



UNIVERSIDADE ESTADUAL DO CEARÁ
CENTRO DE CIÊNCIAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

DIEGO GUIMARÃES DE BARROS

NDN-GAMES: UMA PROPOSTA DE ARQUITETURA HÍBRIDA COM REDES NDN
PARA DISSEMINAÇÃO DE CONTEÚDO MULTIMÍDIA EM JOGOS ONLINE

FORTALEZA – CEARÁ

2015

DIEGO GUIMARÃES DE BARROS

NDN-GAMES: UMA PROPOSTA DE ARQUITETURA HÍBRIDA COM REDES NDN PARA
DISSEMINAÇÃO DE CONTEÚDO MULTIMÍDIA EM JOGOS ONLINE

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Orientador: Prof. Dr. Marcial Porto Fernandez

FORTALEZA – CEARÁ

2015

Dados Internacionais de Catalogação na Publicação

Universidade Estadual do Ceará

Sistema de Bibliotecas

Barros, Diego Guimarães de Barros.

NDN-Games: uma proposta de arquitetura híbrida com redes NDN para disseminação de conteúdo multimídia em jogos online [recurso eletrônico] / Diego Guimarães de Barros Barros. – 2015.

1 CD-ROM: il.; 4 ¾ pol.

CD-ROM contendo o arquivo no formato PDF do trabalho acadêmico com 45 folhas, acondicionado em caixa de DVD Slim (19 x 14 cm x 7 mm).

Dissertação (mestrado acadêmico) – Universidade Estadual do Ceará, Centro de Ciências e Tecnologia, Mestrado Acadêmico em Ciência da Computação, Fortaleza, 2015.

Área de concentração: Redes e Sistemas Distribuídos.
Orientação: Prof. Dr. Marcial Porto Fernandez.

1. Infraestrutura de Redes. 2. Jogos Online. 3. NDN. 4. Internet do Futuro. I. Título.

DIEGO GUIMARÃES DE BARROS

NDN-GAMES: UMA PROPOSTA DE ARQUITETURA HÍBRIDA COM REDES NDN PARA
DISSEMINAÇÃO DE CONTEÚDO MULTIMÍDIA EM JOGOS ONLINE

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Marcial Porto Fernandez (Orientador)
Universidade Estadual do Ceará – UECE

Prof. Dr. André Ribeiro Cardoso
Universidade Estadual do Ceará – UECE

Prof. Dr. Cidcley Teixeira de Souza
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE

RESUMO

O paradigma de redes orientadas a conteúdo é uma abordagem alternativa para redes de computadores, visando como objetivo uma nova arquitetura de comunicação que seja coerente com a natureza dinâmica da Internet atual. Entre esses modelos podemos destacar as redes *Named Data Network* (NDN) que possui uma proposta de recuperação de dados através de nomes de conteúdo, ao invés de endereços de origem e destino. Com essa abordagem, uma rede NDN atinge um ótimo desempenho para a distribuição de conteúdo em larga escala, melhorando o aproveitamento dos recursos da rede. Porém este modelo ainda não apresenta uma eficiência concreta sob aplicações ponto-a-ponto como, e-mail, *Voice over Internet Protocol* (VoIP), jogos online, *stream* de vídeo, músicas, entre outros. Portanto, este trabalho propõe uma arquitetura híbrida de rede, utilizando NDN para a distribuição de conteúdo multimídia em jogos online e rede IP para comunicação de controle. Foi implementado um protótipo em emulação onde foi permitido validar a proposta apresentada, demonstrando como este modelo de rede pode ser utilizado em aplicações web sólidas, bem como, os benefícios dessa abordagem.

Palavras-chave: Infraestrutura de Redes. Internet do Futuro. Jogos Online. NDN.

ABSTRACT

The content-oriented network paradigm is an alternative approach for computer's networks, proposing a new communication architecture compatible with the dynamic nature of the current Internet. Among these models, we can highlight Named Data Network (NDN). Its basic idea is to retrieve data through content names, instead of origin and destination address. This approach allows to achieve good performance to distribute content in large-scale, improving the usage of the network. However, this model is not a consensus on point-to-point applications such as e-mail, VoIP, games and client-server application. This work proposes a hybrid network architecture, using NDN for dissemination of content in online games and point-to-point IP to transmit control messages. A prototype was deployed, by network emulation, demonstrating how NDN networks can be used to improve online game's distribution network.

Keywords: Network infrastructure, Future Internet, Online Games, Named Data Networks

LISTA DE ILUSTRAÇÕES

Figura 1 – Pacote de interesse e Dados.	14
Figura 2 – Exemplo nó NDN.	15
Figura 3 – Fluxo do pacote de interesse.	16
Figura 4 – Tempo de donwload vs. número de clientes.	17
Figura 5 – Arquiteturas de Jogos Online.	19
Figura 6 – Divisão de Conteúdo.	25
Figura 7 – Visão geral NDN-Games.	26
Figura 8 – Arquitetura NDN-Games.	27
Figura 9 – Exemplo NDN-Games.	28
Figura 10 – Arquitetura D-ITG.	31
Figura 11 – Topologias: (a) Tradicional (b) NDN-Games.	33
Figura 12 – Atraso médio em segundos para o fluxo de controle.	34
Figura 13 – Taxa média em Kbit/s para o fluxo de controle.	35
Figura 14 – Tempo médio de download por cliente.	35
Figura 15 – Taxa média de transferência por cliente.	36
Figura 16 – Atraso médio dos pacotes por cliente.	36
Figura 17 – Atraso médio em segundos para o fluxo de controle.	37
Figura 18 – Taxa de Transferência Média em Kbit/s para o fluxo de controle.	38
Figura 19 – Tempo médio de download por cliente.	38
Figura 20 – Taxa média de transferência por cliente.	39
Figura 21 – Atraso médio dos pacotes por cliente.	39
Figura 22 – TPC/IP vs. NDN-Games (Fluxo de Controle).	40
Figura 23 – TPC/IP vs. NDN-Games (Tempo de Download).	40
Figura 24 – TPC/IP vs. NDN-Games (Taxa de Transferência).	41

SUMÁRIO

1	INTRODUÇÃO	9
1.1	MOTIVAÇÃO	10
1.2	OBJETIVOS	11
1.2.1	Objetivo Geral	11
1.2.2	Objetivos Específicos	11
1.3	CONTRIBUIÇÕES DO TRABALHO	11
1.4	ORGANIZAÇÃO DO TRABALHO	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	NAMED DATA NETWORK	13
2.1.1	Pacotes NDN	13
2.1.2	Modelo NDN	13
2.1.3	Fluxo de Interesses e Dados	14
2.1.4	Vantagens NDN	15
2.2	JOGOS ONLINE	17
2.2.1	Arquitetura de rede para jogos Online	18
2.2.2	Peer-to-Peer	19
2.2.3	Cliente e Servidor	19
2.2.4	Híbrido	20
3	TRABALHOS RELACIONADOS	21
3.1	VOZ SOBRE CCN	21
3.2	DOONY BROOK	21
3.3	G-COPSS	22
3.4	MERTS	22
3.5	COMPARTILHAMENTO P2P	22
3.6	RELAÇÕES COM A PROPOSTA	23
4	NDN-GAMES: UMA ARQUITETURA HIBRIDA PARA JOGOS ON- LINE	24
4.1	CONTEÚDO ESTÁTICO E DINÂMICO	24
4.2	ARQUITETURA NDN-GAMES	26
5	AVALIANDO NDN-GAMES	29

5.1	FERRAMENTAS	29
5.1.1	Mininet	29
5.1.2	Simple Router	30
5.1.3	Gerador de Tráfego	30
5.1.4	CCNx e Mini-CCNx	31
5.2	AVALIAÇÃO E RESULTADOS	32
5.2.1	Metodologia	32
5.2.2	Recursos e Medidas	32
5.2.3	Topologia e Enlace	33
5.2.4	Configurações de Fluxo	33
5.3	REDE TRADICIONAL	34
5.4	REDE NDN-GAMES	36
5.5	COMPARATIVO	37
6	CONCLUSÕES E TRABALHOS FUTUROS	42
6.1	TRABALHOS FUTUROS	42
	REFERÊNCIAS	43
	GLOSSÁRIO	45

1 INTRODUÇÃO

A Internet foi projetada para ser uma rede de pesquisa limitada a conectar algumas universidades e ser manuseada por um grupo seleto de usuários técnicos. Porém, hoje boa parte do mundo está conectada à Internet não somente para realizar pesquisas e compartilhar dados, mas também para fornecer uma gama de produtos e serviços como redes sociais, comércio eletrônico, áudio e vídeo sob demanda, jogos online, entre outros. Todavia, para atingir essa complexidade e atender à crescente demanda, a arquitetura original da Internet teve de sofrer uma série de adaptações. Podemos perceber isso através da grande quantidade campos e protocolos de rede existentes atualmente, o que finda por dificultar a implementação e manutenção de novos serviços. Esse processo ficou conhecido como Ossificação da Internet (MOREIRA et al., 2009).

Por conta desse cenário surgiram diversas pesquisas que abordam a reestruturação da Internet. Dentre essas propostas, podemos separar basicamente duas frentes de pesquisa. A primeira defende a reestruturação da Internet em camadas de forma gradual, por outro lado, a segunda também conhecida por "*Clean Slate*" propõe uma reformulação total da arquitetura da Internet, modificando seus princípios mais básicos (FELDMANN, 2007).

Em meio as diversas propostas da linha "*Clean Slate*" podemos destacar as redes orientadas a conteúdo *Named Data Network* (NDN) (JACOBSON et al., 2009b) que se diferencia por propor modificações, não somente em nível de aplicação, mas também em nível de roteamento. Esse modelo trabalha de forma similar ao uso corrente da Internet, através da recuperação de conteúdo pelo nome e não pela sua localização específica. Hoje, de forma geral, os usuários utilizam palavras chaves que descrevem o conteúdo de interesse e como resposta recebem listas de links com possíveis resultados que lhe satisfazem. Dessa maneira fica fácil concordamos que hoje a Internet trabalha em termos de "o que?" e não "onde?", de forma que fica transparente para o usuário a localização específica do conteúdo requisitado.

Entre as vantagens das redes NDN é importante destacar a sua distribuição eficiente de conteúdo. De forma simples este tipo de rede utiliza-se de caches distribuídos que armazenam conteúdos que poderão ser recuperados através de seu nome. Quanto maior o número de usuários interessados em um conteúdo, mais caches serão povoados com o conteúdo, melhorando assim o aproveitamento dos recursos da rede e facilitando o acesso a conteúdos populares.

Para que essa distribuição eficiente seja aproveitada de maneira significativa é de comum acordo que o conteúdo seja também de tamanho e popularidade significativos. De forma simples podemos apontar os arquivos multimídia de áudio, vídeo e imagem como a

grande demanda em distribuidores de conteúdo, sejam eles oficiais ou não. Logo aplicações que envolvam esse tipo de mídia, como *stream* de vídeo e música, jogos em rede ou sob demanda, entre outros, irão se beneficiar com este modelo de rede. Dentre essas aplicações destacamos os jogos online. Basicamente os jogos online atendem a dois requisitos significativos para redes orientadas a conteúdo. Primeiro fornecem um conteúdo que é de múltiplo interesse, impulsionando assim a disseminação dos dados. Segundo, jogos, de forma geral, trabalham com pacote de conteúdo de dados multimídia de tamanho significativo.

Observando a estrutura básica dos conteúdos inerentes a jogos temos: *scripts*, bibliotecas, *plugins*, modelos tridimensionais, texturas, entre outros (BRITO, 2011). Basicamente as texturas e os arquivos de som e vídeo são os componentes que mais influenciam no tamanho final do jogo. Isso é comprovado ao compararmos o tamanho dos jogos do início de 2000, como o *Counter Striker* e *Quake* que variavam entre 700 MB a 1 GB, (FOSTER, 2014) para os jogos atuais como *Battlefield 3* de 2011 que ocupa no mínimo 20GB ou o *Tintam Fall* de 2014 com 48GB (DEBATE, 2014). Grande parte desse aumento de tamanho deve-se a arquivos de textura, som e vídeos em alta qualidade.

De forma realista é pouco provável que a arquitetura da Internet possa ser totalmente reformulada ou que seja implantada uma rede puramente NDN, já que, a mesma ainda possui alguns pontos abertos (JACOBSON et al., 2009a) (LI et al., 2011), em segurança e em aplicações ponto a ponto como, e-mail, *Voice over Internet Protocol* (VoIP), *stream*, entre outros. Porém, assim como as redes P2P, existe a possibilidade da criação de uma rede híbrida *overlay*, funcionando em parceira com o tradicional Internet Protocol (IP), trazendo novas oportunidades de serviços e aplicações.

De acordo com essa perspectiva, a proposta deste trabalho é elaborar e implementar uma arquitetura de rede híbrida voltada para a disseminação de conteúdo para jogos online, usando o melhor dos dois mundos NDN e IP. Com isso pretendemos validar uma aplicação ponto a ponto sólida e popular como, jogos online, utilizando redes NDN. Em seguida, faremos uma comparação de desempenho entre nossa proposta e as arquiteturas tradicionais.

