentados, o desempenho dos sistemas de transmissão de vídeo em execução na rede SDN pode ser aprimorado configurando as rotas pelas quais os pacotes de vídeo são transmitidos para os clientes com base em medições de largura de banda em tempo real.

Um trabalho que utiliza da área de inteligência artificial para balanceamento de carga foi abordado em [Chen-xiao and Ya-bin 2016]. Nele foi utilizada a abordagem SDN para fazer o redirecionamento dos fluxos de vídeo conforme o resultado do algoritmo de balanceamento de carga proposto. Esse algoritmo é baseado em uma rede neural que utiliza como principais métricas a utilização da largura de banda, a taxa de pacotes perdidos, a latência e o número de saltos. Contudo, não tinha como foco redes de distribuição de conteúdo, sendo o principal objetivo do trabalho economizar a largura de banda através da seleção do melhor caminho possível sem congestionamento.

Um exemplo de trabalho que utiliza balanceamento de carga baseado em métricas para escolha do servidor mais adequado é [Carter and Crovella 1997]. Nele, os autores utilizaram ferramentas específicas para medir a latência e a largura de banda disponível do servidor para o cliente. Com essas métricas o algoritmo tomava as devidas decisões a respeito de qual servidor era o mais adequado para transmitir os dados. Entretanto, não foi utilizada a tecnologia VoD e nem do paradigma SDN.

Sendo assim, observando os mais recentes trabalhos que tratam do balanceamento de carga em sistemas de distribuição de conteúdo (especialmente vídeo), é possível notar possibilidade de melhorias. Especialmente, levando em conta a utilização do paradigma SDN para o estabelecimento do balanceamento de carga entre servidores de conteúdo,

executado diretamente em nível da infraestrutura de rede (camadas 2,3 e 4). Além disso, a utilização da métrica de vazão dos servidores envolvidos, bem como o monitoramento e utilização sob demanda desse valor torna o balanceamento de carga altamente dinâmico e escalável, seja na adição de clientes quanto de servidores.

3. Solução Proposta

Esta seção apresenta a abordagem proposta para o balanceamento de carga entre servidores de conteúdo MPEG-DASH. Tal abordagem contempla uma arquitetura de software implementada junto ao controlador SDN, envolvendo passos e ações específicas a serem executadas para o correto redirecionamento das requisições de vídeo do usuário. Estes redirecionamentos, em última análise, executam o balanceamento de carga de acordo com a métrica utilizada.

3.1. Arquitetura Proposta

Na arquitetura proposta, o controlador Ryu é o principal atuador do sistema. Ele é quem executa as principais funções do sistema, como a obtenção das métricas de taxa de transferência dos servidores e toma a decisão de qual é o mais adequado para então poder gerar e instalar as regras de redirecionamento de tráfego no *switch* OpenFlow. A abordagem adotada para a atuação do controlador, na arquitetura proposta, é reativa. Assim, uma vez que o cliente tente acessar o sítio web contendo a página do conteúdo de vídeo, esse pacote de dados da requisição é encaminhado ao controlador, caso no *switch* OpenFlow não haja regra correspondente.

A Figura 1 apresenta a arquitetura da solução proposta, destacando as atividades executadas e suas interações como aplicação pertencente ao controlador. Além disso, a Figura 1 indica a sequência numérica de execução dessas atividades como uma sequência de eventos.

