JULIANO DE MELLO PEDROSO

Arquitetura SDN para Controle de Acesso à Conteúdo em Redes CCN

Curitiba - PR, Brasil

2022

JULIANO DE MELLO PEDROSO

Arquitetura SDN para Controle de Acesso à Conteúdo em Redes CCN

Tese apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de doutor em Ciências.

Pontifícia Universidade Católica do Paraná - PUCPR

Programa de Pós-Graduação em Informática - PPGIa

Orientador: Prof Dr. Marcelo E. Pellenz

Curitiba - PR, Brasil

2022

Aos meus pais, Raquiel (In memoriam) e Raul, que sempre foram anjos na minha vida. Sempre me ensinaram o respeito, dedicação, dignidade e honestidade. Aonde a senhora estiver mamãe obrigado pela liderança, nos encontramos na próxima vida. Papai obrigado pelo suporte, apoio, e me dar a vida. A minha esposa, Eliane, por fazer mais por mim do que para ela. Por me dar o maior presente da minha vida, a Beatriz.

AGRADECIMENTOS

Agradeço a Deus por me dar forças, bênçãos, iluminar meu caminho e encontrar maneiras de evoluir.

Agradeço ao Professor Dr. Edgard Jamhour por me ajudar no começo da pesquisa me fazendo evoluir muito, com conselhos e soluções não ortodoxos. Tenho muito orgulho de ter sido seu aluno.

Agradeço ao Professor Dr. Manoel Camillo de Oliveira Penna Neto por me ajudar todas as vezes que precisei, me fez evoluir muito no desenvolvimento da pesquisa. Tenho muito orgulho de ter sido seu aluno.

Agradeço ao Professor Orientador Dr. Marcelo Eduardo Pellenz por ter paciencia, extrema compentencia técnica e pedagógica fazendo meu trabalho melhorar muito. Tenho muito orgulho de ter sido seu aluno.

Agradeço a professora Sheila Reinehr por me ajudar muito com metodologia cientifica.

Agradeço ao Centro Universitário Internacional por acreditar na minha capacidade e ajudar muito na pesquisa.

Agradeço ao Professor Wilson Picler que também acreditou no meu trabalho e que me ensinou sobre diversos assuntos técnicos.

Agradeço a todos que de alguma forma me ajudaram.

Obrigado!!

Nossas dúvidas são traidoras e nos fazem perder o que, com frequência, poderíamos ganhar, por simples medo de arriscar. William Shakespeare

RESUMO

Contexto: As redes centradas em conteúdo (CCN - *Content Centric Networks*) representam uma importante mudança na forma de acesso à Internet, dando a prioridade ao conteúdo ao invés da comunicação entre dispositivos finais. A maioria dos conteúdos que são entregues atualmente são provenientes de publicadores de vídeo sob demanda e a entrega desse tipo de conteúdo é feita através de redes CDN (*Content Delivery Networks*), onde são implementados diversos servidores de réplicas do conteúdo. Essa abordagem apresenta desvantagens com o aumento da quantidade de conteúdo disponível. Neste sentido os roteadores desempenham um papel vital nas redes orientadas a conteúdo, pois eles podem ser utilizados para armazenar o conteúdo de forma incremental. **Objetivo**: O objetivo desse trabalho é apresentar uma arquitetura baseada em SDN (*Software Defined Networks*) para redes CCN que visa controlar o acesso ao conteúdo. Nessa arquitetura proposta, o controle de acesso passa a ser feito por controladores SDN. **Método**: A arquitetura proposta foi concebida em um ambiente simulado, utilizando a plataforma *Mininet* que fornece um ambiente de teste e desenvolvimento virtual para redes SDN. Neste ambiente foi instalado um *plugin* para dar suporte a redes CCN. A arquitetura proposta foi implementada no simulador para avaliação de segurança e desempenho, utilizando topologias de rede clássicas da internet. **Resultados**: A solução desenvolvida foi validada de forma satisfatória e pode ser utilizada como um controle de acesso ao conteúdo em vídeos sob demanda. Ela possibilita melhorias em termos de redução de tráfego na rede, aumento de disponibilidade do conteúdo e do esquema de controle de conteúdo dentro da topologia. **Conclusão**: A utilização conjunta das duas estratégias (SDN+CCN) possibilita novas abordagens para se melhorar a arquitetura atual da rede. O conceito CCN contribui para a melhora da largura de banda e fluxo de conteúdo. A estratégia SDN melhora o controle de conteúdo e o gerenciamento em um ambiente *multitenant*. A arquitetura proposta demonstrou ser uma solução viável que pode ser implementada como uma alternativa de controle de acesso em ambiente de entrega de vídeo sob demanda.

Palavras-chave:Controle de Acesso, CDN, SDN, CCN, Autorização, Vídeo sob Demanda.

ABSTRACT

Context: *Content-Centric Networks (CCN) represent an important change in the way of accessing the Internet, giving priority to content over communication between end devices. Most of the content currently delivered comes from video-on-demand publishers. This type of content is done through CDN networks (Content Delivery Networks), where several replica servers of the content are implemented. This approach has disadvantages as the amount of content available increases. In this sense, routers play a vital role in content-oriented networks, as they can be used to store content incrementally.* ***Objective***: *The objective of this work is to present an architecture based on SDN (Software Defined Networks) for CCN networks that aim to control access to content. In this proposed architecture, access control is done by SDN controllers.* ***Method***: *The proposed architecture was conceived in a simulated environment, using the Mininet platform that provides a virtual test and development environment for SDN networks. In this environment, a plugin was installed to support CCN networks. The proposed architecture was implemented in the simulator for security and performance evaluation, using classic internet network topologies.* ***Results***: *The developed solution has been satisfactorily validated and can be used as access control to content in on-demand videos. It enables improvements in reducing network traffic, increasing content availability, and the content control scheme within the topology.* ***Conclusion***: *The joint use of the two strategies (SDN+CCN) allows new approaches to improve the current architecture of the network. The CCN concept contributes to the improvement of bandwidth and content flow. The SDN strategy improves content control and management in a multitenant environment. The proposed architecture proved to be a viable solution that can be implemented as alternative access control in a video-on-demand delivery environment.*

Keywords: *Access Control, CDN, SDN, CCN, Authorization, Video-On-Demand.*

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CMMI	*Capability Maturity Model Integration*
TCP/IP	*Transfer Control Protocol/Internet Protocol*
IP	*Internet Protocol*
ICN	*Information Centric Network*
SDN	*Software Defined Network*
ZB	*Zettabyte*
EB	*Exabyte*
GB	*Gigabyte*
CDN	*Content Delivery Network*
Mbps	*Mega bits* por segundo
URL	*Uniform Resource Locator*
QoS	*Quality of Service*
DNS	*Domain Name System*
NDNsim	*NS-3 based named Data Networking*
CCNx	*Software based on CCN*
NDN	*Named Data Network*
RAM	*Random Acess Memory*

LISTA DE SÍMBOLOS

SUMÁRIO

1 INTRODUÇÃO

A arquitetura da *Internet* foi originalmente concebida com o objetivo de compartilhar informações utilizando a arquitetura clássica cliente-servidor. O modelo de comunicação foi projetado para permitir a interação entre duas partes, uma querendo acessar algum dispositivo ou servidor específico e a outra fornecendo esse acesso. A comunicação é suportada pela pilha de protocolo TCP/IP (ZHANG et al., 2014) (QIAO et al., 2019), na qual o endereçamento da camada de rede é fornecido pelo protocolo IP (Internet Protocol). Um endereço IP exclusivo é atribuído a cada dispositivo e os pacotes dependem deles para serem roteados pela rede.

O *streaming* de vídeo sob demanda é uma ferramenta poderosa para a entrega de conteúdo para os usuários e têm crescido de forma muito rápida. Por meio de um *software* de multimídia o usuário pode receber uma grande quantidade de vídeos, filmes e até mesmo jogos disponíveis em VoD (*Video-on-Demand*). O usuário tem total controle sobre o conteúdo que irá assistir, sem precisar aguardar algum tipo de programação ou grade proposta por uma emissora. Por isso a escolha da plataforma de entrega de conteúdo para a audiência se torna um dos componentes mais importantes do serviço.

O atual desenvolvimento da *Internet* nos últimos anos permitiu o avanço e a criação de novas aplicações que requerem cada vez mais distribuição de conteúdo de forma mais segura e onipresente. A partir disso, os usuários estão cada vez mais interessados no tipo e variedade de conteúdos, que possam ser acessados de uma forma eficiente e segura. Neste contexto os provedores de conteúdo devem possibilitar um mecanismo de acesso seguro, rápido e distribuído. Entretanto, para se atingir estes objetivos é necessária uma troca de paradigma com relação ao armazenamento do conteúdo que não pode estar mais centralizado e necessita um mecanismo de controle de acesso distribuído. Nesse sentido, considerando as necessidades atuais, o estudo e utilização de novas tecnologias de redes se torna necessário. As duas novas arquiteturas que podem fornecer esses recursos desejados são as redes ICN (*Information Centric Networks*) (ZHANG et al., 2014) e as redes SDN (*Software Defined Networks*) (DARGAHI et al., 2017) (AMIN; REISSLEIN; SHAH, 2018) (BANNOUR; SOUIHI; MELLOUK, 2018). As redes baseadas em conteúdo (ICN) alteram o paradigma atual da arquitetura centralizada baseada em cliente-servidor para uma nova arquitetura de rede com conteúdo distribuído, com acesso transparente para o usuário, independentemente da sua localização. Já o conceito de rede definida por *software* (SDN) atua de forma a oferecer maior controle e estruturação lógica da arquitetura da rede.

A arquitetura do ICN introduz um novo paradigma de endereçamento para a Internet, com foco na entrega de conteúdo a um solicitante específico, independentemente da

localização do conteúdo. A abordagem tradicional é centrada na identificação e localização de hosts (modelo cliente-servidor), enquanto a ICN usa conteúdo nomeado e roteamento baseado em nome para oferecer suporte à distribuição de conteúdo. Além disso, a segurança de conteúdo é intrínseca à arquitetura ICN, projetada para proteger o conteúdo publicado e os dados armazenados em *cache* ao longo dos elementos da rede em todo o caminho de comunicação (ZHANG et al., 2014). Uma aplicação clássica da arquitetura de rede centrada em informação (ICN) são as redes centradas em conteúdo (*Content-Centric Networking* - CCN) que tem como especificidade a segmentação dos conteúdos, que são especificados por nomes hierárquicos, para serem requisitados pelos consumidores de conteúdo. Especificamente a rede CCN será o foco do nosso estudo.

A arquitetura SDN (OPENFLOW, 2017) (DARGAHI et al., 2017) (AMIN; REISSLEIN; SHAH, 2018) (BANNOUR; SOUIHI; MELLOUK, 2018) aborda várias deficiências da *Internet* centralizando algoritmos de roteamento e comutação em um elemento de controle externo, que toma todas as decisões de roteamento com base no global vista da topologia da rede. O objetivo principal é superar os problemas que surgem pela existência de ativos de rede (*switches* e roteadores) incorporando *softwares* de rede proprietários, cuja programação depende das regras criadas pelo fabricante (OPENFLOW, 2017). Para isso, a arquitetura SDN propõe a separação do plano de dados do plano de controle, em um ambiente independente do fornecedor, que traz o conceito de virtualização de rede que permite a implantação sobre as redes existentes. A SDN permite a simplificação dos elementos da rede, que só executariam as tarefas para as quais seu *hardware* foi projetado. Além disso, toda a camada de *software* da rede é simplificada, tornando a evolução mais efetiva (CISCO, 2016).

Atualmente as redes de distribuição de conteúdo têm crescido bastante, principalmente na área de entretenimento, com até 80% desse montante com o serviço de *streaming* de vídeos, segundo dados de um relatório da CISCO (CISCO, 2016). O serviço de entrega de conteúdo transportará 71% de todo o tráfego da Internet. Estas novas redes de distribuição de conteúdo, que estão em crescente expansão, são tradicionalmente denominadas de *Content Delivery Networks* (CDN) (SALAHUDDIN et al., 2018). Contudo, a expansão das redes CDN apresenta um alto custo de implantação, complexidade de gerência e escalabilidade devido a replicação de conteúdo. Neste contexto, a utilização da arquitetura SDN para as redes CDN pode facilitar a gerência e configuração da rede, e o controle de acesso dos usuários aos servidores distribuídos de conteúdo.

Basicamente uma rede CDN exige uma replicação do conteúdo em grande escala, envolvendo diversos servidores, denominados de pontos de presença (QIAO et al., 2019). Isso pode gerar problemas sérios de escalabilidade em função do aumento da demanda por conteúdo. Neste contexto, a grande vantagem de se utilizar uma arquitetura CCN para substituir a rede CDN, seria que quando fosse acessado o conteúdo de um plublicador

todos os nós envolvidos na conexão poderiam armazenar uma cópia do conteúdo ou parte dele. Isso poderia facilitar os demais acessos ao mesmo conteúdo, reduzindo problemas de escalabilidade, eficiência e desempenho. Os atuais serviços sob demanda para distribuição de conteúdo de vídeo tais como Netflix e Hulu utilizam a arquitetura CDN (ADHIKARI et al., 2015). Contudo, o uso de CCN altera o paradigma atual, propondo uma nova abordagem tendo como foco o conteúdo (ZHANG et al., 2014) (QIAO et al., 2019). No conceito das redes CDN atuais, o lugar onde esta armazenado o conteudo tem grande importância, já nas redes CCN o lugar onde o conteudo está armazenado tem uma menor importância, pois o foco é a disponibilidade do conteúdo em mais lugares. Estas arquiteturas tem como foco a conexão e o compartilhamento de recursos de infraestrutura e de conteúdo (BRITO, 2012) (SALAHUDDIN et al., 2018) (ADHIKARI et al., 2015) (DOAN; BAJPAI; CRAWFORD, 2020a).

Embora a arquitetura clássica cliente-servidor seja predominante para as aplicações atuais, quando pensamos em uma rede orientada a conteúdo (CCN) o principal objetivo é disponibilizar o conteúdo independentemente da localização do mesmo (BRITO, 2012) (QIAO et al., 2019) (JMAL; FOURATI, 2017) (SHINDE; CHAWARE, 2018). A seguir levantamos um problema de pesquisa: As redes CDN que são substituídas por redes CCN necessitam de autenticação de dispositivos para o acesso ao conteudo? A partir desse problema temos a seguinte hipotese: Uma solução SDN pode ser usada para autenticar os dispositivos CCN de forma eficiente. Nesse contexto das redes CDN, propomos nessa tese uma nova abordagem para a entrega de conteúdo sob demanda, com a utilização do paradigma CCN, usando a arquitetura SDN para controle. Entretanto a troca de paradigma CDN pelo CCN traz desafios a serem tratados, um deles é a autenticação do dispositivo que acessará o conteúdo que é disponibilizado por publicadores de conteúdo. Com o conteúdo descentralizado os dispositivos armazenadores deveram ter um mecanismo de autenticação. Nesta tese propomos uma alternativa dessa autenticação utilizando as funções encontradas na rede SDN. A arquitetura proposta foi implementada e avaliada utilizando plataformas e ferramentas *open source* onde foi possível validar a proposta e principalmente o mecanismo de autenticação.

1.1 Motivação

Segundo o sumário executivo desenvolvido pela fabricante de equipamentos de rede CISCO (CISCO, 2016), o tráfego global atingirá mais de 3,3ZB por ano até 2023, com cerca de 30GB per capita. A cada segundo, um milhão de minutos de conteúdo de vídeo irão trafegar pela rede. Globalmente, o tráfego de vídeo IP será de 82% de todo o tráfego de Internet do consumidor em 2021. O tráfego da rede para entrega de conteúdo (CDN) transportará 71% de todo o tráfego da *Internet* até 2023. A partir desses dados do panorama futuro da *Internet* ressalta-se a importância da pesquisa de técnicas de entrega

de conteúdo de forma segura e otimizada.

Nesse sentido uma solução de *software* de código aberto *(Open Source Software - OSS)* pode ser uma saída satisfatória para melhorar o tráfego de vídeo sob demanda.

A solução SDN prevê melhora da utilização de serviços e produtos sem ter grandes preocupações com a rede, podendo inovar independente do *hardware* e *software*.

A escolha da adição dessas duas tecnologias é baseado na soma de vantagens da CCN e da SDN em conjunto. A tecnologia CCN fornece uma rede flexivel e escalável se compararmos com a tecnologia clássica da *Internet*, atendendo os requisitos modernos da *Internet* para a distribuição de conteúdo em grande escala. Já a tecnologia SDN agrega inteligência nas redes junto com o controle de tráfego e nesse nosso caso ajuda no controle de acesso ao conteúdo.

1.2 Objetivos

O objetivo deste trabalho é combinar os conceitos das arquiteturas SDN e CCN, proporcionando uma arquitetura orientada a conteúdo com o controle de acesso assegurado através de um *proxy* SDN. A arquitetura terá uma distribuição de encaminhamento de requisões por conteúdos nomeados nmo plano de controle, o armazenamento será distribuído no plano de dados e logicamente centralizado no plano de controle. Os principais benefícios esperados com esta arquitetura são: aumentar do controle sobre o conteúdo do arquivo em um provedor, melhorar o aproveitamento da largura de banda usando o balanceamento de carga, e resolver o problema de controle de acesso ao conteúdo. Neste contexto, os objetivos específicos deste trabalho de pesquisa são:

1. Pesquisa bibliográfica relacionada com o tema.
2. Definição da arquitetura básica e o ambiente de testes.
3. Especificação da arquitetura de rede.
4. Especificação do protocolo de comunicação entre os dispositivos SDN.
5. Especificação do protocolo de comunicação entre os dispositivos CCN.
6. Definição da metodologia de validação da proposta em um ambiente simulado.
7. Realização de experimentos, coleta de dados e conclusão da análise dos resultados.

Essa tese propõe a utilização de um sistema de autenticação em redes CCN utilizadas em entrega de conteudo de vídeo sob demanda. O sistema tem duas camadas: 1) uma camada CCN com o proposito de entregar conteudo com a melhor popularidade possivel. 2) uma

camada SDN responsavel pela autenticação do dispositivo que quer acessar o conteúdo dentro do dominio de entrega estipulado.

Diante das inúmeras possibilidades de pesquisa neste tema, este projeto de pesquisa foca na proposta de um protocolo de integração CCN/SDN, com a coleta de informações sobre o comportamento de um sistema que tenha somente um controlador SDN, com diversos nós CCN trabalhando num domínio de entrega de conteúdo.

1.3 Estrutura do Documento

Este documento está estruturado da seguinte forma. O Capítulo 2 apresenta a fundamentação teórica, que demonstra a evolução da temática de pesquisa nesta área nos últimos anos. Também são apresentadas definições importantes que formam uma base para este estudo. No Capítulo 3 são apresentados os principais trabalhos da literatura que estão relacionados com o tema desta pesquisa e com a problemática em estudo. No Capítulo 4 é descrita a metodologia de pesquisa adotada, com foco nos softwares utilizados e montagem da estratégia de experimentação utilizada. No Capítulo 5 apresentamos a arquitetura CCN/SDN proposta. No Capítulo 6 apresentamos os resultados dos experimentos consolidados no cenário proposto do Capítulo 4. As conclusões e trabalhos futuros são apresentados no Capítulo 7.

2 FUNDAMENTAÇÃO TEÓRICA

As redes orientadas a conteúdo se mostram promissoras quanto a forma de usar a *Internet*. Esse tipo de rede altera o paradigma atual, propondo uma nova abordagem que possui como foco o conteúdo. As arquiteturas de rede tradicionais focam essencialmente na origem da informação ou no recurso necessário para se tomar uma determinada ação, não levando em consideração o conteúdo requisitado. Esse tipo de arquitetura de rede legada tem como foco a conexão e o compartilhamento de recursos, que eram inicialmente escassos. A arquitetura atual da *Internet* ainda tem o foco principal na conexão entre sistemas que utilizam a arquitetura cliente-servidor. Quando se fala em rede orientada ao conteúdo, o principal objetivo é disponibilizar o conteúdo independentemente da localização deste conteúdo (BRITO, 2012). Neste capítulo abordamos o referencial teórico sobre estas arquiteturas de rede, além de outros tópicos correlatos com este trabalho de pesquisa. São apresentados os conceitos básicos sobre as redes de distribuição de conteúdo, as redes centradas na informação e as redes definidas por *software*.

2.1 Redes de Distribuição de Conteúdo

Uma rede de distribuição de conteúdo (Content Delivery Network - CDN) é um tipo de rede que armazena conteúdo de outros domínios na sua memória *cache* e depois o disponibiliza para os usuários, com base na sua geolocalização. Essa localização é utilizada para reduzir a latência, pois o usuário busca o conteúdo num servidor mais próximo a sua localização. Esse tipo de rede normalmente é usado para distribuir conteúdo estático, como por exemplo *streaming* de vídeo, ou conteúdo dinâmico/interativo. A CDN é um conjunto colaborativo de elementos que abrangem a Internet, onde o conteúdo é replicado em vários servidores espelhados, para se ter uma entrega transparente e eficaz da informação para os usuários. Essa colaboração entre dispositivos CDN pode ocorrer sobre nós em ambientes heterogêneos e homogêneos (PATHAN; BUYYA; VAKALI, 2008).

Esse tipo de rede evoluiu no sentido de melhorar os limites inerentes da Internet quanto ao quesito de qualidade de serviço (*Quality of Service* - QoS). A CDN disponibiliza serviços que melhoram o desempenho da rede, maximizando a largura de banda e melhorando também a acessibilidade (PATHAN; BUYYA; VAKALI, 2008). Uma aplicação CDN fornece serviços de redirecionamento e entrega de conteúdo, direcionando o cliente para um servidor mais próximo. Também oferece serviços de replicação de conteúdo de servidores de origem para servidores cache (servidores de armazenamento), além de serviços de negociação de conteúdo e gerenciamento para atender necessidades específicas e informar sobre estatísticas de acesso ao conteúdo.

A Figura 1 mostra um modelo de CDN onde existe um *cluster* (célula) de distribuição de conteúdo. Existe um servidor de origem, chamado de CSP (*Content Service Provider*), onde é postado o conteúdo original. Após a postagem, o conteúdo é replicado para os servidores de armazenamento (*Surrogate Servers*) em diversos países de forma antecipada. Por fim o conteúdo é copiado para os usuários sob demanda dentro do domínio de distribuição do conteúdo, identificados como CDN´s *Content Distributor*. Dessa forma o usuário não tem a informação direta de qual servidor foi demandado o conteúdo.

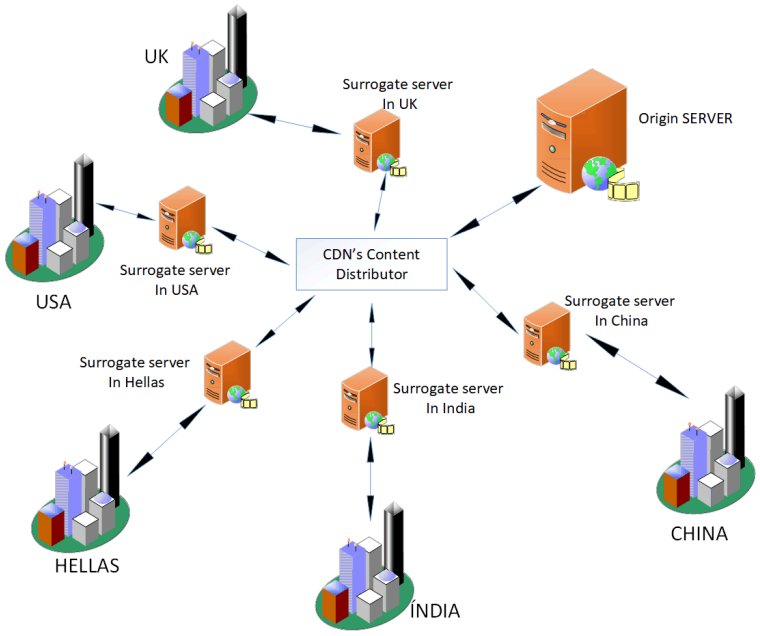

Figura 1 – Modelo de CDN (PATHAN; BUYYA; VAKALI, 2008)

Existem três elementos principais no CDN. O primeiro é o provedor de conteúdo, que é um fornecedor de arquivos ou mídias que serão disponibilizadas na rede. Esses arquivos são delegados com o nome URL (*Uniform Resource Locator*), que na verdade serão objetos WEB a serem distribuídos. Tem a função de manter os arquivos originais e atualizados. O segundo é o provedor CDN, que é uma organização proprietária ou empresa que fornece infraestrutura de rede aos provedores de conteúdo, a fim de fornecer o conteúdo de forma segura. O terceiro são os usuários que acessam o conteúdo do provedor replicados(Surrogates), mais perto da sua localização.

Além destes elementos, existem outros dispositivos que permeiam a CDN. Os provedores de conteúdo usam servidores de armazenamento para controlar réplicas em diversas localidades, e são chamados de servidores *cache*. O termo servidores de borda ou substitutos também são usuais para definir servidores de *cache*. Eles são distribuídos em forma de cluster, de tal maneira que todos os servidores tenham o mesmo conteúdo e URL (PATHAN; BUYYA; VAKALI, 2008). A Figura 2 mostra um exemplo genérico de uma rede CDN, demonstrando suas entidades principais e suas relações.

