

UNIVERSIDADE FEDERAL FLUMINENSE

RICARDO WILLIAMS SANCHEZ JUNIOR

**ADAPTAÇÃO DE TAXA DE PACOTES DE
INTERESSE PARA STREAMING DE VÍDEO EM
REDES CENTRADAS EM CONTEÚDO**

NITERÓI

2015

UNIVERSIDADE FEDERAL FLUMINENSE

RICARDO WILLIAMS SANCHEZ JUNIOR

ADAPTAÇÃO DE TAXA DE PACOTES DE INTERESSE PARA STREAMING DE VÍDEO EM REDES CENTRADAS EM CONTEÚDO

Dissertação de Mestrado apresentada
ao Programa de Pós-Graduação em
Engenharia de Telecomunicações da
Universidade Federal Fluminense como
requisito parcial para a obtenção do Grau de
Mestre em Engenharia de Telecomunicações.
Área de concentração: Comunicação de Da-
dos Multimídia

Orientador:

NATALIA CASTRO FERNANDES

NITERÓI

2015

Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de Computação da UFF

S211 Sanchez Junior, Ricardo Williams

Adaptação de taxa de pacotes de interesse para streaming de vídeo em redes centradas em conteúdo / Ricardo Williams Sanchez Junior.
– Niterói, RJ : [s.n.], 2015.
88 f.

Dissertação (Mestrado em Engenharia de Telecomunicações) -
Universidade Federal Fluminense, 2015.
Orientador: Natalia Castro Fernandes.

1. Arquitetura de redes de computadores. 2. Vídeo digital. 3. Rede Centrada de Conteúdo. I. Título.

CDD 004.6

RICARDO WILLIAMS SANCHEZ JUNIOR

ADAPTAÇÃO DE TAXA DE PACOTES DE INTERESSE PARA STREAMING DE
VÍDEO EM REDES CENTRADAS EM CONTEÚDO

Dissertação de Mestrado

Aprovada em março de 2015.

BANCA EXAMINADORA

Prof. Natalia Castro Fernandes, D.Sc. - Orientador, UFF

Prof. Lisandro Lovisolo, D.Sc. , UERJ

Prof. Débora Christina Muchaluat Saade, D.Sc. , UFF

Niterói

2015

Dedico esta dissertação a minha família, a minha esposa Fernanda que sempre me apoiou e me incentivou e a minha filha Julia que com seu lindo sorriso cativa a todos!

Agradecimentos

Agradeço, primeiramente, a Deus por todas as bênçãos que ele propicia em minha vida.

Agradeço aos meus pais, que me educaram e apoiaram a buscar sempre mais e a vencer os desafios da vida. Se hoje sou a pessoa que sou, devo muito a eles.

Agradeço à minha esposa, que sempre me apoiou e esteve ao meu lado nos momentos de maior necessidade ao longo deste curso, sempre cobrando e incentivando. Agradeço especialmente a ela pela alegria maior do mundo de poder ser pai da nossa linda filha e fruto do nosso amor!

Agradeço à minha família, à minha irmã, aos meus avós, tios e tias.

Um agradecimento especial a minha orientadora Natalia Castro Fernandes, por toda a sua atenção, sua dedicação e disponibilidade em me ajudar a vencer o grande desafio de concluir este trabalho de dissertação e principalmente de me formar.

Agradeço às professoras Leni Joaquim de Matos e Débora Christina Muchaluat e ao professor Julio Cesar Dal Bello que ao longo desses anos sempre acreditaram em mim e me ajudaram a concluir o curso.

Agradeço à banca pela oportunidade de apresentar este trabalho desafiador e motivante.

Não poderia deixar de agradecer às pessoas do meu trabalho que me ajudaram ao longo de todo o curso: aos meus chefes Rafael Freitas de Amorim e Marcos Henrique Wandenkolk Vieira por me possibilitarem realizar o curso de Mestrado em conjunto com o trabalho e as ajudas sempre essenciais em Linux de Fabio Teixeira Xavier.

É chegada a hora da comemoração!

Resumo

As redes evoluíram do conceito de apenas permitir conectar computadores para realizar a distribuição de conteúdos para todo o mundo, a qualquer dispositivo e a qualquer momento que o usuário desejar. Para isso acontecer, novos serviços e tecnologias foram desenvolvidos utilizando as redes existentes. Os serviços de *streaming* de vídeo são um exemplo dessa evolução, ao mesmo tempo que representam um desafio, devido à forma como consomem os recursos da rede. Neste cenário foi desenvolvido o MPEG-DASH, tecnologia responsável pela melhora na experiência de uso de *streamings* de vídeo na Internet e que possibilita uma maior eficiência no consumo de recursos da rede, ao utilizar transmissões com taxas de bits adaptativas. Apesar de tamanha evolução nas aplicações, a arquitetura existente na Internet é a mesma utilizada há 40 anos, de forma que a comunicação está atrelada aos pontos de origem e de destino, mesmo o padrão das aplicações tendo mudado de ponto a ponto para a distribuição de conteúdo. Nas redes IP, o conhecimento da informação trafegada é invisível ou limitada, dificultando uma distribuição mais eficiente dos dados na rede. Diante deste cenário, foi concebida a Rede Centrada em Conteúdos (CCN), que é uma nova arquitetura de rede planejada para trafegar melhor as informações e não simplesmente realizar a conexão entre duas máquinas. Para isso, uma nova abordagem de rede foi concebida, voltada para a entrega mais eficiente de conteúdos, onde o que importa é a disponibilidade do conteúdo e não a fonte original dos dados. Sendo assim, o uso de tecnologias de *streaming* de vídeo tais como o MPEG-DASH torna-se pertinente, principalmente objetivando o uso de mecanismos de controle de taxas de bit adaptativas em conjunto com as redes CCN.

Embora o *streaming* com taxa de bits adaptativa seja de grande relevância dentro do contexto de CCNs, faltam estudos, em especial por meio de simulação, que permitam avaliar como funcionam as CCNs em diferentes cenários de transmissão de vídeo. Dessa forma, nesta dissertação, é realizado um estudo prático da transmissão de vídeos em CCN, assim como é proposto um mecanismo de controle adaptativo de taxa de pacotes de interesse para transmissão de vídeo em CCN que se assemelha ao MPEG DASH para redes TCP/IP tradicionais. São propostos e avaliados dois mecanismos para a adaptação da taxa de requisição de interesses, os quais são disparados sempre que eventos que indiquem congestionamentos na rede forem detectados. Em especial, neste estudo, foi controlada a taxa de NACKs como forma de monitorar a rede e adaptar a taxa do conteúdo. Foi desenvolvido um módulo de simulação para o ndnSIM para avaliação de adaptação de taxa de dados em *streamings* de vídeo, observando o seu desempenho em diferentes cenários e utilizando diferentes configurações de parâmetros.

Palavras-chave: CCN, MPEG-DASH, *Streaming*, Vídeo, Taxas Adaptativas.

Abstract

Networks have evolved from their original concept of connecting just computers to distribute contents for the whole world, to any device at any moment that a user wants. To make this happen, new services and technologies were developed using the existent networks. Video streaming is one example of this evolution. At the same time, they represent a challenge because of the way they consume network resources. In this scenario MPEG-DASH was developed, a technology responsible for video streaming improvements over the Internet by allowing a better efficiency on the consumption of network resources with the usage of adaptive bit rate transmissions. Despite the huge evolution on applications, the existing IP architecture is the same over 40 years, in a way that the communication is tied to endpoints, even though application standards had changed from a point to point distribution to content distribution. In IP networks, the knowledge of the transmitted information is invisible or limited, making it harder to distribute contents within the network. Facing this scenario, Content Centric Networks was conceived as a new network architecture, planned to better transmit the information and not only to simply connect two machines. For this, a new network approach was done, aimed at making content delivery more efficient, where what matters is the content availability and not the content original source. Therefore, the usage of video streaming technologies such as MPEG-DASH became important, especially targeting the use of adaptive bit rate control together with CCN networks.

Although streaming with adaptive bit rate has a big relevance within the CCN context, there is a lack of studies, in special simulations, that allows evaluating how CCN works in different video transmission scenarios. This way, in this dissertation a practical study of video transmission over CCN is done. A new mechanism for adaptive control of interest packet rate for video transmission over CCN is also proposed that resembles MPEG DASH of traditional TCP/IP networks. It is proposed and evaluated two mechanisms of adaptive rate of interest requests, which are always triggered in indication of network congestion event. In special, in this study, NACK rate was controlled as a way to monitor the network and adapt the content rate. A new simulation module from ndnSIM was developed to make an evaluation of adaptive data rate on video streaming, observing different performance scenarios and using different parameter configurations.

Keywords: Adaptive Rate, CCN, MPEG-DASH, Streaming, Video.

Lista de Figuras

1.1	Analogia entre o atendimento de uma requisição em uma rede CDN e uma rede CCN	18
2.1	No modo <i>multicast</i> , apenas um fluxo de vídeo é trafegado na rede para os <i>hosts</i> que estão conectados a transmissão	22
2.2	Ilustração do uso do <i>buffer</i> de exibição no dispositivo cliente	24
2.3	Estrutura em blocos do DASH	27
2.4	Exemplo de funcionamento da CDN ilustrando o processo de redirecionamento de requisições e entrega de conteúdo	29
3.1	Comparação entre a pilha de protocolos de uma rede TCP/IP com uma CCN [26].	33
3.2	Tabelas usadas para gerência dos pacotes de interesse e dos conteúdos armazenados em um nó CCN [56].	36
3.3	Representação do funcionamento de um nó CCN	39
4.1	Representação das similaridades na nomeação dos conteúdos no DASH e na CCN	43
4.2	Proposta de integração do DASH com CCN	44
5.1	Diagrama em blocos dos módulos que compõem a arquitetura da proposta	47
5.2	Lógica detalhada do funcionamento da arquitetura proposta	49
5.3	Detalhamento da implementação utilizada no mecanismo de controle . . .	53
5.4	Componentes do ndnSIM - http://ndnsim.net/	53
5.5	Módulos do Simulador ndnSIM e as classes modificadas.	54
6.1	Topologia para o teste de desempenho do DASH para CCN	57
6.2	Parâmetros do Vídeo de Teste	58

6.3	Estatísticas de reprodução padrão do VLC.	60
6.4	Percentual Médio de Perda de Vídeo	62
6.5	Percentual Médio de Atraso na Reprodução do Vídeo	63
6.6	Tempo de <i>Freezing</i> na Reprodução do Vídeo	63
6.7	Topologia do simulador ndnSIM para os Testes de Simulação	64
6.8	Frequência de envio de Pacotes de Interesse no Tempo - Topologia de Rede sem Gargalos e Frequência Inicial de 100 pacotes.	68
6.9	Frequência de envio de Pacotes de Interesse no Tempo - Topologia de Rede com Gargalos e Frequência Inicial de 100 pacotes de interesse enviados. . .	69
6.10	Taxa de Recebimento de Pacotes de Dados no Tempo - Topologia de Rede sem Gargalos e Frequência Inicial de 100 pacotes de interesse enviados. . .	70
6.11	Taxa de Recebimento de Pacotes de Dados no Tempo - Topologia de Rede com Gargalos e Frequência Inicial de 100 pacotes de interesse enviados. . .	70
6.12	Frequência de envio de Pacotes de Interesse no Tempo - Topologia de Rede sem Gargalos e Frequência Inicial de 500 pacotes de interesse enviados. . .	72
6.13	Frequência de envio de Pacotes de Interesse no Tempo - Topologia de Rede com Gargalos e Frequência Inicial de 500 pacotes de interesse enviados. . .	73
6.14	Taxa de Recebimento de Pacotes de Dados no Tempo - Topologia de Rede sem Gargalos e Frequência Inicial de 500 pacotes de interesse enviados. . .	74
6.15	Taxa de Recebimento de Pacotes de Dados no Tempo - Topologia de Rede com Gargalos e Frequência Inicial de 500 pacotes de interesse enviados. . .	74
6.16	Frequência de envio de Pacotes de Interesse no Tempo - Topologia de Rede sem Gargalos e Frequência Inicial de 2000 pacotes de interesse enviados. . .	75
6.17	Frequência de envio de Pacotes de Interesse no Tempo - Topologia de Rede com Gargalos e Frequência Inicial de 2000 pacotes de interesse enviados. . .	76
6.18	Taxa de Recebimento de Pacotes de Dados no Tempo - Topologia de Rede sem Gargalos e Frequência Inicial de 2000 pacotes de interesse enviados. . .	77
6.19	Taxa de Recebimento de Pacotes de Dados no Tempo - Topologia de Rede com Gargalos e Frequência Inicial de 2000 pacotes de interesse enviados. . .	77

6.20 Média da Taxa de Pacotes de Dados Recebidos por Segundo.	79
6.21 Buffer de Recepção Acumulado pela Frequência Inicial	80

Lista de Tabelas

6.1	Cenários de testes de simulação de transmissão de vídeo para CCN.	59
6.2	Cenários de Teste de Simulação	65
6.3	Índices de aumento e decréscimo da frequência.	65

Lista de Abreviaturas e Siglas

AIMD	:	<i>Additive Increase Multiplicative Decrease;</i>
AISD	:	<i>Additive Increase Subtrative Decrease;</i>
ARP	:	<i>Address Resolution Protocol;</i>
CCN	:	<i>Content Centric Network;</i>
CDN	:	<i>Content Distribution Network;</i>
CS	:	<i>Content Store ;</i>
DASH	:	<i>Dynamic Adaptive Streaming over HTTP;</i>
DCT	:	<i>Discrete Cosine Transform;</i>
FIB	:	<i>Forwarding Information Base;</i>
GOP	:	<i>Group Of Pictures;</i>
HD	:	<i>High Definition;</i>
HTTP	:	<i>HyperText Transfer Protocol;</i>
ICN	:	<i>Information-Centric Networking;</i>
IETF	:	<i>Internet Engineering Task Force;</i>
IP	:	<i>Internet Protocol;</i>
IPTV	:	<i>Internet Protocol Television;</i>
LFU	:	<i>Least Frequently Used ;</i>
LRU	:	<i>Least Recently Used;</i>
LTS	:	<i>Long Term Support;</i>
MPEG	:	<i>Motion Pictures Expert Group;</i>
MTU	:	<i>Maximum Transmission Unit;</i>
NACK	:	<i>Negative Acknowledgement ;</i>
NAT	:	<i>Network Address Translation;</i>
NDN	:	<i>Named Data Networking;</i>
ndnSIM	:	<i>NDN Simulator ;</i>

NS-3	:	<i>Network Simulator 3;</i>
OSI	:	<i>Open Systems Interconnection;</i>
OTT	:	<i>Over The Top;</i>
PIT	:	<i>Pending Interesse Table;</i>
RAM	:	<i>Random-Access Memory ;</i>
RFC	:	<i>Request For Comments;</i>
RTCP	:	<i>Real-Time Transport Control Protocol ;</i>
RTP	:	<i>Real-Time Transport Protocol ;</i>
RTSP	:	<i>Real Time Streaming Protocol;</i>
RTT	:	<i>Round Trip Time;</i>
SD	:	<i>Standard Definition;</i>
SDN	:	<i>Software-Defined Networking;</i>
TCP	:	<i>Transmission Control Protocol;</i>
TOS	:	<i>Type Of Services;</i>
TTL	:	<i>Time To Live;</i>
UDP	:	<i>User Datagram Protocol;</i>
URI	:	<i>Uniform Resource Identifier;</i>
URL	:	<i>Uniform Resource Locator;</i>
UTF	:	<i>Unicode Transformation Format;</i>
VLC	:	<i>Video Lan Client;</i>
VM	:	<i>Virtual Machine;</i>
VOD	:	<i>Video On Demand;</i>

Sumário

1	Introdução	14
1.1	Motivação	15
1.2	Objetivos	17
1.3	Organização do texto	18
2	Modelos de Comunicação IP, Streamings Adaptativos e Redes CDN	20
2.1	Modelos de Comunicação IP	20
2.1.1	Comunicações <i>Unicast</i> e <i>Multicast</i>	20
2.1.2	Comunicação por Streaming	23
2.1.2.1	Streaming por HTTP	25
2.1.2.2	Streaming por HTTP Adaptativo e o padrão MPEG DASH	25
2.2	Redes de Distribuição de Conteúdo	28
2.3	Internet do Futuro	30
3	Fundamentos de Redes CCN	32
3.1	A Arquitetura CCN	32
3.2	Fundamentos de Redes CCN	34
3.2.1	Os Pacotes CCN	34
3.2.2	Nomeação dos conteúdos nas redes CCN	35
3.2.3	O funcionamento dos nós CCN	35
3.2.4	Equilíbrio do Fluxo de Transportes	40
4	Trabalhos Relacionados	42

5	Proposta	46
5.1	Arquitetura da Proposta	47
5.1.1	Módulo de Monitoração da Rede	48
5.1.2	Módulo do Mecanismo de Controle	48
5.1.3	Módulo de Adequação das Requisições e Agendamento de Envio . .	50
5.1.4	Escalonador	52
5.2	Implementação da proposta	52
5.3	O Simulador ndnSIM	53
6	Análise da transmissão de vídeo em redes CCN	56
6.1	Etapa 1 – Análise da transmissão de vídeo em redes CCN	56
6.2	Etapa II - Análise do Controle da Taxa de Interesse	64
6.2.1	Resultados Obtidos	67
7	Conclusão	81
7.1	Trabalhos Futuros	82
	Referências	84

Capítulo 1

Introdução

As redes IP, amplamente utilizadas hoje em dia, tiveram o início de seu desenvolvimento nas décadas de 1960 e 1970, com o objetivo de fazer duas máquinas compartilharem recursos computacionais que, na época, eram escassos. Hoje, a finalidade no acesso a rede é outra. Os usuários acessam a rede buscando informações, interações sociais e, principalmente, entretenimento, sendo o vídeo o serviço de principal destaque. Segundo a previsão apresentada pela Cisco, o uso das redes IP em 2018 será caracterizado por: 79% do tráfego IP global sendo vídeo sobre IP (eram 66% em 2013), 83% do tráfego de um usuário será vídeo sobre IP (eram 73% em 2013) e um tráfego global total da ordem de 103 Exabytes por mês, mais do que o triplo do tráfego presente em 2013 que era da ordem de 33 Exabytes por mês. Esse aumento é caracterizado pela popularização de serviços como Netflix e Youtube e presença destes tipos de serviços nos mais diversos dispositivos. Neste cenário, os usuários possuem o controle de qual conteúdo específico eles querem ver, o local e em qual dispositivo desejarem, e principalmente no momento quiserem [43]. Esses recursos multimídia aumentam significativamente a complexidade requerida na transmissão, tornando-a cada vez mais customizada de acordo com a experiência do usuário. Essa customização na transmissão dos dados para os usuários é possibilitada, principalmente, pelos mecanismos de taxas de transmissão adaptativas (destacando-se o padrão MPEG-DASH (*Motion Pictures Expert Group - Dynamic Adaptive Streaming over HTTP*)), os quais transmitem aos usuários a melhor taxa que o seu dispositivo está habilitado a receber, de acordo com o desempenho instantâneo da rede, melhorando a eficiência na entrega dos conteúdos. Além dos mecanismos de transmissão adaptativas, o uso de técnicas de armazenamento de conteúdos, como os empregados nas redes CDN (*Content Distribution Network*) possibilitam o aumento da eficiência na entrega de conteúdos populares, assim como minimizam o impacto no consumo de recursos na rede. Observando-se o cenário de

utilização das redes CDN em conjunto com o uso de MPEG DASH, ocorre de os *caches* da rede CDN armazenarem todos os segmentos de vídeo solicitados durante o intervalo de armazenamento permitido pela política definida pelo administrador da rede. Neste caso, novas requisições HTTP de um segmento já armazenado no próprio nó serão entregues diretamente, sem a necessidade de encaminhar a requisição através de toda a rede até o servidor contendo o conteúdo. Diante dos novos requisitos dos atuais serviços, uma nova abordagem de arquitetura de rede está sendo planejada. As redes centradas em conteúdos (*Content Centric Network* - CCN) são uma nova arquitetura de rede pensada para a entrega de dados em função do conteúdo requisitado, em vez do uso do endereço de destino. Como será apresentado no decorrer desta dissertação, essa nova arquitetura possui, nativamente, diversos recursos existentes nas redes IPs atuais e outros que ainda necessitam de desenvolvimento, tais como um mecanismo de controle das taxas para envio/recepção de conteúdo. Dada a sua importância e a sua capacidade, o uso das redes CCN para a entrega de vídeo possui um grande campo de desenvolvimento e usabilidade.

1.1 Motivação

As redes IP apresentam diferentes mecanismos para transmissão de vídeo, tais como fluxos *unicast/multicast*, os *streamings* em UDP, TCP (HTTP) e as redes CDN (*Content Distribution Network*). Cada um deles possui características próprias que os tornam mais indicados para determinados tipos de serviços. Para os serviços de vídeo baseados no modelo de transmissão a cabo tradicional, tais como IPTV (*Internet Protocol Television*) e transmissões de canais ao vivo através da Internet, o *multicast* apresenta-se como o modelo de transmissão padrão. Neste caso, o controle da taxa de transmissão é todo feito pelo servidor a partir de informações enviadas pelo dispositivo do usuário, podendo inclusive o servidor transmitir *stream multicast* sem que exista uma requisição de conteúdo ativa. Dependendo do protocolo utilizado pelo cliente, têm-se o uso de mecanismos de controle da taxa de transmissão e de retransmissão.

Para os serviços de vídeo sob demanda, tais como VOD (*Video on Demand*), além de serviços de *streaming* via Internet como o Netflix e Youtube, onde a transmissão do vídeo é específica e direcionada para cada usuário, o *unicast* é o modelo de transmissão adotado. Neste caso, em muitas das vezes, o controle da taxa de transmissão é feito principalmente pelo cliente, de acordo como o *buffer* é utilizado. Se a taxa de consumo do *buffer* for maior do que a taxa em que ele é preenchido, a reprodução do vídeo poderá ser afetada com interrupções na exibição até que *buffer* armazene uma parcela do conteúdo

suficiente para retomar a exibição. Caso o *buffer* seja enchido mais rápido que a taxa em que ele é consumido, talvez seja necessário informar à fonte transmissora para suspender a transmissão de forma a evitar a sobrecarga no *buffer* e consequente descarte de pacotes.

O controle pelo cliente também é percebido nos *streams* em HTTP (*Hypertext Transfer Protocol*) adaptativos, onde a taxa de transmissão é variável não apenas pelo controle do *buffer*, como também de acordo com o estado da rede. Desta forma, os *streams* em HTTP adaptativos fazem a requisição de segmentos do mesmo conteúdo com perfis de qualidade de imagem diferentes, de forma a manter a continuidade da exibição do conteúdo. Os perfis de maior qualidade de imagem requerem uma maior taxa e, consequentemente, acarretam em um maior consumo de recursos da rede. Caso o estado da rede degrade, a aplicação cliente poderá requisitar segmentos do conteúdo de qualidade menor, o que requer uma menor taxa de dados e, consequentemente, acarreta em um menor consumo de recursos da rede a fim de evitar impactos na exibição do conteúdo.

Conforme será detalhado no Capítulo 3, as redes CCN possuem dois tipos de pacotes, os pacotes de interesse e os pacotes de dados. Os pacotes de interesse são os pacotes responsáveis pela requisição de um conteúdo e os pacotes de dados são pacotes que contêm o conteúdo requisitado ou fragmentos dele. No caso de transmissões em redes CCN, a condição fundamental para a transmissão de um pacote de dados é a existência, obrigatoriamente, de um pacote de interesse associado. Assim, o modelo de transmissão CCN pode ser tratado de forma análoga a um fluxo *unicast* no qual a transmissão é iniciada somente com a existência de uma requisição inicial e todo controle da taxa de recebimento de dados é feito diretamente pelo lado cliente, que ao controlar a taxa de envio de requisições de interesses, irá consequentemente controlar a taxa de recebimento de pacotes de dados. Como o tamanho do pacote de interesse é menor do que o tamanho do pacote de dados, ao realizar o controle da taxa de envio de interesses, acarreta-se indiretamente em um controle contra congestionamentos na rede, pois se diminui o consumo de recursos na rede ao diminuir a taxa de envio de pacotes de dados.

Conforme ilustrado na Figura 1.1, o cenário de uma transmissão CCN pode ser comparado ao cenário de uma rede CDN com o uso de MPEG-DASH. A primeira requisição de um conteúdo feito pelo cliente em uma rede CDN será redirecionada até o servidor de origem do conteúdo (adotando-se como critério o fato de os nós intermediários não possuírem o arquivo requisitado). Ao ser transmitido pela rede desde o servidor de origem, o conteúdo requisitado será armazenado durante um período de tempo no *cache* dos nós intermediários da rede CDN. Se durante este período de tempo, uma segunda requisição

do mesmo conteúdo for recebida, o nó que possuir uma cópia do conteúdo em seu cache, será escolhido para atendimento da nova solicitação. A seleção do nó para atender a entrega do conteúdo será feita de acordo com as políticas definidas pelo administrador da rede. No caso da rede CCN, caso o conteúdo requisitado não esteja armazenado nos repositórios locais dos nós intermediário, o pacote de interesse somente será atendido pelo nó que publicou o conteúdo (servidor de origem). Os pacotes de dados ao serem transmitidos pelo caminho de volta até o cliente, também serão armazenados nos repositórios locais dos nós intermediários. Neste caso, ao receber uma segunda requisição do mesmo conteúdo, o primeiro nó que tiver uma cópia do conteúdo em seu repositório local, será o responsável pela entrega do pacote de dados associado, sem a necessidade de buscar o conteúdo diretamente no nó que publicou o conteúdo. Além da similaridade no modelo de entrega de conteúdos, a analogia também é notada em relação à forma de nomeação dos conteúdos transmitidos na rede. Nas redes CCN, utiliza-se as URI (*Uniform Resource Identifier*) para padrão de nomeação dos conteúdos e em uma rede CDN, como o endereçamento é baseado na localização do conteúdo, utiliza-se uma URL (*Uniform Resource Locator*) para o endereçamento das requisições HTTP, tendo ambas uma estrutura bastante similar entre si.

Apesar de ser uma rede própria para a distribuição de conteúdos, as redes CCN ainda carecem de uso em larga escala para transmissões de vídeo. Uma das possibilidades motivadoras deste fato, pode ser devido a quantidade limitada de propostas de mecanismos de controle de taxa. Sendo assim, embora existam implementações que realizem a transmissão de vídeo em redes CCN, ainda não está padronizado um mecanismo que faça o controle adaptativo da taxa de pacotes de interesse de tal forma que a transmissão de vídeo respeite as condições de congestionamento da rede e não comprometa a qualidade da experiência do usuário final.

1.2 Objetivos

Diante deste fato, a principal motivação desta dissertação é viabilizar uma técnica de controle adaptativo de taxa de pacotes de interesse para *streaming* de vídeo em redes centradas em conteúdo. Dessa forma, são alvo dessa dissertação os seguintes objetivos:

- Propor um mecanismo de controle adaptativo de taxa de pacotes de interesse para a transmissão de vídeo em CCN que se assemelhe ao MPEG DASH;
- Desenvolver um módulo de simulação para o ndnSIM para avaliação de adaptação

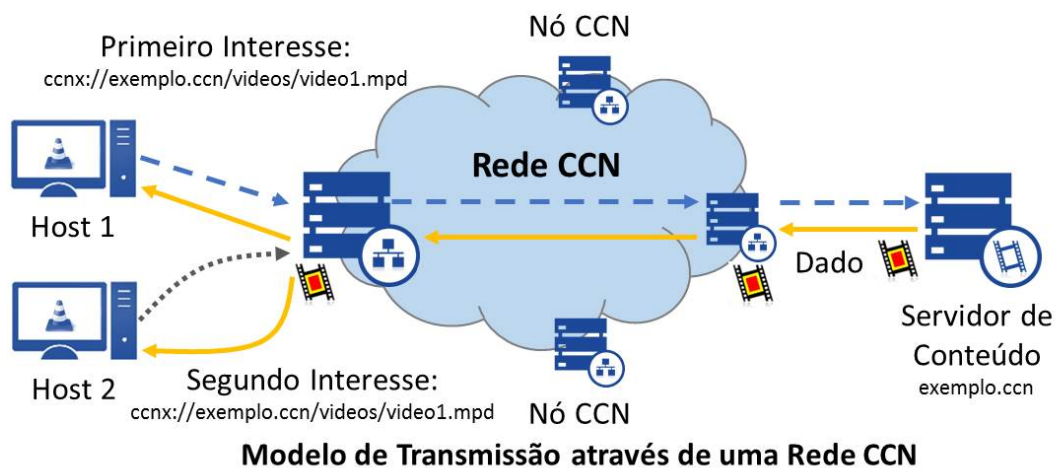
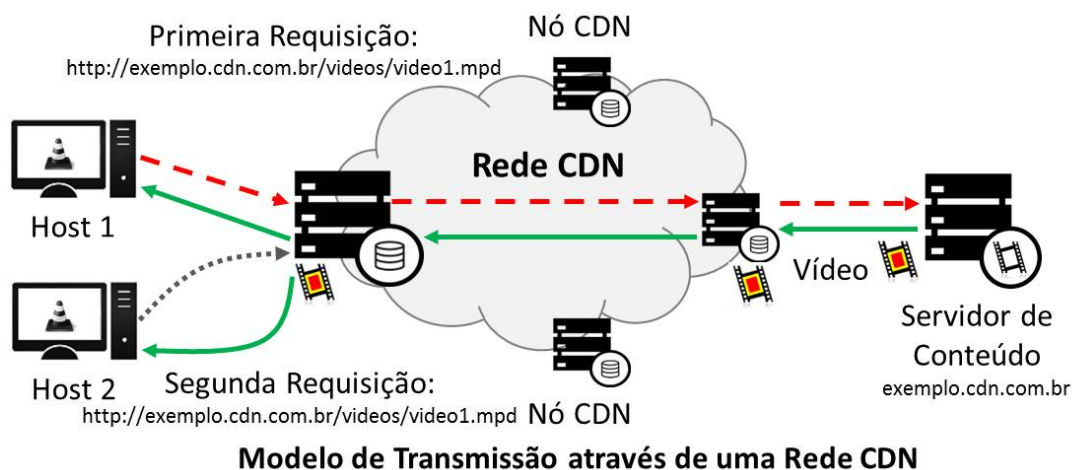


Figura 1.1: Analogia entre o atendimento de uma requisição em uma rede CDN e uma rede CCN

de taxa de dados em *streamings* de vídeo; e

- Analisar a proposta, observando o seu desempenho em diferentes cenários e utilizando diferentes configurações de parâmetros da rede.

