UNIVERSIDADE FEDERAL DE SANTA CATARINA

QUALIDADE DE SERVIÇO E NEUTRALIDADE DE REDE EM REDES DE ACESSO DEFINIDAS POR SOFTWARE

Evandro Sasse

Florianópolis

2017

Evandro Sasse

QUALIDADE DE SERVIÇO E NEUTRALIDADE DE REDE EM REDES DE ACESSO DEFINIDAS POR SOFTWARE

Trabalho Conclusão do Curso de Graduação em Ciências da Computação do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito para a obtenção do Título de Bacharel em Ciências da Computação. Orientador: Prof. Dr. Mario Antonio Ribeiro Dantas.

Florianópolis
2017

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Sasse, Evandro
QUALIDADE DE SERVIÇO E NEUTRALIDADE DE REDE EM REDES DE ACESSO DEFINIDAS POR SOFTWARE / Evandro Sasse ; orientador, Mario A R Dantas, 2017.
79 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Ciências da Computação, Florianópolis, 2017.

Inclui referências.

1. Ciências da Computação. 2. Redes Definidas por Software. 3. Qualidade de Serviço. 4. Neutralidade de Rede. 5. Redes de Acesso Definidas por Software. I. A R Dantas, Mario. II. Universidade Federal de Santa Catarina. Graduação em Ciências da Computação. III. Título.

Evandro Sasse

QUALIDADE DE SERVIÇO E NEUTRALIDADE DE REDE EM REDES DEFINIDAS POR SOFTWARE

Este Trabalho Conclusão de Curso foi julgado adequado para obtenção do Título de “Bacharel em Ciências da Computação” e aprovado em sua forma final pelo Departamento de Informática e Estatística.

Florianópolis, 17 de novembro de 2017.

_____

Prof. Rafael Luiz Cancian, Dr.

Coordenador do Curso

Banca Examinadora:

_____

Prof. Mario Antonio Ribeiro Dantas, Dr.

Orientador

Universidade Federal de Santa Catarina - UFSC

_____

Prof.ª Patrícia Della Méa Plentz, Dr.ª

Universidade Federal de Santa Catarina – UFSC

_____

Prof. Roberto Willrich, Dr.

Universidade Federal de Santa Catarina - UFSC

AGRADECIMENTOS

À universidade e ao corpo docente, que criaram tantas oportunidades na busca pelo conhecimento.

Ao meu professor orientador, Mario Dantas, pelo suporte. E aos colegas Felipe Volpato e Wander Queiroz, pelas horas de discussão.

Ao PET Computação, onde fiz diversos amigos que marcaram a minha caminhada acadêmica.

À oportunidade de participar do programa Ciências Sem Fronteiras, e aos amigos que fiz na ocasião.

Aos colegas de trabalho na SumOne, que propiciaram tão sólida experiência profissional.

À minha família pelo incentivo e apoio incondicional.

E todos os outros que participaram direta ou indiretamente dessa jornada.

RESUMO

As redes de computadores estão cada dia mais presentes, indiretamente, nas nossas vidas. Para os consumidores dessas redes a primeira preocupação, até então, é a performance. Enquanto os operadores da rede também tem sua atenção voltada para a facilidade de manutenção. Tentando resolver essas questões, precisamos classificar o conteúdo que está circulando, e portanto acabamos tendo que discutir os tópicos de privacidade e competição na Internet.

Para facilitar o gerenciamento e configuração dos equipamentos de rede, a abordagem de Redes Definidas por Software (SDN) foi criada, possibilitando aos operadores administrar a rede de maneira rápida e centralizada. (RFC7426)

Enquanto o conceito de Qualidade de Serviço(QoS), que descreve a medição da performance dos serviços, já era usado pelas redes de computadores, seu uso ainda é pouco prático. Mas quando combinado com as Redes Definidas por Software o QoS se torna muito mais poderoso. (H E Egilmez; S Civanlar; A M Tekalp, 2013)

Recentemente, visando a proteção dos usuários e consumidores da rede a ideia de Neutralidade de Rede vem sendo discutida. Com a intenção de que os dados circulados nas redes sejam tratados da mesma maneira, independente do conteúdo, fonte ou destino, pelas operadoras de rede.

Portanto existe uma grave discordância entre argumentos. Em um campo, do QoS, queremos classificar o conteúdo circulado, para aumentar a performance. Mas no outro, da Neutralidade de Rede, queremos privacidade e que o conteúdo não seja alvo de discriminação.

Palavras-chave: Redes Definidas por Software, Qualidade de Serviço, Neutralidade de Rede, Proposta de Soluções.

ABSTRACT

Computer networks are everyday more present, indirectly, in our lifes. For the customers of these networks, the first worry, until now, is performance. While for the net operators the big worry is maintenance ease. Trying to solve these points, we need to classify the content that is trafficking, so e also need to discuss topics such as privacy and competition on the Internet.

To ease the management and configuration of the network equipments, the concept of Software Defined Networks(SDN) was created, making possible to oversee the network in a fast and centralized manner. (RFC7426)

While the concept of Quality of Service(QoS), that describes the measurement of the performance of the services, was already used by legacy computer networks, it still is not practical. When we combine the Software Defined Networks with QoS it becomes much more powerful. (H E Egilmez; S Civanlar; A M Tekalp, 2013)

Recently, trying to protect the users and customers of the networks the idea of Network Neutrality is been discussed. With intentions to make all data traffic to be treated equally, no matter the content, origin or recipient, by the network operators.

Therefore there is a big gap between these arguments. In one side, of the QoS, we want to know the content that is circulating, so we can bring more performance. But on the other side, of Net Neutrality, we do not want the content to be target of discrimination.

Keywords: Software Defined Networks, Quality of Service, Network Neutrality, Proposed Solutions.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE REDUÇÕES

FCC *Federal Communications Commission*; Comissão Federal de Comunicações.

HGW *Home gateway*; Ponto de acesso residencial.

NN *Net Neutrality*; Neutralidade de Rede.

QoE *Quality of Experience*; Qualidade de Experiência.

QoS *Quality of Service*; Qualidade de Serviço.

SDN *Software Defined Network*; Rede Definida por Software.

SNMP *Simple Network Management Protocol*; Protocolo Simples de Gerenciamento de Rede.

SUMÁRIO

1 INTRODUÇÃO

Este capítulo visa dar uma ideia geral dos temas que serão abordados, estabelecer os objetivos que estão sendo buscados, e delimitar o escopo do trabalho.

1.1 APRESENTAÇÃO

As Redes Definidas por Software são uma abordagem moderna para problemas de controle e gerenciamento de redes de computadores. Utilizando uma interface aberta e conhecida o comportamento da rede pode ser controlado por software externo ao equipamento. Esses seriam os equipamentos intermediários, *switches*, que realizam as trocas de pacotes entre a origem e destino. Isso é, cria-se um novo nível ou plano de abstração, onde o equipamentos intermediários são apenas responsáveis por executar o encaminhamento de pacotes do modo que lhe é repassado; enquanto a decisão de quais devem ser esses encaminhamentos é de decisão de um software, em outro plano. Esses planos são então conhecidos como Encaminhamento, do equipamento da rede, e de Controle, do software. A separação permite que o plano de controle, relacionado ao gerenciamento da rede, esteja em constante e rápida evolução, que reflete em um mudanças dos encaminhamentos, tornando-os dinâmicos conforme o contexto de cada momento. (RFC7426)

O conceito de Qualidade de Serviço já existe a muito tempo, mesmo nas redes comuns não-SDN. Consistindo de todas as características do serviço de telecomunicação prestado, com o objetivo de melhorar essas métricas/indicadores. Por exemplo, já era utilizado quando havia apenas serviços de telefonia, medindo ruído e *delay*. Compreende todas as métricas de performance das conexões, como *delay*, velocidade, disponibilidade, confiabilidade, e outras (ITU E.800 09/08). Com o advento das Redes Definidas por Software a manutenção da QoS torna-se muito poderosa pois pode ser automatizada, diminuindo a complexidade de implementação. Pelo motivo que a implantação de melhorias de Qualidade de Serviço ainda é muito complexa, e requer muito trabalho manual na configuração de cada equipamento, nas redes comuns (W Kim; P Sharma; J Lee; S Banerjee; J Tourrilhes; S Lee; and P Yalagandula, 2010).

Com o aumento da facilidade de controle da rede, preocupações com problemas de privacidade, censura e competição vem sendo criadas e crescendo. Exemplos desses problemas estão sendo observados pelo mundo inteiro: votações no congresso dos EUA para permitir que as operadoras de Internet possam vender as informações de navegação de seus clientes (ACLU, 2017); governos de países como Líbia, Egito e China bloqueando acesso a grande parte da Internet (M Bailey; C Labovitz, 2012); operadoras de Internet dando preferência dos pacotes de um provedor de serviço sobre outro (H K Cheng; S Bandyopadhyay; H Guo, 2011).

O debate dessas preocupações recebe o nome de Neutralidade de Rede, que recebe diversas definições, mas pode ser condensado em “maximizar a utilidade da rede de informações públicas, aspirando tratar todo o conteúdo, páginas e plataformas igualmente” (H K Cheng; S Bandyopadhyay; H Guo, 2011). Buscando

que o conteúdo que é transportado pela rede não seja filtrado ou diminuído em facilidade, qualidade e velocidade, por quem o transporta.

Vê-se então uma dicotomia entre aquilo que é buscado pela QoS e pela Neutralidade de Rede, já que na QoS buscamos classificar o conteúdo circulado para que seja realizada a priorização, mas com a Neutralidade de Rede queremos privacidade na circulação de nosso conteúdo e que ele não sofra despriorização (TECHTARGET, 2017). Mas alguns trabalhos já tentam atender os dois lados do espectro como C Liss; T Fendler; D Gajic; A Vensmer (2015), e H H Gharakheili; A Vishwanath, V Sivaraman (2016).

Este trabalho visa apresentar e propor soluções que possam aliar estes conceitos, compatibilizando toda a facilidade de configuração e performance que pode ser atingida pelas SDNs e mantendo os princípios da Neutralidade de Rede. Maximizando a utilidade obtida pelos usuários/consumidores.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Mostrar se é possível aliar e compatibilizar os conceitos de Qualidade de Serviço e Redes Definidas por Software (SDN) à Neutralidade de Rede, analisando soluções que possivelmente poderiam ser utilizadas por operadoras e clientes no futuro. Comparando as soluções de maneira qualitativa, e demonstrando como poderiam ser comparadas quantitativamente utilizando um simulador de SDN. E por fim então validar se a compatibilização desses conceitos é possível e aplicável.

1.2.2 Objetivos Específicos

A. Mostrar a importância da Neutralidade de Rede;
B. Sinalizar os problemas da abordagem atual de Qualidade de Serviço quanto à Neutralidade de Rede;
C. Propor novas soluções, com características e problemáticas de implementação;
D. Explanar e comparar soluções propostas entre si e com as soluções existentes, qualitativamente;
E. Demonstrar como pode ser realizada a comparação quantitativa;
F. Apresentar as conclusões obtidas em termos de possibilidade e aplicabilidade.

1.3 JUSTIFICATIVA

É fácil ficar confuso sobre como os elementos de Redes Definidas por Software, Qualidade de Serviço e Neutralidade de Rede (NN) interagem, por serem muito dinâmicos, e a NN não ser muito formalizada. Esse trabalho deixa mais claro essas interações, e as restrições que um causa sobre o outro.

E sabendo que a Neutralidade de Rede parece se opor à Qualidade de Serviço, esse trabalho apresenta e propõe soluções que diminuem a tensão entre esses termos, tentando compatibilizá-los. Tentando maximizar a utilidade que o usuário final da rede vai obter.

1.4 METODOLOGIA

Durante todo o período desde a idealização do tema até o fim do desenvolvimento deste trabalho foi organizado a leitura de artigos e materiais diversos relacionados aos tópicos. A leitura foi organizada através de reuniões e discussões semi-semanais com o professor orientador que também contaram com a presença do mestrando Felipe Volpato, e Wander Queiróz

Paralelamente, foram acompanhadas notícias sobre a Neutralidade de Rede e imaginadas as soluções propostas posteriormente. Também contando com o estudo de como usar os softwares necessários para simulação de SDNs.

Após um primeiro momento de estudo e escrita da fundamentação teórica, estado da arte e trabalhos relacionados, foram analisadas as propriedades em alto nível de soluções já existentes. De modo a propor soluções diferentes, que também envolvem os problemas já conhecidos.

A análise das soluções propostas então se estendeu até o fim do trabalho, então realizado de maneira individual. As simulações de redes foram realizadas com o programa Mininet. Utilizando das simulações e das características qualitativas, comparamos as soluções e sua aplicabilidade para tirar as conclusões finais do trabalho.

2. FUNDAMENTAÇÃO TEÓRICA

Esta seção introduz os conceitos que devem ser previamente conhecidos para o entendimento do desenvolvimento.

2.1 CONCEITOS BÁSICOS

A seguir são feitas as definições de alguns termos que são necessários conhecer para entendimento completo de outros conceitos e do desenvolvimento. Mas não são específicos, ou são tangenciais, de um dos conceitos principais.

2.1.1 Rede comum, *legacy*, ou não-SDN

A partir desse ponto os termos comum, *legacy* e não-SDN serão utilizados para se referir à redes de computadores que não se utilizam do conceito de Redes Definidas por Software.

2.1.2 Métricas de rede

Diversas características de uma rede podem ser medidas para obtermos informação de uso e performance dessa. Algumas dessas características e suas definições são apresentadas à seguir.

O termo velocidade, largura de banda e *bandwidth* refere-se a taxa de transmissão (ITU E.800 09/08). Comumente é medido em *bits* por segundo(b/s ou

bps), ou algum múltiplo como *kilobits* por segundo(kb/s ou kbps) ou *megabits* por segundo(Mb/s Mbps), por exemplo “10Mbps”.

Delay ou latência refere-se ao tempo entre o envio de uma mensagem e o recebimento dela no destino. *Ping, round-trip delay* ou tempo de resposta especificam o tempo entre o envio de uma mensagem e o recebimento de uma confirmação. E *jitter* remete a variação do *delay* ao longo do tempo. Normalmente esses termos são medidos em milissegundos(ms), por exemplo “120ms de *ping*”.

Uptime refere-se à medida de tempo que um serviço fica ativo ou disponível. Em contrapartida o *downtime* é a medida de tempo que um serviço fica inativo ou indisponível. Geralmente descritos por porcentagens, por exemplo “99.9% de *uptime*”.