1.1 MOTIVAÇÃO

Uma vez entendido o contexto dessa pesquisa e de acordo com diversos trabalhos, (JACOBSON et al., 2009a), (LI et al., 2011) e (CHEN et al., 2011), ainda não está tão claro o quão provável e bom pode ser o aproveitamento de uma rede NDN em aplicações ponto a

ponto. Por conta disso, muitas pesquisas foram desenvolvidas para a validação deste paradigma, focando em diferentes aplicações como o VoIP, por exemplo. Entretanto, até então, nenhum trabalho havia focado no aproveitamento específico de uma rede NDN em jogos online. Devido a este cenário, é relevante elaborar e validar uma arquitetura de rede que possa aproveitar os benefícios do NDN, ao mesmo tempo que este possa ser implantado em uma infraestrutura já existente, beneficiando aplicações que realmente possam aproveitar todo o potencial das redes orientadas a conteúdo.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

O objetivo geral deste projeto é propor um modelo de infraestrutura de rede para o fornecimento de conteúdo digital para jogos online, utilizando o paradigma das redes orientadas a conteúdo NDN.

1.2.2 Objetivos Específicos

Dentre os objetivos específicos podemos listar:

- a) Definir uma arquitetura padrão: modularizar todo o processo de distribuição de conteúdo em jogos online.
- b) Levantar requisitos: ferramentas necessárias para o suporte de uma aplicação de jogos neste contexto.
- c) Comparar desempenho: avaliar se o NDN-Games pode ser tão bom ou melhor quanto a rede tradicional de jogos.

1.3 CONTRIBUIÇÕES DO TRABALHO

Este trabalho consegue demonstrar o uso do paradigma de rede NDN dentro de uma aplicação ponto a ponto sólida como os jogos online. Esta pesquisa gerou uma publicação no congresso *International Conference on Networks* (ICN 2015) (BARROS; FERNANDEZ, 2015). Neste artigo é apresentado uma visão geral sobre a proposta NDN-Games, definindo sua arquitetura e testes preliminares.

1.4 ORGANIZAÇÃO DO TRABALHO

O restante deste trabalho está organizado da seguinte forma, primeiramente, no Capítulo 2 abordaremos nossa Fundamentação Tórica com os conhecimentos básicos sobre NDN e Jogos Online. Em seguida, apresentamos os trabalhos relacionados, Capítulo 3. No Capítulo 4 apresentamos nossa arquitetura NDN-Games e seu funcionamento básicos sobre conteúdo Estático e Dinâmico. Em seguida, no Capítulo 5 abordaremos a avaliação de nossa proposta através de emulação, construindo uma rede Tradicional (*Legacy*) e uma rede NDN-Games, os resultados são computados e comparados. Por fim, no Capítulo 6 apresentamos nossas Conclusões e trabalhos Futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Esta Seção trata de descrever o funcionamento básico de um nó NDN e seus principais componentes. Em seguida, é abordada uma visão geral sobre arquitetura de rede para jogos online, mostrando assim a compatibilidade de interesses entre essa aplicação e as redes NDN.

2.1 NAMED DATA NETWORK

A ideia básica sobre uma rede orientada a conteúdo é antiga, trabalhos como projeto *Translating Relaying Internet Architecture integrating Active Directories* (TRIAD) em 1999 e o *Data-Oriented Network Architecture* (DONA) em 2006 já sugeriam o uso de nomes de objetos para o roteamento de pacotes (KOPONEN et al., 2007). Em 2009 o grupo PARC publicou sua proposta o CCN *Content-Centric Networking* que em seguida ficou conhecida como NDN *Named Data Network* (JACOBSON et al., 2007). Este modelo se destaca devido a não necessidade de um endereço de origem e de destino como o IP utilizado nas redes tradicionais. Sendo assim uma rede NDN precisa apenas do nome do conteúdo para recuperá-lo. Sua filosofia é simples e teoricamente soluciona diversas questões como: disponibilidade, segurança, dependência de localização entre outras (ZHANG et al., 2010).

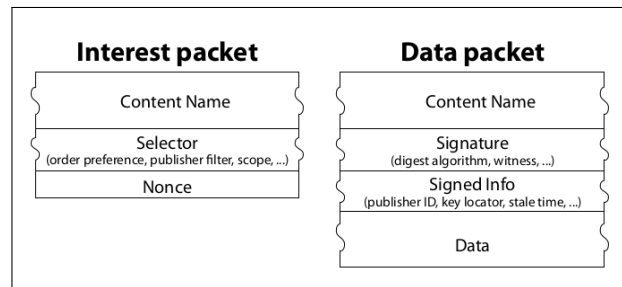
2.1.1 Pacotes NDN

Para um bom entendimento do trabalho é necessário conhecermos como funciona um nó NDN. Porém, antes de definirmos um nó modelo NDN, é importante entendermos que este funciona basicamente com apenas dois tipos de pacotes: um de interesse e outro de dados. O pacote de interesse define o nome do conteúdo desejado, permitindo assim a sua busca. O pacote de dados é o conteúdo que satisfaz o interesse. Tecnicamente isso somente ocorre quando ambos possuem o mesmo nome. O NDN trabalha com o conceito de "face", que é uma referência da origem do pacote de interesse, podendo ser um endereço IP, MAC, proxy, entre outros. Na Figura 1 podemos ver um exemplo desses pacotes.

2.1.2 Modelo NDN

Um nó NDN é composto pelas seguintes entidades: *Content Store* (CS), *Pending Interest Table* (PIT) e *Forward Information Base* (FIB). A CS funciona basicamente como um

Figura 1 – Pacote de interesse e Dados.



Fonte: (JACOBSON et al., 2007)

buffer, porém com políticas de reposição diferentes. Ela possui a propriedade de ser idempotente, ou seja, para diferentes requisições ela poderá retornar o mesmo resultado, por exemplo: um vídeo sobre notícias é interesse para o usuário A também poderá atender ao interesse dos usuários B,C,...

A PIT é essencialmente uma tabela de interesses que ainda não foram atendidos. Quando um interesse é difundido na rede NDN a PIT correspondente em cada nó armazena o nome do interesse e a face que o requisitou, sendo assim quando um conteúdo atende a um determinado interesse este segue o caminho de volta descrito em cada nó NDN, é o que o autor chama de “*bread crumbs*” ou migalha de pão.

Por fim, a FIB é uma base de dados usada para o encaminhamento de interesses a até suas prováveis fontes de conteúdo. É semelhante a IP FIB, porém, podendo usar múltiplos destinos ao invés de um único. Isso se deve ao fato de o NDN não ser limitado pelo spanning tree.

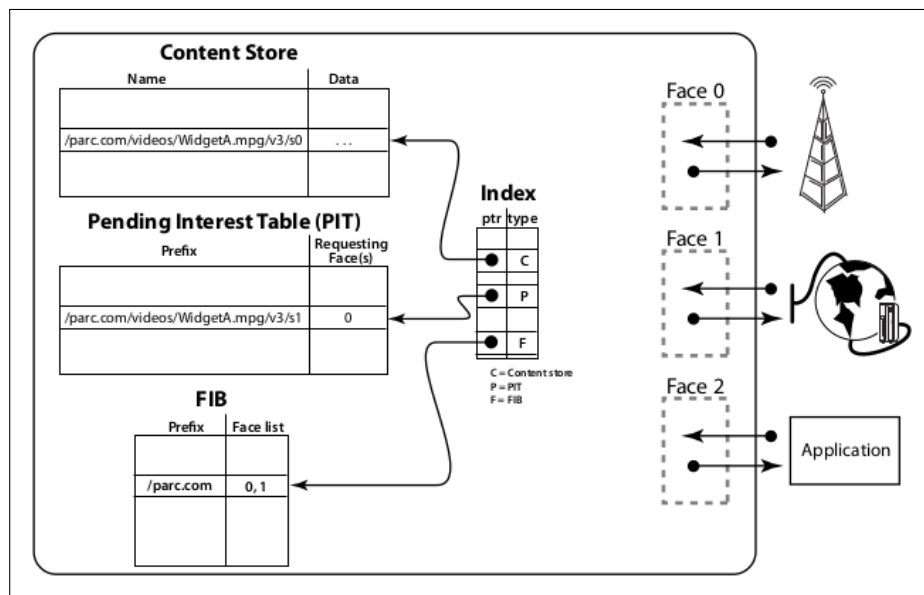
Quando um pacote de Interesse chega de alguma face é feita uma pesquisa *longest-match* em seu nome de conteúdo. Existe uma estrutura índice que serve para definir uma precedência de pesquisa, tendo como preferência primeiramente a CS, em seguida a PIT e por último a FIB. Na Figura 2 podemos ver um exemplo desses pacotes.

2.1.3 Fluxo de Interesses e Dados

Quando um pacote de interesse chega em um nó NDN é feita uma consulta na CS. Se o conteúdo correspondente existir, é enviado então um D-packet com o conteúdo pela face que o requisitou. Caso contrário, é observado se existe algum registro na PIT correspondente ao interesse recém chegado. Se existir, o pacote de interesse é descartado e a face que o requisitou é adicionada na lista de faces deste interesse pendente.

Caso contrário é feita uma consulta na FIB, se existir uma entrada correspondente o

Figura 2 – Exemplo nó NDN.



Fonte: (JACOBSON et al., 2007)

pacote de interesse é encaminhado para a face que em potencial poderá atender a este interesse. Em seguida é feita uma entrada na PIT com o interesse pendente.

Caso não exista nenhuma entrada na FIB o pacote de interesse é descartado, pois não existe nenhum dado de referência e ele não sabe como obter nenhuma informação a respeito.

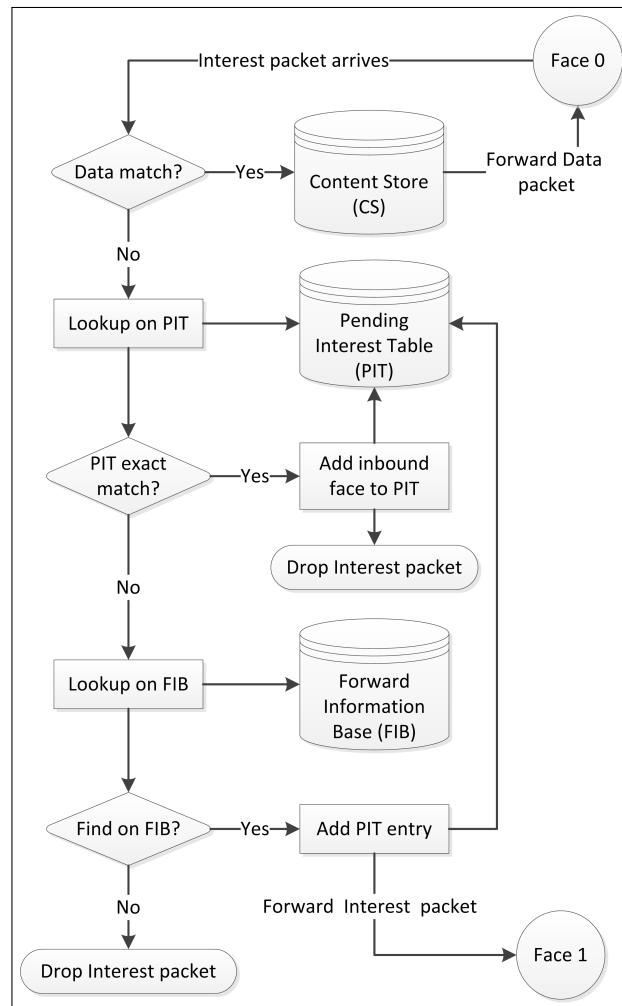
O pacote de dados possui uma mecânica bem simples, pois não precisa ser roteado. Basta apenas seguir o caminho reverso deixado pelo pacote de interesse em cada nó NDN. Esse caminho é traçado através da cadeia de entradas de cada FIB até a origem do pedido.

Quando um pacote de dados chega em nó NDN é feita uma pesquisa pelo nome do conteúdo na CS, se existir uma entrada, o conteúdo está duplicado e então é feito o descarte. Outro caso de descarte é quando não existe uma entrada na PIT correspondente, logo aquele pacote de dados recém chegado não foi requisitado. Caso exista uma entrada na PIT, os dados foram requisitados por um pacote de interesse que passou por este nó. Os dados são autenticados e adicionados a CS. Em seguida é criada uma lista com todas as faces que requisitaram o pacote de dados em questão. Logo o pacote de dados é enviado para cada face da lista.

2.1.4 Vantagens NDN

Diferente do IP o NDN não é limitado por loops na camada de enlace, portanto é possível tomar vantagem do uso de múltiplas interfaces de origem, atendendo as requisições em paralelo. O NDN trabalha apenas de um nó para outro sob a informação, não há a necessidade

Figura 3 – Fluxo do pacote de interesse.



Fonte: Elaborado pelo autor

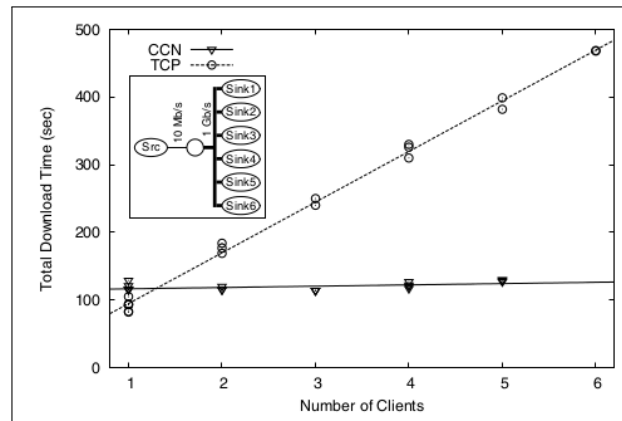
de relacionar um endereço de camada 3, como o IP, com uma identidade de camada 2, como MAC, dessa forma mesmo que a conectividade mude o NDN sempre entrega os dados, assim que for fisicamente possível.