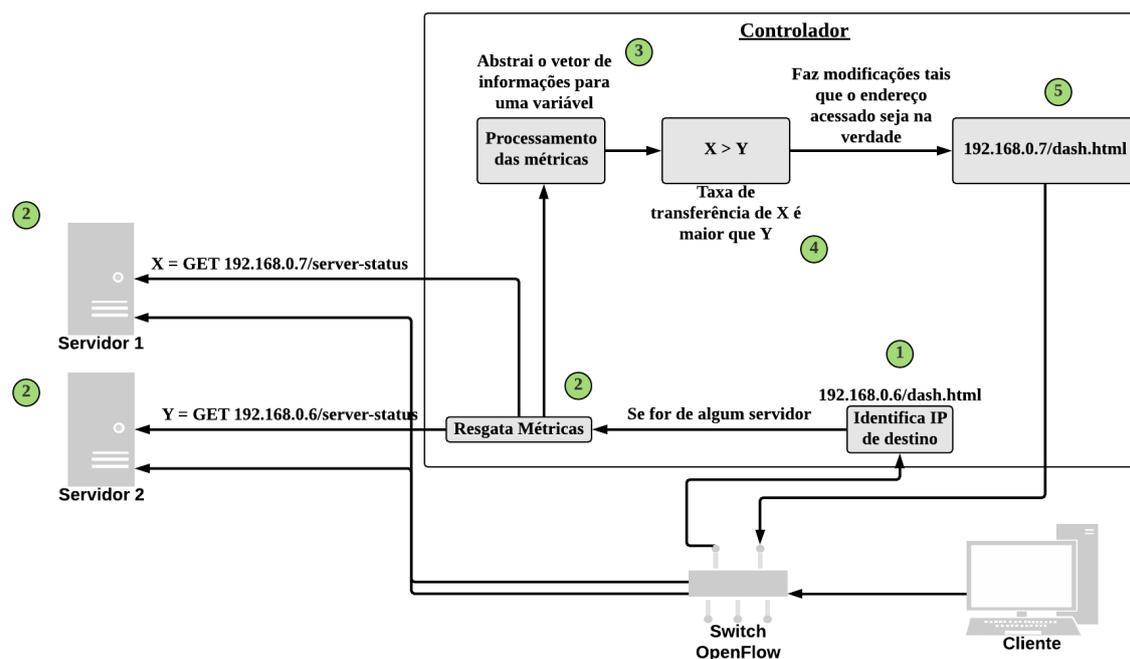


Figura 1. Arquitetura de balanceamento de carga proposta

Assim, de acordo com a arquitetura proposta, o pacote de dados representando a requisição do cliente, ao chegar ao *switch* OpenFlow, com o seu direcionamento ao controlador, põe em marcha a identificação do cliente requisitante e o conteúdo a ser entregue. Para tal, por meio do estabelecimento da conexão TCP e execução da requisição HTTP do conteúdo, uma função identifica o endereço IP de origem (1). Com o endereço de origem, o próximo passo é verificar o endereço de destino (1). Caso o endereço destino seja de um dos servidores de conteúdo conhecidos e disponíveis, o próximo passo compreendendo a obtenção das métricas desses servidores, é iniciado (2). A obtenção dos valores da taxa de transferência dos servidores é realizada pela aplicação de balanceamento desenvolvida junto ao controlador, obtida uma única vez durante a execução do vídeo, logo após o processo de obtenção do IP de origem. Ou seja, a cada nova conexão TCP endereçada a um dos servidores, os valores da métrica utilizada são resgatados. Portanto, o balanceamento de carga ocorre apenas em função do estabelecimento da conexão (e consequente sessão HTTP).

Após a obtenção dos valores da métrica utilizada, a aplicação de balanceamento desenvolvido junto ao controlador avalia qual servidor possui a maior taxa de transferência (3,4), presumindo que aquele que possui maior taxa de transferência está apto a receber mais conexões e propenso a menor latência. O próximo e último passo é a geração e instalação das regras no *switch* OpenFlow referentes aos endereços IP e MAC e à porta de destino do tráfego de conteúdo com base no melhor servidor escolhido (5).

A Figura 2 ilustra a sequência de eventos do sistema proposto. Primeiramente, o cliente faz a requisição do conteúdo. Essa requisição passa então pelo *switch* OpenFlow que a reencaminha ao controlador para tratamento e execução das medidas cabíveis. Assim, o controlador solicita aos servidores disponíveis a métrica de taxa de transferência. Com a métrica de todos os servidores, é feita a tomada de decisão de atendimento pelo que possui a taxa mais alta. Após isso, são geradas e executada a instalação das regras de fluxo no *switch*. Assim, com as regras instaladas, é de fato realizada a requisição ao servidor escolhido e este por sua vez retorna o conteúdo ao cliente utilizando-se das regras recentemente instaladas no *switch*.

3.2. Implementação

Para a instanciação da arquitetura proposta, além da implementação da aplicação de balanceamento de carga junto ao controlador Ryu, na falta de uma CDN para testes, uma infraestrutura que simule o acesso de clientes a vários servidores de conteúdo, em condições manipuláveis da rede, é necessária. Nesse sentido, o cenário de implementação adotado tem como elementos n servidores de conteúdo, um controlador, um *switch* OpenFlow e n clientes conforme Figura 3.

Assim, para o caso particular deste trabalho, devido a limitação de hardware, porém sem perda da objetividade, apenas dois servidores de conteúdo MPEG-DASH foram criados, ambos idênticos. Isto é, ambos possuem o mesmo conteúdo a ser disponibilizado. Com relação aos clientes, mesmo estando na rede Mininet, e portanto virtualizados, foi disponibilizado a eles acesso externo e portanto à Internet para que acessassem os servidores de conteúdo.

O cenário proposto baseado em rede foi elaborado de modo que a rede criada pelo emulador Mininet consiga acessar a rede LAN física (representando a rede WAN em