2.1.3 Inspeção Profunda de Pacotes (DPI)

Ou *Deep Packet Inspection*(*DPI*), consiste de analisar(*“parsear”* do inglês *to parse*) o conteúdo inteiro dos pacotes sendo transmitidos na rede. O objetivo original do *DPI* é melhorar as capacidades de equipamentos como o *firewall* da rede, permitindo a detecção e bloqueio de tentativas de intrusão e ataques. Analisando o conteúdo dos pacotes para buscar comportamento suspeito. (SYMANTEC, 2017)

O motor de inspeção, como é conhecida a ferramenta que inspeciona os pacotes, pode se utilizar de técnica como comparação da “assinatura” de pacotes, técnicas heurísticas e estatísticas, e qualquer anomalia nos fluxos. Analisando não apenas os cabeçalhos, mas todo o conteúdo dos pacotes aumenta a quantidade de

padrões que podem ser encontrados para identificar ataques, mas também torna a tarefa muito mais computacionalmente cara. Visa então proteger os usuários e a rede de ataques externos. (SYMANTEC, 2017)

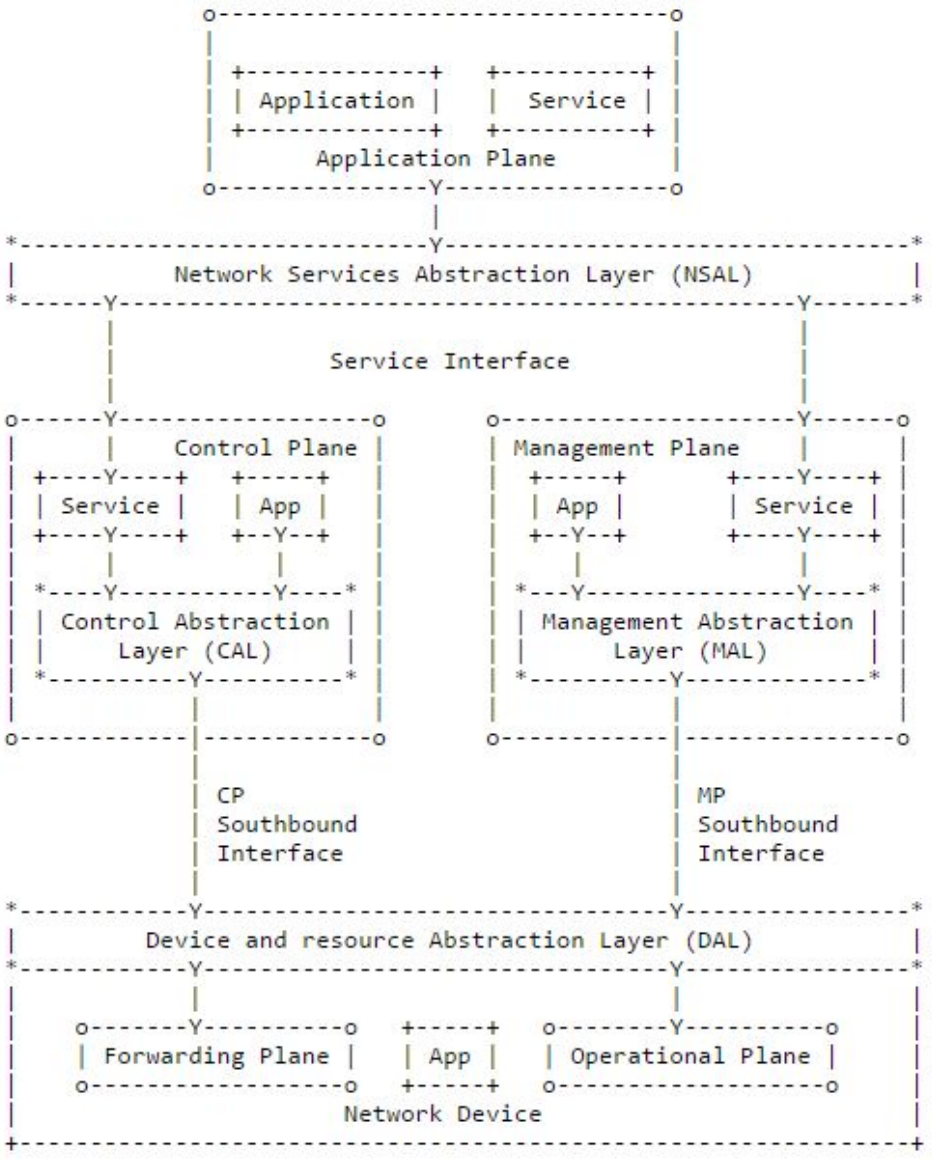
2.2 REDES DEFINIDAS POR SOFTWARE (SDN)

As Redes Definidas por Software estão dentro do paradigma de redes programáveis. E refere-se à habilidade de uma rede prover interfaces para que aplicações configurem cada equipamento da rede, de forma dinâmica, de modo a mudar o comportamento da mesma. Uma das ideias principais é a divisão entre o plano de controle e de encaminhamento.(RFC7426)

A ideia de abstrair a arquitetura da rede em diferentes planos não é nova das SDNs. Foi pensada para prover flexibilidade, facilidade de implementação e rápida inovação. O conceito já foi usado por diversas outras soluções e pode envolver diversos planos. Exemplo disso é a proposta “4D” por Greenberg, et al (2005), que leva esse nome por causa dos seus 4 planos: Decisão, Disseminação, Descoberta e Dados.

Na *Figura 1* são vistos os Planos de Controle (*Control Plane*) e de Encaminhamento (*Forwarding Plane*). Além disso são mostrados diversos outros Planos, como o de Gerenciamento (*Management Plane*), de Aplicação (*Application Plane*) e o Operacional (*Operational*), e como cada um se encaixa na arquitetura. A partir desse momento o Plano de Encaminhamento também será referido como Plano de Dados (*Data Plane*).

Figura 1 - A arquitetura de uma SDN.

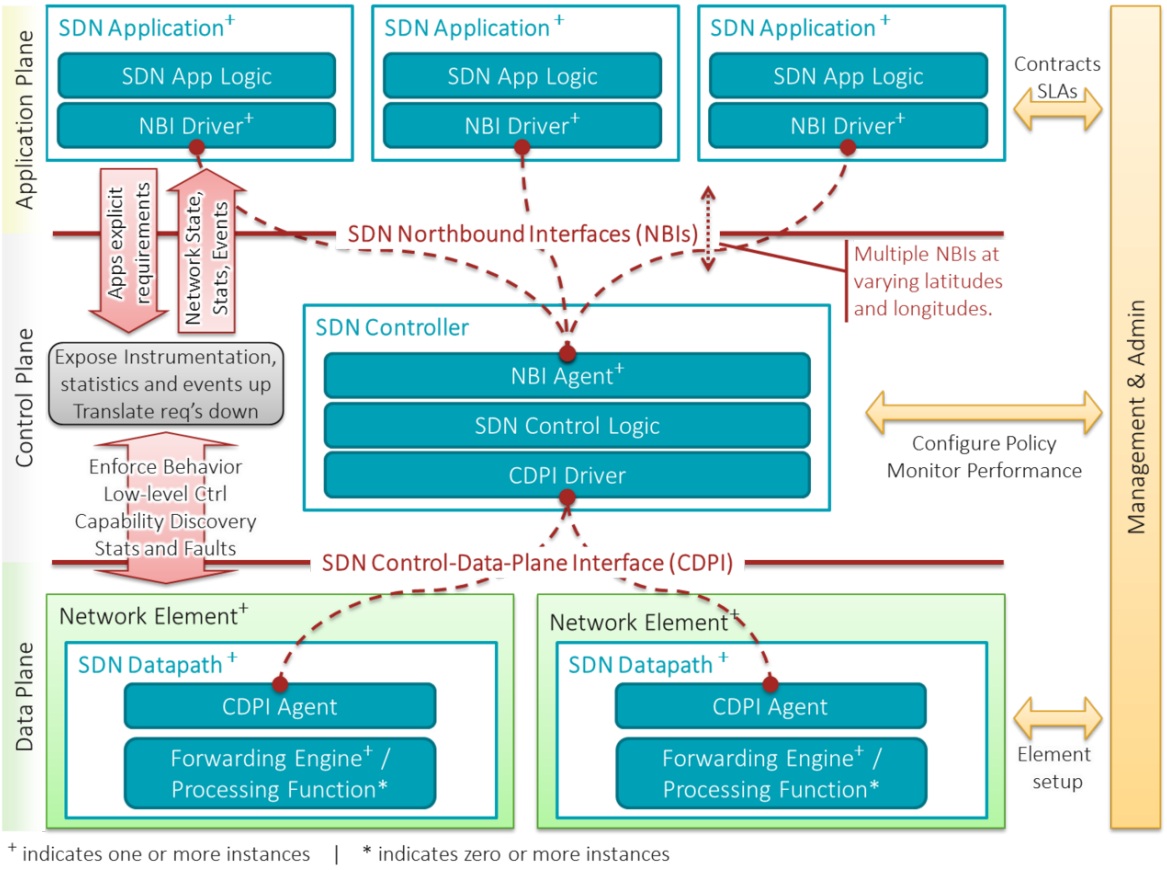


Fonte: (RFC7426).

A abstração mostrada na Figura 2 também é muito comum, escondendo elemento pouco referidos, como o Planos Operacional e de Gerenciamento, e as camadas de abstração. E simplificando o Plano de Controle à apenas um

elemento, o Controlador SDN. Os papéis dos elementos escondidos são incorporados por elemento próximos da Figura 1.

Figura 2 - Simplificação da arquitetura de uma SDN.



Fonte: WIKIPEDIA, 2017.

O Plano Operacional é responsável por gerenciar o estado de um dispositivo da rede, por exemplo quantas portas o dispositivo possui, qual o *status* de cada porta, entre outras coisas (RFC7426). Na simplificação mostrada na Figura 2 essas funcionalidades são incorporadas no Plano de Dados.

O Plano de Gerenciamento realiza o monitoramento e configuração dos dispositivos, comumente através do Plano Operacional (RFC7426). Portanto está

ligado à configurações que seriam feitas manualmente por um administrador da rede (SDXCENTRAL, 2017). A diferença dos objetivos deste plano para o Plano de Controle original ficará mais clara posteriormente. Na Figura 2 este plano é englobado pelo Plano de Controle.

É interessante ressaltar que toda a abstração da SDN não requer distinção entre implementações físicas/*hardware* ou virtuais/*software.* Portanto pode ser utilizada para descrever redes físicas, virtuais e híbridas. (RFC7426)

2.2.1 Plano de Controle

O Plano de Controle é principalmente responsável por gerenciar e configurar o Plano de Encaminhamento, então ele instrui como deverão serão manuseados os pacotes, qual a topologia atual da rede, escolha e mudança de rotas, entre outros (RFC7426).

As interfaces do Plano de Controle são popularmente conhecidas como *Northbound* (De direção norte) e *Southbound* (De direção sul) (WEBWERKS, 2017). Essas interfaces aparecem na *Figura 1* como *Service Interface* e *CP Southbound Interface* respectivamente. E na *Figura 2* como *SDN Northbound Interfaces* e *SDN Control-Data-Plane Interface* respectivamente.

Utilizando a abstração da Figura 2, observamos que através da interface Northbound o Plano de Controle se comunica com o Plano de Aplicação. E através da interface Southbound se comunica com o Plano de Dados.

Os elementos pertencentes ao Plano de Controle são conhecidos como Controladores, como pode ser visto na *Figura 2*, sob *SDN Controller*. Teoricamente

poderíamos ter diversos Controladores, até mesmo com diferentes implementações, em uma mesma SDN. Mas a partir deste ponto do trabalho iremos considerar a presença de apenas um Controlador, como na figura.

2.2.2 Plano de Aplicação e a Interface Northbound

O Plano de Aplicação abstraí as aplicações que irão configurar e monitorar a rede dinamicamente, utilizando dos serviços dos Planos de Controle e Gerenciamento. Algumas aplicações também podem ser utilizadas como serviços por outras aplicações. Podem por exemplo ser aplicações que irão configurar rotas prioritárias para certos fluxos. (RFC7426)

São as aplicações deste plano que programaticamente ditam o comportamento esperado da rede. Irão portanto usar a interface Northbound para consumir os dados disponibilizados pelo Plano de Controle e fazer decisões (OPEN NETWORKING, 2013). É neste plano que as aplicações de Qualidade de Serviço, em uma SDN, trabalham.

A interface Northbound pode ser implementada de diversas maneiras, mas é a maneira mais popular é através de uma API REST, implementada no Plano de Controle (W. Zhou; L. Li; M. Luo; Wu Chou, 2014). Portanto as aplicações podem, teoricamente, ser independentes da implementação do Controlador escolhido.

2.2.3 Plano de Dados e a Interface Southbound

O Plano de Dados ou Plano de Encaminhamento, engloba os diversos equipamentos da rede, físicos ou virtuais, e sua habilidade de receber, manusear, filtrar e encaminhar pacotes. (RFC7426)

Os equipamentos de rede(físicos ou virtuais) para SDNs precisam conseguir conversar através da interface Southbound. O primeiro protocolo para isso, e que criou diversas especificações dissidentes, é o OpenFlow. Mas é necessário explicitar, pois existe confusão nesse aspecto, que as SDNs não estão necessariamente conectadas as características do OpenFlow. (WIKIPEDIA, 2017)

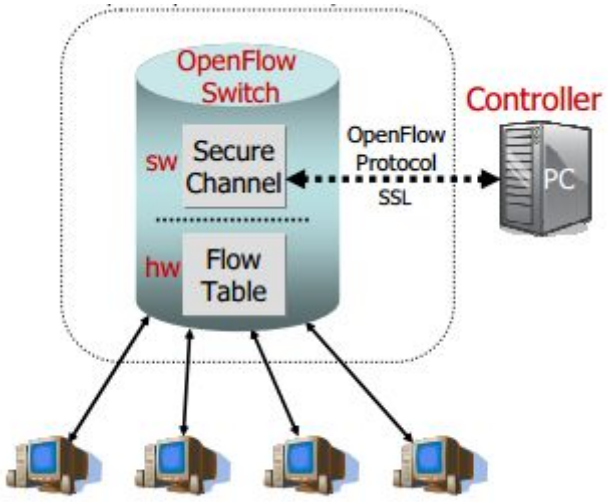
Outros protocolos para a interface Southbound também podem podem ser utilizados, mantendo as ideias de programabilidade, abstração, e velocidade de mudança. Por exemplo *OpFlex* é uma alternativa declarativa, criada pela Cisco, que tenta aumentar a performance aumentando a responsabilidade dos elementos da rede (CISCO, 2017).

O Controlador e o elemento da rede devem possuir protocolos compatíveis. Portanto acabam sendo criados controladores capazes de suportar diversos protocolos, como o *OpenDaylight* que tem plugins para suportar *SNMP* (Simple Network Management Protocol)*, OpFlex*, *OpenFlow* (e suas diversas versões), entre outros (OPENDAYLIGHT, 2017). E elementos de rede que também suportam diversos protocolos, como o simulador Mininet que suporta diversas versões do OpenFlow (GITHUB, 2017).

2.2.4 OpenFlow

O protocolo OpenFlow descreve uma maneira segura de transmitir as rotas que os elementos da rede deverão usar. Mantendo uma interface única para que o Controlador possa comandar diversos elementos sem precisar conhecer detalhes de implementação de cada um. Foi pensado inicialmente apenas como uma maneira de pesquisadores atualizarem em tempo real as rotas dos equipamentos, que estavam sendo utilizados para testes ao mesmo tempo que tráfego real também acontecia e não deveria ser interferido. (N McKeown; T Anderson; H Balakrishnan; G Parulkar; L Peterson; J Rexford; S Shenker; J Turner, 2008)

Figura 3 - O protocolo *OpenFlow* idealizado.



Fonte: (N McKeown; T Anderson; H Balakrishnan; G Parulkar; L Peterson; J Rexford; S Shenker; J Turner, 2008).

Como mencionado anteriormente, o elemento de rede deve ser compatível com o protocolo. Portanto, podemos ver na Figura 3 que o equipamento é descrito como um *OpenFlow Switch*, ou seja um *switch* de rede que suporta o protocolo

OpenFlow. Também vemos na figura onde o protocolo se encaixa, fazendo a comunicação entre o Controlador e o equipamento.