Outra vantagem eminente desse modelo é que cada nó pode acessar informações sobre o resultado de uma requisição de interesse, como o tempo de resposta e alcance, com isso é possível mensurar qual o melhor caminho para atender determinado interesse.

As alterações propostas pelo modelo NDN refletem de forma mais significativa na hora de distribuir o conteúdo. Segundo a avaliação feita por Jacobson (JACOBSON et al., 2007), onde é comparado o desempenho de uma rede TCP/IP e uma rede NDN, para a disseminação de dados com um número crescente de usuários. Basicamente podemos concluir que para um simples cliente o TCP apresenta uma desempenho melhor, porém quando o número de clientes aumenta o tempo total de download do TCP aumenta linearmente e diretamente proporcional ao

número de cliente, enquanto o NDN se mantém um tempo quase constante, assim como visto na Figura 4.

Figura 4 – Tempo de donwload vs. número de clientes.



Fonte: (JACOBSON et al., 2007)

De acordo com o que foi visto o NDN apresenta um modelo de rede que em teoria é simples e que tecnicamente recupera conteúdo somente pelo nome do conteúdo, atingindo a flexibilidade necessária para a natureza dinâmica da Internet. Porém é um desafio relevante aplicar este modelo de forma pura, sendo que ainda existem muitos pontos abertos em segurança e na implementação de aplicações ponto-a-ponto, como VoIP, e-mail, *stream*, entre outros. Logo seria necessário verificar como este modelo de rede poderia ser adaptado a aplicações web já consolidadas.

Dessa forma podemos propor que a melhor forma de utilizar o NDN seja em conjunto com o próprio IP, para a disseminação de conteúdo com tamanho e popularidade significativa. Desta maneira pretendemos reduzir o tempo de download por parte dos usuários, bem como, reduzir os investimentos necessários por parte dos produtores de conteúdo, para alocação e disseminação de vídeo, músicas e jogos.

2.2 JOGOS ONLINE

No ano de 2012 o mercado de vídeo games faturou 68.40 bilhões de dólares, um crescimento de 10.04% quando comparado com o ano de 2007 (CARON, 2008). Segundo dados (THEESA, 2014) 59% dos americanos jogam algum tipo de vídeo game, gastando em 2013 um total de 21.53 bilhões, sendo 15.39 bilhões relacionado a compra de conteúdo.

Para atender as necessidades deste mercado foram criados as distribuidoras digitais,

que possuem o papel de entregar o conteúdo de forma digital sem a necessidade de uma mídia física. Essa modo de distribuição se tornou mais evidente ao longo dos anos 2000 devido ao desenvolvimento das redes de telecomunicação e aumento da largura de banda para usuários da Internet.

Com a intenção de melhorar as vendas várias empresas de jogos desenvolveram sua própria plataforma para a distribuição de conteúdo (WIKIPEDIA, 2014). Com isso surgiram as bibliotecas de conteúdo digital, como a Steam, Origin, Live, Play Station Store (PSN), entre outras. A intenção é oferecer um serviço centralizado de compras e armazenamento de jogos em mídia digital. Além disso é possível comprar conteúdo relacionado aos jogos como filmes, trilhas sonoras, customização, entre outros pacotes de dados. A arquitetura de rede usada por esses serviços para a distribuição de conteúdo é o tradicional cliente servidor (YINANC, 2012) com o auxílio do *Content Delivery Networking* (CDN), no caso da Steam são utilizados servidores Akami (AKAMI, 2015).

Para prover um serviço de distribuição de conteúdo eficiente é necessário um investimento significativo em infraestrutura de rede que será proporcional a demanda de conteúdo. Como podemos ver no dados fornecidos pela Steam, com aproximadamente 8 milhões de clientes, o fluxo de conteúdo disseminado somente nos EUA, por semana, atinge facilmente a casa dos 13.5 PB, representando 21.2% do fluxo global (STEAM, 2014).

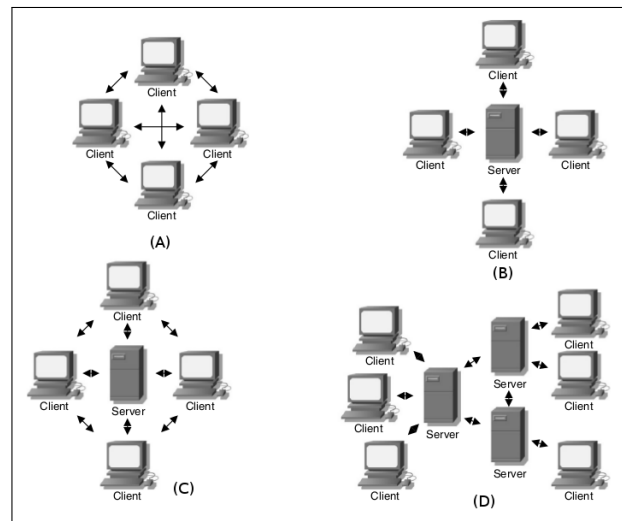
De acordo com os dados apontados fica evidente que as necessidades do mercado de distribuição de jogos podem ser sanadas pelas as redes orientadas a conteúdo, que podem contribuir de forma significativa na distribuição dos jogos, impactando no custo da infraestrutura de rede e melhorando a qualidade do serviço prestado.

2.2.1 Arquitetura de rede para jogos Online

Esta Seção ira apresentar uma visão geral sobre a evolução dos jogos online bem como os ponto mais relevantes para nossa pesquisa.

Os modelo de comunicação para jogos não se diferenciam muito do que já é conhecido em aplicações distribuídas, com o tradicional cliente-servidor e o peer-to-peer. Porém para cada modelo são inferidos pequenas modificações que são relevantes para cada classe de jogos. Podemos ter uma visão geral das arquiteturas na Figura 5.

Figura 5 – Arquiteturas de Jogos Online.



Fonte: (ARMITAGE et al., 2006)

2.2.2 Peer-to-Peer

DOOM em 1993 foi um dos primeiros títulos de *First Personal Shooter* (FPS) com modo *multiplayer* cooperativo com quatro jogadores. O jogo usava o modelo peer-to-peer em LAN sob o protocolo *Internetwork Packet Exchange* (IPX) com transmissão em *Broadcast*. Cada jogador rodava uma instância do jogo e devia trocar mensagens de controle e atualização diretamente com o demais jogadores de forma descentralizada (ARMITAGE et al., 2006).

Os principais desafios desse modelo estão relacionados diretamente ao sincronismo. É de significativa dificuldade assegurar que o jogo seja completamente determinístico, ou seja, cada máquina deve executar o mesmo conjunto de instruções no mesmo intervalo de tempo, independente das eventualidades da rede. Para garantir isto é necessário esperar que cada ação executada por um jogador seja atualizada e processada pelos demais antes do turno ser simulado e o estado do jogo ser modificado. Este modelo ainda é utilizado por jogo do tipo *Real Time Strategy* (RTS), devido a natureza desse gênero (GAFFERONGAMES, cited July 2008).

2.2.3 Cliente e Servidor

Com a intenção de superar os problemas das redes peer-to-peer em 1996 foi disponibilizados o jogo Quake (SANGLARD, 2012) utilizando a arquitetura cliente-servidor. Agora temos um modelo centralizado de comunicação, o cliente envia suas informações de entrada, teclado e mouse, apenas para o servidor, este recebe as informações inerente ao jogador e pro-

cessa a atualização do estado de jogo. O cliente recebe como resposta o seu estado atualizado. A máquina cliente possui apenas uma aproximação do estado real do jogo, funcionando, de forma parecida a um terminal burro. Com isso não há a necessidade de garantir o determinismo em cada máquina, pois o estado real do jogo acontece somente no servidor. Com isso a *Quality of Experience* (QoE) e de um jogo esta relacionado apenas com a qualidade de comunicação entre o cliente e o servidor. Este é modelo mais adotado pelo mercado de jogos online (GAFFERONGAMES, cited July 2008).

Uma variação do modelo cliente-servidor é o server-pool onde temos uma rede de servidores remotos (SMED et al., 2002). Um cliente se conecta a um servidor local, este se conecta a outro servidor remoto. Temos então uma rede de servidores ligado a uma sub-rede de clientes. Esse modelo visa amenizar a carga e no servidores locais e aumenta a qualidade de serviço da rede.

2.2.4 Híbrido

Por fim temos uma arquitetura híbrida entre P2P e cliente-servidor. Existe um servidor que continua desempenhando seu papel tradicional, sendo utilizado para processar os dados do estado do jogo, através das mensagens enviadas pelo cliente. Porém os cliente são habilitados a trocarem informações entre si, que não são importantes para o processamento do estado do jogo, por exemplo, chat para troca de mensagens de texto ou comunicação por VoIP (ARMITAGE et al., 2006).

3 TRABALHOS RELACIONADOS

Nesta seção iremos abordar os trabalhos relacionados, mostrando um resumo com as principais idéias que ajudaram a compor nossa arquitetura NDN-Games.

3.1 VOZ SOBRE CCN

Como visto no trabalho de Jacobson sobre a implementação de comunicação de voz sobre redes CCN (JACOBSON et al., 2009a), existe uma significativa importância em validar aplicações ponto-a-ponto, como e-mail, *stream*, *Voice over Internet Protocol* (VoIP), entre outras para redes orientadas a conteúdos. Nesse artigo o autor adapta uma aplicação VoIP para uma rede de conteúdo. A ideia se baseia em usar um sistema de publicação sob demanda que serve como ponto de contato para o serviço, possibilitando o início de uma sessão. Devido ao uso de nomes em rede NDN ao invés de endereços IP's é introduzido o conceito de nomes construíveis, onde é possível construir nomes de conteúdos desejáveis sem ter visto o conteúdo previamente. Assim através de um algoritmo determinístico tanto consumidor quanto produtor podem chegar em um mesmo nome através de informações disponíveis para ambos.

3.2 DOONY BROOK

O uso de arquiteturas alternativas podem melhorar o desempenho de rede de um determinado jogo, bem como, otimizar sua escalabilidade. Donny Brook (BHARAMBE et al., 2008) é uma arquitetura para jogos baseada em redes peer-to-peer implementado sob o *Quake 3*. Para melhorar o uso da banda é utilizado um grupo reduzido de objetos de interesses, como consequência, temos uma redução significativa das mensagens de atualização. Outro ponto forte dessa proposta é o uso de um sistema *multicast* de mensagens, permitindo o uso de múltiplas fontes de atualização, sensíveis ao tempo de resposta. Outro benefício é o balanceamento de carga, onde um nó com menos recursos pode ser auxiliado por outros nós em melhor estado. Esse trabalho é importante para nossa pesquisa pois demonstra uma implementação válida e significativo sob o *Quake 3* com objetivos parecidos que é o uso de uma arquitetura alternativa da cliente-servidor com o ganho de desempenho e aproveitamento da rede.

3.3 G-COPSS

Outro trabalho que se destaca por usar redes NDN é o *Game Content Publisher/-Subscriber System* (G-COPSS) (CHEN et al., 2012) uma plataforma de jogos distribuída que utiliza as vantagens de rede orientada a conteúdo e adapta o COPSS (CHEN et al., 2011) para prover escalabilidade, característica desejadas em ambientes de jogos. Utilizando a topologia da rede para oferecer um eficiente e rápido sistema de disseminação de conteúdo, o G-COPSS usa um descritor hierárquico de conteúdo e também um *framework* de disseminação baseado em publicação por demanda. O usuário expressa o interesse em CD's (*Content Describe*), por exemplo: /esportes/futebol. Os publicadores de conteúdo enviam anúncios relacionados a um CD quando novas parte do conteúdo chegam. Os CDs são organizados hierarquicamente. Usuários de uma nível superior podem receber anúncios de um nível diferente (a baixo) ex: /esportes recebem /esportes/futebol ou /esportes/natação. NDN requer uma nova engine de encaminhamento. Os roteadores são equipados com uma nova tabela de assinatura STs (*Subscription Table*). As STs mantêm uma base de CDs com informações do assinantes, funcionando de maneira distribuída e agregada, assim como um IP *multicast*.

3.4 MERTS

Mais um trabalho que aborda o tema de aplicações fim-a-fim em tempo para redes NDN é o More Efficient Real-time Traffic Support (MERTS) (LI et al., 2011). Nessa pesquisa é proposto um esquema que suporte de maneira eficiente o tráfego de tempo real em redes NDN. Basicamente é feita a classificação do tráfego em tempo-real e não-tempo-real. Com a intenção de reduzir o número de mensagens de requisição repetidas é proposto um novo modo chamado one-request-n-packets.

3.5 COMPARTILHAMENTO P2P

O compartilhamento de dados através de redes P2P é uma das principais formas de disseminação de conteúdo, sendo responsável por uma grande fatia do tráfego da internet (KUROSE, 2005). Logo, é interessante destacarmos a significativa semelhança das duas arquiteturas NDN e P2P, principalmente, quando observamos o protocolo Gnutella (RIPEANU, 2001). Um cliente que executa o Gnutella é uma par dentro de uma rede lógica, também chamada de rede de sobre posição, formada por outros clientes Gnutella. Quando um cliente necessita requisitar um

conteúdo, ele envia uma mensagem Gnutella *Query* similar ao pacote de Interesse do NDN, pois essa mensagem contém a palavra chave sobre o conteúdo desejado. A Gnutella *Query* é enviada para todos os nós vizinhos, por inundação, até que um nó possa atender a requisição ou que o número limite de saltos seja esgotado. Quando um par pode atender a requisição é enviado uma mensagem de resposta *QueryHit* que apenas segue o caminho de volta, assim como, o pacote de Dados NDN, não precisa ser roteado. É interessante ressaltar que alguns distribuidores de jogos independentes, utilizam o P2P como uma das opções para o Download de conteúdo, porém quando comparado com os grandes serviços disponíveis no mercado, ainda é uma opção pouco adotada pelos meios oficiais de distribuição de jogos.