1.3 Organização do texto

Esta dissertação está estruturada como definido a seguir.

No Capítulo 2, são apresentados os padrões de comunicação, os *streams* adaptativos e as redes de distribuição de conteúdos.

No Capítulo 3 são apresentados os fundamentos das redes centradas em conteúdos e suas funcionalidades. Serão discutidas suas principais funcionalidades, vantagens e

desvantagens.

No Capítulo 4 são discutidos os trabalhos relacionados do tema da proposta, principalmente no que diz respeito ao uso do controle de transmissão nas redes CCN, transmissões de vídeo em redes CCN e estudos sobre MPEG-DASH para CCN.

No Capítulo 5, é apresentada a proposta e sua arquitetura, assim como são detalhados os módulos para o controle e a transmissão adaptativa de vídeo em redes CCN. Também é apresentada a implementação da proposta, usando o simulador ndnSIM. Aspectos práticos e simplificações da implementação da proposta para análise de desempenho são explicados.

No Capítulo 6, são apresentados todos os cenários para emulação e simulação que foram utilizados nessa dissertação. Nesse sentido, são analisadas a transmissão de vídeo em CCN e a proposta de controle adaptativo de taxa de interesses. Os resultados obtidos são apresentados e discutidos.

Por fim, no Capítulo 7, são apresentadas as conclusões e as propostas para evolução futura da pesquisa.

Capítulo 2

Modelos de Comunicação IP, Streamings Adaptativos e Redes CDN

2.1 Modelos de Comunicação IP

Na televisão tradicional, todos os telespectadores recebem o mesmo sinal, ao mesmo tempo. Para que essa mesma condição seja mantida para a transmissão de vídeo sobre IP, a distribuição simultânea de um sinal de vídeo para vários usuários, deverá utilizar um mecanismo *multicast*. Praticamente, todas as transmissões de conteúdos de vídeo lineares utilizam *multicast*, incluindo os serviços de TV a cabo, TV por satélite, *Broadcast TV* Digital Terrestre e a grande maioria dos serviços de IPTV. No entanto, atualmente os espectadores estão assistindo aos conteúdos de vídeo que eles desejam, no momento que eles querem e em novos tipos de dispositivos. Neste cenário onde os fluxos de vídeo são personalizados, o *multicast* não é a regra e sim as comunicações em *unicast*. Para atender a essa nova gama de serviços, o uso de *streaming* de vídeo e seus diversos formatos estão se popularizando nas transmissões de vídeo.

2.1.1 Comunicações *Unicast* e *Multicast*

Para facilitar o entendimento sobre os conceitos básicos de *multicast*, é preciso primeiramente compará-lo com o processo de *unicast*.

Unicast em uma rede IP é a maneira tradicional pela qual os pacotes são enviados a partir de uma origem até um único destino final. Cada pacote IP é formatado com os endereços IP da origem e do destino antes de sua transmissão através da rede. Quando os mesmos dados precisam ser enviados para vários destinos distintos, diferentes pacotes

são criados contendo o mesmo endereço de origem e o endereço de cada destino.

Para o cenário de um serviço de vídeo, quando os usuários desejam ver conteúdos específicos, cada um deles deve fazer uma solicitação distinta. A fonte então deverá criar diferentes fluxos *unicast* contendo os endereços IPs de destino de cada um dos usuários. Esse controle individual de cada transmissão *unicast*, possibilita o uso de recursos personalizados, como pausar, retroceder e avançar a reprodução do vídeo.

Caso o número de telespectadores simultâneos aumente, a carga de processamento da fonte e o volume de dados trafegados na rede aumentarão, a fim de comportar o aumento dos fluxos contínuos de pacotes requisitados por cada espectador. Por exemplo, uma fonte de vídeo transmitindo um sinal de vídeo HD codificado em H.264 a uma taxa de 6 megabits por segundo (Mbps) para 20 usuários diferentes, necessita uma conexão de rede de pelo menos 120 Mbps. Caso, tenhamos 500 espectadores requisitando o mesmo vídeo, teremos uma carga de 3 Gbps na rede. Esse aumento no consumo de banda transmitida é um fator de grande preocupação para provedores de serviços de rede. Para mitigar esse problema, diferentes recursos e tecnologias de transmissão são empregados a fim de manter a possibilidade de personalização das transmissões de cada usuário, conforme será apresentado na seção 2.2, assim como nas próprias redes CCN como será mostrado no Capítulo 3.

Diferentemente do *unicast*, onde a responsabilidade da criação dos fluxos de vídeo para cada usuário é feita pela fonte, no *multicast* a responsabilidade pela replicação dos fluxos de vídeo para cada usuário passa a ser da rede. Protocolos permitem que os roteadores identifiquem os pacotes *multicast* e façam o seu envio para os vários destinos solicitantes. Os pacotes *multicast* possuem um endereçamento especial a fim de facilitar a sua identificação. As transmissões *multicast* são unidirecionais, ou seja, não existe nenhum mecanismo de coleta e envio de dados dos diferentes destinos para a origem (exceto algumas estatísticas de desempenho de rede, tais como a perda de pacotes).

Durante uma conexão *multicast*, todos os usuários recebem ao mesmo tempo o mesmo fluxo de vídeo. Assim, quando eles querem assistir a um programa, eles devem fazer uma requisição de conexão (realizar um *join*) ao fluxo *multicast* para que possam receber os dados da transmissão. Neste caso, os usuários têm a opção de conectar ou desconectar de um fluxo *multicast* que já esteja fluindo na rede, o que eles não podem é iniciar ou parar um fluxo com a transmissão iniciada.

Nas Figuras 2.1(a) e 2.1(b) podemos ver uma representação que ilustra a diferença conceitual entre uma transmissão *unicast* e uma transmissão *multicast*.

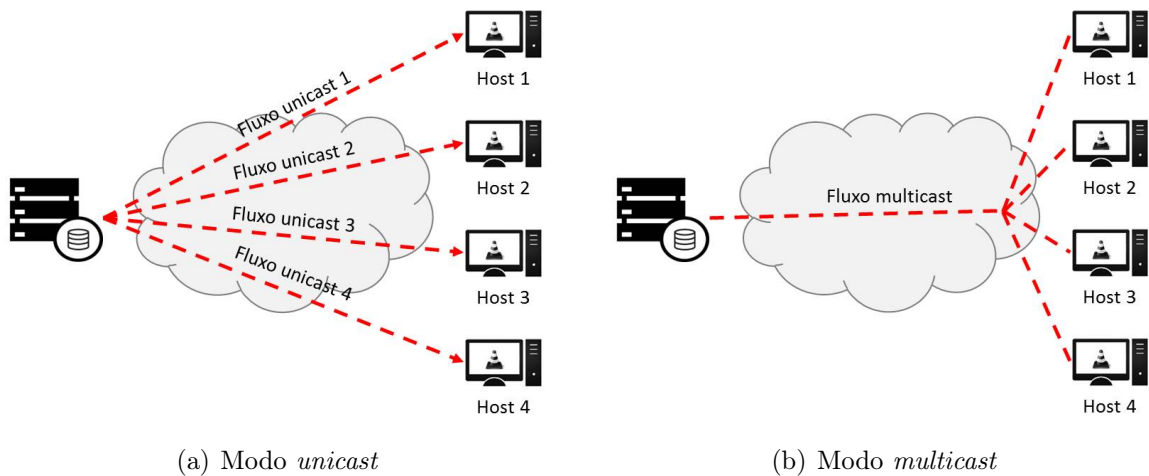


Figura 2.1: No modo *multicast*, apenas um fluxo de vídeo é trafegado na rede para os *hosts* que estão conectados a transmissão

A origem do *multicast* é responsável por anunciar periodicamente a disponibilidade de seu fluxo de dados para os dispositivos da rede. Os dispositivos interessados em realizar uma conexão *multicast* devem ouvir esses anúncios, que contêm detalhes sobre como o multicast está configurado. Sabendo destes detalhes, o dispositivo envia um pedido de *join* para o roteador mais próximo. Quando um roteador recebe um *join* de participação em um fluxo multicast, primeiramente ele deve determinar se o *multicast* requisitado já está sendo processado para outro usuário. Se for o caso, então o roteador simplesmente precisa fazer uma cópia do fluxo de *multicast* e enviá-lo para o usuário solicitante. Se não for o caso, o roteador deve solicitar o fluxo para um roteador que esteja transmitindo o fluxo *multicast*. Uma vez que esta solicitação é atendida, o roteador pode enviar o fluxo para o usuário. Observe que neste cenário, os pedidos são feitos a partir de roteador para roteador. Este procedimento pode ser repetido quantas vezes for necessário até encontrar uma rota desde o usuário até a fonte do *multicast*.

É importante observar que neste processo, a cada roteador, deve ser feito o acompanhamento do fluxo *multicast* que está sendo processado. Se ele já está recebendo um fluxo, então ele deve fazer uma cópia para os novos usuários que o solicitarem. Somente no caso em que ele não está recebendo o fluxo é que o roteador está autorizado a solicitar o fluxo de outro roteador. Esta é a grande vantagem do *multicast*, somente uma cópia do fluxo de *multicast* deve ser enviada para roteadores que têm usuários conectados ao fluxo *multicast*. Isto significa que a largura de banda entre os roteadores é utilizada de forma muito eficiente.

2.1.2 Comunicação por Streaming

Durante muitos anos, especialmente no final dos anos 90 e início dos anos 2000, o conceito difundido era o de que dadas as características de controle de transmissão existente, o TCP inseria muito jitter na transmissão e que seu uso para transmissões de vídeo não seria possível devido aos requisitos necessários. Em contra partida, devido à sua simplicidade, o UDP seria o protocolo mais indicado para utilização, mesmo tendo todos os problemas devido a falta de controle e garantia de entrega. Com o passar do tempo, e com o aumento e a popularização de diversos tipos de serviços de vídeo, os desenvolvedores de sistemas de transmissão de vídeo mudaram sua opinião em relação ao uso do TCP e o passaram a ver como uma vantagem em relação ao UDP, principalmente como forma de garantir a qualidade na transmissão de vídeo para conteúdos gravados tais como serviços de VoD ou sites como Youtube ou Netflix. Pois seus mecanismos de controle de transmissão, garantem uma entrega mais confiável do vídeo.

Para minimizar o jitter na utilização do TCP e problemas de taxas de transmissão variável, a solução adotada foi a utilização do conceito de buffers de exibição no lado cliente conforme ilustrado na Figura 2.2. Neste caso, todos os sistemas de *streaming* iniciam o armazenamento de 5 a 10 segundos de conteúdo antes da sua exibição e posteriormente utilizam o *buffer* para exibição. Esse atraso inicial na transmissão possibilita que o buffer seja minimamente preenchido até que o limite mínimo seja atingido. Quando a taxa utilizada pelo cliente for maior que a taxa de exibição do conteúdo, os quadros de vídeo serão armazenados automaticamente no *buffer* para sua posterior exibição. Este armazenamento no *buffer* ocorre automaticamente e possibilita que o *player* seja capaz de corrigir atrasos na entrega dos quadros de vídeo pela rede, ou devidos a congestionamentos. Caso o limite de armazenamento seja alcançado, o cliente informa ao servidor de mídia ou interrompe as suas requisições de forma a evitar um estouro no *buffer* e o consequente descarte de dados. Nas situações em que a taxa de recebimento de dados fica menor que a taxa de exibição por curtos intervalos de tempo, o *buffer* é consumido. Caso a taxa de exibição fique maior que a taxa de recebimento de dados por longos períodos de tempo, o *buffer* do cliente diminuirá e poderá esvaziar. Caso o buffer fique vazio, a exibição do conteúdo será interrompida até que o buffer seja novamente preenchido até atingir o mínimo para exibição. Quando a média da taxa de transmissão do TCP for o dobro da média da taxa de exibição, temos que a transmissão em TCP terá pequenos atrasos e uma baixa probabilidade de os *buffers* ficarem vazios.

A determinação do tamanho do *buffer* necessário ou a velocidade na qual o *buffer* será

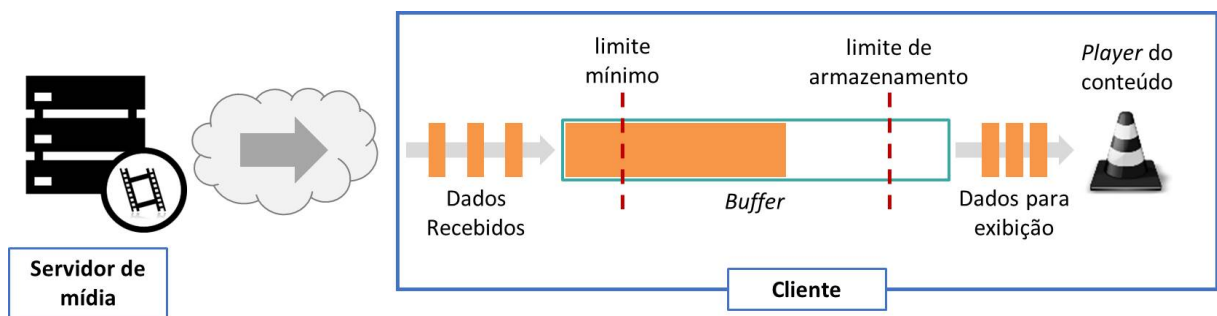


Figura 2.2: Ilustração do uso do *buffer* de exibição no dispositivo cliente

enchido, dependem diretamente do tipo de protocolo que será utilizado na comunicação.

No *Streaming* por UDP, o servidor transmite o vídeo a uma taxa equiparável à taxa na qual o cliente reproduz o vídeo e não emprega nenhum mecanismo de controle de congestionamento. Como a exibição do vídeo ocorrerá em um curto intervalo de tempo após o seu recebimento, o *buffer* necessário para o armazenamento dos pacotes de vídeo é pequeno. Ao manter a taxa de transmissão constante, o *streaming* em UDP fica suscetível a problemas de exibição decorrentes de perdas ou sobrecargas na rede que fazem com que a taxa de transmissão fique menor que a taxa necessária para a exibição. Neste caso, teremos interrupções ou erros na exibição do conteúdo podem ser são comuns.

Já as transmissões baseadas em TCP, apresentada nas seções 2.1.2.1 e seção 2.1.2.2, que utilizam o HTTP como protocolo de transmissão, teremos a princípio um aumento no *jitter* de transmissão devido aos mecanismos de controle do TCP. Para isso, será necessário um limite mínimo maior e um limite de armazenamento do *buffer* maior. Em contrapartida, o TCP enviará os dados na maior taxa possível de forma que se contemplem os atrasos inerentes ao protocolo, assim como o tempo necessário para o reenvio de qualquer pacote perdido.

Tanto *streamings* em UDP como em TCP, quando possuem taxas de download maiores que as taxas de exibição do conteúdo, garantem que o buffer será capaz de encher e reproduzir o conteúdo sem perdas. Neste cenário, como o enchimento do *buffer* será mais rápido que o seu consumo, pode acontecer de todo o arquivo do conteúdo ter seu download concluído muito antes do término de sua exibição. O que faz com que os dispositivos receptores tenham a necessidade de possuir um armazenamento grande o suficiente para conter todo o arquivo.

2.1.2.1 Streaming por HTTP

A convenção na década de 1990 foi que *streaming* de vídeo nunca iria funcionar bem sobre TCP. Ao longo do tempo, no entanto, os arquitetos de sistemas de *streaming* de vídeo aprenderam que os mecanismos existentes TCP para o controle de congestionamento e de transferência de dados confiável não necessariamente impossibilitavam a reprodução contínua de um vídeo. O uso de *buffers* e o armazenamento inicial do conteúdo antes de sua exibição viabilizaram o uso do TCP que por ser um protocolo mais robusto que o UDP, tornava a transmissão mais confiável e menos suscetível a perdas durante a transmissão.

No caso de um usuário desejar ver um vídeo armazenado em um servidor HTTP, o cliente estabelece uma conexão TCP com o servidor e emite uma solicitação HTTP GET para aquela URL. Em seguida o servidor envia o mais rapidamente possível o arquivo de vídeo dentro de uma mensagem de resposta HTTP. Os mecanismos de controle de congestionamento intrínseco do TCP como o início lento e a prevenção de congestionamento que respectivamente incrementam exponencialmente a janela de congestionamento e a reduzem pela metade quando um timeout é detectado, garantem que a entrega dos dados ocorra da maneira mais confiável. No entanto, estes mecanismos fazem com que a taxa de transmissão do servidor para o cliente varie significativamente, visto que a cada segmento recebido a taxa de transmissão é incrementada e quando um *timeout* é detectado, a mesma é reduzida pela metade. Além disso, os pacotes podem também ser atrasados devido ao mecanismo de retransmissão do TCP.

O uso de HTTP sobre TCP também permite que vídeos passem a trafegar mais facilmente por *firewalls* e NAT (*Network Address Translation*) (que muitas vezes são configurados para bloquear a maioria do tráfego UDP, mas permitem a maioria do tráfego HTTP).

2.1.2.2 Streaming por HTTP Adaptativo e o padrão MPEG DASH

O *streaming* por HTTP apresenta uma desvantagem: todos os clientes recebem o mesmo conteúdo codificado em uma mesma taxa, independente da largura de banda disponíveis para cada um deles. Neste caso, clientes que possuem uma alta largura de banda, receberão o vídeo desejado mais rapidamente e possivelmente, sua exibição será menos suscetível a interrupções, visto que o *buffer* será enchido a uma taxa maior que a taxa de exibição. No caso de clientes com uma largura de banda baixa, podem ocorrer situações em que a taxa de enchimento do buffer seja menor que a taxa de exibição do conteúdo, com isso

interrupções no vídeo serão notadas.

Para minimizar os impactos na exibição e possibilitar um melhor uso dos recursos de rede disponíveis, foi desenvolvido o mecanismo de *streaming* por HTTP adaptativo. Em um *streaming* por HTTP adaptativo, o mesmo vídeo é codificado em diferentes taxas de bits (correspondendo a diferentes perfis de qualidade). Cada cópia gerada do vídeo original é dividida em pequenos intervalos fixo de tempo. Com isso, um único vídeo será fragmentado em pequenos segmentos de vídeo e cada segmento será replicado nos diferentes perfis de qualidade. Todos os segmentos de vídeo gerados são indexados em um único arquivo, denominado arquivo de manifesto.

Como o conteúdo é replicado em diversas cópias correspondentes aos diferentes perfis de qualidade, a demanda por armazenamento será elevada à medida que outros conteúdos também sejam disponibilizados para transmissão. A fim de reduzir a demanda de armazenamento, foi desenvolvido o mecanismo de *live transcoding* [50] que utiliza o conteúdo original como referência e conforme as requisições do conteúdo nos diferentes perfis forem recebidas, este mecanismo codifica o conteúdo original nos novos perfis conforme a demanda. Esta dissertação abordará apenas experimentos em que todas cópias do conteúdo nos diferentes perfis são armazenadas previamente a transmissão.

Alguns padrões de *streaming* por HTTP adaptativo foram desenvolvidos tais como os padrões proprietários: Adobe HTTP Dynamic Streaming [1], Apple HTTP Live Streaming [5] e Microsoft Smooth Streaming [42], além do padrão aberto DASH (*Dynamic Adaptive Streaming over HTTP*). Por se tratar de um padrão aberto, esta dissertação abordará apenas o padrão DASH para seus experimentos.

A Figura 2.3 apresenta a estrutura em blocos do funcionamento do DASH, para o *streaming* entre um servidor HTTP e um cliente DASH. A representação do conteúdo no servidor é composta de duas partes: o bloco descritor de apresentação da mídia (*Media Presentation Description* – MPD) que representa o arquivo de manifesto do conteúdo disponível, seus diferentes perfis de qualidade, as respectivas URL associadas de cada segmento e outras informações complementares. A outra parte do MPD é compreendida pelos diversos segmentos de vídeo.

Para reproduzir o conteúdo, primeiramente o cliente DASH necessita obter o arquivo de manifesto através de uma requisição HTTP GET. Desta maneira, o cliente DASH é informado sobre os diversos perfis de qualidade existentes para o conteúdo, os tempos envolvidos para sua exibição, a resolução da imagem e a localização de cada segmento. Através do uso dessa informação, o cliente DASH seleciona o perfil de qualidade mais apro-

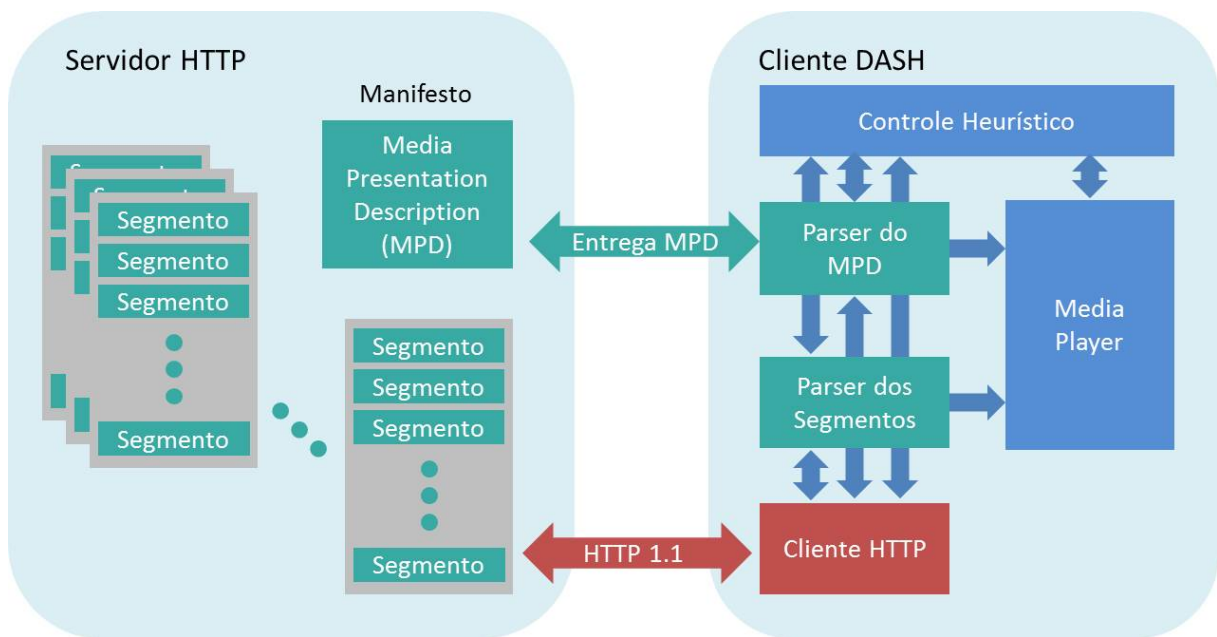


Figura 2.3: Estrutura em blocos do DASH

priado e inicia o *streaming* do conteúdo através de requisições HTTP GET, especificando a URL e um intervalo de bytes para cada segmentos de vídeo [4].

Enquanto recebe o download fracionado, o cliente analisa sinteticamente os segmentos e o manifesto, respectivamente pelo *Parser* dos segmentos e pelo *Parser* do manifesto, com isso os dados resultantes são disponibilizados para *media player*. O controlador heurístico é responsável pelo monitoramento dinâmico do nível do buffer e da largura de banda disponíveis. Seu algoritmo é capaz de determinar qual o próximo segmento de vídeo melhor atende as condições da rede e de uso do buffer.

Portanto o DASH permite que o cliente alterne livremente entre os diferentes níveis de qualidade do conteúdo, mantendo uma reprodução contínua e com o melhor nível de qualidade possível, evitando a ocorrência de congelamentos ou saltos na exibição. Qualidade visual da reprodução do conteúdo pode sofrer uma degradação quando ocorrer uma variação súbita da condição da rede, provocando uma variação significativa da taxa de um segmento para o outro. Essa percepção visual pode ser minimizada com o uso diversos perfis de qualidade intermediários, com isso o impacto visual da transição de um segmento para o outro será suavizado.

Por permitir que um cliente possa se adaptar à largura de banda disponível, o DASH permite que clientes com diferentes taxas de acesso à Internet possam receber vídeo em diferentes qualidades de imagens com uma reprodução satisfatória. Este recurso é parti-

cularmente importante para os usuários das redes móveis celulares, onde as condições da rede variam a medida em que se locomovem em relação às estações base. Por exemplo, clientes em conexões 3G de baixa capacidade podem receber uma versão de baixa qualidade de um conteúdo e clientes em conexões em 4G de alta capacidade pode receber uma versão de alta qualidade do mesmo conteúdo.

2.2 Redes de Distribuição de Conteúdo

O objetivo das redes de distribuição de conteúdo (*Content Distribution Networks* - CDN) é melhorar a entrega dos dados ao cliente de forma a aumentar qualidade e eficiência na transmissão. Com isso, diminuem-se os gargalos nas redes de acesso, reduz-se a latência no acesso dada a proximidade entre os sistemas finais envolvidos e consequentemente aumenta-se as taxas de transferência de dados.

Em uma rede CDN o conteúdo é replicado através de diversos nós interconectados pela Internet de forma que a entrega seja feita através do servidor da localidade que melhor pode atender a solicitação [46].

A arquitetura de uma CDN contempla dois blocos funcionais inter-relacionados, o serviço de redirecionamento de requisições e políticas de acesso e o bloco de distribuição e replicação de conteúdos. O bloco de redirecionamento de requisições e políticas de acesso é responsável pela forma como a CDN irá tratar as requisições de conteúdo, através da seleção do servidor mais adequado para atender à demanda. O serviço de distribuição e replicação de conteúdos é composto pela forma como os conteúdos serão replicados na rede de acordo com a definição do produtor do conteúdo. Os dois blocos devem ser integrados de modo que o bloco que trata do redirecionamento de requisições deve ter conhecimento de como o conteúdo requisitado é distribuído na rede. E o bloco de distribuição e replicação deve alocar e realocar os conteúdos de acordo com as demandas recebidas pelo bloco do redirecionamento de requisições.

Tipicamente existem três tipos de servidores em uma rede CDN: o(s) servidor(es) de controle responsáveis pelo recebimento e redirecionamento das requisições, os servidores de origem que fazem a requisição ao produtor do conteúdo e os disponibiliza na rede e os servidores de réplica e entrega que fazem a transmissão do conteúdo para o cliente final. Os servidores de réplica e entrega, ao receberem uma demanda de entrega de um conteúdo que não está presente no seu *cache*, fazem requisições internas à CDN de forma a buscar esse conteúdo de outros servidores de réplica ou então do servidor de origem.

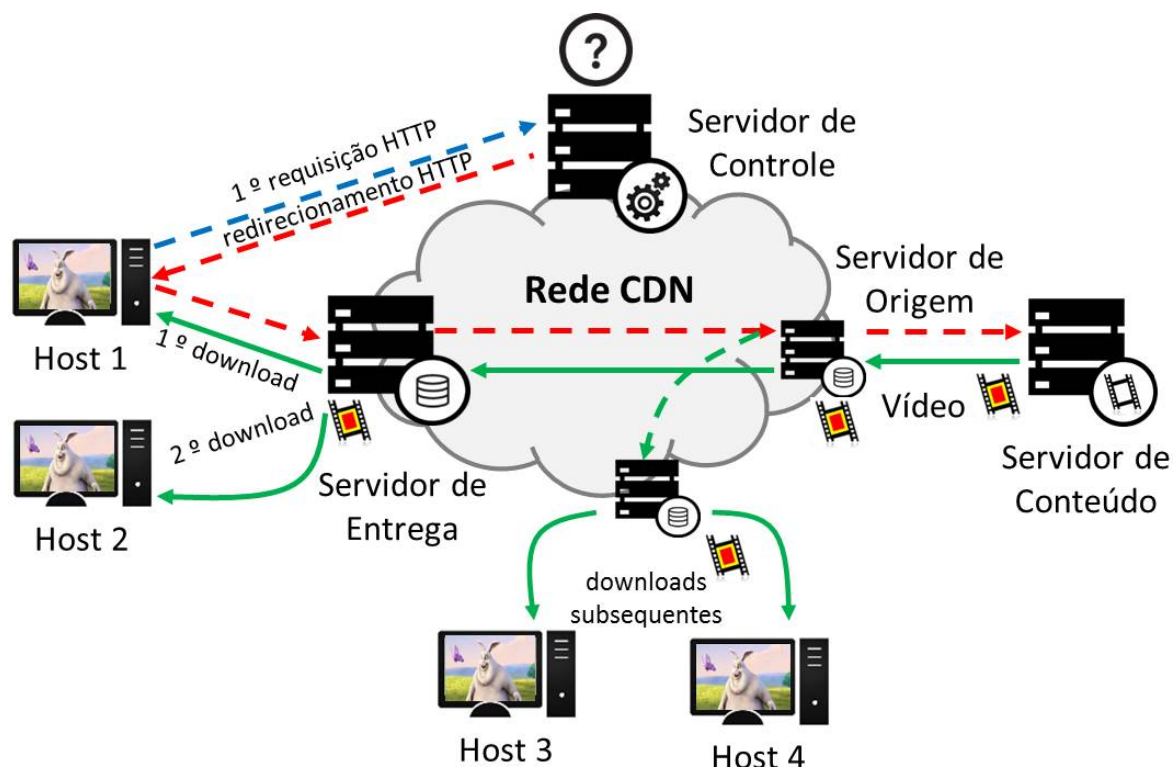


Figura 2.4: Exemplo de funcionamento da CDN ilustrando o processo de redirecionamento de requisições e entrega de conteúdo

A Figura 2.4 exemplifica o funcionamento de uma rede CDN, onde um cliente realiza a requisição de um conteúdo e o servidor de controle, responsável pelo recebimento e redirecionamento das requisições, analisa a requisição HTTP recebida, seleciona o melhor servidor de entrega de acordo com os critérios de avaliação definidos e realiza o redirecionamento da requisição HTTP.