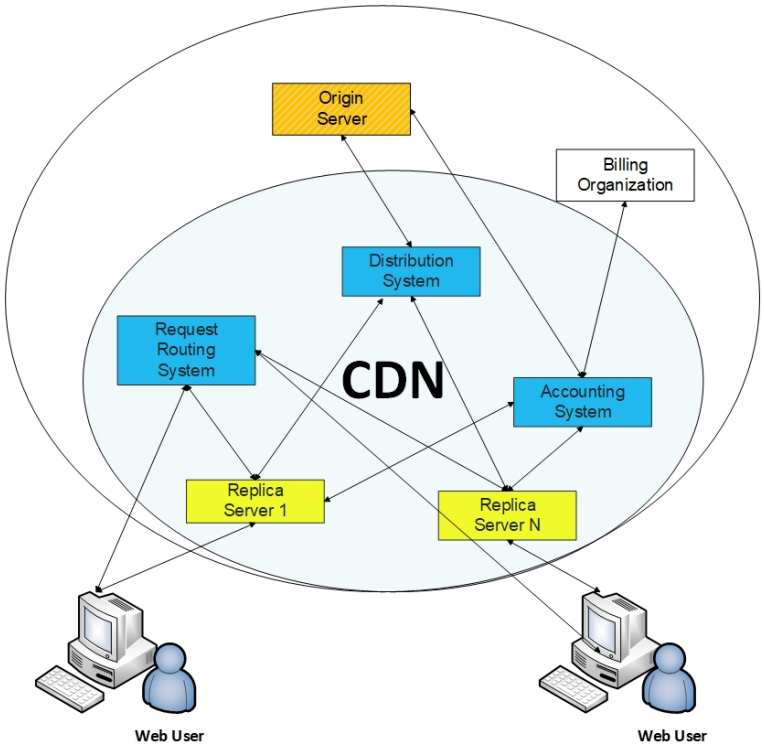

Figura 2 – Arquitetura CDN (PATHAN; BUYYA; VAKALI, 2008).

O *Origin Server* é responsável pela entrega de conteúdo, sendo composto por um servidor de origem e um conjunto de servidores de armazenamento que entregam o conteúdo aos usuários. O *Request Routing System* é responsável pela direção dos pedidos de usuários para escolher o melhor servidor de armazenamento que contemplará o usuário. O *Distribution System* é um sistema de distribuição, que move o conteúdo do servidor de origem para os servidores de armazenamento. O *Accounting System* é o componente que mantém os *logs* de acessos e registros de usuários que acessaram o conteúdo CDN (*Accounting System*). Os *Replicas Servers* são os servidores responsáveis por armazenar as réplicas dos conteúdos disponibilizados pelos servidores de origem (*Origin Server*). Já o *Billing Organization* é a entidade responsável pelo registro dos usuários (PATHAN; BUYYA; VAKALI, 2008).

Arquitetura atual da internet estabelecida a partir do TCP/IP obteve grande sucesso e se tornou infraestrutura indispensável na vida diária. Entretanto, tendências para novas arquiteturas crescem dia a dia. Os protocolos da camada de transporte e rede, como o TCP/IP, são difíceis de serem modificados. Normalmente para se resolver os problemas de escalabilidade é necessário se fazer adaptações adicionais de softwares. Contudo, a camada de aplicação se desenvolve de forma muito mais rápida e eficaz. Quando se fala em entrega de conteúdo, temos várias oportunidades de melhoria, porque o TCP/IP é orientado a conexão e não ao conteúdo propriamente dito. Isso significa que depois de estabelecida a conexão, o protocolo da camada de rede não se preocupa mais com o que

está trafegando na rede.

Contudo, quando falamos em entrega de conteúdo, o conceito atual da tecnologia é a CDN. A tecnologia CDN pode ser dividida em três categorias básicas que nos interessam nessa pesquisa (SALAHUDDIN et al., 2018)(ADHIKARI et al., 2015)(DOAN; BAJPAI; CRAWFORD, 2020a):

- CDN de Uso Geral – Esse tipo de CDN implementa o que é mais frequentemente chamado de aceleração da *Web*. Em geral, isso é realizado por um CDN com vários servidores em diferentes locais. Idealmente eles devem estar próximos aos grandes pontos de conexão entre provedores de serviços de *Internet* (ISPs) ou nos mesmos centros de dados de um site ou provedor de jogos/aplicativos. O CDN armazena em *cache* uma cópia do conteúdo que será frequentemente solicitado por um grande número de usuários.

- CDN de Vídeo sob Demanda - Algumas CDNs de uso geral também fornecem serviços CDN para conteúdo de vídeo sob demanda. O raciocínio é que o conteúdo de vídeo é apenas um grande arquivo para *download*, como um jogo ou aplicativo. Portanto, a veiculação de conteúdo de vídeo não deve ser muito diferente de outro conteúdo. Há alguns anos, a desconexão entre CDNs de propósito geral e de vídeo sob demanda era bastante distinta, já que a entrega de vídeo exigia o uso de um servidor de streaming. Os servidores de *streaming* entregam o conteúdo no momento de uma solicitação, e tem a possibilidade de entregar trechos ou porções solicitadas do vídeo, ao invés do vídeo completo. Isso foi útil para o proprietário do conteúdo que estava pagando a CDN pela entrega, pois o usuário que escolher abandonar a exibição de um vídeo na metade da duração, não teria baixado o clipe inteiro, independentemente da velocidade de conexão da *Internet* do visualizador. Uma inovação importante implementada na arquitetura CDN foi a taxa de *bits* adaptativa (Adaptive Bit Rate - ABR), que converte um fluxo de vídeo em fragmentos ou pedaços, geralmente de 2 a 10 segundos de duração. A ABR cria fluxos discretos em várias taxas de *bits* e, em seguida, usa o *feedback* do *player* de vídeo do usuário da Internet para detectar dinamicamente a velocidade de rede ideal para a entrega do vídeoclipe. Atualmente existem várias soluções ABR proprietárias no mercado, incluindo soluções de grandes empresas como Adobe, Apple, Microsoft e *Move Networks*. Todas funcionam de maneira bastante semelhante, e algumas até permitem a entrega via *streaming* HTTP ou por *download* progressivo.

- Ao Vivo - Além do uso das estratégias de *streaming* usando ABR e *streaming* HTTP, ainda há uma necessidade de entrega de vídeo ao vivo, já que o vídeo ao vivo não pode ser armazenado em *cache* como um conteúdo que foi gravado previamente. Os modelos CDN para este tipo de aplicação ainda não sofreram grandes evoluções, por

três razões principais. Primeiro, a grande maioria do conteúdo de vídeo fornecido pelos CDNs é um vídeo sob demanda. Algumas estimativas indicam que cerca de 95% de todos os vídeos *on-line* estão sendo entregues como conteúdo de vídeo sob demanda. Em segundo lugar, como o vídeo ao vivo não pode ser armazenado em *cache*, é necessário modificar a infraestrutura CDN básica para ter canais de largura de banda muito alta entre um local central e o usuário final visualizando o conteúdo. Outra abordagem seria ter canais com largura de banda ligeiramente menor, que enviem a transmissão ao vivo para um repetidor que esteja mais próximo do usuário final. Em terceiro lugar, considerando os dois pontos acima, o custo de construir e manter uma solução de *streaming* ao vivo para eventos ao vivo muito populares é extremamente alto. Construir e manter uma solução de *streaming* ao vivo de mais de um milhão de espectadores é muito caro (SALAHUDDIN et al., 2018)(ADHIKARI et al., 2015)(DOAN; BAJPAI; CRAWFORD, 2020a).

Particularmente nesta pesquisa, focamos no modelo CDN para vídeo sob demanda.

2.2 Redes Orientadas a Conteúdo

O princípio clássico de entrega de conteúdo na *Internet* foi fundamentado na arquitetura cliente-servidor, pois a rede tinha como foco resolver o compartilhamento de recursos remotos, que eram escassos e de alto custo. O modelo resultante é uma comunicação entre duas máquinas, uma desejando utilizar o recurso e a outra provendo acesso a este recurso. Assim pacotes IP contêm dois identificadores (endereços), um para a origem e outro para o destino. A maioria do tráfego TCP/IP consiste na comunicação entre pares de computadores (ZHANG et al., 2012). A rede CCN (*Content Centric Networking*) é uma arquitetura de comunicação construída para trabalhar com dados nomeados. Tal rede sugere que dados nomeados são uma abstração melhor para se resolver algumas questões atuais relacionadas com a comunicação, tais como disponibilidade, segurança e dependência de localização (ZHANG et al., 2012). A rede CCN têm sido impulsionada pelos consumidores de dados.

A rede CCN pode trabalhar de forma *overlay* ou *sobreposta* [1] com outras arquiteturas de redes, como por exemplo as redes IP. A Figura 3 compara a pilha de protocolo CCN com a arquitetura clássica das redes IP (Modelo OSI/ISO). O CCN se diferencia de várias maneiras, principalmente com relação as estratégias de transmissão e de segurança (ZHANG et al., 2012). Conforme ilustrado na Figura 4, existem basicamente dois tipos de pacotes na arquitetura CCN: pacotes de interesse (*Interest*) e pacotes de dados (*Data*).

[1] A sobreposição de rede é um método de usar um software para criar camadas de abstração de rede que podem ser usadas para executar múltiplas camadas de rede virtualizadas separadas sobre a rede física, geralmente fornecendo novos aplicativos ou benefícios de segurança.

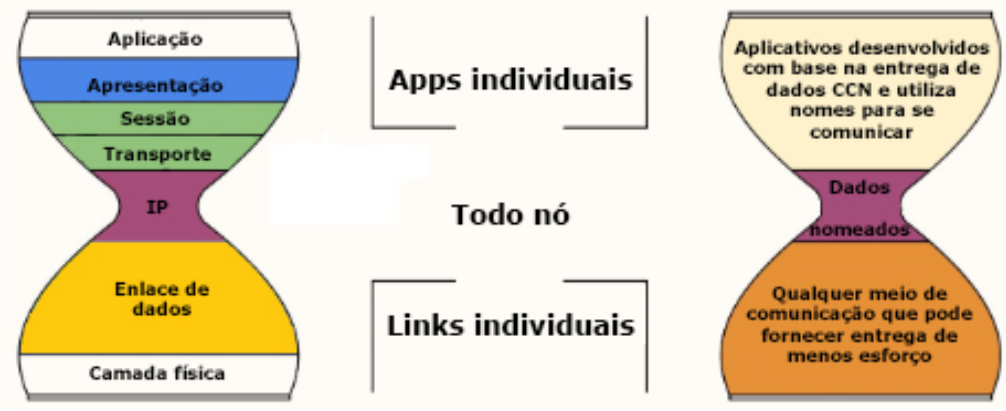

Figura 3 – Arquitetura IP versus CCN (ZHANG et al., 2012).

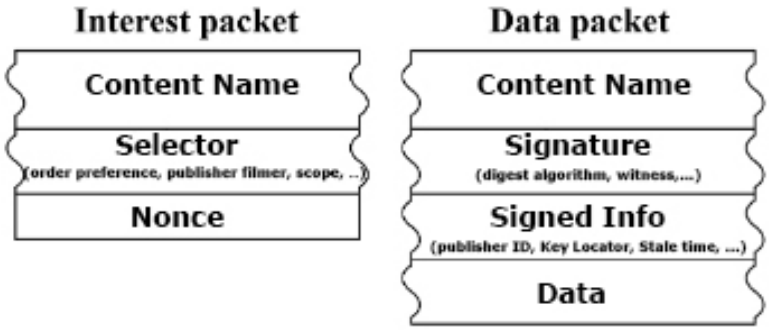

Figura 4 – Tipos de pacotes CCN (ZHANG et al., 2012).

Um usuário ou consumidor de dados, pergunta por um conteúdo e por *broadcast* transmite seu interesse para todo mundo que tem conectividade. Qualquer nó que recebe o interesse e tenha o conteúdo (dado) pode responder com o pacote de dados. O pacote de dados é transmitido somente em resposta a um pacote de interesse, ou seja, um para um. Um pacote de dados satisfaz um interesse se o *Content Name* no pacote de Interesse é um prefixo de um *Content Name* no pacote de dados.

Esses nomes são definidos por um esquema semelhante ao usado no sistema operacional Linux. Os nomes são constituídos por uma série de caracteres separados por "/" (barra), tendo hierarquia. Essa hierarquia é relevante apenas para a rede, pois ativos de rede classicos não identifcam esse parâmetro nominal. Por conveniência notacional, escrevem-se os nomes como se fossem URLs com significado para o usuário (ZHANG et al., 2012).

Na Figura 4 temos um exemplo de nome hierárquico nomeado. Estes nomes são humanamente legíveis, que normalmente refletem o significado do conteúdo. No caso da figura temos o nome humanamente legível de /parc.com/vídeo/WidgetA.mpg/_V/_s3. A

parte /parc.com constitui o nome globalmente roteável e a parte /vídeo/WidgetA.mpg constitui a parte do nome organizacional. A partir desse ponto temos as versões e segmentações. Por exemplo, se estivessem procurando o quarto segmento da segunda versão desse vídeo, poderíamos ter o seguinte nome: /parc.com/vídeo/WidgetA.mpg/2/4. A conversão codificada do conteúdo pode ser usada para a transmissão e sistemas de resolução de nomes(ZHANG et al., 2012).

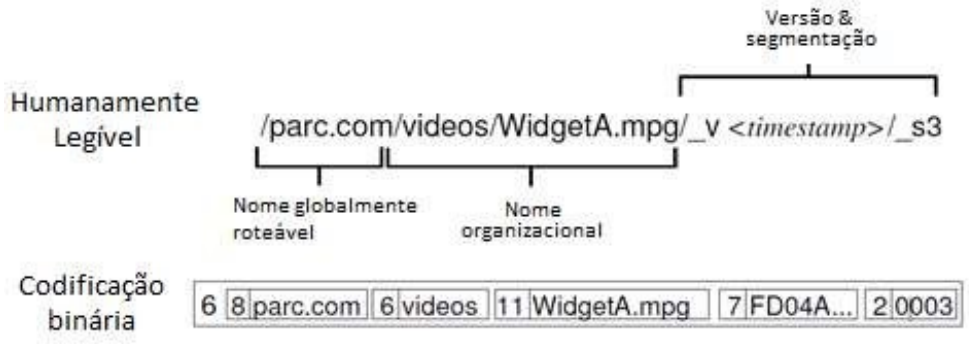

Figura 5 – Exemplo de dado nomeado na CCN (ZHANG et al., 2012).

Esses nomes são hierárquicos, e tem a vantagem de os interesses poderem ser recebidos para conteúdos que ainda nem existem. Isso permite que um editor possa gerar conteúdo simultâneo em resposta a um pedido em particular. Tais nomes permitem a rede CCN suportar uma mistura de *cache* estático e de conteúdo gerado automaticamente. Prefixos de nomes também podem ser susceptíveis ao contexto, como exemplo, o nome local/friends poderia ser usado para trocar informações sobre amigos num determinado ambiente (ZHANG et al., 2012).

A Figura 6 apresenta uma arquitetura base do CCN, que mostra o mecanismo de encaminhamento de pacotes, que contém 3 estruturas de dados: A FIB (*Forwarding Information Base*), *Content Store* (armazenamento) e a PIT (*Pending Interest Table*). A FIB é usada para encaminhar os pacotes de interesse em relação a origem que tenham correspondência de dados em potencial. Ela é quase idêntica a uma FIB IP, exceto que ela permite uma lista de faces [1] de saída, em vez de uma única, o que possibilita que várias fontes de dados sejam consultadas em paralelo. O *content store* é o mesmo que o *buffer* de memória de um roteador IP, mas com uma política de substituição diferente. Uma vez que cada pacote IP pertence a uma única comunicação ponto-a-ponto, o mesmo não tem mais valor depois de encaminhado ao destino. Já na rede CCN o *cache* armazena pacotes que podem ser acessados futuramente por novos pacotes de interesse. A PIT mantém o controle de interesses encaminhando o pedido de conteúdo para a origem, então os dados podem ser enviados para seus solicitantes (ZHANG et al., 2012).

[1] Usa-se o termo face ao invés de interface porque os pacotes não são transmitidos por interfaces físicas de rede, mas também trocados diretamente por processos de aplicações dentro de uma máquina.

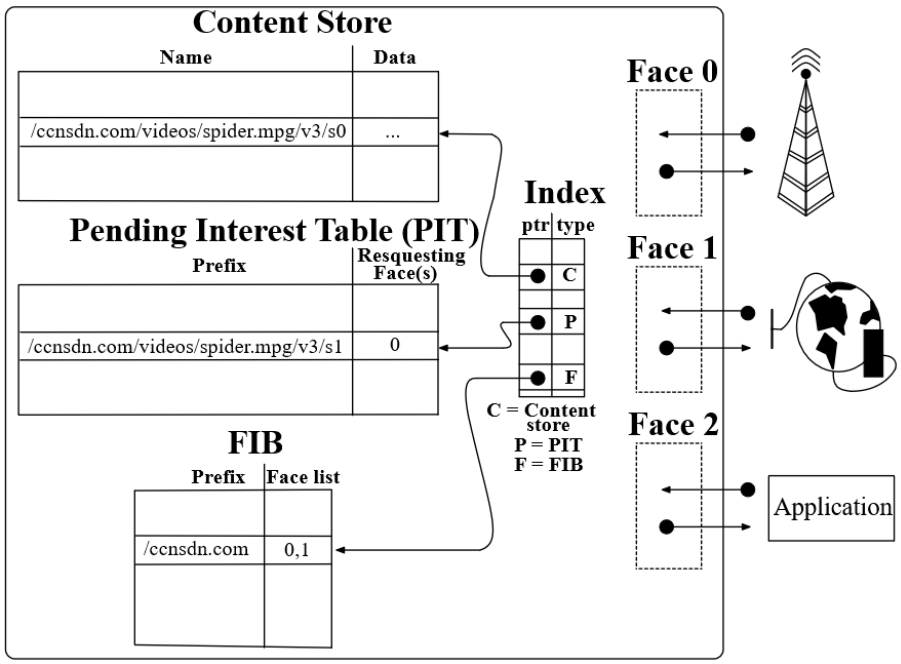

Figura 6 – Mecanismo de encaminhamento CCN (ZHANG et al., 2012).

Em uma rede CCN, somente os pacotes de interesse são roteados, sendo propagados a possíveis fornecedores de dados, deixando uma indicação de rota para que um pacote de dados possa seguir de volta para seu requisitante original. Cada entrada na PIT é uma indicação de rota. As entradas na PIT são apagadas tão logo o cliente recebe o pacote de dados. Entradas na PIT, para as quais não são encontradas correspondências ou para as quais o tempo de espera é esgotado, são apagadas da PIT, e o cliente é responsável pelo reenvio se ainda estiver interesse no dado. Quando um pacote de interesse chega em alguma face, uma consulta completa é feita no seu *content name*. A estrutura de índice utilizada para a pesquisa é ordenada da seguinte maneira: *content store*, depois a PIT e por último a FIB. Assim, se já existe um pacote de dados na *content store* que coincide com o interesse, o mesmo será enviado para a face requisitante e o interesse será descartado (desde que satisfeito) (ZHANG et al., 2012).

Por outro lado, se houver uma correspondência igual na tabela PIT, a face será adicionada a entrada numa lista de requisição de faces e o interesse será descartado. Caso contrário, se há uma entrada correspondente na FIB, o interesse deve ser enviado ao cliente. A face de chegada é removida da lista de entrada da FIB, em seguida, se a lista de entrada não está vazia, o interesse é enviado para todas as faces que ainda permanecem e uma nova entrada na PIT é criada. Se não houver nenhuma correspondência para o interesse ele é descartado, ou seja, esse nó não tem nenhuma correspondência e não sabe como encontrá-lo. A estrutura de identificação da rede CCN é baseada em conteúdo e não em um endereço fixo do dispositivo. Isso permite a concepção e implementação de algoritmos de *caching* e balanceamento de carga, com base no conteúdo e, principalmente, esquemas

de segurança baseado no conteúdo (ZHANG et al., 2012).

2.3 Rede Definida por Software

O processo atual de instalação e configuração da rede requer alta qualificação e o conhecimento de vários ativos de redes especializados, onde as interações entre os nós, como por exemplo *switches* e roteadores são complexas. Além disso, os custos operacionais envolvidos na implantação e gerenciamento de projetos que abrangem equipamentos de diversos fabricantes também é um limitante na implementação do projeto. Entretanto, podemos ter problemas depois da rede implantada. Considere como exemplo uma rede implantada em uma grande empresa. Fica extremamente complicado se fazer testes de novas tecnologias ou alterações que possam vir a impactar no funcionamento normal dessa rede.

Neste contexto surge o conceito da rede definida por *software* (Software Defined Network - SDN). Esse tipo de rede foi proposto pela Universidade de *Stanford* e posteriormente assumido pela *Open Networking Foundation* (ONF, 2017), que atua num consórcio com diversas empresas como *Facebook*, *Google*, *Microsoft*, entre outras. Essas empresas já têm projetos implementados de SDN em sua infraestrutura. O SDN é *open source*[1] nativo. Nesse tipo de arquitetura temos a separação do plano de dados (*Data Plane*) do plano de controle (*Control Plane*). Com a alteração do paradigma atual de redes, o SDN abre possibilidade para se pesquisar e projetar novas aplicações que gerenciem os elementos ativos da rede de maneira diferente das tradicionais. Na Figura 7 temos a comparação entre um modelo atual e o modelo SDN, onde podemos ver também que o plano de controle fica centralizado num dispositivo separado (MCKEOWN et al., 2008) (DARGAHI et al., 2017)(AMIN; REISSLEIN; SHAH, 2018)(BANNOUR; SOUIHI; MELLOUK, 2018).

O plano de dados tem como objetivo encaminhar os pacotes de dados, o que pode ser feito com o *hardware* disponibilizado em *switches* ou roteadores. O encaminhamento de pacotes são ações tomadas pelas ativos de rede, que podem ser escolher enviar para alguma porta especifica, enviar para várias portas, todas as portas, ou mesmo descartar o pacote. Já o plano de controle toma decisões de como as operações escolhidas no plano de dados serão feitas. Essas decisões abrangem encaminhamento, filtro de pacotes, priorização, etc. O plano de controle tem características de sistemas de tempo real (MCKEOWN et al., 2008) (DARGAHI et al., 2017)(AMIN; REISSLEIN; SHAH, 2018)(BANNOUR; SOUIHI; MELLOUK, 2018).

O SDN torna a rede programável. Esse atributo aumenta a flexibilidade no gerenciamento da rede e o plano de controle separado, cria um dispositivo logicamente centralizado

[1] A definição de software livre foi criada pela *Open Source Iniciative* (OSI), que diz que um programa de código aberto (*open source*) deve garantir que a distribuição seja livre, assim como o acesso ao código fonte.

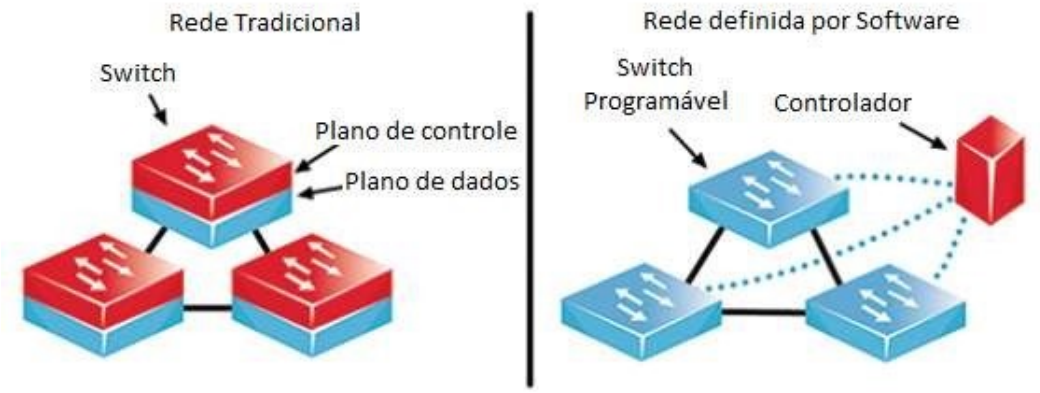

Figura 7 – Rede Tradicional versus SDN. Adaptado de (OPENFLOW, 2017).

na rede para ser usado como controlador. Na rede definida por *software* existem dois componentes da rede que naturalmente exigem que sejam conhecidos: Fluxo e Tabela de fluxo. O fluxo (*flow*) é uma cadeia de pacotes que são enviados de um dispositivo final até outro dispositivo final, e a tabela de fluxo tem a composição das entradas dos *flows* e qual ação a ser tomada no envio desses fluxos(MCKEOWN et al., 2008) (DARGAHI et al., 2017)(AMIN; REISSLEIN; SHAH, 2018)(BANNOUR; SOUIHI; MELLOUK, 2018).

Existem algumas estratégias para se implementar a rede SDN. A principal delas é o uso do protocolo *Openflow* (PITT, 2015). Publicado em meados de 2008, ele permite a criação de experimentos com novos protocolos em redes tradicionais. O principal componente do protocolo *Openflow* é o *Openflow controller* que é um item primordial nessa arquitetura (OPENFLOW, 2017).

O *openflow* pode disponibilizar descoberta automática de dispositivos, políticas de encaminhamento da empresa implementadas de forma mais fácil e sendo independente de fornecedores e/ou fabricantes. Por outro lado, a comutação via *software* é bem mais lenta. Detalhes inerentes a cada fornecedor existem e são necessários ajustes de algumas características do protocolo de roteamento rodando no dispositivo.