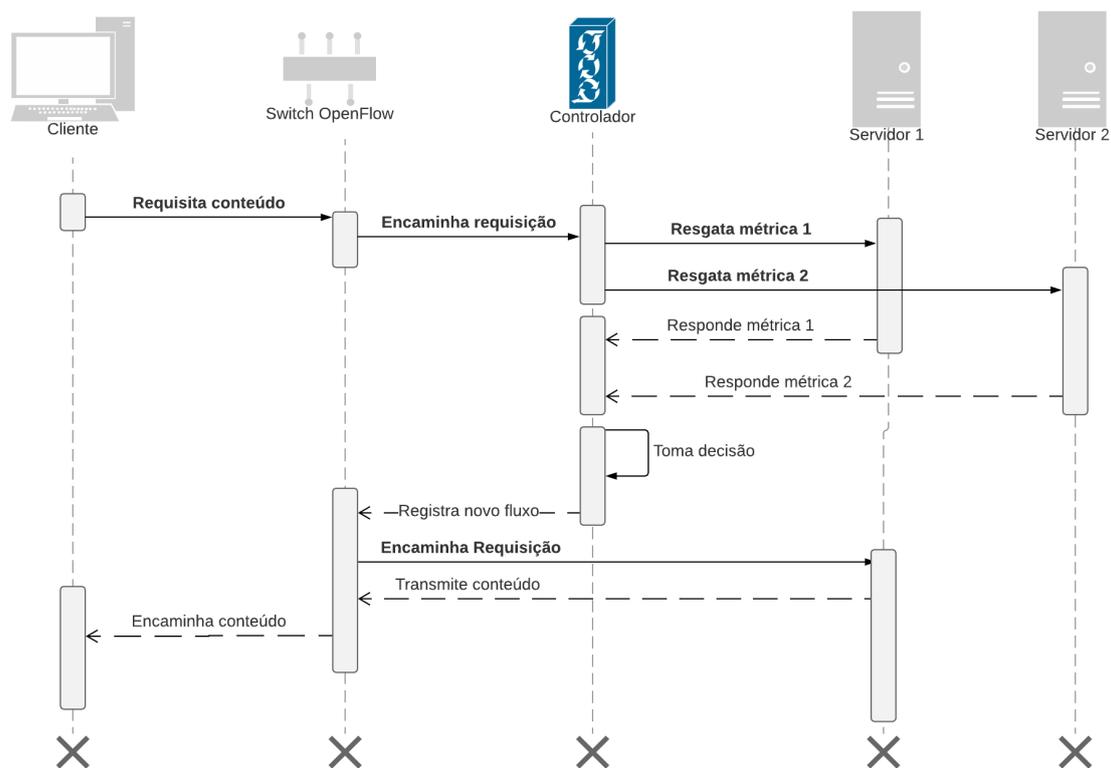


Figura 2. Sequência de eventos da solicitação do conteúdo vídeo

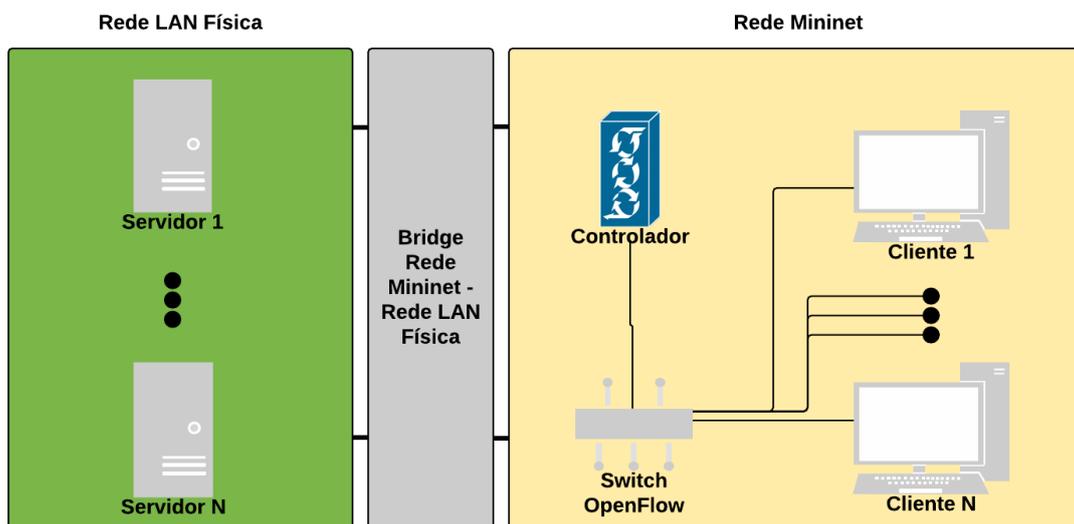


Figura 3. Cenário de implementação baseado em redes distintas

cenário real), na qual os servidores estão hospedados. Essa conexão ocorre por meio de uma ponte (*bridge*) que conecta essas duas interfaces através do *switch* OVS.

Mais especificamente, é realizada a configuração dos clientes (consumidores de conteúdo de vídeo MPEG-DASH) virtualizados, existentes na rede Mininet, de forma

que os mesmos recebam endereços IP da rede LAN física real que tem acesso a Internet, conforme Figura 4. Assim, no passo 1, ao ser inicializado, o cliente recebe endereço IP aleatório associado a rede interna Mininet (*h1-eth0*). No passo 2, é executado o comando para desassociação desse IP (*ifconfig h1-eth0 0*). Logo, esse cliente (*host*) não possui endereço IP. No passo 3, é adicionada a interface de rede do equipamento físico que está executando o programa Mininet ao *switch* OVS utilizado pela rede Mininet (*sudo ovs-vsctl add-port s1 enp0s3*). Desse modo, todos os dispositivos conectados a esse *switch* (*hosts* na rede Mininet) terão acesso a Internet quando feita a nova associação de IP no passo 4. Essa nova associação, se dá agora pelo serviço de DHCP (*Dynamic Host Configuration Protocol*) executando no dispositivo físico que executa o Mininet e a rede emulada por ele.