O redirecionamento da requisição HTTP é feito através de configurações dinâmicas do DNS e do uso da mensagem HTTP 302 (Redirecionamento), sendo necessário que o dispositivo ou o aplicativo que tenha feito a solicitação seja capaz de entender e processar a mensagem. O redirecionamento HTTP é uma das técnicas mais simples utilização, pois todas as requisições de objetos web são feitas através de serviços web, seja um navegador de Internet, um aplicativo customizado ou um *player* de conteúdos.

No caso do conteúdo requisitado não estar disponível nos *caches* da CDN, os servidores de entrega solicitarão o conteúdo ao servidor de origem que solicitará ao servidor de conteúdo. Uma vez que o conteúdo esteja disponível na CDN, o mesmo será replicado entre os servidores de entrega conforme a demanda do conteúdo. Caso novas requisições do mesmo conteúdo sejam recebidas pelo servidor de controle, o conteúdo poderá ser

entregue diretamente pelo servidor de entrega selecionado.

Para o cliente, o uso da CDN é transparente, pois a maneira na qual o cache da CDN é utilizado, faz com que a única percepção para o assinante seja a melhora na sua transmissão. As redes CDN são utilizadas amplamente para a entrega de *streamings* de vídeo devido a sua baixa latência e a possibilidade de o cliente realizar downloads com taxas de transferências mais elevadas. Desta forma, devido aos requisitos que uma transmissão necessita para manter uma boa experiência de uso, a cada dia mais as redes CDN são mais empregadas.

2.3 Internet do Futuro

Manipular e distribuir informações é uma das principais aplicações da internet hoje em dia. Além da distribuição de conteúdo web, outras tecnologias de distribuição foram desenvolvidas tais como *peer-to-peer* e as redes CDN apresentadas anteriormente. Essas tecnologias promovem um modelo de comunicação que acessa os dados através do nome, independentemente da localização do servidor de origem. Recursos como *caching* e replicação são utilizados a fim de atender o aumento no tráfego atual na internet. Esses recursos são empregados independentemente (por exemplo entre diferentes provedores de CDN), desta forma não é possível identificar de maneira única a informação independentemente do meio de distribuição.

Além disso, a arquitetura estática das redes convencionais não condiz com o cenário atual de computação dinâmica em que aplicações geralmente acessam bancos de dados distribuídos geograficamente e em que servidores necessitam de flexibilidade na comunicação entre nuvens públicas e privadas.

De forma a atender a estas necessidades e não ficar restrito as limitações existentes na arquitetura atual da rede, novos modelos de comunicação e novas arquiteturas passaram a ser estudadas no tema que passou a ser conhecido como Internet do Futuro. Diversas arquiteturas e modelos estão sendo desenvolvidos, no entanto destaca-se a arquitetura SDN (*Software-Defined Networking*) [16] que caracteriza-se por ser dinâmica, gerenciada e adaptável e a arquitetura baseada de forma a identificar de maneira única a informação (*Information-Centric networking* - ICN) [18, 3].

A arquitetura SDN é caracterizada pela separação entre o controle da rede e as funções de encaminhamento. Com isso, o controle da rede pode ser diretamente programado e a infraestrutura subjacente pode ser abstraída nas aplicações e nos serviços de rede.

Neste caso, essa separação permite que os administradores da rede possam dinamicamente configurar, gerenciar e otimizar o tráfego da rede de acordo com as variações na demanda através da automatização de programas específicos para SDN. Para facilitar o gerenciamento, a inteligência da rede é logicamente centralizada em controladores SDN específicos. Estes controladores possuem uma visão global de toda a rede. A adoção comercial de SDN já está sendo iniciada por parte de diversos fabricantes de equipamentos de rede.

Na arquitetura ICN o dado deixa de ser dependente da localização e do armazenamento, possibilitando o *caching* e a replicação interna na rede. Os benefícios esperados são o aumento da eficiência, melhor escalabilidade na demanda por informação em relação ao consumo de banda e mais robustez em cenários de comunicação críticos. Os conceitos de ICN podem ser implementados sob a infraestrutura IP ou podem ser utilizados no nível de pacotes através da interconexão das tecnologias o que pode ocasionar mudanças no roteamento e encaminhamento existente para pacotes IP.

Espera-se que a arquitetura ICN evolua a arquitetura da internet em diferentes níveis e segmentos. Os conceitos de ICN podem ser identificados em diferentes novas arquiteturas de rede, sendo a principal delas as redes centradas em conteúdo (CCN). As redes CCN são o foco de estudo desta dissertação e será detalhada no capítulo 3.

Capítulo 3

Fundamentos de Redes CCN

3.1 A Arquitetura CCN

Primeiramente apresentada por Jacobson et al [26] como uma nova arquitetura de rede voltada para a distribuição eficiente e segura de conteúdo, as denominadas redes CCN (*Content Centric Network*) apresentam-se como uma mudança no paradigma atual de arquitetura de rede, de forma a torná-la mais adequada às demandas de serviço existentes. Ao defini-la toda em torno do nome do pacote e não mais sendo restrita ao conhecimento dos endereços de origem e destino, os nós da rede passam a realizar o encaminhamento entre os consumidores de conteúdo e os respectivos provedores de conteúdo em função do nome dos pacotes. A Figura 3.1 mostra uma comparação entre a pilha de protocolos de uma rede IP tradicional, comparada a uma rede CCN.

Um recurso nativo nas redes CCN é a realização do cache transparente do conteúdo à medida que ele está sendo trafegado na rede. Desta forma, nós passam a ter conhecimento do conteúdo por eles trafegado, o que não ocorre nativamente nas redes IP. Caso seja necessário e possível, os próprios roteadores respondem a requisições de conteúdos previamente armazenados em seus caches [55, 25, 49, 28].

A arquitetura CCN foi elaborada de acordo com cinco princípios fundamentais:

- **Independência da Localização:** O conteúdo pode ser entregue a partir de qualquer nó que o tenha disponível, independentemente da localização ou de qual nó seja este.
- **Invariância do Conteúdo:** Todos os conteúdos publicados na CCN são inalteráveis. Quaisquer alterações que venham a ser realizadas serão consideradas novas versões de objetos já existentes.

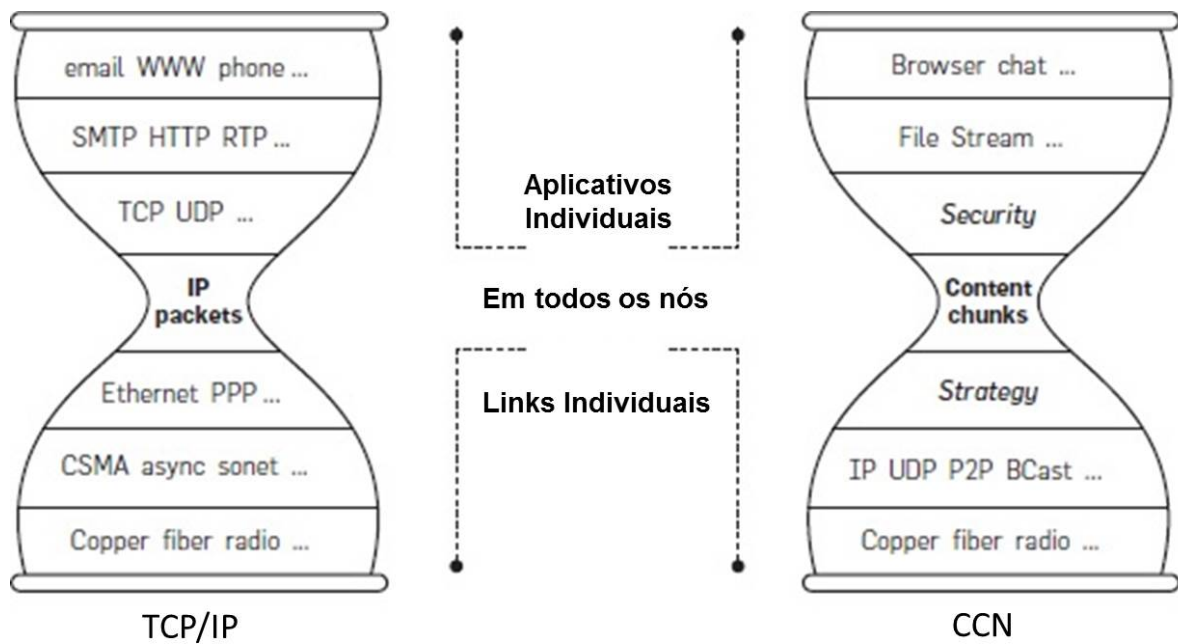


Figura 3.1: Comparação entre a pilha de protocolos de uma rede TCP/IP com uma CCN [26].

- Controle de Fluxo: Todo conteúdo somente é enviado caso exista uma requisição associada a ele. Essa requisição só pode ser atendida por, no máximo, uma resposta de conteúdo.
- Extensibilidade: Apenas os mecanismos que suportam distribuição eficiente e segura de conteúdo fazem parte da arquitetura CCN. Serviços de nível superior podem ser implementados em uma camada acima da CCN.
- Segurança: A integridade do conteúdo depende de assinatura criptográfica realizada pelo seu fornecedor e da verificação feita por parte do consumidor. Essa condição é chave para que a CCN possa receber um conteúdo de qualquer nó CCN, sem contato com a fonte do conteúdo. Garante-se a segurança do conteúdo propriamente dito e não da conexão ou topologia utilizada (o armazenamento e os enlaces utilizados pela rede não são considerados seguros) [17, 41, 51].

A CCN pode ser implantada sobre qualquer tecnologia de camada 2 de enlace de dados do modelo OSI, da mesma forma que acontece com o IP ou até mesmo sendo utilizada como uma camada superior ao IP [26, 8, 48]. A fim de garantir a interoperabilidade com as redes existentes, para a implantação inicial da CCN na Internet, todos os pacotes deverão se encapsulados no IP e com o uso de tunelamento para a ligação entre nós CCN remotos. Desta forma, a adesão das redes CCN é facilitada.

3.2 Fundamentos de Redes CCN

3.2.1 Os Pacotes CCN

Existem dois tipos de pacotes na CCN, os pacotes de interesse e os pacotes de dados. Quando um consumidor deseja um conteúdo, é feita uma requisição na rede através do envio de um pacote de interesse. Qualquer nó da rede que receba o interesse e consiga atendê-lo com o conteúdo requisitado, envia um pacote de dados de volta para o nó que enviou o pacote de interesse. O pacote de interesse e o pacote de dados terão o mesmo nome associado. Toda a comunicação na CCN baseia-se na troca de pacotes de interesse e dados associados, desde que ambos tenham o mesmo nome.

Os pacotes de interesse também podem ser utilizados para a realização do controle de fluxo da transmissão, uma vez que para o recebimento de um pacote de dados, é condicionada a existência de pelo menos um pacote de interesse associado.

Existe um tipo específico de pacote de interesse que representa os casos em que uma requisição de um conteúdo não pode ser atendida. Estes pacotes de interesse específicos são denominados pacotes NACK e devido ao motivo pelo o qual a requisição de conteúdo não pode ser atendida, os pacotes NACKs recebem uma classificação específica. Os casos em que foi identificada a presença de congestionamentos na rede acarretam na ocorrência de pacotes do tipo NACK CONGESTION. Os casos em que foi identificados loops nas requisições de interesses, isto é, quando um interesse é encaminhado através dos nós da rede até retornar ao seu nó de origem acarretam na geração de pacotes do tipo NACK LOOP e os casos em que a requisição de conteúdo não pode ser atendida devido ao fato de o conteúdo requisitado não ser encontrado na rede, recebem a classificação de NACK GIVEUP PIT. O pacote NACK será fundamental para o desenvolvimento do mecanismo de controle proposto no Capítulo 5.

Devido ao fato de as mensagens CCN poderem utilizar nomes relativamente grandes e terem o *overhead* devido ao uso da chave de identificação e do Hash, podem ocorrer casos em que uma mensagem possa ter um tamanho maior do que o MTU permitido em uma rede. Neste caso, mecanismos de fragmentação de pacotes são empregados a fim de reduzir o tamanho dos pacotes transmitidos. Desta maneira, é possível notar que muitas vezes, os dados são transmitidos em pedaços ou em *chunks*.

3.2.2 Nomeação dos conteúdos nas redes CCN

Os nomes na CCN servem a dois propósitos: a parte inicial do nome é utilizada como informação para o roteamento na rede através da localização do prefixo do nome e a parte final é utilizada para a identificação única do conteúdo [6, 12, 11].

O conteúdo possui mais duas identificações: a chave de identificação e o Hash do conteúdo. A chave de identificação é a chave de segurança utilizada por quem está publicando o conteúdo na CCN e devido a sua criptografia, a identificação de quem o disponibilizou é conhecida. O Hash do conteúdo é utilizado para a criptografia de todo o conteúdo. Os pacotes de interesse que possuem a chave de identificação do conteúdo desejado, só serão atendidos pelos pacotes de dados que possuem a mesma chave de identificação associada ao conteúdo. Desta forma, garante-se que somente serão recebidos pacotes de dados associados aos pacotes de interesse do conteúdo desejado.

Não existe a obrigatoriedade de os nomes serem legíveis. Caso seja necessário que o nome seja legível, atualmente utiliza-se a codificação UTF-8. Um nome CCN é semelhante a uma URI, onde segmento tem uma etiqueta e um valor. A etiqueta identifica o propósito do componente do nome, tal como um componente genérico utilizado para o encaminhamento, ou um componente especializado utilizado para sequenciamento ou *timestamps*. Há também rótulos específicos de aplicação. Por exemplo, o nome pode ser relativo a um conteúdo específico como: /exemplo.com/CCN/vídeos/filme.avi

A correspondência de nomes é realizada pela igualdade exata de cada segmento que contempla um nome. Por exemplo, para que ocorra a correspondência exata entre dois nomes, é necessário que todos os segmentos dos dois nomes sejam exatamente iguais entre si [29].

3.2.3 O funcionamento dos nós CCN

A operação básica de um nó CCN é semelhante à de um nó IP. Os nós CCN recebem e enviam pacotes através das faces. A face é um ponto de conexão para um aplicativo, ou outro nó CCN, ou algum outro tipo de canal. Uma face pode ter atributos que indicam a latência e largura de banda esperada, a capacidade de realizar *multicasts*, *broadcasts* ou outros recursos úteis.

Cada nó CCN possui três estruturas internas de dados: O repositório local CS (*Content Store*), a FIB (*Forwarding Information Base*) e a PIT (*Pending Interest Table*), como

mostrado na Figura 3.2.

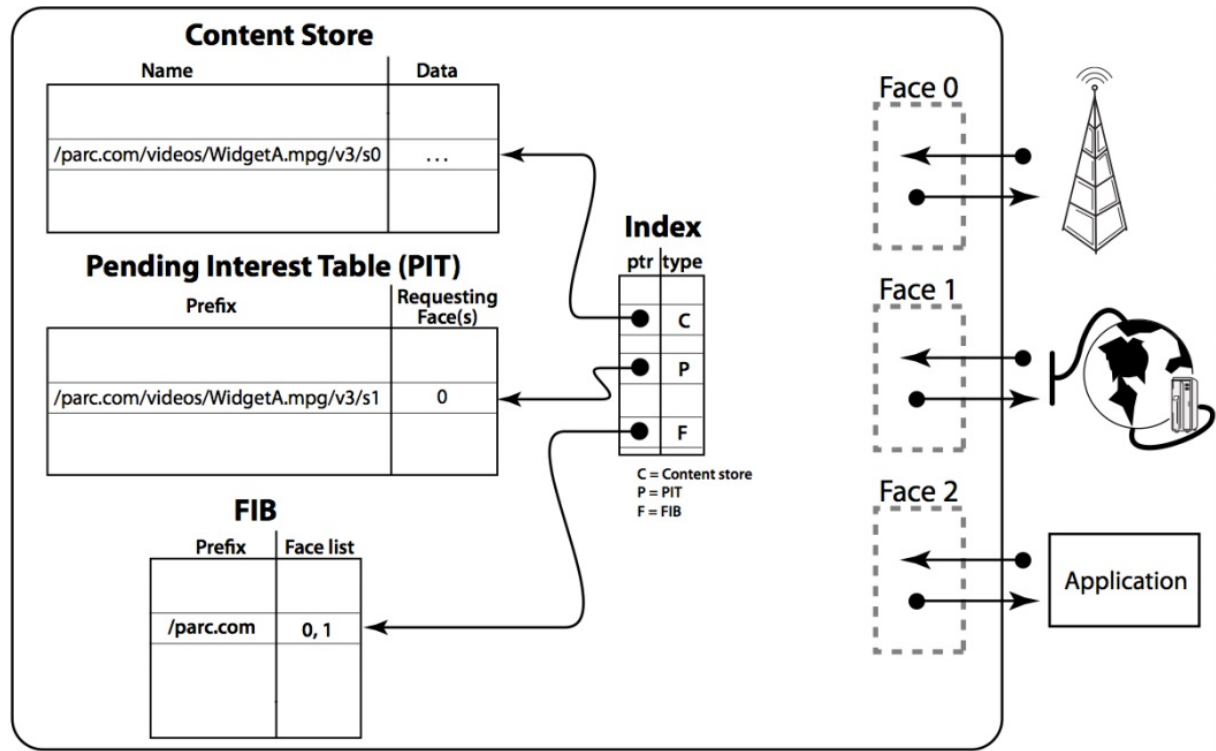


Figura 3.2: Tabelas usadas para gerência dos pacotes de interesse e dos conteúdos armazenados em um nó CCN [56].

O repositório local (CS) mantém uma tabela de conteúdos transmitidos ou em *cache*, indexados pelo nome do pacote. Além de proporcionar o *buffering* da comunicação, o CS serve como um cache de conteúdo. O repositório local tem um funcionamento análogo ao do *buffer* de memória de um roteador IP, com exceção da política de substituição diferente. Como cada pacote IP pertence a uma única conversa ponto-a-ponto, depois que ele for encaminhado, ele não tem mais valor. Ao contrário do IP, os pacotes CCN são auto identificáveis e auto autenticáveis e podem ser úteis para muitos usuários. Sendo assim, a fim de aperfeiçoar o uso da largura de banda e reduzir a latência percebida pelo usuário, os nós CCN podem armazenar os pacotes de conteúdo no seu CS.

Na CCN, apenas os pacotes de interesse são roteados nó-a-nó. Os pacotes de dados ao trafegarem, seguem o caminho inverso ao feito pelo pacote de interesse. O controle dos pacotes de interesse pendentes é feito pela PIT. Quando um pacote de interesse é encaminhado, a PIT armazena as informações da face pela qual o pedido foi recebido no nó e a face pela qual ele foi encaminhado. A PIT pode conter informações extras tais como o tempo limite no qual o pacote fica marcado como pendente. Neste caso, o nó informa a origem que o pedido não pode ser atendido. Pode haver informações adicionais,

tais como valores de tempo limite, que afetam as entradas.

A FIB é usada para o encaminhamento de pacotes de interesse para potenciais fontes de dados, de forma análoga à FIB em roteadores IP. A FIB detém um conjunto de entradas, onde cada entrada contém um prefixo de nome e uma lista das faces que podem fornecer dados para esse prefixo.

O procedimento realizado por um nó quando um pacote de interesse é recebido por uma de suas faces é:

- Verificar a existência de pacotes de interesse duplicados: caso pacotes de interesse com o mesmo nome sejam recebidos ou se o pacote de interesse atingiu o seu limite de saltos, o mesmo é descartado.
- Verificar a existência de dados: caso exista um conteúdo na CS, cujo nome corresponde exatamente ao nome recebido no pacote de interesse, é enviado em sequência um pacote de dados e o pacote de interesse é descartado.
- Verificar a existência de pacotes de interesses duplicados pendentes: caso o nome constando no pacote de interesse recebido corresponda exatamente ao nome de uma entrada da PIT, então a face pela qual o interesse foi recebido é adicionada à lista de faces e o interesse é descartado.
- Encaminhar o interesse para um potencial fornecedor: caso existam entradas na FIB com prefixos de nomes iguais ao informado no pacote de interesse, a face pela qual o interesse foi recebido é adicionada na PIT e, na sequência, o interesse é encaminhado para o potencial fornecedor pela a face informada na FIB.

O procedimento realizado por um nó quando um pacote de dados é recebido por uma de suas faces é:

- Descarte de respostas duplicadas: caso tenha uma entrada exata na CS, o pacote de dados é descartado.
- Descarte de respostas não solicitadas: caso não tenha uma entrada exata na PIT, o pacote de dados é descartado.
- Responder ao solicitante: existe uma entrada correspondente na PIT. Assim, a lista de faces é consultada de forma a encaminhar o dado pela face informada na PIT. Após o encaminhamento, a entrada na PIT é descartada.

- Armazenar na CS: Opcionalmente pode-se escolher armazenar o dado na CS. Esse procedimento não é especificado por padrão, mas define o quão eficiente será a rede ao prover o conteúdo rapidamente, sem precisar consultar a fonte [36, 37, 24].

O processo de encaminhamento dos pacotes em um nó CCN pode ser resumidamente representado na Figura 3.3, onde um nó ao aceitar um pacote de interesse por uma de suas faces, verifica se ele consegue atender ao interesse diretamente, caso o conteúdo solicitado esteja armazenado na sua CS. Caso contrário, a FIB é consultada e o interesse é encaminhado à face de saída indicada e a PIT é atualizada com a informação da face na qual o interesse foi recebido (Figura 3.3 (a) a (b)). Caso novos interesses do mesmo conteúdo sejam recebidos pela mesma face anterior, os mesmos serão descartados (Figura 3.3 (c)). Caso receba um novo pacote de interesse para o mesmo conteúdo por outra face, a PIT é atualizada com a número da nova face (Figura 3.3 (d)). Ao receber o pacote de dados pela face na qual os pacotes de interesse foram encaminhados, o mesmo é encaminhado as faces listadas na PIT e suas entradas são atualizadas (Figura 3.3 (e)) [37, 30, 21, 57]. Uma vez que o pacote de dados é recebido pelo nó, o conteúdo associado é armazenado na CS. Caso um novo pacote de interesse para este conteúdo seja recebido por uma de suas faces, o nó fará o envio do pacote de dados diretamente para a face, sem a necessidade de encaminhamento da requisição na rede (Figura 3.3 (f)).

Um nó CCN pode transmitir pacotes de interesse a um ou mais nós conectados. Normalmente, a melhor abordagem é transmitir o interesse em primeiro lugar para a face mais provável para responder, com base nas estatísticas das respostas anteriores. No entanto, por vezes, o encaminhamento de pacotes para outras faces, ou mesmo para todas as faces, é útil para detectar problemas ou para atualizar estatísticas associadas aos *links*.

Ao ter o controle exato por onde o pacote de interesse trafegou, o nó, ao receber um pacote de dados, verifica o nome associado e, na sequência, utiliza as informações presentes na FIB para encaminhar o pacote de dados pela interface na qual ele recebeu o pedido de interesse. Desta maneira, o pacote de dados realiza o percurso até o consumidor que realizou a requisição.

Uma vez que apenas os pacotes de interesse são reencaminhados (roteados) na CCN, os pacotes de dados devem tomar o caminho inverso do percorrido pelos pacotes de interesse. Os pacotes de dados podem ser armazenados em todos os nós CCN intermediários entre o consumidor do conteúdo e produtor do conteúdo. A política de cada nó e o tipo de conteúdo no pacote de dados serão determinantes na escolha do que armazenar. Normalmente, os dados na CS são removidos usando políticas de limpeza, como os re-

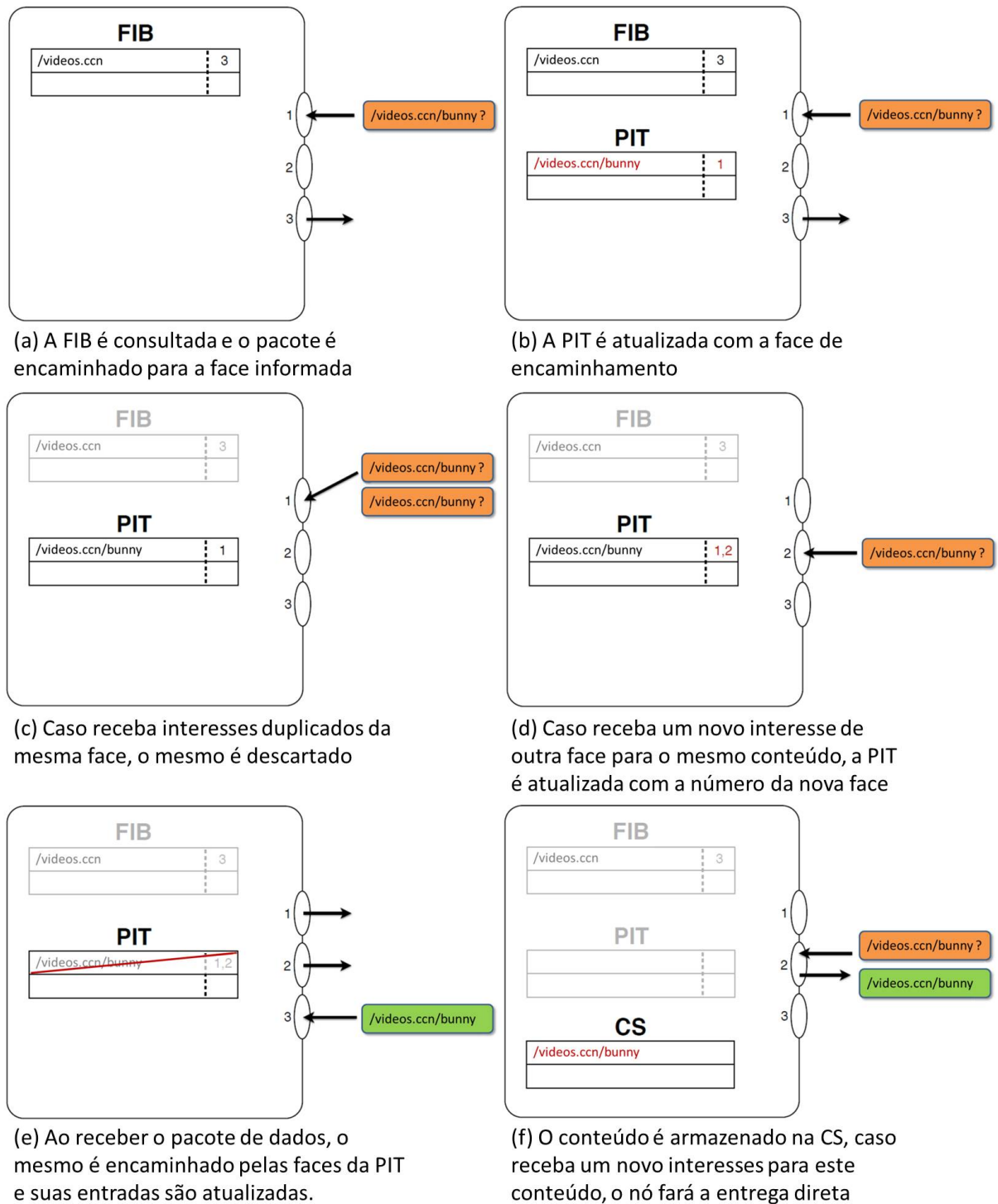


Figura 3.3: Representação do funcionamento de um nó CCN

centemente menos utilizado (LRU – *Least Recently Used*) ou os frequentemente menos utilizado (LFU – *Least Frequently Used*), embora outras políticas possam ser usadas também [44, 22, 31, 15]. Conteúdos populares tipicamente ficam na CS e raramente são substituídos, portanto interesses para conteúdos populares provavelmente serão servidos

a partir da CS do nó CCN mais próximo, reduzindo assim o uso de banda de rede.

O mecanismo de recuperação de conteúdo na CCN garante que os dados populares estarão tipicamente no caminho entre o consumidor do conteúdo e produtor do conteúdo, desde de que o armazenamento esteja em toda parte na rede. Isso é interessante, pois cria estruturas de distribuição para os conteúdos mais populares similares a árvores *multicast*, de tal forma que os conteúdos podem ser providos pelo nó mais próximo que contenha os conteúdos armazenados. Assim, o número de pedidos que chegam até o servidor do conteúdo é drasticamente reduzido, assim como a banda gasta com o envio do mesmo conteúdo por meio de diversas conexões *unicast*. Neste sentido, as CCN funcionam como uma generalização das CDNs, pois ao invés de armazenarem os principais conteúdos por servidores estrategicamente espalhados pela rede, na CCN, qualquer roteador pode armazenar o conteúdo, garantindo que os dados fiquem o mais próximo possível dos usuários finais.