Na Figura 8 temos um esquemático de funcionamento de uma estrutura *openflow*, que inclui os seguintes componentes (MCKEOWN et al., 2008) (DARGAHI et al., 2017):

- Controlador – É o dispositivo que coordena todas as ações dos demais equipamentos, preenchendo a tabela de fluxos.
- *Switch Openflow* – É um dispositivo normalmente composto por no mínimo três partes: (1) uma tabela de fluxos, onde cada fluxo é associado a uma regra de como deve ser processado se for requisitado, (2) Canal seguro, pelo qual são passados os comandos provenientes do controlador, (3) Protocolo *Openflow*, que provê uma maneira do controlador se comunicar com o switch de forma remota.

O *switch Openflow* deve ser capaz de executar as seguintes tarefas básicas:

- Comutar o fluxo de entrada para uma determinada porta de saída, respeitando regra implementada na tabela de fluxos.
- Se não tiver uma regra implementada, o *switch* deve enviar ao controlador o pacote através do canal seguro.
- O *switch* deverá descartar pacotes se for o caso.

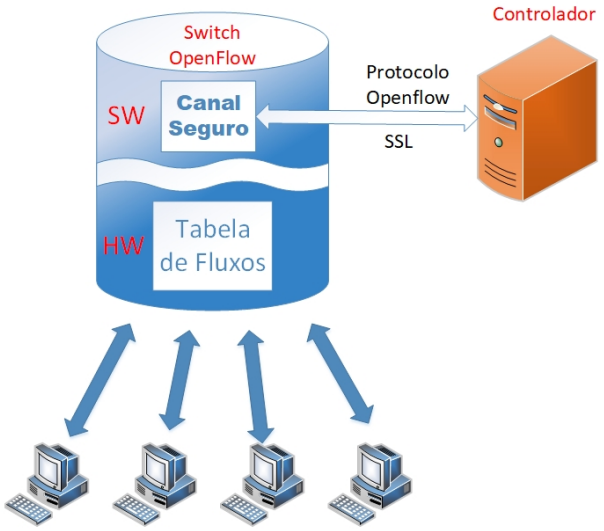

Figura 8 – Funcionamento do *Openflow*. Adaptado de (MCKEOWN et al., 2008).

Cada entrada na tabela de fluxos é composta por vários campos que são utilizados para categorização de um fluxo, ou seja, uma ação para cada fluxo (OPENFLOW, 2017). A Tabela 1 apresenta um exemplo de entradas na tabela de fluxos. Na tabela temos o número da regra que diz a ordem de execução, a regra de compatibilidade de fluxos, as ações a serem tomadas com o pacote caso a regra tenha correspondência e o contador que fornece estatísticas que também podem ser usadas para implementar regras. Na Tabela 2 temos os campos do cabeçalho *Openflow*, cada campo pode ser ou não definido. Por exemplo se usarmos o campo IP o *switch Openflow* pode ter um fluxo de roteamento. Esses campos são usados como parâmetros para a criação das regras. A Figura 9 apresenta exemplos de fluxos do protocolo *openflow*, onde cada linha é um fluxo e os asteriscos significam qualquer valor entre os intervalos possíveis (MCKEOWN et al., 2008).

Em resumo, o encaminhamento nas redes SDN é realizado por uma entidade denominada controlador. Ele possui uma visão global da topologia da rede, e que é capaz de programar os fluxos de conexão na rede, "programando" os *switches* através de inclusão e remoção de entradas em suas tabelas de fluxo. As principais vantagens trazidas pelo

Tabela 1 – Exemplos de Fluxo

Número da Regra	**Regra**	**Ação**	**Contador**
1	Porta de origem TCP = 673	Comutar para a porta 400	0
2	IP de origem = 5.5.5.5	Encaminhar para o controlador	15
3	IP destino = 153.10.10.15	Bloquear	650

Fonte: O autor

Tabela 2 – Campos do Cabeçalho *Openflow*

Porta de entrada	MAC de Origem	MAC de Destino	Ethertype	ID da Vlan	IP de Origem	IP de Destino	Porta de origem TCP	Porta de destino TCP

Fonte: (MCKEOWN et al., 2008).

Port	VLAN	ETH SRC	ETH DST	IP SRC	IP DST	IP Proto	L4 SRC	L4 DST	Ações
*	*	*	00:4F:...	*	*	*	*	*	Port 4
*	*	*	*	*	*	TCP	*	22	DROP
*	1	*	00:4F:...	*	*	*	*	*	Port 4, 6, 9

Figura 9 – Exemplo de tabela de fluxos no *switch openflow*. Adaptado de (OPENFLOW, 2017).

SDN são a facilidade de criação de *clusters* em nuvem, maior controle do tráfego, redução de custos e aumento sensível na inteligência da rede.

2.4 Conceitos e Mecanismos de Rede

Atualmente ainda não existe uma solução completa que resolva todas as implicações de se escolher uma rede orientada a conteúdo. Existem algumas estratégias para se atender de forma parcial essas implicações, contornando as limitações da arquitetura clássica de Internet. Como já mencionado, a utilização de CDNs tem sido uma solução muito atrativa e amplamente utilizada no mundo todo. Entretanto esse tipo de arquitetura contém problemas de segurança dos conteúdos, assim como questões de disponibilidade e persistência dos conteúdos. Mudar o paradigma e utilizar redes especializadas em conteúdo podem melhorar radicalmente a eficiência na localização e na entrega do conteúdo. Neste capítulo serão descritos os conceitos e mecanismos clássicos utilizados em redes orientadas

a conteúdo. As seções a seguir descrevem as tecnologias do uso de *proxy* reverso, AAA e aplicações *Multitenant*.

2.4.1 Proxy Reverso

Classicamente um servidor *proxy* é um tipo de servidor que redireciona as requisições de um cliente por algum serviço fornecido por outro servidor, em uma outra rede. Na Figura 10 temos um esquema de ligação de um servidor proxy na rede. Nesse cenário temos a rede interna que é composta por diversos computadores, tais computadores estão conectados fisicamente a equipamentos ativos de rede exemplificados por um switch, um servidor proxy, um modem ADSL (*Asymmetric Digital Subscriber Line*) que fornece acesso à intranet e a internet. Se qualquer um dos computadores que fazem parte da rede interna quiser acessar algum serviço da internet, primeiramente vai enviar a requisição para o servidor proxy, no servidor proxy estarão programadas regras de acessibilidade desse computador, providenciando acesso e anonimato ao usuário.

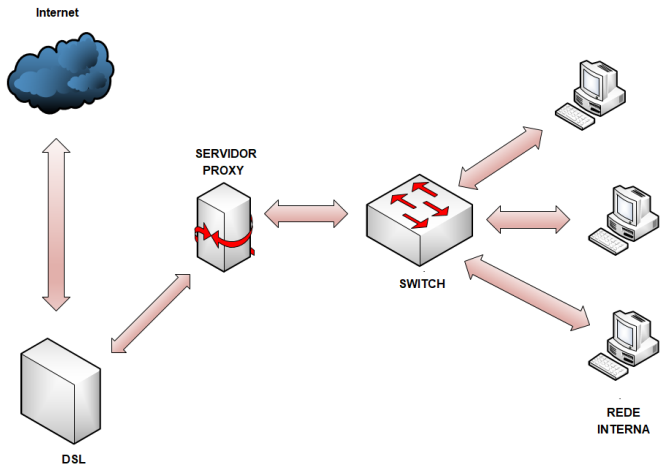

Figura 10 – Servidor *Proxy* (o Autor)

.

Um servidor *proxy* faz um papel intermediário entre o cliente e o servidor que o cliente quer acessar o serviço. Nesse ponto o servidor *proxy* pode ser configurado com especificações que irão filtrar o tráfego da rede. Essas especificações são escolhidas pelo administrador de rede e tem o papel de estabelecer regras de entrada e saída dos pacotes de rede. Assim os pacotes indesejados são travados nesse dispositivo e descartados (JEFFERY; DAS; BERNAL, 1996).

Um servidor proxy pode ser configurado para trabalhar de forma reversa, chamado *proxy* reverso. Este tipo de *proxy* atua de forma contrária a definição clássica de *proxy*. Enquanto o *proxy* filtra requisições do cliente (LAN) para a rede externa, um *proxy* reverso

trata de requisições da *internet* para o destino de um cliente numa rede interna (LONG; LI, 2012). Na Figura 11 temos um esquema de *proxy* reverso, que provê e controla o acesso proveniente da rede externa(LONG; LI, 2012).

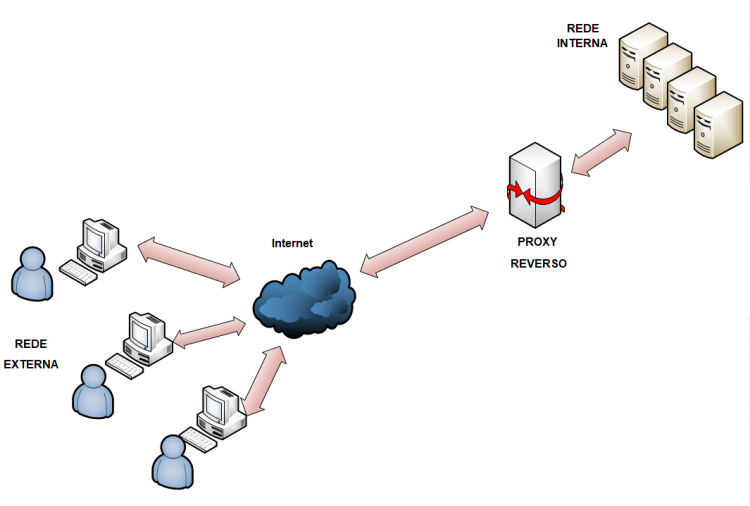

Figura 11 – Esquema de um *Proxy* Reverso (o Autor).

2.4.2 Autenticação, Autorização e Auditoria (AAA)

Na segurança da informação é responsabilidade do administrador de rede ter o controle de acesso as informações de forma eficiente. O controle de acesso vai desde o controle ao acesso físico até as ferramentas de controle de acesso à informação propriamente dita. Uma estratégia de controle de acesso largamente utilizada é a aplicação de protocolos que implementem as funções de Autenticação, Autorização e Auditoria (Authentication, Authorization and Accounting - AAA). Existe uma família de protocolos que oferecem estes três serviços. A autenticação é o processo de verificação de algum tipo de identificação digital do usuário. É o momento em que o cliente do software prova quem realmente ele é.

Depois de validada a identidade do usuário ocorre o processo de autorização que concede determinados privilégios pertinentes a esse usuário. Nesse momento atribui-se ao usuário as permissões de acesso ao serviço para o qual foi feito o *login*.

Por fim o processo da auditoria ocorre depois da autenticação e da autorização, e tem por objetivo obter informações sobre as atividades que o usuário fez e as armazena num servidor chamado servidor de *log* (METZ, 1999).

2.4.3 Aplicações *Multitenant*

Atualmente estão se popularizando os serviços de computação em nuvem, juntamente com o conceito de Software as a Service (SaaS). Nesse caso, o usuário paga pela utilização do *software*, invés de comprar o *software* propriamente dito. A partir desse

momento surge o termo *multitenant*, frequentemente utilizado por provedores de nuvem. Por definição um *software* ou uma arquitetura *multitenant* é criada para trabalhar com vários "inquilinos". Dizer que uma arquitetura é *multitenant* significa que se pode atender na mesma instância diversos clientes de origens e níveis diferentes. Na Figura 12 temos um esquema diferenciando uma aplicação *single-tenant*, em que cada usuário tem acesso a sua aplicação, que por sua vez tem um banco de dados separado. Já na aplicação *multitenant*, temos diversos usuários (inquilinos) compartilhando uma aplicação que por sua vez compartilha os bancos de dados (VELKOSKI et al., 2013).

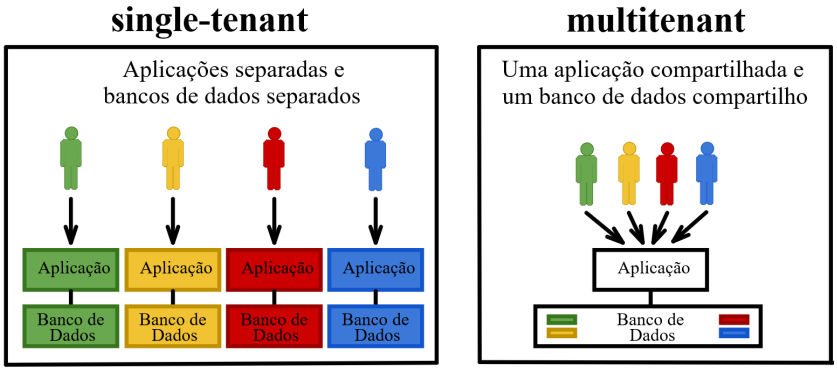

Figura 12 – Esquema *single-tenant* e *multitenant* (o Autor).

2.5 Considerações Finais

Este capitulo apresentou um conjunto de definições e conceitos importantes que compõe o referencial teórico necessário para fundamentar o projeto proposto na tese. Falar a importancia da rede ICN.

3 TRABALHOS RELACIONADOS

Neste capítulo apresentamos uma síntese dos principais trabalhos relacionados diretamente com a pesquisa, que ajudaram na definição das escolhas e do caminho a ser seguido para resolver o problema proposto. Também apresentamos um breve resumo de cada artigo relevante para a pesquisa e quais as estratégias utilizadas na arquitetura proposta.

Uma arquitetura chamada *AuthFlow* utilizando o IEEE 802.1X e o EAP (*Extensible Authentication Protocol*) foi prosposta em (MATTOS; DUARTE; PUJOLLE, 2018), conforme ilustrado na Figura 13. Tal proposta considera três componentes principais: controlador *Openflow*, um servidor autenticador, e um servidor RADIUS. Nessa proposta uma VM (*Virtual Machine*) querendo se autenticar envia uma mensagem IEEE 802.1X para o servidor RADIUS e a resposta é enviada para a análise e tomada de decisão no controlador *Openflow*.

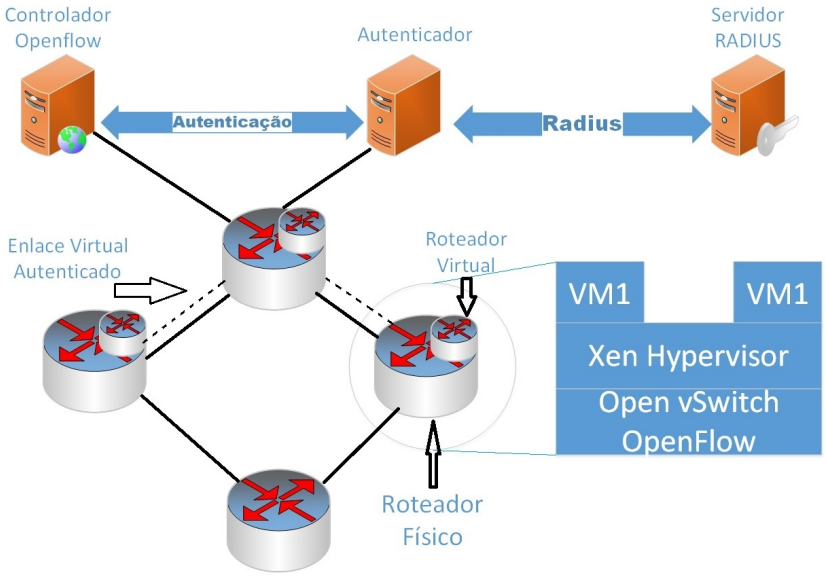

Figura 13 – Arquitetura AuthFlow (MATTOS; DUARTE; PUJOLLE, 2018).

Uma segunda proposta de arquitetura que combina redes orientadas a conteúdo e redes definidas por software para atuar com big data foi apresentada em (YAO et al., 2016). Esta proposta adota um controle centralizado, com o SDN combinado com o *cache* distribuído fornecido pelo CCN. O estudo indica que esse paradigma que muda para a rede centrada em conteúdo tem muitas vantagens, tais como redução da carga da rede e baixa latência. Esse recurso tem impactos significativos no fornecimento de conteúdo aos usuários SDN. O SDN pode ser a solução de uma estratégia de controle no gerenciamento de conteúdo. O SDN pode obter grandes quantidades de dados das diferentes camadas,

com granularidade arbitrária. Usando a análise de big data na camada de controle do SDN, pode-se extrair conhecimento dos grandes volumes de dados, de forma a ajudar o controlador na tomada de decisões. Por exemplo, com a análise de big data, pode-se saber qual conteúdo em um determinado switch tem alta popularidade.

No terceiro artigo (ROWSHANRAD; PARSAEI; KESHTGARI, 2016) são discutidos alguns métodos de implementação do NDN (*Named Data Network*). O NDN é um nome correlato ao CCN, utilizando a estratégia SDN. Especificamente, esta estratégia introduz 4 métodos relevantes para este tipo de implementação:

- Método 1 - Alterando os comutadores *Openflow*
- Método 2 - Uso de funções *Hash* para *hash* de pacotes em campos de cabeçalho IP
- Método 3 - Usando proxy para se comunicar entre o switch OF e o CCNx (ou NDN)
- Método 4- Usando o campo Opção IP de um pacote como o campo de nome

3.1 Método 1

No Método 1 (CARVALHO et al., 2012), denominado *ContentFlow*, os comutadores *Openflow* são modificados para trabalhar com conteúdo, conforme ilustrado na Figura 14. O nó CCN é o comutador Openflow modificado que é programável através do controlador de rede e suporta transações NDN. Este nó inclui FIB, PIT e armazenamento de conteúdo. O nó *ContentFlow* é o controlador que se comunica com nós CCN e o servidor de conteúdo para gerenciar a rede. Os nós do CCN transferem periodicamente para o servidor de contúdo, uma cópia do conteúdo que eles têm, e de suas informações atualizadas. O controlador pode consultar a lista de conteúdo e suas informações no servidor de conteúdo. Isso dá ao controlador a capacidade de ter uma visão global da rede. O servidor de conteúdo também é responsável por tomar decisões sobre a localização do conteúdo na rede e sua duração de armazenamento em *cache* no nó CCN (CARVALHO et al., 2012). O ponto forte da arquitetura é que, embora possa implantar todas as funcionalidades do NDN, ele também suporta as vantagens do SDN, como uma visão global centralizada da rede a partir do controlador. Além disso, ele cria a capacidade de adicionar novos protocolos e serviços à rede, independente da mudança de dispositivos de hardware. Isso é uma grande vantagem, pois o NDN ainda está evoluindo e pode enfrentar muitas mudanças. Entretanto, alterar os *switches Openflow* para criar os nós da CCN depende de alterações de hardware, que são demoradas e envolvem alto custo.

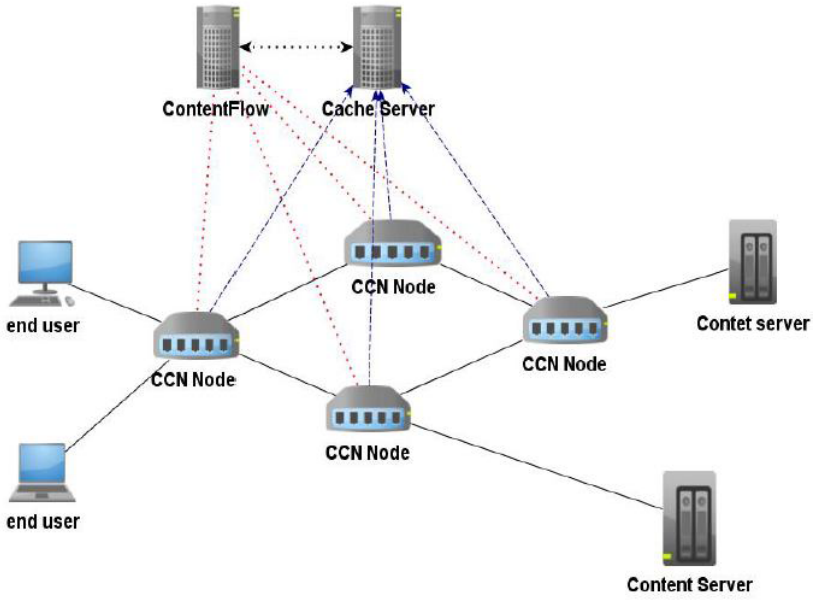

Figura 14 – Arquitetura ContentFlow (CARVALHO et al., 2012).

3.2 Método 2

O Método 2 (OOKA et al., 2013) propõe uma arquitetura baseada em *hash* de nomes de conteúdo.Para implementar esse método, foram utilizados os protocolos UDP (*User Datagram Protocol*) e o *Openflow* 1.0. Os pacotes NDN são encapsulados no payload do pacote UDP. Dois números de porta diferentes são dedicados aos pacotes de dados e de interesse. Isto torna os nós capazes de distinguir entre esses dois pacotes NDN, e também distinguir entre pacotes IP e pacotes NDN. Portanto, os comutadores podem trabalhar com NDN e arquiteturas IP. Os roteadores NDN devem poder executar o LPM (Longest prefix match) nos nomes de pacotes, mas não existem regras no protocolo *Openflow* para correspondência de cargas úteis de pacotes.

Como resultado, os nomes são *hashes* e colocados no campo IP no cabecalho dos pacotes. Como cada parte de um nome seria dividido em um número de 4 bits, um nome pode consistir em 8 partes, no máximo. Ocorre um problema porque os nomes na NDN podem ter comprimentos diferentes, e estudos mostram que 99,9% dos nomes têm menos de 30 partes. Portanto, na maioria dos casos, 8 partes são insuficientes. A probabilidade de colisão também depende do número de bits, pois quanto maior o número de bits para cada parte do nome, menor a probabilidade de colisão. Uma solução melhor é considerar o *Openflow* 1.3 para explorar campos IPv6 ou MAC. Em (OOKA et al., 2013) é proposto o uso do cache de software no controlador como o cache em rede, o que claramente tem um desempenho melhor do que o cache de hardware nos roteadores. Em vez de PIT, FIB e armazenamento de conteúdo, ele considera um sinalizador de estado no controlador para cada pacote (OOKA et al., 2013).

3.3 Método 3

O Método 3 além de ter uma implementação SDN e CCN, tem uma abordagem de resolução do problema de cache citado no Método 2. Para resolver esse problema em (NGUYEN; SAUCEZ; TURLETTI, 2013), uma nova estratégia é proposta para "cache fora do caminho"na NDN, usando funcionalidades SDN. Essa estratégia reduz a replicação de conteúdo. Para integrar SDN e NDN, este artigo propôs um método no qual os comutadores de rede são compostos de um comutador *Openflow* e CCNx que se comunicam entre si através de um proxy. O proxy mapeia os comutadores CCNx e *Openflow* e faz o *hash* dos nomes de conteúdo NDN em campos que podem ser processados por switches *Openflow*. A vantagem desse método é que não há necessidade de alterar os comutadores CCNx ou Openflow e o proxy pode ser atualizado facilmente. O cache de hardware *Openflow* pode ser usado como FIB enquanto o *daemon* CCNx pode ser usado como PIT e armazenamento de conteúdo. Mas o proxy pode diminuir o desempenho da rede devido a operações extras que ele deve executar. Para resolver esse problema, três módulos foram adicionados ao controlador. O primeiro é um módulo de medição que identifica periodicamente o conteúdo mais popular consultando os comutadores. O segundo é um módulo de otimização que encontra a melhor localização para o cache de conteúdo com base em latência e congestionamento. O terceiro é um módulo de deflexão que cria um mapeamento entre o nome do *hash* do conteúdo e as interfaces de saída em que ele foi armazenado em cache. Os resultados da avaliação mostram que o proxy diminui a eficiência em apenas 5% no pior caso (NGUYEN; SAUCEZ; TURLETTI, 2013).

3.4 Método 4

O Método 4, denominado arquitetura CoNet (SALSANO et al., 2013), implementada no banco de testes OFELIA, nesta arquitetura o campo onde iria o endereço IP vai o nome do conteúdo. Devido à incapacidade do OpenFlow em processar opções de IP, o nome também é codificado e marcado nos primeiros 4 bytes do cabeçalho UDP. Portanto, um IP reservado no campo de destino IP é usado para distinguir entre esses dois pacotes. Dois outros IP reservados também são usados para os pacotes de interesse e dados da CoNet (SALSANO et al., 2013). Uma parte detalhada da arquitetura CoNet é mostrada na Figura 15.

Ao receber um interesse no nó de extremidade do sistema, se os dados solicitados não estiverem armazenados em cache em qualquer nó, o interesse será roteado para o destino apropriado que pode ser um nó interno do sistema ou nó de extremidade quando os dados pertencem a um sistema externo. Se os dados solicitados forem armazenados em cache em um servidor de cache, o interesse será roteado para esse servidor. Quando o pacote de dados é roteado para o consumidor, um dos comutadores também envia os

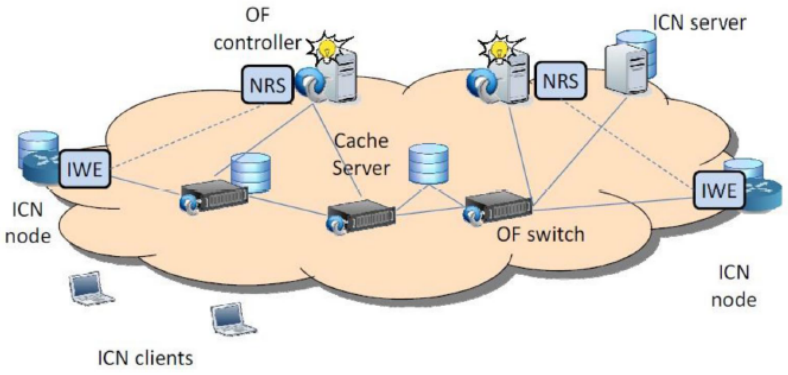

Figura 15 – Arquitetura CoNet (SALSANO et al., 2013).

dados para serem armazenados em cache em um servidor de cache. Isso deve ser feito com a ajuda do controlador (nó NRS). O controlador também é responsável por verificar e garantir que as tags sejam únicas em seus domínios. O CoNet suporta a maioria das funcionalidades da NDN, como assinatura digital para segurança e armazenamento em rede. Também oferece algumas outras funcionalidades, como controle de congestionamento TCP (SALSANO et al., 2013).