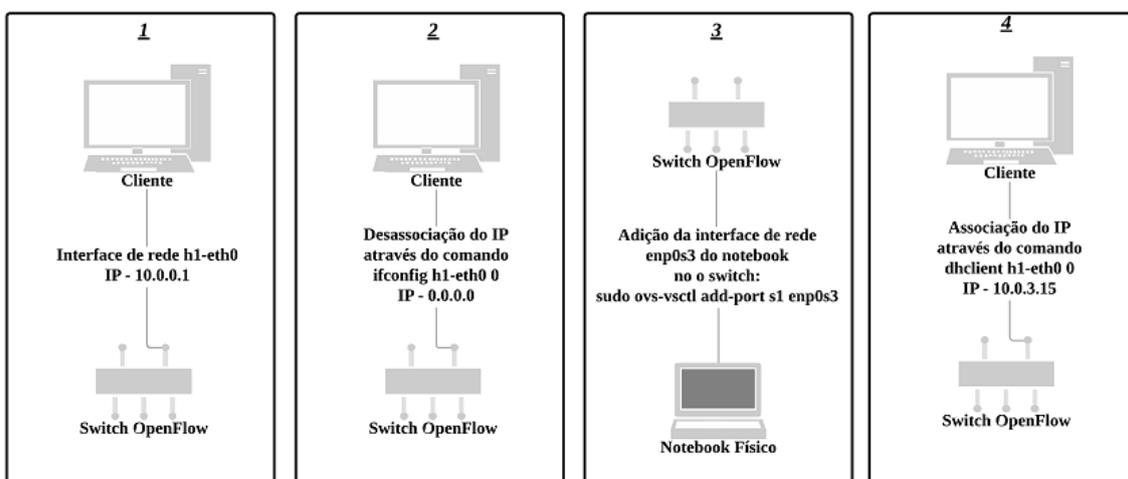


Figura 4. Passo a passo configuração IP dos *hosts*

Para simular um cenário mais realista em termos de consumo de vídeo MPEG-DASH, foi idealizado que todos os *hosts* (clientes) criados dentro da rede Mininet possam utilizar um navegador de Internet para executar a visualização do conteúdo recebido. Nesse caso, o navegador Google Chrome, com alterações para execução em modo administrador, foi utilizado pois permite a utilização facilitada de *player* que permite a visualização de conteúdo MPEG-DASH.

A instalação do *player* MPEG-DASH é um passo importante na implementação do trabalho. Isso se deve ao fato de que reprodutores de vídeo/áudio convencionais não tem suporte para reprodução de tal padrão.

O *player Dash.js* é, de fato, um arquivo em *javascript* que pode ser descarregado pelo sítio web de origem [dash.js 2012]. Sua instalação se dá pela inclusão desse pacote (arquivo) no cabeçalho (*header*) da página web que disponibiliza o reprodutor ao usuário.

Para o resgate dos valores da métrica taxa de transferência nos servidores de vídeo MPEG-DASH, o servidor Apache fornece uma ferramenta conhecida como *server-status* que ativa uma página HTML (*Hyper Text Markup Language*) com todas as possíveis informações e métricas de desempenho do servidor. Essa página pode ser acessada remotamente. Assim, o resgate, executado pela aplicação de balanceamento de tráfego desenvolvida junto ao controlador Ryu, dos valores da métrica taxa de transferência é realizado

por meio da requisição HTTP GET, dessa página, nos servidores designados. Portanto, quando retornada tal página web, é executado o tratamento dos dados recebidos para que o valor da taxa de transferência seja o único a ser comparado. Um ponto importante a ser destacado é que, para que o *status* do servidor possa ser acessado, nas configurações do servidor Apache (*apache.conf*), devem ser adicionados os endereços que têm acesso ao servidor, pois caso contrário a requisição será negada.

A elaboração do algoritmo balanceador de carga proposto foi realizada com base na aplicação de *switch* camada 2 da biblioteca disponível do controlador Ryu. A partir dessa aplicação, foram executadas as modificações necessárias de tal modo que quando uma requisição de conteúdo MPEG-DASH é recebida no *switch* OpenFlow, todos os pacotes TCP enviados para o IP do servidor de conteúdo destinatário que pretendem iniciar uma conexão TCP, não vão ser de fato entregues diretamente ao servidor, mas redirecionados para o controlador para processamento.

A aplicação de balanceamento de carga então requisita os dados de taxa de transferência (por meio da página HTML de *status*) dos servidores através de requisições HTTP GET. Após receber a resposta, e executar o tratamento para encontrar os valores da métrica taxa de transferência, ela poderá decidir qual servidor real é o mais adequado para entregar o conteúdo. Em seguida, a aplicação de balanceamento de carga modifica o plano de dados da rede instalando regras de fluxo cuja ação deve reescrever o endereço IP, MAC e a porta de destino de todos os pacotes futuros do cliente para o endereço IP, MAC e porta do servidor escolhido. O mesmo deve ser realizado no sentido contrário, isto é, do servidor para o cliente. Isso deve ocorrer para que o cliente não perceba que o endereço IP do servidor foi redirecionado.

Portanto, é esperado que o usuário não perceba qualquer tipo de mudança na reprodução do conteúdo. Caso o cliente (*player*) MPEG-DASH executando no navegador Google Chrome utilize, por exemplo, o endereço 10.0.0.1 para acessar o servidor de conteúdo, mas o controlador decidir que o servidor 10.0.0.2 é mais adequado, o redirecionamento será executado automaticamente e o usuário/cliente não saberá que essa troca foi feita.

4. Experimentos e Resultados

Todos os experimentos idealizados tem como principal objetivo observar o comportamento da reprodução do vídeo no computador do usuário final. Deste modo, todas as métricas de avaliação do balanceamento de carga proposto para as quais os experimentos foram elaborados, tem relação direta ou indireta com a qualidade de experiência do usuário final.