3.2.4 Equilíbrio do Fluxo de Transportes

Como informado na Seção 3.1, uma CCN pode rodar sobre qualquer camada-2 ou tecnologia acima. O protocolo CCN requer muito pouca funcionalidade para o transporte dos pacotes. A CCN simplesmente assume a entrega de pacotes sem estado, não confiável, desordenadamente e por melhor esforço. Assim, os pacotes de interesse e/ou dados podem ser perdidos ou danificados em trânsito, ou o conteúdo solicitado pode não estar disponível. Na CCN, o caminho de retorno para o pacote de dados é o inverso do caminho para o pacote de Interesse. Contudo, nós intermediários da CCN podem falhar, resultando, assim, em pacotes de dados que não chegam ao consumidor.

Nesse caso, a confiabilidade na CCN é conseguida através da retransmissão de pacotes de interesses que ainda não tenham sido satisfeitos. A responsabilidade recai sobre o consumidor de conteúdo, que deve retransmitir pacotes de interesse não respondido. O consumidor pode incorporar um tempo de *timeout* para cada pacote de interesse que ele envia. Após o tempo de *timeout*, um pacote de interesse não respondido pode ser retransmitido. Outras lógicas podem ser, naturalmente, implementadas para otimizar o funcionamento do cliente.

A CCN mantém um equilíbrio no fluxo de transmissão, de modo que um pacote de interesse resulte em no máximo um pacote de dados. Como no TCP, não é necessário, ou mesmo desejável, aguardar uma resposta antes de enviar o próximo interesse, portanto vários interesses podem estar em transmissão simultaneamente. Com isso, a taxa de

envio de pacotes de interesse é equiparável a taxa de envio de pacotes de dados. Como o tamanho em bytes dos pacotes de dados é mais representativo que o tamanho dos pacotes de interesse, ao multiplicar o seu valor em bytes pela sua taxa de envio teremos aproximadamente a taxa de transmissão. É importante salientar que no caso de uma transmissão de vídeo, a taxa de bits para a reprodução do vídeo é definida no momento em que o vídeo é codificado e não tem relação com a taxa de envio dos pacotes CCN e da taxa de transmissão.

Ao manter o equilíbrio no fluxo de transmissão, permite-se a comunicação eficiente através de conexões de diferentes latências e larguras de banda e nós com recursos variados. O equilíbrio do fluxo em cada nó permite técnicas simples e eficazes para evitar o congestionamento na rede. Como será mostrado no Capítulo 5, esta é a base da proposta desta dissertação de um mecanismo para a realização do controle da taxa de transmissão dos pacotes de interesse a fim de evitar congestionamentos.

Capítulo 4

Trabalhos Relacionados

Sendo a principal motivação desta dissertação a viabilização de uma técnica de controle adaptativo de taxa de pacotes de interesse para streaming de vídeo em redes centradas em conteúdo, foram analisados diversos trabalhos que apresentam estudo sobre transmissões de vídeo em CCN, incluindo transmissões ao vivo e de conteúdos gravados assim como o suporte a *streamings*.

Em [33] e [32], Kulinski e Burke apresentam propostas de aplicação para a realização de *streamings* de vídeo ao vivo e gravado para múltiplos clientes, neste trabalho é possível verificar que o conceito proposto para as redes CCN possibilitam a transmissão de vídeos. Os autores analisam também a habilidade da rede para sincronizar os caches para a exibição entre os clientes. Em [13], Ciancaglini et al apresentam uma proposta de *streaming* em tempo real, similar a um serviço de TV, onde a eficiência do controle da transmissão é feita através do controle do tempo de *timeout*. O controle do tempo de *timeout* foi utilizado como base também em [40], Longzhe et al apresentam um mecanismo de retransmissão para redes sem fio com perdas também baseado no controle do tempo de *timeout*.

Como apresentado nos Capítulo 2 e no Capítulo 3 as características dos *streamings* adaptativos e das redes CCN são bastante similares, o que torna o estudo de transmissões de *streamings* de vídeo adaptativo em redes CCN um assunto em voga em diversos estudos. Em [52], Xu et al apresentam um comparativo entre as vantagens e as desvantagens do uso de *streamings* em tempo real nas transmissões em CCN e em HTTP. Já em [38], Yaning et al apresentam uma proposta de aplicação para a realização de transmissões de *streamings* adaptativos em DASH com foco na análise do *overhead* de transmissão. Lederer et al em [34], Bing et al em [23] e Detti et al em [14] apresentam propostas de uso do DASH para CCN em dispositivos móveis.

Lederer et al, em [35] e em [39], também apresentam um estudo experimental comparativo onde o *overhead* na transmissão criado com a adição do empacotamento CCN em uma integração do DASH com a CCN é avaliado e seu resultado é analisado em relação ao *overhead* do HTTP. As diversas similaridades entre o DASH e a CCN são apresentadas, tais como a condição necessária para que as requisições sejam iniciadas e controladas pelo cliente, a maneira na qual o conteúdo é fragmentado na rede e a analogia entre a estrutura de nomeação e indexação dos conteúdos DASH e CCN. A Figura 4.1 apresenta essas similaridades, onde as diferentes versões de um conteúdo e a sua estrutura de nomeação, podem ser comparados aos diferentes segmentos de vídeo existentes do DASH e que são indexados pelo arquivo de manifesto. O estudo também apresenta o suporte a potenciais fontes múltiplas de um conteúdo e a maneira eficiente de replicação e distribuição dos fragmentos dos conteúdos dentro da rede. Desta maneira, verifica-se que o cliente pode requisitar qualquer fragmento do conteúdo de qualquer lugar da rede.

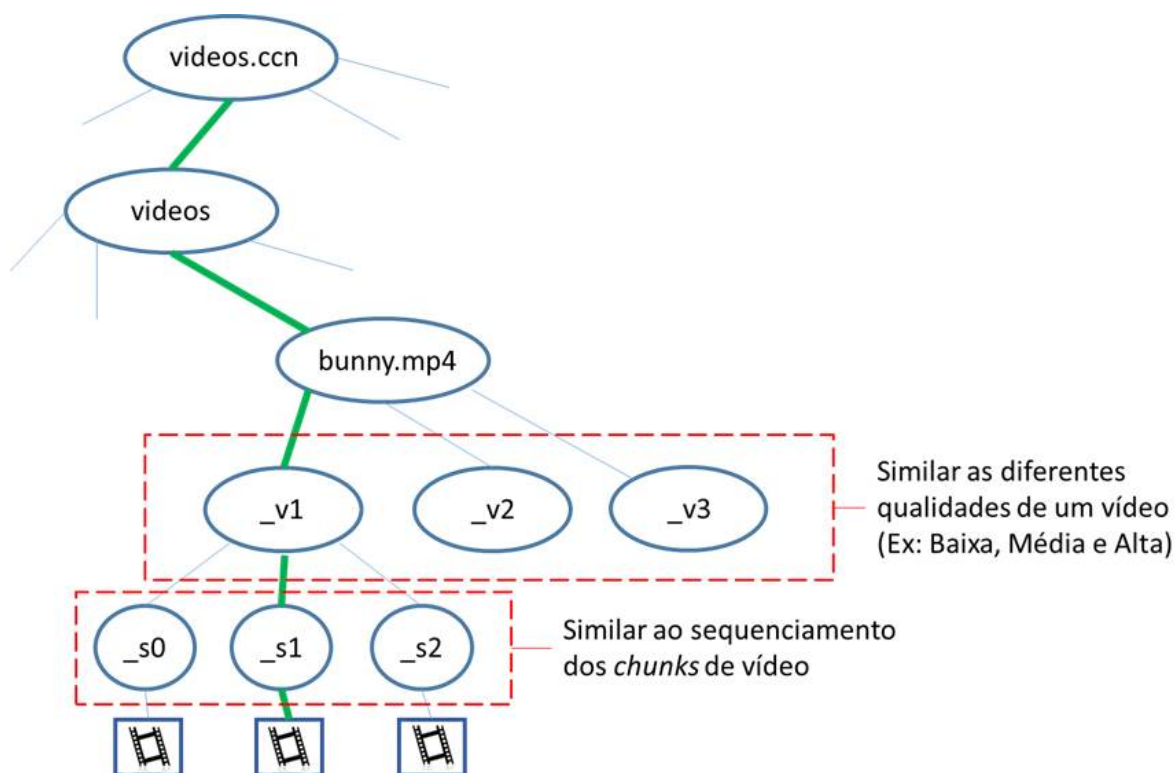


Figura 4.1: Representação das similaridades na nomeação dos conteúdos no DASH e na CCN

Em seguida, é apresentado um estudo de desempenho na transmissão de *streaming* adaptativo em DASH em um rede CCN, levando-se em consideração na análise a variação do estado da rede. Para este experimento, foi apresentada uma proposta de integração do DASH com a CCN conforme apresentado na Figura 4.2 onde a entrega do arquivo de

manifesto é feita através da rede CCN e um controlador de conexões CCN adicionado ao cliente DASH será responsável pela requisição e controle das conexões CCN para o recebimento dos segmentos de vídeo através da rede CCN. Durante o desenvolvimento desta Dissertação, este experimento foi utilizado inicialmente para análise e desenvolvimento do mecanismo de controle adaptativo da taxa de envio pacotes de interesse, sendo portanto replicado no ambiente de simulação conforme apresentado no Capítulo 6. No entanto, os resultados encontrados e a análise dos pacotes trafegados na rede apresentaram que o mecanismo proposto era fundamentado no uso de em requisições HTTP, adequadas de maneira a serem similares a uma transmissão CCN, o que descaracteriza seu uso para obter o objetivo proposto.

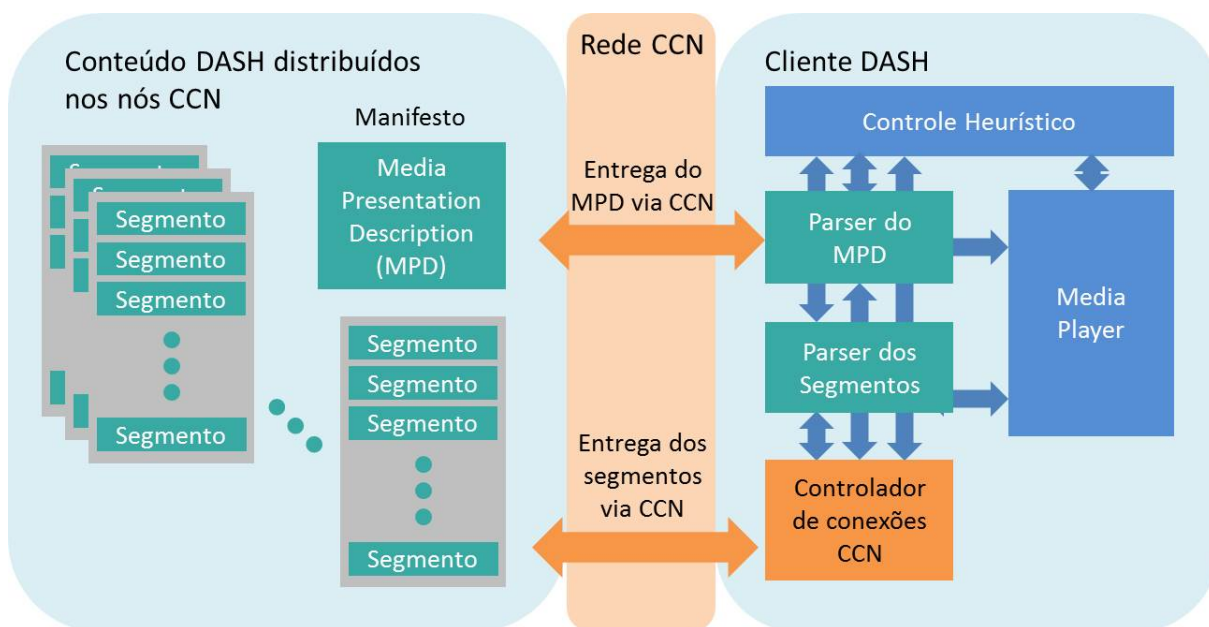


Figura 4.2: Proposta de integração do DASH com CCN

Sendo assim, buscando viabilizar uma técnica de controle adaptativa de envio dos pacotes de interesse, similar ao mecanismo do DASH, foi proposto um mecanismo novo de controle da taxa de envio de interesses. O controle proposto é feito com base nos recebimentos de NACK, em vez de ser baseado exclusivamente em *timeouts*. Com isso, é possível obter uma resposta mais rápida ao estado da rede.

O estudo [53] apresentado por Yi et al ilustra o processo de encaminhamento de interesses e define um diagrama de estado de cada face de forma que o conceito de encaminhamento adaptativo seja baseado nas informações de desempenho das entradas constantes na tabela de encaminhamento (FIB). Além das informações de encaminhamento, a FIB possui informações relativas ao estado de cada face. O estudo também apresentou os

conceitos relativos aos tipos de pacotes NACK existentes, assim como a proposta para a realização da estratégia de encaminhamento baseada nos pacotes NACK que serviram de modelo para o desenvolvimento do mecanismo de controle proposto.

Um outro ponto relativo a transmissão de vídeo em redes CCN é a análise da característica nativa que a rede possui de realizar o *cache* interno. Em [19], Grandl et al apresentam o relacionamento entre a utilização do cache na rede e os impactos na adaptação da taxa de transmissão, assim como o sucesso de requisitar os conteúdos armazenados em cache.

Como os mecanismos de controle existentes atualmente no TCP devem coexistir com as redes CCN, em [8], Braun et al apresentam alguns modelos de coexistência entre a CCN e os mecanismos de controle do TCP. Os autores apresentam um cenário de como os estudos de [54], [45], [27] e [9] se correlacionam com a estratégia de aumento aditivo e decremento multiplicativo para a dinâmica do controle de fluxo, mesmo em um ambiente nativo CCN, ou seja, que não utiliza o TCP ou o UDP. Esses estudos foram utilizados como base para a elaboração do mecanismo de controle proposto no Capítulo 5 e nos experimentos realizados no Capítulo 6.

Feixiong et al também apresentam um mecanismo de aumento aditivo e decremento multiplicativo para o controle explícito de congestionamento [56]. Porém, nessa proposta, o controle da taxa de envio dos pacotes de interesse é feito diretamente pelo roteador e não controlado independentemente pelo cliente conforme o mecanismo proposto Capítulo 5. Outros estudos [47] e [10] também foram realizados baseados no aumento aditivo e decremento multiplicativo de forma evitar congestionamentos.

Capítulo 5

Proposta

Nesta dissertação, propõe-se um mecanismo de controle adaptativo da taxa de envio de pacotes de interesse em redes CCN. Como a CCN possui a condição fundamental de que para todo pacote de dados é necessária a existência de um pacote de interesse associado, e como os pacotes de dados possuem um tamanho maior que os pacotes de interesse, ao controlar o envio de interesses, conseqüentemente, se controla o recebimento de dados. Dessa maneira, todo o controle é feito pelo cliente e não pelo servidor, possibilitando assim um consumo mais eficiente dos recursos da rede. Para que este controle ocorra, o cliente deverá possuir maneiras de verificar o status de rede e adequar a taxa de envio dos pacotes de interesse de acordo com o mecanismo proposto.

Dada a característica de fragmentação dos pacotes de dados e o uso de indexação para a nomeação dos conteúdos, a natureza de uma transmissão CCN assemelha-se a uma transmissão em *streaming* adaptativo, de forma que em ambos os casos, as requisições de dados são iniciadas pelo cliente. Como não existe um controle da taxa de envio de interesses implementado, é proposto neste trabalho um mecanismo de controle adaptativo de taxa de pacotes de interesse para a transmissão de vídeo em CCN que se assemelhe ao MPEG DASH.

O mecanismo proposto foi desenvolvido utilizando como fundamento o controle dos pacotes de dados e dos pacotes NACKs recebidos pelo cliente, de forma a adaptar a frequência de envio dos pacotes de interesse.

As vantagens do mecanismo proposto são tanto para o funcionamento da rede quanto para o cliente. Fazendo a requisição de pacotes de vídeo sem o *feedback* da rede, pode-se sobrecarregar a rede, prejudicando outras transmissões que também estão em curso. Por outro lado, ao se adaptar a taxa do vídeo de acordo com a característica da rede, o ganho

é evidente para o cliente, pois esse não observará tantas interrupções no seu *streaming*. O mecanismo proposto adapta a taxa do vídeo de acordo com a quantidade de NACKs que são recebidos, mudando o nome do conteúdo a ser baixado e a frequência da taxa de interesses conforme a taxa de vídeo selecionada.

A inovação da proposta reside justamente na criação de um mecanismo de controle da taxa de transmissão semelhante ao controle presente MPEG DASH para as redes CCN. CCNs foram desenvolvidas de forma a dar suporte a busca e distribuição de conteúdo independentemente de onde esse conteúdo está armazenado. Contudo, essa rede de nova geração, que pode trazer muitas vantagens em termos de eficiência na transmissão, ainda carece de propostas e implementações de controle de taxa que sejam eficientes, especialmente dentro do contexto de *streaming* de vídeo.

5.1 Arquitetura da Proposta

A arquitetura foi dividida em quatro módulos de acordo com o papel desempenhado para a realização do controle da taxa de interesses: o módulo de monitoração da rede, o módulo do mecanismo de controle, módulo de adequação das requisições e módulo escalonador, conforme apresentado na Figura 5.1:

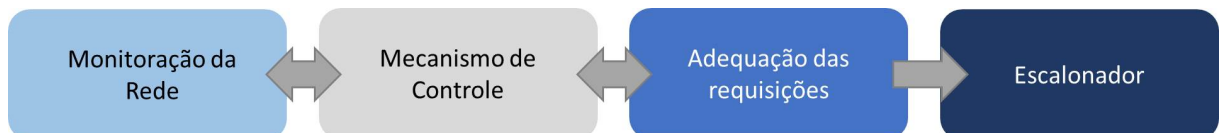


Figura 5.1: Diagrama em blocos dos módulos que compõem a arquitetura da proposta

O módulo de monitoração da rede é responsável por informar ao módulo do mecanismo de controle o estado atual da rede de forma a poder avaliar o melhor modo de realizar as requisições, ou seja, se a taxa de envio dos interesses deve aumentar ou diminuir, ou dependendo do caso, realizar a requisições de novos conteúdos. Após a avaliação do mecanismo de controle, o módulo de adequação das requisições poderá ser acionado e atuará conforme solicitado. Na sequência o escalonador será acionado para agendamento do envio dos pacotes.

5.1.1 Módulo de Monitoração da Rede

O módulo de monitoração da rede é responsável pela verificação do estado da rede. A verificação se dá através da coleta de um conjunto de informações relativas às transmissões realizadas e aos pacotes trafegados na rede.

Similar ao que acontece nas redes IP, o RTT (*Round Trip Time*) também é medido nas redes CCN através da medição da variação de tempo necessário para que um pacote de dados seja recebido, desde o momento em que o seu respectivo pacote de interesse foi enviado. Caso o tempo de RTT seja maior que um limite pré-estabelecido, ocorre um *timeout* na transmissão sendo necessário o reenvio do pacote de interesse. Desta maneira o módulo de monitoração é encarregado de informar ao mecanismo de controle que algum evento ocorreu na rede. Este evento pode estar atrelado a uma gama de problemas, porém muitos deles são informados pela própria rede CCN através dos pacotes NACK.

O controle dos pacotes de dados recebidos possibilita ao controlador alterar a taxa de requisição dos pacotes de interesse ou dependendo do caso, realizar requisições de outros repositórios.

Os pacotes NACK são classificados de acordo com três tipos de ocorrências: NACK CONGESTION, NACK LOOP e o NACK GIVEUP PIT que representam respectivamente os casos em que foi identificada a ocorrência de congestionamentos na rede, os casos em que um pacote de interesse entra em *loop* na rede, isto é, quando um interesse é encaminhado através dos nós da rede até retornar ao seu nó de origem e por fim o tipo de NACK quando o interesse não pôde ser atendido. De acordo com o tipo de pacote NACK recebido, o mecanismo de controle pode realizar uma determinada ação.

5.1.2 Módulo do Mecanismo de Controle

O mecanismo de controle é a parte central da arquitetura, sendo responsável pela decisão de controle em função das informações recebidas pela monitoração da rede. Sua atuação ocorre na variação da frequência de envio dos pacotes de interesse ou na mudança do nome requisitado. Sua funcionalidade pode variar também de acordo com o serviço ou aplicação que utiliza a rede CCN.

Sua lógica de funcionamento é ilustrada de acordo com o apresentado na Figura 5.2.

Ao receber as informações coletadas pelo módulo de monitoração da rede, o mecanismo de controle realiza diferentes avaliações de acordo com tipo de informação recebida. Ao

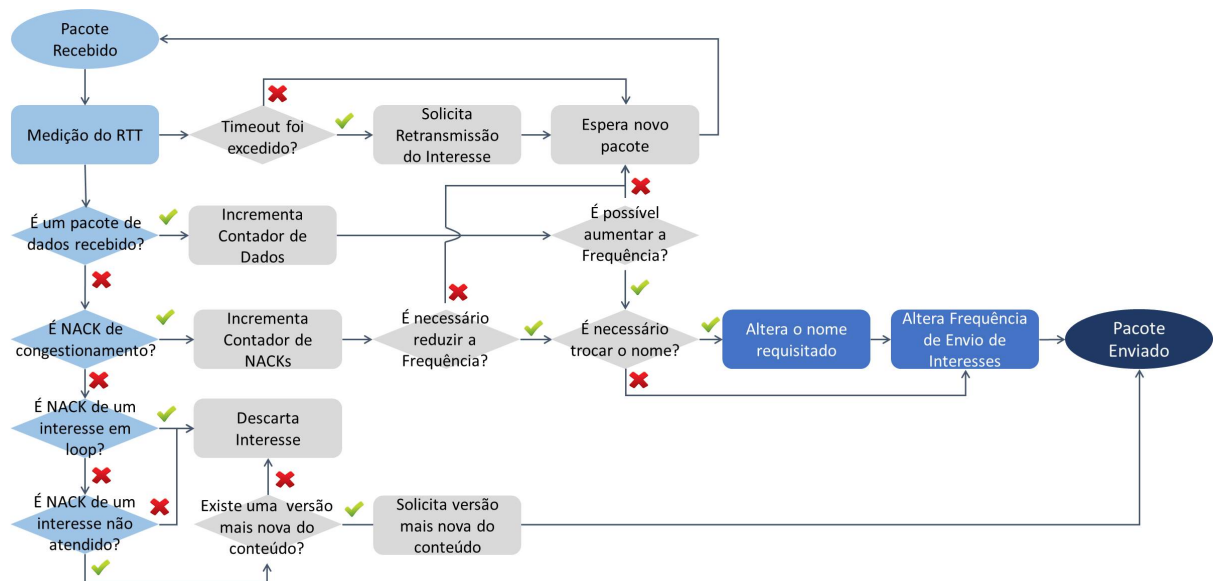


Figura 5.2: Lógica detalhada do funcionamento da arquitetura proposta

receber a medição do tempo de RTT, verifica-se se o tempo de *timeout* foi excedido ou não, caso tenha ocorrido um *timeout*, faz-se a solicitação de retransmissão e o novo pacote de interesse é agendado para envio. Caso contrário, aguarda-se o recebimento de um novo pacote.

Ao receber pacotes de dados, o contador de recebimento de pacotes de dados recebidos é incrementado e posteriormente verifica-se se é possível o aumento da frequência de envio dos pacotes de interesse. Caso seja possível esse aumento, verifica-se se existe a necessidade de troca do nome do conteúdo desejado. Se for necessária a troca do nome, o mesmo é feito e a frequência é aumentada. Se não for necessária a troca do nome, apenas a frequência de envio é alterada. Caso não seja possível o aumento da frequência de envio, faz-se o agendamento do próximo pacote de interesse.

Caso seja recebido um pacote NACK de congestionamento, o contador interno de NACKs é incrementado e em seguida verifica-se se é necessário reduzir a frequência de envio dos pacotes de interesse. Caso seja necessária essa redução, é verificada também a necessidade de alteração no nome do conteúdo solicitado. Se também for necessário trocar o nome, a frequência de envio e o nome são alterados. Caso não seja necessário trocar o nome, apenas a frequência de envio é alterada. Se não for necessário reduzir a frequência de envio, espera-se o recebimento do próximo pacote de interesse.

Caso seja recebido um pacote NACK devido a um interesses em *loop*, o interesse em questão é descartado e se o pacote NACK recebido for devido ao caso de um interesse

não atendido, verifica-se se existe uma versão mais recente do conteúdo. Caso exista uma nova versão do conteúdo, faz-se uma nova requisição para o conteúdo mais novo. Caso não exista uma nova versão do conteúdo, o interesse é descartado.

5.1.3 Módulo de Adequação das Requisições e Agendamento de Envio

O módulo de adequação das requisições e agendamento de envio é responsável por receber as tarefas designadas pelo mecanismo de controle e realizar as adequações das requisições quando for o caso e o agendamento do envio dos pacotes de interesse.

As adequações das requisições de interesse podem contemplar o aumento ou redução da frequência na qual os pacotes de interesses são enviados, o que impacta diretamente na transmissão de interesses. O agendamento do envio dos pacotes é a tarefa responsável pela ordenação da fila de envio para a face de saída.

De acordo com o tipo de pacote recebido, a taxa de envio interesse poderá ser atualizada, como detalhado na Figura 5.2. Os Algoritmos 1 e 2 mostram as propostas de controle da taxa deste trabalho, dentre as quais, no Algoritmo 1 considera-se as premissas do aumento aditivo e de decremento subtrativo e no Algoritmo 2 considera-se as premissas do TCP, de aumento aditivo e de decremento multiplicativo.

O aumento da frequência de envio dos pacotes de interesse ocorre enquanto o contador de pacotes de dados recebidos for igual ou maior que o contador de pacotes NACKs recebidos, ou seja, enquanto o cliente está recebendo mais pacotes de dados do que pacotes NACK. Quando o contador de pacotes NACK recebido for maior que o contador de pacotes de dados recebidos, o módulo de controle irá verificar se o contador de pacotes NACK está maior que o limite de pacotes NACK que podem ser recebidos. Caso esse valor seja menor, a frequência de envio dos pacotes de interesse não é alterada. Se o contador de pacotes NACKs recebidos for maior que o limite de NACKs recebidos, a frequência de envio dos pacotes de interesse é diminuída. Foi adotado na proposta o limite de 500 pacotes NACKs recebidos de forma a eliminar as variações iniciais na rede.

Dependendo do serviço ou aplicação que utilizará a rede CCN, as tarefas de alteração do nome requisitado podem representar um ganho de funcionalidade para o mecanismo de controle, uma vez que, a alteração do nome pode ser feita sem a necessidade de intervenção do serviço ou aplicação.

Algoritmo 1 Mecanismo de aumento e decremento aditivo do controle da taxa de envio de interesses

Entradas: *Frequência Inicial, Índice de Aumento, Índice de Redução da Frequência, Contador de Dados, Contador de NACK*

Saídas: *Frequência de Transmissão*

Zerar todos dos contadores

Frequência de Transmissão = Frequência Inicial

se *Contador de Dados $\geq$ Contador de NACKs* **então**

Frequência de Transmissão = Frequência de Transmissão + Índice de Aumento

senão

se *Contador de NACK > Limite de NACKs Recebidos* **então**

Frequência de Transmissão = Frequência de Transmissão - Índice de Redução

fim se

fim se

Algoritmo 2 Mecanismo de controle da taxa de envio de interesses

Entradas: *Frequência Inicial, Índice de Aumento, Índice de Redução, Contador de Dados, Contador de NACK*

Saídas: *Frequência de Transmissão*

Zerar todos dos contadores

Frequência de Transmissão = Frequência Inicial

se *Contador de Dados $\geq$ Contador de NACKs* **então**

Frequência de Transmissão = Frequência de Transmissão + Índice de Aumento

senão

se *Contador de NACK > Limite de NACKs Recebidos* **então**

Frequência de Transmissão = Frequência de Transmissão - Índice de Redução

fim se

fim se

5.1.4 Escalonador

O módulo Escalonador é praticamente um módulo independente dos eventos da rede, sendo responsável pelo agendamento do envio de pacotes de maneira periódica e constante. O módulo é chamado de acordo com a taxa atual de interesses para envio do conteúdo que for explicitado. Os eventos da rede só podem mudar o período de chamada escalonador. Uma vez que o módulo for chamado, o pacote será enviado.

5.2 Implementação da proposta

A proposta para controle adaptativo de *streaming* foi desenvolvida utilizando o artifício da simulação, a qual imita uma implementação em um sistema real [7]. Este conceito não deve ser confundido com o artifício da emulação que possibilita um hardware ou software de um *hosts* se comportar de maneira similar a outro *host* [20]. Nesta proposta foi utilizado o simulador ndnSIM com o objetivo de avaliar o impacto dos parâmetros e a eficiência do sistema de controle proposto.