A ideia inicial é simples. Os equipamentos de rede até então já utilizavam uma tabela chamada *flowtable* para manter a sua lista de rotas, mas elas são um pouco diferentes dependendo da implementação. Foram identificadas um conjunto de operações que podem ser feitas, em diversas implementações de *flowtables*, o suficiente para criar os comportamentos mais simples. Essas operações descrevem o requerimento mínimo para um equipamento dar suporte ao OpenFlow.

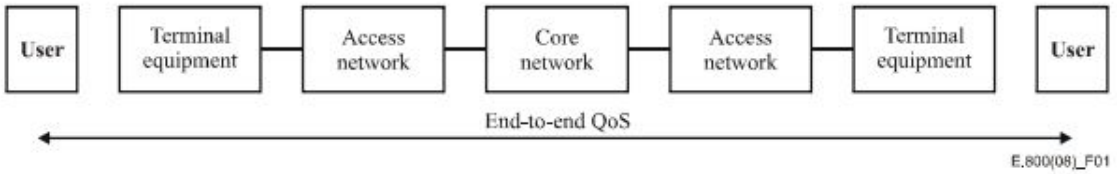
As ações básicas que precisam ser implementadas são de: encapsular os pacotes de um fluxo e encaminhar para o Controlador; encaminhar os pacotes de um fluxo para um ou mais portas; ou jogar fora os pacotes de um fluxo. Desse modo podemos estabelecer o ciclo de vida de um fluxo: o fluxo primeiramente é desconhecido, portanto é encaminhado ao Controlador para ser feita a decisão de uma rota; uma rota já foi decidida para o fluxo, os pacotes são encaminhados para as respectivas portas; ou foi decidido que o fluxo não de interesse e pode ser jogado fora.

A *flowtable* possui três campos em cada uma das entradas: o *header* dos pacotes que identificam um fluxo; a ação/operação a ser tomada para os pacotes deste fluxo; e estatísticas, como o número de pacotes e de bytes que passaram por este fluxo, ou o horário do último pacote desse fluxo. (N McKeown; T Anderson; H Balakrishnan; G Parulkar; L Peterson; J Rexford; S Shenker; J Turner, 2008)

2.3 QUALIDADE DE SERVIÇO (QoS)

A Qualidade de Serviço refere-se à características do serviço de rede que ativamente afetam as necessidades do usuário. Comumente está conectado à aspectos como velocidade, simplicidade, confiabilidade, precisão, corretude, entre outros. (ITU E.800 09/08)

Figura 4: Simplificação da arquitetura fim-a-fim.

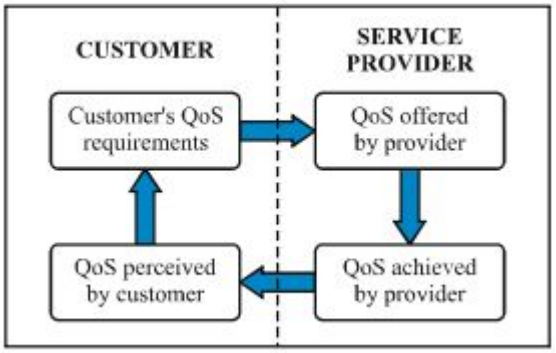


Fonte: (ITU E.800 09/08).

Tais aspectos afetam então todo o caminho que os pacotes que saem e chegam ao usuário passam. Na Figura 4 temos uma simplificação desse caminho. O única parte da rede que está sob controle dos usuários/clientes estão na borda da rede, na figura chamados de *Terminal equipment*. As redes de acesso, *Access networks*, estão sob poder das operadoras ou provedoras de serviço. E o núcleo, *Core network*, está sob poder de um grupo menor de operadoras.

Na Figura 4 podemos observar alguns pontos de vista sobre a Qualidade de Serviço, divididos verticalmente em pontos de vista do cliente, à esquerda, e da provedora do serviço, à direita. Mas também podemos fazer uma divisão horizontal com o Acordo de Nível de Serviço, ou *Service Level Agreement*(SLA) na parte superior, e a Qualidade de Experiência, ou *Quality of Experience*(QoE) na parte inferior. Estes pontos serão mais explorados posteriormente.

Figura 5 - 4 pontos de vista da Qualidade de Serviço.



Fonte: (ITU E.800 09/08).

A manutenção da Qualidade de Serviço então visa cobrir o Acordo de Nível de Serviço de todos os clientes e manter a Qualidade de Experiência agradável. É necessário ir além da ideia do modelo de “melhor esforço”, onde todos os fluxos da rede são tratados da mesma maneira. É necessário tomar decisões inteligentes, conhecendo as necessidades de cada fluxo (NETLAB, 2017). As características de alguns tipos de fluxos serão explicados posteriormente.

Algumas mecanismos simples são empregadas atualmente. Canais ou fluxos reservados para comunicações de altíssima prioridade, como realizado pelo protocolo *RSVP*; onde o resto do fluxo da rede é limitado temporariamente garantindo que sempre exista a disponibilidade da rede para a comunicação de certo fluxo (RFC 2205). Ou um esquema de prioridade simples, onde as comunicações de um cliente tem prioridade sobre as de outro cliente, independente das características de cada fluxo de comunicação (Jon Crowcroft, 2007).

Outro mecanismo é o *over-provisioning* (Murali Kodialam; T. V. Lakshman; S Sengupta, 2006). Existem 10 clientes em uma rede, com um limite de 10Mbps para cada um, idealmente a rede deveria suportar 100Mbps, mas isso se tornaria muito

caro. A provedora pode optar por suportar uma capacidade média, do comportamento que seja mais observado na rede; por exemplo, talvez o cenário mais comum seja que apenas um cliente esteja usando seu limite máximo de 10Mbps, outros 4 clientes estejam usando metade, 5Mbps, e os outros 5 clientes estejam usando pouco, 1Mbps, para um total de 35Mbps; isso poderia suportar a necessidade dos clientes a maior parte do tempo. Mas a qualquer pico de utilização que acontecesse, o serviço se tornaria lento e ruim para todos os clientes. O *over-provisioning* então ainda leva em conta que nem todos os clientes estarão utilizando o seu limite máximo em todos os momentos, mas a provedora deve se basear nos picos de utilização para estabelecer qual deve ser a capacidade da rede.

Historicamente o protocolo *IPv4* possuía um campo chamado *Type Of Service*(*TOS*) no cabeçalho, onde armazenava informações do tipo de serviço que o pacote requer, mapeando os parâmetros: precedência, latência, banda e confiabilidade (RFC791). E no protocolo *IPv6* o campo é dividifo nos bits *Differentiated Services Codepoint*(*DSCP*) (6 bits) e *Explicit Congestion Notification*(*ECN*) (2 bits) (RFC2474) (RFC 3168). Tais bits possibilitam o uso do mecanismo *DiffServ* e a priorização de um pacote sobre outro a cada encaminhamento, mas ainda não permitem uma visão global da priorização de um pacote na rede. A evolução desse tipo de marcação de pacote seria o *Multiprotocol Label Switching*(*MPLS*) onde então encontramos problemas de para se adaptar e reconfigurar em tempo real.

2.3.1 Acordo de Nível de Serviço (SLA)

O Acordo de Nível de Serviço é a formalização, contratual, das obrigações da provedora e cliente quanto à Qualidade de Serviço da rede. São definidas quantitativamente as métricas que devem ser atingidas. Também está ligado a contratos de manutenção, quando terceirizados pela provedora. (D.C. Verma, 2004)

As métricas utilizadas são qualquer uma das características quantitativas da rede. Devem ser apresentados os métodos de medição e resolução de problemas, como também as consequências para quando as medidas não são atingidas. Ou seja, um contrato de prestação de serviço. Isso é necessário pois dependendo da natureza do serviço, ele pode afetar vidas e finanças. Por exemplo. O serviço de Internet de uma casa pode possuir cláusulas como largura de banda máxima que possa ser atingida pelo menos 50% do tempo, largura de banda mínima que possa ser atingida 95% do tempo. E o serviço de Internet de um banco seria mais restrito com cláusulas como 99% de *uptime* com certa largura de banda mínima garantida, e consequências financeiras caso a provedora não consiga cumprir o contrato, portanto o serviço do banco também é muito mais caro.

2.3.2 Qualidade da Experiência (QoE)

A Qualidade de Experiência define aspectos da experiência do usuário com um sistema ou serviço, expandindo também para a qualidade do conteúdo provido. Descrevendo o encanto ou desprazer do usuário ao utilizar o serviço, observando a satisfação das expectativas dele. (K Brunnström; S Beker; K De Moor, 2013)

Lidando com o ponto de vista subjetivo dos usuários do serviço, que pode mudar conforme experiências prévias de cada usuário, expectativas geradas,

personalidade, etc. Portanto são difíceis de ser metrificados, pois normalmente só podem ser obtidos de maneira descritiva de um usuário. (K Brunnström; S Beker; K De Moor, 2013)

Mas, as causas de problemas observados na QoE podem ser medidos na fonte. Isto é, a causa de o conteúdo de um serviço estar lento, por exemplo, provêm de que alguma métrica da Qualidade de Serviço esteja fora esperado. Seja algum problema com a largura de banda ou perda de pacotes, deveríamos retroalimentar essas métricas de modo a melhorar a Qualidade de Experiência.

2.3.3 Tipos de fluxo

Diferenciar as características de cada fluxo é necessário para que possamos aperfeiçoar a Qualidade de Serviço. Por exemplo, podemos atrasar pacotes de um fluxo que não tenha perfil de tempo-real, ou diminuir a largura de banda de fluxo que seja uma transferência em *background*.

Alguns exemplos mais específicos são:

- Jogos: características de tempo-real como *delay* e *jitter* são muito importantes para não frustrar o usuário, mas a largura de banda utilizada é pequena.
- *Backup* e armazenamento: podem precisar transferir muitos dados e precisam de ampla largura de banda, mas são feitos em *background* e independem do *delay*.
- Vídeo e voz: podemos diferenciar entre “sob demanda”(*on-demand*) e “ao vivo”, mas ambos podem requerer grande largura de banda,

enquanto também podem aumentar ou diminuir a qualidade do vídeo, portanto largura de banda, dependendo da conexão.

- *On-demand*: Serviços onde o conteúdo já existe armazenado previamente, como Youtube ou Netflix. Como um buffer de até alguns minutos pode ser armazenado, aspectos de tempo-real não são muito importantes.
- Ao vivo: Videoconferências ou *streaming* ao vivo, são altamente suscetíveis a *delay*. Pequenas mudanças de tempo podem tornar serviços completamente inutilizáveis.

- Páginas da Internet: Altamente dependente de características de tempo-real, e podem requerer que o conteúdo seja baixado rapidamente, mas não costuma ser muito grande.

Uma maneira das maneiras de classificar cada tipo de tráfego, como os dos exemplos anteriores, é proposta em 802.1Q, sem relação com a cadama de rede mencionada acima. Essa classificação pode ser observada na Tabela 1.

Tabela 1 - Tipos de tráfego

Prioridade	**Tipo de tráfego**
1 *	*Background*
0 *	*Best effort*
2	*Excellent effort*
3	*Critical Applications*
4	*Video,* menos de 100ms de latência
5	*Voice,* menos de 10ms de latência

6	*Internetwork Control*
7	*Network Control*

Fonte: (802.1Q).

Da Tabela 1 é interessante destacar alguns tipos de tráfego. Com a menor prioridade temos os serviços que executam em plano de fundo, mas tem que tem seu valor de prioridade como *1*, de modo a diferenciar do valor padrão *0*, que pertence aos serviços mais comuns, classificados como de melhor esforço. Com maior prioridade temos duas classes para os serviços de transmissão de voz e vídeo em tempo real, relacionando-as com valores esperados de latência. E duas classes para pacotes relacionados com o gerenciamento da rede, considerados como de altíssima prioridade.

2.3.5 Qualidade de Serviço em Redes Definidas por Software

Para irmos além dos mecanismos apresentados anteriormente, queremos conhecer as características de cada fluxo em nossa rede, sua prioridade em comparação com os outros fluxos daquele cliente, sua prioridade em comparação com os fluxos de outros clientes, e quais características mais afetam a Qualidade de Experiência para aquele fluxo.

Novos fluxo são introduzidos a qualquer momento, fluxos até então desconhecidos, que podem nunca ter aparecido anteriormente na rede. Fluxos podem mudar suas necessidades a qualquer momento, podem ter picos de uso de

banda, etc. Além disso, abstraímos a necessidade de uma prioridade para os pacotes relacionados com o gerenciamento da rede, já que separamos logicamente o Plano de Controle do Plano de Dados.

Portanto nossos cenários são extremamente dinâmicos. E esse é o poder que a as Redes Definidas por Software trazem para a Qualidade de Serviço, permitindo que as mudanças de rota e prioridade aconteçam em tempo real na rede, com facilidade de configuração e manutenção, flexibilidade, rápida iteração, e todas as outras características que vêm sendo desenvolvidas nas SDNs. (W Kim; P Sharma; J Lee; S Banerjee; J Tourrilhes; S Lee; and P Yalagandula, 2010)

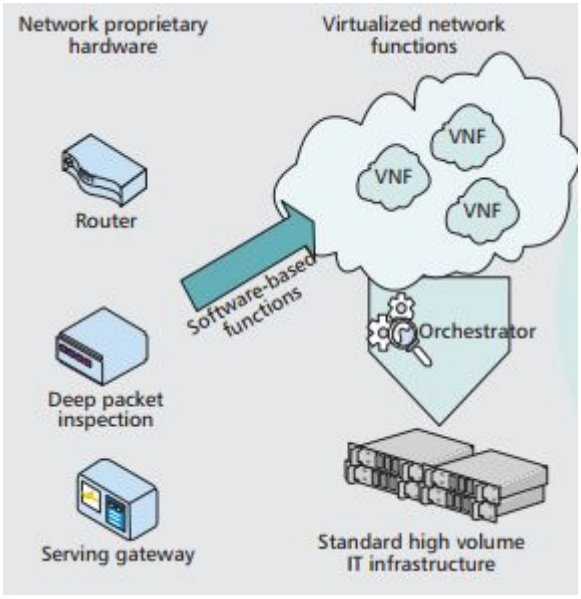
2.3.6 Virtualização da Função da Rede (NFV)

Outro conceito, separado de Redes Definidas por Software, mas que ganha poder ao ser utilizado conjuntamente, é o *Network Function Virtualization*(*NFV*). Separando as funcionalidades de equipamentos especializados de rede, dependentes de hardware proprietário, e transformando-as em software que pode ser rodado em equipamentos *off-the-shelf*. Aumenta-se a flexibilidade e diminui-se o tempo de desenvolvimento de novas funcionalidades, também diminuindo os custos de compra de equipamentos e de gerenciamento deles. (HHawil; A Shami; M Mirahmadi; R Asal, 2014)

Quando aliado com o conceito de SDN o conjunto de funções virtualizadas operam como o controlador da SDN, e o processamento de cada nodo pode ser transferido para a nuvem. Ou seja, agora os nodos da rede receberão quais funções devem executar, e essas funções são atualizadas sempre que necessário,

rapidamente. Podemos ver essa diferença na Figura 6, onde a primeira coluna mostra o cenário antes da virtualização, a segunda coluna mostra a virtualização aliada com as Redes Definidas por Software. (HHawil; A Shami; M Mirahmadi; R Asal, 2014)

Figura 6 - Virtualizar funções e servir via SDN.