O quarto trabalho (SON et al., 2016) utiliza o o CCN trsdicional para transmitir os conteúdos. Uma vez que os pedidos dos clientes chegam ao servidor que possui o conteúdo solicitado, os pacotes de dados são entregues aos clientes pelo caminho reverso usando a PIT e FIB nos roteadores CCN. Enquanto os pacotes de dados são entregues aos clientes, eles são armazenados em cache nos roteadores intermediários para que, se houver as mesmas solicitações posteriormente, os pedidos não precisam ir até o servidor em que o conteúdo solicitado é armazenado. Em vez disso, os clientes podem obter respostas dos roteadores que já armazenaram o conteúdo em cache no armazenamento de Conteúdo (CS). No entanto, pode haver problemas se considerarmos aplicar CCN em uma rede de larga escala. Uma desvantagem da Arquitetura CCN original é que todos os roteadores num ambiente de tranmissão devem encaminhar os pacotes de interesse para todos os roteadores vizinhos.Isso significa que o tráfego de rede se torna incontrolável, pois o tamanho da rede aumenta e o número de solicitações aumenta. Em outras palavras, a utilidade dos recursos da rede diminuiu. O outro problema é que, à medida que o tamanho da rede aumenta, o tempo necessário para os clientes receberem os dados solicitados será maior e toda a rede se torna ineficaz no fornecimento de conteúdo. Neste artigo, é proposto uma arquitetura CCN baseada em SDN e uma nova estratégia de encaminhamento que pode lidar com esses problemas. Cada cluster de roteadores é gerenciado por um controlador SDN local e aplicativo que tenha o índice, contendo informações para interesse e encaminhamento. Os roteadores CCN não são mais usados para direcionar pacotes. A introdução do controlador SDN à rede torna possível enviar pacotes, tanto de interesse quanto de dados, em vez de inundá-los por toda a rede. Isso pode ajudar a reduzir tráfego e melhorar a eficiência em

termos de fornecimento de pacotes de dados.

No quinto artigo (LIU; WADEKAR, 2016) trouxe um projeto e desenvolvimento do SDAR, uma Plataforma de Controle de Roteamento Intra-Domínio Definido por Software em NDN, que pode gerenciar roteadores em toda a rede, o mesmo cita que no NDN, os roteadores encaminham pacotes de interesse e mantêm informações de estado para os pacotes de dados correspondentes o caminho inverso.

O recurso que os roteadores NDN mantêm informações de encaminhamento completas através de dados de interesse simétricos troca permite que o plano de encaminhamento possa detectar e recuperar de falhas ou alterações na rede explorando caminhos alternativos múltiplos sem assistência do plano de controle.

Como resultado, esse recurso de encaminhamento adaptável relaxa o requisito de protocolos de roteamento originalmente projetado para reagir à dinâmica de curto prazo da participação de pares e, portanto, melhora significativamente a escalabilidade e estabilidade da rede. Por outro lado, o recurso exclusivo abre portas para uso de outros protocolos de roteamento anteriormente considerados inviáveis ou ineficazes para redes IP, como protocolos de roteamento de domínio centralizado. O SDN e os padrões relacionados foram criados para resolver as limitações das redes atuais, como complexidade, políticas inconsistentes e incapacidade de dimensionar a dependência do fornecedor.

Observa-se que o SDN é uma definição do codificador em vez da tradicional orientação do *Openflow* SDN. Na verdade, uma das principais contribuições é usar uma abordagem não-*Openflow* para implementar o a gestão roteamento centralizado.

Encontram-se três grandes desafios ao implementar SDAR da seguinte forma: (1) Design do modelo de comunicação: estrutura foi motivada pelo SDN em redes IP, mas o SDAR modelo de comunicação, que estará dentro do contexto da NDN, não pode ser transformado diretamente do IP tradicional baseado em modelo *push* de comunicação. (2) Integração da plataforma de roteamento: o segundo desafio é projetar e desenvolver uma abordagem genérica e plataforma de roteamento centralizado a partir do zero. A plataforma será capaz de lidar com o roteamento de entrada e saída tráfego, para armazenar informações de roteamento em toda a rede em várias estruturas de dados eficientes, para fornecer interfaces uniformes para algoritmos de roteamento diferentes e para calcular rapidamente e vários caminhos mediante alterações de topologia e link. Apesar o trabalho existente tentou integrar o NDN ao *Open-Based* baseado em IP. Estrutura de fluxo, essas funcionalidades centralizadas não foram implementadas a partir de uma perspectiva pura centrada em dados no NDN comunidades de pesquisa. (3) Detecção de Link e manipulação de dinâmica de rede: não há um protocolo centralizado de descoberta de camada de link orientado para NDN, como o *Link Layer Discovery Protocol* (LLDP) em SDN, para detectar se um link está vivo ou nem um protocolo que possa lidar com mudanças de topologia e vincular as mudanças de custos de forma dinâmica e inteligente em um ambiente

centralizado. Portanto, o terceiro desafio é como obter o status dos links entre vizinhos e como lidar com várias dinâmicas de rede a partir do zero. Como tal, implementa-se um protocolo de detecção de link que pode identificar falhas de link ou nós.

No sexto e último artigo (CHARPINEL et al., 2016) é apresentado uma combinação entre SDN e CCN. O projeto é chamado de SDCCN (*Software Defined Content Centric Network*), e possui suporte a roteadores e cache com programação, e um ambiente de trabalho para a realização de diversos experimentos. A proposta têm por objetivo eliminar o mapeamento do nome do conteúdo e seu identificador, trabalhando diretamente com os nomes de conteúdos utilizando o protocolo POF (*Protocol Oblivious Forwarding*). Foi utilizado o ambiente de prototipação chamado Mininet nos testes executados. A arquitetura proposta é apresentada na Figura 16, e possui os seguintes elementos:

- *Switch* - Um comutador que foi alterado para encaminhar conteúdos nomeados, e que tem em sua programação, uma FIB, nesse caso chamada de Tabela de Encaminhamento (TE), responsável pelo encaminhamento dos pacotes de interesse. Uma tabela com as regras estabelecidas para o manuseio da cache (TRC), Tabela de Conteúdo que faz o papel da *content store* e uma tabela de interesses (TI) chamada de PIT no CCN.
- Controlador – É um dispositivo que tem o poder de alterar o conteúdo das tabelas supracitadas, assim como obter as informações pertinentes em qualquer ponta da comunicação.

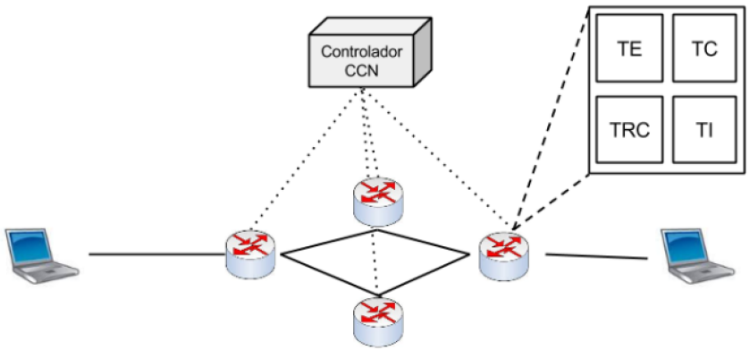

Figura 16 – Arquitetura proposta SDCCN (CHARPINEL et al., 2016).

O ponto inicial é o cliente pedir um conteúdo através de um pacote de interesse. Após o pedido o primeiro switch verifica em sua tabela de conteúdo se tiver o mesmo é enviado ao cliente, senão é encaminhado ao próximo switch e assim sucessivamente até o conteúdo ser encontrado ou não. O diferencial dessa proposta comparado a arquitetura original CCN (JACOBSON et al, 2009) é que existe uma estratégia de otimização no encaminhamento dos pacotes de interesse, assim como alteração na estratégia de controle

de armazenamento de conteúdo (cache). A solução para essas otimizações foi a implantação de uma camada de estratégia dentro do modelo TCP/IP implementada em SDN.

Trabalhos	Fluxo de SDN	Cache de conteúdo	Segurança	Contribuição
(MATTOS; DUARTE; PUJOLLE, 2018)	S	S	Servidor de autenticação	Implementação de um método de autenticação para redes SDN
(YAO et al., 2016)	S	S	SEM	Implementação de controle de big data usando uma estratégia SDN combinada com a CCN.
(CARVALHO et al., 2012)	S	N	SEM	Discussão das estratégias de implementação do SDN com a CCN – Estratégia 01 - Mudar os switches openflow
(OOKA et al., 2013)	S	S	SEM	Discussão das estratégias de implementação do SDN com a CCN – Estratégia 02 Utilizando funções de hash de cabeçalho IPs
(NGUYEN; SAUCEZ; TURLETTI, 2013)	S	N	SEM	Discussão de estratégias de implementação de SDN com CCN - Estratégia 03 - uso de proxy para comunicação entre CCN e SDN
(SALSANO et al., 2013)	S	S	Assinatura digital no arquivo	Discussão das estratégias de implementação do SDN com a CCN – Estratégia 04 -Utilização do campo Opção no pacote IP como nome do arquivo CCN.
(SON et al., 2016)	S	S	SEM	Implementação de uma estratégia de melhoria do controle de inundação de interesse da rede
(LIU; WADEKAR, 2016)	N	N	SEM	Implementação de uma arquitetura SDAR com um novo protocolo de roteamento.
(CHARPINEL et al., 2016)	S	N	SEM	Implementação de uma arquitetura SDN e CCN utilizando o protocolo POF
Arquitetura Proposta	S	S	Autenticação do usuário sem usar um servidor de autenticação separado	Controle da entrega de conteúdo usando as configurações do controlador

Tabela 3 – Quadro resumo do referencial teórico - Fonte: O Autor

Por fim, o trabalho apresentado em (GULATI et al., 2021) relata uma utilização contemporânea do uso das tecnologias SDN e CCN. O trabalho aborda as redes veiculares para aplicações em saúde. A Internet dos Veículos de Saúde (IoHV) é um paradigma em evolução que conta com o transporte inteligente envolvendo ambulâncias e outros veículos de saúde, para se conectarem através da Internet, especialmente em casos como testes de COVID19 e rastreamento de contatos. No entanto, as aplicações de IoHV não são semelhantes a outras redes e, portanto, trazem novos desafios técnicos devido a alta mobilidade e as rápidas mudanças de topologia. Portanto, para lidar com esses desafios o trabalho propõe um novo esquema que encontra um caminho ótimo. Usando o caminho ideal, o pacote de interesse é encaminhado entre os veículos de saúde em um cenário de cidade inteligente. O esquema proposto maximiza a probabilidade de encontrar os dados solicitados enquanto minimiza a latência. Como o esquema proposto não envolve a transmissão de pacotes de interesse, o problema de congestionamento da rede no CCN pode ser resolvido em grande parte. Para facilitar a interoperabilidade de dispositivos de saúde heterogêneos na rede e melhorar a flexibilidade de roteamento, um controlador SDN é implantado no cenário IoHV. Ele mantém as informações globais da rede em suas tabelas de fluxo. Além disso, para obter um tempo de resposta mais rápido e menor latência no ambiente IoHV, são utilizados dispositivos de borda em vez da nuvem central.

A Tabela 3 apresenta um resumo com as características relevantes do referencial teórico, que investigamos para o desenvolvimento desta proposta.

3.5 Considerações Finais

Este capitulo foi constituido por uma série de trabalhos que contribuiram com a tese aqui descrita. Tais artigos serviram de estado da arte e foram extremamente úteis no desenvolvimento do experimento prático simulado.

4 METODOLOGIA

Neste capítulo apresentamos uma breve descrição do conjunto de técnicas e processos empregados para a realização da pesquisa. Este capítulo foi estruturado em quatro seções. Na primeira seção descreve a caracterização da pesquisa em termos da fundamentação bibliográfica. Na segunda seção apresentamos a estratégia de pesquisa e a metodologia utilizada. Na terceira seção apresentamos as ferramentas utilizadas na implementação da simulação. Na quarta seção apresentamos as considerações finais.

4.1 Caracterização da Pesquisa Bibliográfica

Esta pesquisa seguiu os preceitos de revisão bibliográfica exploratória, por meio de pesquisa desenvolvida a partir de material já elaborado. Primeiramente foi executado a leitura exploratória de todo material selecionado nas bases de dados previamente escolhidas. A partir dessa seleção foi feita uma leitura seletiva dos artigos relevantes para a pesquisa. Por fim, realizamos o registro das informações extraídas das fontes analisadas, levando em consideração os artigos que ajudaram a responder o problema proposto na pesquisa.

A partir dessa análise, os artigos pesquisados contribuíram de forma relevante para a resolução do problema proposto na tese. O artigo (MATTOS; DUARTE; PUJOLLE, 2018) forneceu diversos subsídios para nossa proposta, especificamente a inserção do uso do controlador POX. As abordagens pesquisadas resolviam em partes o problema proposto nessa tese e de forma separada. O artigo (YAO et al., 2016), assim como diversos outros pesquisados, revelaram ser relevante a escolha do tema de combinação da rede SDN com a CCN, para a utilização em alguns segmentos de vídeo sob demanda. O artigo (ROWSHANRAD; PARSAEI; KESHTGARI, 2016) foi relevante pois ajudou na escolha da arquitetura e a implementação do projeto prático utilizado na tese. O artigo (SON et al., 2016) exibiu uma nova estratégia de encaminhamento e roteamento, ajudando a abrir novos horizontes nesse tema. O quinto artigo (LIU; WADEKAR, 2016) também exibiu melhorias no roteamento e na forma de como encaminhar os pacotes. E o último artigo (CHARPINEL et al., 2016), entre outros não citados, foram relevantes para a definição da arquitetura básica e o ambiente de rede.

4.2 Estratégia de Pesquisa

O primeiro desafio a ser superado foi a escolha de um ambiente que simulasse uma rede de entrega de vídeo sob demanda, e que fizesse a entrega do conteúdo através de um serviço CCN, através do controle de fluxos com a tecnologia SDN. Depois do primeiro

desafio superado, a questão seria controlar a entrega do conteúdo para os usuários dentro dessa rede. A partir da revisão bibliográfica, o trabalho proposto consiste numa estratégia de controle de acesso num ambiente de simulação constituído por uma rede de entrega de conteúdo com o controle de fluxos SDN. Essa experimentação é composta por um ambiente simulado usando softwares nativos das duas tecnologias citadas. Foi implementada uma arquitetura de redes simulando um ambiente de entrega de vídeo sob demanda para o levantamento dos dados e avaliação dos resultados. A metodologia de desenvolvimento do trabalho foi definida em 12 etapas, conforme descrito a seguir:

1. Estudo e avaliação da revisão bibliográfica nas bases definidas em conjunto com o orientador.
2. Levantamento do estado da arte dessas tecnologias e soluções para os problemas que foram elencados na pesquisa.
3. Estudo das ferramentas nativas de cada uma das tecnologias contidas na proposta (MININET para o SDN e CCNX para o CCN).
4. Avaliação das outras possibilidades existentes para compor o ambiente de simulação.
5. Projeto e implementação da ferramenta proposta.
6. Definição dos requisitos e das ferramentas que irão compor o ambiente de testes.
7. Definição de métricas e critérios de avaliação.
8. Planejamento dos experimentos.
9. Implementação do cenário de avaliação de acordo com as premissas definidas no quesito 6.
10. Execução dos experimentos e organização dos dados.
11. Avaliação dos resultados sob a perspectiva dos objetivos da pesquisa.
12. Documentação e análise dos resultados.

4.3 Ferramentas de Simulação

Através dos subsídios adquiridos com a pesquisa bibliográfica, definimos a escolha de um ambiente simulado. Esse ambiente simulado traria a facilidade de escalabilidade e a não necessidade de equipamentos físicos com as tecnologias SDN e a CCN. Assim se chegou ao ambiente chamado Mininet (ONF, 2021) (Mininet, 2021). O Mininet fornece um ambiente de teste e desenvolvimento virtual para redes definidas por software (SDN).

O Mininet habilita prototipagem rápida de redes definidas por software com teste de topologia complexo sem a necessidade de conectar uma rede física e vários desenvolvedores simultâneos para trabalhar de forma independente na mesma topologia.

As redes Mininet executam códigos reais, incluindo aplicativos de rede Unix/Linux, bem como o kernel Linux real e a pilha de rede. O Mininet fornece uma API Python extensível para criação e experimentação de redes. Ele é lançado sob uma licença BSD Open Source permissiva e é ativamente desenvolvido e apoiado pela comunidade de entusiastas de redes e SDN. Um ponto importante desse simulador é a possibilidade de se utilizar um software chamado wireshark, que é conhecido e amplamente utilizado como analisador de protocolos de redes. Ele permite que se veja o que está acontecendo na rede em diversos níveis. Esse tipo de software registra a duração de fluxos, o registro de pacotes e diversas outras diversas informações sobre o SDN. Conforme demonstrado na figura 17, o mininet tem hosts que trabalham de forma isolada e que possuem interfaces, portas e tabelas de roteamento para serem utilizadas. Os links são emulados e podem ser configurados com taxas de tráfego diferentes e cada switch (Open Vswitch ou linux default Brigde) é utilizado para comutar os pacotes que trafegam na rede. O Mininet é útil para o desenvolvimento em ensino e pesquisa, pois podemos interagir com ele através de CLI e GUI, ou seja, com uma interface de linha de comando ou uma interface gráfica para usuário (ONF, 2021) (Mininet, 2021).

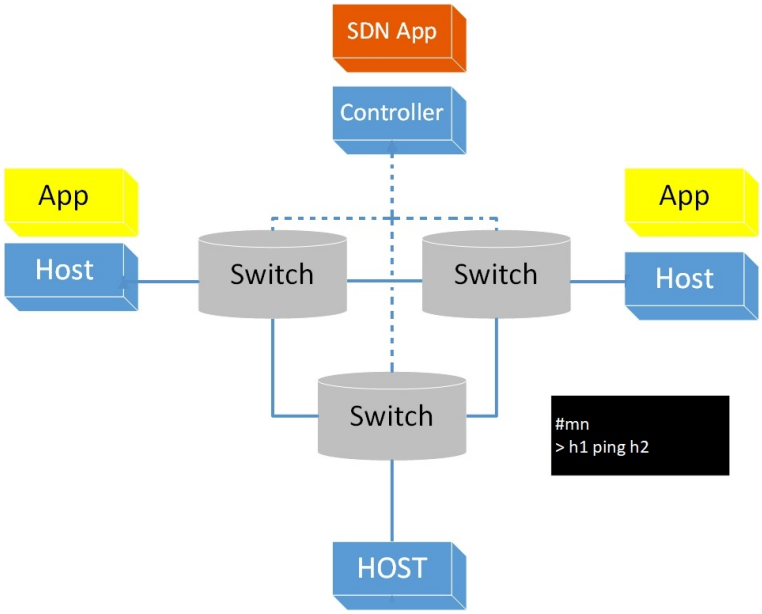

Figura 17 – Representação de uma rede no Mininet (O Autor).

Depois da instalação do Mininet foram criadas diversas topologias de rede, tanto para aprendizado quanto para os testes reais implementados para a tese. Essas topologias são implementadas em scripts de linguagem de programação *Python*. Entretanto o mininet não trabalha de forma nativa com o CCN, dessa forma é necessário a instalação de um

pluguin que habilita características do protocolo CCN as máquinas virtuais do Mininet. A implementação de referência da CCNx fornece as bibliotecas e componentes necessários para construir e executar aplicativos que utilizam e demonstram os protocolos básicos do CCNx para fins experimentais e de pesquisa. Depois da instalação do pluguin as máquinas virtuais enviam os pacotes de interesse e de dados como acontece numa rede CCN (ProjectCCNx, 2021) (Cicn, 2021). Nesse momento o ambiente base toma forma e fica preparado para receber os experimentos dessa tese. O último passo foi a instalação de um controlador SDN chamado POX, que tem maior flexibilidade de programação pois seu objetivo de uso tem viés didático. POX é um controlador de rede de código aberto baseado em Python(SDN). O POX é usado para desenvolvimento mais rápido e prototipagem de novas aplicações de rede. O controlador POX vem pré-instalado com a máquina virtual mininet. Usando o controlador POX, podemos transformar dispositivos de fluxo aberto em dispositivos de hub, switch, balanceador de carga e firewall. O controlador POX possibilita uma maneira fácil de executar experimentos OpenFlow/SDN. O POX pode ser passado por diferentes parâmetros de acordo com topologias reais ou experimentais, permitindo que possamos executar experimentos em hardware real, *testbeds* ou em emulador de rede. O POX é um tipo de controlador SDN que fornece uma estrutura de comunicação com os *switches* SDN usando o protocolo *openflow* (KAUR; SINGH; GHUMMAN, 2014).

Desta forma, temos um ambiente especificado e preparado para todos os testes realizados na proposta, como por exemplo as estratégias de controle de acesso aos dados CCN no controlador POX.

4.4 Considerações Finais

Nesse capítulo foram descritos a passos principais da metodologia utilizada. Primeiramente foi caracterizado o escopo da pesquisa, depois como se deu a estratégia das pesquisas elucidando cada uma das etapas necessárias para se atingir os objetivos. Após isso houve um detalhamento da estratégia da implementação prática, descrevendo o ambiente completo utilizado nos testes práticos.

5 ARQUITETURA PROPOSTA

Redes de fornecimento de conteúdo (Content Delivery Networks - CDNs), foi um termo criado no final da década de 90 para descrever computadores em rede (Internet), que colaboram de um modo transparente para fornecer conteúdo, normalmente grandes conteúdos multimídia para clientes finais. Esse tipo de rede tem diversas vantagens, como a realização de downloads em paralelo, e a possibilidade de o navegador já ter o arquivo em cache, o que ajuda muito a diminuir o consumo de tráfego do servidor principal e das CDNs. Um bom exemplo do êxito dessa estratégia é que ela é aplicada por grandes provedores de serviços de informação, como o Google, a Microsoft e a Amazon, que costumam ter latência e taxa de transferência muito boa.

Porém essa maneira de resolver a entrega de conteúdo tem algumas desvantagens tais como: O conteúdo fica centralizado em alguns servidores replicados, um problemas nas empresas fornecedoras dessas redes temos lentidão e mal funcionamento das redes sociais por exemplo e usar um CDN não confiável poderá tornar o site perigoso para os usuários que o visitam. Entretanto, o maior problema dessa abordagem é que o conteúdo é replicado ao nível da camada de rede, isto é, o conteúdo vai para os servidores de conteúdo espalhados pela rede, próximos as grandes concentrações de usuários. Como os servidores de conteúdo são identificados pelos seus endereços IP, é possível dizer que a replicação é feita ao nível de rede. A escolha do servidor de conteúdo é feita no momento que o usuário resolve o nome DNS para um endereço IP. Nesse caso, dependendo da região onde está localizado, um endereço IP diferente é escolhido. Observe que dependendo da sua localização, o usuário será designado sempre para o mesmo servidor, independente do conteúdo que esteja solicitado.

Uma abordagem alternativa é fazer a replicação ao nível de aplicação. Nesse caso a granularidade é a informação, e informações diferentes podem vir de servidores de conteúdo diferentes. Isso dá muito mais flexibilidade para a abordagem. Isso permite também que a replicação seja feita de forma incremental, na medida em que os usuários solicitem o conteúdo.

A abordagem proposta também leva em consideração o suporte a multitenant, ou seja, múltiplos inquilinos. Os inquilinos ou nesse caso os fornecedores de serviços de origens e naturezas diferentes, que em conjunto fornecem um produto final ao usuário. Neste sentido, o objetivo desta pesquisa é combinar o paradigma SDN numa arquitetura CCN e propor um esquema de roteamento centralizado com uma estratégia de controle de acesso ao conteúdo usando um proxy SDN.

5.1 Framework SDN/CCN

Esta pesquisa possui como foco um tipo específico de rede CDN, ou seja, a rede CDN de entrega de vídeo sob demanda. Neste sentido propomos um framework que tem a rede CCN como base e a redes SDN adicionando recursos importantes para melhorar o serviço de entrega conteúdo, mas especificamente melhorar o processo de entrega do vídeo sob demanda. As duas tecnologias têm demonstrado vantagens que justificam suas escolhas.

Em primeiro lugar, CCN foi considerado como uma das arquiteturas mais promissoras para a distribuição eficiente de conteúdo na Internet. Mudando o paradigma de centralizar a rede no host, para centralizar no conteúdo, muitas vantagens são vistas, tais como redução da carga da rede e baixa latência (WANG et al., 2016).

A principal característica do CCN é o armazenamento de cache ao longo da rede, o que resulta em impactos significativos no fornecimento de conteúdo aos usuários. Já o SDN pode trazer benefícios ao modelo CCN, como a construção de regras que ajudem a achar caminhos alternativos para se fazer o balanceamento de carga e também a validação do conteúdo pelo usuário e vice-versa. Isso significa que fornecedor de conteúdo terá o aumento de controle sobre quem está acessando o conteúdo, mesmo centralizando este controle.