Para a implementação da proposta, como o objetivo do trabalho é propor um mecanismo adaptativo da taxa de envio dos pacotes de interesse para o controle de *streamings* de vídeo, as informações relativas à medição do RTT e os NACKs devido a pacotes de interesse em *loop* e os casos de pacotes de interesse não atendidos não são considerados na implementação, uma vez que essas condições não impactam na alteração da frequência de envio dos pacotes de interesse. Neste cenário, o mecanismo de controle realiza sua avaliação considerando as informações relativas aos pacotes de dados recebidos e aos NACKs devidos a congestionamentos. A Figura 5.3 apresenta o detalhamento dos componentes que foram desenvolvidos no simulador para o mecanismo de controle.

Para a realização dos experimentos, será utilizado o simulador ndnSIM [2] na versão 1.0 devido a sua escalabilidade em relação à quantidade de parâmetros a serem configurados e também pela facilidade na alteração do código fonte.

Embora seja baseado em CCN, a NDN (*Named Data Networking*) é um projeto de Internet do futuro que até 2013 utilizava como código fonte o CCNx, posteriormente o desenvolvimento da NDN foi individualizado em relação ao CCNx, porém seus conceitos e funcionalidades são similares aos da CCN, sendo portanto utilizado ao longo da análise da proposta. Os módulos de monitoração, adequação das requisições e escalonador, da Figura 5.1, foram implementados integralmente.

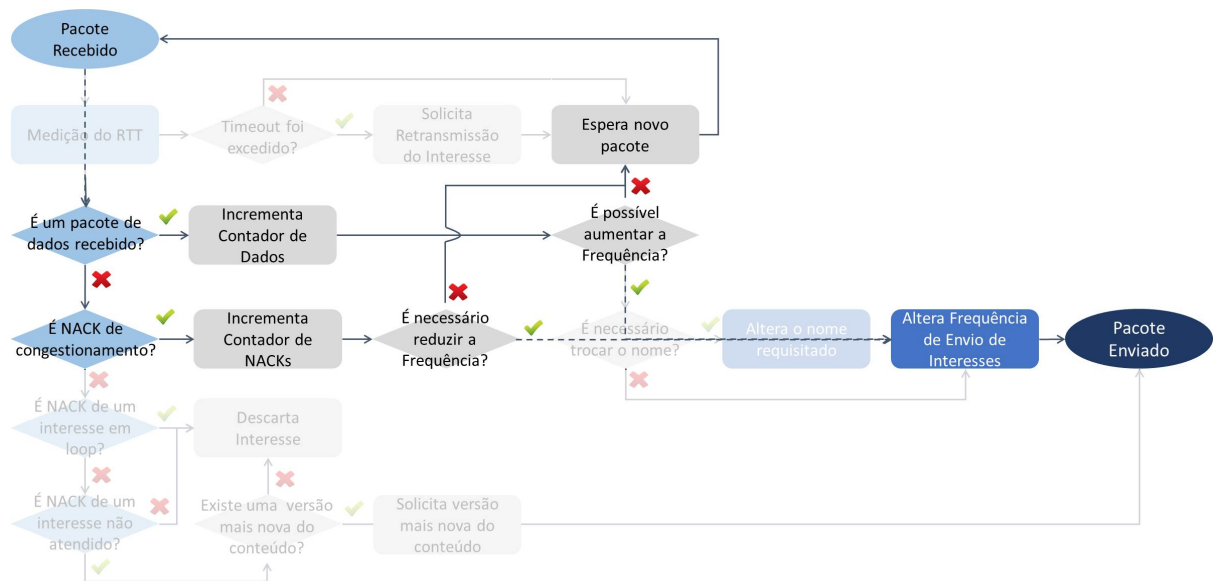


Figura 5.3: Detalhamento da implementação utilizada no mecanismo de controle

5.3 O Simulador ndnSIM

O ndnSIM é um simulador de rede baseado no NS-3 e de estrutura modular conforme apresentado na Figura 5.4. Através do uso de classes C++ individuais é possível modelar o comportamento de cada entidade de rede na CCN: PIT – Tabela de Interesses Pendentes, a FIB – Base de Informações de Encaminhamento, CS - Repositório Local, interfaces de rede e aplicação e estratégias de encaminhamento de interesses, fazendo com que essa estrutura modular possibilite que qualquer componente possa ser facilmente modificado ou substituído sem nenhum ou com mínimo impacto nos outros componentes.

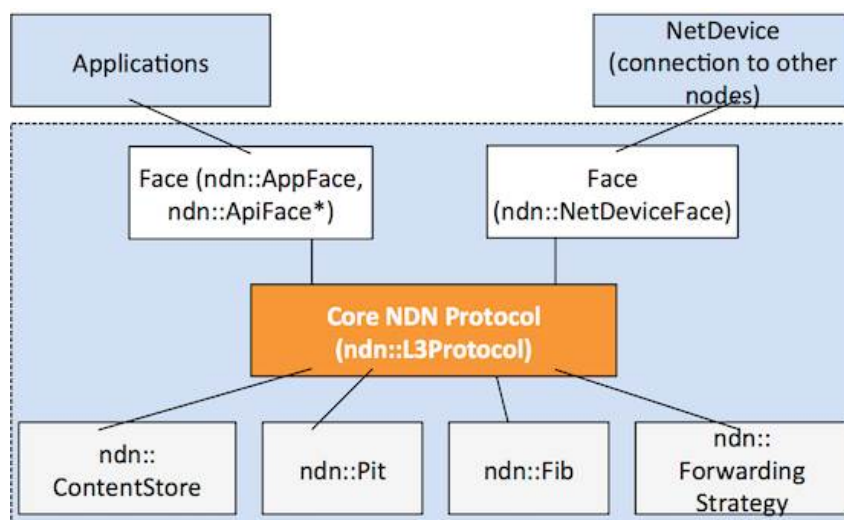


Figura 5.4: Componentes do ndnSIM - <http://ndnsim.net/>

Para o desenvolvimento do mecanismo de controle, foram feitas alterações nas classes `ns3::ndn::Consumer` e `ns3::ndn::ConsumerCbr`, pertencentes ao módulo de aplicações conforme indicado na Figura 5.5.

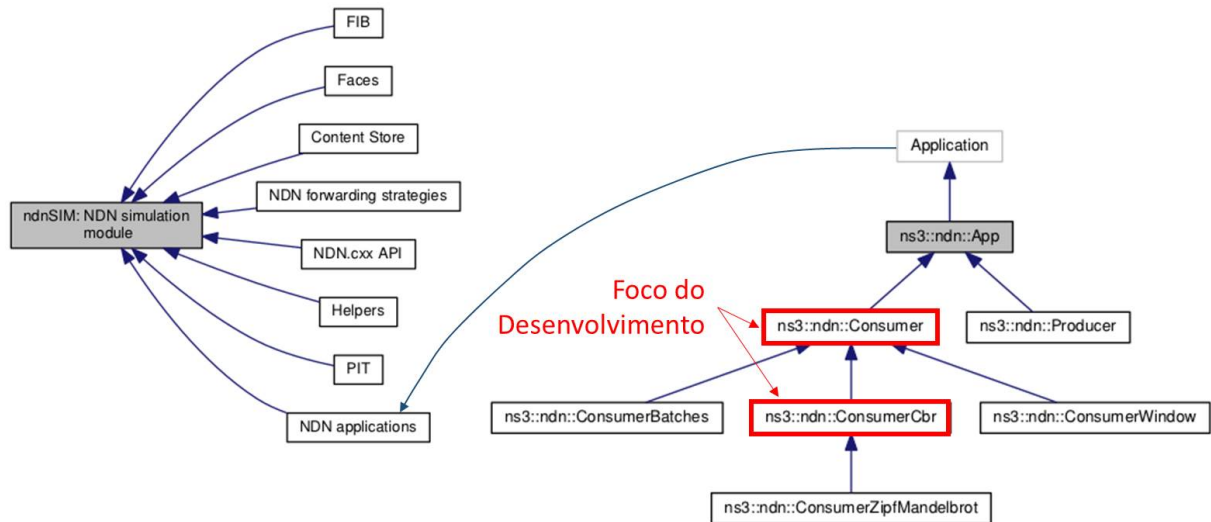


Figura 5.5: Módulos do Simulador ndnSIM e as classes modificadas.

A classe `ns3::ndn::Consumer` possui algumas funções internas responsáveis pelo processamento dos pacotes recebidos, tais como as funções `OnData`, que é chamada toda vez que um pacote de dados é recebido e a função `OnNack`, que é chamada toda vez que um NACK é recebido. Além de funções para o envio de pacotes como a `SendPacket` e funções para reenvio de pacotes.

Conforme apresentado no algoritmo do mecanismo de controle, na Figura 5.2, para que verifique se é necessário ou não aumentar ou reduzir a frequência de envio de pacotes de interesse, é preciso que se faça o controle dos contadores dos pacotes de dados e dos pacotes NACK. No desenvolvimento realizado, as funções `OnData`, `OnNack` e `SendPacket` foram alteradas de forma a implementar um contador de pacotes indicando o número sequencial de cada pacote.

A classe `ns3::ndn::ConsumerCbr` é responsável pelo agendamento do envio dos pacotes de acordo com a frequência inicial. O código da função `ScheduleNextPacket` foi alterado, tendo o valor da frequência inicial adaptado de acordo com os contadores de pacotes presentes nas funções `OnData`, `OnNack` e `SendPacket` e os parâmetros de aumento e diminuição da taxa de envio dos interesses. Os Algoritmos 1 e 2 apresentam as lógicas utilizadas pelo mecanismo de controle da taxa de interesses na função `ScheduleNextPacket`.

Os resultados obtidos nas simulações serão apresentados em detalhes na Seção 6.2.

Capítulo 6

Análise da transmissão de vídeo em redes CCN

A análise da transmissão de vídeo em redes CCN foi dividida em duas etapas. Primeiramente, buscou-se analisar o seu funcionamento em redes CCN através do uso de um cenário simulado com máquinas virtuais. Ainda nesse contexto, também foram realizados testes com a proposta de Ledere et al. [35], que primeiramente utiliza o MPEG-DASH em CCN e em sequência o teste com o uso de um *plugin* de CCN para o VLC e outro *plugin* para MPEG DASH. Na segunda etapa da análise, buscou-se avaliar a efetividade do mecanismo proposto para controle de taxa de vídeo adaptativo. Para tanto, utilizou-se o simulador ndnSIM, utilizando a proposta descrita na Seção 5.2.

6.1 Etapa 1 – Análise da transmissão de vídeo em redes CCN

Com o objetivo de verificar a qualidade da experiência da transmissão de um vídeo para um usuário em redes CCN, foi utilizado um cenário de simulação com um cliente realizando uma requisição de um vídeo armazenado em um repositório local de outro nó da rede CCN. A simulação foi realizada através do uso do simulador CCNx na versão 0.8.2, além de quatro máquinas virtuais no Oracle Virtualbox na versão para Windows 4.2.16 r86992, sendo todas as VMs com a mesma configuração virtual de hardware e com 4 GB de RAM. O sistema operacional utilizado foi o Ubuntu 14.04 LTS. As quatro máquinas foram configuradas de acordo com a topologia apresentada na Figura 6.1 de forma similar ao experimento apresentado em [35]. Todo o endereçamento IP e as configurações de rotas foram definidos manualmente conforme o apresentado na topologia. Mesmo o

experimento sendo em CCN, houve a necessidade de a conectividade IP estar totalmente estabelecida para o seu correto funcionamento.

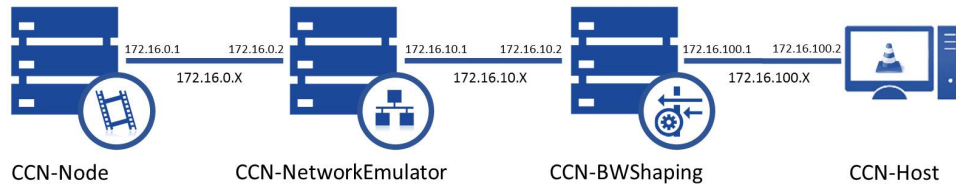


Figura 6.1: Topologia para o teste de desempenho do DASH para CCN

Para que os nós tenham conhecimento dos demais nós da rede e com isso realizarem o correto encaminhamento dos pacotes de interesse e de dados, foram configuradas manualmente as entradas da tabela de encaminhamento (FIB) em cada um dos nós. As entradas contemplaram tanto rotas baseadas em TCP como em UDP. Os resultados para cada um dos casos são analisados na sequência. Foram analisados três tipos de transmissão:

- Transmissão usando o *plugin* para MPEG-DASH em CCN descrito em [35];
- Transmissão usando o VLC com *plugins* para MPEG-DASH e CCN;
- Transmissão de streaming com o VLC com o *plugin* para CCN.

Assim, foram configuradas duas máquinas virtuais: uma máquina virtual para a publicação do conteúdo no repositório local (máquina CCN-Node) e outra como uma máquina hospedeira que utiliza como aplicação cliente o VLC para Ubuntu em duas versões: uma com *plugin* customizado para DASH em CCN e outra com outro *plugin* customizado apenas para CCN. Em virtude dos *plugins* não funcionarem na versão corrente do VLC, foi necessária a compilação e utilização de uma versão de desenvolvedor anterior do VLC. Foi também configurada uma máquina virtual usando o *plugin* de MPEG-DASH para CCN descrita em [35].

Em ambas as topologias de simulação há duas máquinas virtuais que desempenham o mesmo comportamento: simular a rede através da configuração de atrasos (*delay*), além de perda de pacotes (máquina CCN-NetworkEmulator) e também a largura de banda do enlace (máquina CCN-BWShaping).

O conteúdo utilizado para a análise da transmissão foi um trecho do vídeo *opensource* Big Buck Bunny. Foi utilizado um trecho do vídeo de 44 segundos com os parâmetros de áudio, vídeo e codificação apresentados na Figura 6.2. Os parâmetros foram identificados utilizando-se o software MediaInfo v0.7.63 presente no K-Lite Mega Codec Pack.

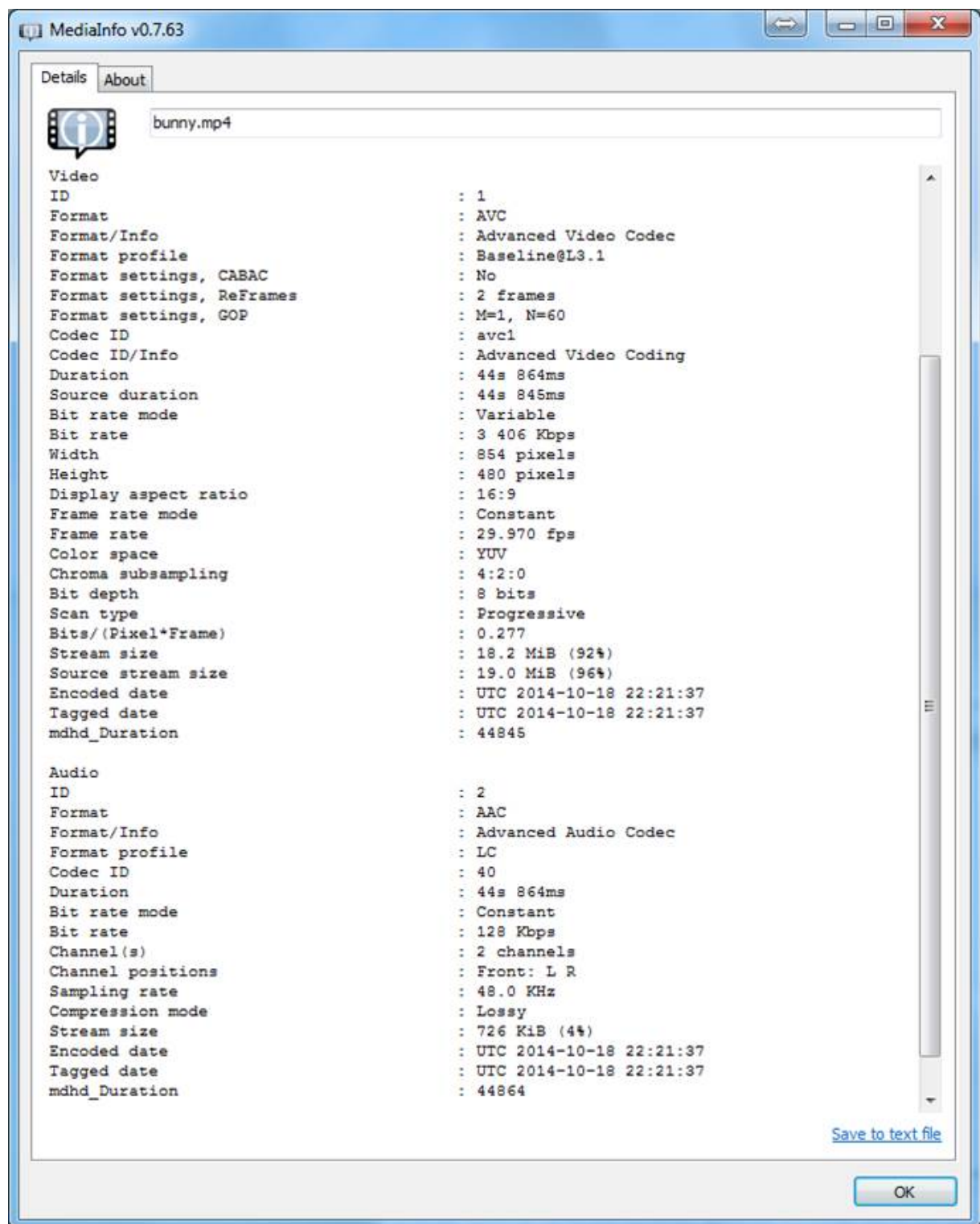


Figura 6.2: Parâmetros do Vídeo de Teste

Para a realização do processo de *encoding*, foi criada uma nova máquina virtual, igual as outras VMs (*Virtual Machines*), com exceção de que o sistema operacional utilizado foi o Ubuntu na versão 12.04 LTS. O software para a formatação do conteúdo DASH

Tabela 6.1: Cenários de testes de simulação de transmissão de vídeo para CCN.

Cenários de Testes de Simulação Modelo I		BW Shapping			
		BW Variável	BW 2000kbps	BW 3000kbps	BW 4000kbps
NetworkEmulator	Delay 0ms Loss 0%	Cenário A1	Cenário A2	Cenário A3	Cenário A4
	Delay 50ms Loss 0,5%	Cenário B1	Cenário B2	Cenário B3	Cenário B4
	Delay 100ms Loss 1%	Cenário C1	Cenário C2	Cenário C3	Cenário C4
	Delay 200ms Loss 2%	Cenário D1	Cenário D2	Cenário D3	Cenário D4

utilizado foi o DashCast.

Para a análise dos pacotes recebidos e enviados, foi utilizada a versão 1.8.6 do Wireshark para desenvolvedores, devido a necessidade de instalação de um *plugin* específico para análise dos pacotes CCN.

Em virtude da característica de armazenamento (*caching*) nos nós da rede CCN, somente as primeiras transmissões realizadas contemplariam a transmissão fim a fim, pois a repetição da transmissão teriam as requisições subsequentes atendidas pelo repositório local do próprio nó do cliente. Para evitar que diferentes rodadas de teste sofressem interferência das rodadas anteriores, antes de cada rodada, todas as VMs tinham seu estado (*snapshot*) reiniciados para o momento anterior a transmissão. Dessa forma, os repositórios locais estavam sempre vazios no início da simulação.

Para as simulações, foram definidos cenários de testes variando os parâmetros de atraso (*delay*) dentre 0 ms (sem atrasos), 50 ms, 100 ms e 200 ms e perdas (*loss*) dentre 0% (sem perdas), 0,5%, 1% e 2% no nó emulador da rede (máquina CCN-NetworkEmulator) e também a largura de banda dentre 2000 kbps, 3000 kbps, 4000 kbps e banda variável (sem limitação de banda) no nó que controla a largura de banda dos enlaces (máquina: CCN-BWShaping). O controle do *buffer* de reprodução e a definição do seu tamanho, foram feitos diretamente pela aplicação e não houve qualquer alteração na sua configuração original. Os cenários foram organizados de acordo com as variações dos parâmetros informados. Para simplificar o entendimento, na Tabela 6.1 é apresentado um quadro com o descritivo de cada cenário.

Para a medição do resultado, os seguintes indicadores foram empregados com o objetivo de mensurar a qualidade da reprodução no âmbito do usuário: percentual de perda de vídeo, o atraso no vídeo e o tempo de *freezing*. Os indicadores foram calculados de acordo com as informações do vídeo coletadas na tela de estatísticas de reprodução do VLC, conforme mostra a Figura 6.3, tais como quantidade de quadros de vídeo decodificados, quantidade de quadros de vídeo exibidos e quantidade de quadros de vídeo perdidos. As informações de áudio, devido à baixa taxa de transmissão, não foram consideradas na análise dos resultados.

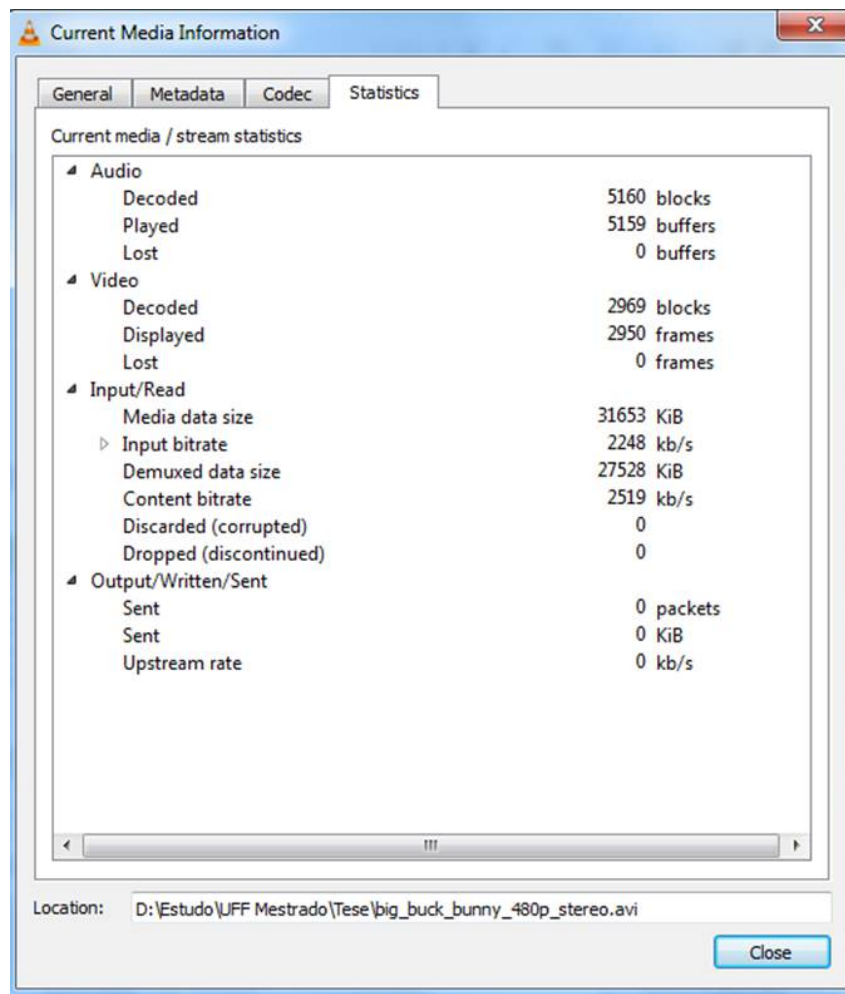


Figura 6.3: Estatísticas de reprodução padrão do VLC.

O percentual de perda de vídeo é definido como:

$$Perda\ de\ Video\ (\%) = \frac{Quantidade\ de\ Quadros\ de\ Vdeo\ Perdidos}{Quantidade\ de\ Quadros\ de\ Vdeo\ Decodificados}, \quad (6.1)$$

onde o valor da quantidade de quadros de vídeo decodificados indica o total de quadros

recebidos pela aplicação e o valor da quantidade de quadros de vídeo decodificados o total de quadros perdidos. Caso a quantidade de quadros de vídeo exibidos seja maior que a quantidade de quadros de vídeo decodificados, define-se o percentual de atraso no vídeo como:

$$\text{Atraso no Video (\%)} = \frac{\text{Quantidade de Quadros de Vdeo Exibidos}}{\text{Quantidade de Quadros de Vdeo Decodificados}} - 1, \quad (6.2)$$

onde a quantidade de quadros de vídeo exibidos indica o total de quadros exibidos. Como durante um congelamento na imagem, mantêm-se a exibição do mesmo quadro de vídeo, temos a quantidade de quadros de vídeo exibidos maiores do que a quantidade de quadros de vídeo decodificados. Sendo assim, para que ocorram atrasos na exibição é necessário que ocorram congelamentos. Logo, se a quantidade de quadros de vídeo exibidos for menor ou igual ao valor da quantidade de quadros de vídeo decodificados, o atraso é zero e a exibição acontece normalmente.

O tempo de *freezing* necessita da mesma condição do cálculo do atraso, sendo definido como:

$$\text{Tempo de Freezing (seg)} = \frac{\text{Qde. de Quadros Exibidos} - \text{Qde. de Quadros Decodificados}}{\text{Frequencia de Exibio de Quadros}}. \quad (6.3)$$

Para o cálculo do indicador em segundos, é feito o cálculo da quantidade de quadros exibidos durante um congelamento (a subtração da quantidade de quadros de vídeo exibidos menos a quantidade de quadros de vídeo decodificados), sendo esse valor dividido pela frequência de exibição de quadros que por padrão é de 60 quadros/segundo. Como um congelamento na exibição de um vídeo é facilmente notado pelo usuário, nos casos em que o tempo de *freezing* é maior que zero, a experiência do usuário foi afetada.

Observando os resultados com o *plugin* para MPEG-DASH proposto em [35] com o Wireshark, notou-se que, surpreendentemente, essa implementação não satisfaz a condição necessária de utilizar os pacotes CCN para a transmissão do conteúdo DASH e sim pacotes IP com adaptações de forma a simular pacotes CCN. Dessa forma, pacotes de interesse não são gerados para requisitar o conteúdo em MPEG-DASH. Sendo assim, depois de constatada esta condição, estes resultados foram descartados, já que não representam a transmissão utilizando CCN.

Como alternativa, buscou-se reproduzir o conteúdo DASH através da versão do VLC customizado com o *plugin* apenas para CCN, visto que a partir da versão 2.0 do VLC, o mesmo passou a ter suporte nativo ao DASH. Ao verificar as requisições feitas no wireshark, notou-se que, ao realizar a primeira troca de perfil de qualidade do vídeo, a reprodução do conteúdo era interrompida e o VLC parava de controlar as requisições de interesse, fazendo com que todos os demais pedidos fossem feitos no mesmo perfil. Com isso, este resultado também se mostrou insatisfatório, já que o *plugin* de DASH não se mostrou compatível com a CCN.

Como a reprodução do DASH não ocorreu com sucesso, foram analisados o comportamento da reprodução para o caso de uma transmissão não adaptativa, conforme apresentados nas Figuras 6.4, 6.5 e 6.6. Observa-se que o uso do TCP como encapsulamento dos pacotes CCN pelo emulador CCNx torna seu desempenho inviável visto que o percentual de perda e os tempos de atraso e *freezing* foram muito elevados. Ao utilizar o encapsulamento UDP, nota-se um melhor comportamento na reprodução do vídeo para os casos em que o gargalo na rede diminui, o que era esperado. Verifica-se também pelos valores do atraso percentual da reprodução do vídeo e o tempo de *freezing* que a transmissão ainda é muito ruim. A expectativa seria que com o uso do DASH, este comportamento fosse notadamente melhor.