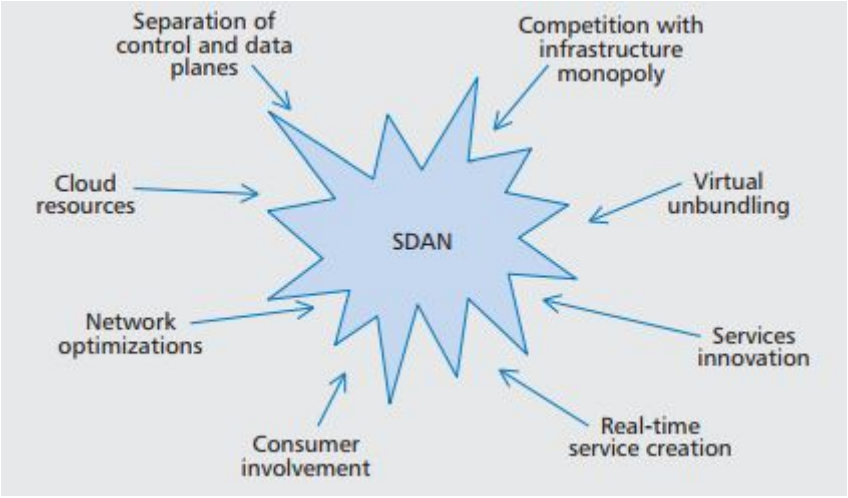
Fonte: HHawil; A Shami; M Mirahmadi; R Asal (2014).

2.3.7 Redes de Acesso Definidas por Software (SDAN)

As facilidades de gerenciamento e flexibilidades que o uso da Virtualização das Funções de Rede e das Redes Definidas por Software se aplicam muito bem sobre as grandes redes que as provedoras gerenciam. Alguns benefícios podem ser vistos na Figura 7. A utilização desses conceitos na *Access Network* fica então conhecido como *Software Defined Access Network*, ou Redes de Acesso Definidas por Software. Mas o maior benefício, que pode não ser claro a primeira vista, é a

integração do gerenciamento da rede com outros serviços, por exemplo com a rede de outras operadoras; posteriormente mostramos alguns exemplos de como poderíamos ter esse tipo de integração. (K J Kerpez, J M Cioffi, G Ginis, M Goldburg, S Galli, P Silverman, 2014)

Figura 7 - Benefícios das operadoras ao usar o SDAN



Fonte: K J Kerpez, J M Cioffi, G Ginis, M Goldburg, S Galli, P Silverman (2014).

2.4 NEUTRALIDADE DE REDE (NN)

O conceito de Neutralidade de Rede (NN), ou *Net Neutrality*, diz respeito ao ideal de que todos os dados devem ser tratados de maneira igual pela rede. Não favorecendo uma ou discriminando outra aplicação. O argumento se centraliza em uma ideia de livre competição, em que o meio onde ela acontece deve ser neutro e portanto meritocrático. (Tim Wu, 2003)

É um ideal de inovação, baseado em modelos evolutivos que são observados em mudanças tecnológicas. Isto é, pode-se descrever o processo de inovação

como“sobrevivência do mais apto” entre os diversos desenvolvedores concorrentes em uma tecnologia. Neste caso a Internet é o novo “meio” onde esses diversos concorrentes vivem, com suas aplicações competindo por espaço. E esse meio tem a sua melhor performance geral quando não tende à nenhum dos competidores. (Tim Wu, 2003)

E diverge do sentimento de que uma autoridade com controle sobre o meio poderia tomar decisões ótimas sobre ele. Pois essa autoridade também teria outros interesses que afetariam o seu compromisso com ele. Ou simplesmente porque o caminho mais promissor simplesmente é muito difícil de ser predito (Tim Wu, 2003). Indo contra a possibilidade de que uma empresa ou governo deveria ter controle do comportamento da rede.

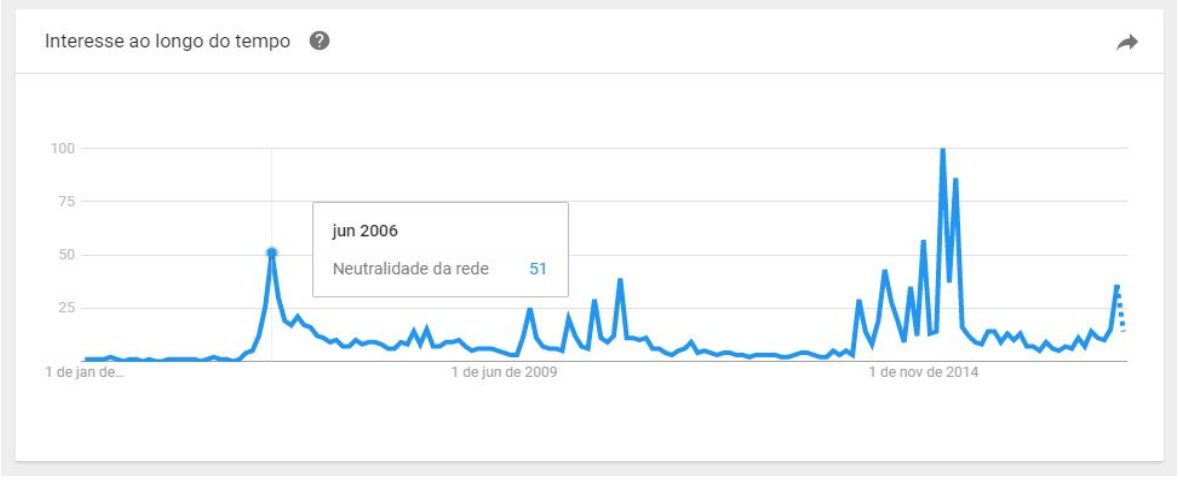
Um exemplo desses argumentos são os dados estudados da plataforma de compras Amazon, onde foi analisado com o uso de *big data* que um aumento de 100 milissegundos no tempo de carregamento da página custava 1% do total de vendas (GIGASPACES, 2017). Outro exemplo são os experimentos realizados pelo Google, que concluíram que um tempo de carregamento meio segundo maior diminuía em 20% o número de acessos (GLINDEN, 2017). Pode-se inferir então que, caso a rede não fosse idealmente neutra, quem tivesse controle sobre as prioridades dela poderia afetar o comportamento dos usuários, e portanto a competição e a inovação.

Segundo Tim Berners-Lee em CSAIL, 2017: “Se eu pago para me conectar à rede com uma certa qualidade de serviço, e você pagar para se conectar com a mesma qualidade de serviço ou superior, nós podemos nos comunicar com nível de qualidade da primeira. Isso é tudo. É responsabilidade dos provedores interoperar

para que isso aconteça. A Neutralidade de Rede NÃO está pedindo por Internet gratuita. A Neutralidade de Rede NÃO está dizendo que alguém não deveria pagar mais por um serviço de qualidade superior. Sempre fizemos isso, e sempre iremos fazer. Existem sugestões de que não precisamos de legislação porque nunca realmente tivemos [Neutralidade de Rede]. Elas são absurdas, porque nós já tivemos neutralidade de rede no passado -- apenas recentemente que as ameaças mais explícitas ocorreram.”.

Os primeiros picos de interesse na questão de Neutralidade de Redes começaram em 2006 (TRENDS GOOGLE, 2017). Mesma época em que Berners-Lee escreveu o artigo citado anteriormente, e provavelmente o motivo de ele ter escrito-o.

Figura 8 - Histórico de interesse pelo tópico de Neutralidade de Rede, de 2004 até 2017.



Fonte: (TRENDS GOOGLE, 2017).

Na Figura 8 podemos ver picos de interesse pelo tópico de NN em 2006, 2009, 2010, 2014, 2015 e 2017. Os valores do gráfico representam o interesse relativo de uma data ao ponto máximo, tal ponto máximo, com interesse relativo 100, é o maior pico observado em 2015.

Em 2006 surgiram as primeiras discussões importantes do tópico quando as empresas AT&T, Comcast, Google e Microsoft, nos Estados Unidos, entraram em uma batalha jurídica para com a regulação da Neutralidade de Rede. De um lado AT&T e Comcast, provedoras de Internet, buscavam o direito de poder cobrar além dos consumidores também os donos de websites, para garantir que o conteúdo do website não tenha menor prioridade na rede. Do outro lado Google e Microsoft defendem a Neutralidade de Rede e a natureza livre da Internet. Neste caso AT&T e Comcast ganharam, e puderam implementar as cobranças. (RECLAIM THE MEDIA, 2017)

Em 2009 e 2010 diversos eventos de menores proporções ocorreram nos Estados Unidos. A provedora de rede Comcast é investigada, faz um acordo e posteriormente é absolvida da acusação de discriminar tráfego do protocolo *BitTorrent* (WSJ, 2017). É criado o "*The Internet Freedom Preservation Act of 2009*" que estabelece alguns pontos da NN que as provedoras de rede devem obedecer (H.R.3458). É publicado Gary S. Becker, Dennis W. Carlton & Hal S. Sider (2010), por Gary Becker, Nobel em economia, com diversos argumentos contra a Neutralidade de Rede que serão explorados adiante.

Em 2014 e 2015 ocorreu uma nova onda de interesse no tópico nos Estados Unidos, após o *FCC* (*Federal Communications Commission*) anunciar regras que iriam completamente contra o espírito da NN. Então *FCC* abre para o público uma

seção de comentários sobre a nova legislação, que acaba recebendo cerca de 4 milhões de respostas, com menos de 1% delas sendo contra a Neutralidade de Rede. O *FCC* volta atrás da sua posição e reclassifica a *Internet* de modo a manter a NN.

Em 2017 os EUA, com um novo presidente do *FCC*, muda novamente a sua posição, e volta atrás da reclassificação de 2015. Uma nova seção de comentários é aberta ao público, e as respostas em maioria, novamente, pró-NN foram ignoradas pelo *FCC*.

Enquanto isso no Brasil, foi criado o Marco Civil da Internet, que além de outros pontos, como liberdade de expressão e privacidade, também visa garantir a preservação da Neutralidade de Rede. Como podemos ver no conteúdo do Artigo 9: “O responsável pela transmissão, comutação ou roteamento tem o dever de tratar de forma isonômica quaisquer pacotes de dados, sem distinção por conteúdo, origem e destino, serviço, terminal ou aplicação.”. (PLANALTO, 2014)

2.4.2 Privacidade, Competição, Censura e a Inspeção Profunda de Pacotes

Como explicado anteriormente, a Inspeção Profunda de Pacotes parece ser uma técnica benéfica, protegendo a rede e seus usuários de ataques externos (SYMANTEC, 2017). Mas a técnica vem sido utilizada de maneiras inesperadas pelas provedoras de Internet, lendo o conteúdo dos pacotes dos usuários, e até mesmo os modificando. (H Abelson, K Ledeen, H Lewis, 2009)

Sobre a questão de privacidade. Os usuários não esperam que as operadoras de rede estejam lendo os dados que estão sendo transmitidos, da

mesma maneira que não esperam que a operadora telefônica esteja escutando as suas ligações. O argumento das operadoras de rede gira em torno de que o tráfego precisa ser lido para que seja priorizado; mas novamente o argumento pode ser espelhado para o exemplo absurdo de que a operadora telefônica escutaria suas ligações para decidir qual deve ser a qualidade delas (H Abelson, K Ledeen, H Lewis, 2009). Além das recentes iniciativas das operadoras de tentar viabilizar a venda das informações de navegação de seus clientes (M Bailey; C Labovitz, 2012).

Sobre a questão de competição. Diversas operadoras de rede oferecem também serviços que possuem concorrentes na Internet, como serviços de telefonia, e televisão. A análise intrusiva realizada com *DPI* permite que as operadoras influciem no tráfego relacionado aos seus concorrentes, onde a simples despriorização de uma classe de pacotes resulta em uma grande vantagem competitiva (H Abelson, K Ledeen, H Lewis, 2009).

Sobre a questão de censura, H Abelson, K Ledeen, H Lewis (2009) trás um exemplo simples disso: em 2007 a operadora de Internet *Verizon*, nos EUA, bloqueou a utilização de mensagens de texto de um grupo pró-aborto pois considerava o assunto controverso. Hoje em dia ainda observamos a utilização da censura de conteúdo realizada por governos com razões políticas, e a interferência de operadoras em tráfego *P2P* (M Bailey; C Labovitz, 2012).

Em H Abelson, K Ledeen, H Lewis (2009) conclui-se então que a Inspeção Profunda de Pacotes deveria ser banida no nível das operadoras de Internet. Como uma das razões é colocado o argumento de que não é possível prever como as tecnologias vão se desenvolver, portanto a interferência, causada pelas práticas

mencionadas acima, pode causar *bottlenecks* inesperados, estaríamos desperdiçando o potencial de tecnologias que deixariam de ser criadas.

3 TRABALHOS RELACIONADOS

Essa seção traz alguns trabalhos que exploram objetivos semelhantes. Serão observados pontos em comum, que também aparecerão posteriormente no desenvolvimento deste trabalho, e explicados os pontos divergentes.

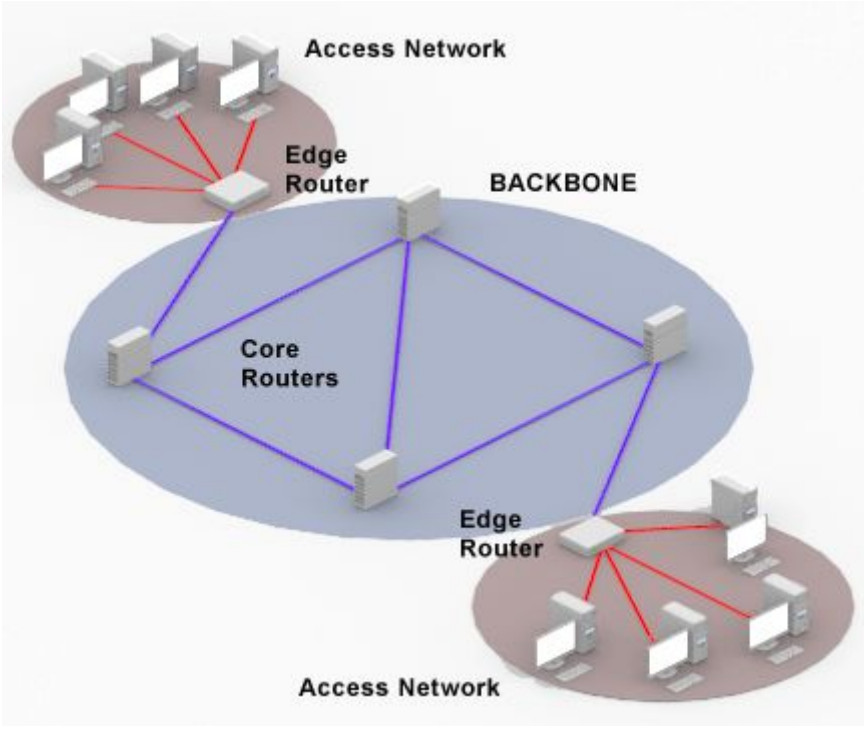
3.1 YouQoS (C Liss; T Fendler; D Gajic; A Vensmer, 2015)

Este trabalho desenvolvido na Universidade de Stuttgart, Alemanha, leva o nome de “YouQoS - Combining Quality of Service with Network Neutrality”(YouQoS - Combinando Qualidade de Serviço com Neutralidade de Rede), e cria um conceito chamado de YouQoS, onde um usuário pode gerenciar o seu tráfego em uma rede externa.