4.1. Ambiente de Testes

Todos os experimentos foram realizados em máquinas virtuais com o auxílio da ferramenta VirtualBox. O primeiro dispositivo utilizado para executar a máquina virtual é um computador de mesa. Este tem como papel hospedar a rede Mininet, contendo os *hosts*, o *switch* OpenFlow (OpenVSwitch), o controlador e o servidor de conteúdo que está separado do equipamento que contém o ambiente virtual Mininet. O sistema operacional utilizado foi o Ubuntu com 3 GB de RAM com processador de 4 núcleos a 3.5 GHz. O segundo dispositivo utilizado é um notebook. As especificações da máquina virtual são

semelhantes as anteriores. Porém, por dispor de um processador inferior executando a apenas 3.2 GHz, foi elencado somente como servidor de conteúdo.

O *player* MPEG-DASH utilizado é o *dash.js* por ser de código aberto. Logo, é possível realizar modificações conforme necessário. Particularmente, modificações (funções) foram implementadas para permitir boa parte da análise da qualidade de experiência do usuário, ou seja, permitir acompanhar o comportamento de certas estatísticas de transmissão do vídeo durante a reprodução do conteúdo. Entre as funções desenvolvidas estão:

- Tempo para aparição do primeiro quadro de vídeo no reproduutor. Como discutido anteriormente, se trata de uma métrica de QoE.
- A cada segundo é impresso o *bitrate* atual do vídeo e áudio, juntamente com a respectiva resolução.
- A cada 15 segundos é feito um teste de velocidade da largura de banda.
- Opcionalmente, no início do vídeo é possível ativar uma função que mostra todas as qualidades disponíveis do vídeo e áudio do conteúdo (resolução, *bitrate* de áudio e vídeo).

Para criar a rede no Mininet criam-se inicialmente os *hosts* (clientes) que executarão os navegadores e por conseguinte o *player* MPEG-DASH. Cria-se em seguida o *switch* virtual com o open vswitch e conectam-se cada um desses *hosts* às portas do *switch*. Após as devidas configurações, o cliente (navegador executando no *host* do Mininet) está pronto para acessar o conteúdo disponibilizado na página do servidor Apache.

Com relação aos conteúdos de vídeo/áudio disponibilizados, foram escolhidos dois *datasets* de vídeos de licença *Creative Commons* (CC) oferecidos para testes pelo YouTube. O primeiro *dataset* é chamado de **CAR CENC**. O vídeo possui 181 segundos de duração, e é disponibilizado em 6 qualidades (diferentes resoluções) de vídeo com variação de 256×144 à 1920×1080 pixels, e uma faixa de áudio. Nesse caso, tanto o vídeo quanto o áudio estão disponibilizados em segmentos (chunks) de 2 segundos.

O segundo *dataset* é chamado de **FEELINGS VP9**. Este *dataset* possui compressão de vídeo pelo método VP9 e por isso, seu tamanho é consideravelmente menor em relação ao anterior. Porém, apresenta como qualidade máxima somente alta definição, ou seja, não é disponibilizado *Full HD*. O vídeo possui 136 segundos de duração, e é disponibilizado em 6 qualidades de vídeo com variação de 426×240 à 1280×720 pixels.¹ e uma faixa de áudio.

4.2. Avaliação de QoE

Para além da abordagem de balanceamento de carga orientada a tráfego proposta, baseada na métrica taxa de transferência, outras duas abordagens foram desenvolvidas. Uma delas utiliza um algoritmo de escolha aleatória do servidor e a outra, o algoritmo *Round Robin* entre os servidores de conteúdo MPEG-DASH. O algoritmo de escolha aleatória foi desenvolvido de maneira simplista, como aplicação do controlador. Assim, a cada nova conexão tratada pela aplicação de balanceamento junto ao controlador, duas variáveis, cada uma representando um servidor, recebem um número aleatório de 0 a 1000. Aquela

¹Nos resultados, quando se falar em *dataset* carro, significa que trata-se do dataset **CAR CENC**. Já o **FEELINGS VP9** é lido como *paraquedas*

que apresentar o maior número aleatório, é o servidor escolhido. Já para a abordagem Round Robin, foi criada uma variável global que é incrementada em um a cada nova conexão tratada. Como os testes foram realizados contando apenas com dois servidores de conteúdo, quando o número for par o servidor 1 é escolhido. Consequentemente, o servidor 2 é escolhido caso o número seja ímpar. É importante ressaltar que para a realização dos experimentos com as abordagens sem a utilização de SDN, a taxa de transferência foi monitorada manualmente, para que quando houvesse a requisição de conexão, ela fosse feita para o servidor com a menor taxa de transferência. A intenção é demonstrar que existirão atrasos caso a taxa de transferência for baixa.

Os resultados a seguir apresentados foram obtidos da seguinte maneira. No total, foram realizados 10 testes para cada *dataset* de cada método/abordagem proposta sendo que o valor da média dessas execuções é usado como resposta. A cada teste, o controlador era encerrado assim como a rede Mininet e o *cache* do navegador era esvaziado.

As métricas para avaliação dos da abordagem proposta por meio dos experimentos realizados estão relacionadas com a mensuração da QoE do cliente, e são:

- A média do *bitrate* para cada segundo de execução de vídeo. Assim, quanto maior a taxa, melhor a qualidade do vídeo exibido. Para tal, foram criados dois gráficos, um para cada *dataset*, com a média geral de todos os métodos envolvidos.
- O número de vezes que a qualidade do vídeo mudou. Quanto menor esse número, melhor a qualidade de experiência do usuário final.
- O tempo de carregamento da página. Quanto menor, melhor.
- O tempo para a aparição do primeiro quadro do vídeo no *player*. Quanto menor, melhor.
- O número de congelamentos (*stalls*) no vídeo. Quanto menor melhor. O ideal é zero.
- A média de largura de banda utilizada pelo cliente.

