Centro Federal de Educação Tecnológica de Minas Gerais

Simon Lopes Isenmann

# Estudo de uma ferramenta para a simulação de Redes de Dados Nomeados (NDN)

Timóteo, MG

2023

Simon Lopes Isenmann

# Estudo de uma ferramenta para a simulação de Redes de Dados Nomeados (NDN)

Trabalho de Conclusão de Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais

Orientador: Prof. Me. Adilson Mendes Ricardo

Timóteo - Minas Gerais

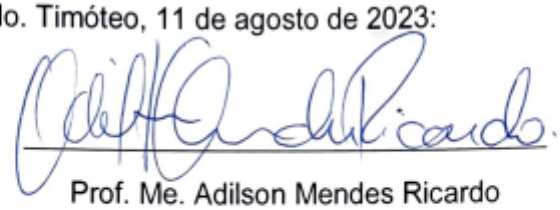
2023

Simon Lopes Isenmann

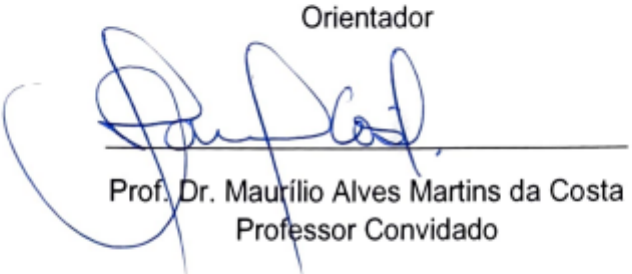
# Estudo de uma ferramenta para a simulação de Redes de Dados Nomeados (NDN)

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, campus Timóteo, como requisito parcial para obtenção do título de Engenheiro de Computação.

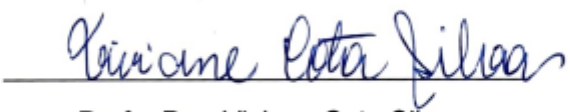
Trabalho aprovado. Timóteo, 11 de agosto de 2023:



Prof. Me. Adilson Mendes Ricardo
Orientador



Prof. Dr. Maurílio Alves Martins da Costa
Professor Convidado



Profa. Dra. Viviane Cota Silva
Professora Convidada

Timóteo
2023

# Agradecimentos

Aos meus familiares, pela compreensão, apoio e suporte nesta importante etapa de minha jornada acadêmica.

Ao professor Adilson Mendes Ricardo, por ter sido um orientador confiável e de grande auxílio.

Aos professores do CEFET-MG Campus Timóteo, em particular Viviane Cota, Maurilio Alves, Luciano Moreira, e o previamente mencionado Adilson Mendes, pelo apoio, reforço e, acima de tudo, paciência.

Aos meus colegas do curso de Engenharia de Computação, pela ajuda e companheirismo providenciados ao longo de minha jornada acadêmica.

A Guilherme Braga Araújo, pela ideia inicial a respeito do tema da tese, e o auxílio providenciado ao longo de sua elaboração.

Aos amigos que permaneceram ao meu lado, providenciando o encorajamento necessário para seguir em frente com este projeto.

## Resumo

Redes de Dados Nomeados, ou NDN, são uma alternativa teórica baseada em conteúdo às arquiteturas de redes de distribuição de informação definidas pelo protocolo TCP/IP (*Transmission Control Protocol/Internet Protocol,* ou Protocolo de Transmissão de Controle/Protocolo de Internet). Enquanto as redes atualmente predominantes baseiam-se em endereços, redes NDN são baseadas em nomeação de pacotes de dados, servindo como uma alternativa possivelmente mais otimizada para a forma como a Internet é predominante utilizada. Porém, devido à prevalência de estruturas já funcionais, tanto em software quanto em hardware, feitas para funcionar com redes TCP/IP, uma conversão em larga escala para esta nova arquitetura é inviável; portanto, redes NDN são primariamente tratadas como objeto de estudo com mais valor acadêmico do que prático. Tendo isso em mente, este trabalho teve como objetivo encontrar uma ferramenta funcional para simular uma rede NDN, bem como explorar o seu funcionamento e utilizá-la para adquirir uma base teórica a respeito de redes NDN.

## Abstract

Named Data Networks, or NDN, are a theoretical, content-based alternative to the information distribution networks defined by the TCP/IP protocol. While the currently predominant networks are based on addresses, NDN-type networks are based around naming data packets, serving as an alternative to said networks that is possibly more optimized in regards to how the Internet is currently mainly used. However, due to the prevalence of already functioning structures designed to function with TCP/IP, both in software and hardware, a large-scale conversion to this new network architecture is not viable; thus, Named Data Networks are primarily treated as an object of study with more academic value than practical value. With that in mind, this paper had the goal of finding a functional software tool for simulating an NDN-type network, as well as to explore its functionalities and utilize it to gain a theoretical understanding of such networks.