A ideia é que a rede de acesso(*access network*), da operadora, tenha um interface de gerenciamento, que seria o árbitro YouQoS. Um usuário dessa rede poderia prover informações de preferências gerais e de situações específicas, de como o seus dados devem ser tratados, e o árbitro trataria de repassar configurações aos equipamentos da *acess network* da operadora. Portanto o usuário está indiretamente controlando a prioridade de seus próprios fluxos.

A Figura 9 mostra uma simplificação da arquitetura de uma rede, diferenciando entre o *edge* e o *core* dela. No *edge* estão os equipamentos mais próximos dos usuários, são os computadores, celulares e a *access network*. E no *core* está toda a infraestrutura que é transparente ao usuário, diversos roteadores interconectados, conectando à outros pontos de acesso, e outras redes.

Figura 9 - O *edge* e o *core* de uma rede.



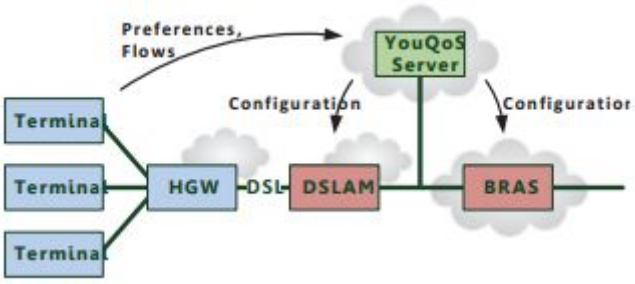
Fonte: C Liss; T Fendler; D Gajic; A Vensmer, 2015.

Segundo C Liss; T Fendler; D Gajic; A Vensmer, 2015 a fronteira entre o *edge* e o *core* está se tornando embaçada, até então os equipamentos da rede local pertenciam ao usuário, mas está se tornando comum serviços que fornecem esses equipamentos, como o *home gateway*(*HGW*). Isso possibilita que a rede local e *access network* tenham configurações compatíveis, que tornaria possível através do YouQoS uma configuração de prioridade de tráfego, que até então era aplicada somente na rede local, seja também aplicada, para o fluxos daquele usuário, na *access network*. Por exemplo, através desse mecanismo.

Esse processo de manejo das configurações pode ser visto na Figura 10. O árbitro YouQoS repassa as preferências do usuário configurando os equipamentos

da rede externa. No caso mostrado, o DSLAM(*Digital Subscriber Line Access Multiplexer*) refere-se ao equipamento multiplexador que recebe a conexão de diversos *home gateways*, e o BRAS(*Broadband Remote Access Server*) seria o *edge router* da *Figura 6* conectando ao núcleo da rede.

Figura 10 - As preferências são passadas para o árbitro.



Fonte: C Liss; T Fendler; D Gajic; A Vensmer, 2015.

3.1.1 Integrando com Redes de Acesso Definidas por Software

Como mencionado anteriormente, um dos benefícios da utilização de Redes de Acesso Definidas por Software é a possibilidade de uma fácil integração com outros serviços. Um exemplo disso é a proposta que o YouQoS de funcionar como um serviço que pode ser integrado ao resto das funcionalidades virtualizadas que estariam rodando na nuvem. Também fica mais claro aqui como a Virtualização das Funções da Rede facilitam outra etapa da manutenção, já que você pode tratar aquilo que era feito pelo *DSLAM* e o *BRAS* de maneira homogênea.

3.2 APPLICATION-AWARENESS IN SDN (Ayyub Qazi; Zafar; Lee, 2013)

O trabalho de Ayyub Qazi, Zafar; Lee (2013) apresenta um framework chamado *Atlas*, que através de *Machine Learning* consegue classificar o tráfego da rede, reconhecendo qual aplicação está sendo utilizada. Citando as aplicações críticas dessa técnica seria realizar a cobrança por uso de serviços, segurança e Qualidade de Serviço. Sabendo a qual aplicação os pacotes pertencem podemos priorizá-los conforme necessário.

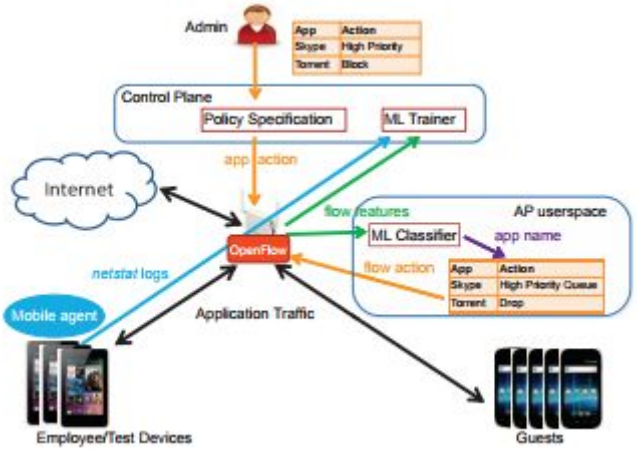
A identificação da aplicação apenas através do cabeçalho ou porta de acesso é muito imprecisa. A utilização de *Deep Packet Inspection*, enquanto fiel requer muito processamento e manutenção da assinatura de cada aplicação, e também o aumento de aplicações que utilizam criptografia fim-a-fim torna isso praticamente impossível.

A utilização de de *Machine Learning*(*ML*) pode utilizar o tamanho dos pacotes de cada fluxo junto com algumas informações do cabeçalho, como endereço e porta de destino, para agrupar os diferentes tráfegos. Mas ainda é difícil de estabelecer controle fino, aplicação a aplicação, já que os tráfegos não são anotados a qual aplicação pertencem. Isto é, conseguimos diferenciar as aplicações umas das outras, mas sem identificá-las, como em “esse fluxo pertence ao aplicativo/programa X”.

Com o aumento do número de aplicações torna-se difícil que a identificação dos fluxos seja feita de maneira manual. Portanto o framework *Atlas* se utiliza de *crowd-sourcing*, instalando em alguns equipamento que utilizam a rede uma

aplicação que consegue coletar os dados dos programas que estão rodando. Podemos ver na Figura 11 onde os *Test Devices* possuem neles um agente que envia logs do uso de rede de cada aplicação rodando no equipamento, que são então correlacionados pelo *ML Trainer* as informações de cada fluxo observado na rede.

Figura 11 - O *framework Atlas* em funcionamento.



Fonte: K J Kerpez, J M Cioffi, G Ginis, M Goldburg, S Galli, P Silverman (2014).

Com isso é obtido o relacionamento da identificação, nome do aplicativo, com as características necessárias para realizar a classificação futura de cada fluxo. Possibilitando que o administrador da rede realize então o papel de priorizar um fluxo sobre outro.

3.2 MININET

O Mininet é um sistema que permite rápida prototipação e simulação de Redes Definidas por Software, com facilidades para a criação de propostas de controladores de rede. Permitindo a criações que podem ser facilmente transportadas para o ambiente de produção. (GITHUB, 2017)

Conforme descrito pelo projeto Mininet (GITHUB, 2017), os equipamentos virtuais, *hosts*, *switches*, *links*, e controladores são reais, apenas implementados em software, e não hardware. Portanto a performance dos equipamentos está limitada ao computador que está realizando a simulação. Portanto é necessário manter em mente durante as simulações para utilizar *links* mais lentos que mas que ainda possam atingir as propriedades visadas pelo simulação, ou seja, utilizando links de 100Mbps e não 100Gbps.

Cada host virtual criado pelo Mininet pode executar programas, ler e escrever arquivos, etc. Isso torna as possibilidades de experimentos muito amplas. (GITHUB, 2017)

O Miniedit, também desenvolvido pelo próprio projeto Mininet, é uma interface gráfica simples que permite a prototipação da arquitetura de uma rede, conectando hosts com switches e switches com controladores. Permite gerar código que compatível com o Mininet para ainda maior comodidade de simulação. (GITHUB, 2017)

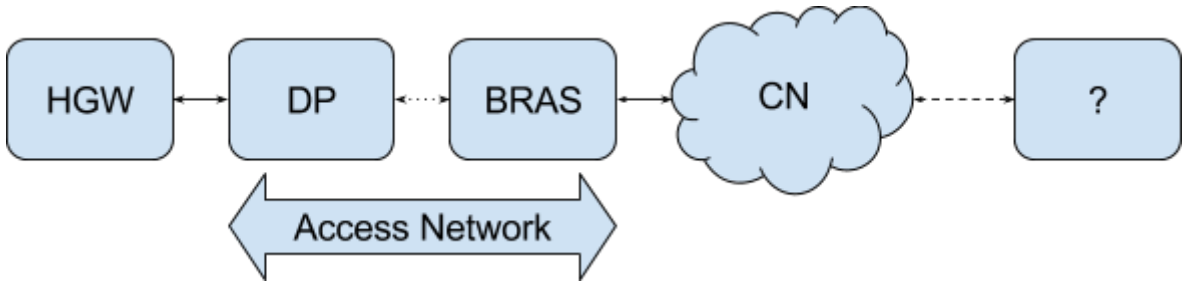
4 DESENVOLVIMENTO

Essa seção traz as propostas de soluções, e o processo de criação delas, visando compatibilizar as o conceito de Neutralidade de Rede com a Qualidade de Serviço em Redes de Acesso Definidas por Software. Começamos as propostas de soluções das mais simples, incrementando em complexidade e mecanismos conforme observamos as problemáticas vistas na fundamentação e trabalhos relacionados.

4.1 OS ELEMENTOS DA REDE

Nos diversos trabalhos vistos até agora há uma grande variedade de nomes e siglas dados a elementos da rede que possuem a mesma funcionalidade. Aqui é estabelecido a nomenclatura que será utilizada, e como eles serão representados nas propostas. Na Figura 12 podemos observar esses elementos, e a seguir temos definição de cada um deles.

Figura 12 - Representação dos elementos da rede.



Fonte: desenvolvido pelo autor.

HGW: *Home Gateway*. Anteriormente visto também como *Terminal* ou *Terminal equipment*. É o equipamento de rede que pertence ao usuário/cliente e o conecta à rede da operadora. O usuário teoricamente tem completo acesso ao equipamento e suas configurações. Seria o *modem*/ponto de acesso na casa do cliente.

DP: *Distribution Point*. Anteriormente também visto como *DSLAM* (*Digital Subscriber Line Access Multiplexer*). É o equipamento da operadora que se conecta aos equipamentos dos diversos usuário de uma região pequena, como algumas ruas. São conhecidos popularmente como “armários”.

BRAS: *Broadband Remote Access Server*. Conecta a rede da operadora, de uma grande região, com o núcleo da rede.

Access Network ou Rede de Acesso. Compreendendo a infraestrutura local de uma operadora, possui diversos equipamentos intermediários entre o ponto de entrada (DP) e de saída (BRAS). Aqui os nodos são abstraídos visualmente, mas compreendem uma parte importante na questão de priorização dos pacotes, pois podem consistir de diversos *hops.*

CN: *Core Network* ou Núcleo da Rede. São os *backbones*, de alta performance*,* que interconectam as diversas rede locais das operadoras criando a infraestrutura da Internet.

Além disso na Figura 12 temos uma caixa vazia, conectada por uma linha tracejada à rede, representando algum outro cliente ou serviço que também está conectado à Internet, e portanto similarmente possui uma série de equipamentos (HGW, DP, BRAS) que estão abstraídos.

4.2 *FAIRNESS*

Iremos observar que nas soluções muitas vezes não temos como ter certeza de que os pacotes estão da melhor maneira possível, “justa”, *fair*. Classificamos então o nível de certeza de “justiça”, *fairness*, que podemos obter em cada solução em três níveis: Confiar, Monitorar e Forçar.

4.2.1 Confiar

Podemos apenas confiar que operadora esteja tratando os nossos pacotes da melhor maneira possível. Quando possibilitamos que a própria operadora realize a classificação de prioridade. Similar aquilo que é feito atualmente.

Podemos apenas comparar a performance obtida com a performance contratada. Não conseguimos saber quais foram as prioridades utilizadas, impossibilitando que conheçamos problemas de performance pois não sabemos a causa das flutuações.

4.2.2 Monitorar

Estabelecemos a prioridade relativa de nossos pacotes, repassando a decisão para a rede da operadora. Esperamos que as prioridades relativas sejam respeitadas.

Conseguimos descobrir caso a operadora esteja interferindo com o tráfego pois as obtemos performance diferente daquela priorizada, comparando o tráfego de diferentes conteúdos.

4.2.3 Forçar

As prioridades atribuídas aos nossos pacotes são utilizadas na rede da operadora, mas não estão sob controle dela. Existe alguma autoridade confiável que regula o repasse de prioridades para os equipamentos. A operadora ainda possui controle relativo sobre o funcionamento dos equipamentos, mas as mudanças estão sob supervisão.

4.3 AS SOLUÇÕES INGÊNUAS

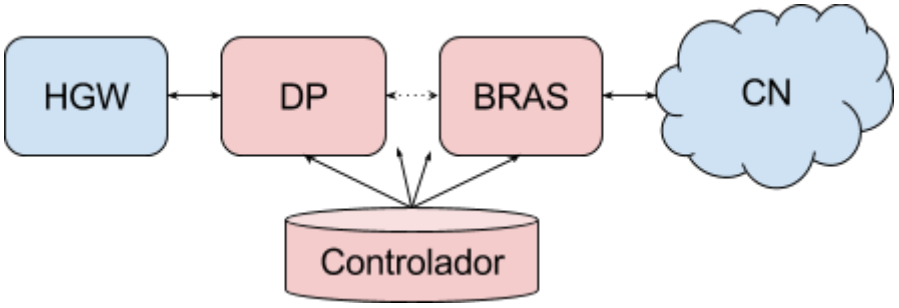
Algumas soluções mais simples facilmente caem em armadilhas que foram vistas na fundamentação e trabalhos relacionados. Mesmo assim são interessantes pois são tomadas como base para entender as soluções mais complexas apresentadas posteriormente..

4.3.1 A Rede de Acesso com equipamentos SDN

Imaginando que substituíssemos os equipamentos da Rede de Acesso por nodos que possibilitem o uso de Redes Definidas por Software, isso nos permite que introduzíssemos a ideia de inspeção de prioridade desde o primeiro nodo da

rede da operadora, o *Distribution Point*. Na Figura 13 temos como isso seria representado.

Figura 13 - A rede de acesso com equipamentos SDN.



Fonte: desenvolvido pelo autor.

Os novos fluxos primeiramente são desconhecidos e portanto precisam de que o Controlador da rede os classifique. É colocado então na *flowtable* de cada de cada equipamento o encaminhamento que os fluxos devem tomar, e priorização que deve ser dada. Necessário ressaltar que o *Distribution Point* e *BRAS* possuem diversos outros equipamentos entre eles, a troca de todos esses equipamentos por novos capazes de trabalhar com os protocolos de SDN.

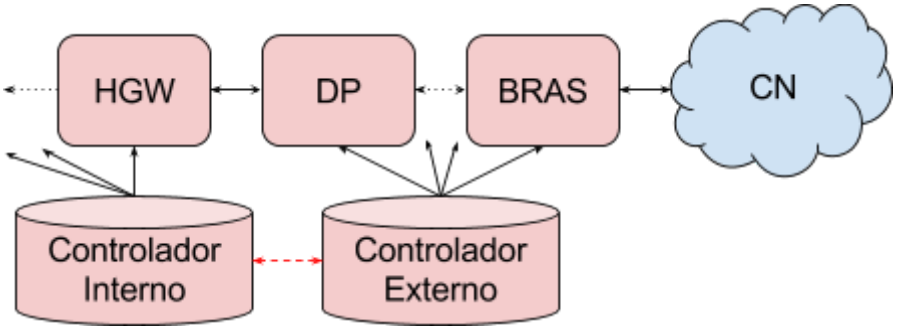
Caímos então sobre o problema de que a Rede de Acesso, sob controle da operadora, precisa realizar a priorização dos pacotes, abrindo a possibilidade de Inspeção Profunda de Pacotes e nos coloca sob a classificação de Confiar no *fairness* da priorização.