3.6 RELAÇÕES COM A PROPOSTA

Cada trabalho apresentado neste capítulo contribui para a construção da ideia sobre o NDN-Games. Primeiramente, o trabalho sobre VoCCN deixa claro a necessidade de se comprovar a viabilidade da arquitetura NDN para aplicações ponto-a-ponto. Segundo, tanto o Doony Brook quanto o G-COPSS, apresentam a ideia sobre conjunto de interesse reduzido, onde uma parte do jogo ou um pacote de mensagens é de interesse comum de uma parte do grupo de jogadores e não do grupo inteiro. Adaptando essa ideia com a proposta do MERTS, de classificar tráfego, temos nossa arquitetura que divide o tráfego entre dados (texturas, mapas, vídeos) e controle (mensagens de atualização, autenticação) e aloca cada tráfego na rede que melhor atende suas necessidades.

4 NDN-GAMES: UMA ARQUITETURA HÍBRIDA PARA JOGOS ONLINE

Esta Seção abordar as principais ideias e métodos utilizados para se alcançar os objetivos desta pesquisa. Primeiramente, descrevendo uma visão geral sobre nossa proposta de arquitetura de jogos online usando o paradigma de rede orientada a conteúdo.

Como visto na Seção 2.2 o modelo cliente-servidor é o mais utilizado pelo mercado de jogos. Porém problemas de escalabilidade e ponto central de falha são inerentes a este tipo de arquitetura (TANENBAUM; STEEN, 2007). Por exemplo, quando ocorrer o lançamento de um novo jogo ou *Downloadable Content* (DLC) é comum que a carga nos servidores de conteúdo aumente além do esperado, ocasionando falhas e até a interrupção do serviço (APPLETON, 2014). Para minimizar esse risco são feitos investimentos em infraestrutura de rede, como aumentar a quantidade servidores, alocação de máquinas virtuais, armazenamento em nuvem, entre outros.

Observando esse cenário nossa proposta visa minimizar essa problemática, utilizando um modelo de rede híbrida. Nossa ideia se baseia que um servidor local pode trabalhar com um número limitado de clientes. Porém se o conteúdo for compartilhado entre os clientes, através de uma camada rede que trabalhe com NDN, acreditamos que será possível atender a uma demanda maior com a mesma infraestrutura. Neste caso um publicador usaria um servidor local que povoaria os primeiros caches com o seu conteúdo, em seguida cada cliente que baixasse esse conteúdo ajudaria a disseminá-lo, dessa forma, quanto maior a demanda maior a disponibilidade. Logo temos uma arquitetura escalável sem um ponto central de falha.

4.1 CONTEÚDO ESTÁTICO E DINÂMICO

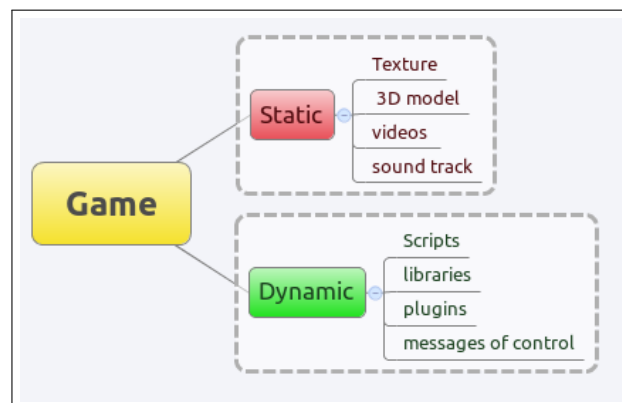
Para entendermos melhor esta proposta é necessário abstrairmos a estrutura de um jogo em dois tipos de conteúdo, estático e dinâmico. O pacote multimídia de um jogo é composto basicamente por arquivos relacionados a ambientação e caracterização do jogo, como as texturas que servem para a construção de cenários, personagens e objetos 3D. Arquivos de vídeos e efeitos sonoros completam esse pacote. Essas estruturas representam a carga mais significativa no tamanho final de um jogo e não variam ao longo de uma partida, todo o processo de armazenamento e renderização é feito localmente. Portanto, este tipo de conteúdo é caracterizado como estático, podendo ser compartilhado entre os clientes de forma independente.

Por outro lado a parte lógica, como arquivos de instalação, *scripts*, bibliotecas,

plugins e mensagens de controle e atualização, podem variar no decorrer de uma partida e ainda podem sofrer variações de acordo com o hardware ou SO do cliente. Logo definimos esse conteúdo como dinâmico.

Dessa forma nossa proposta visa aprimorar a disseminação do conteúdo para jogos online, dividindo esses tipos de conteúdo em dois fluxos distintos de dados que serão trabalhados em camadas diferentes de nossa arquitetura.

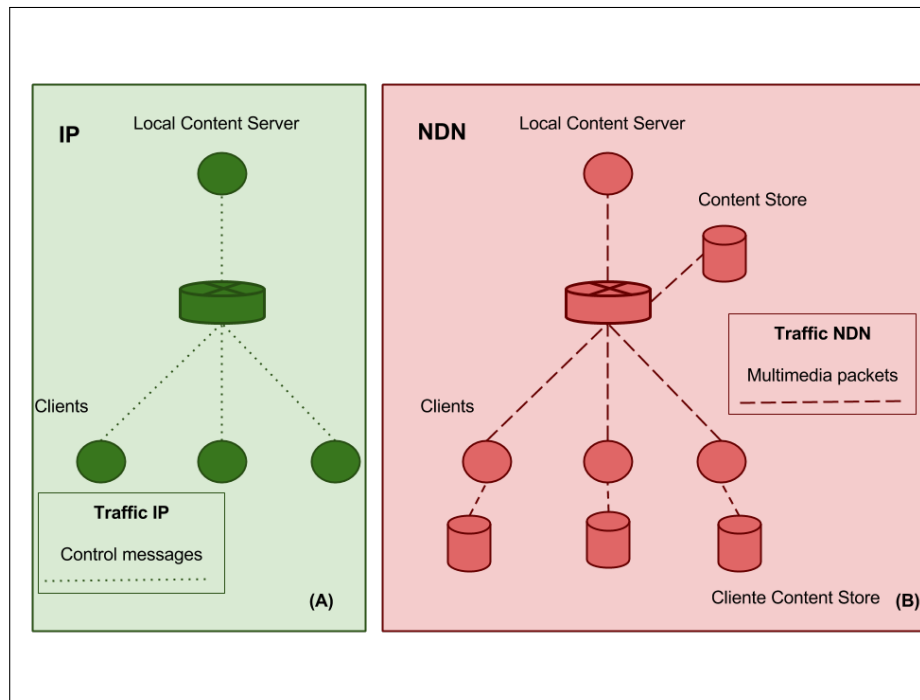
Figura 6 – Divisão de Conteúdo.



Fonte: Criado pelo Autor

O conteúdo dinâmico possui considerável sensibilidade a latência da rede (CHEN et al., 2006), necessitando do melhor tempo de resposta possível. Logo podemos propor que a melhor forma de disponibilizarmos este tipo de conteúdo é utilizando os protocolos TCP/UDP de uma rede IP tradicional no modelo cliente-servidor. Por outro lado os pacotes multimídia possuem necessidades diferentes. Como este tipo de conteúdo possui impacto maior na carga da rede, devido a sua natureza robusta, acreditamos que a melhor estratégia é dividir essa responsabilidade entre os clientes. Com isso os cliente de uma mesma distribuidora de conteúdo estão habilitados a armazenar e compartilhar os dados de pacote multimídia dos jogos, aliviando a carga nos servidores locais. Em uma visão geral de nossa arquitetura temos uma rede tradicional cliente-servidor utilizando TCP/IP sobreposta por uma rede *overlay* NDN. Basicamente dividimos o tráfego de conteúdo, estático e dinâmico, para a rede que melhor atende as necessidades de cada um. Podemos ter uma visão geral na Figura 7

Figura 7 – Visão geral NDN-Games.



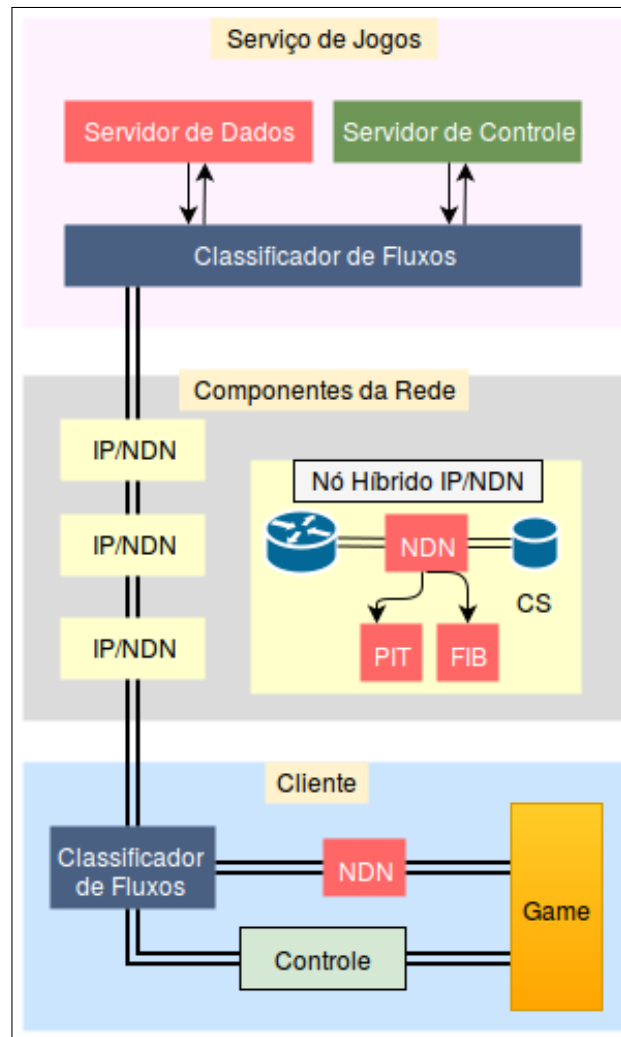
Fonte: Elaborado pelo autor

4.2 ARQUITETURA NDN-GAMES

Em uma abstração mais detalhada desta proposta Figura 8, temos os seguintes módulos: servidor de dados e controle, classificador de fluxo, nós híbridos (NDN/IP) e cliente. O serviço de jogos online possui na verdade o servidor de dados que recebe o fluxo de dados estático formado por pacotes de Interesse/Dados, em paralelo temos o servidor de controle que recebe o fluxo de dados dinâmico dos pacotes TCP/IP. O classificador de fluxo faz a separação dos pacotes de interesse e pacotes de rede TCP/IP tradicionais, encaminhando para o ip e porta do servidor correspondente. Este pacotes que chegam são oriundos da rede composta por nós híbridos, habilitados tanto para o encaminhamento IP quanto NDN. Quando um pacote chega a interface desses nós híbridos, existe uma função que diferencia entre pacotes IP e pacotes NDN, encaminhando para seus respectivos módulos de roteamento. No caso dos pacotes NDN, temos uma aplicação, rodando em *background*, que trata os pacotes. Basicamente, com essas configurações o nó híbrido será capaz de armazenar dados em cache e implementará busca por nome de conteúdo. Uma vantagem desse modelo é que o cliente também podem rodar o modulo NDN e funcionar como um nó Híbrido (NDN/IP) atendendo as requisições e aumentando a disponibilidade de conteúdo na rede.

Para uma melhor compreensão podemos acompanhar o exemplo na Figura 9 o

Figura 8 – Arquitetura NDN-Games.