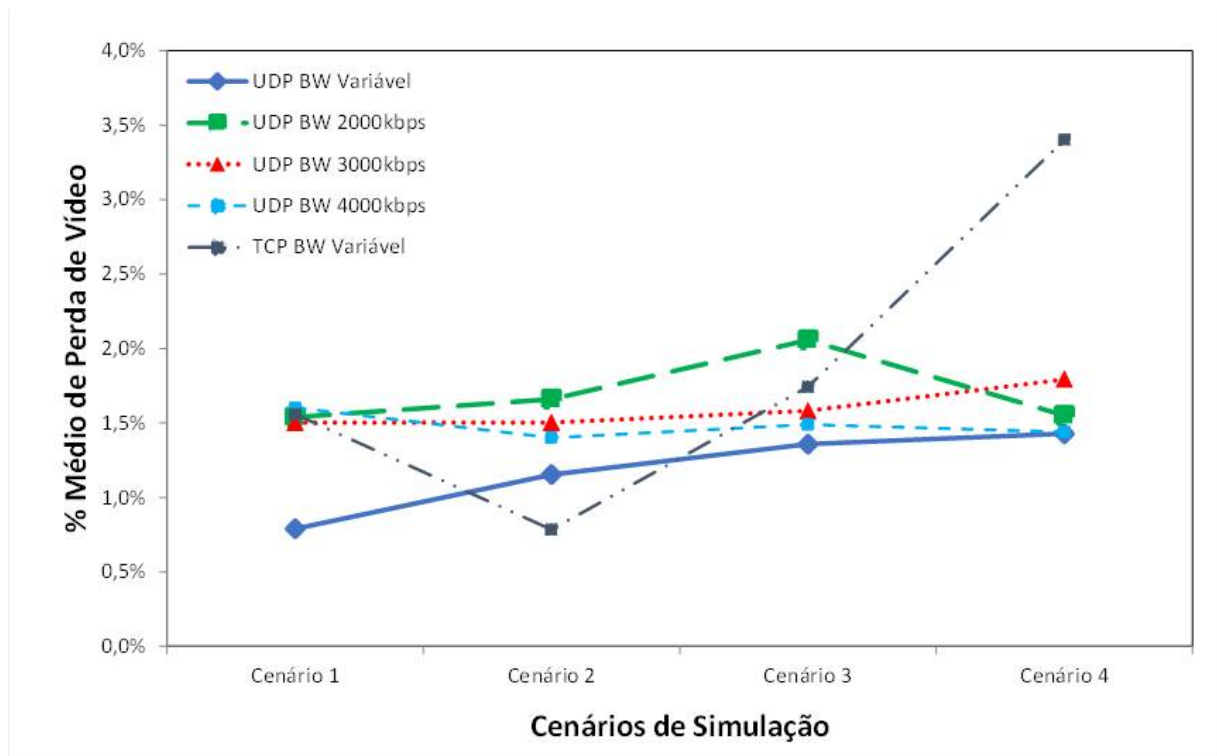


Figura 6.4: Percentual Médio de Perda de Vídeo

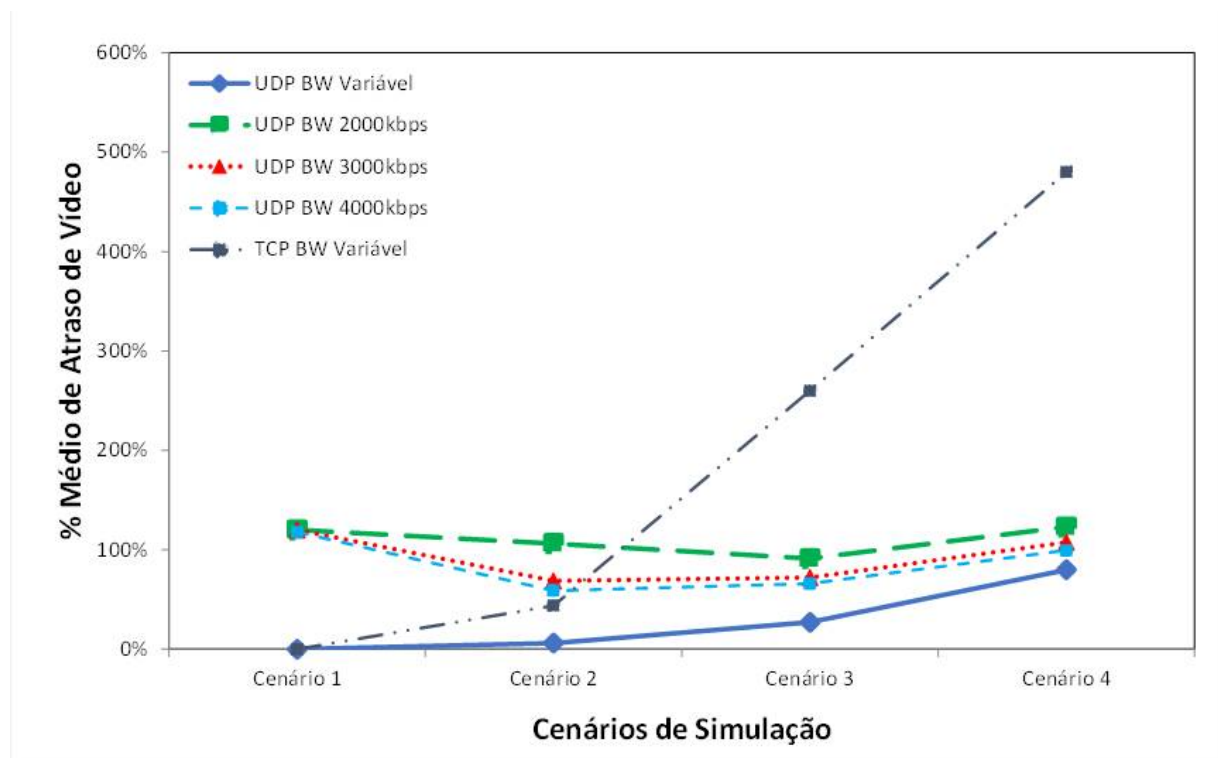
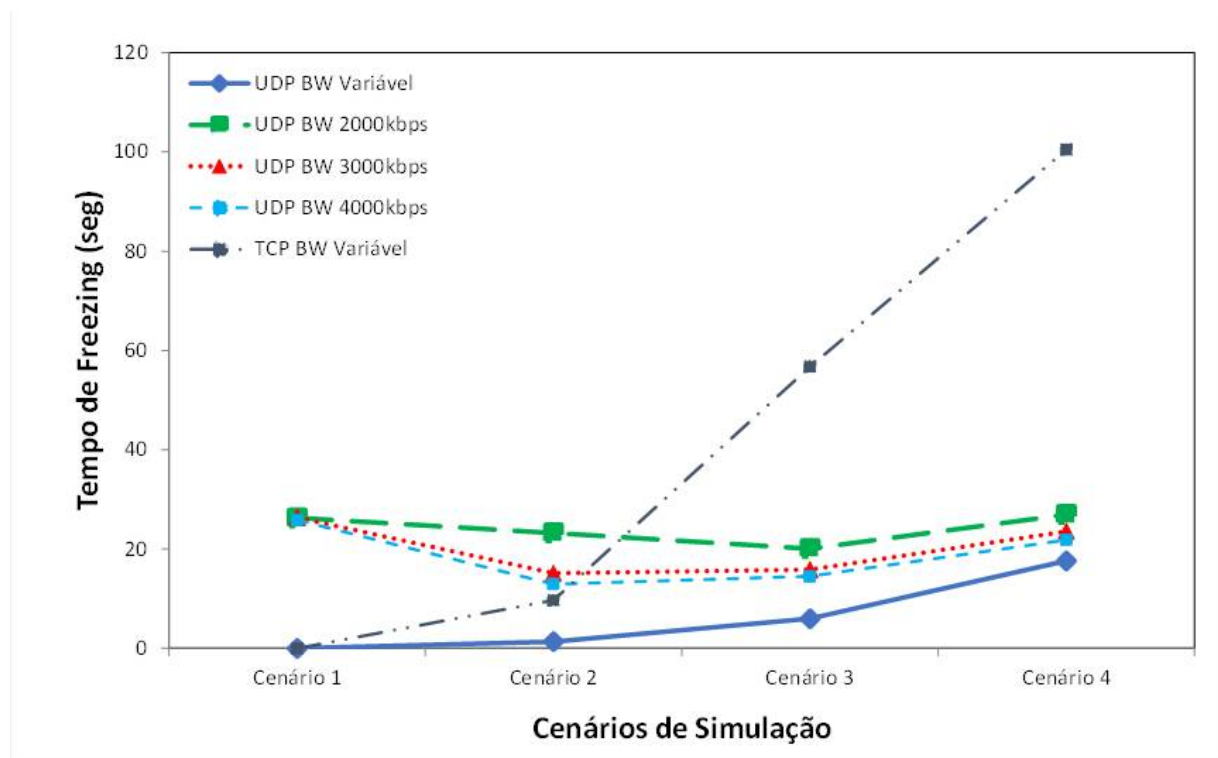


Figura 6.5: Percentual Médio de Atraso na Reprodução do Vídeo

Figura 6.6: Tempo de *Freezing* na Reprodução do Vídeo

Estes resultados foram motivadores para o estudo do controle da taxa de interesse na transmissão de vídeo, assim como foram importantes para observar que o uso do UDP é superior ao do TCP na CCNx. Estes resultados também foram observados na Etapa II dos testes, mostrada a seguir, de forma que o mecanismo proposto foi avaliado utilizando apenas a CCN sobre UDP.

6.2 Etapa II - Análise do Controle da Taxa de Interesse

Nesta análise, utilizou-se um ambiente de simulação para implementar a proposta de controle de taxa de interesse, de forma a analisar o *streaming* adaptativo em redes CCN de acordo com o algoritmo proposto. Como foi explicado na Seção 5.2, utilizou-se para esse fim o simulador ndnSIM. A simulação foi feita utilizando uma máquina virtual com 4 GB de RAM no Oracle Virtualbox na versão para Windows 4.2.16 r86992. O sistema operacional da máquina virtual utilizado foi o Ubuntu 14.04 LTS.

Para a realização das simulações de controle da taxa de interesse, foi utilizada uma topologia contemplando um nó consumidor responsável pela realização das requisições de interesse, dois roteadores intermediários e um nó produtor responsável pela entrega dos pacotes de dados. O *link* entre o nó consumidor e o roteador ao qual está conectado e o *link* entre o nó produtor e o roteador ao qual está conectado, possui largura de banda de 10Mbps e atraso de 2 ms. O link entre os roteadores foi configurado de duas maneiras, a primeira também com 10 Mbps de largura de banda e 2 ms de atraso e uma segunda configuração gerando um gargalo na rede, com 1 Mbps de largura de banda e 5 ms de atraso, conforme apresentado na Figura 5.4.

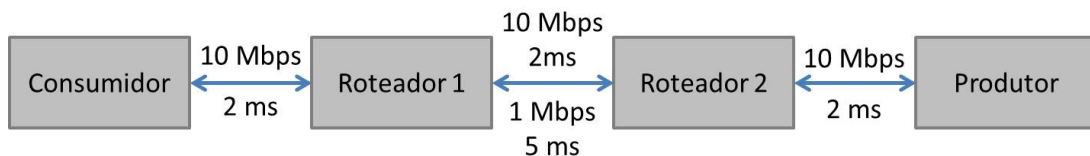


Figura 6.7: Topologia do simulador ndnSIM para os Testes de Simulação

A frequência inicial de envio de pacotes de interesse varia de um cenário de experimento para o outro, sendo os valores considerados de 100 pacotes/segundo, 500 pacotes/segundo e 2000 pacotes/segundo. A aplicação produtora dos pacotes de dados também foi configurada com o mesmo prefixo de nome a fim de criar os pacotes de dados para satisfazer os pacotes de interesse. Os pacotes de dados apresentam o tamanho fixo de 1024 bytes. Com isso, as taxas de vídeo variaram de 100 KB/s, 500 KB/s e 2 MB/s de

acordo com as frequências iniciais de envio de pacotes de interesse. Estas taxas foram utilizadas de maneira a representar a transmissão de um vídeo de baixa resolução.

Os cenários de simulação foram organizados de forma a simplificar o entendimento de acordo com as variações dos parâmetros de transmissão. A Tabela 6.2 apresenta o descritivo de cada cenário. Cada cenário de simulação foi repetido 10 vezes, sendo que cada rodada teve 30 segundos de duração.

Tabela 6.2: Cenários de Teste de Simulação

Cenários de Testes de Simulação Etapa II		Frequência Inicial		
		100 interesses	500 interesses	2000 interesses
Rede sem Gargalo Delay 2ms Link 10MB	Com Mecanismo	Cenário A1X	Cenário A1Y	Cenário A1Z
	Sem Mecanismo	Cenário B1X	Cenário B1Y	Cenário B1Z
	Com Mecanismo AIMD	Cenário C1X	Cenário C1Y	Cenário C1Z
Rede com Gargalo Delay 5ms Link 1MB	Com Mecanismo	Cenário A2X	Cenário A2Y	Cenário A2Z
	Sem Mecanismo	Cenário B2X	Cenário B2Y	Cenário B2Z
	Com Mecanismo AIMD	Cenário C2X	Cenário C2Y	Cenário C2Z

Os índices de aumento e decréscimo da frequência de interesse foram definidos conforme mostra a Tabela 6.3. Nessa tabela, o mecanismo com o aumento por soma e o decremento por subtração (Algoritmo 1) é chamado de 'Mecanismo AISD' e o mecanismo com soma e divisão (Algoritmo 2) é chamado de 'Mecanismo AIMD'. Esses nomes serão mantidos nos demais resultados.

O *script* de simulação foi configurado de forma que todo o roteamento global fosse automaticamente configurado entre os nós. A aplicação de consumidores e a aplicação de produtores foram configuradas nos nós consumidores e produtores respectivamente. Para a estratégia de encaminhamento, foi configurado o parâmetro de *flooding* no encaminhamento dos pacotes, definido o limite por face de saída através do controle da janela com o número de pacotes de interesses não atendidos e habilitados os pacotes NACKs.

Tabela 6.3: Índices de aumento e decréscimo da frequência.

	Mecanismo AISD	Mecanismo AIMD
Índice de aumento	1	1
Índice de decréscimo	1	0,99

Além dos pacotes NACK CONGESTION, a aplicação possui mais dois tipos de pacotes NACK: O NACK LOOP e o NACK GIVEUP PIT, que representam, respectivamente, casos em que ocorrem loops nas requisições de interesse e quando o interesse não pode ser atendido. Foi verificado que a atual versão do ndnSIM não possui implementado o envio de pacotes de congestionamento. Sendo assim, foi definido que o recebimento de qualquer tipo de pacote NACK como critério de identificação de um congestionamento e, com isso, proceder com a alteração da taxa de transmissão, sem que sejam gerados impactos no objetivo do experimento. Pois, devido a topologia proposta no experimento, a ocorrência de pacotes de interesse em *loop* não é verificada e os casos em que o pedido de interesse não pode ser atendido ocorrem quando o congestionamento acarreta a indisponibilidade da face.

Para a medição do resultado, foram utilizados indicadores como a frequência de pacotes de interesse enviados por segundo, a vazão de recebimento de dados, a taxa média de recebimento de dados ao variar a frequência de envio inicial e utilização do *buffer*. O indicador de uso do *buffer* foi desenvolvido de forma a mensurar a utilização do mesmo ao longo do experimento. Para isso foi necessário o cálculo da quantidade de pacotes de dados necessários para a reprodução de 1 segundo de vídeo. Como os pacotes de dados possuem o tamanho fixo de 1024 Bytes e o vídeo utilizado como referência possui 44 segundos de duração e o tamanho de 20689282 Bytes, logo serão necessários 20205 pacotes de dados para a reprodução de 44 segundos de vídeo. Portanto, a quantidade mínima de pacotes de dados para a reprodução de 1 segundo vídeo é de 460 pacotes de dados por segundo. Como o vídeo foi codificado a uma taxa variável, esse valor de pacotes por segundo pode sofrer alteração ao longo do tempo, porém é esperado em uma transmissão real que o *buffer* compense essa variação.

Logo, calculou-se a utilização do *buffer* da seguinte forma:

$$\text{Tamanho do Buffer (seg)} = \frac{\text{Qde de Pacotes de Dados Recebidos /seg}}{\text{Qde Min de Pacotes de Dados p/ exibir 1 seg video}} - 1 \quad (6.4)$$

Onde o resultado da fração representa o total de segundos de vídeo recebidos em um segundo de transmissão, sendo subtraído 1 segundo separado para a exibição, encontramos o total de segundos de vídeo armazenados no *buffer*

O detalhamento dos resultados obtidos é apresentado na sequência.

6.2.1 Resultados Obtidos

Dada a correlação comportamental existente entre os indicadores, seus resultados não podem ser analisados independentemente. Primeiramente, são apresentados os resultados dos experimentos que medem a variação da frequência de envio de das requisições de interesse para os cenários de simulação propostos. O primeiro cenário a ser avaliado é o caso em que é utilizado como mecanismo de controle o uso do aumento aditivo e redução subtrativa (*Additive-Increase Subtrative-Decrease* – AISD) como regra para adaptação da taxa. Na sequência, é apresentado o cenário em que não existe variação da frequência de envio, caso este em que não estão presentes os mecanismos de controle. Por fim, é apresentado o cenário em que o mecanismo de controle adotado baseia-se no aumento aditivo e na redução multiplicativa (*Additive-Increase Multiplicative-Decrease* – AIMD). Para melhor visualização dos gráficos, os cenários das topologias de rede com e sem gargalos foram separados.

Os resultados foram comparados de forma a avaliar o desempenho de cada um deles em relação ao cenário em que não é feito o controle da taxa. São observadas as vazões de dados das transmissões e avalia-se o impacto dos resultados obtidos para uma transmissão de vídeo.

O primeiro cenário avaliado, cujos resultados estão na Figura 6.8, é aquele onde a frequência inicial de envio de interesses é de 100 pacotes/segundo e sem a presença de gargalos na rede. Observa-se que o mecanismo aditivo/subtrativo apresentou um desempenho consideravelmente melhor que o cenário sem a adoção do mecanismo AISD e também apresentou um comportamento mais estável em relação ao mecanismo AIMD. No mecanismo AIMD, observa-se que nos segundos 1,6s, 12s, 23,4s e 29,2s houve ocorrências de frequências próximas a zero.

Ao realizar o mesmo experimento com a presença do gargalo na rede, o resultado apresenta um outro comportamento conforme apresentado na Figura 6.9. Neste caso os dois mecanismos de controle apresentam um comportamento bastante similar. Em ambos os casos a frequência foi melhor que o valor inicial e crescente ao longo do experimento.

Através da análise dos gráficos da vazão de dados recebidos tanto para as simulações sem gargalos na rede e com gargalos na rede, respectivamente Figuras 6.10 e 6.11, nota-se que o comportamento do mecanismo AISD resulta em uma taxa de recebimento de dados mais estável do que o mecanismo AIMD e em uma taxa 10 vezes maior que o cenário original quando não ocorrem gargalos na rede. Ao introduzir o gargalo na rede,

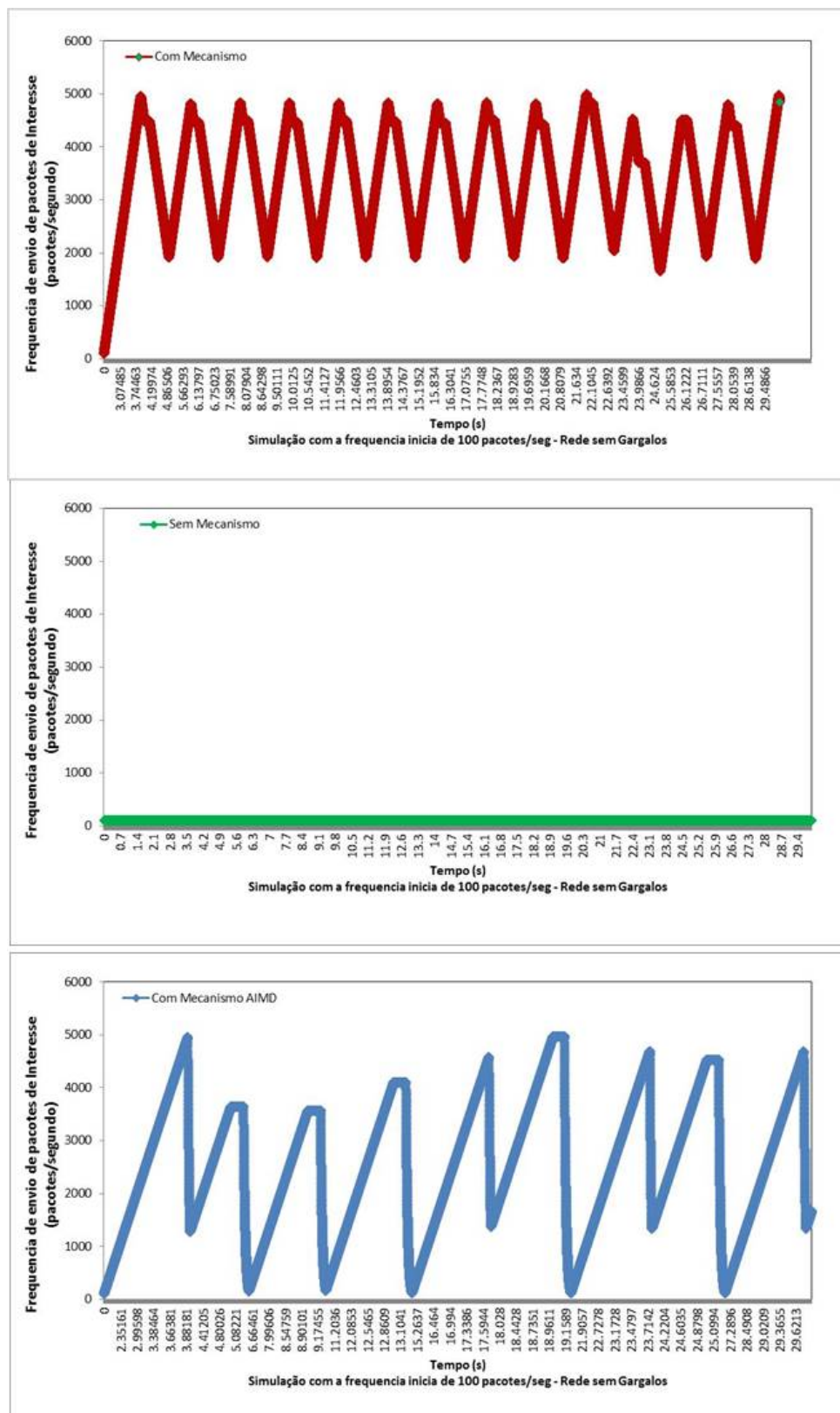


Figura 6.8: Frequência de envio de Pacotes de Interesse no Tempo - Topologia de Rede sem Gargalos e Frequência Inicial de 100 pacotes.

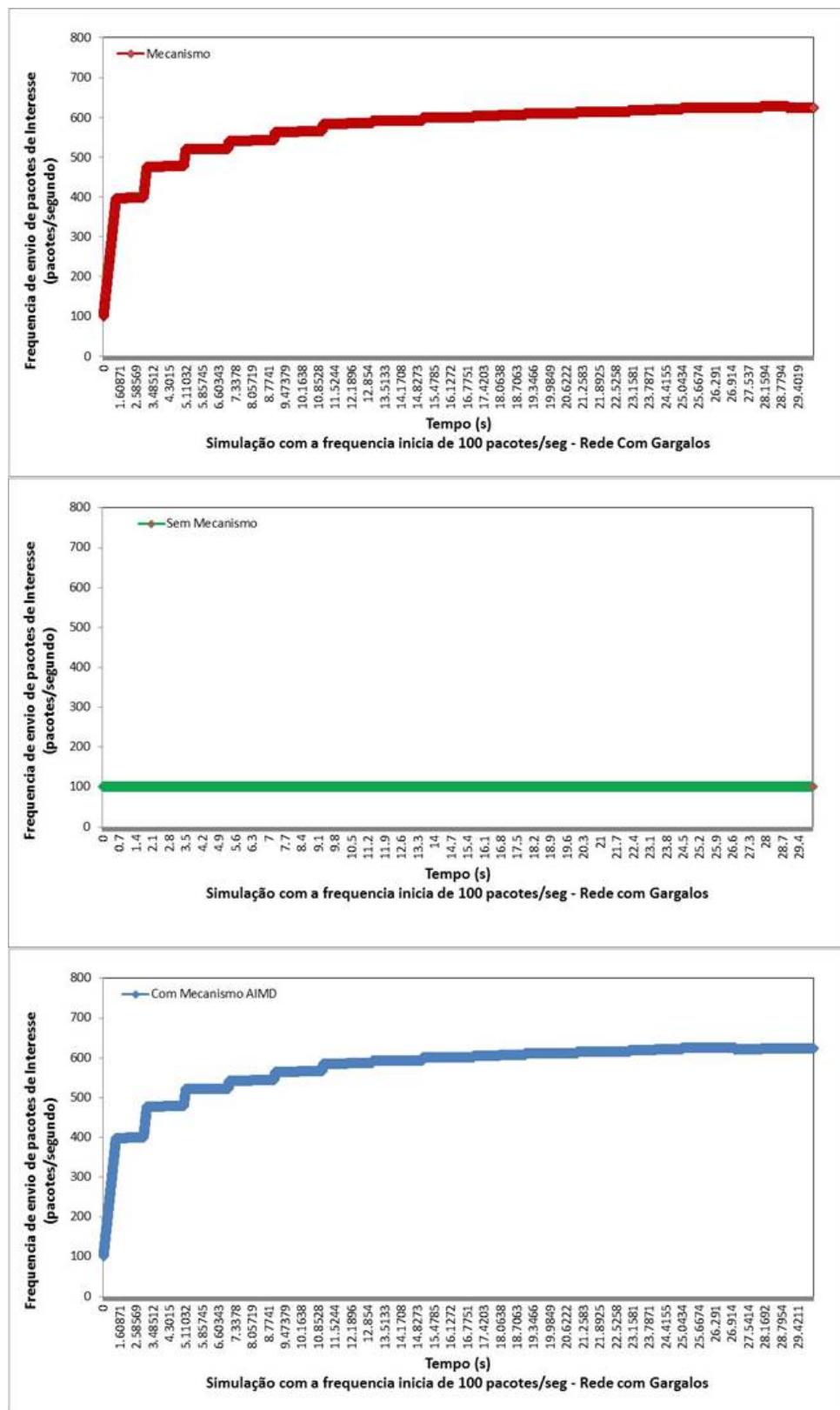


Figura 6.9: Frequência de envio de Pacotes de Interesse no Tempo - Topologia de Rede com Gargalos e Frequência Inicial de 100 pacotes de interesse enviados.

os três cenários ficam praticamente os mesmos, com uma pequena melhora percebida no mecanismo AISD.

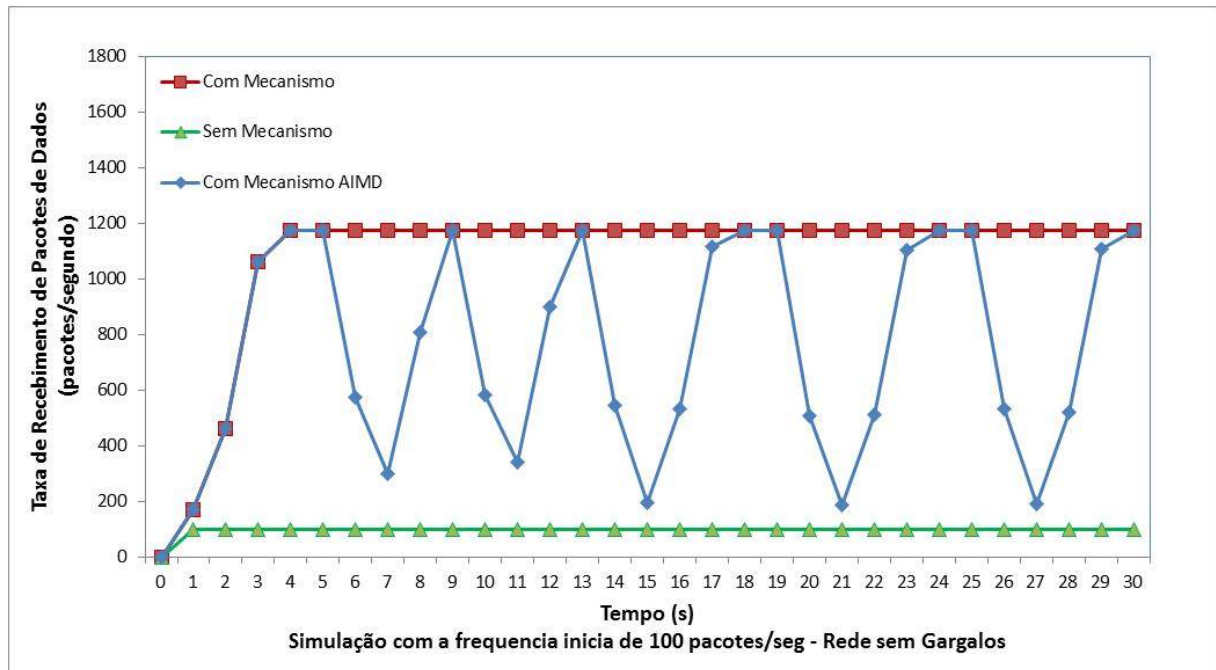


Figura 6.10: Taxa de Recebimento de Pacotes de Dados no Tempo - Topologia de Rede sem Gargalos e Frequência Inicial de 100 pacotes de interesse enviados.

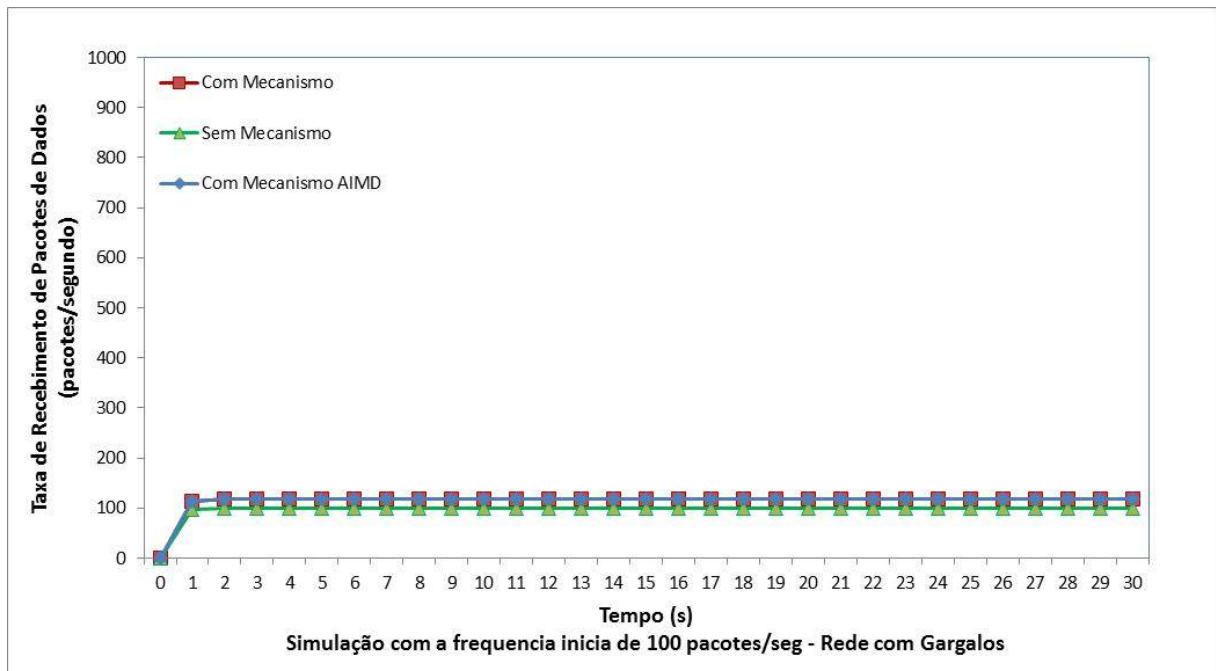


Figura 6.11: Taxa de Recebimento de Pacotes de Dados no Tempo - Topologia de Rede com Gargalos e Frequência Inicial de 100 pacotes de interesse enviados.