# LISTA DE ABREVIATURAS E SIGLAS

CCN - Content-Centric Network

CS - Content Store

FIB - Forward Interest Base

GB - Gigabyte(s)

HTTP - Hypertext Transfer Protocol

NDN - Named Data Network

P2P - Peer-to-Peer

PIT - Pending Interest Table

RAM - Random Access Memory

TCP/IP - Transmission Control Protocol/Internet Protocol

URL - Uniform Resource Locator

VoIP - Voice over Internet Protocol

# SUMÁRIO

**Agradecimentos** ..... **2**
**Resumo** ..... **3**
**Abstract** ..... **4**
**1) Introdução** ..... **7**
1.1) Tema ..... 7
1.2) Problema ..... 8
1.3) Objetivos ..... 8
1.4) Justificativa ..... 8
**2) Revisão de Literatura** ..... **9**
2.1) Referencial Teórico ..... 9
2.1.1) Rede NDN ..... 9
2.1.2) Conhecimento Teórico Prévio: HTTP ..... 9
2.1.3) Como a NDN Funciona ..... 10
2.1.4) Estrutura de funcionamento ..... 11
2.2) Trabalhos Correlatos ..... 13
**3) Procedimentos Metodológicos** ..... **15**
3.1) Ferramentas e Métodos ..... 15
3.1.1) ndnSIM ..... 15
3.2) Análise do Funcionamento da Ferramenta ..... 24
**4) Considerações Finais** ..... **30**
**5) Referências** ..... **31**

# 1) Introdução

A crescente importância da internet em várias facetas, tanto dentro da vida cotidiana como em outras porções do funcionamento mundial, leva a uma conclusão lógica simples: a eficiência da rede e de seu funcionamento são os gargalos para o avanço de melhores tecnologias e do acesso a elas. A atualmente predominante arquitetura, TCP/IP, foi desenvolvida para criar uma rede de comunicação, onde apenas importa para um determinado conjunto de dados os endereços dos pontos de origem e destino; porém, com o crescimento do novos usos para redes, como redes sociais e comercio digital, a internet vem sendo utilizada mais como uma rede de distribuição. (ZHANG et al. 2014)

O acesso à internet para uso pessoal no Brasil chegou a 90% em 2021, o que significa que quaisquer impactos na rede afetam a maioria da população brasileira. (INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA, 2022) Ao longo dos anos, tecnologias para usos diversos, como Internet das Coisas (*Internet of Things*, ou IoT), servidores em nuvem e identificação por radiofrequência (RFID), foram criadas e disseminadas, sendo vistas como uma grande utilidade apesar de necessitarem de métodos para funcionar de forma eficiente mesmo com a arquitetura TCP/IP não sendo idealmente compatível. (ZHANG et al., 2014)

Tendo tal situação em vista, foi criada uma arquitetura alternativa: a Rede de Dados Nomeados, ou NDN, que funciona como uma rede de distribuição de conteúdo, e visa remediar uma porção dos desafios proporcionados à estrutura de rede predominante atual. (ZHANG et al., 2014)

Dito isso, informações a respeito de redes NDN em língua portuguesa parecem ser escassas; portanto, este trabalho teve como objetivo encontrar uma ferramenta que possa auxiliar no aprendizado a respeito do assunto em questão. Uma vez encontrada a ferramenta ndnSIM e estabelecido que ela pode ser utilizada para o propósito desejado, foi feita e documentada uma análise de seu funcionamento.

## 1.1) Tema

Análise da funcionalidade de ferramentas para a simulação de Redes de Dados Nomeados.

## 1.2) Problema

A arquitetura NDN, ou Named Data Network, foi criada para servir como uma alternativa à arquitetura TCP/IP, que atualmente domina o mercado (AFANASYEV, 2018). Porém, há uma escassez de materiais tratando-se dessa nova arquitetura no idioma português, bem como a realização de testes centrados em arquiteturas de rede costumam demandar grande quantidade de recursos caso sejam realizados em ambiente real. Tendo isto em mente, a pergunta deste trabalho é: Como utilizar uma ferramenta de simulação para o estudo de redes NDN?

## 1.3) Objetivos

Geral:

- Encontrar uma ferramenta que auxilie no estudo de redes NDN por meio de simulação.

Específicos:

- Estudar e caracterizar a arquitetura NDN;
- Encontrar e aprender sobre simuladores de redes NDN;
- Definir um simulador que seja mais útil para compreensão geral;
- Descrever sobre o funcionamento do simulador definido.

## 1.4) Justificativa

A arquitetura de rede TCP/IP é predominante sobre o mercado de redes, e talvez por consequência costuma ser a única que é ensinada. Considerando-se que, em comparação com a NDN, ela possui não apenas pontos fortes mas também pontos fracos (BARROS, 2015), o estudo de outra arquitetura como alternativa pode ser útil para a formação de profissionais na área de redes. Para a realização viável de estudos com uma alternativa a algo tão unânime em seu campo como TCP/IP, o uso de softwares de simulação providenciam uma alternativa muito mais acessível, portanto o objetivo deste artigo caracterizando e explicando o uso de um deles é justificado.