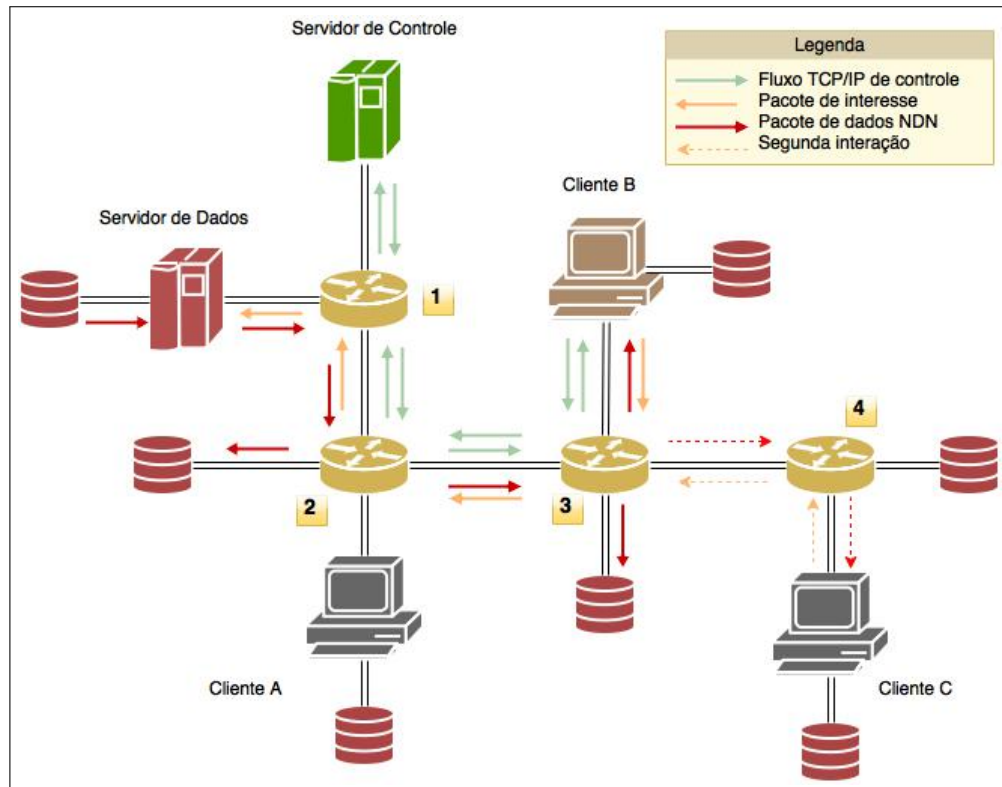


Fonte: Elaborado pelo autor

cliente B realiza uma autenticação no servidor de controle, tráfego em verde, através da rede IP tradicional. Em seguida B envia um pacote de interesse, destacado em vermelho, com o nome do pacote estático do jogo desejado. O pacote de interesse chega ao roteador 3 e então é realizada uma busca no cache pelo nome do conteúdo desejado. Como este pacote de dados ainda não existe neste nó, a face pela qual o pacote de interesse chegou é adicionada a PIT e o pacote de interesse encaminhado ao próximo nó. Neste exemplo, esse processo se repete nos roteadores 2 e 1 até chegar ao servidor de dados, que por sua vez, possui o conteúdo estático do jogo e o envia por meio dos pacotes de dados. Estes refazem o caminho de volta, passando pelos roteadores 1,2 e 3, povoando suas base dados até chegar ao cliente B. Em uma possível segunda interação, caso o cliente A ou C desejem o mesmo jogo o processo de download será otimizado, pois ocorreu o povoamento dos caches na rede e agora tanto os roteadores 2, 3 e o cliente B possuem estes

dados e podem atender essa requisição. Adotando esta estratégia de divisão de fluxo espera-se reduzir o fluxo da rede no servidor de dados, aumentar a disponibilidade do conteúdo e diminuir o tempo de download.

Figura 9 – Exemplo NDN-Games.



Fonte: Elaborado pelo autor

5 AVALIANDO NDN-GAMES

Esta Seção tem como objetivo validar a proposta NDN-Game através de emulação, apresentando as ferramentas necessárias para recriar uma rede tradicional de jogos. Em seguida mostramos como é possível adaptar essa rede para o modelo NDN.

5.1 FERRAMENTAS

O principal objetivo nesta etapa é encontrar uma maneira adequada de por em prática a arquitetura sugerida neste trabalho. Para isso é necessário criar uma rede com topologia e tráfego similares a de uma rede real de jogos online. Muitas ferramentas como *Testbeds*, Simuladores e Emuladores foram avaliadas e o perfil que mais se encaixa nas necessidades desta pesquisa é o de Emuladores, pois apresentam baixo custo, alto realismo e alta/média escalabilidade (CABRAL et al., 2013). A vantagem de se trabalhar com emulação é que a pilha de protocolos, o tráfego de pacotes e os recursos de hardware utilizados são reais e este fator torna nosso ambiente de teste mais consistente e os resultados mais confiáveis.

5.1.1 Mininet

Dentre as ferramentas de emulação de rede podemos destacar o Mininet que foi inicialmente desenvolvido para o estudo e ensino do protocolo *Open Flow* (MCKEOWN et al., 2008) e hoje é reconhecido como uma poderosa ferramenta de virtualização leve em contêiner que permite a criação rápida de topologias de rede. Contêineres são grupos de processos, e no Mininet esses processos possuem uma visão independente dos recursos do sistema, como CPU, memória, HD, ID de processos. Com essa estratégia se ganha muito desempenho, flexibilidade e escalabilidade quando comparado com as virtualizações baseadas em *Hypervisors* como, *Virtualbox* e *VMware*. O Mininet trabalha com *scripts* em *Python* que define as configurações de topologia e enlace da rede. Com isso é possível limitar o consumo de recursos que cada nó poderá usar como CPU, memória, disco e também configurar as variáveis de enlace como largura de banda, taxa de perdas e atraso. Outro atrativo é a possibilidade de rodar qualquer aplicação que esteja instalada na máquina hospedeira, já que o *Kernel* é compartilhado. Para mais detalhes dessa ferramenta consulte (TEAM, 2015).

5.1.2 Simple Router

O Mininet foi desenvolvido para trabalhar com o *Open Flow* que é um protocolo de rede que trabalha com a divisão de fluxos. Uma rede tradicional possui tráfegos de Dados e Controle passando pelo mesmo dispositivo *switch* ou roteador. Uma rede *Open Flow* possui um controlador, um servidor tradicional, que toma as decisões de roteamento e controla o fluxo da rede. Esse controlador se comunica com um *Switch Open Flow* através do protocolo *Open Flow*, dessa forma os equipamentos do núcleo de uma rede *Open Flow* podem atingir até a camada 3 (transporte). Portanto é necessário encontrar um módulo de roteamento que trabalhe até a camada 2 (rede) no Mininet.

Para realizar essa tarefa encontramos uma implementação feita em C, chamada de SR (*Simple Router*), que se comunica com controlador da rede *OpenFlow*. O SR escuta em um porta padrão e através de uma tabela de rotas estáticas processa pacotes como um roteador tradicional *layer 2*. É necessário um arquivo de configuração dos IPs de cada nó da rede, de uma tabela de rotas para cada roteador e de uma *script* em *Python* que crie uma topologia no Mininet.

5.1.3 Gerador de Tráfego

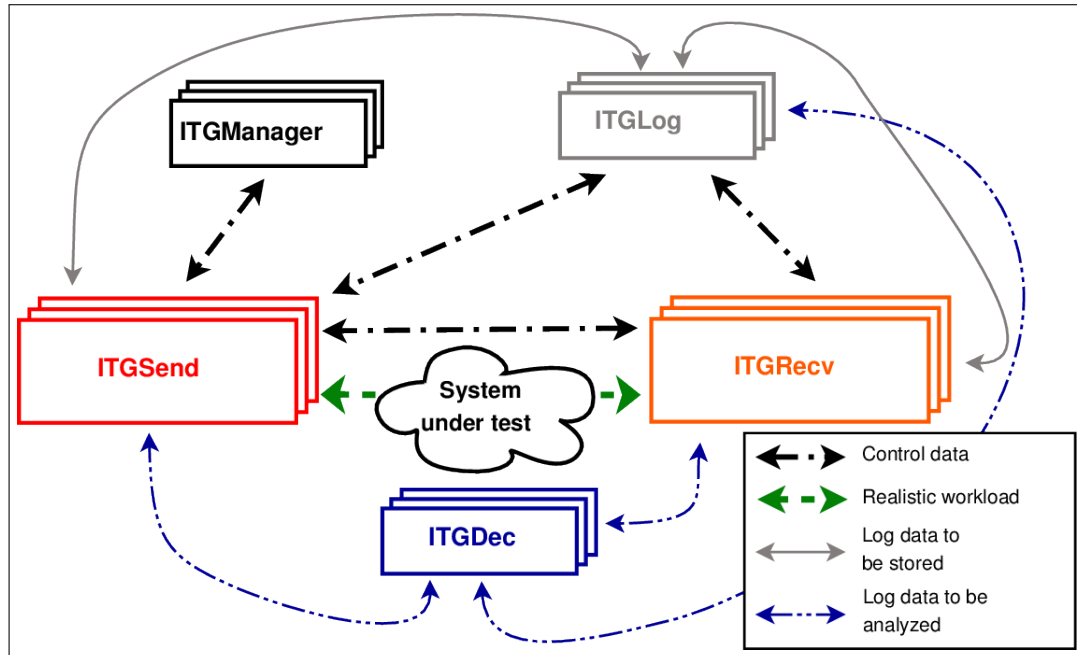
Então, com a combinação do Mininet e do módulo de roteamento SR podemos emular uma rede tradicional *layer 2*. Agora é preciso criar um cenário de tráfego similar a de uma rede de jogos online. Para atender essa tarefa usamos um gerador de tráfego D-ITG (*Distributed Internet Traffic Generator*) (AVALLONE et al., 2004).

Essa plataforma é capaz de gerar tráfego de pacotes IPV4/IPV6 de forma configurável, seguindo modelos estocásticos para o tamanho e tempo de partida do pacote. Por conta dessas características o D-ITG é capaz de retratar diferentes tipos de aplicações da Internet, bem como, VoIP, Telnet, *streams* e jogos online. Além de gerar os pacotes o D-ITG também fornece módulos que permitem a análise de métricas de desempenho da rede.

O D-ITG possui três módulos que são de nosso interesse, o ITGSend, o ITGRecv e o ITGDec. O tráfego pode ser configurado através de linha de comando ou script, onde o ITGSend recebe os parâmetros como, ip de origem e destino, protocolo de transporte (TCP/UDP), tamanho do pacote, duração do fluxo entre outros. Com base nessas características o ITGSend cria o fluxo de dados desejado que será recebido pelo ITGRecv. Cada módulo, ITGSend e ITGRecv, gera um arquivo de log do tráfego que é armazenado pelo ITGLog e poderá ser analisado pelo ITGDec

mostrando métricas de cada fluxo como, taxa média de transmissão, atraso e perda de pacotes. É possível criar múltiplos fluxos para um ou mais ITGRecv. Gerenciando todos esses módulos temos o ITGManager, uma visão geral dessa interação pode ser visto na Figura 10.

Figura 10 – Arquitetura D-ITG.



Fonte: (AVALLONE et al., 2004)

5.1.4 CCNx e Mini-CCNx

O CCNx é a implementação de referência baseada no modelo NDN, esta ferramenta disponibiliza bibliotecas e recursos necessários para a construção e execução de um ambiente experimental de estudo e pesquisa do protocolo NDN (PARC, 2014). Seu funcionamento é baseado na execução de *daemons*, inicialmente o **ccnd** que fica escutando em uma porta padrão esperando por pacotes de Interesse/Dados. O encaminhamento é feito através de um arquivo de configuração (*ccn.conf*) que é similar a uma tabela de rotas, porém no lugar do IP de destino temos um prefixo referente ao nome do conteúdo, por exemplo, *ccn:/uece/ndngames*. Fora os componentes necessários para a execução de um nó NDN, o CCNx, conta com ferramentas para a análise do estado da rede e processamento de pacotes, como o *ccnping* e *ccndump*, que serão muito úteis no decorrer da avaliação da rede NDN-Games.

O trabalho (CABRAL et al., 2013) disponibiliza uma ferramenta de prototipagem rápida para redes NDN o Mini-CCNx que na realidade é o Mininet habilitado para o encami-

nhamento de pacotes NDN. Com isto é possível criar uma rede NDN-Games em condições idênticas a uma rede tradicional de jogos, porém com o adicional uso do protocolo NDN. O Mini-ccnx trabalha com apenas um arquivo simples de configuração, onde é determinado desde as configurações de topologia e enlace até a tabela de rotas de cada nó.

5.2 AVALIAÇÃO E RESULTADOS

Esta Seção detalha os métodos usados para executar os experimentos de emulação das redes Tradicional e NDN-Games, em seguida avaliamos os resultados de cada modelo, e por fim fazemos um comparativos dos pontos mais significantes de cada rede. É interessante destacar que por conta de problemas de compatibilidade entre Mininet, *OpenFlow* e Miniccnx, o ambiente de testes somente se demosntrou estável utilizando as versões, de cada compoentes, descritas no decorrer desta Seção.

5.2.1 Metodologia

A metodologia de avaliação consiste basicamente em emular uma rede de jogos Tradicional e NDN-Games, usando as ferramentas Mininet, *Simple Router* e Mini-ccnx descritas anteriormente. Para cada rede serão gerados dois fluxos Controle e Dados, utilizando o D-ITG. Para cada fluxo de cada rede serão avaliados os seguintes aspectos: (1) atraso médio dos pacotes, (2) taxa média de transferência e, especificamente para o fluxo de dados, (3) tempo médio de download.

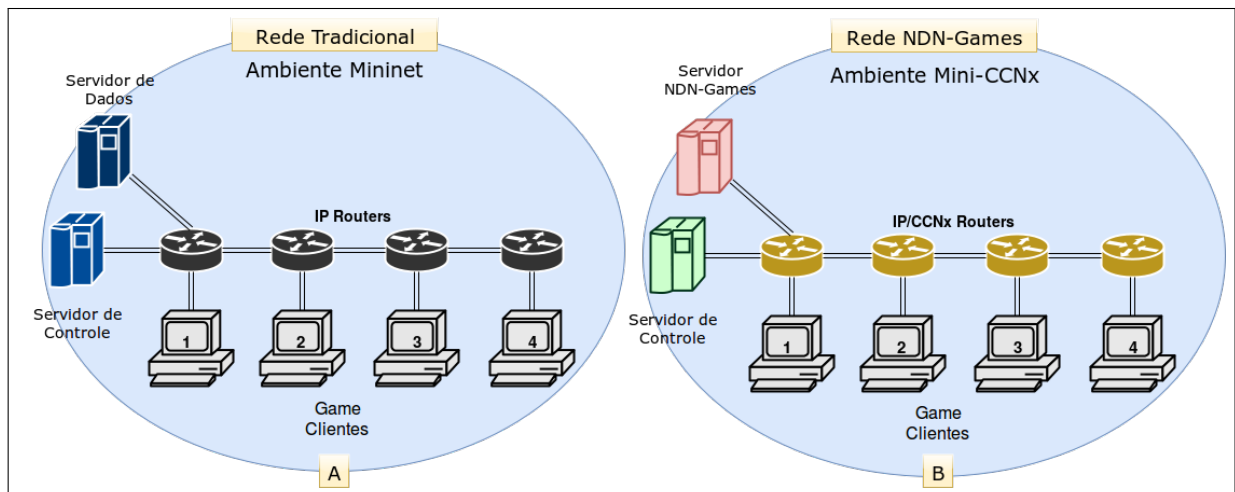
5.2.2 Recursos e Medidas

O ambiente de teste foi criado utilizando o *Virtualbox* versão 4.3.26, através de um servidor Intel Core i7 3770 3.4ghz e 16 GB RAM. As medições foram feitas através dos arquivos de log criados pelo próprio D-ITG. Os dados coletados foram trados estatisticamente e apresentam resultados dentro de um intervalo de confiança de aproximadamente 95%. Durante os teste o consumo de CPU e memória foi monitorado e controlado dentro de uma margem de segurança de 80%, garantido que os resultados não sofram alterações devido a limitações de hardware.