O paradigma SDN na arquitetura proposta neste trabalho está representado principalmente pela capacidade de decisão centralizada e não distribuída pelos nós da rede e acesso ao controle dos usuários que acessam o conteúdo. Dessa maneira a tecnologia CCN padrão terá as funcionalidades de ter informações sobre balanceamento de carga e validação dos usuários, sendo implementadas pelo SDN.

Na Figura 18 apresentamos o framework proposto. A camada de aplicação é responsável pelo cálculo das rotas, pela criação de uma tabela de roteamento, pelo balanceamento de carga e pelo controle de acesso de usuários. A camada de controle fica responsável pela descoberta da topologia e controle de conteúdo, assim como a construção da tabela de fluxo que será enviada para o proxy reverso, que funcionará como um controlador alternativo. Por fim, a camada de infraestrutura que é responsável pelo cache e pela entrega de conteúdo aos usuários finais.

5.2 Arquitetura Proposta

A capacidade de serviço das redes CDN têm sido levada ao extremo devido ao aumento significativo de tráfego de vídeo (BALACHANDRAN et al., 2013). Se levarmos em conta também o crescimento das taxas geradas pelas redes sociais, essa situação piora ainda mais. Para minimizar esse problema, tem sido organizada federações de CDN, que

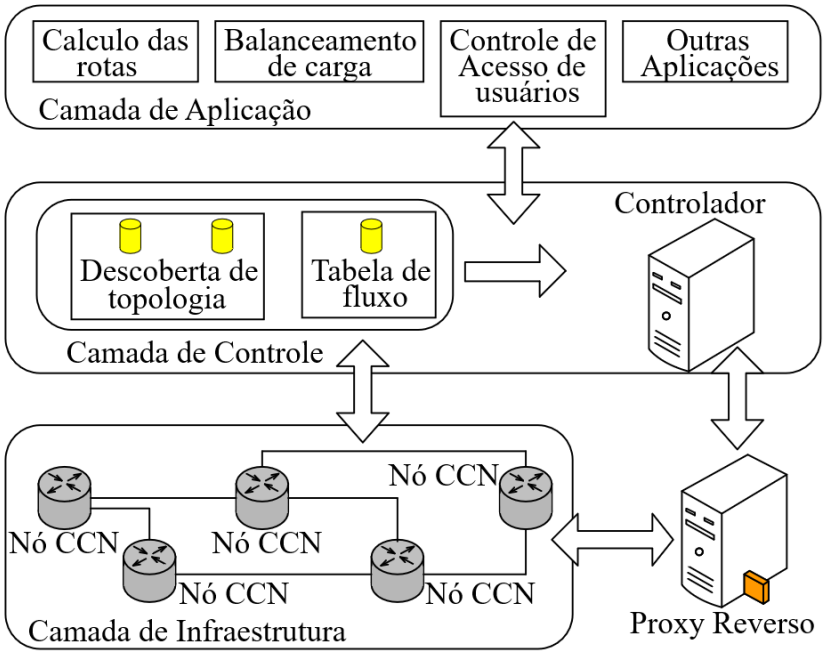

Figura 18 – Framework Proposto (O Autor).

tem a capacidade de direcionar solicitações de conteúdo para servidores mais próximos dos clientes. Porém o problema de escalabilidade da rede CDN tende a crescer à medida que ocorra o aumento da quantidade de vídeo e numa perspectiva de crescimento do número de dispositivos moveis. A construção de uma rede CDN é bem complexa e questões estratégicas tem que ser definidas antes da implementação de uma rede CDN (WANG; SHARMAN; RAMESH, 2008), tais como:

- Qual o número de servidores de cópia deve ser escolhido e qual sua localização (QIU; PADMANABHAN; VOELKER, 2001) (DOAN; BAJPAI; CRAWFORD, 2020b);
- Quais conteúdos serão replicados e quais servidores receberão a cópia dos conteúdos (QIU; PADMANABHAN; VOELKER, 2001) (DOAN; BAJPAI; CRAWFORD, 2020b);
- Que tipo de plano deve ser seguido para se manter o conteúdo dos servidores de cópia de forma consistente. Pode-se utilizar as seguintes estratégias: time-based consistency, value-based consistency e orderbased consistency (BHIDE et al., 2002). A forma de atualização pode ser: pull-based, push-based e híbrida (BHIDE et al., 2002) (DOAN; BAJPAI; CRAWFORD, 2020b);
- Qual a estratégia de escolha do servidor de cópia assim como redirecionar para outro servidor se for preciso. Um algoritmo de roteamento de requisição pode ser adaptativo ou não-adaptativo. Além disso, pode ser transparente ou não transparente (CARDELLINI; COLAJANNI; YU, 2003) (DOAN; BAJPAI; CRAWFORD, 2020b).

A arquitetura proposta é composta pelos seguintes componentes: Controlador SDN, Proxy SDN, Nó intermediário, Publicador de Conteúdo e o Cliente, conforme apresentado na Figura 19.

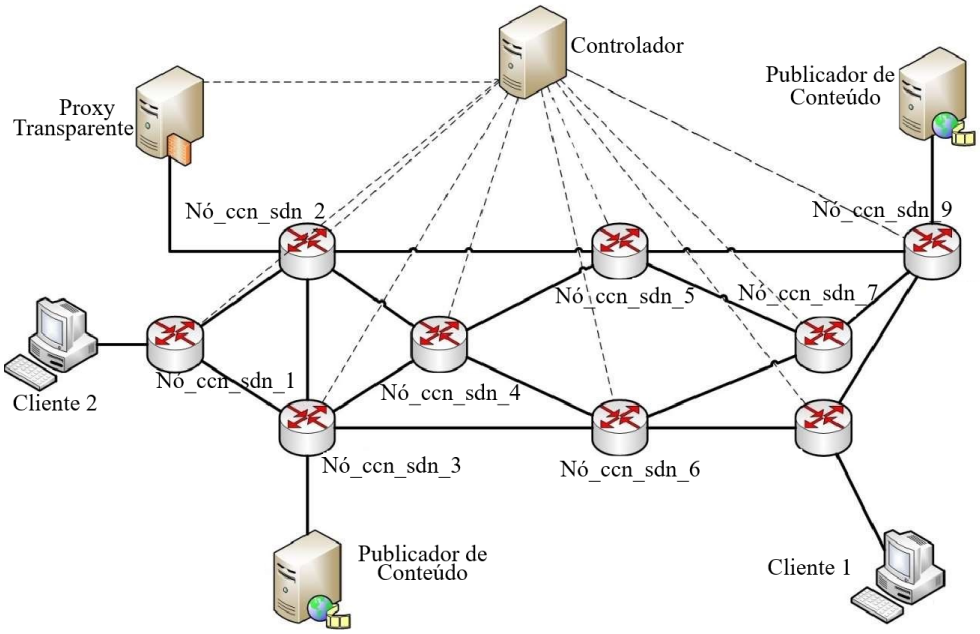

Figura 19 – Elementos da Rede (O Autor).

5.2.1 Controlador SDN

O controlador SDN é o elemento central de uma rede SDN. É o elemento que tem como objetivo coordenar a troca de mensagens com todos os outros elementos programáveis desse tipo de rede, disponibilizando uma visão centralizada da rede. No caso especifico dessa proposta, o controlador SDN será responsável por diversas funções:

- Descoberta de topologia.
- Enviar as tabelas de fluxos a todos os dispositivos intermediários que compõe a infraestrutura de rede.
- Atualizar novos fluxos.
- Delegar ao *proxy* tarefas de controle de acesso dos usuários.
- Informar aos *nós_ccn_sdn* qual será o *proxy* reverso que deverá acessar.

5.2.2 Nós Intermediários

Na abordagem utilizada pelas redes centradas no conteúdo, os nós intermediários se adaptam para armazenar o conteúdo ao longo da rede. Na etapa 1 da Figura 20 temos

um publicador de conteúdo e três usuários. Na etapa 2 o usuário 1 faz uma requisição de um conteúdo. Na etapa 3 ocorre a busca do conteúdo até o publicador de conteúdo. Na etapa 4 o usuário 1 recebe o conteúdo. Na etapa 5 fica demonstrado que o conteúdo ficou armazenado no nós intermediários A, B e D. Na etapa 6 ocorre outra busca pelo mesmo conteúdo só que agora feita pelo usuário 2. A etapa 7 demonstra que o conteúdo buscado é encontrado no nó intermediário B e por fim a etapa 8 demonstra que o conteúdo ficou armazenado nos nós intermediários A, B, D e E.

Na arquitetura proposta, os nós intermediários, chamados nós_ccn_sdn, tem características de ambas as arquiteturas de rede, tanto SDN como CCN. Nesta arquitetura proposta o nó intermediário terá duas funções distintas:

- Trabalhar como um nó CCN transmitindo interesses e conteúdos enviados na rede.
- Trabalhar como um comutador SDN, que são dispositivos de encaminhamento, roteadores, comutadores ou software especializados em encaminhar pacotes.

5.2.3 Proxy SDN

O proxy SDN é um tipo especial de controlador, que tem como especificidade atuar como um intermediário entre um comutador SDN e um Controlador SDN. Isso significa que ele atua como um mediador entre as requisições de um controlador e as ações de um comutador. Desta maneira o proxy SDN permite que a rede seja dividida em partes e tenha um controle de certa forma distribuído, pois quando um controlador SDN repassa um comando para o comutador SDN, ele passa antes pelo proxy que determina se o comutador tem autorização ou não para realizar esta ação. Essa interação pode ser visualizada na Figura 21.

5.2.4 Publicador de Conteúdo

O procedimento de requisição de um conteúdo no CCN é parecido com modelo de requisição-resposta que é usado no protocolo HTTP, no qual componentes, chamadas publicadores de conteúdo, promovem conteúdos na rede e os usuários, denominados consumidores, requisitam os mesmos. Os nós publicadores de conteúdo são elementos que atuam como fonte de dados permanentes, que respondem aos interesses enviados pelos clientes relacionados aos segmentos de dados desejados.

5.2.5 Cliente

Diferente das redes tradicionais, a rede CCN provê aos usuários (consumidores/clientes) a possibilidade de requisição de conteúdos através de nomes, sem a preocupação de onde estão armazenados. Os nós clientes são elementos da rede que desejam obter

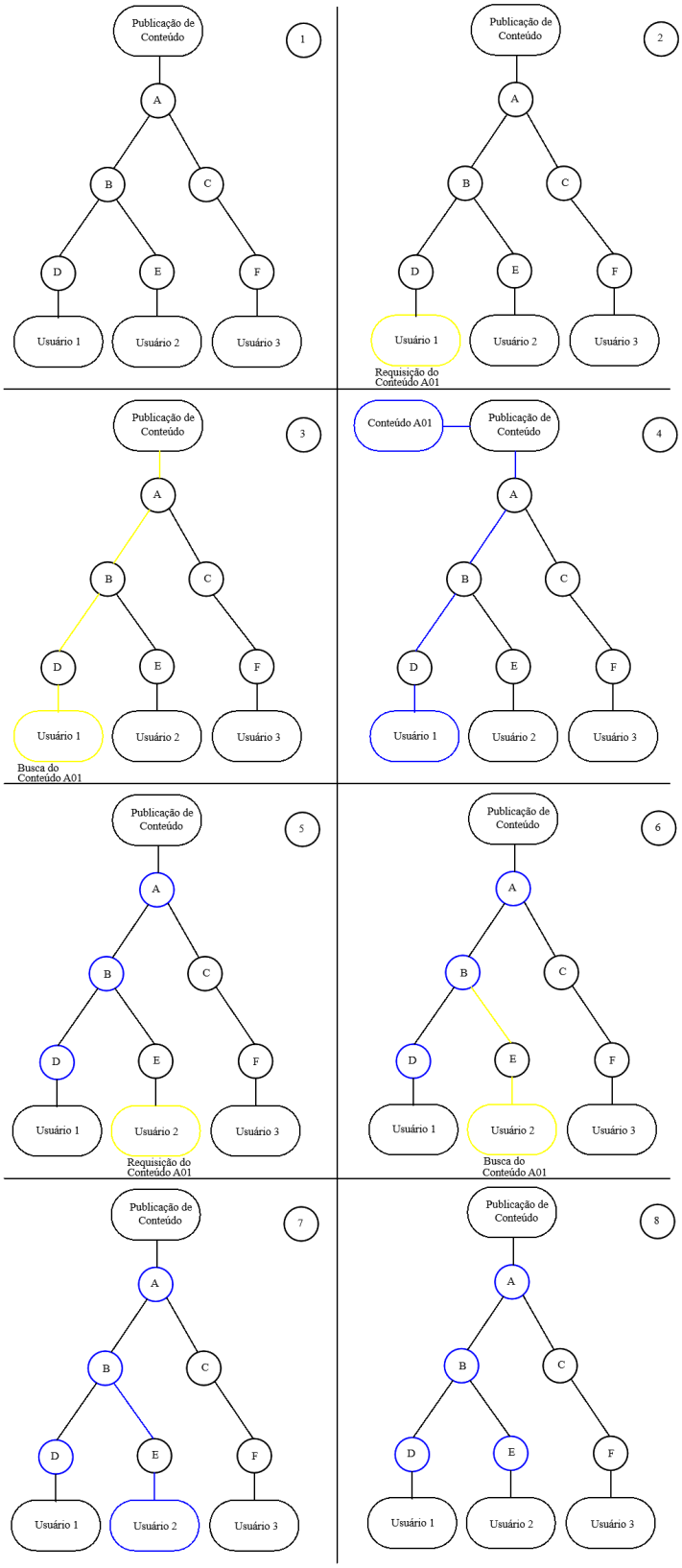

Figura 20 – Funcionamento da requisição de conteúdo e busca de um mesmo conteúdo por dois usuários diferentes (O Autor)

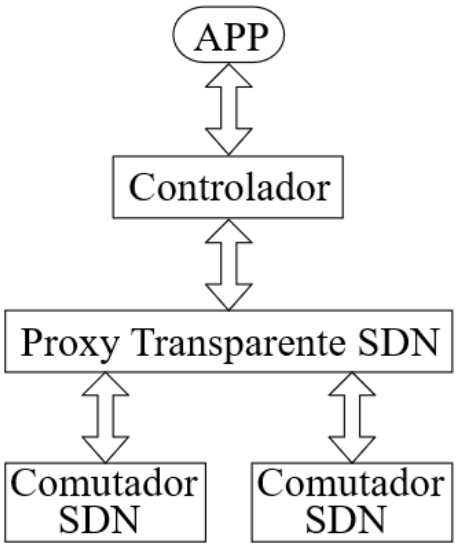

Figura 21 – Proxy SDN (O Autor).

uma cópia de um determinado conteúdo ou até mesmo um pedaço específico dele (*chunk*). Os nós clientes são responsáveis pelo reenvio de pacotes de interesse para conteúdo que por ventura não forem entregues ou cujas cópias entregues sejam por qualquer motivo invalidadas.

5.3 Etapas de Funcionamento da Arquitetura

Nesta seção apresentamos um esquema de roteamento com balanceamento de carga baseado em SDN para CCN, com controle de acesso ao conteúdo, descrevendo todas as mensagens de controle usada pela alteração do protocolo. Primeiramente detalhamos a construção da topologia da rede que permite ao controlador descobrir os nós envolvidos no processo, criar uma topologia, vincular os nós e realizar o roteamento. Depois descrevemos como é feito a etapa de encaminhamento de conteúdo e autenticação de cliente, descrevendo de forma detalhada como o controlador pode preencher a tabela FIB de cada nó. Finalmente, apresentamos as mensagens que o controlador envia atualizando conteúdo e como é feito o balanceamento de carga pelo proxy SDN.

5.3.1 Descoberta de Topologia

Esta etapa é iniciada pelo controlador que primeiramente envia uma mensagem para todos os nós da rede. Essa mensagem descrita na Figura 22 é propagada pela rede e tem por objetivo vincular os nós da rede ao controlador.

A mensagem *Control_announce* tem um parâmetro único que identifica o controlador. Nesse caso foi usado o maior endereço MAC das suas *ifaces* (*Controller_Id*).

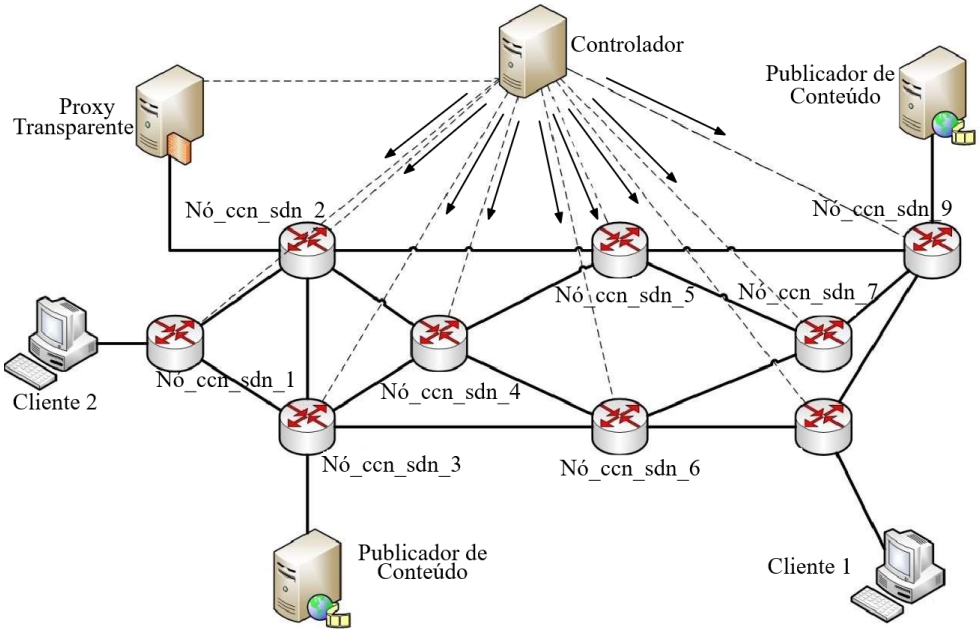

Figura 22 – Mensagem do Controlador (Control_announce (O Autor).

Quando um nó recebe essa mensagem pode executar duas ações distintas: Se já estiver vinculado a um controlador descarta-se o pacote, caso contrário se vincula ao controlador (enviando uma mensagem de resposta *Node_Link* com as variáveis *Node_Id* e *Node_Type*) e encaminha a mensagem para todas as outras faces. A variável *Node_Id* contém o identificador do nó que é constituído pelo endereço MAC da interface que recebeu a mensagem *Control_annouce* e a variável *Node_type* contém o tipo do nó, que pode ser: um nó CCN, um proxy SDN ou um publicador de conteúdo.

A partir do momento que um nó se vincula a um controlador, o mesmo envia para todas suas faces uma mensagem de reconhecimento de vizinhança. Essa mensagem tem a identificação do controlador e uma requisição de qual controlador o nó vizinho está vinculado. Essa mensagem é identificada com o nome *Find_Neighbours*. Depois de todas as requisições respondidas, o nó envia uma mensagem chamada *Neighbours_List* para o Controlador, que constrói o mapa da topologia que será usada para a instalação dos fluxos de dados.

5.3.2 Roteamento

O protocolo de roteamento usado é o NLSR (*Named-data Link State Routing Protocol*), que também é um protocolo link state, porém além de encaminhar os LSAs o protocolo distribui os prefixos nomeados. O encaminhador NDN (*Named data Network*) no qual está implementado no NLSR, anuncia os prefixos nomeados que estão em sua FIB. Essas entradas podem ser adicionadas de forma dinâmica ou de forma estática por meio de qualquer publicador diretamente ligado ao encaminhador. Da mesma forma que é

adicionado quando um prefixo não estiver mais disponível, ocorre a disseminação de um novo LSA que muda o estado do prefixo nas tabelas de roteamento da rede. A grande diferença entre o NLSR e um protocolo OSPF clássico é o fato de não apenas ter o anúncio de melhor próximo salto, mas sim de vários saltos em diferentes interfaces (*ifaces*). Esse protocolo utiliza os pacotes de dados e interesse para enviar esses anúncios, ou seja, utiliza as características da rede para mandar os anúncios. O encaminhador que utiliza esse protocolo é identificado pelo nome e não pelo IP.

5.3.3 Encaminhamento de Conteúdo e Autenticação de Cliente

Esta etapa consiste em se criar um caminho para rotear uma mensagem de interesse para um nó que tenha o conteúdo em seu *Content Store*, ou seja, o nó requisita regras para o controlador para achar o caminho de um interesse até o conteúdo.

Considere como exemplo que o cliente 2 da Figura 19 está interessado em um conteúdo ou *chunk* qualquer. Então ele enviará uma mensagem de interesse (mensagem padrão do CCN) ao nó ccn_sdn_1 ao qual está ligado. Nessa arquitetura temos três situações diferentes em relação ao encaminhamento de conteúdo:

- Na primeira situação, o cliente 2 requisita o conteúdo e envia uma mensagem de interesse ao primeiro nó que não tem no seu CS o arquivo em questão, não tem rota na sua FIB e não tem nenhum proxy identificado para autenticar o cliente 2. Nesse caso, o nó ccn_sdn_1 irá mandar uma mensagem (*find_proxy*) ao controlador requisitando a identificação de um proxy ativo para que seja possível a autenticação do cliente. Depois da informação do proxy, o Nó ccn_sdn_1 deve mandar uma mensagem ao proxy autenticando a possibilidade de enviar o conteúdo ao cliente com a mensagem (*Auth_client*), a mensagem anterior é respondida com outra mensagem chamada User_autentication, com dois parâmetros: autenticado ou não autenticado. Se o cliente 2 for autenticado pelo proxy, o nó ccn_sdn_1 irá mandar uma mensagem ao controlador requisitando um fluxo de dados ao controlador destinado ao conteúdo indicado na mensagem de interesse, senão a conexão é desfeita e o cliente recebe uma mensagem de conteúdo não autorizado. O proxy também é responsável por passar as estatísticas de balanceamento de carga para o controlador quando tiver o mesmo conteúdo disponível em dois publicadores diferentes através da mensagem Load_balance_Chunk.

- A segunda situação se dá se o nó mais próximo (ccn_sdn_1) não tiver o conteúdo em sua CS mas possui em sua FIB um caminho que pode ser usado para encontrar o conteúdo requisitado. Nesse caso o nó que contém a rota encaminha o interesse até o nó que tenha na sua CS o conteúdo requisitado para o mesmo pedir o processo de autenticação de usuário para o nó para o proxy SDN.

- A terceira situação se dá se o nó mais próximo (ccn_sdn_1) tiver o conteúdo em sua CS, nesse caso só ocorre o processo de autenticação no proxy SDN pelo usuário que está requisitando o conteúdo através do cliente 2. Na Figura 23 temos um fluxograma demostrando o procedimento descrito anteriormente.

Uma das grandes contribuições desse método no quesito segurança, é que se o cliente não for autenticado, ele não receberá os caminhos para encontrar o conteúdo, ou seja, ele não participa do domínio de entrega do conteúdo.

5.4 Validação e Avaliação do Protocolo

Para avaliar o desempenho de uma rede CDN, alguns tipos de métricas podem ser utilizadas (SIVASUBRAMANIAN et al., 2004)):

- Temporais: tempo de espera do atendimento de uma requisição. (OLSHEFSKI; NIEH; AGRAWAL, 2004)(HARCHOL-BALTER et al., 2003);
- Espaciais: distância geográfica entre os nós ccn_sdn ou número de saltos (HUFFAKER et al., 2002);
- Utilização de recursos da rede: largura de banda usada, atraso, tempo de manutenção, atualização de publicadores de conteúdo, entre outros (SIVASUBRAMANIAN et al., 2004);
- Financeiras: associadas com o custo dos servidores e sua manutenção;
- Consistência dos dados: nível de consistência do conteúdo acessado.