Assim, foram executados os experimentos de requisição e reprodução dos vídeos dos *datasets* utilizados, que capturaram os valores para as métricas elencadas anteriormente. A Tabela 1 apresenta os resultados obtidos. Assim, é possível notar que para a métrica tempo de carregamento da página, a abordagem proposta é mais eficiente do que todas as outras abordagens quando o *dataset* é o **FEELINGS VP9** (paraquedas). Em relação ao outro *dataset*, ele é inferior somente a abordagem sem SDN. Uma das possíveis justificativas é que na abordagem sem SDN, não existe escolha de servidor baseado em métrica e processamento e portanto, tende a ser mais ágil.

A mesma situação ocorre com a terceira métrica de avaliação: o tempo de espera para o primeiro *frame* do vídeo ser visualizado. O método proposto é mais eficiente no *dataset* do paraquedas. Contudo, perde para o método sem SDN no *dataset* carro. A justificativa para tal, é idêntica a da métrica anterior.

Na quarta métrica, foi contabilizado o número de vezes em que a qualidade do vídeo foi alterada durante a reprodução no cliente. Antes de tudo, é válido ressaltar que é esperado que o vídeo tenha um número mínimo de troca de qualidades. Nesse caso, essa quantidade é 3. Isso se deve ao fato de que, quando o vídeo é solicitado, a transmissão é iniciada em qualidade inferior, geralmente a mínima no primeiro *frame* pois o *player* precisa se certificar que existe de fato largura de banda suficiente para aumentar a qualidade. Por isso, geralmente são notadas 3 trocas de qualidade nos segundos iniciais.

	Sem SDN		Randômico		Round Robin		Abordagem Proposta	
	Carro	Paraquedas	Carro	Paraquedas	Carro	Paraquedas	Carro	Paraquedas
Carregar pagina (ms)	574,30 $\sigma = 47,46$	542 $\sigma = 37,38$	819,90 $\sigma = 243,78$	819,10 $\sigma = 298,97$	987,10 $\sigma = 227,51$	1066,40 $\sigma = 49,84$	679,80 $\sigma = 354,21$	525,80 $\sigma = 450,55$
Primeiro frame (ms)	1078 $\sigma = 55,81$	1144,10 $\sigma = 118,53$	1371,80 $\sigma = 402,50$	1242,30 $\sigma = 312,76$	1486,20 $\sigma = 278,96$	1679,40 $\sigma = 402,50$	1141 $\sigma = 410,58$	1011,80 $\sigma = 582,86$
Trocas de qualidade	5,3 $\sigma = 2,86$	3 $\sigma = 0$	3,7 $\sigma = 2,14$	3 $\sigma = 0$	3,6 $\sigma = 1,01$	3 $\sigma = 0$	3,5 $\sigma = 1,36$	3 $\sigma = 0$
<i>Stalls</i>	0,8 $\sigma = 0,97$	0	1 $\sigma = 1,26$	0	0	0,2 $\sigma = 0,6$	0,6 $\sigma = 0,91$	0,4 $\sigma = 0,8$
Largura de banda (Mbps)	2,39 $\sigma = 0,26$	0,30 $\sigma = 0,09$	2,11 $\sigma = 1,36$	1,08 $\sigma = 1,29$	3,88 $\sigma = 0,24$	2,41 $\sigma = 0,10$	3,47 $\sigma = 0,59$	2,39 $\sigma = 0,15$

Tabela 1. Tabela de resultados finais

Como o segundo *dataset* é mais “leve”, todos os métodos implementados conseguiram o melhor resultado (3 mudanças de qualidade). No *dataset* carro, no entanto, sendo um vídeo de qualidade máxima de 1920×1080 pixels, sem compressão, portanto mais “pesado”, houve mais trocas de qualidade em todos os métodos. Isso ocorre pois esse vídeo em sua qualidade máxima exige muita largura de banda, e conforme o funcionamento do *player* MPEG-DASH, quando não há largura de banda suficiente, a qualidade dos segmentos de vídeo requisitados é diminuída. Em alguns casos, em especial no método sem SDN, há mais trocas de qualidade principalmente ao meio da reprodução, o que afeta negativamente a qualidade de experiência do usuário. Uma das explicações para tal, se deve ao fato de que os testes foram realizados com o limite de largura de banda imposto pelo programa *Traffic Monitor* e em alguns momentos houve instabilidade da largura de banda, forçando o *player* a diminuir a qualidade requisitada dos vídeos.

Em relação a quinta métrica, o número de *stalls*, é importante ressaltar que um *stall* geralmente ocorre quando não há largura de banda suficiente para consumir o vídeo num determinado espaço de tempo e o *player* não consegue diminuir a qualidade de vídeo em tempo ótimo para que a reprodução continue sem interrupções. Os resultados obtidos foram positivos em todos os testes, isto é, houve nenhum ou baixo número de *stalls*. Embora em algumas situações ocorreram de fato *stalls*, a ocorrência deles foi no início do vídeo (aos zero segundos) de vídeo, logo, era pouco perceptível ao usuário. Esse comportamento de *stall* no segundo zero do vídeo se deve ao comportamento escada que o *player* utiliza para reprodução, especialmente nos segundos iniciais. Por conta disso, pouco afetou a qualidade de experiência. O melhor resultado foi da abordagem *Round Robin* com nenhum *stall* no *dataset* carro e uma média de 0,2 no *dataset* paraquedas. Por outro lado, o pior resultado foi da abordagem aleatória. Com pelo menos um *stall* por reprodução no *dataset* carro.