5.2.3 Topologia e Enlace

Para avaliarmos nossa proposta foram elaborados dois contextos básicos representados na Figura 11: (a) uma rede tradicional com roteadores IP e (b) uma rede NDN-Games com roteadores híbridos IP/CCNx. Em ambas as rede temos a mesma topologia: dois servidores, Controle e Dados, ligados a roteadores em série. Cada roteador conecta um cliente. Através desse cenário base construímos variações em que se muda o número de clientes, cada variação multiplica-se o número de cliente por dois, ou seja, inicia-se com 1 cliente, depois 2,4 e 8.

Figura 11 – Topologias: (a) Tradicional (b) NDN-Games.



Fonte: Elaborado pelo autor

No que se trata das configurações de enlace temos: 0% de perda de pacotes em todos os enlaces, atraso de 5 ms entre os roteadores e 10 ms entre os clientes e roteadores, largura de banda de 300 Mbit/s para servidor/roteador, roteador/roteador e de 3 Mbit/s para cliente/roteador. O *script* em *Python* que cria e configura topologia e enlace está em anexo, foi utilizado a versão 2.0.0 do Mininet no Ubuntu 12.10.

5.2.4 Configurações de Fluxo

Para o fluxo de controle configuramos o D-ITG versão 2.8.1 em nível de aplicação, permitindo gerar um fluxo de rede de jogos, baseado no game *Counter Strike*, em modo ativo, isso significa que o tráfego gerado é equivalente ao de uma jogador ativo em uma partida online. A duração do tráfego foi estabelecida dentro de um tempo fixo de 10 min. Este fluxo por padrão só funciona com UDP. As configurações do fluxo de controle serão iguais em ambas as rede

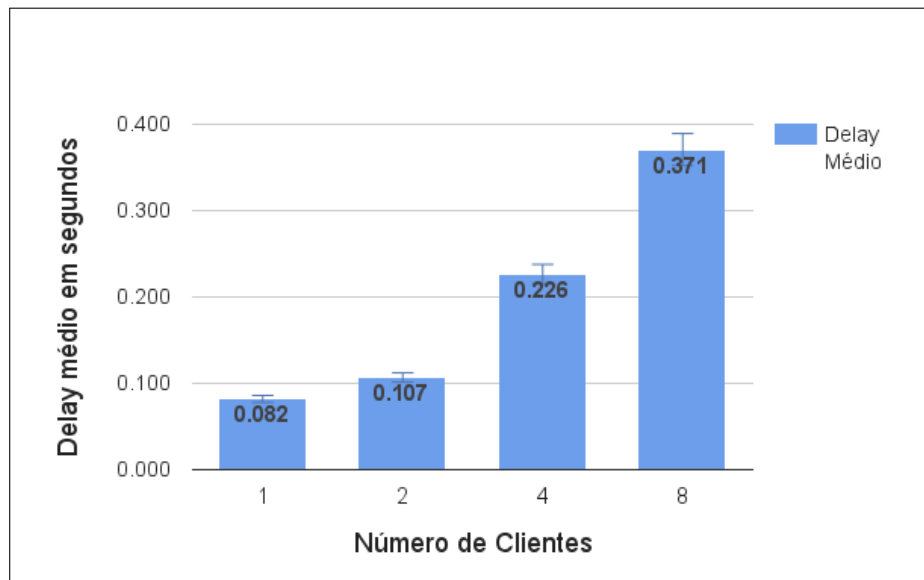
tradicional e NDN-Games. A configuração do fluxo de dados é detalhada nas seções seguintes.

5.3 REDE TRADICIONAL

Emulamos uma rede com dois fluxos distintos, controle e dados, onde o fluxo de dados é gerado da seguinte forma: usa-se o protocolo TCP, com pacotes de tamanho fixo 512 Bytes, taxa constante de 1000 pacotes/s. Um pacote de dados para download equivalente a 10000 Kbytes. O tamanho desses pacotes de dados equivale a proporção da rede, onde um pacote maior ocasionaria em maior tempo de emulação, menor escalabilidade e menor desempenho, como poderemos constatar mais adiante. Dessa forma podemos avaliar os seguintes aspectos para cada fluxo: atraso médio em segundos, a taxa média de dados em Kibt/s e o tempo de download em segundos.

Analisando, inicialmente o fluxo de controle nas Figuras 12 e 13, temos que o atraso na rede aumenta conforme o número de clientes cresce, porém a taxa de transferência se manteve estável se considerarmos a margem de erro.

Figura 12 – Atraso médio em segundos para o fluxo de controle.

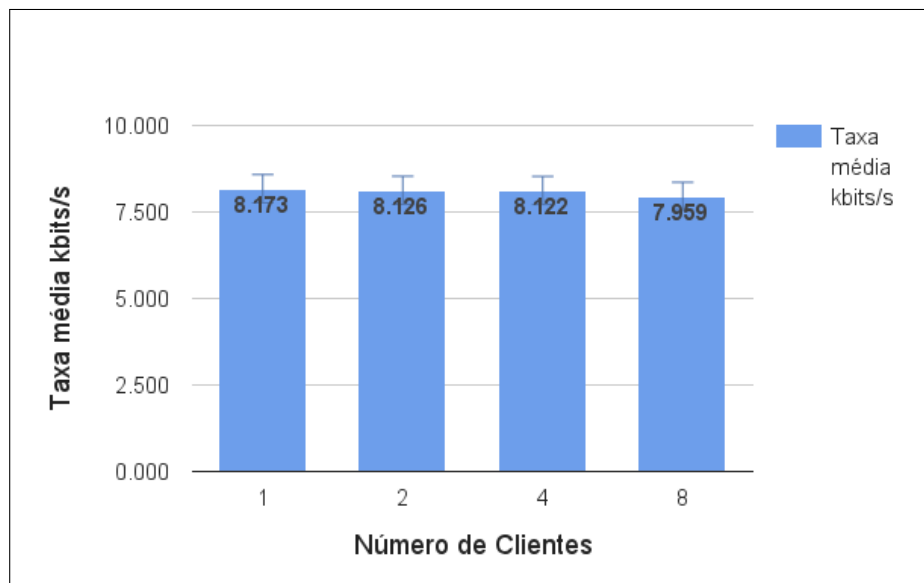


Fonte: Elaborado pelo autor

Para o fluxo de dados podemos observar o tempo de download em segundos, a taxa média de transferência em Kbit/s e o atraso médio dos pacotes em segundos, respectivamente nas Figuras 14, 15 e 16.

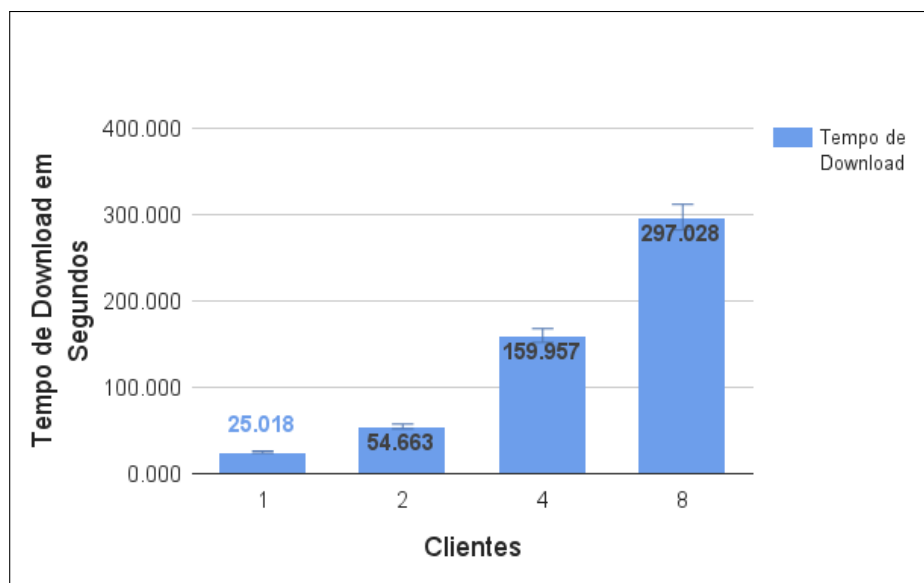
Com base nesses resultados percebemos que o acréscimo de clientes resulta di-

Figura 13 – Taxa média em Kbit/s para o fluxo de controle.



Fonte: Elaborado pelo autor

Figura 14 – Tempo médio de download por cliente.

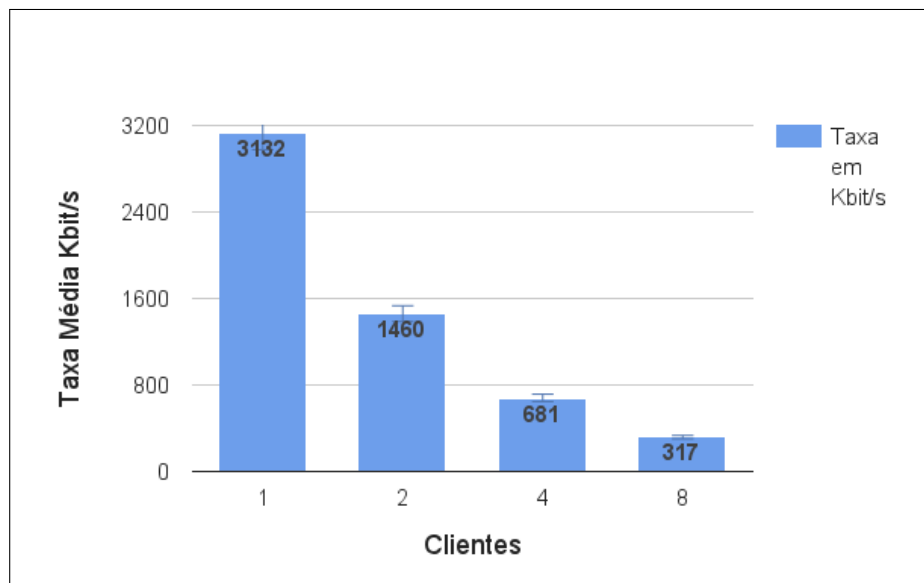


Fonte: Elaborado pelo autor

retamente no aumento do tempo total para o download do pacote de dados e diminuição do aproveitamento da largura de banda. A latência da rede aumenta significativamente também.

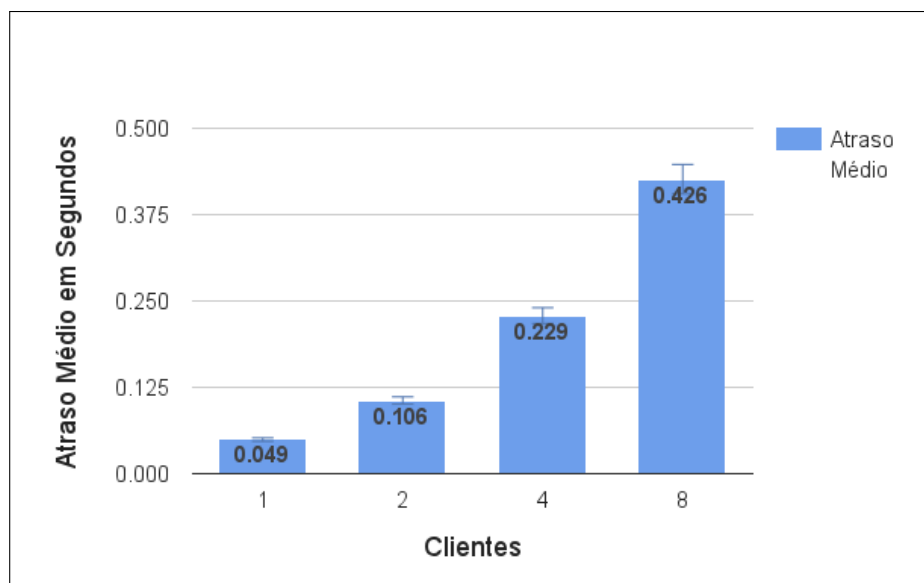
Uma vez que avaliamos esses dados iniciais é interessante deixar claro que para os pacote de controle a latência da rede é vital para um bom *Quality of Experience* (QoE) em jogos online porém, para os pacotes de dados o tempo de download e o aproveitamento da banda são mais significativos e em todos os cenários essas variáveis de rede foram afetadas com o aumento do número de clientes.

Figura 15 – Taxa média de transferência por cliente.



Fonte: Elaborado pelo autor

Figura 16 – Atraso médio dos pacotes por cliente.