Ao elevar a frequência inicial para 500 pacotes de interesse por segundo, tanto o comportamento da variação da frequência, como a taxa de dados recebidos (Figuras 6.12 e 6.14), para os casos sem a ocorrência de gargalos na rede, permaneceram praticamente os mesmos que os anteriormente apresentados para a frequência inicial de 100 pacotes/segundo. Para as simulações com a presença de gargalos na rede, nota-se que o comportamento das frequências na Figura 6.13 indicam uma maior estabilização em torno do valor de 600 pacotes/segundo e uma redução aproximadamente no segundo 2. Nesta redução, é possível observar a diferença entre a redução subtrativa que reduz em 1 a frequência a cada envio de pacote e o mecanismo AIMD que foi configurado a reduzir 1% a frequência a cada envio de pacotes. Ao verificar a taxa recebida na Figura 6.15, verifica-se que o comportamento não sofreu alteração em relação à simulação anterior.

Ao realizar as simulações com a frequência inicial de 2000 pacotes de interesse por segundo, observa-se na Figura 6.16 que, no caso da rede não apresentar gargalos, a frequência do modelo AISD permanece similar ao comportamento das outras simulações com taxas mais baixas (Figuras 6.8 e 6.12), porém a amplitude da frequência do modelo AIMD apresenta maiores variações próximas à frequência zero. Ao realizar as simulações com o gargalo na rede, Figura 6.17, percebe-se uma redução contínua na frequência de envio de interesses e em ambos os casos, tanto no mecanismo AISD, como no AIMD, houve a saturação do *link* em próximo a 600 pacotes/segundo. As taxas de recebimentos de dados sem gargalo na rede Figura 6.18 permanecem as mesmas, e em torno da taxa de 1200 pacotes de dados/segundo. As taxas de recebimentos de dados com gargalo na rede Figura 6.19 permanecem as mesmas, e em torno da taxa de 100 pacotes de dados/segundo.

Com o resultado desses indicadores, é possível notar que quando ocorrem gargalos no *link*, é notada a sua saturação, com o envio de aproximadamente 600 pacotes de interesse/segundo e o recebimento de apenas 100 pacotes de dados/segundo. Essa condição possibilita a identificação da relação entre o tamanho dos pacotes de interesse e os pacote de dados como sendo da ordem de 1 para 6. No caso de não haver gargalos nos *links*, foi notada a saturação apenas quando houve o recebimento de aproximadamente 1200 pacotes de dados por segundo, o que, de acordo com a proporção encontrada anteriormente, indicaria uma saturação de pacotes de interesses da ordem de 7200 pacotes de interesse sendo enviados por segundo.

Ao analisar como varia a taxa média de recebimento de pacotes de dados, como mostrado na Figura 6.20, verifica-se que o mecanismo AISD tem um desempenho melhor do que quando comparados com o mecanismo AIMD e a transmissão sem o uso de um

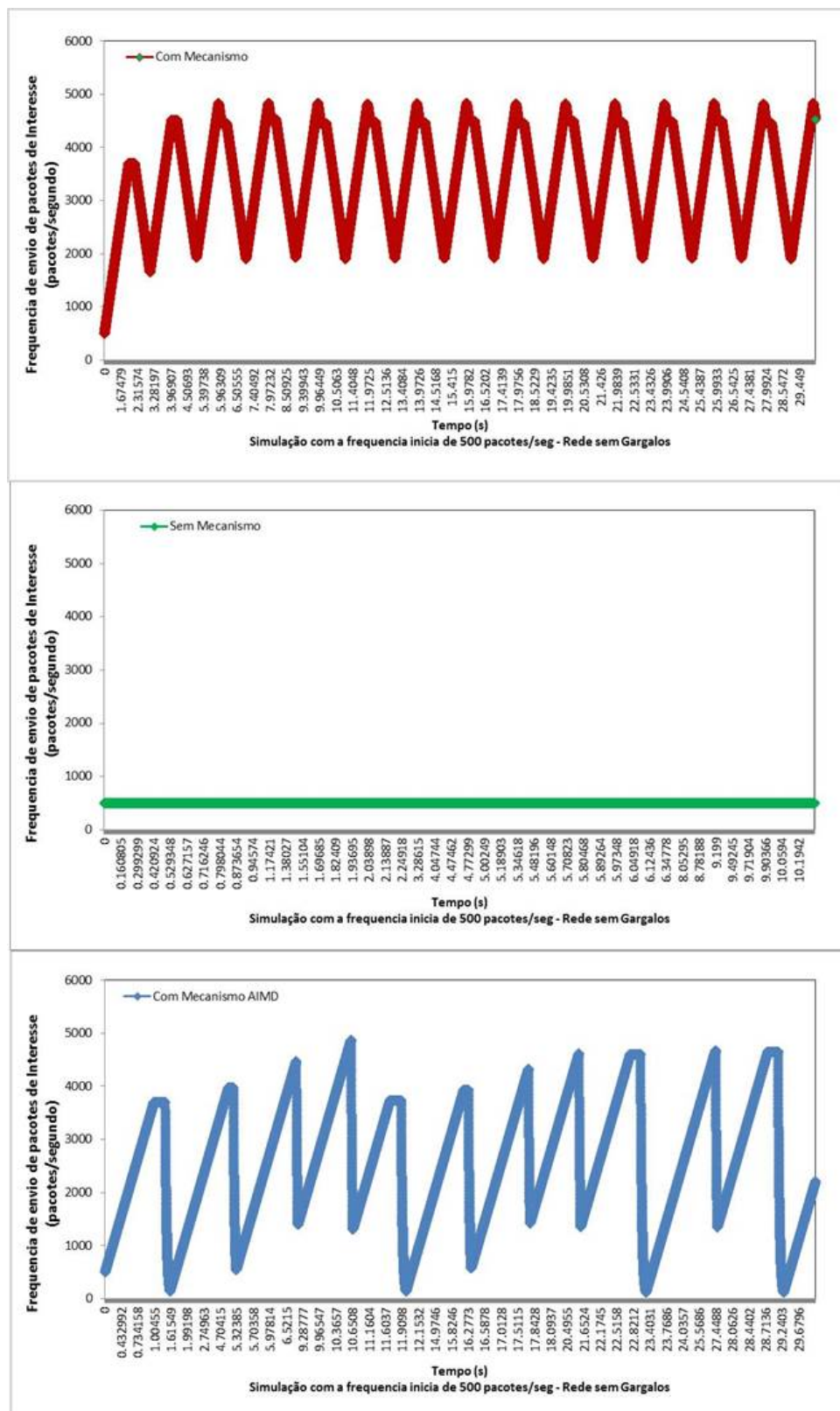


Figura 6.12: Frequência de envio de Pacotes de Interesse no Tempo - Topologia de Rede sem Gargalos e Frequência Inicial de 500 pacotes de interesse enviados.

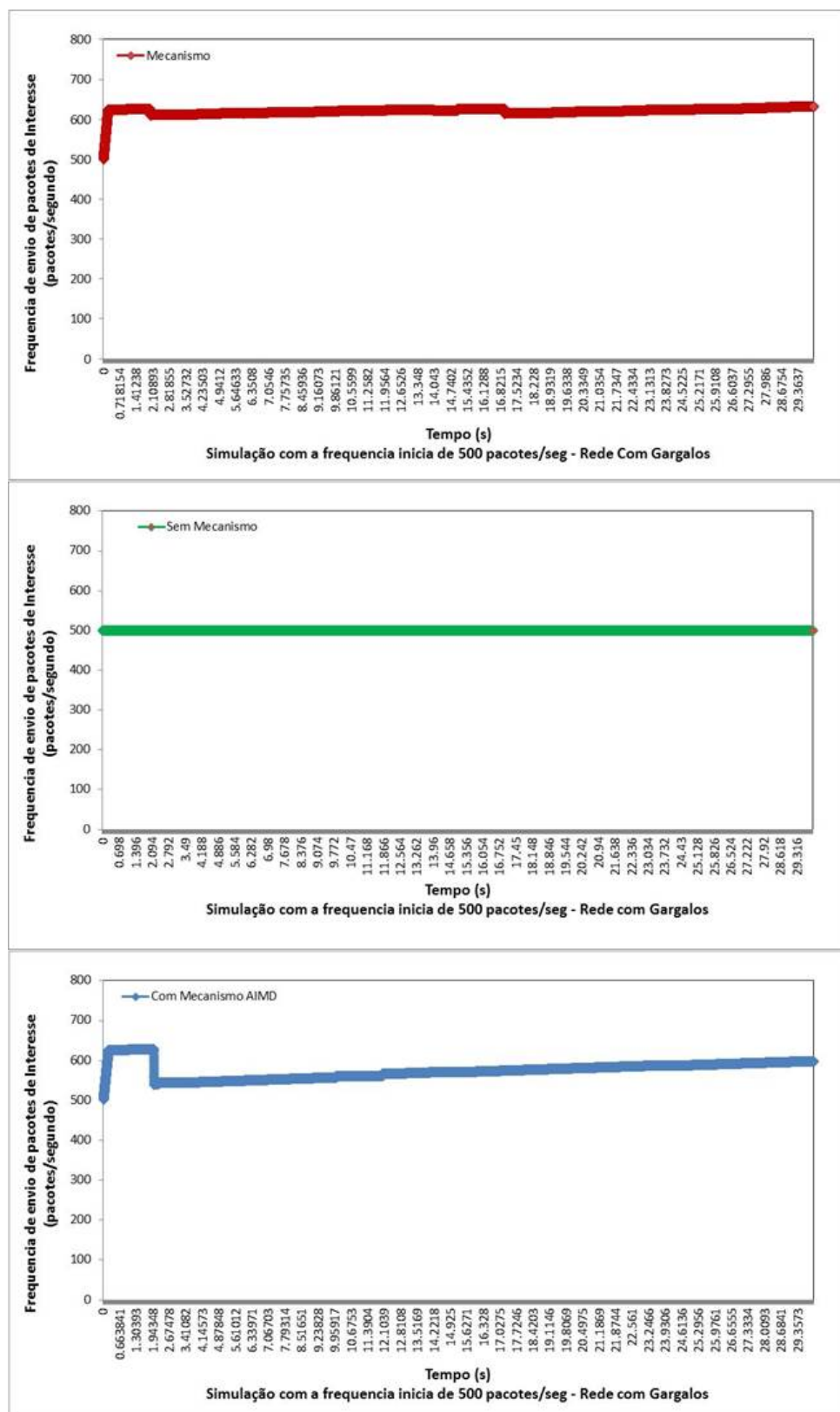


Figura 6.13: Frequência de envio de Pacotes de Interesse no Tempo - Topologia de Rede com Gargalos e Frequência Inicial de 500 pacotes de interesse enviados.

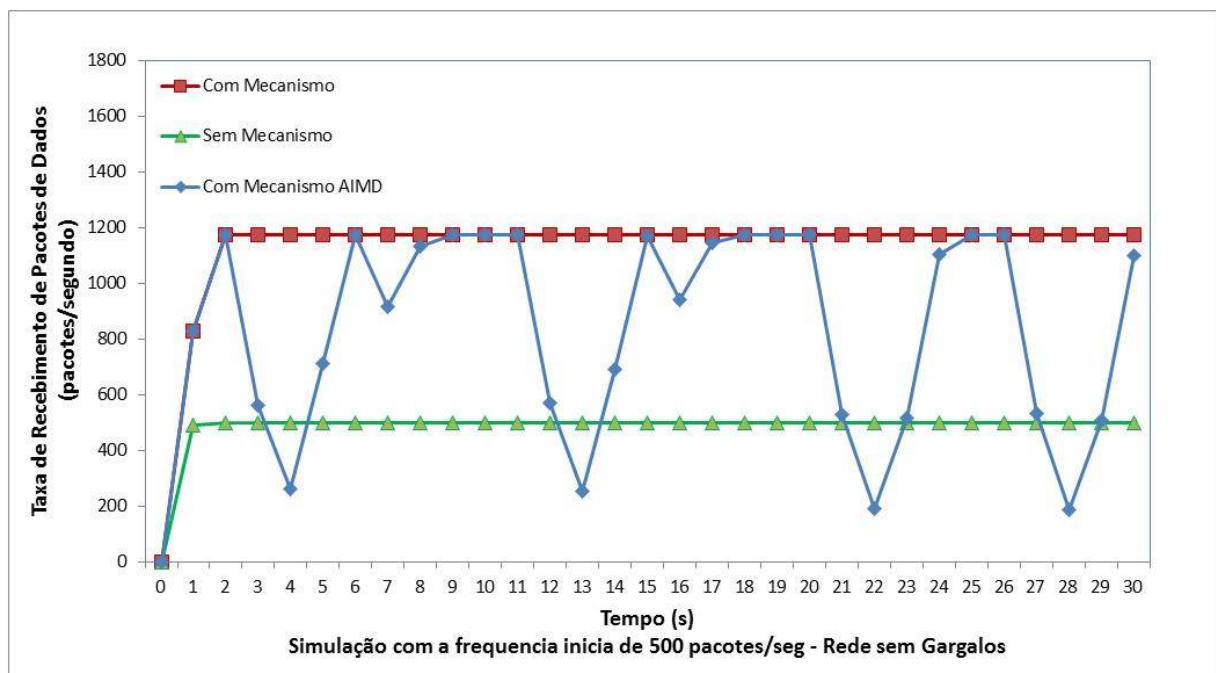


Figura 6.14: Taxa de Recebimento de Pacotes de Dados no Tempo - Topologia de Rede sem Gargalos e Frequência Inicial de 500 pacotes de interesse enviados.

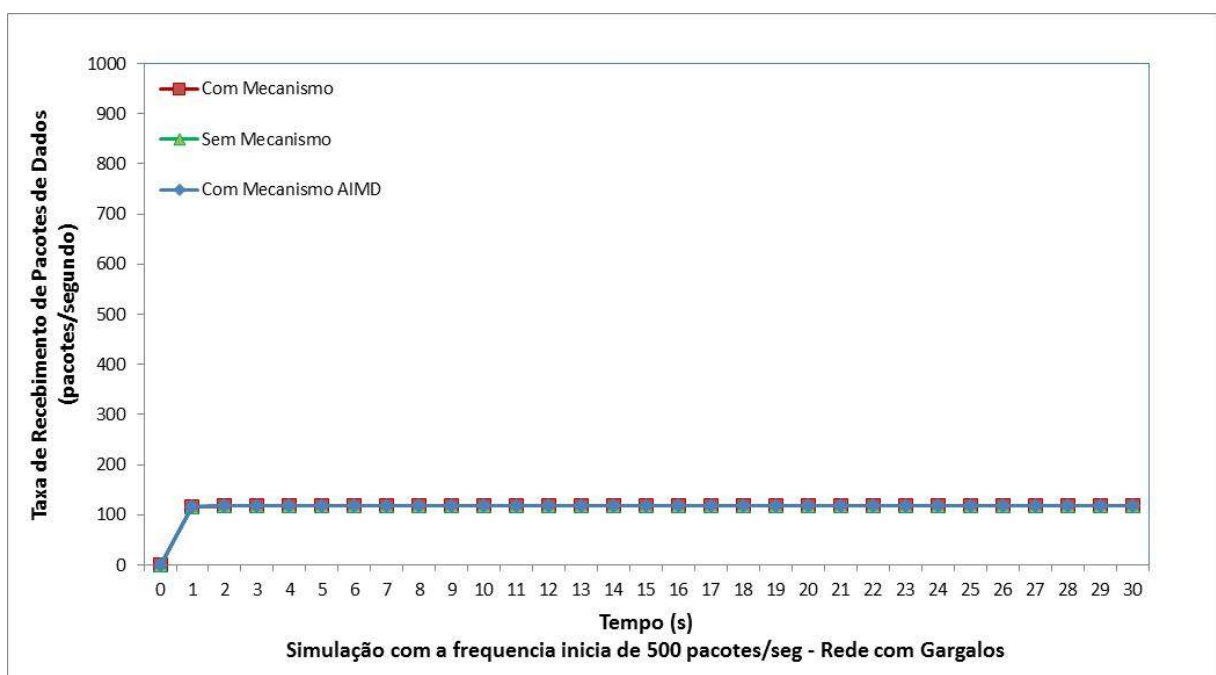


Figura 6.15: Taxa de Recebimento de Pacotes de Dados no Tempo - Topologia de Rede com Gargalos e Frequência Inicial de 500 pacotes de interesse enviados.

mecanismo de controle no envio dos pacotes de interesse. Seu desempenho é sempre próximo da taxa de saturação do link, o que indica um excelente desempenho. Embora

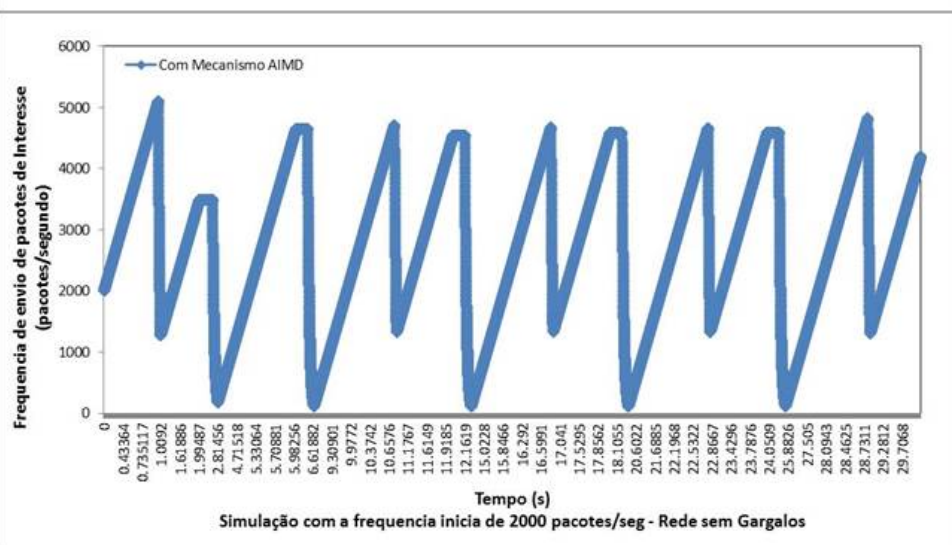
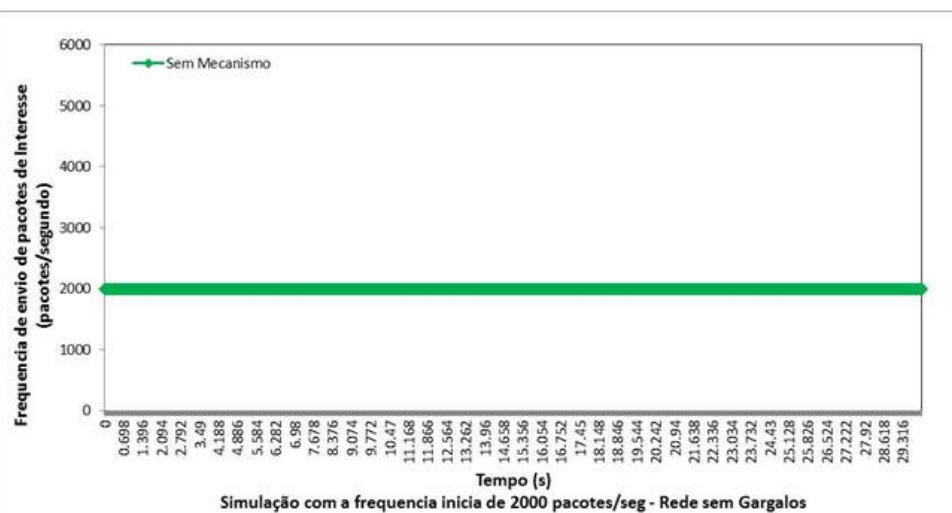
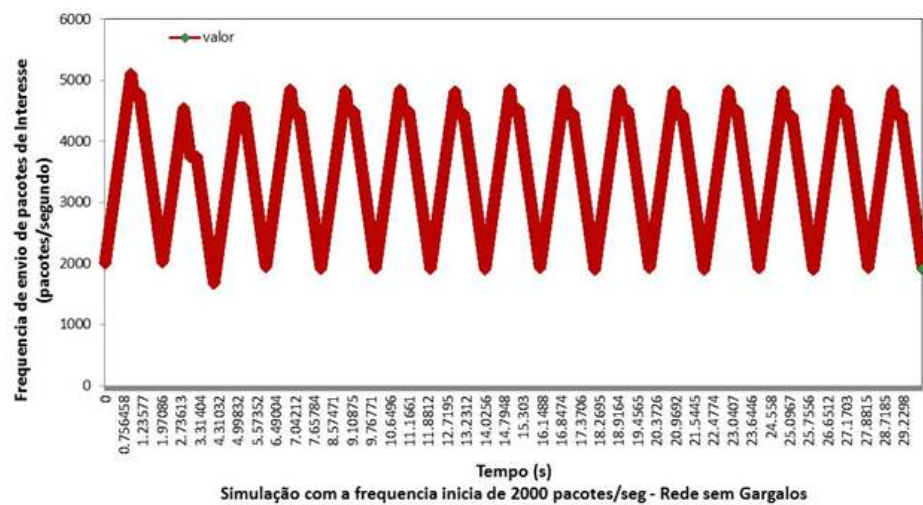


Figura 6.16: Frequência de envio de Pacotes de Interesse no Tempo - Topologia de Rede sem Gargalos e Frequência Inicial de 2000 pacotes de interesse enviados.

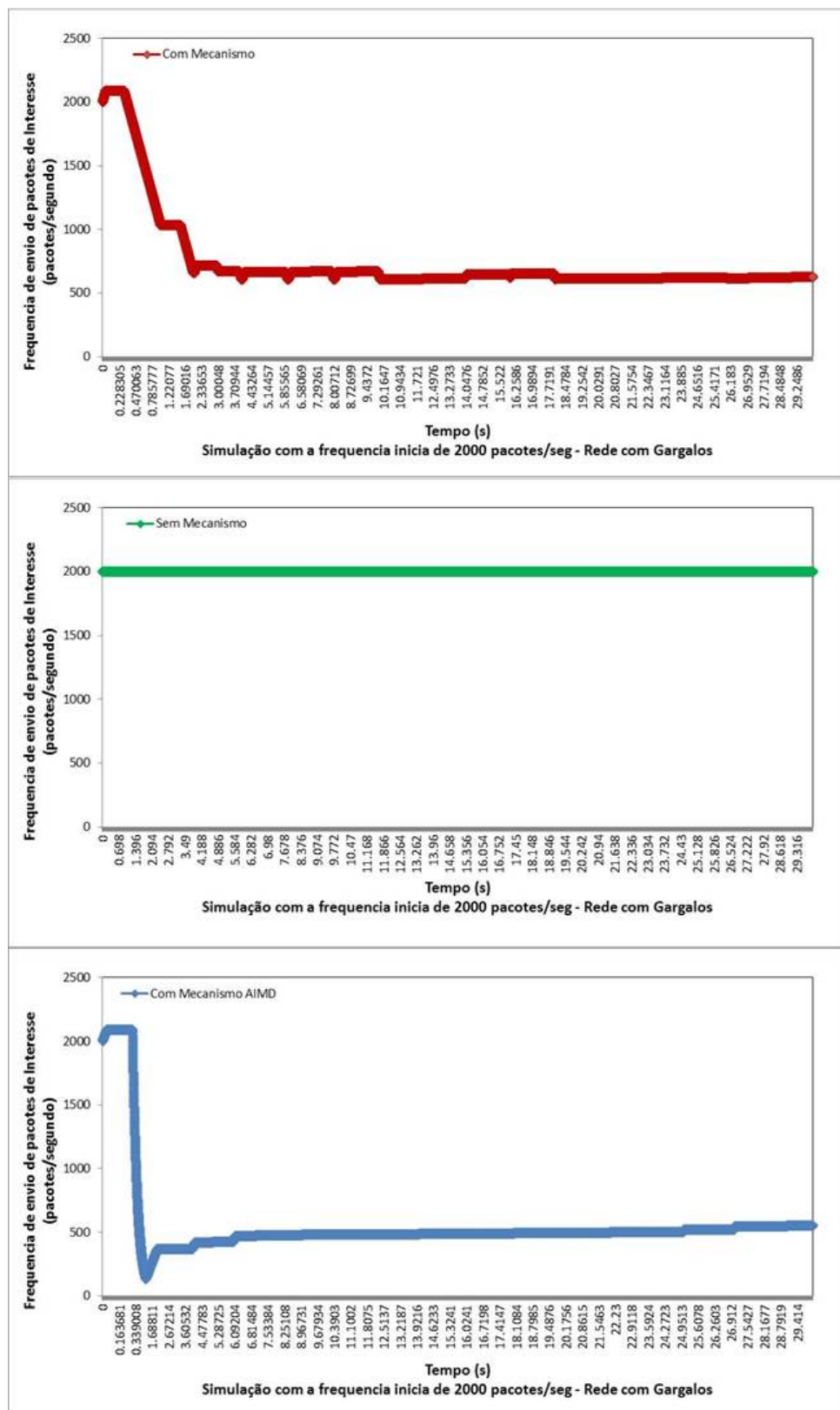


Figura 6.17: Frequência de envio de Pacotes de Interesse no Tempo - Topologia de Rede com Gargalos e Frequência Inicial de 2000 pacotes de interesse enviados.

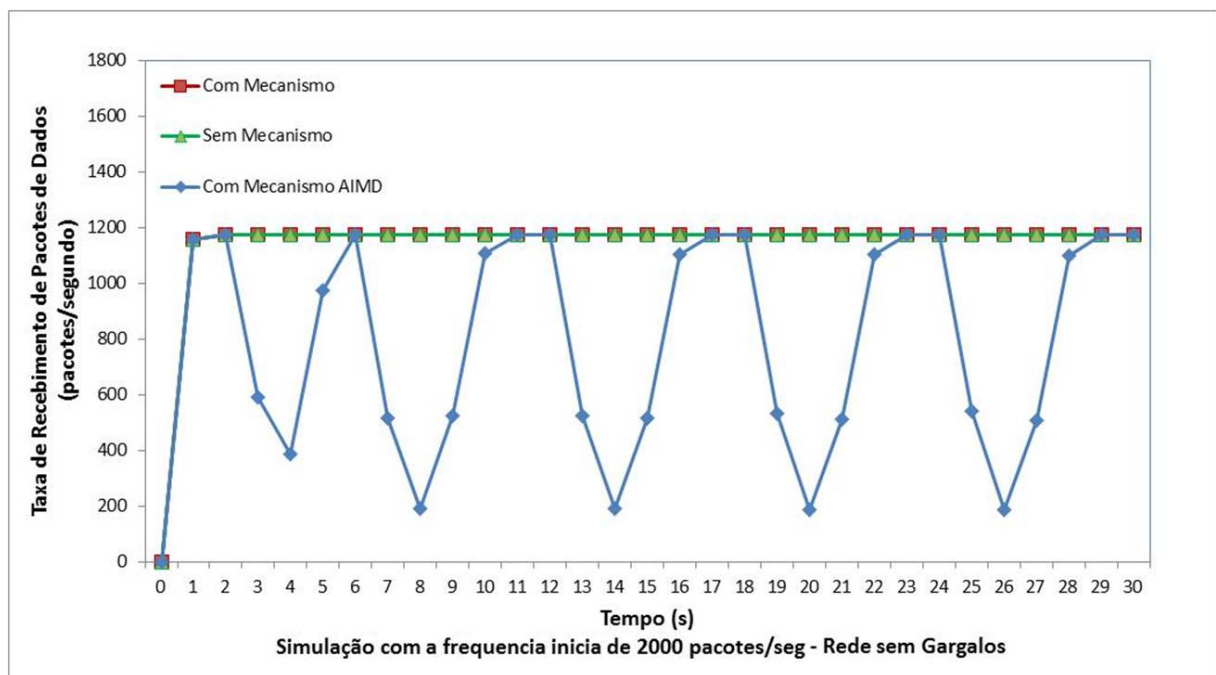


Figura 6.18: Taxa de Recebimento de Pacotes de Dados no Tempo - Topologia de Rede sem Gargalos e Frequência Inicial de 2000 pacotes de interesse enviados.