Essas métricas são importantes porque a nova abordagem deve fornecer subsídios para atende-las, porém, algumas métricas da rede CCN também podem ser usadas tais como:

- Número médio de saltos;
- Tempo médio de recuperação, que é contado do envio do primeiro interesse até a resposta do primeiro chunk;
- Número médio de pacotes de interesse;
- Número médio de pacotes de dados;

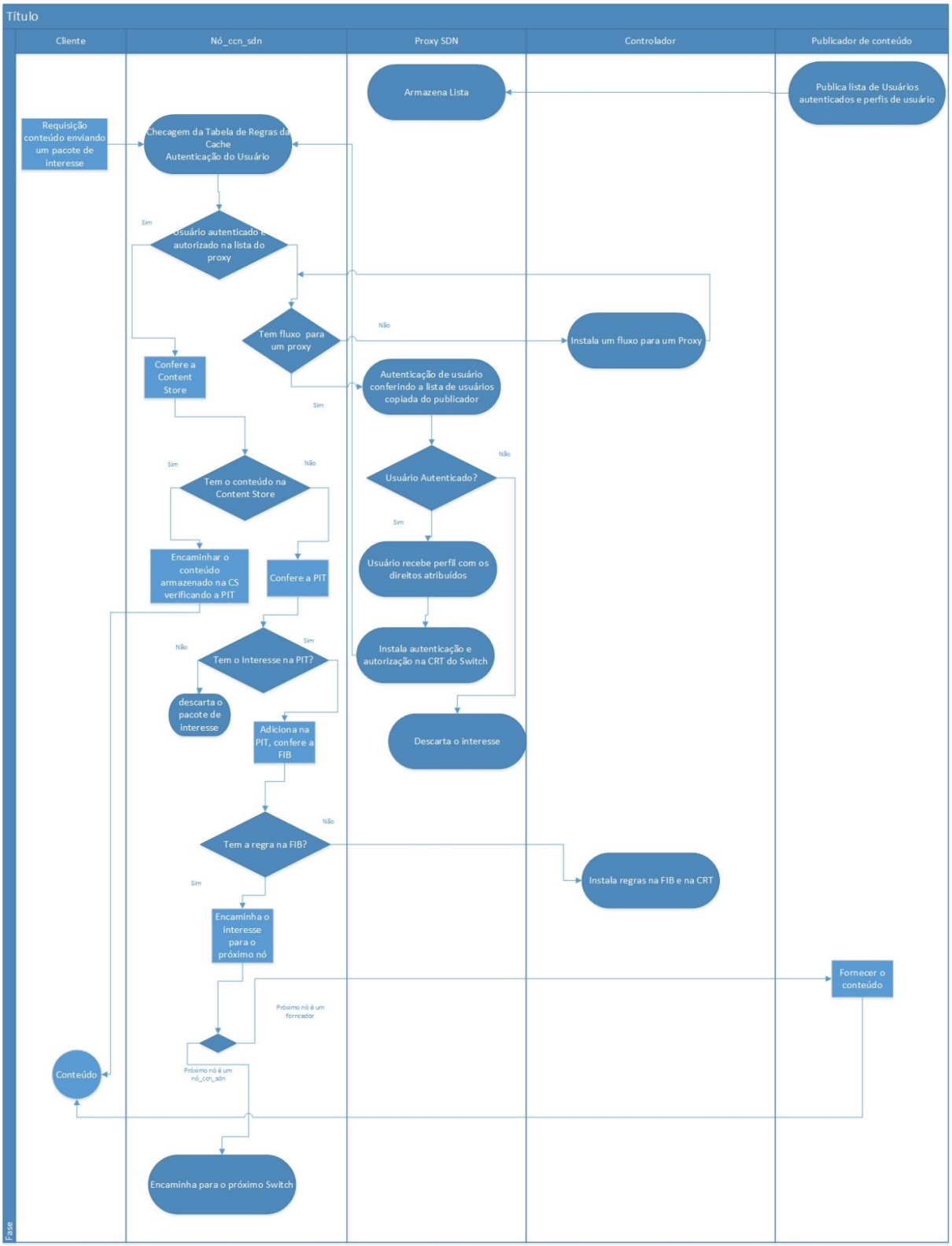

Figura 23 – Fluxograma do pedido de conteúdo na arquitetura CCN+SDN (O Autor)

No caso da arquitetura proposta foram validas utilizando métricas: temporais - onde comparamos o tempo de download de um arquivo, número médio de saltos - a partir dos downloads feitos obtivemos o numero de saltos de cada uma das iterações. E por

ultimo validamos o sistema de controle de acesso fornecendo acesso e retirando de usuários pré estabelecidos no ambiente.

5.5 Cenário da Experimentação

O cenário utilizado foi virtualizado numa máquina física com processador Intel i7 8GB de memória RAM e SSD 240GB, com o sistema operacional Windows. O software de virtualização utilizado foi o Wmware Workstation Pro 16 e foi instalado sistema operacional Linux Ubuntu 18.04. Os softwares utilizados foram:

- Controlador *Pox* (KAUR; SINGH; GHUMMAN, 2014)
- Banco de Dados Nativo *Mysql* (ORACLE, 2022) nativo do Linux
- *Python*(PYTHON, 2022)
- *Open vSwitch*(LINUXFOUNDATION, 2022)
- *Mininet*(ONF, 2022)
- *Packet Tracer*(CISCO, 2022)

5.6 Autorização do Cliente

A etapa de autorização é uma das mais importantes desta arquitetura, pois nessa etapa temos o repasse ao usuário dos direitos que ele terá ao acessar determinado roteador. Considere como exemplo a Figura 24 temos a configuração simples de requisição de conteúdo. Em primeiro lugar o cliente 1 é autenticado e recebe os fluxos com o caminho de rede do publicador de conteúdo. Em segundo lugar o cliente 1 requisita qualquer conteúdo ou *chunk* ao publicador e recebe, sendo copiado nos dois nós intermediários (nó_ccn_sdn_1 e nó_ccn_sdn_2).

Considere agora que o cliente 2 deseje uma cópia do mesmo conteúdo. Assim que buscar ele já vai achar uma cópia no nó_ccn_sdn_1, porém é necessária uma estratégia de controle de acesso ao conteúdo. Levando em consideração um cenário *multi-tenant* propomos em nossa arquitetura que o controle de acesso seja feito pelo controlador SDN para serviços de vídeo sob demanda. Nesse caso, depois da autenticação do cliente 2, acontece a autorização por parte do controlador para que o mesmo tenha acesso a esse conteúdo ou a parte dele.

O componente central de uma estrutura SDN é o controlador, que é um dispositivo que tem a função de centralizar a comunicação dos outros componentes programáveis da

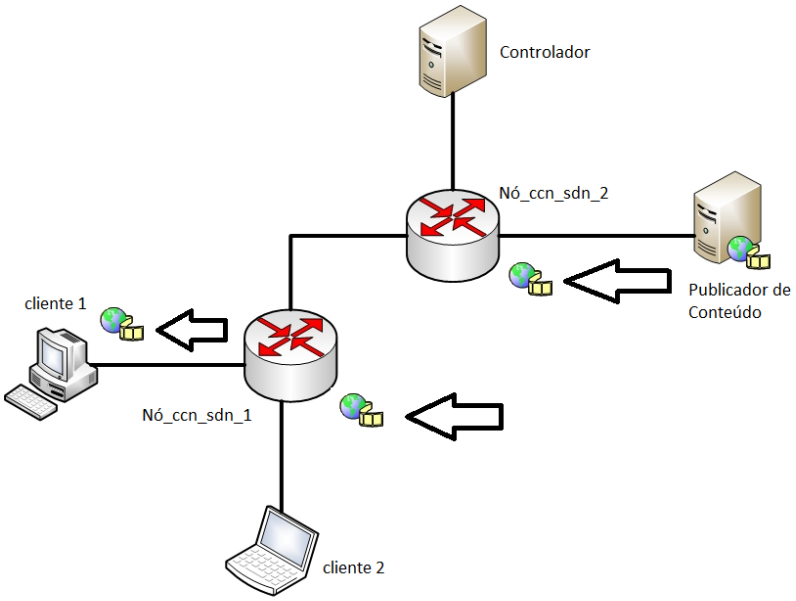

Figura 24 – Requisição e entrega de conteúdo ao cliente 1 (O Autor)

rede. O POX é uma alternativa ao controlador NOX e foi utilizado para o desenvolvimento da arquitetura proposta.

Com a utilização do SDN, tem-se a possibilidade de controlar a autorização do fluxo entre componentes da rede, e em consequência disso estabelecer direitos correlatos a inserção de tal dispositivo na rede, os encaixando em perfis propostos pelo administrador.

A arquitetura proposta tem o seguinte funcionamento. Quando um componente quer fazer parte da rede, o controlador recebe uma notificação. A seguir, o controlador coleta informações sobre o dispositivo e um banco de dados (Mysql) é consultado. Se o componente tiver um perfil de usuário cadastrado, o controlador avalia se o usuário tem direitos na rede, levando em consideração o horário em que foi acessado. Se não tiver nenhum perfil, o controlador o associa a um usuário padrão sem nenhum direito na rede. Na Figura 25 temos o fluxograma desse procedimento.

5.7 Considerações Sobre o Capítulo

Nesse capitulo forma descritos a arquitetura proposta, a função de cada entidade da rede, estratégias de descoberta de topologia, roteamento e qual cenário prático pode ser utilizado.

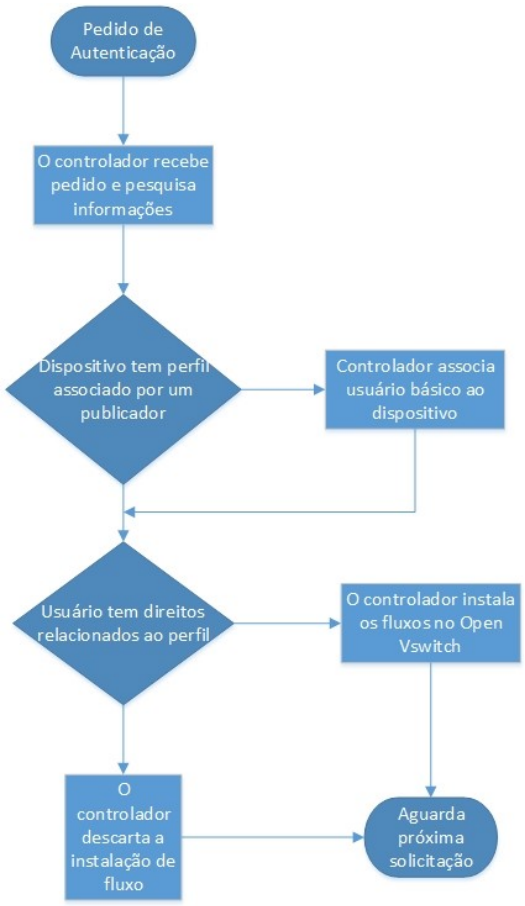

Figura 25 – Fluxograma de autorização de perfil de usuário (O Autor)

6 EXPERIMENTOS E RESULTADOS

Os experimentos realizados visam avaliar o desempenho da arquitetura proposta e a eficiência no controle de conteúdo, utilizando a instalação ou não de fluxos através de um controlador SDN. A gestão dos perfis de usuário são feitos por um administrador de rede, ou seja, o administrador de rede pode estabelecer privilégios aos dispositivos da rede através da liberação de direitos aos perfis de usuário. A troca de pacotes acontece se, e somente se, o usuário relacionado aos dispositivos tiver privilégios na rede, no momento da requisição, senão os pacotes são descartados. Este capítulo esta estruturado da seguinte forma:

Na Seção 6.1 foi montado um cenário na arquitetura clássica TCP/IP da Internet, onde se pode testar os tempos de download e o número de saltos que são efetuados em uma tecnologia já consolidada. Na Seção 6.2 temos a troca da arquitetura clássica por uma que utiliza as duas tecnologias SDN e CCN, na mesma topologia utilizada no item anterior. Por último, na Seção 6.3, temos a inserção do controle de acesso na tecnologia SDN e CCN na mesma topologia da Seção 6.2.

6.1 Topologia TCP/IP

A topologia TCP/IP é a base da internet atual. Nesse cenário foi utilizado o programa de simulação *Packet Tracer* na versão 7.2.2, concebido pela fabricante de equipamentos de rede Cisco. Na Figura 26 temos a topologia utilizada como base de testes dos protocolos clássicos utilizados atualmente. Nessa figura foram utilizados primeiramente dez roteadores cisco modelo 2911, três servidores de arquivos, 8 switches camada 2 e dez computadores pessoais que fizeram o papel dos clientes de vídeo sob demanda.

As conexões feitas entre os computadores pessoais e o switch foram feitas através das portas gigabit ethernet, assim como as ligações entre os switches e os roteadores (routers). Ligações entre os roteadores foram feitas através de conexões seriais com links de largura de banda de 2Mbps. As subredes foram feitas com o endereço 192.168.0.0, com máscaras de subrede 255.255.255.0, que é padrão neste tipo de endereço. A primeira utilizada na LAN do router 0 foi a 192.168.0.0, já a ligação entre o router 0 e o router 0 (2) foi utilizado o endereço 192.168.1.0 e assim por diante.

O protocolo de utilizado na topologia foi o OSPF (*Open shortest path Fisrt*), de fácil configuração e acessível ao aumento do número de roteadores. A topologia escolhida foi utilizada pela facilidade de aumento do número de roteadores havendo a necessidade. Nesse caso aumenta-se o número de roteadores em série no meio da topologia.

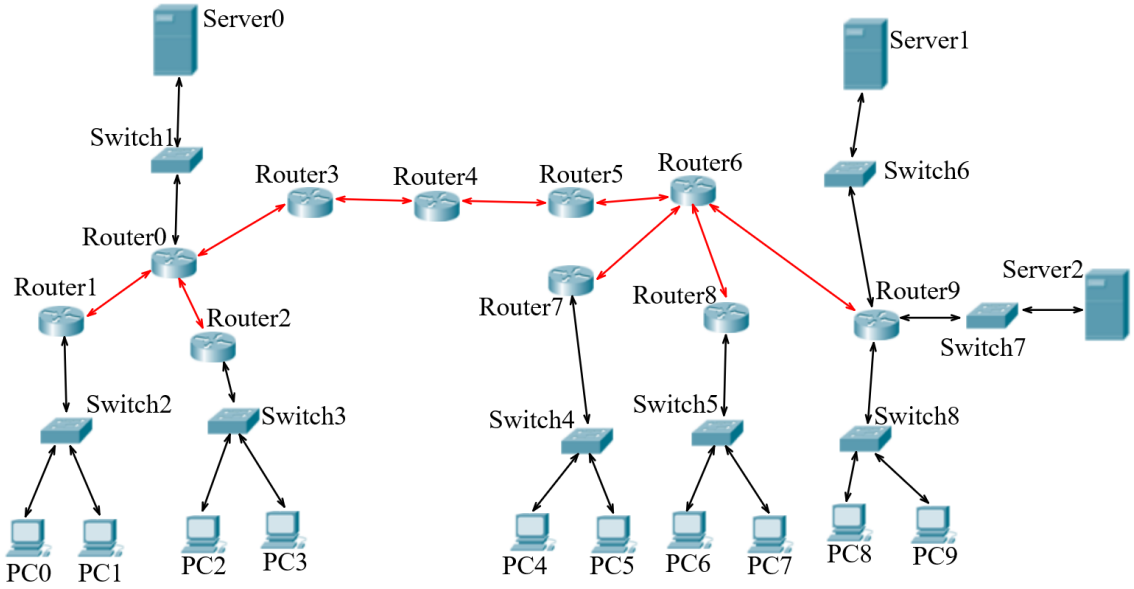

Figura 26 – Topologia tcpip utilizada no comparativo (O Autor).

6.2 Topologia SDN-CCN

Para a concepção desse projeto foi utilizado o simulador Mininet modificado com a adição de um *plugin* CCN, propondo uma solução inovadora orientada a conteúdo. Essa arquitetura é composta por clientes que são dispositivos que requisitam conteúdo. Três controladores SDN, um central e outros dois que recebem cópia dos conjuntos de regras para fornecimento de fluxos. Também foram utilizados switches CCN preparados para atuar com conteúdo ou segmentos do mesmo. Essa topologia está demonstrada na Figura 27. A topologia foi criada no Miniedit, que é uma interface gráfica para o Mininet. Essa ferramenta foi utilizada pela facilidade de construir topologias e aumentar o número de dispositivos de rede, salvando-os em topologias completas e scripts de topologia.

A arquitetura construída suporta trabalhar com *cache* de forma dinâmica, a arquitetura original utilizada em (CHARPINEL et al., 2016) trabalhou com experimentos utilizando algoritmos de controle de *cache* diferentes e as comparou, nesta contribuição não foram levadas em consideração alteração nas caracteristicas de *cache*, ou seja, foram utilizados as estratégias de *cache* padrão da tecnologia CCN.

No host h1 roda uma aplicação chama ccnxget e o host 8 atua como publicador de conteúdo e utiliza uma aplicação ccnxgetserver. O resultado da aplicação ccnxget está descrito na Figura 28. As ferramentas utilizadas no CCNx 0.8 foram o ccnputfile e ccngetfile, para a realização do *download* de um arquivo de 30MB. Foram feitas 6 repetições.

O resultado encontrado pode ser visto no gráfico da Figura 29, comparando as arquiteturas TCP/IP, SDN-CCN.

Na arquitetura TCP/IP, os conteúdos percorrem sempre o mesmo caminho, dessa

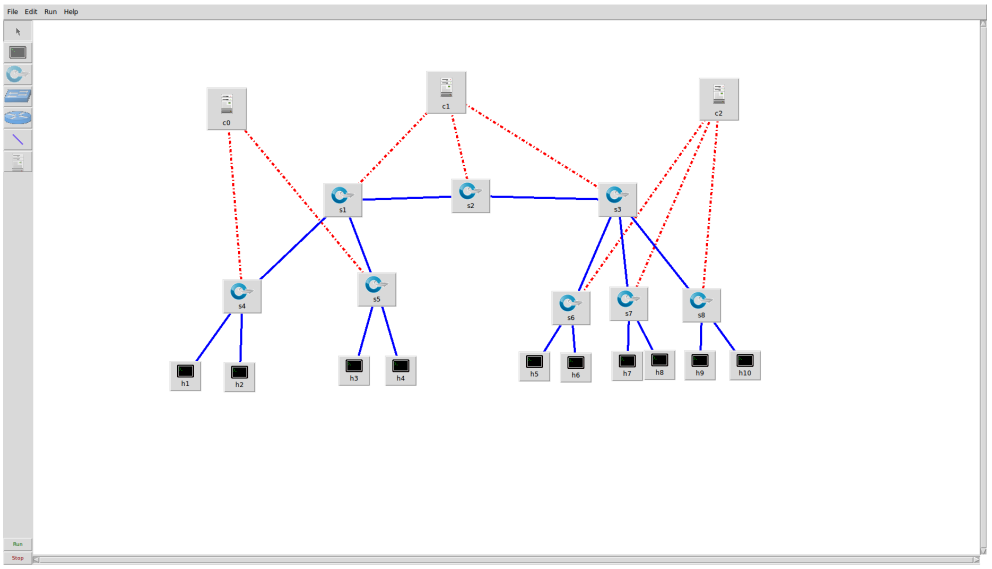

Figura 27 – Topologia utilizada na configuração SDN-CCN (O Autor)

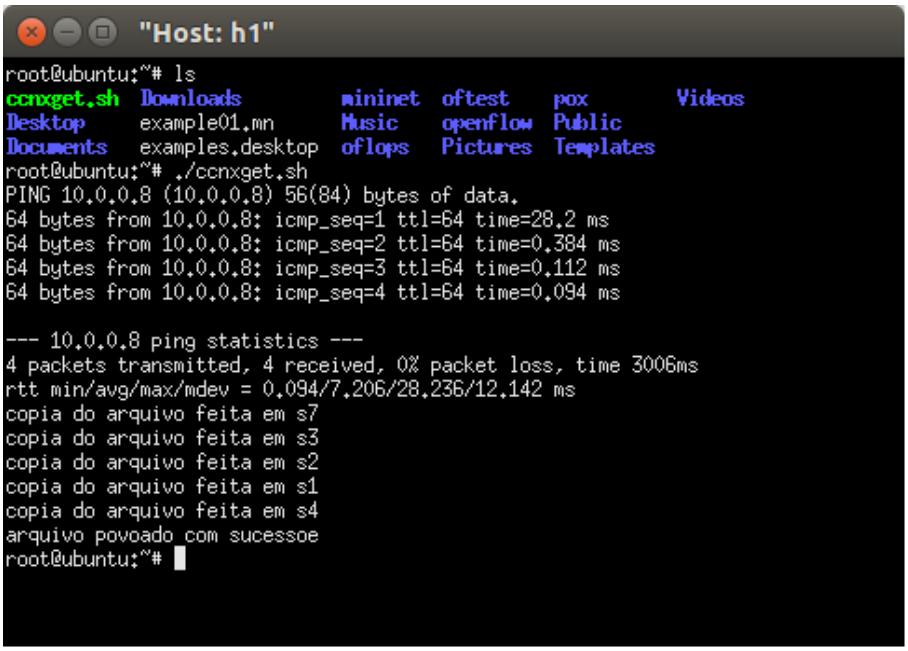

Figura 28 – Resultado do script de pedido de conteúdo (O Autor)

forma quanto mais aumentar o número de requisições, mais se aumentará o tempo total de *downloads*. No ambiente SDN-CCN, a partir da segunda requisição, se mantêm constante pois o primeiro download é feito no publicador e todos os outros em *cache* no *switch* mais perto. Para o controle de tráfego foram utilizadas as ferramentas *TCPdump* e *Wireshark*.

A rede ccn armazena o arquivo em cada dispositivo, sendo muito vantajoso pois diminui o tempo de aquisição dos dados. No segundo teste temos a comparação dos tempos de *download* de um arquivo, enquanto o número de clientes cresce. Esse resultado está demonstrado no gráfico da figura 29. Atente-se para o detalhe inicial da tecnologia CCN

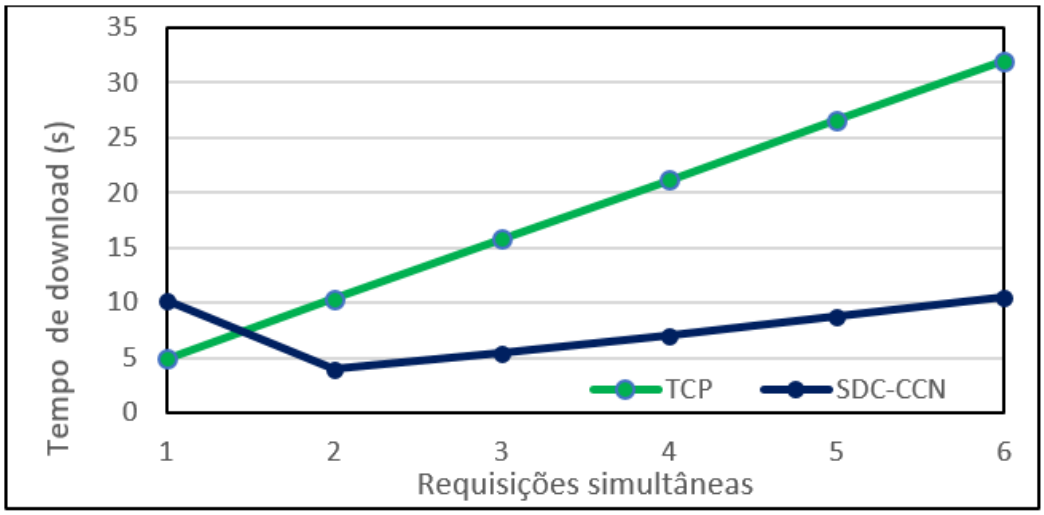

Figura 29 – Gráfico comparativo das arquiteturas (O Autor)

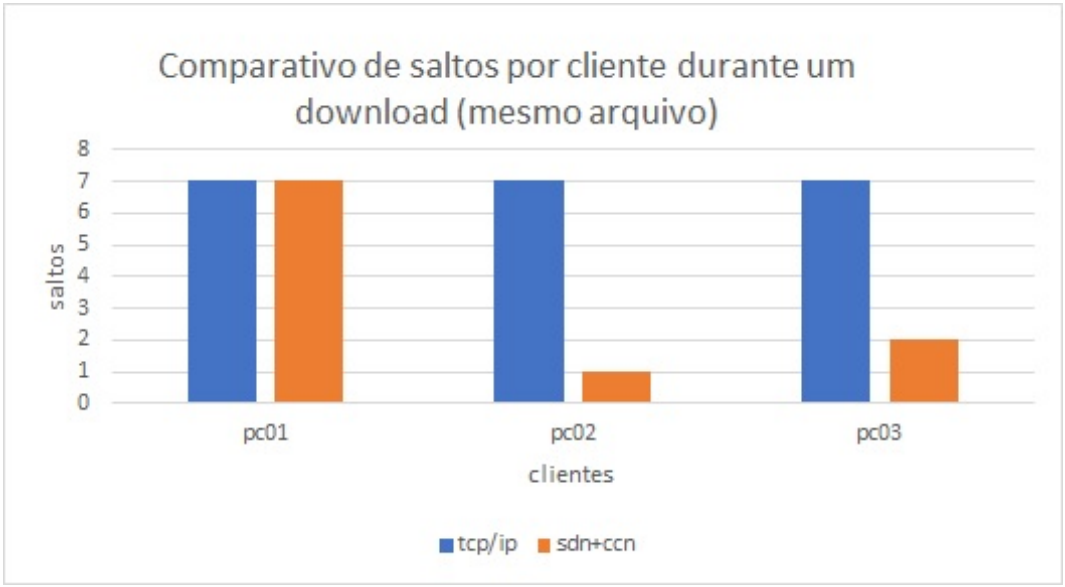

Figura 30 – Comparativo de saltos por cliente durante um *download* (O Autor)

que está com um tempo maior por causa do *overhead* causado pela implementação do CCN sobre o TCP/IP.

Observe que no primeiro teste foi feita a comparação de *download* de um mesmo arquivo, por três clientes diferentes, nas duas arquiteturas. Quando se fez o *download* com a arquitetura clássica TCP/IP, tivemos o número máximo de saltos ao contraria da arquitetura proposta que a partir do segundo *download* sempre diminuiu consideravelmente o número de saltos. Essa comparação está demonstrada no gráfico da 30

No segundo cenário de testes realizamos procedimentos de *download* de forma aleatória entre os hosts. Os dez computadores foram numerados de 0 a 9 e tais números

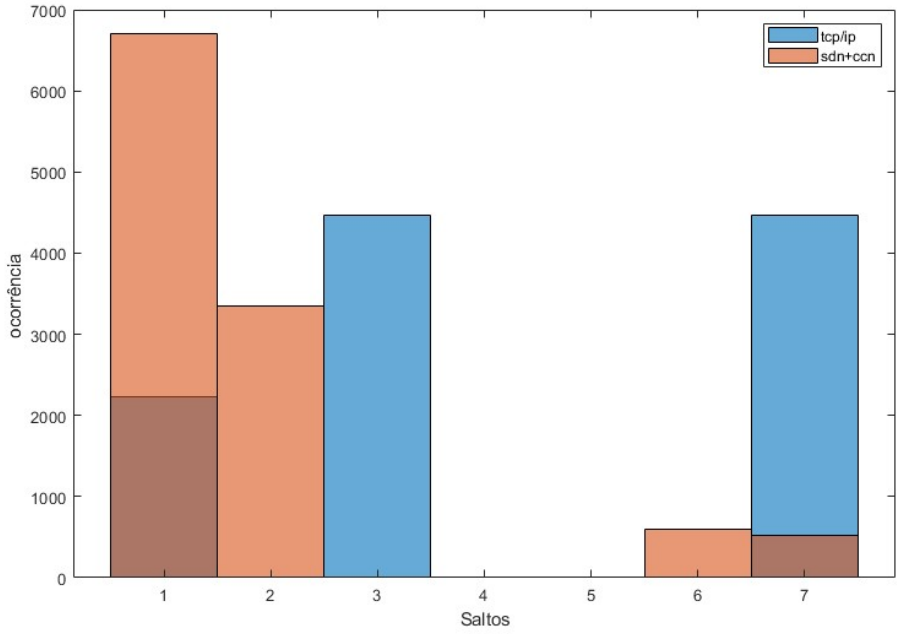

Figura 31 – Número de ocorrências por protocolo (O Autor)

foram permutados sem repetição. Para cada combinação era feito o *download* de um arquivo teste (todas as iterações baixaram o mesmo arquivo). A cada rodada era feito o armazenamento do numero de saltos que cada um dos hosts executava para fazer o *download* desse arquivo teste, em cada uma das arquiteturas.