Fonte: Elaborado pelo autor

5.4 REDE NDN-GAMES

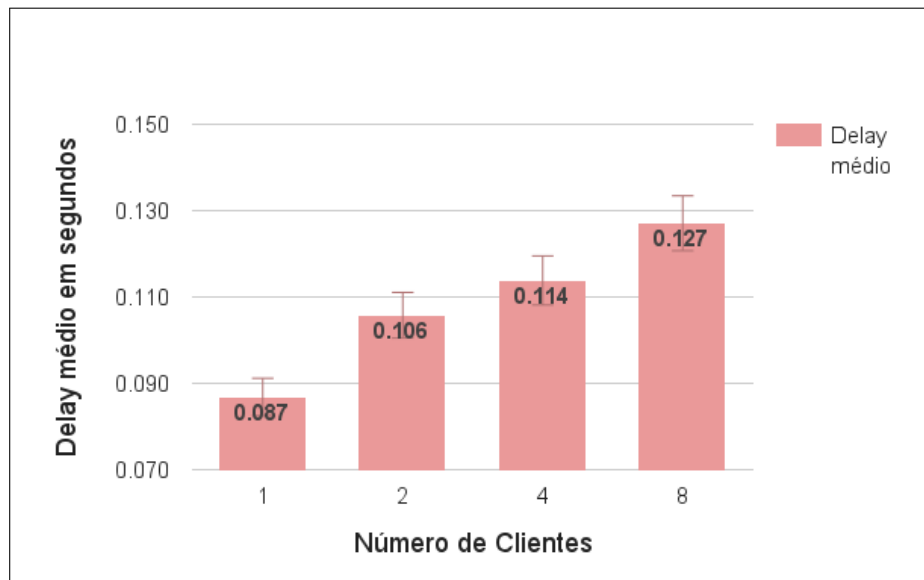
Nesta etapa do trabalho iremos recriar a rede de jogos online, porém usaremos um módulo de roteamento de pacotes diferente o Mini-ccnx. Com essa plataforma será possível emular a mesma topologia com a vantagem de poder trabalhar com pacotes de Interesse e Dados NDN. Outra vantagem é a possibilidade de usar ferramentas do modelo NDN como: *ccnping*, para a análise de conectividade e latência da rede, o *ccngetfile/ccnputfile* para o envio e

requisição de pacotes e *ccndump* para inspeção de pacotes. Em geral as informações sobre atraso e processamento de pacote serão retiradas dos logs do *daemon ccnd* que roda em cada nó NDN.

Basicamente o fluxo de dados será trabalhado com um arquivo de interesse com o nome teste, seu tamanho será igual ao do cenário anterior 10000 kbytes. Já para o fluxo de controle continuamos usando o D-ITG nas mesmas configurações da Seção anterior.

Como podemos avaliar os resultados do fluxo de controle nas Figuras 17 e 18 respectivamente, temos um aumento leve na latência e a taxa de transferência estável conforme o número de clientes cresce.

Figura 17 – Atraso médio em segundos para o fluxo de controle.



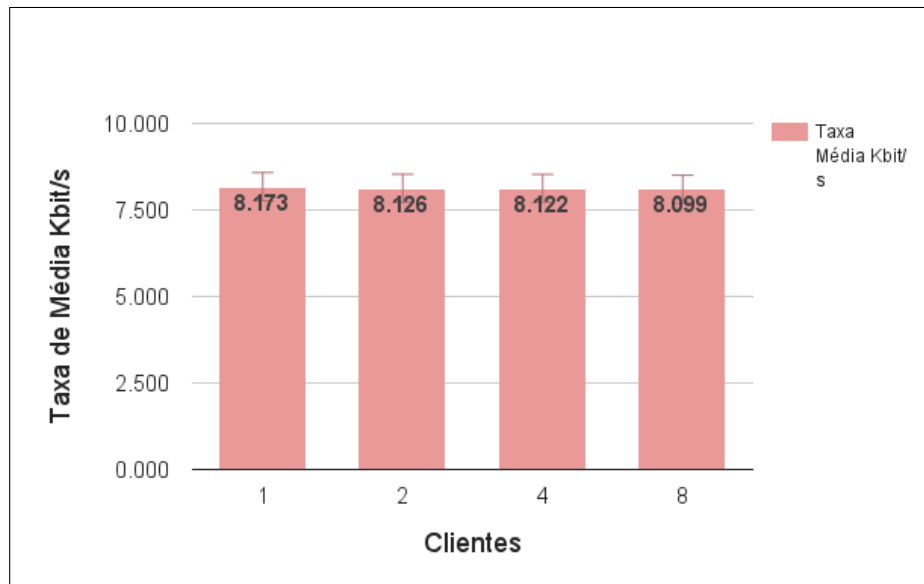
Fonte: Elaborado pelo autor

O ponto chave deste cenário é a avaliação do fluxo de dados NDN-Games, como podemos ver nas Figuras 19, 20 e 21, o aproveitamento da largura de banda é significativamente maior, resultando em tempo de download e latência da rede que não aumentam de maneira proporcional ao número de clientes.

5.5 COMPARATIVO

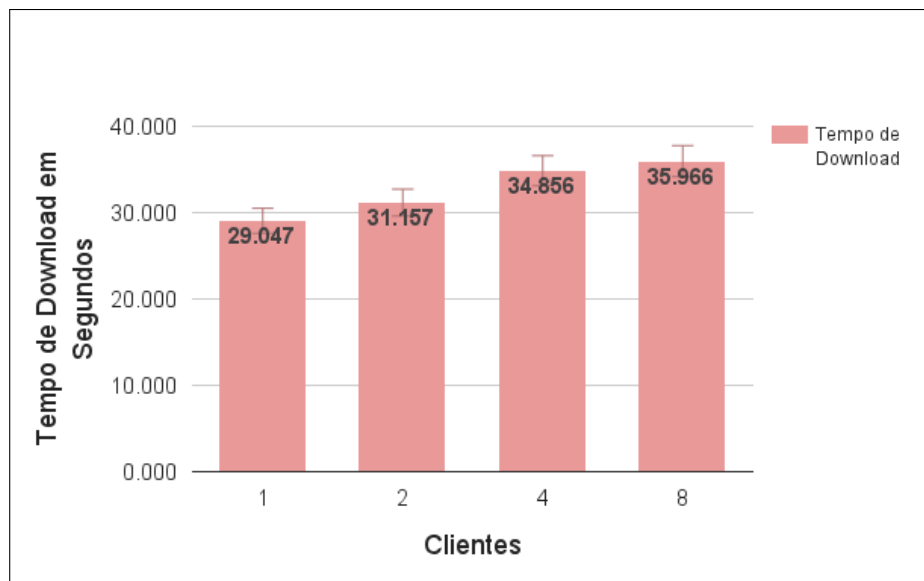
Agora podemos fazer um comparativo final dos resultados sobre, o atraso médio do fluxo de controle, tempo de download e taxa de transmissão do fluxo de dados, nas Figuras 22, 23 e 24. Antes de iniciarmos nossa análise sobre os gráficos a seguir, é interessante destacar que para o caso de apenas um cliente o modelo Tradicional mostra-se mais eficiente do que o

Figura 18 – Taxa de Transferência Média em Kbit/s para o fluxo de controle.



Fonte: Elaborado pelo autor

Figura 19 – Tempo médio de download por cliente.

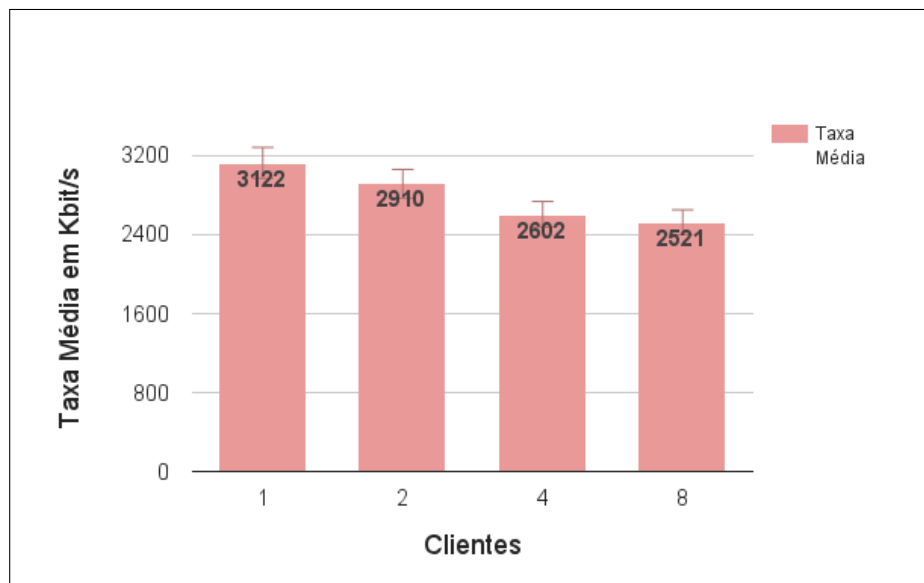


Fonte: Elaborado pelo autor

NDN-Games. Isto ocorre por conta do *overhead* gerado pelo acréscimo de mais uma pilha de protocolo. Este resultado era esperado, como visto no trabalho de Voz sobre CCN (JACOBSON et al., 2009a), o modelo TCP/IP é maduro e bastante otimizado, logo é normal que ele apresente um desempenho melhor, porém quando o número de clientes aumenta o *overhead* se torna menos perceptível e o ganho de desempenho do NDN-Games é significativamente melhor.

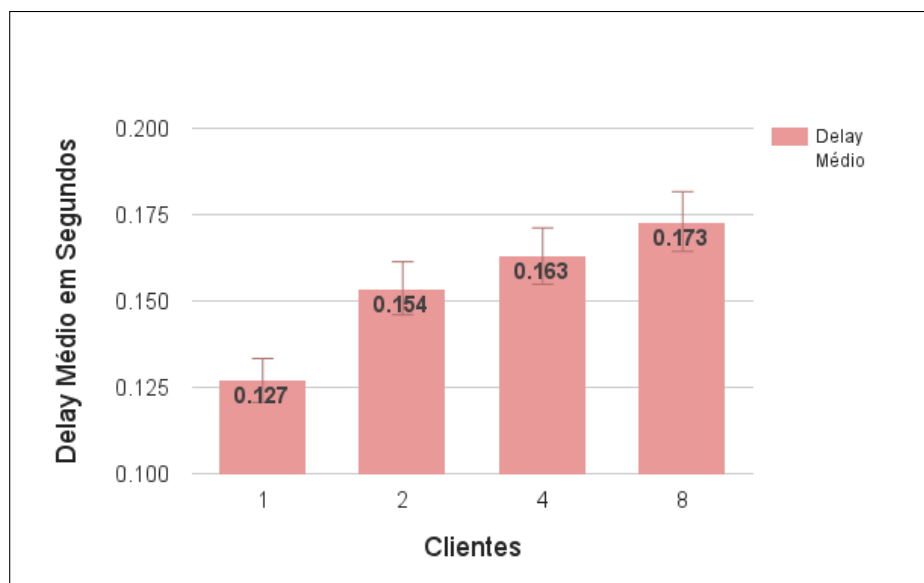
Agora analisando cada aspecto da rede temos, no que se refere a latência do tráfego de controle, temos um empate técnico em cenários com até 2 clientes, porém a partir de 4 ou

Figura 20 – Taxa média de transferência por cliente.



Fonte: Elaborado pelo autor

Figura 21 – Atraso médio dos pacotes por cliente.

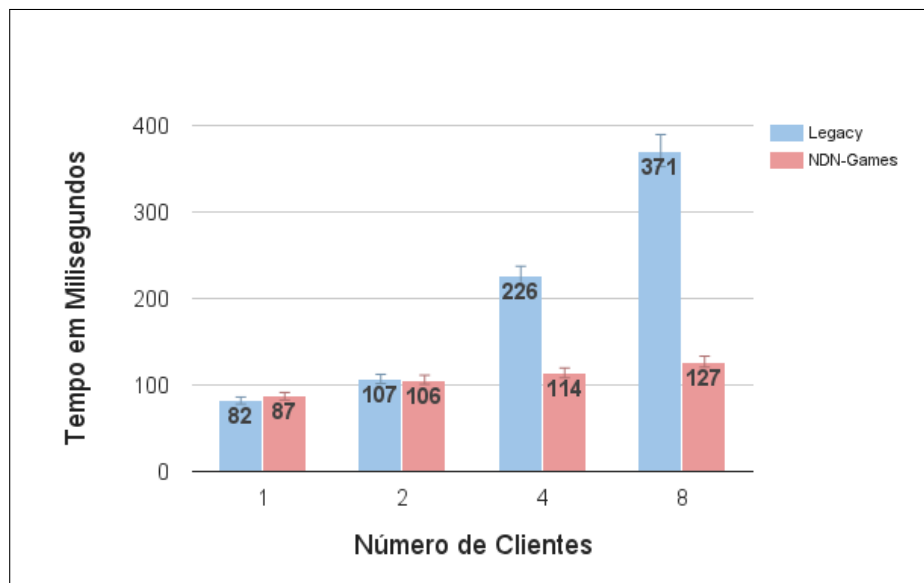


Fonte: Elaborado pelo autor

mais clientes a latência é significativamente menor na rede NDN-Games. Isto ocorre devido a redução da carga sobre o servidor de conteúdo Estático, uma vez que os clientes também estão habilitados a atender as requisições desse tipo de conteúdo, o fluxo médio de pacotes ao longo do tempo é menor, ocasionado em uma disponibilidade maior do enlace e menor tempo de resposta.