Por último, o consumo de largura de banda durante a reprodução do vídeo pode ser observado. A abordagem sem SDN se mostrou superior nessa métrica no *dataset* do paraquedas com um consumo médio de apenas 0,3 Mbps. Já o método proposto, é o

segundo mais custoso, com 2,39 Mbps sendo utilizados.

Foi realizado também, o monitoramento da variação do *bitrate* do vídeo em relação ao tempo. Para tal, o vídeo é executado e consumido por inteiro. Haja vista as funcionalidades de monitoramento implementadas no *player*, neste caso em especial, a de monitoramento do *bitrate*, a cada execução é gerado um relatório com todas as características durante a execução. Após 10 execuções para cada *dataset* e para cada abordagem, é executado o cálculo da média de *bitrate* em cada segundo de execução. Por isso, em algumas situações nos gráficos apresentados nas Figuras 5 e 6 o *bitrate* correspondente a abordagens diferentes, encontra-se sobreposto.

De qualquer forma, ao observar a Figura 5 nota-se que dentre as abordagens testadas, a que obteve a menor média de *bitrate* foi a sem SDN. As outras abordagens, contudo, foram bastante similares. No início, do segundo 1 ao 10, todos tem comportamento semelhante devido ao efeito escada comentado anteriormente. Já a partir do segundo 10 é notável uma leve variação entre eles. Entretanto, do segundo 40 ao segundo 70, é possível notar uma diminuição expressiva da média do *bitrate* da abordagem sem SDN.

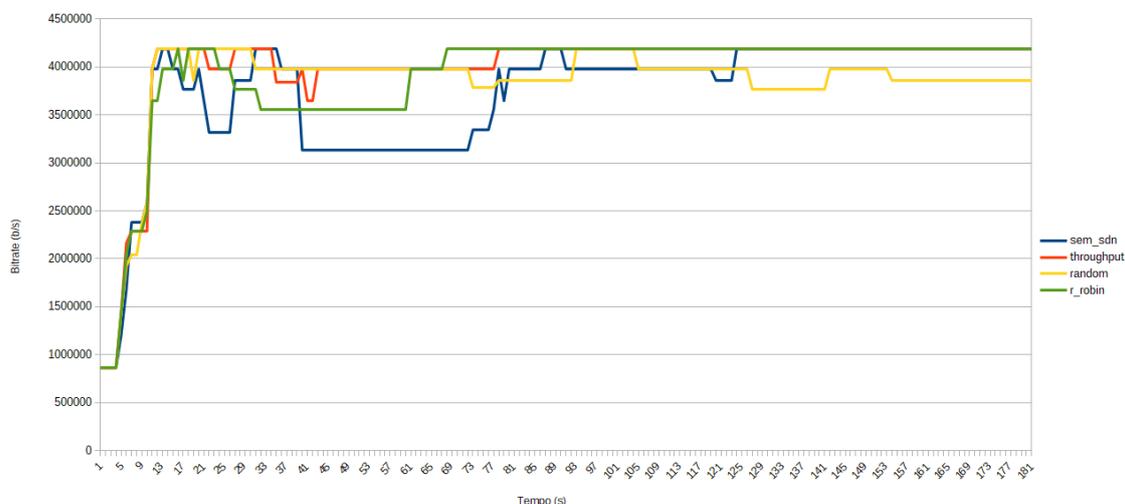


Figura 5. Resultado Final, média bitrate por método - Carro

Já no segundo *dataset* (paraquedas), conforme a Figura 6, o comportamento de todas as abordagens foi extremamente parecido. Isso se deve ao fato de que esse *dataset* era um vídeo cuja resolução mais alta é de 1280×720 *pixels*, menor portanto que o *dataset* carro, além da compressão VP9. Por essas razões o descarregamento desse vídeo era rápido.