Foram realizadas 1100 combinações diferentes de ordens nas quais os computadores requisitam o conteúdo, nas duas estratégias. Os resultados são apresentados na Figura 34, na forma de um histograma.

Nesse histrograma temos as seguintes características, dentro desse universo de 1100 *downloads* aleatórios do mesmo arquivo temos o acumulado de saltos feitos por cada arquitetura, na cor azul temos as ocorrências dos saltos utilizados para o *download* de um arquivo na tecnologia TCP/IP, na cor marrom claro temos as ocorrências de saltos utilizados para o *download* de um arquivo na tecnologia SDN+CCN. Na cor marrom temos uma sobreposição onde as duas tecnologias em determinados momentos tiveram o mesmo número de saltos. Podemos observar que a arquitetura proposta tem um número de saltos bem reduzido se comparado com a arquitetura clássica TCP/IP, ou seja, o número de saltos para se baixar um arquivo na tecnologia SDN+CCN é menor.

6.3 Topologia SDN-CCN com Controle de Acesso

Nesse momento a arquitetura SDN+CCN está apta e funcionando, ou seja, o cenário foi testado e funciona conforme a literatura. Os aquivos são copiados nos equipamentos

intermediários de rede, e podem ser acessados por outros dispositivos que quiserem o mesmo conteúdo. Porém utilizar esse cenário numa rede de entrega de video sob demanda nos remete ao problema de pesquisa central, o próximo componente a acessar o arquivo que está salvo em *cache* tem a autorização necessária para esse acesso?

Nas redes de computadores, a proteção dos dados sempre teve grande importância. Um dos grandes desafios na segurança dos dados está relacionado aos diretos de uso determinado conteúdo ou informação. Nesse sentido para poder ter o controle de acesso ao arquivo salvo em *cache* para os usuários foi desenvolvido uma estratégia de configuração e alteração do código do controlador SDN (POX) em conjunto com banco de dados *Mysql* de usuários e direitos sobre os arquivos e um *front end WEB*. A arquitetura proposta tem o seguinte funcionamento, a partir do momento que um componente entra na rede de conteúdo o controlador recebe uma mensagem. Informações sobre esse componente são adquiridas pelo controlador se o componente tiver um usuário associado lhe é repassado seus privilégios, senão é associado a um usuário padrão que tem pouquíssimos diretos na rede. Conforme descrito no diagrama de sequência demonstrado na Figura 32.

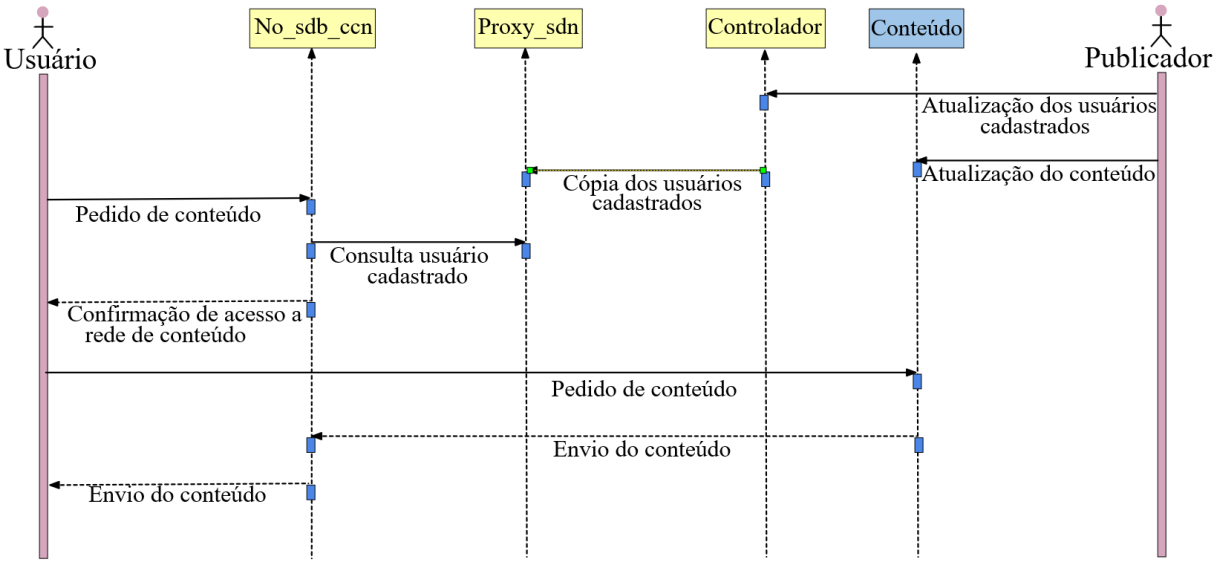

Figura 32 – Diagrama de sequência do controle de acesso (O Autor)

Assim que o componente que adentra a rede for associado a um perfil de usuário, o gerenciamento será feito através de uma aplicação simples e fácil utilização do administrador de rede. Foi concebido uma aplicação que tem como *front end* uma página da *Web*. Essa aplicação está demonstrada na Figura 33.

A primeira aba é chamada de *Status*, nessa aba tem-se uma lista de todos os dispositivos que fazem parte da rede. Foram criados diversos perfis de usuário padrão, daqueles que não tem direito a nada na topologia, pois além de servirem como objeto de teste para clientes externos inseridos após a concepção da arquitetura, os *switches*

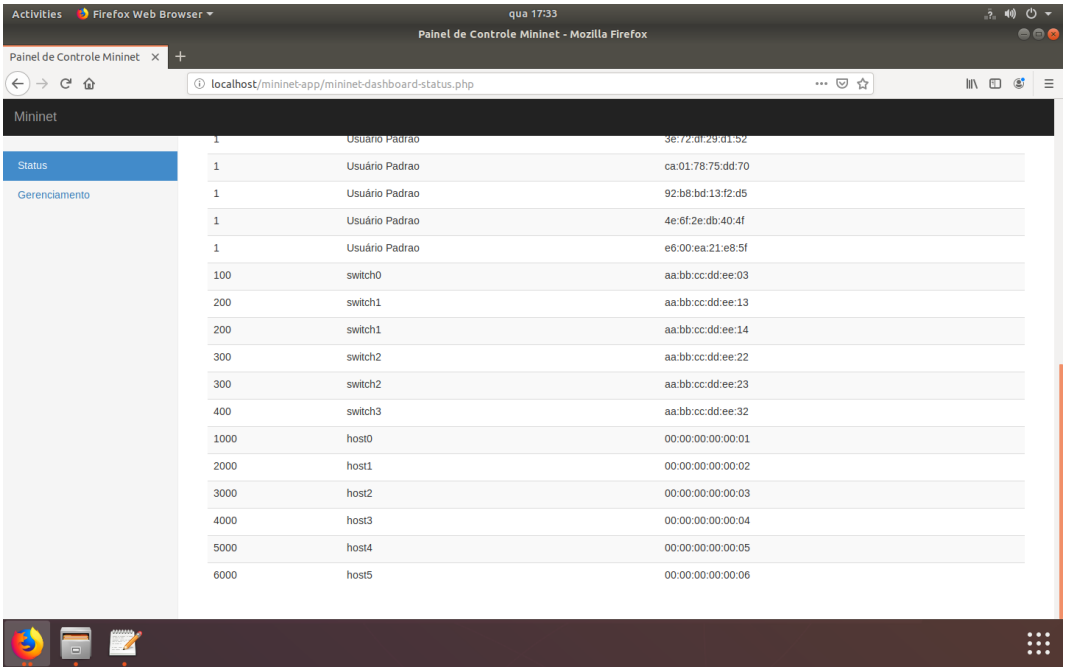

Figura 33 – Painel de controle e gerência da Rede 2 (O Autor)

inicializam com endereços de forma aleatória devendo ser associados a esses perfis no início, a seguir podem e devem ser alterados para endereços com perfis atualizados e com direitos irrestritos.

A segunda aba é chamada de Gerenciamento tem as principais funcionalidades da aplicação, tais como: a criação de usuários (criação de perfis), associação de dispositivos aos usuários, edição de privilégios dos perfis de usuário e remoção do usuário do banco de dados. Esta aba esta demonstrada na Figura 34.

Para adicionar um usuário primeiramente basta colocar um número id de identificação e um nome para este usuário, essas opções que podem ser vistas na figura 39. Para alterar privilégios, adicionar dispositivos ou remover usuários podemos utilizar a opção listar usuário no item Procurar Usuário. Procura-se o usuário pelo nome, se procurarmos um usuário e não digitarmos nenhuma entrada serão listados todos os usuários existentes na Figura 34 temos um exemplo de listagem dos usuários existentes.

Ao lado do nome de cada um dos usuários foram criados três botões: Um azul um verde e um vermelho. O botão azul abre o menu de alterações de privilégios, nesse botão pode-se especificar qual dia da semana aquele perfil de usuário está possibilitado a utilizar a rede e acessar os conteúdos. Na Figura 35 tem-se essa alteração.

O botão verde abre um menu de associar e desassociar dispositivos. Infelizmente não ficou otimizado da melhor forma, mas não altera a eficácia da aplicação. Ao clicar em 'Find' uma busca será feita, o menu irá fechar, mas ao clicar no botão verde o menu ira novamente abrir com todos os dispositivos listados, associados ou não associados. Para

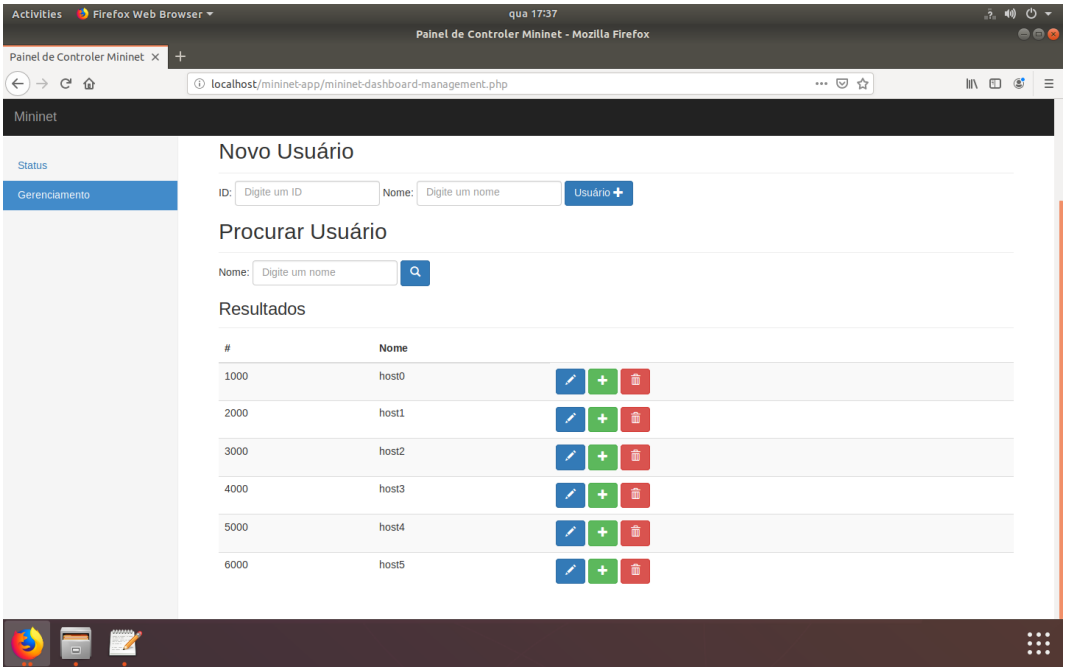

Figura 34 – Listagem de usuários (O Autor)

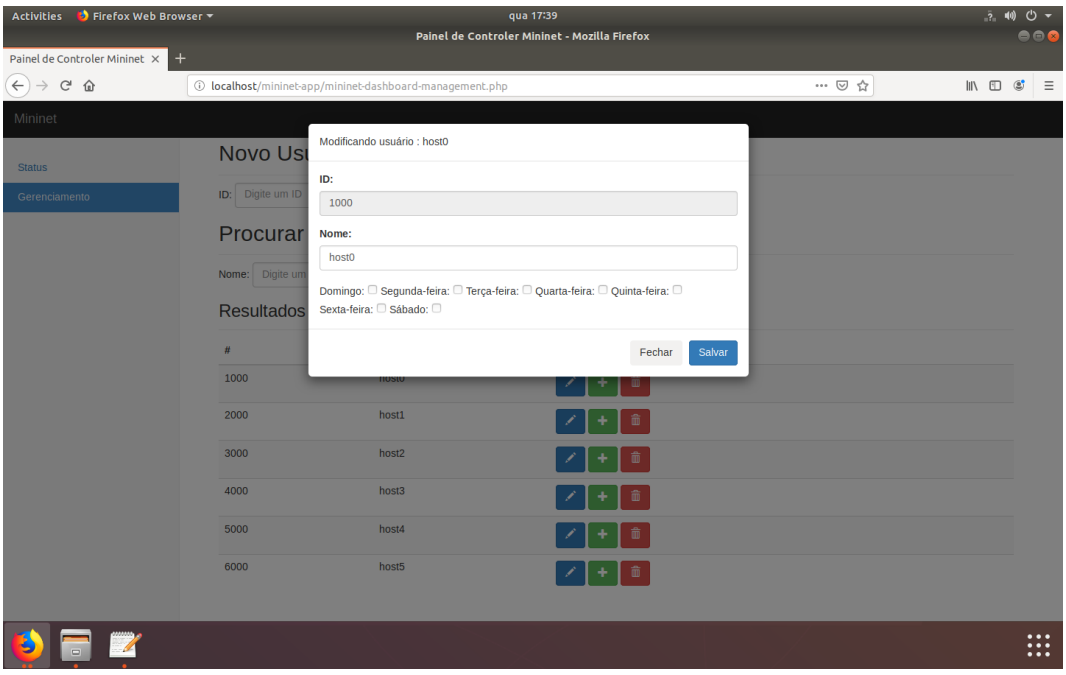

Figura 35 – Privilégios de usuário (O Autor)

associar o dispositivo basta clicar no botão azul '+' e para desassociar, clicar no botão vermelho 'x'.

O botão vermelho irá excluir o usuário (perfil de usuário). Nesta aplicação, quando um usuário é criado, este se torna também um perfil de usuário. Desta forma, para esta aplicação apenas é possível um usuário por perfil de usuário.

Nesse momento deve-se inicializar o controlador POX. Se o controlador não for inicializado antes do Minient, o mesmo iniciará o controlador padrão que nesse caso, seria o Nox. Neste momento com o Pox preparado a proxima etapa é iniciar o programa de simulação.

O programa de simulação utilizado é o Mininet. Foi utilizado o seguinte comando para a inicialização do programa.

```
sudo mn –custom /home/sdn/sdn-experiment/
mininet-topo/showTopo.py –topo showtopo
–controller remote –switch ovsk –mac
```

Onde a saída desse comando tem como principal função iniciar o Mininet com uma topologia predeterminada, o arquivo *showTopo.Py* é o *script* onde é salvo a topologia que foi utilizada na simulação. Após a inicialização os comandos são executados dentro do Mininet.

A topologia utilizada para os testes de autorização de perfis de usuário é a demonstrada na Figura 27, essa topologia foi configurada no arquivo *Show.Topo.py*.

A partir desse momento é necessário trocar os endereços assumidos pelos switches provenientes do controlador. Por limitações encontradas no processo, não é possível a troca dos endereços MAC dos *switches* durante o processo de inicialização o que deve ser feito após a mesma. Desta forma deve-se forçar cada interface a ter um endereço específico para facilitar o debug (Ação de extrema importância, aprendido durante a pesquisa).

A seguir foi feito um teste de conexão através do comando pingall, onde todos os hosts utilizam essa ferramenta para testar a conexão entre eles. Na Figura 36 que todos os hosts enviaram pacotes e receberam a resposta porque todos os perfis de usuários estão com o privilégio de conexão.

Também pode-se ver através de um terminal de controle do controlador POX que todos os usuários foram autorizados.

A seguir volta-se para a aplicação Web e retira-se todos os privilégios do *host3*, como demostrado na Figura 37 e a seguir o administrador recebe uma resposta que o procedimento deu certo.

Volta-se ao Mininet e se executa o mesmo teste com o comando *Pingall* e se percebe que o host3 não completa as conexões com nenhum dos outros *hosts*. Teste demonstrado na Figura 38.

Também consegue se ver pelo terminal do controlador o perfil de usuário do *host3* não autorizado conforme descrito na Figura 39.

Outro experimento comprobatório feito foi de deixar um *script* de *download* de conteúdo para o host3 com os direitos de perfis de usuário liberados, esse processo está

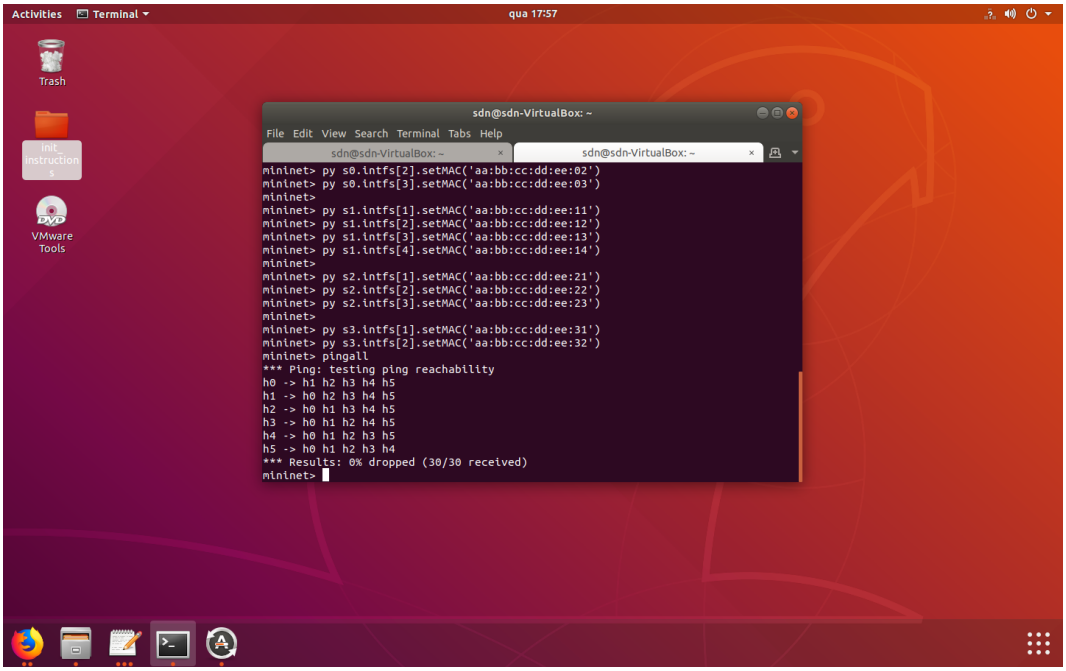

Figura 36 – Teste de conexão com a ferramenta Pingall (O Autor).

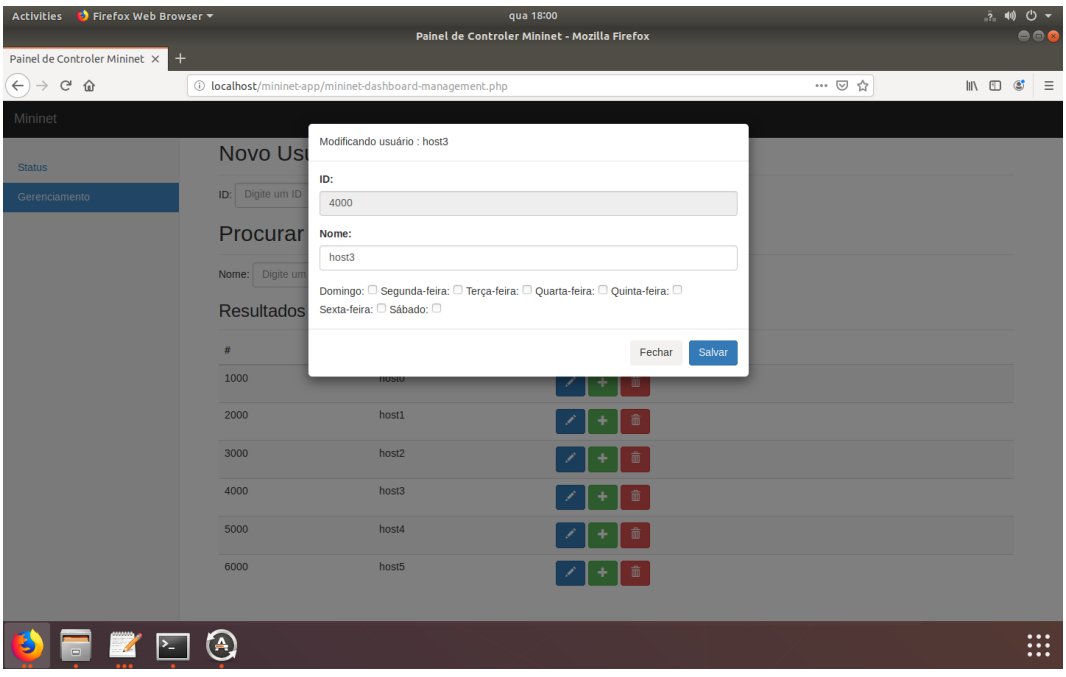

Figura 37 – Retirada de todos os privilégios do *Host3* (O Autor).

demonstrado na Figura 40.

Na Figura 41 temos o momento que bloqueia o perfil do *host3* e percebesse o fluxo de dados se esvair.

A liberação e o bloqueio do perfil do *host3* foi feito de forma manual, mas não teve influência direta pois não era tempo que se precisava testar e sim a liberação ou não dos

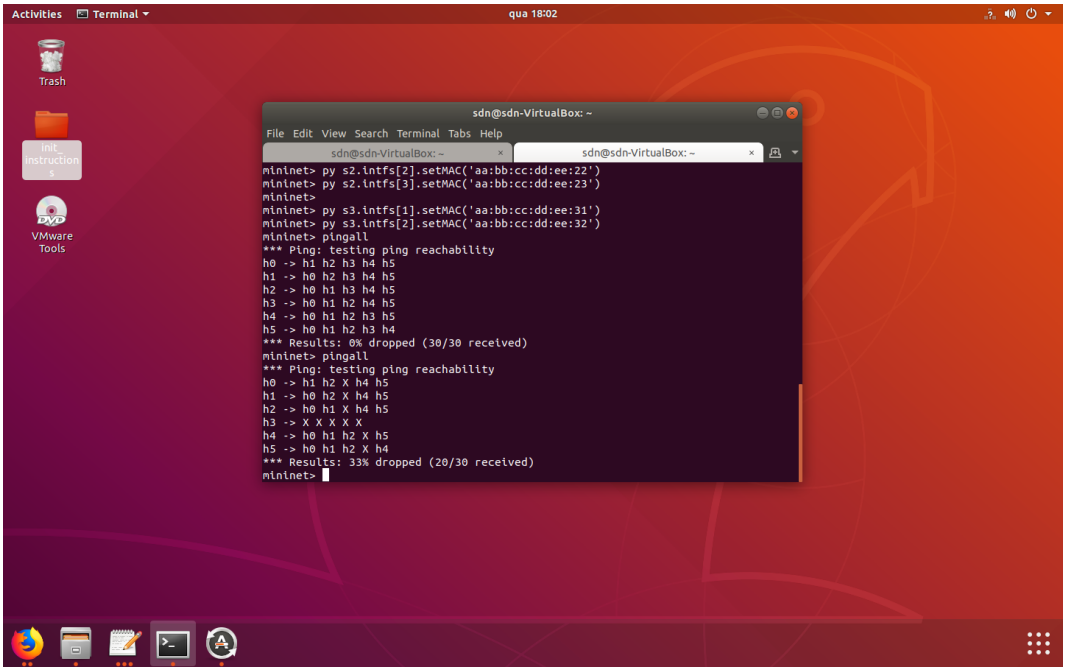

Figura 38 – Teste de conexão do host3 (O Autor).