No que se trata sobre o fluxo de dados os pontos mais relevantes são sobre o de tempo de download e aproveitamento de banda e nesses aspectos a rede NDN-Games mostrou

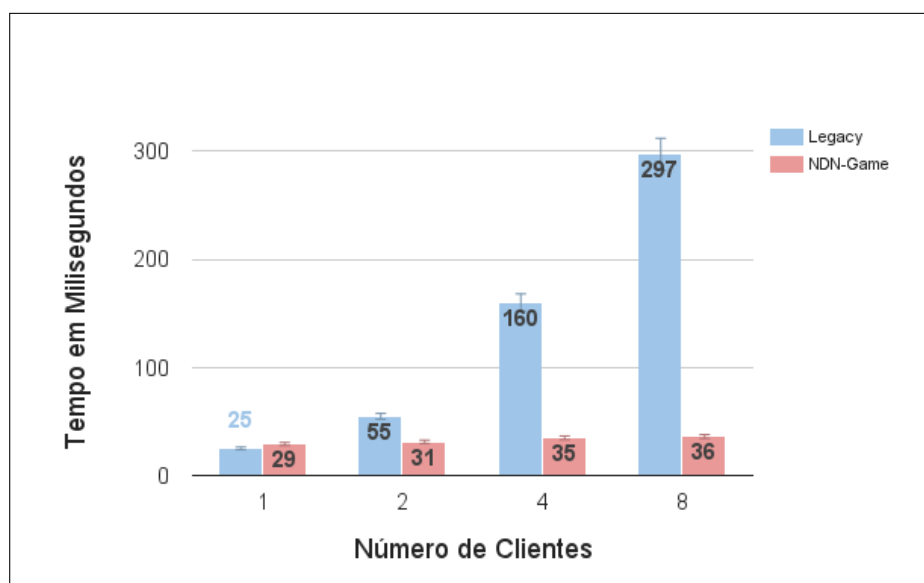
Figura 22 – TPC/IP vs. NDN-Games (Fluxo de Controle).



Fonte: Elaborado pelo autor

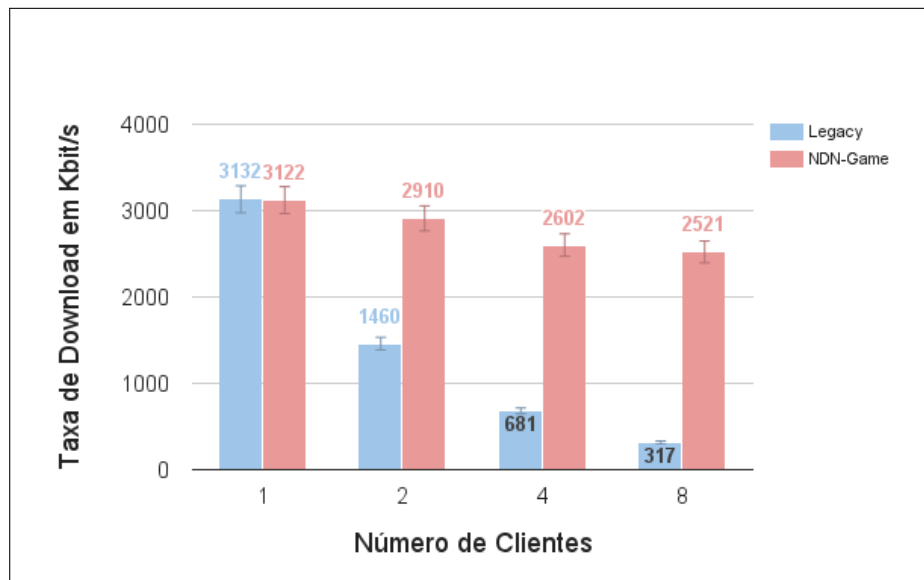
um desempenho significativamente melhor. Com uma larga vantagem, para o pior caso com 8 clientes, a rede NDN apresenta um aproveitamento de 78% da banda disponível contra apenas 9% do modelo tradicional, resultando em um tempo muito próximo do melhor caso de apenas 1 cliente. Porém era esperado que o uso da banda fosse um pouco melhor em ambos os casos o que necessita de uma atenção especial para trabalhos futuros com topologias mais povoadas.

Figura 23 – TPC/IP vs. NDN-Games (Tempo de Download).



Fonte: Elaborado pelo autor

Figura 24 – TPC/IP vs. NDN-Games (Taxa de Transferência).



Fonte: Elaborado pelo autor

6 CONCLUSÕES E TRABALHOS FUTUROS

O paradigma de rede NDN apresenta-se como uma ótima solução para a distribuição de conteúdo em larga escala, porém seu uso para aplicações ponto-a-ponto ainda não é um consenso e muitos aspectos técnicos devem ser tratados antes de uma real implantação. Tendo em vista esse cenário este trabalho propõe uma abordagem mista onde a rede tradicional TCP/IP pode coexistir com o modelo NDN dentro de uma arquitetura híbrida voltada para jogos online e sua disseminação de conteúdo. Deixamos as mensagens de controle ponto-a-ponto trafegarem com o encaminhamento IP enquanto o conteúdo multimídia é distribuído através de um rede *overlay* NDN.

Construímos um modelo de rede de jogos online tradicional e em seguida um modelo híbrido com nós IP/NDN. Os primeiros testes mostraram uma significativa melhora do aproveitamento da largura de banda no enlace dos clientes, resultando em um menor tempo de download do pacote de dados. Munidos dessas informações podemos concluir que existe uma viabilidade na adoção do NDN para a disseminação de conteúdo multimídia para jogos online e que sua contribuição é bastante significativa para um melhor aproveitamento dos recursos da rede.

6.1 TRABALHOS FUTUROS

Inicialmente como trabalhos futuros, é interessante avaliarmos nossa proposta em outros cenários, com maiores variações na topologias e com uma densidade maior de cliente e servidores. Em seguida pretendemos avaliar o impacto da pilha de dois protocolos IP e NDN sobre um único *switch*. Outra questão importante que deve ser trabalhada é a segurança que provavelmente pode ser tratada através do fluxo de controle.

REFERÊNCIAS

- AKAMI. **Akami site oficial**. 2015. <<https://www.xcdn.exceda.com>>. Acessado em outubro de 2015.
- APPLETON, R. **Titanfall's PC servers are overloaded**. 2014. <<http://www.gamerheadlines.com/2014/03/titanfalls-pc-servers-are-overloaded-real-fix-for-xbox-live-coming-later-tonight/>>. Acessado em setembro de 2014.
- ARMITAGE, G.; CLAYPOOL, M.; BRANCH, P. **Networking and online games: understanding and engineering multiplayer Internet games**. [S.l.]: John Wiley & Sons, 2006.
- AVALLONE, S.; GUADAGNO, S.; EMMA, D.; VENTRE, G. et al. D-itg distributed internet traffic generator. In: IEEE. **Quantitative Evaluation of Systems, 2004. QEST 2004. Proceedings. First International Conference on the**. [S.l.], 2004. p. 316–317.
- BARROS, D. G.; FERNANDEZ, M. P. Ndn game: A ndn-based architecture for online games. **ICN 2015**, p. 77, 2015.
- BHARAMBE, A.; DOUCEUR, J. R.; LORCH, J. R.; MOSCIBRODA, T.; PANG, J.; SESHAN, S.; ZHUANG, X. Donnybrook: enabling large-scale, high-speed, peer-to-peer games. **ACM SIGCOMM Computer Communication Review**, ACM, v. 38, n. 4, p. 389–400, 2008.
- BRITO, A. Blender 3d: Jogos e animações interativas. **São Paulo: Novatec**, 2011.
- CABRAL, C.; ROTHENBERG, C. E.; MAGALHÃES, M. F. Mini-ccnx: fast prototyping for named data networking. In: ACM. **Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking**. [S.l.], 2013. p. 33–34.
- CARON, F. **Gaming expected to be a 68 billion business by 2012**. 2008. <<http://www.arstechnica.com/gaming>>. Acessado em 30 de agosto de 2014.
- CHEN, J.; ARUMAITHURAI, M.; FU, X.; RAMAKRISHNAN, K. G-copss: a content centric communication infrastructure for gaming applications. In: IEEE. **Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on**. [S.l.], 2012. p. 355–365.
- CHEN, J.; ARUMAITHURAI, M.; JIAO, L.; FU, X.; RAMAKRISHNAN, K. Copss: An efficient content oriented publish/subscribe system. In: IEEE. **Architectures for Networking and Communications Systems (ANCS), 2011 Seventh ACM/IEEE Symposium on**. [S.l.], 2011. p. 99–110.
- CHEN, K.-T.; HUANG, P.; LEI, C.-L. How sensitive are online gamers to network quality? **Communications of the ACM**, ACM, v. 49, n. 11, p. 34–38, 2006.
- DEBATE, G. **Titanfall System Requirements**. 2014. <<http://www.game-debate.com/games>>. Acessado em 30 de agosto de 2014.
- FELDMANN, A. Internet clean-slate design: what and why? **ACM SIGCOMM Computer Communication Review**, ACM, v. 37, n. 3, p. 59–64, 2007.
- FOSTER, D. **Counter-Striker Requirements**. 2014. <http://www.ehow.com/list_6665806_counter_strike-1_5-requirements.html>. Acessado em junho de 2014.

GAFFERONGAMES. **What every programmer needs to know about game networking**. cited July 2008. <<http://gafferongames.com/networking-for-game-programmers/what-every-programmer-needs-to-know-about-game-networking/>>. Acessado em 30 de agosto de 2014.

JACOBSON, V.; MOSKO, M.; SMETTERS, D.; GARCIA-LUNA-ACEVES, J. Content-centric networking. **Whitepaper, Palo Alto Research Center**, p. 2–4, 2007.

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: ACM. **Proceedings of the 2009 workshop on Re-architecting the internet**. [S.l.], 2009. p. 1–6.

JACOBSON, V.; SMETTERS, D. K.; THORNTON, J. D.; PLASS, M. F.; BRIGGS, N. H.; BRAYNARD, R. L. Networking named content. In: ACM. **Proceedings of the 5th international conference on Emerging networking experiments and technologies**. [S.l.], 2009. p. 1–12.

KOPONEN, T.; CHAWLA, M.; CHUN, B.-G.; ERMOLINSKIY, A.; KIM, K. H.; SHENKER, S.; STOICA, I. A data-oriented (and beyond) network architecture. **ACM SIGCOMM Computer Communication Review**, ACM, v. 37, n. 4, p. 181–192, 2007.

KUROSE, J. F. **Computer Networking: A Top-Down Approach Featuring the Internet, 3/E**. [S.l.]: Pearson Education India, 2005.

LI, H.; LI, Y.; LIN, T.; ZHAO, Z.; TANG, H.; ZHANG, X. Merts: A more efficient real-time traffic support scheme for content centric networking. In: IEEE. **Computer Sciences and Convergence Information Technology (ICCIT), 2011 6th International Conference on**. [S.l.], 2011. p. 528–533.

MCKEOWN, N.; ANDERSON, T.; BALAKRISHNAN, H.; PARULKAR, G.; PETERSON, L.; REXFORD, J.; SHENKER, S.; TURNER, J. Openflow: enabling innovation in campus networks. **ACM SIGCOMM Computer Communication Review**, ACM, v. 38, n. 2, p. 69–74, 2008.

MOREIRA, M. D.; FERNANDES, N. C.; COSTA, L.; DUARTE, O. Internet do futuro: Um novo horizonte. **Minicursos do Simpósio Brasileiro de Redes de Computadores-SBRC**, v. 2009, p. 1–59, 2009.

PARC. **CCNx**. 2014. <<http://www.ccnx.org>>. Acessado em setembro de 2014.

RIPEANU, M. Peer-to-peer architecture case study: Gnutella network. In: IEEE. **Peer-to-Peer Computing, 2001. Proceedings. First International Conference on**. [S.l.], 2001. p. 99–100.

SANGLARD, F. **Quake 3 Source Code Review: Network Model**. 2012. <<http://fabiansanglard.net/quake3/network.php>>. Acessado em junho de 2014.

SMED, J.; KAUKORANTA, T.; HAKONEN, H. Aspects of networking in multiplayer computer games. **Electronic Library, The**, MCB UP Ltd, v. 20, n. 2, p. 87–97, 2002.

STEAM. **Steam Download Stats**. 2014. <<http://store.steampowered.com/stats/content>>. Acessado em 3 de setembro de 2014.

TANENBAUM, A.; STEEN, M. V. **Distributed systems**. [S.l.]: Pearson Prentice Hall, 2007.

TEAM, M. **Mininet**. 2015. <<http://www.mininet.org>>. Acessado em agosto de 2015.

THEESA. **Essential Facts About the Computer and Video Game Industry**. 2014. <http://www.theesa.com/facts/pdfs/esa_ef_2014.pdf>. Acessado em 30 de agosto de 2014.

WIKIPEDIA. **Digital distribution in video games**. 2014. <http://en.wikipedia.org/wiki/Digital_distribution_in_video_games>. Acessado em 30 de agosto de 2014.

YINANC, D. **technical-architecture-behind-Valves-Steam**. 2012. <<https://www.quora.com/What-is-the-technical-architecture-behind-Valves-Steam>>. Acessado em outubro de 2015.

ZHANG, L.; ESTRIN, D.; BURKE, J.; JACOBSON, V.; THORNTON, J. D.; SMETTERS, D. K.; ZHANG, B.; TSUDIK, G.; MASSEY, D.; PAPADOPOULOS, C. et al. Named data networking (ndn) project. **Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC**, Citeseer, 2010.