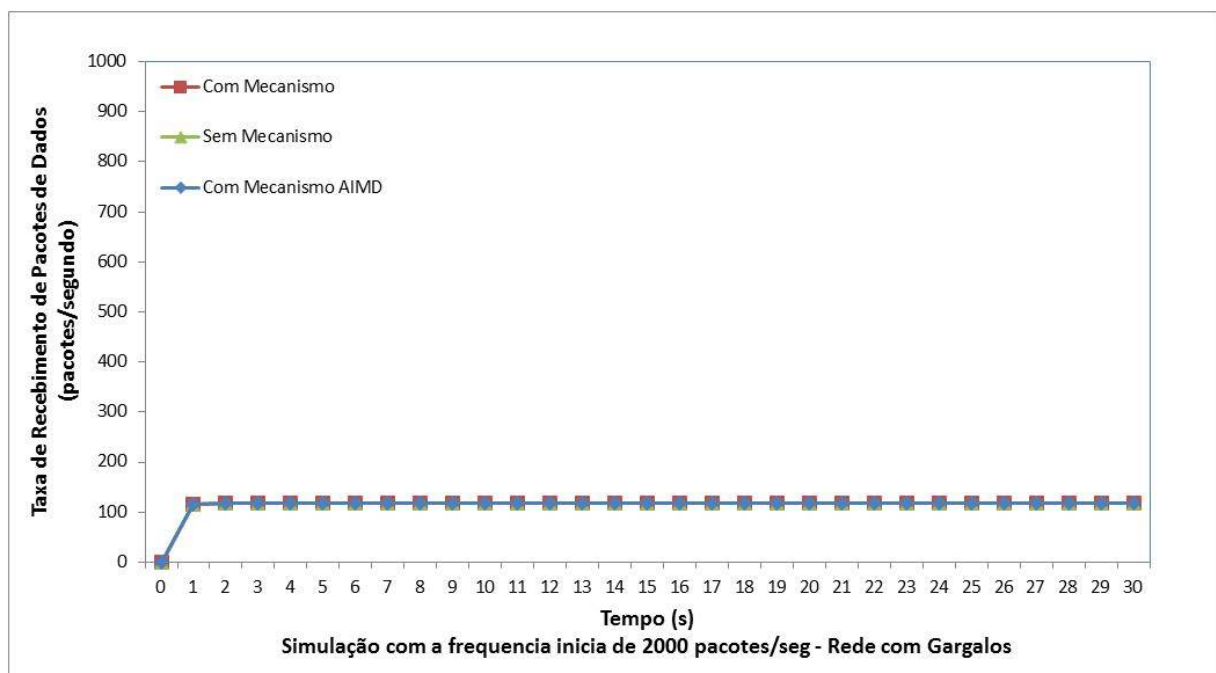


Figura 6.19: Taxa de Recebimento de Pacotes de Dados no Tempo - Topologia de Rede com Gargalos e Frequência Inicial de 2000 pacotes de interesse enviados.

o comportamento do mecanismo AIMD também seja superior em relação à transmissão sem o uso de mecanismos de controle em frequências mais baixas de envio de pacotes de

interesse, nota-se que, ao elevar a frequência inicial de envios de interesse, a transmissão com taxa constante passa a ter um desempenho melhor. Ao inserir o gargalo no link, nota-se que todos os cenários atingiram a taxa de saturação do link.

As análises de como essas simulações se comportariam caso fossem utilizadas para uma transmissão de vídeo, através da análise do buffer acumulado, são apresentadas na Figura 6.21. Essa figura mostra que na condição de não haver gargalos na rede, os dois mecanismos de controle propostos, o AISD e o AIMD, apresentam um buffer acumulado ao longo da simulação o que representa um ganho real em relação à transmissão sem a presença dos mecanismos, pois com a taxa de envio de interesses menor, foi possível adaptar a transmissão de forma que, para o usuário, o conteúdo seria reproduzido sem interrupções e sem atrasos.

Com isso, o uso do mecanismo AISD é o mais indicado para o atendimento do objetivo desta dissertação de propor um mecanismo de controle adaptativo de taxa de pacotes de interesse para a transmissão de vídeo em CCN que se assemelhe ao MPEG DASH, utilizando o desenvolvimento de um módulo de simulação para o ndnSIM para avaliação de adaptação de taxa de dados em *streamings* de vídeo.

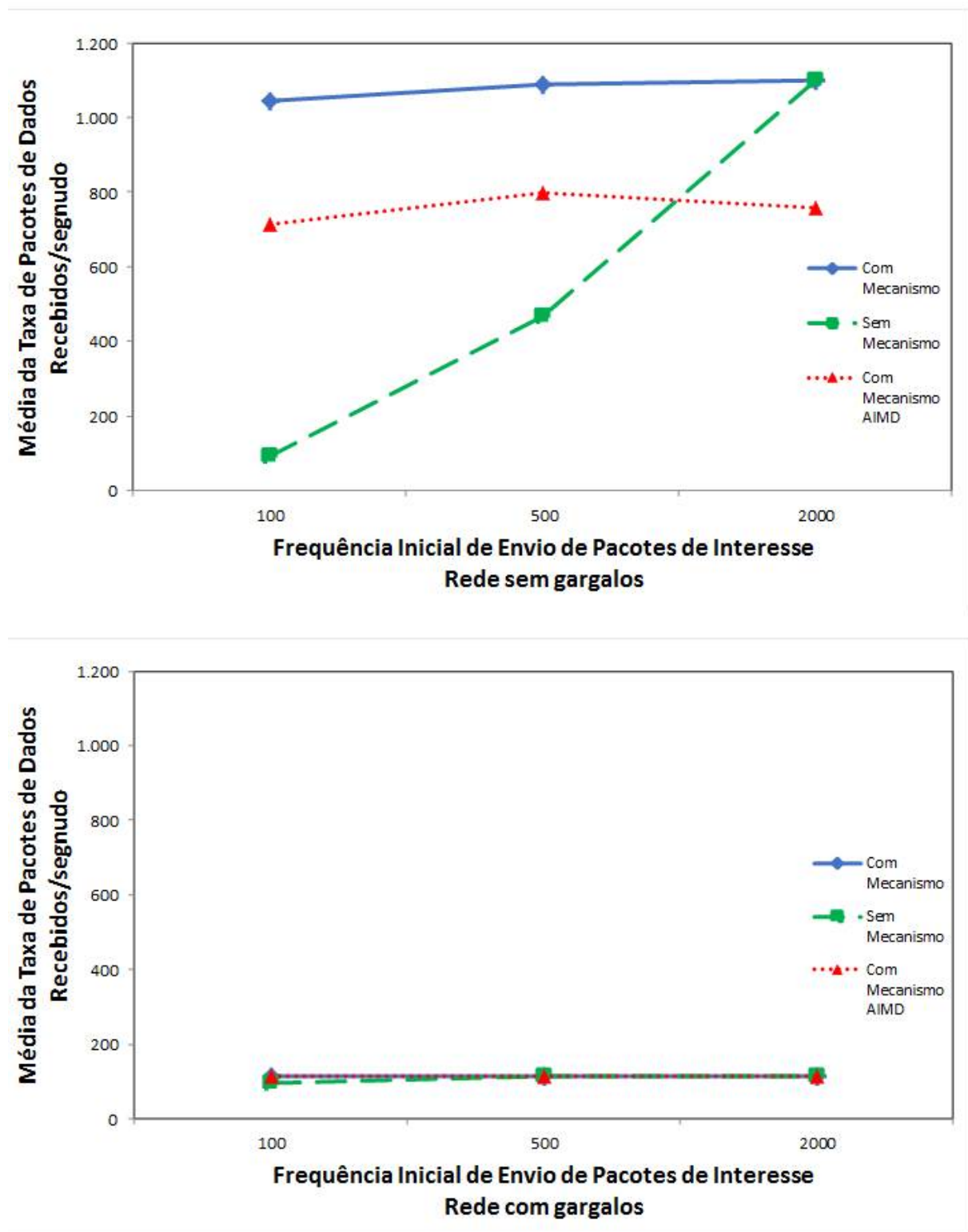


Figura 6.20: Média da Taxa de Pacotes de Dados Recebidos por Segundo.

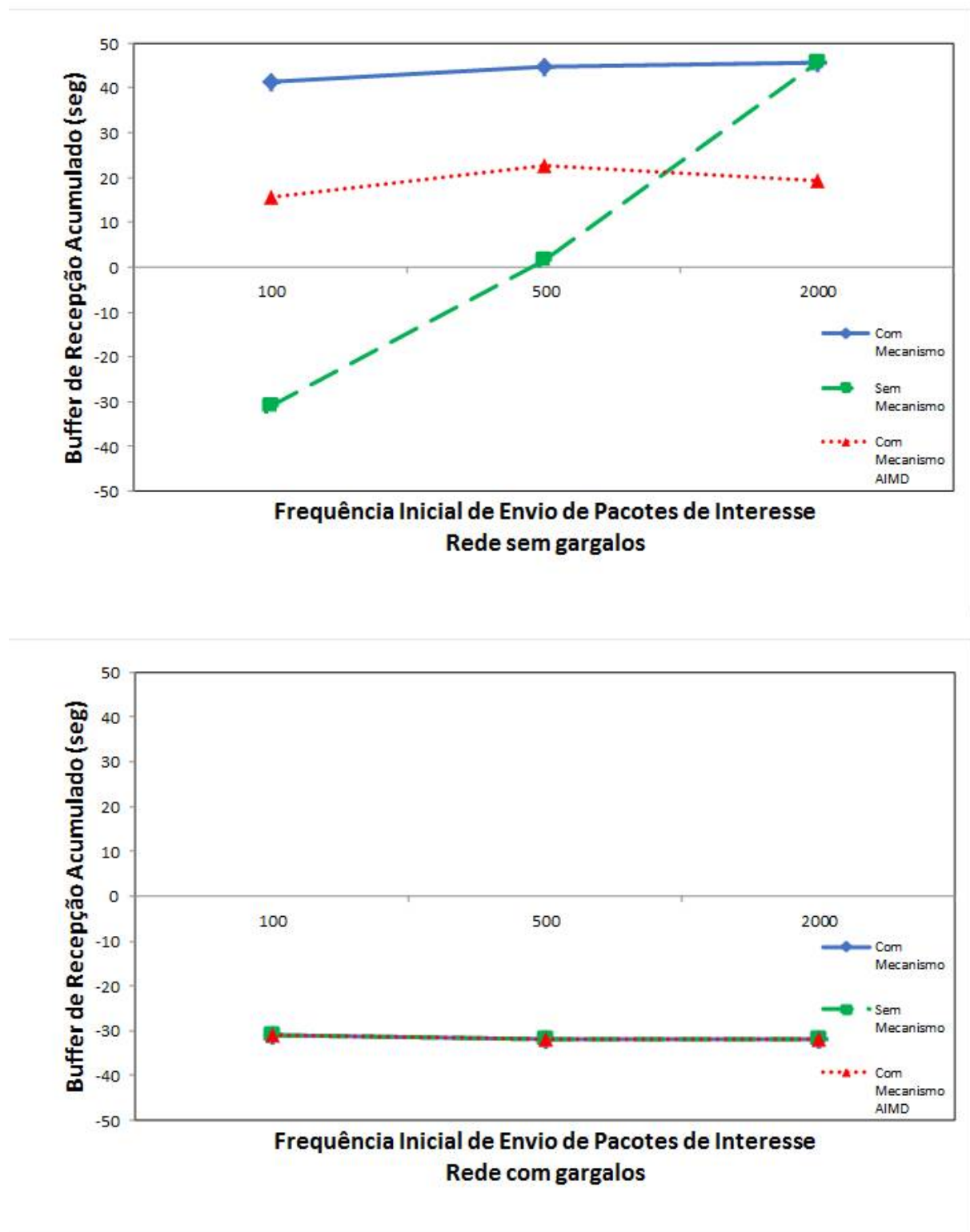


Figura 6.21: Buffer de Recepção Acumulado pela Frequência Inicial

Capítulo 7

Conclusão

Conforme apresentado ao longo desta apresentação, as redes CCN são um campo de estudo promissor e com aplicabilidade a diversos serviços existentes hoje em dia. Para sua coexistência com as tecnologias de rede já empregadas, é necessário que as suas funcionalidades sejam compatíveis com a tecnologia em produção de forma a facilitar a sua adoção.

Aplicações de transmissão de conteúdo multimídia são chaves para as redes CCN. Contudo, observa-se que existe muito pouco progresso nessa área. As propostas ainda são bem iniciais, restringindo-se apenas à transmissão de vídeo ou, em outros casos, não atendendo os requisitos básicos de uma rede CCN. De fato, até o momento, a análise de *streamings* com taxa adaptativa é restrita, por falta de ferramentas que permitam testes deste tipo.

O objetivo da proposta foi apresentar um mecanismo de controle adaptativo da taxa de envio de pacotes de interesse para redes CCN em um ambiente de simulação. Foram analisados trabalhos relacionados ao objetivo proposto, porém nenhum deles apresentava um mecanismo controlado diretamente pelo cliente. Os trabalhos que analisaram a transmissão de *streaming* adaptativo em DASH em redes CCN não eram implementados conforme o objetivo proposto requiritava. Ao buscar a realização da transmissão da maneira correta, foram encontrados problemas na reprodução do vídeo pela aplicação. Desta maneira, foi necessária a simulação de uma transmissão de maneira similar a uma transmissão em *streaming* DASH.

Cabe ressaltar que a taxa de bits necessária para a reprodução do vídeo em DASH é definida no momento em que o vídeo é codificado de acordo com os diferentes perfis de qualidade desejados. Uma vez definida essa taxa, a mesma não é alterada. A al-

ternância da taxa de transmissão é que possibilita a variação de um perfil para outro e consequentemente um consumo mais eficiente da rede.

Para o desenvolvimento do mecanismo de controle, foi apresentada uma arquitetura completa de controle que utiliza as informações da rede de forma direta, sem a necessidade do recebimento de dados do lado servidor para que o mecanismo funcione.

O mecanismo é capaz de reduzir e aumentar a taxa de envio dos pacotes de interesse automaticamente, de acordo com as variações da rede. Os dados coletados nos experimentos mostraram que ao ativar o mecanismo de controle, houve um ganho significativo na entrega dos pacotes de dados, tornando a transmissão como um todo melhor. Os mecanismos propostos são capazes de se ajustar à condição da rede, apresentam um resultado melhor quando comparados a transmissão de vídeo sem o uso de mecanismos de controle e com a taxa de envio de pacotes de interesse constante.

Os ganhos encontrados eram esperados, uma vez que são conhecidas as vantagens do MPEG DASH com relação a um *streaming* com taxa constante em redes IP. O benefício aqui apresentado é a análise desse comportamento adaptativo dentro da CCN, onde a taxa da transmissão é controlada exclusivamente pelo cliente, através do controle da taxa de envio de pacotes de interesse. Portanto, cabe à aplicação do cliente tanto o controle do conteúdo quanto o controle do congestionamento da rede. Os algoritmos propostos apresentaram um bom resultado, se mostrando eficientes para o controle de um *streaming* com taxa variável em redes CCN, sem sobrecarregar a rede com a tentativa de transmissão de vídeo a uma taxa superior àquela suportada pela rede. Portanto, o mecanismo proposto foi capaz de controlar a taxa de envio de pacotes de interesse conforme os objetivos propostos nesta dissertação.

7.1 Trabalhos Futuros

A fim de melhorar o desempenho do mecanismo de controle da taxa de envio dos pacotes de interesses, alguns campos de estudos são promissores, como a identificação de valores ideais para os índices de variação da frequência de envio, o uso de novos repositórios de nomes de forma que não só a frequência seja controlada, como também os tipos de pacotes de dados serão requisitados. O controle do conteúdo do cache, ao se considerar uma rede com mais clientes, também é um objeto de estudo interessante.

A integração desses mecanismos de *streaming* adaptativo e CCN já foi abordada, porém não de uma maneira conclusiva. O trabalho apresentado traz uma contribuição

para esse campo, através tanto da proposta quanto da análise realizada. Contudo, a integração do controle do nome do conteúdo ao simulador também é interessante e será realizada em futuros trabalhos. De fato, a realização de experimentos de que utilizem todos os blocos componentes da arquitetura proposta pode gerar novos resultados tão motivadores como os apresentados nesta dissertação.

Outro trabalho futuro seria a implementação dos algoritmos analisados e do controle do nome do conteúdo no CCNx, de forma a poder avaliá-la dentro de uma rede real.

Referências

- [1] ADOBE. Http dynamic streaming specification version 3.0 final, 2013. Disponível em <http://www.images.adobe.com/content/dam/Adobe/en/devnet/hds/pdfs/adobe-hds-specification.pdf>.
- [2] AFANASYEV, A.; MOISEENKO, I.; ZHANG, L. ndnSIM: NDN simulator for NS-3. Technical Report NDN-0005, NDN, October 2012.
- [3] AHLGREN, B.; DANNEWITZ, C.; IMBRENDA, C.; KUTSCHER, D.; OHLMAN, B. A survey of information-centric networking. *Communications Magazine, IEEE* 50, 7 (July 2012), 26–36.
- [4] AKHSHABI, S.; BEGEN, A. C.; DOVROLIS, C. An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems* (New York, NY, USA, 2011), MMSys '11, ACM, pp. 157–168.
- [5] APPLE. Http live streaming overview, 2014. Disponível em <https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/Introduction/Introduction.html>.
- [6] BAE, Y.; KIM, J.; JANG, M.-W.; LEE, B.-J. Ccn naming scheme for scalable routing and incremental deployment. In *Consumer Electronics (ICCE), 2014 IEEE International Conference on* (Jan 2014), pp. 422–423.
- [7] BANKS, J. *Discrete-Event System Simulation*, 3 ed. Prentice Hall, 2001.
- [8] BRAUN, S.; MONTI, M.; SIFALAKIS, M.; TSCHUDIN, C. Ccn and tcp co-existence in the future internet: Should ccn be compatible to tcp? In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on* (May 2013), pp. 1109–1115.
- [9] CAROFIGLIO, G.; GALLO, M.; MUSCARIELLO, L. Icp: Design and evaluation of an interest control protocol for content-centric networking. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on* (March 2012), pp. 304–309.
- [10] CAROFIGLIO, G.; GALLO, M.; MUSCARIELLO, L.; PAPALINI, M.; WANG, S. Optimal multipath congestion control and request forwarding in information-centric networks. In *Network Protocols (ICNP), 2013 21st IEEE International Conference on* (Oct 2013), pp. 1–10.
- [11] CHEN, J.; ZHOU, H.; ZHANG, H. Ditnm: Dynamic interest transmission scheme in content centric networking with name mapping. In *Communications (ICC), 2014 IEEE International Conference on* (June 2014), pp. 3197–3202.

- [12] CHI, J.; ZHANG, Y.; PENG, J.; TANG, H.; ZHOU, X. Design and analysis on routing by hierarchical and flat names. In *Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on* (Oct 2011), pp. 452–457.
- [13] CIANCAGLINI, V.; PIRO, G.; LOTI, R.; GRIECO, L.; LIQUORI, L. Ccn-tv: A data-centric approach to real-time video services. In *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on* (March 2013), pp. 982–989.
- [14] DETTI, A.; RICCI, B.; BLEFARI-MELAZZI, N. Peer-to-peer live adaptive video streaming for information centric cellular networks. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on* (Sept 2013), pp. 3583–3588.
- [15] DRAXLER, M.; KARL, H. Efficiency of on-path and off-path caching strategies in information centric networks. In *Proceedings of the 2012 IEEE International Conference on Green Computing and Communications* (Washington, DC, USA, 2012), GREENCOM '12, IEEE Computer Society, pp. 581–587.
- [16] FOUNDATION, O. N. Sdn architecture overview. Tech. Rep. ONF TR-504, Nov. 2014.
- [17] GHALI, C.; TSUDIK, G.; UZUN, E. Network-layer trust in named-data networking. *SIGCOMM Comput. Commun. Rev.* 44, 5 (Oct. 2014), 12–19.
- [18] GHODSI, A.; SHENKER, S.; KOPONEN, T.; SINGLA, A.; RAGHAVAN, B.; WILCOX, J. Information-centric networking: Seeing the forest for the trees. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks* (New York, NY, USA, 2011), HotNets-X, ACM, pp. 1:1–1:6.
- [19] GRANDL, R.; SU, K.; WESTPHAL, C. On the interaction of adaptive video streaming with content-centric networking. In *Packet Video Workshop (PV), 2013 20th International* (Dec 2013), pp. 1–8.
- [20] GRANGER, S. Digital preservation emulation, 2001.
- [21] GUAN, J.; WANG, X.; XIA, Y.; HUANG, T.; WEI, L. Status-based content sharing mechanism for content-centric network. In *Communication Technology (ICCT), 2012 IEEE 14th International Conference on* (Nov 2012), pp. 357–363.
- [22] HAN, B.; WANG, X.; CHEN, X.; KWON, T. T.; CHOI, Y.; MAU, D. O.; CHEN, M. Ppp: Prefix-based popularity prediction for effective caching in content-centric networking. In *Proceedings of The Ninth International Conference on Future Internet Technologies* (New York, NY, USA, 2014), CFI '14, ACM, pp. 7:1–7:6.
- [23] HAN, B.; WANG, X.; CHOI, N.; KWON, T.; CHOI, Y. Amvs-ndn: Adaptive mobile video streaming and sharing in wireless named data networking. In *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on* (April 2013), pp. 375–380.

- [24] ISHIYAMA, R.; TSUKAMOTO, K.; KOIZUMI, Y.; OHSAKI, H.; HATO, K.; MURAYAMA, J.; IMASE, M. On the effectiveness of diffusive content caching in content-centric networking. In *Information and Telecommunication Technologies (APSITT), 2012 9th Asia-Pacific Symposium on* (Nov 2012), pp. 1–5.
- [25] JACOBSON, V.; SMETTERS, D. K.; BRIGGS, N. H.; PLASS, M. F.; STEWART, P.; THORNTON, J. D.; BRAYNARD, R. L. Voccn: Voice-over content-centric networks. In *Proceedings of the 2009 Workshop on Re-architecting the Internet* (New York, NY, USA, 2009), ReArch '09, ACM, pp. 1–6.
- [26] JACOBSON, V.; SMETTERS, D. K.; THORNTON, J. D.; PLASS, M.; BRIGGS, N.; BRAYNARD, R. Networking named content. *Commun. ACM* 55, 1 (Jan. 2012), 117–124.
- [27] JOKELA, P.; ZAHEMSZKY, A.; ESTEVE ROTHENBERG, C.; ARIANFAR, S.; NIKANDER, P. Lipsin: Line speed publish/subscribe inter-networking. *SIGCOMM Comput. Commun. Rev.* 39, 4 (Aug. 2009), 195–206.
- [28] KAMIYAMA, N. Ccn economics: Can isps profit by introducing ccn? In *Network and Service Management (CNSM), 2014 10th International Conference on* (Nov 2014), pp. 242–247.
- [29] KAZI, A.; BADR, H. Some observations on the performance of ccn-flooding. In *Computing, Networking and Communications (ICNC), 2014 International Conference on* (Feb 2014), pp. 334–340.
- [30] KIM, Y.; AN, J.; LEE, Y.-H. Ccnfr: Fast one-hop re-route in ccn. In *Communications (ICC), 2012 IEEE International Conference on* (June 2012), pp. 5799–5803.
- [31] KIM, Y.; YEOM, I. Performance analysis of in-network caching for content-centric networking. *Comput. Netw.* 57, 13 (Sept. 2013), 2465–2482.
- [32] KULINSKI, D.; BURKE, J. Ndnvideo: Random-access live and pre-recorded streaming using ndn. Tech. Rep. 1, September 2012.
- [33] KULINSKI, D.; BURKE, J.; ZHANG, L. Video streaming over named data networking, July 2013.
- [34] LEDERER, S.; MUELLER, C.; RAINER, B.; TIMMERER, C.; HELLWAGNER, H. Adaptive streaming over content centric networks in mobile networks using multiple links. In *Communications Workshops (ICC), 2013 IEEE International Conference on* (June 2013), pp. 677–681.
- [35] LEDERER, S.; MUELLER, C.; RAINER, B.; TIMMERER, C.; HELLWAGNER, H. An experimental analysis of dynamic adaptive streaming over http in content centric networks. In *Multimedia and Expo (ICME), 2013 IEEE International Conference on* (July 2013), pp. 1–6.
- [36] LEE, J.; LIM, K.; YOO, C. Cache replacement strategies for scalable video streaming in ccn. In *Communications (APCC), 2013 19th Asia-Pacific Conference on* (Aug 2013), pp. 184–189.

- [37] LEE, S.-W.; KIM, D.; KO, Y.-B.; HOON KIM, J.; JANG, M.-W. Cache capacity-aware ccn: Selective caching and cache-aware routing. In *Global Communications Conference (GLOBECOM), 2013 IEEE* (Dec 2013), pp. 2114–2119.
- [38] LIU, Y.; GEURTS, J.; POINT, J.-C.; LEDERER, S.; RAINER, B.; MULLER, C.; TIMMERER, C.; HELLWAGNER, H. Dynamic adaptive streaming over ccn: A caching and overhead analysis. In *Communications (ICC), 2013 IEEE International Conference on* (June 2013), pp. 3629–3633.
- [39] LIU, Y.; GEURTS, J.; RAINER, B.; LEDERER, S.; MUELLER, C.; TIMMERER, C. Dash over ccn: A ccn use-case for a social media based collaborative project. In *CCNx Community Meeting (CCNxConn 2012)* (Sophia Antipolis, sep 2012), G. Carofiglio, Ed., Parc, pp. 1–1.
- [40] LONGZHE HANA, S.-S. K.; IN, H. P. Efficient retransmission mechanism for video streaming over multi-hop wireless content-centric networks. pp. 349–351.
- [41] MAHADEVAN, P.; UZUN, E.; SEVILLA, S.; GARCIA-LUNA-ACEVES, J. Ccn-krs: A key resolution service for ccn. In *Proceedings of the 1st International Conference on Information-centric Networking* (New York, NY, USA, 2014), INC '14, ACM, pp. 97–106.
- [42] MICROSOFT. Live and on-demand video with silverlight and iis smooth streaming, 2010. Disponível em http://download.microsoft.com/download/3/A/4/3A4A066C-6543-4BC1-A8BA-965D0E55F296/Live_and_On_Demand_Video_with_Silverlight_and%20IIS_Smooth_Streaming_FINAL.pdf.
- [43] MONTPETIT, M.-J.; KLYM, N.; MIRLACHER, T. The future of iptv. *Multimedia Tools and Applications* 53, 3 (2011), 519–532.
- [44] ONG, M. D.; CHEN, M.; TALEB, T.; WANG, X.; LEUNG, V. C. Fgpc: Fine-grained popularity-based caching design for content centric networking. In *Proceedings of the 17th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems* (New York, NY, USA, 2014), MSWiM '14, ACM, pp. 295–302.
- [45] OUESLATI, S.; ROBERTS, J.; SBIHI, N. Flow-aware traffic control for a content-centric network. In *INFOCOM, 2012 Proceedings IEEE* (March 2012), pp. 2417–2425.
- [46] PASSARELLA, A. Review: A survey on content-centric technologies for the current internet: Cdn and p2p solutions. *Comput. Commun.* 35, 1 (Jan. 2012), 1–32.
- [47] ROZHNOVA, N.; FDIDA, S. An effective hop-by-hop interest shaping mechanism for ccn communications. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on* (March 2012), pp. 322–327.
- [48] SAUCEZ, D.; GRIECO, L. A.; BARAKAT, C. Aimd and ccn: Past and novel acronyms working together in the future internet. In *Proceedings of the 2012 ACM Workshop on Capacity Sharing* (New York, NY, USA, 2012), CSWS '12, ACM, pp. 21–26.

- [49] SMETTERS, D. K.; JACOBSON, V. Securing network content. Tech. Rep. PARC TR-2009-1, Oct. 2009.
- [50] TIMMERER, C.; WEINBERGER, D.; SMOLE, M.; GRANDL, R.; MUELLER, C.; LEDERER, S. Live transcoding and streaming-as-a-service with mpeg-dash. In *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)* (Los Alamitos, CA, jun 2015), E. Magli, S. Tubaro, and A. Vetro, Eds., IEEE, pp. 1–4.
- [51] WOOD, C.; UZUN, E. Flexible end-to-end content security in ccn. In *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th* (jan 2014), pp. 858–865.
- [52] XU, H.; CHEN, Z.; CHEN, R.; CAO, J. Live streaming with content centric networking. In *Networking and Distributed Computing (ICNDC), 2012 Third International Conference on* (Oct 2012), pp. 1–5.
- [53] YI, C.; AFANASYEV, A.; MOISEENKO, I.; WANG, L.; ZHANG, B.; ZHANG, L. A case for stateful forwarding plane. *Comput. Commun.* 36, 7 (Apr. 2013), 779–791.
- [54] YI, C.; AFANASYEV, A.; WANG, L.; ZHANG, B.; ZHANG, L. Adaptive forwarding in named data networking. *ACM Computer Communication Reviews* 42, 3 (July 2012), 62–67.
- [55] ZHANG, L. AND ESTRIN, D. B. J.; JACOBSON, V.; THORNTON, J. D.; SMETTERS, D. K.; ZHANG, B.; TSUDIK, G.; CLAFFY, K.; KRIOUKOV, D.; MASSEY, D.; PAPADOPOULOS, C.; ABDELZAHER, T.; WANG, L.; CROWLEY, P.; YEH, E. Named data networking (ndn) project. Tech. Rep. PARC TR-2010-3, Oct. 2010.
- [56] ZHANG, F.; ZHANG, Y.; REZNIK, A.; LIU, H.; QIAN, C.; XU, C. A transport protocol for content-centric networking with explicit congestion control. In *Computer Communication and Networks (ICCCN), 2014 23rd International Conference on* (Aug 2014), pp. 1–8.
- [57] ZHANG, Y.; LIU, J.; HUANG, T.; CHEN, J.; LIU, Y. Multi-path interests routing scheme for multi-path data transfer in content centric networking. In *Information and Communications Technology 2013, National Doctoral Academic Forum on* (Aug 2013), pp. 1–6.