Outro problema é que todos os computadores da rede interna do usuário tem a mesma prioridade na rede externa, isso pode parecer bom à primeira vista, mas diminui os cenários de configuração possível.

4.3.2 Marcação de prioridade no *Home Gateway*

Realizamos então a troca do equipamento do usuário por um de Redes Definidas por Software. Passamos a inspeção de prioridade para equipamento do cliente, e resolvemos parte do problema com Inspeção Profunda de Pacotes (DPI). Afinal, não existe problema em utilizarmos a técnica sobre os nosso próprios pacotes, estaríamos violando apenas a nossa própria privacidade. Atente de que o DPI não é completamente resolvido, já que a operadora ainda pode ler o conteúdo dos pacotes, mas não deveria ter motivos para fazer isso.

Figura 14 - Marcação dos pacotes no *Home Gateway.*



Fonte: desenvolvido pelo autor.

Na Figura 14 podemos observar que um novo controlador é introduzido. Podemos aproveitar a oportunidade e substituir também os outros equipamentos do usuário, já que estaremos marcando os pacotes com a priorização que eles devem ter na rede externa, podemos nos utilizar de tal priorização também na rede interna.

Precisamos então que as prioridades definidas pelo controlador interno sejam utilizadas pelo controlador externo. Para isso é necessário que algum padrão seja seguido, abstraímos o formato dessa comunicação; mas seria ou alguma maneira

indireta de comunicação dos dois controladores; ou a marcação de algum cabeçalho no pacote, que ao entrar na Rede de Acesso teria a informação informação necessário para o controlador externo (similar ao *MPLS*).

Essa abordagem está sujeita a um novo problema, onde um cliente mal-intencionado marca que todos os seus pacotes devem ter prioridade máxima. Desfavorecendo os pacotes de outros usuários.

Também não podemos contar com que a rede da operadora irá obedecer as prioridades repassadas. Mas podemos obter um nível de *fairness* Monitorar, já que sabemos quais deveriam ser as prioridades dentre os nosso próprios pacotes, podemos comparar a performance atingida por cada fluxo.

4.4 SOLUÇÕES RELEVANTES

Soluções que poderiam realmente ser implementadas e utilizadas sem maiores problemas. Utilizam dos diversos métodos vistos na fundamentação teórica e trabalhos relacionados para obtenção dos melhores resultados.

4.4.1 Normalização das prioridades na Rede de Acesso

Resolvemos um dos problemas encontrados na seção 4.3.2 utilizando uma função para normalizar as prioridades na Rede de Acesso que foram originalmente recebidas dos clientes. Isto é, resolvemos o caso um cliente malicioso que esteja enviando diversos pacotes como de alta prioridade desfavorecer os outros usuários.

Para tal normalização, podemos pensar em termos da quantidade de “utilidade” que é obtida por cada usuário. Podemos estabelecer que utilidade máxima que um usuário pode obter em um certo momento seja relativo à quantidade de banda contratada. E para cada nível de prioridade, mais utilidade é utilizada, através de um multiplicador. Lembrando devemos permitir que o usuário possa utilizar toda sua banda contratada, logo, quando o usuário ultrapassa o máximo de sua utilidade, apenas multiplicamos as prioridades por uma fator que corrige a utilidade, diminuindo as prioridades.

Utilizando as classes de tráfego estabelecidas em 802.1Q, na Tabela 2 temos um exemplo simples de como poderia ser feita a multiplicação da utilidade do usuário, que poderia ser facilmente substituída ou extendida. Utilizando esses valores segue um exemplo demonstrando como a normalização poderia ocorrer: considerando que o usuário tem uma banda contratada de 10Mbps, e ele está utilizando no em um certo momento 1Mbps priorizada como *Voice*, sua utilidade utilizada é de 5Mbps, como descrito na tabela 2; em dado um momento o usuário começa a utilizar também uma aplicação que utiliza 5Mbps priorizada como *Best effort*, sua utilidade sendo utilizada cresce então para 10Mbps; em um próximo momento o usuário utiliza também uma terceira aplicação que utiliza 5Mpbs priorizada como *Background*, o uso de utilidade elevaria para 15Mbps, então um fator de normalização é calculado (10mpbs/15mpbs = 0.66) para diminuir a prioridade de cada um dos tráfegos.

Apesar de um dos problemas ter sido resolvido, a característica de *fairness* Monitorar é mantida.

Tabela 2 - Exemplo de multiplicadores de utilidade.

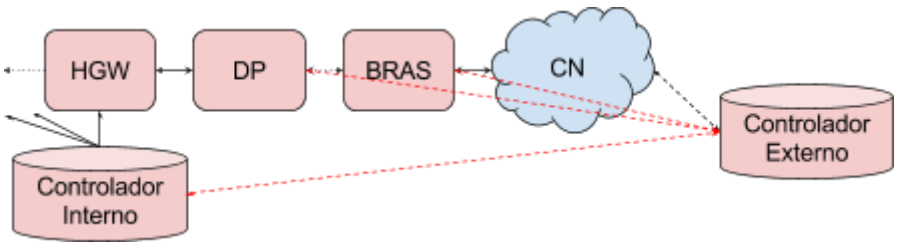
Multiplicador de utilidade	**Tipo de tráfego**
1	*Background*
1	*Best effort*
2	*Excellent effort*
3	*Critical Applications*
4	*Video,* menos de 100ms de latência
5	*Voice,* menos de 10ms de latência

Fonte: 802.1Q.

4.4.2 Utilizando Redes de Acesso Definidas por Software

Como visto na seção 2.3.7 o conceito de *Software Defined Access Networks* utiliza-se também das Funções Virtualizadas de Rede (NFV). Na Figura 15 podemos como o controlador da rede poderia se tornar um serviço externo. Onde a operadora pode encontrar maior facilidade em operar diversas Redes de Acesso.

Figura 15 - A rede de acesso virtualizada.



Fonte: desenvolvido pelo autor.

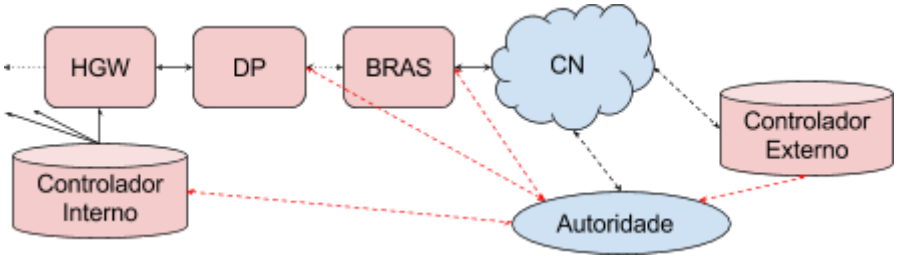
Não resolvemos nenhum problema do ponto de vista do usuário, apenas facilitamos a operação da provedora de rede, e essa virtualização se torna um trampolim necessário para a próxima solução. Novamente o *fairness* é Monitorar.

4.4.3 Supervisão por uma autoridade

Estabelecendo que a supervisão por uma autoridade é necessária para garantir que as preferências estabelecidas sejam cumpridas sempre que possível. Abstraímos qualquer tipo de falha proveniente da autoridade, sendo ela sempre confiável. Um exemplo de autoridade que poderia ser estabelecido é utilização de uma entidade governamental, como a Anatel no Brasil ou o FCC nos EUA, que já são responsáveis por realizar auditoria sobre a operadoras de Internet.

Como pode ser visto na Figura 16, a autoridade funciona como um outro serviço externo. Enviando a prioridades para a Autoridade, ela funciona como um serviço que acrescenta às funções de virtualização geradas pelo controlador da operadora as características de prioridade estabelecidas pelos usuários.

Figura 16



Fonte: desenvolvido pelo autor.

Conseguimos atingir nessa proposta o objetivo de *fairness* Forçar.

4.5 COMO TESTAR UMA SOLUÇÃO

Utilizando o Mininet podemos facilmente simular algumas arquiteturas de redes simples muito facilmente, assim como controladores e seus funcionamentos. Mas a introdução de elementos como núcleo da rede, a Autoridade e a Virtualização de funções da rede infelizmente ainda são bastante complexas.

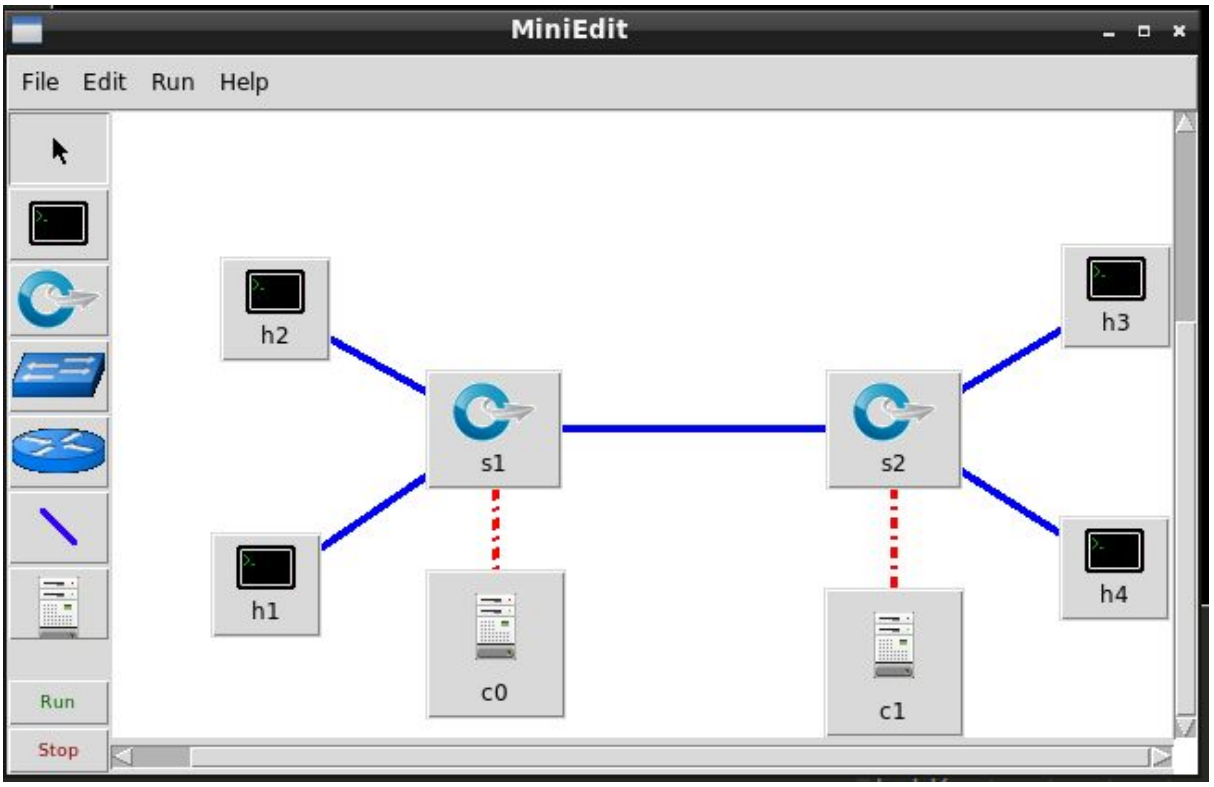
Aqui então é proposto uma maneira de obter as todas as métricas da rede que poderiam ser utilizadas para a comparação de soluções, mas as soluções não puderam ser implementadas para uma comparação quantitativa.

4.5.1 Realizando uma simulação

Começamos a simulação de um cenário criando a arquitetura da rede, dependendo da complexidade dela podemos utilizar o Miniedit, que vem junto do código do projeto Mininet, para desenhar a nossa arquitetura. Na Figura 17, podemos ver o exemplo de uma arquitetura simples, com 4 *hosts*, e 2 *switches*, cada um com um controlador SDN, onde cada *link* tem banda de 10mpbs.

Gerando o código do a partir do Miniedit, temos o código que está pode ser visto no Apêndice 7.1, exceto pelo segmento delimitado como *CUSTOM EXPERIMENT*.

Figura 17



Fonte: desenvolvido pelo autor.

Escrevemos então o nosso experimento, conhecendo os equipamento da nossa arquitetura. Um exemplo simples disso é então o segmento *CUSTOM EXPERIMENT* do Apêndice 7.1. Onde abrimos um servidor do serviço *iperf* de mensuração de taxa de transmissão, nos hosts *h3* e *h4*, que podem ser observado à direita da nossa arquitetura na Figura 17. Iniciamos o serviço *bwm-ng* como superusuário da máquina que está rodando as simulações, deste modo capturamos um log, de todas as métricas em todas os links virtuais que serão criados, segundo-a-segundo. Começamos a transmissão de 100Mb de dados, do *host h1* para o *host h3*, que caso não houvesse interferência levaria cerca de 80 segundos para serem transmitidos, pelos links de 10Mbps. E 15 segundos depois, começamos

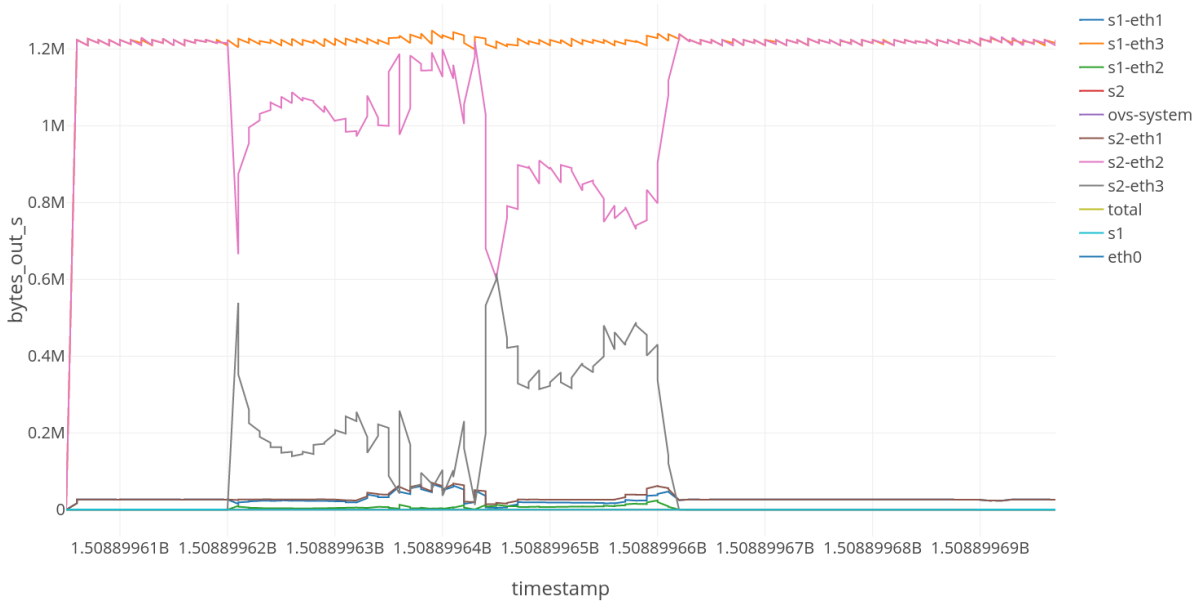
uma transferência de 10Mb do *host h2* para o *host h4*. Quando as duas transferências terminam, desligamos o serviço de logs de métricas, e os servidores de do serviço *iperf*.