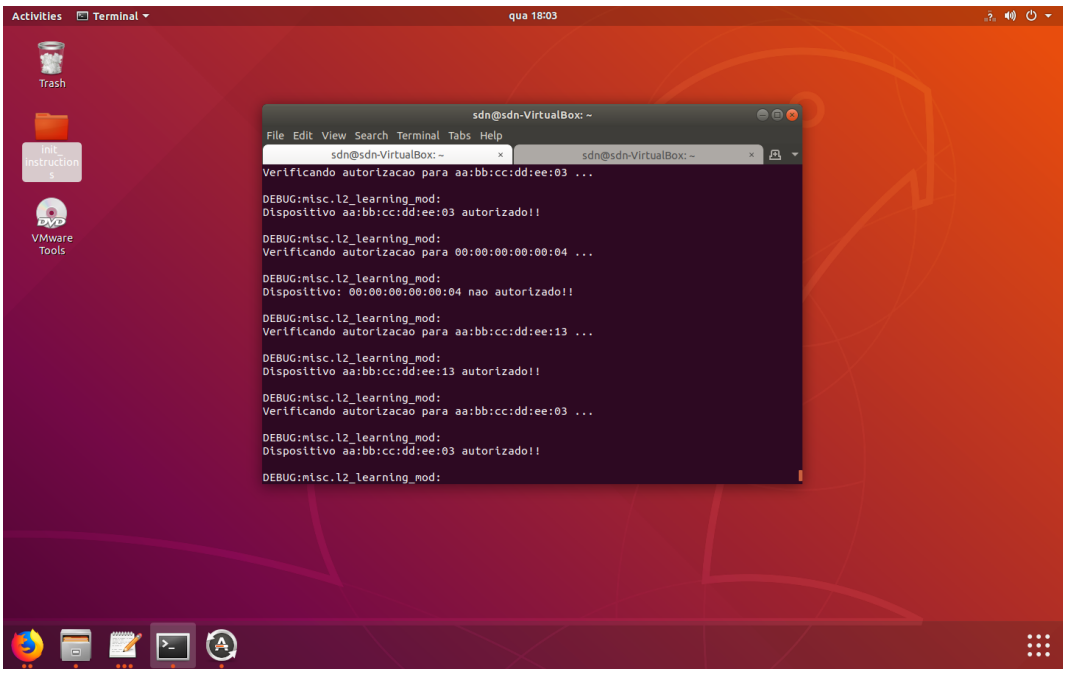

Figura 39 – Mensagens de autorização do Controlador POX 2 (O Autor).

privilégios do host em questão.

Nesse ponto tem-se um cenário que funciona numa arquitetura com as duas tecnologias (SDN+CCN), um sistema de controle de acesso funcionando no controlador POX e um sistema Web controlando os perfis de usuário facilmente gerido por um administrador de redes.

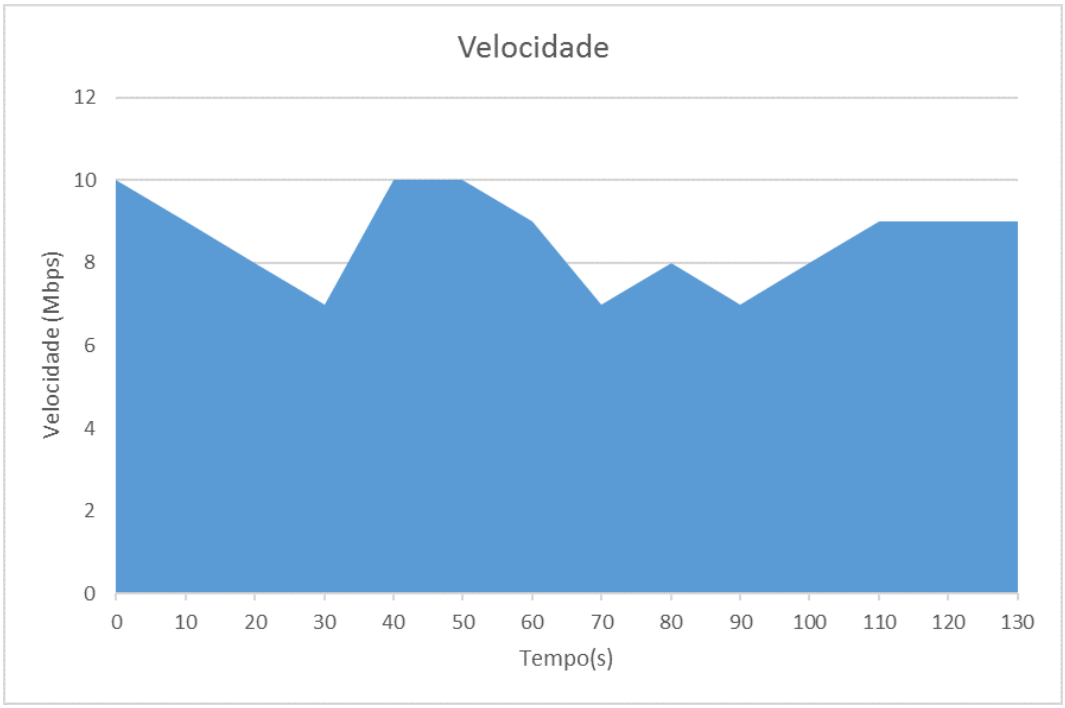

Figura 40 – Trafego Liberado host3 (O Autor).

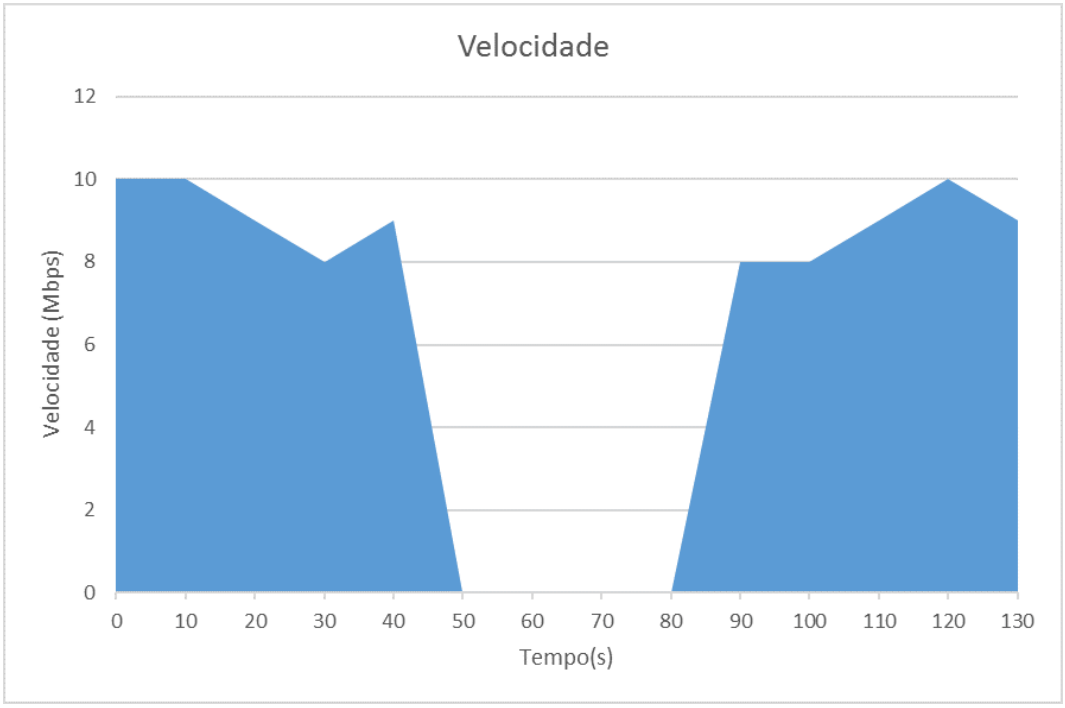

Figura 41 – Trafego não permitido host3 (O Autor).

7 CONCLUSÃO E TRABALHOS FUTUROS

A mudança de paradigma utilizando redes orientadas a conteúdo proporcionam um melhor uso para a Internet, principalmente na área de entrega de vídeo sob demanda. Porém, inserem problemas de implantação, pois trocar a toda a infraestrutura é muito caro ainda. Diversas dúvidas sobre como utilizar e manter os arquivos em cache ainda pairam sobre esse tipo de arquitetura, além disso a programação ainda tem diversos fatores a serem melhorados.

Existem diversos estudos sobre a utilização de redes orientadas a conteúdo, entretanto há diversas perguntas a serem respondidas e pesquisadas para se achar um modelo que seja viável para substituir a internet do modelo atual. Levando em consideração que o modelo atual é antigo e deve ser melhorado.

Os estudos nessa área ainda são iniciais, podendo ser explorada de modo amplo e profundo, os modelos propostos trazem várias soluções para melhorar o trafego de dados e diversos trabalhos futuros. Isso é notório quando se analisa as soluções propostas pois além de diferentes cada uma tem suas qualidades e suas oportunidades de melhoria.

A utilização das redes SDN contribuem de forma muito positiva e agrega diversos benefícios, de modo geral nas redes programáveis. Esse tipo de paradigma insere características uteis ao gerenciamento de aplicações na área de entrega de conteúdo. Utilizando uma interface amigável na camada de aplicação se torna mais fácil.

A utilização das duas tecnologias juntas abriu um novo leque de opções para melhorar a arquitetura atual. A parte CCN ajuda na largura de banda e fluxo de conteúdo, já a rede SDN melhora controle de conteúdo e o gerenciamento em um ambiente multi-tenant.

Essa tese apresentou uma proposta de controle de conteúdo em redes CCN usando estratégias SDN, a arquitetura proposta provê uma aplicação para gerenciar perfis de usuário para controlar o acesso de conteúdo.

Os resultados encontrados provaram que a solução tem utilidade e pode ser implementada como uma ferramenta real e pode ser utilizada como ferramenta de controle de acesso em ambiente de entrega de vídeo sob demanda.

Como trabalhos futuros se ressalta uma quantidade maior de testes para o armazenamento de cache, estratégias de fragmentação de conteúdo para melhoria da utilização da largura de banda, alteração dos protocolos de rede, localização de cache de forma inteligente, controle de acesso por conteúdos e não por fluxo e aplicação de testes em redes sem fio.

REFERÊNCIAS

ADHIKARI, Vijay K.; GUO, Yang; HAO, Fang; HILT, Volker; ZHANG, Zhi-Li; VARVELLO, Matteo; STEINER, Moritz. Measurement study of netflix, hulu, and a tale of three cdns. *IEEE/ACM Transactions on Networking*, v. 23, n. 6, p. 1984–1997, 2015. Citado 3 vezes nas páginas 16, 22 e 23.

AMIN, Rashid; REISSLEIN, Martin; SHAH, Nadir. Hybrid sdn networks: A survey of existing approaches. *IEEE Communications Surveys Tutorials*, v. 20, n. 4, p. 3259–3306, 2018. Citado 4 vezes nas páginas 14, 15, 27 e 28.

BALACHANDRAN, Athula; SEKAR, Vyas; AKELLA, Aditya; SESHAN, Srinivasan. Analyzing the potential benefits of cdn augmentation strategies for internet video workloads. In: *Proceedings of the 2013 Conference on Internet Measurement Conference*. New York, NY, USA: Association for Computing Machinery, 2013. (IMC '13), p. 43–56. ISBN 9781450319539. Disponível em: <https://doi.org/10.1145/2504730.2504743>. Citado na página 48.

BANNOUR, Fetia; SOUIHI, Sami; MELLOUK, Abdelhamid. Distributed sdn control: Survey, taxonomy, and challenges. *IEEE Communications Surveys Tutorials*, v. 20, n. 1, p. 333–354, 2018. Citado 4 vezes nas páginas 14, 15, 27 e 28.

BHIDE, M.; DEOLASEE, P.; KATKAR, A.; PANCHBUDHE, A.; RAMAMRITHAM, K.; SHENOY, P. Adaptive push-pull: disseminating dynamic web data. *IEEE Transactions on Computers*, v. 51, n. 6, p. 652–668, 2002. Citado na página 49.

BRITO, P. B. VELLOSO E I. M. MORAES G. M. DE. Redes orientadas a conteúdo: Um novo paradigma para a internet. Minicursos do Simpósio Brasileiro de Redes de Computadores-SBRC, v. 2012, n. capitulo 5, p. 211–264, jul. 2012. Disponível em: <http://www2.ic.uff.br/~igor/cursos/files/BVM12.pdf>. Citado 2 vezes nas páginas 16 e 19.

CARDELLINI, Valeria; COLAJANNI, Michele; YU, Philip S. Request redirection algorithms for distributed web systems. *IEEE Trans. Parallel Distrib. Syst.*, IEEE Press, v. 14, n. 4, p. 355–368, abr. 2003. ISSN 1045-9219. Disponível em: <https://doi.org/10.1109/TPDS.2003.1195408>. Citado na página 49.

CARVALHO, I; FARIA, F; CERQUEIRA, E; ABELEM, A. Contentflow: An introductory routing proposal for content centric networks using openflow. In: *API, 7th Think-Tank Meeting*. [S.l.: s.n.], 2012. p. 1–2. Citado 4 vezes nas páginas 7, 35, 36 e 41.

CHARPINEL, Sergio; SANTOS, Celso Alberto Saibel; VIEIRA, Alex Borges; VILLACA, Rodolfo; MARTINELLO, Magnos. Sdccn: A novel software defined content-centric networking approach. In: *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*. [S.l.: s.n.], 2016. p. 87–94. Citado 5 vezes nas páginas 7, 40, 41, 43 e 62.

Cicn. *Cicn*. 2021. Disponível em: <https://wiki.fd.io/view/Cicn>. Acesso em: 07 outubro 2021. Citado na página 46.

CISCO. Cisco visual networking index: Forecast and methodology, 2016–2020. jul. 2016. Disponível em: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networkingindex-vni/complete-white-paper-c11-481360.htm>. Citado 2 vezes nas páginas 15 e 16.

CISCO. *Instaling Packet tracer*. 2022. Urlhttps://www.netacad.com/courses/packet-tracer. Citado na página 58.

DARGAHI, Tooska; CAPONI, Alberto; AMBROSIN, Moreno; BIANCHI, Giuseppe; CONTI, Mauro. A survey on the security of stateful sdn data planes. *IEEE Communications Surveys Tutorials*, v. 19, n. 3, p. 1701–1725, 2017. Citado 4 vezes nas páginas 14, 15, 27 e 28.

DOAN, Trinh Viet; BAJPAI, Vaibhav; CRAWFORD, Sam. A longitudinal view of netflix: Content delivery over ipv6 and content cache deployments. In: *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*. [S.l.: s.n.], 2020. p. 1073–1082. Citado 3 vezes nas páginas 16, 22 e 23.

DOAN, Trinh Viet; BAJPAI, Vaibhav; CRAWFORD, Sam. A longitudinal view of netflix: Content delivery over ipv6 and content cache deployments. In: *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*. [S.l.: s.n.], 2020. p. 1073–1082. Citado na página 49.

GULATI, Amuleen; AUJLA, Gagangeet Singh; KUMAR, Neeraj; GARG, Sahil; KADDOUM, Georges. Software-defined content dissemination scheme for internet of healthcare vehicles in covid-like scenarios. *IEEE Internet of Things Magazine*, v. 4, n. 3, p. 34–40, 2021. Citado na página 41.

HARCHOL-BALTER, Mor; SCHROEDER, Bianca; BANSAL, Nikhil; AGRAWAL, Mukesh. Size-based scheduling to improve web performance. *ACM Trans. Comput. Syst.*, Association for Computing Machinery, New York, NY, USA, v. 21, n. 2, p. 207–233, maio 2003. ISSN 0734-2071. Disponível em: <https://doi.org/10.1145/762483.762486>. Citado na página 56.

HUFFAKER, Bradley; FOMENKOV, Marina; PLUMMER, Daniel; MOORE, David; CLAFFY, K. Distance metrics in the internet. 08 2002. Citado na página 56.

JEFFERY, C.L.; DAS, S.R.; BERNAL, G.S. Proxy-sharing proxy servers. In: *Proceedings of COM'96. First Annual Conference on Emerging Technologies and Applications in Communications*. [S.l.: s.n.], 1996. p. 116–119. Citado na página 31.

JMAL, Rihab; FOURATI, Lamia Chaari. Content-centric networking management based on software defined networks: Survey. *IEEE Transactions on Network and Service Management*, v. 14, n. 4, p. 1128–1142, 2017. Citado na página 16.

KAUR, Sukhveer; SINGH, Japinder; GHUMMAN, Navtej. Network programmability using pox controller. In: . [S.l.: s.n.], 2014. Citado 2 vezes nas páginas 46 e 58.

LINUXFOUNDATION. *HOw to use Open vSwitch*. 2022. Urlhttps://www.openvswitch.org/. Citado na página 58.

LIU, Yaoqing; WADEKAR, Hitesh. Sdar: Software defined intra-domain routing in named data networks. In: *2016 IEEE 15th International Symposium on Network Computing and Applications (NCA)*. [S.l.: s.n.], 2016. p. 158–161. Citado 3 vezes nas páginas 39, 41 e 43.

LONG, Wen-Guang; LI, Jian-Ping. Designing secure session based on reverse proxy. In: *2012 International Conference on Wavelet Active Media Technology and Information Processing (ICWAMTIP)*. [S.l.: s.n.], 2012. p. 299–301. Citado na página 32.

MATTOS, Diogo; DUARTE, Otto Carlos; PUJOLLE, Guy. Segurança em redes definidas por software: Autenticação, controle de acesso e consistência com plano de controle eficientemente distribuído. In: *Anais Estendidos do XXXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. Porto Alegre, RS, Brasil: SBC, 2018. ISSN 2177-9384. Disponível em: <https://sol.sbc.org.br/index.php/sbrc_estendido/article/view/14185>. Citado 4 vezes nas páginas 7, 34, 41 e 43.

MCKEOWN, Nick; ANDERSON, Tom; BALAKRISHNAN, Hari; PARULKAR, Guru; PETERSON, Larry; REXFORD, Jennifer; SHENKER, Scott; TURNER, Jonathan. Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, Association for Computing Machinery, New York, NY, USA, v. 38, n. 2, p. 69–74, mar. 2008. ISSN 0146-4833. Disponível em: <https://doi.org/10.1145/1355734.1355746>. Citado 5 vezes nas páginas 7, 27, 28, 29 e 30.

METZ, C. Aaa protocols: authentication, authorization, and accounting for the internet. *IEEE Internet Computing*, v. 3, n. 6, p. 75–79, 1999. Citado na página 32.

Mininet. *Mininet*. 2021. Disponível em: <http://mininet.org/>. Acesso em: 07 outubro 2021. Citado 2 vezes nas páginas 44 e 45.

NGUYEN, Xuan Nam; SAUCEZ, Damien; TURLETTI, Thierry. Efficient caching in content-centric networks using openflow. In: *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. [S.l.: s.n.], 2013. p. 67–68. Citado 2 vezes nas páginas 37 e 41.

OLSHEFSKI, David; NIEH, Jason; AGRAWAL, Dakshi. Using certes to infer client response time at the web server. *ACM Trans. Comput. Syst.*, Association for Computing Machinery, New York, NY, USA, v. 22, n. 1, p. 49–93, fev. 2004. ISSN 0734-2071. Disponível em: <https://doi.org/10.1145/966785.966787>. Citado na página 56.

ONF. *Mininet*. 2021. Disponível em: <https://opennetworking.org/mininet/>. Acesso em: 07 outubro 2021. Citado 2 vezes nas páginas 44 e 45.

ONF. *Using Mininet*. 2022. Urlhttps://opennetworking.org/mininet/. Citado na página 58.

OOKA, Atsushi; ATA, Shingo; KOIDE, Toshio; SHIMONISHI, HIDEyuki; MURATA, Masayuki. Openflow-based content-centric networking architecture and router implementation. In: *2013 Future Network Mobile Summit*. [S.l.: s.n.], 2013. p. 1–10. Citado 2 vezes nas páginas 36 e 41.

OPENFLOW. The openflow switch specification. november 2017. Disponível em: <http://OpenflowSwitch.org>. Citado 5 vezes nas páginas 7, 15, 28, 29 e 30.

ORACLE. *Installing Mysql*. 2022. Urlhttps://dev.mysql.com/doc/refman/8.0/en/. Citado na página 58.

PATHAN, Mukaddim; BUYYA, Rajkumar; VAKALI, Athena. Content delivery networks: State of the art, insights, and imperatives. In: _____. *Content Delivery Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 3–32. ISBN 978-3-540-77887-5. Disponível em: <https://doi.org/10.1007/978-3-540-77887-5_1>. Citado 4 vezes nas páginas 7, 19, 20 e 21.

PITT, Dan. *SDN Networks*. 2015. Urlhttps://www.networkcomputing.com/networking/understanding-openflow-vxlan-and-ciscos-aci. Citado na página 28.

ProjectCCNx. *ProjectCCNx*. 2021. Disponível em: <https://github.com/ProjectCCNx/ccnx>. Acesso em: 07 outubro 2021. Citado na página 46.

PYTHON. *Programming Python*. 2022. Urlhttps://www.python.org/. Citado na página 58.

QIAO, Xiuquan; WANG, Hongyi; TAN, Wei; VASILAKOS, Athanasios V.; CHEN, Junliang; BLAKE, M. Brian. A survey of applications research on content-centric networking. *China Communications*, v. 16, n. 9, p. 122–140, 2019. Citado 3 vezes nas páginas 14, 15 e 16.

QIU, Lili; PADMANABHAN, V.N.; VOELKER, G.M. On the placement of web server replicas. In: *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*. [S.l.: s.n.], 2001. v. 3, p. 1587–1596 vol.3. Citado na página 49.

ROWSHANRAD, Shiva; PARSAEI, Mohamad Reza; KESHTGARI, Manijeh. Implementing ndn using sdn: A review on methods and applications. *IIUM Engineering Journal*, v. 17, n. 2, p. 11–20, Nov. 2016. Disponível em: <https://journals.iium.edu.my/ejournal/index.php/iiumej/article/view/590>. Citado 2 vezes nas páginas 35 e 43.

SALAHUDDIN, Mohammad A.; SAHOO, Jagruti; GLITHO, Roch; ELBIAZE, Halima; AJIB, Wessam. A survey on content placement algorithms for cloud-based content delivery networks. *IEEE Access*, v. 6, p. 91–114, 2018. Citado 4 vezes nas páginas 15, 16, 22 e 23.

SALSANO, S.; BLEFARI-MELAZZI, N.; DETTI, A.; MORABITO, G.; VELTRI, L. Information centric networking over sdn and openflow: Architectural aspects and experiments on the ofelia testbed. *Computer Networks*, Elsevier BV, v. 57, n. 16, p. 3207–3221, Nov 2013. ISSN 1389-1286. Disponível em: <http://dx.doi.org/10.1016/j.comnet.2013.07.031>. Citado 4 vezes nas páginas 7, 37, 38 e 41.

SHINDE, Anita; CHAWARE, S. M. Content centric networks (ccn): A survey. In: *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on*. [S.l.: s.n.], 2018. p. 595–598. Citado na página 16.

SIVASUBRAMANIAN, Swaminathan; SZYMANIAK, Michal; PIERRE, Guillaume; STEEN, Maarten van. Replication for web hosting systems. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 36, n. 3, p. 291–334, set. 2004. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/1035570.1035573>. Citado na página 56.

SON, Jaehyeok; KIM, DoHyeon; KANG, Hyo Sung; HONG, Choong Seon. Forwarding strategy on sdn-based content centric network for efficient content delivery. In: *2016 International Conference on Information Networking (ICOIN)*. [S.l.: s.n.], 2016. p. 220–225. Citado 3 vezes nas páginas 38, 41 e 43.

VELKOSKI, Goran; SIMJANOSKA, Monika; RISTOV, Sasko; GUSEV, Marjan. Cpu utilization in a multitenant cloud. In: *Eurocon 2013*. [S.l.: s.n.], 2013. p. 242–249. Citado na página 33.

WANG, Jingguo; SHARMAN, Raj; RAMESH, Ram. Shared content management in replicated web systems: A design framework using problem decomposition, controlled simulation, and feedback learning. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, v. 38, p. 110 – 124, 02 2008. Citado na página 49.

WANG, Kan; LI, Hongyan; YU, F. Richard; WEI, Wenchao. Virtual resource allocation in software-defined information-centric cellular networks with device-to-device communications and imperfect csi. *IEEE Transactions on Vehicular Technology*, v. 65, n. 12, p. 10011–10021, 2016. Citado na página 48.

YAO, Haipeng; QIU, Chao; FANG, Chao; CHEN, Xu; YU, F. Richard. *Content-Centric and Software-Defined Networking with Big Data*. 2016. Citado 3 vezes nas páginas 34, 41 e 43.

ZHANG, Lixia; AFANASYEV, Alexander; BURKE, Jeffrey; JACOBSON, Van; CLAFFY, kc; CROWLEY, Patrick; PAPADOPOULOS, Christos; WANG, Lan; ZHANG, Beichuan. Named data networking. *SIGCOMM Comput. Commun. Rev.*, Association for Computing Machinery, New York, NY, USA, v. 44, n. 3, p. 66–73, jul. 2014. ISSN 0146-4833. Disponível em: <https://doi.org/10.1145/2656877.2656887>. Citado 3 vezes nas páginas 14, 15 e 16.

ZHANG, Lixia; ESTRIN, Deborah; BURKE, Jeffrey; JACOBSON, Van; THORNTON, James; SMETTERS, Diana; ZHANG, Beichuan; TSUDIK, Gene; CLAFFY, Kc; KRIOUKOV, Dmitri; MASSEY, Dan; PAPADOPOULOS, Christos; ABDELZAHER, Tarek; WANG, Lan; CROWLEY, Patrick; YEH, Edmund. Named data networking (ndn) project. 05 2012. Citado 6 vezes nas páginas 7, 23, 24, 25, 26 e 27.