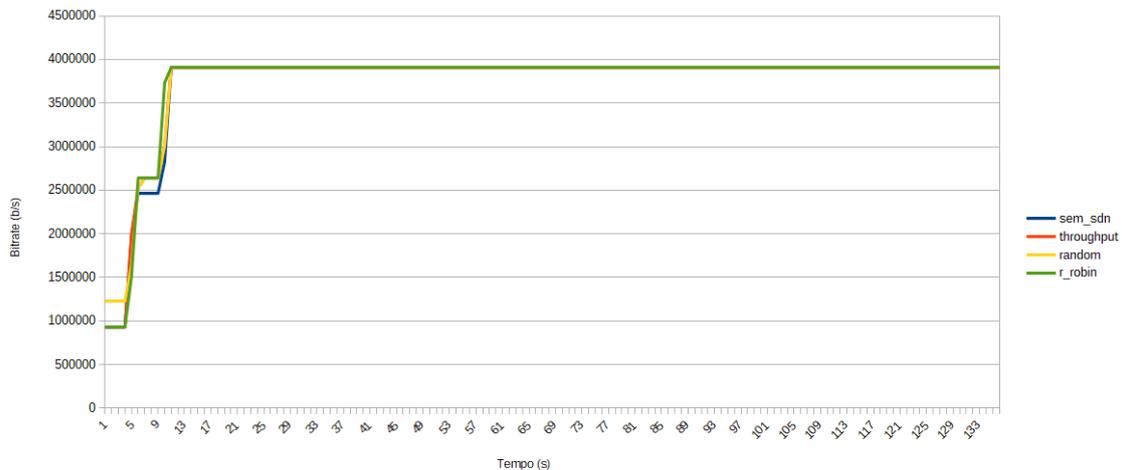


Figura 6. Resultado Final, média bitrate por método - Paraquedas

De modo geral, de acordo com os experimentos realizados, o método proposto se mostra promissor, haja visto que consegue diminuir o tempo de resposta da requisição de conteúdo de vídeo MPEG-DASH, mantendo média de *bitrate* elevada, preservando assim a qualidade de experiência do usuário. Entretanto, mesmo com desempenho elevado é o segundo método que consome mais largura de banda, o que é um fator importante a ser levado em consideração caso seja implementado em produção.

5. Conclusão

Com a popularização da Internet e avanços tecnológicos referentes a arquitetura e estrutura de rede, o consumo de vídeo se tornou uma prática diária no cotidiano de muitas pessoas. Contudo, apesar desses avanços e constantes melhorias, ainda há diversos desafios a serem superados. Um deles é o balanceamento de carga de modo que a rede de distribuição de conteúdo consiga suportar uma alta demanda de requisições e não prejudique a qualidade de experiência do usuário final.

Este trabalho apresenta uma possível solução ou pelo menos atenuação a esse problema. Para tal, foi idealizado um sistema que utiliza o paradigma Redes Definidas por Software para interceptar os pacotes durante a reprodução do conteúdo em vídeo, e através da análise da métrica taxa de transferência dos servidores de conteúdo disponíveis, efetuar a mais adequada escolha de servidor, para servir o conteúdo, de maneira a maximizar sua taxa de transferência. A avaliação realizada teve como principal objetivo destacar que apesar de se tratar de uma solução que modifica os fluxos de conexão, não há percepção desse redirecionamento na reprodução do conteúdo consumido pelo usuário e portanto da qualidade de experiência vivenciada. Além disso, a abordagem proposta apresenta ganhos em diversos aspectos como tempo de resposta, número de trocas de qualidade e *stalls*.

Contudo, são necessários mais experimentos referentes a escalabilidade horizontal da abordagem proposta para a constatação de sua completa aplicabilidade. Ainda, é intencionado realizar comparações com outras abordagens, bem como incrementar a decisão de balanceamento agregando valores de mais métricas relevantes tanto à QoE do usuário final, como de ganho em melhor aproveitamento dos recursos computacionais dos servidores de vídeo.

Referências

- Bourke, T. (2001). *Server load balancing*. O'Reilly, 1st edition.
- Cardellini, V., Colajanni, M., and Yu, P. (1999). Dynamic load balancing on web-server systems. *IEEE Internet Computing*, 3(3):28–39.
- Carter, R. L. and Crovella, M. E. (1997). Server selection using dynamic path characterization in wide-area networks. In *Proceedings of INFOCOM '97*, volume 3, pages 1014–1021 vol.3.
- Chen-xiao, C. and Ya-bin, X. (2016). Research on load balance method in SDN. *International Journal of Grid and Distributed Computing*, 9(1):25–36.
- Cisco (2018). Cisco Visual Networking Index: Forecast and Trends, 2017–2022 White Paper.
- dash.js (2012). A reference client implementation for the playback of MPEG DASH via javascript and compliant browsers.: Dash-industry-forum/dash.js. Disponível em <https://github.com/Dash-Industry-Forum/dash.js>. Acesso 13 Out. de 2018.
- Doshi, B., Kumar, C., Piyush, P., and Vutukuru, M. (2015). Webq: A virtual queue for improving user experience during web server overload. In *2015 IEEE 23rd International Symposium on Quality of Service (IWQoS)*, pages 135–140.
- Duanmu, Z., Ma, K., and Wang, Z. (2017). Quality-of-experience of adaptive video streaming: Exploring the space of adaptations. In *MM '17 Proceedings of the 2017 ACM on Multimedia Conference*, pages 1752–1760. ACM Press.
- Jiang, J.-R., Yahya, W., and Ananta, M. (2015). Load balancing and multicasting using the extended dijkstra's algorithm in software defined networking. *Frontiers in Artificial Intelligence and Applications*, 274:2123–2132.
- Kaysudu, E., Çetinkaya, C., Hergüner, K., and Sayıt, M. (2016). Server selection for video streaming applications over software defined networks. In *2016 24th Signal Processing and Communication Application Conference (SIU)*, pages 1965–1968.
- Nielsen, T. C. (2016). How worldwide viewing habits are changing in the evolving media landscape. <http://www.nielsen.com/content/dam/nielsen-global/eu/docs/pdf/Nielsen-global-video-on-demand.pdf>, 73, Avenue des Pléiades B-1200 Brussels. Acesso 15 Ago. de 2018.
- Sodagar, I. (2011). The mpeg-dash standard for multimedia streaming over the internet. *IEEE MultiMedia*, 18(4):62–67.
- Stockhammer, T. (2011). Dynamic adaptive streaming over http –: Standards and design principles. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems, MMSys '11*, pages 133–144, New York, NY, USA. ACM.
- Wichtlhuber, M., Reinecke, R., and Hausheer, D. (2015). An sdn-based cdn/isp collaboration architecture for managing high-volume flows. *IEEE Transactions on Network and Service Management*, 12:15.