4.5.2 Observando os resultados

O arquivo de log gerado pelo *bwm-ng* possui todas as métricas que foram capturadas, portanto todos os dados que poderíamos querer utilizar para comparar as soluções.

Utilizando o código no Apêndice 7.2 convertemos o arquivo de logs para um arquivo *CSV* que pode ser utilizado a criação de gráficos, como o da Figura 18, representando velocidade de bytes por segundo saindo de cada *link* da rede durante o experimento. Podemos observar no centro do gráfico quando as transmissões concorrentes acontecem.

Figura 18



Fonte: desenvolvido pelo autor.

5 CONCLUSÕES

Analisando o objetivo geral do trabalho, podemos concluir que ele foi atingido. Pudemos propor diversas soluções e compará-las de maneira qualitativa. Compatibilizando, com um porém, a Neutralidade de Rede com a Qualidade de Serviço nas Redes Definidas por Software. O porém é que ainda realizamos a priorização de fluxos, mas essa pode ser realizada pela parte interessada, o usuário.

Dentro dos objetivos específicos. Não foi possível comparar as soluções quantitativamente devido a complexidade de que elas atingiram. Mas foi um elaborado método através do qual seria possível, caso as soluções fossem arquitetadas utilizando o simulador Mininet. Nem comparar quantitativamente com as soluções existentes, mas foi possível observar as diferenças das características qualitativas.

As soluções propostas na seção 4.4 seria aplicáveis no mundo real. Mas como o resto das soluções, são caras, requerem a troca de muitos equipamentos de rede antigos. Futuramente, quando os equipamentos antigos forem naturalmente substituídos seria interessante validar o quão aplicável as soluções são.

6 REFERÊNCIAS BIBLIOGRÁFICAS

RFC7426. *Software-Defined Networking (SDN): Layers and Architecture Terminology* - https://tools.ietf.org/html/rfc7426.

ITU E.800 09/08: *Definitions of terms related to quality of service* - http://www.itu.int/rec/T-REC-E.800-200809-I/en.

W Kim, P Sharma, J Lee, S Banerjee, J Tourrilhes, S Lee, and P Yalagandula (2010). *Automated and Scalable QoS Control for Network Convergence* - Princeton University.

Jon Crowcroft (2007). *Net neutrality: the technical side of the debate: a white paper*. University of Cambridge, Cambridge, UK. Volume 37 Issue 1, January 2007.

Hsing Kenneth Cheng, Subhajyoti Bandyopadhyay, Hong Guo, (2011) *The Debate on Net Neutrality: A Policy Perspective*. Information Systems Research 22(1):60-82. http://dx.doi.org/10.1287/isre.1090.0257.

TECHTARGET. *It'll take NFV and SDN to kill Net Neutrality and the open Internet.* Disponível em: <http://searchsdn.techtarget.com/opinion/Itll-take-NFV-and-SDN-to-kill-Net-Neutrality-and-the-open-Internet>. Acesso em: 12/10/2017.

A Greenberg, G Hjalmtysson, D. A Maltz, A Myers, J Rexford, G Xie, H Yan, J Zhan, H Zhang (2005). *A Clean Slate 4D Approach to Network Control and Management*. ACM SIGCOMM Computer Communication Review.

N McKeown, T Anderson, H Balakrishnan, G Parulkar, L Peterson, J Rexford, S Shenker, J Turner (2008). OpenFlow: *Enabling Innovation in Campus Networks*. ACM SIGCOMM Computer Communication Review archive.

WIKIPEDIA. *SDN*. Disponível em <https://en.wikipedia.org/wiki/Software-defined_networking>. Acesso em: 14/09/2017.

SDXCENTRAL. *Does SDN Make My Network Management Look Fat?*. Disponível em <https://www.sdxcentral.com/articles/contributed/does-sdn-make-my-network-management-look-fat/2012/09>. Acesso em: 16/09/2017.

WEBWERKS. *Southbound vs. Northbound SDN: What are the differences?*. Disponível em: <http://blog.webwerks.in/data-centers-blog/southbound-vs-northbound-sdn-what-are-the-differences>. Acesso em: 18/09/2017.

OPEN NETWORKING (2013). *SDN Architecture Overview*. Disponível em: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/SDN-architecture-overview-1.0.pdf>.

W. Zhou, L. Li, M. Luo, and Wu Chou (2014). *REST API design patterns for SDN Northbound API*. in Proc. 28th IEEE Int. Conf. Adv. Inf. Netw. Appl. Workshops (AINA'14), Victoria, BC, Canada, May 13–16, 2014, pp. 358–365.

CISCO. *OpFlex: An Open Policy Protocol White Paper*. Disponível em: <http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-731302.htm>. Acesso em: 20/10/2017.

OPENDAYLIGHT. *OpenDaylight Project List*. Disponível em: <https://wiki.opendaylight.org/view/Project_list>. Acesso em: 24/09/2017.

GITHUB. *Mininet FAQ*. Disponível em <https://github.com/mininet/mininet/wiki/FAQ>. Acesso em: 12/10/2017.

OPENFLOW. *OpenFlow Learn more*. Disponível em <http://archive.openflow.org/wp/learnmore/>. Acesso em 18/10/2017.

NETLAB. *IP Quality of Service*. Disponível em: <https://www.netlab.tkk.fi/~puhuri/htyo/Tik-110.551/iwork/iwork.html>. Acesso em: 26/09/2017.

D.C. Verma (2004). *Service level agreements on IP networks*. Proceedings of the IEEE (Volume: 92, Issue: 9, Sept. 2004). 10.1109/JPROC.2004.832969.

K Brunnström, S Beker, K De Moor (2013). *Qualinet White Paper on Definitions of Quality of Experience*. Output from the fifth Qualinet meeting, Novi Sad.

Tim Wu (2003). *NETWORK NEUTRALITY, BROADBAND DISCRIMINATION*. Disponível em: <http://www.jthtl.org/content/articles/V2I1/JTHTLv2i1_Wu.PDF>.

CSAIL. Net Neutrality: This is serious. Disponível em: <http://dig.csail.mit.edu/breadcrumbs/node/144>. Acesso em: 27/03/2017.

TRENDS GOOGLE. *Google Trends Neutralidade de Rede*. Disponível em <https://trends.google.com/trends/explore?date=all&q=%2Fm%2F04zdxn>. Acesso em: 10/10/2017.

RECLAIM THE MEDIA. *AT&T, Comcast Rout Google, Microsoft in `Net Neutrality' Battle*. Disponível em: <http://www.reclaimthemedia.org/broadband_cable/at_t_comcast_rout_google_microsoft_in_net_neutrality_battle.html> Acesso em: 18/10/2017.

STANFORD. *Make Data Useful*. Disponível em: <https://www.gduchamp.com/media/StanfordDataMining.2006-11-28.pdf>. Acesso em: 20/09/2017.

GIGASPACES. *Amazon found every 100ms of latency cost them 1% in sales*. Disonível em: <https://blog.gigaspaces.com/amazon-found-every-100ms-of-latency-cost-them-1-in-sales/>. Acesso em: 24/09/2017.

GLINDEN. *Marissa Mayer at Web 2.0*. Disponível em: <http://glinden.blogspot.com.br/2006/11/marissa-mayer-at-web-20.html>. Acesso em: 12/08/2017.

WSJ. *Court Backs Comcast Over FCC on 'Net Neutrality'*. Disponível em: <https://www.wsj.com/articles/SB10001424052702303411604575167782845712768>. Acesso em: 28/08/2017.

H.R.3458 - *Internet Freedom Preservation Act of 2009*. Disponível em: <https://www.congress.gov/bill/111th-congress/house-bill/3458>. Acesso em: 24/10/2017.

Gary S. Becker, Dennis W. Carlton & Hal S. Sider (2010). *NET NEUTRALITY AND CONSUMER WELFARE*. Published by Oxford University Press.

PLANALTO. *LEI Nº 12.965. Marco Civil da Internet*. Disponível em: <http://www.planalto.gov.br/CCIVIL_03/_Ato2011-2014/2014/Lei/L12965.htm>. Acesso em: 02/08/2017.

Christian Liss, Tamas Fendler, Danica Gajic & Alexander Vensmer (2015). *YouQoS - Combining Quality of Service with Network Neutrality*. 978-3-8007-3925-7.

Hassan Habibi Gharakheili, Arun Vishwanath, Vijay Sivaraman (2016). *Perspectives on Net Neutrality and Internet Fast-Lanes*. University of New South Wales, IBM Research-Australia.

Rogerio Leao Santos de Oliveira, Ailton Akira Shinoda, Christiane Marie Schweitzer (2014). *Using Mininet for emulation and prototyping Software-Defined Networks*. 10.1109/ColComCon.2014.6860404.

Hilmi E. Egilmez, Seyhan Civanlar & A. Murat Tekalp (2013). *An Optimization Framework for QoS-Enabled Adaptive Video Streaming Over OpenFlow Networks*. IEEE Transactions on Multimedia. 10.1109/TMM.2012.2232645.

ACLU. *Congress: Don't Let Internet Providers Sell Our Data to the Highest Bidder*. Disponível em: <https://www.aclu.org/blog/privacy-technology/internet-privacy/congress-dont-let-internet-providers-sell-our-data-highest>. Acesso em: 14/10/2017.

Michael Bailey, Craig Labovitz (2012). *Censorship and Co-option of the Internet Infrastructure*.

RFC1122. *Requirements for Internet Hosts - Communication Layers*. Disponível em: <https://tools.ietf.org/html/rfc1122>. Acesso em: 24/10/2017.

Murali Kodialam T. V. Lakshman Sudipta Sengupta (2006). *Efficient and Robust Routing of Highly Variable Traffic*. Doctoral Dissertation.

RFC 2205. *Resource ReSerVation Protocol (RSVP)*. Disponível em: <https://tools.ietf.org/html/rfc2205>. Acesso em: 21/10/2017.

RFC 3168. *Historical Definitions for the IPv4 TOS Octet*. Disponível em: <https://tools.ietf.org/html/rfc3168#section-22>. Acesso em: 12/07/2017.

RFC791. *INTERNET PROTOCOL*. Disponível em: <https://tools.ietf.org/html/rfc791>. Acesso em: 23/09/2017.

RFC2474. *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*. Disponível em: <https://tools.ietf.org/html/rfc2474>. Acesso em: 30/09/2017.

RFC2460. *Internet Protocol, Version 6 (IPv6) Specification*. Disponível em: <https://tools.ietf.org/html/rfc2460>. Acesso em: 14/10/2017.

802.1Q. *LOCAL AND METROPOLITAN AREA NETWORK STANDARDS*. Disponível em: <http://standards.ieee.org/getieee802/download/802.1Q-2005.pdf>. Acesso em: 07/10/2017.

Hassan Hawilo, Abdallah Shami, Maysam Mirahmadi, and Rasool Asal (2014). *NFV: State of the Art, Challenges, and Implementation in Next Generation Mobile Networks (vEPC)*. 0890-8044/14/ © 2014 IEEE.

Kenneth J. Kerpez, John M. Cioffi, George Ginis, Marc Goldburg, Stefano Galli, and Peter Silverman (2014). *Software-Defined Access Networks*. 0163-6804/14/ © 2014 IEEE.

Ayyub Qazi, Zafar & Lee, Jeongkeun & Jin, Tao & Bellala, Gowtham & Arndt, Manfred & Noubir, Guevara. (2013). *Application-awareness in SDN*. ACM SIGCOMM Computer Communication Review. 43. 487-488. 10.1145/2486001.2491700.

SYMANTEC. *The Perils of Deep Packet Inspection*. Disponível em: <https://www.symantec.com/connect/articles/perils-deep-packet-inspection>. Acesso em: 23/10/2017.

Hal Abelson Ken Ledeen, Harry Lewis (2009). *Just Deliver the Packets*. Disponível em: <https://www.priv.gc.ca/en/opc-actions-and-decisions/research/explore-privacy-research/2009/ledeen-lewis_200903>. Acesso em: 23/10/2017.

GITHUB. *Introduction to Mininet*. Disponível em: <https://github.com/mininet/mininet/wiki/Introduction-to-Mininet>. Acesso em: 08/08/2017.

APÊNDICE A

A.1 UM EXPERIMENTO NO MININET

```
#!/usr/bin/python

import time
import subprocess

from mininet.net import Mininet
from mininet.node import Controller, RemoteController, OVSController
from mininet.node import CPULimitedHost, Host, Node
from mininet.node import OVSKernelSwitch, UserSwitch
from mininet.node import IVSSwitch
from mininet.cli import CLI
from mininet.log import setLogLevel, info
from mininet.link import TCLink, Intf
from subprocess import call

def myNetwork():

    net = Mininet( topo=None,
                   build=False,
                   ipBase='10.0.0.0/8')

    info( '*** Adding controller\n' )
    c0=net.addController(name='c0',
                         controller=Controller,
                         protocol='tcp',
                         port=6633)

    c1=net.addController(name='c1',
                         controller=Controller,
                         protocol='tcp',
                         port=6634)

    info( '*** Add switches\n')
    s2 = net.addSwitch('s2', cls=OVSKernelSwitch)
    s1 = net.addSwitch('s1', cls=OVSKernelSwitch)

    info( '*** Add hosts\n')
    h4 = net.addHost('h4', cls=Host, ip='10.0.0.4', defaultRoute=None)
    h2 = net.addHost('h2', cls=Host, ip='10.0.0.2', defaultRoute=None)
    h1 = net.addHost('h1', cls=Host, ip='10.0.0.1', defaultRoute=None)
    h3 = net.addHost('h3', cls=Host, ip='10.0.0.3', defaultRoute=None)

    info( '*** Add links\n')
    h1s1 = {'bw':10,'delay':'10'}
    net.addLink(h1, s1, cls=TCLink , **h1s1)
    h2s1 = {'bw':10,'delay':'10'}
    net.addLink(h2, s1, cls=TCLink , **h2s1)
```

```
    s1s2 = {'bw':10,'delay':'10'}
    net.addLink(s1, s2, cls=TCLink , **s1s2)
    s2h3 = {'bw':10,'delay':'10'}
    net.addLink(s2, h3, cls=TCLink , **s2h3)
    s2h4 = {'bw':10,'delay':'10'}
    net.addLink(s2, h4, cls=TCLink , **s2h4)

    info( '*** Starting network\n')
    net.build()
    info( '*** Starting controllers\n')
    for controller in net.controllers:
        controller.start()

    info( '*** Starting switches\n')
    net.get('s2').start([c1])
    net.get('s1').start([c0])

    info( '*** Post configure switches and hosts\n')

    ## START CUSTOM EXPERIMENT ##

    bwm_ng_file = open("bwm-ng.log", "w")

    h3.cmd('iperf -s &')
    h4.cmd('iperf -s &')

    bwm_ng = subprocess.Popen(['bwm-ng','-o','csv','-I','%lo'],
stdout=bwm_ng_file)

    h1.sendCmd('iperf -n 100M -c %s' % h3.IP())

    time.sleep(15)

    h2.sendCmd('iperf -n 10M -c %s' % h4.IP())

    h2.waitOutput()
    h1.waitOutput()

    time.sleep(5)

    bwm_ng.kill()

    h3.cmd('pkill iperf')
    h4.cmd('pkill iperf')

    ## END CUSTOM EXPERIMENT ##

    # CLI(net)
    net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' )
    myNetwork()
```

A.2 OBSERVANDO RESULTADOS DE UM EXPERIMENTO

```
import matplotlib.pyplot as plt
import csv

IGNORE = ['ovs-system', 'eth0', 'total']

bwm_log = open("bwm-ng.log", "r")

ifaces = {}

class InterfaceData:
      def __init__(self, name):
            self.name = name
            self.timestamp = []

            self.bytes_out_s = []
            self.bytes_in_s = []
            self.bytes_total_s = []
            self.bytes_in = []
            self.bytes_out = []

            self.packets_out_s = []
            self.packets_in_s = []
            self.packets_total_s = []
            self.packets_in = []
            self.packets_out = []

            self.errors_out_s = []
            self.errors_in_s = []
            self.errors_in = []
            self.errors_out = []

      def append(self, *args):
            self.timestamp.append(args[0])

            self.bytes_out_s.append(args[1])
            self.bytes_in_s.append(args[2])
            self.bytes_total_s.append(args[3])
            self.bytes_in.append(args[4])
            self.bytes_out.append(args[5])

            self.packets_out_s.append(args[6])
            self.packets_in_s.append(args[7])
            self.packets_total_s.append(args[8])
            self.packets_in.append(args[9])
            self.packets_out.append(args[10])

            self.errors_out_s.append(args[11])
            self.errors_in_s.append(args[12])
            self.errors_in.append(args[13])
            self.errors_out.append(args[14])
```

```
	def to_rows(self):
		rows = []
		for i in range(len(self.timestamp)):
			row = [self.name, self.timestamp[i]]

			row.append(self.bytes_out_s[i])
			row.append(self.bytes_in_s[i])
			row.append(self.bytes_total_s[i])
			row.append(self.bytes_in[i])
			row.append(self.bytes_out[i])

			row.append(self.packets_out_s[i])
			row.append(self.packets_in_s[i])
			row.append(self.packets_total_s[i])
			row.append(self.packets_in[i])
			row.append(self.packets_out[i])

			row.append(self.errors_out_s[i])
			row.append(self.errors_in_s[i])
			row.append(self.errors_in[i])
			row.append(self.errors_out[i])

			rows.append(row)

		return rows

#timestamp;iface_name;
#bytes_out/s;bytes_in/s;bytes_total/s;bytes_in;bytes_out;
#packets_out/s;packets_in/s;packets_total/s;packets_in;packets_out;
#errors_out/s;errors_in/s;errors_in;errors_out
for row in csv.reader(bwm_log, delimiter=";"):
	# print(row)
	try:
		timestamp = int(row[0])
		iface_name = row[1]

		bytes_out_s = float(row[2])
		bytes_in_s = float(row[3])
		bytes_total_s = float(row[4])
		bytes_in = int(row[5])
		bytes_out = int(row[6])

		packets_out_s = float(row[7])
		packets_in_s = float(row[8])
		packets_total_s = float(row[9])
		packets_in = int(row[10])
		packets_out = int(row[11])

		errors_out_s = float(row[12])
		errors_in_s = float(row[13])
		errors_in = int(row[14])
		errors_out = int(row[15])
	except IndexError:
		continue

	iface = ifaces.get(iface_name)
	if not iface:
```

```
            iface = InterfaceData(iface_name)
            ifaces[iface_name] = iface

      iface.append(timestamp,

                        bytes_out_s,
                        bytes_in_s,
                        bytes_total_s,
                        bytes_in,
                        bytes_out,

                        packets_out_s,
                        packets_in_s,
                        packets_total_s,
                        packets_in,
                        packets_out,

                        errors_out_s,
                        errors_in_s,
                        errors_in,
                        errors_out)

bwm_log.close()

for iface in ifaces.values():
      if iface.name in IGNORE:
            continue

      plt.plot(iface.timestamp, iface.bytes_out_s, label=iface.name)

plt.legend()
plt.show()

bwm_csv_f = open('bwm-ng.csv', 'w')

bwm_csv = csv.writer(bwm_csv_f, delimiter=',')

bwm_csv.writerow(['name', 'timestamp',
                        'bytes_out_s', 'bytes_in_s', 'bytes_total_s',
                        'bytes_in', 'bytes_out',
                        'packets_out_s', 'packets_in_s', 'packets_total_s',
                        'packets_in', 'packets_out',
                        'errors_out_s', 'erros_in_s',
                        'errors_in', 'erros_out'])

for iface in ifaces.values():
      for row in iface.to_rows():
            bwm_csv.writerow(row)

bwm_csv_f.close()
```

Qualidade de Serviço de Neutralidade de Rede em Redes de Acesso Definidas por Software

Evandro Sasse[1], **Mario A. R. Dantas** [1]

[1]Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – 88.040-900 – Florianópolis – SC – Brazil

Abstract. *Some solutions to Network Neutrality are going to be presented, for Access Networks controlled by Internet Service Providers. There are problems with Net Neutrality that can be aggravated by using Software Defined Access Networks. Some solutions are proposed to be implemented, using of this technology.*

Resumo. *Serão apresetadas aqui algumas soluções para resolver problemas quanto à Neutralidade de Rede em Redes de Acesso, controladas por operadoras de Internet. É visto que os problemas relacionados à Neutralidade de Rede poderiam aumentar com a implantação de Redes de Acesso Definidas por Software. Propomos então soluções que poderiam ser implementadas aproveitando também das facilidades trazidas por essa tecnologia.*

1. Introdução

Para facilitar o gerenciamento e configuração dos equipamentos de rede, a abordagem de Redes Definidas por Software (SDN) foi criada, possibilitando aos operadores administrar a rede de maneira rápida e centralizada. [Haleplidis et al. 2015]

Enquanto o conceito de Qualidade de Serviço(QoS), que descreve a medição da performance dos serviços, já era usado pelas redes de computadores, seu uso ainda é pouco prático. Mas quando combinado com as Redes Definidas por Software o QoS se torna muito mais poderoso. [Egilmez et al. 2013]

Recentemente, visando a proteção dos usuários e consumidores da rede a ideia de Neutralidade de Rede vem sendo discutida. Com a intenção de que os dados circulados nas redes sejam tratados da mesma maneira, independente do conteúdo, fonte ou destino, pelas operadoras de rede.

Portanto existe uma discordância entre argumentos. Em um campo, do QoS, queremos classificar o conteúdo circulado, para aumentar a performance. Mas no outro, da Neutralidade de Rede, queremos privacidade e que o conteúdo não seja alvo de discriminação.

2. Fundamentação Teórica

2.1. Redes Definidas por Software e suas facilidades

As Redes Definidas por Software trazem diversas facilidades aos operadores de rede quanto ao gerenciamento e administração. Separando a rede em diversos planos de interesse, repassamos as responsabilidades para um controlador central, tal que facilita a

manutenção dos equipamentos, por traduzir os nossos objetivos em configuração de cada *switch* da rede.

No plano de encaminhamento, onde cada equipamento anteriormente repassava os pacotes sem qualquer conhecimento de contexto, agora podemos facilmente realizar priorizações dinâmicas de fluxos. Aumentando e diminuindo a velocidade e/ou chance de um pacote ser perdido, conforme a prioridade de seu fluxo. Enquanto cada equipamento ainda não sabe o contexto dos fluxos, o controlador central conhece à todos e repassa as configurações para tornar isso possível.

2.2. Qualidade de Serviço e o aumento de sua responsabilidade

A Qualidade de Serviço refere-se à características do serviço de rede que ativamente afetam as necessidades do usuário. Comumente está relacionado à aspectos como velocidade, simplicidade, confiabilidade, precisão, corretude, entre outros, dos serviços de telecomunicação. [ITU 2008]

Buscando melhorar a experiências dos usuários da rede, e diminuir os gastos dos responsáveis por operar a rede, a Qualidade de Serviço é relacionada com o priorização de fluxos. Fluxos que obtêm maior prioridade são então dados maiores larguras de banda e menores *delays*. Enquanto fluxos que são considerados de menor prioridade ficam mais lentos, quando seus pacotes não são até perdidos.

Até então a manutenção da Qualidade de Serviço em redes é um serviço pouco prático, necessitando de muitas configurações, que são traduzidas em muita horas de trabalho para a configuração de muitos equipamentos. Com o advento das Redes Definidas por Software, que podem de maneira facilitada modificar a configuração de cada equipamento, não seguindo apenas o gerenciamento de um operador, mas de um controlador programado para realizar tal tarefa, a manutenção de Qualidade de Serviço ganha muito poder. A classificação e priorização de fluxos se torna muito masi fácil.

2.3. Neutralidade de Rede e problemas com a priorização de fluxos

O conceito de Neutralidade de Rede (NN), ou Net Neutrality, diz respeito ao ideal de que todos os dados devem ser tratados de maneira igual pela rede. Não favorecendo uma ou discriminando outra aplicação. O argumento se centraliza em uma ideia de livre competição, em que o meio onde ela acontece deve ser neutro e portanto meritocrático. [Wu 2003]

Tendo uma facilitação da priorização de fluxos na rede, podemos observar práticas mau vistas, sob a questão de Neutralidade de Rede, das operadoras se tornando mais comuns. Priorizando os serviços prestados pela própria operadora ou aliados, sobre seus concorrentes. Acabando com a competição justa de serviços para os consumidores daquela operadora.

Problemas observados são por exemplo: diminuição da prioridade de pacotes de concorrentes diretos, no caso de serviços de vídeo e telefonia via Internet; prioridade para serviços de aliados, no caso de não cobrar pelo uso de dados em certos serviços; discriminação de conteúdo, no caso de despriorização ou bloqueio de serviços como *torrent* ou conteúdo político.

3. Desenvolvimento

3.1. A métrica *Fairness*

Para realizar uma comparação qualitativa quanto à ”justiça” de que nossos pacotes podem ser priorizados, é criada a métrica *Fairness*. Dividida nos níveis **Confiar**, **Monitorar**, **Forçar**.

Quando no nível **Confiar**, podemos apenas esperar que o operador da rede está fazendo o melhor trabalho possível na priorização de nosso pacotes. Sendo o mais imparcila possível.

Passando para o nível **Monitorar** esperamos que o operador da rede esteja cumprindo com requisitos repassados pelo usuário final. Mas podemos apenas monitorar e não há certeza de que os requisitos sejam cumpridos.

No último nível, **Forçar**, seria possível obrigar a utilização de prioridades repassadas, mas normalizadas, do usuário final, na Rede de Acesso.

3.2. Montando uma base

Começando apenas com uma Rede de Acesso Definida por Software comum, a priorização dos pacotes ainda seria responsabilidade dos equipamentos das rede da operadora, já que os equipamentos da rede local não possuem a habilidade de se comunicar com a rede externa. Configurando um nível de **Confiar**.

Realizando a marcação de prioridades na rede local, que, através da utilização de Redes Definidas por Software, possa repassar as marcações para a Rede de Acesso, obtemos um nível **Confiar**. Já que podemos esperar que a Rede de Acesso poderia utilizar de nossas configurações. Mas precisamos realizar uma normalização das prioridades repassadas, na Rede de Acesso, já que um cliente mal intencionado poderia passar apenas as mais altas prioridades, prejudicando o serviço de outros clientes. Isso pode ser visto na Figura 1.

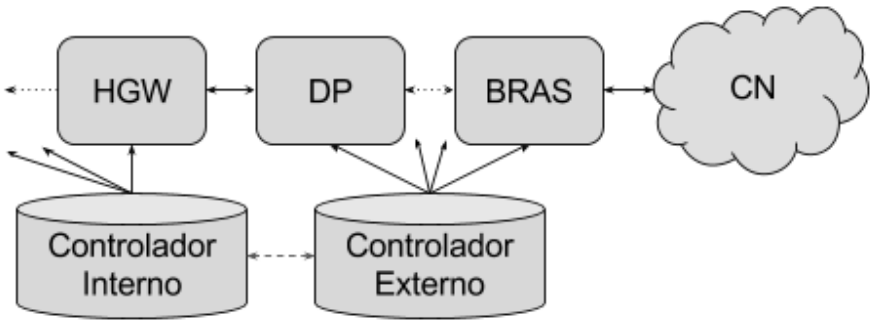


Figure 1. O repasse de prioridade pela abstração dos controladores de cada Rede Definida por Software

Terminamos então utiliando a virtualização das funções da Rede de Acesso Definida por Software, onde uma autoridade terceira, confiada pelo usuário e pela operadora, como um orgão governamental, realiza a captação e configuração da prioridades na Rede de Acesso, enquanto a operadora é responsável por qualquer outra configuração dos equipamentos da rede de acesso. Podemos imaginar isso então como uma função

composta, que determina a função virtualizada final do equipamento. Como na fórmula que pode ser vista na Figura 2. Na Figura 3 temos como esses cálculos seriam deslocados entre os controladores.

$$((autoridade(prioridades)) \circ operadora)(fluxo)$$

Figure 2. A função composta para a função virtualizada dos equipamentos da Rede de Acesso

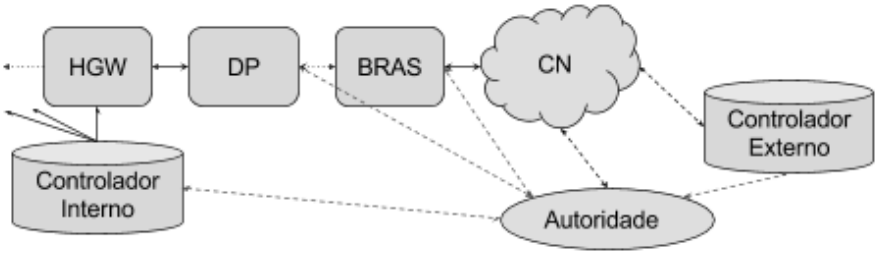


Figure 3. O repasse de prioridade para a autoridade, e da função virtualizada para os equipamentos

4. Conclusão

Utilizando a métrica estabelecida *Fairness* podemos considerar que o problema de Neutralidade de Rede estaria resolvido dentro dela. Podemos ainda utilizar da Qualidade de Serviço, mas utlizando das prioridades estabelecidas pelo próprio usuário. Ou quando não estabelecidas fornecendo um serviço comum de melhor esforço.

Essas soluções poderiam ser aplicáveis no mundo real, mas necessitariam de grandes investimentos para a transformação das Redes de Acesso em Redes de Acesso Definidas por Software. Mas poderiam ser aplicadas em simuladores como o Mininet para testar as suas prioridades.

References

Egilmez, H. E., Civanlar, S., and Tekalp, A. M. (2013). An optimization framework for qos-enabled adaptive video streaming over openflow networks. *IEEE Transactions on Multimedia*, 15(3):710–715.

Haleplidis, E., Pentikousis, K., Denazis, S., Salim, J. H., Meyer, D., and Koufopavlou, O. (2015). Software-defined networking (sdn): Layers and architecture terminology. RFC 7426, RFC Editor. `http://www.rfc-editor.org/rfc/rfc7426.txt`.

ITU (2008). Itu-t e.800 e.800 : Definitions of terms related to quality of service.

Wu, T. (2003). Network neutrality, broadband discrimination. *Journal of Telecommunications and High Technology Law*, 2:141–176.