# 2) Revisão de Literatura

## 2.1) Referencial Teórico

### 2.1.1) Rede NDN

NDN, uma ideia de internet do futuro, deriva de um projeto anterior, CCN (*content-centric network*), publicado inicialmente por Van Jacobson em 2006; a ideia investiga a evolução de uma rede centrada em hosts como a de hoje para uma rede centrada em conteúdo, proposta por Jacobson (JACOBSON, 2006). Como diferenciais principais, primeiramente os nomes dos dados e das aplicações que os usam são usados diretamente no *forwarding* de pacotes de dados (envio a destinatários interessados), visto que os solicitantes requisitam dados desejados por nome.

Em segundo lugar, as comunicações são asseguradas de forma centrada em dados e conteúdo através de assinaturas criptográficas de produtores. Em terceiro lugar, NDN utiliza um plano de *forwarding* baseado em estados, sendo que cada requisição mantém um estado que é apagado quando o pacote de dados correspondente chegar. (ZHANG et al., 2014)

### 2.1.2) Conhecimento Teórico Prévio: HTTP

Há um protocolo da camada de aplicação cujo uso muito difundido leva ele a ser conhecido comumente: o HTTP (*Hypertext Transfer Protocol*, ou Protocolo de Transferência de Hipertexto). Ele é primariamente utilizado para sistemas de informação de hipermídia, isto é, formas de comunicação distintas unidas em um único local por ligações lógicas, e serve como uma base para comunicação de dados pela Internet. (BERNERS-LEE, FIELDING, FRYSTYK, 1996) Inicialmente tendo seu desenvolvimento começado em 1989 por Tim Berners-Lee, ele veio a receber atenção constantemente da comunidade de desenvolvimento para Web, tendo diversas atualizações e versões.

O funcionamento do protocolo HTTP é feito de forma *request-response*, isto é, uma máquina envia uma requisição (*request*) de certos dados para outra, que por sua vez processa a requisição e envia de volta uma resposta (*response*). É um funcionamento análogo a uma ligação telefônica, onde a origem da ligação deve aguardar até que ela seja atendida por um recipiente para que haja comunicação.(HOHPE, 2015)

### 2.1.3) Como a NDN Funciona

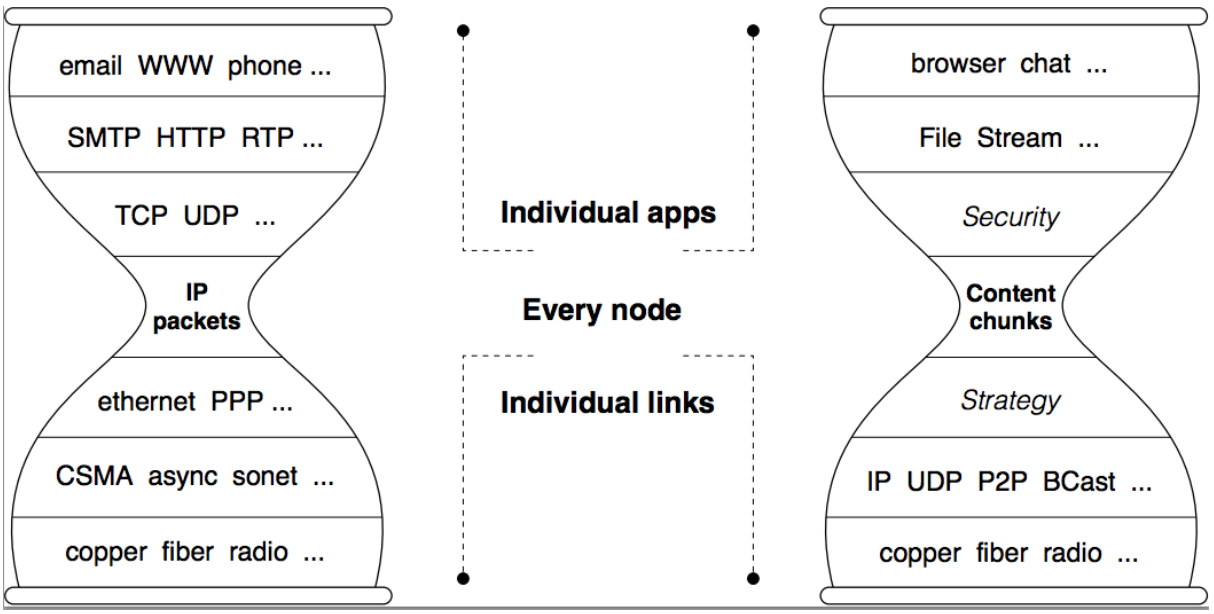
O funcionamento de NDN na camada de rede possui semelhanças com o funcionamento do protocolo HTTP da camada de aplicação: a NDN adota o modelo de comunicação *request-reply*, e usa nomes de dados de aplicações na camada de rede para fazer com que os serviços de rede combinem com os padrões de comunicação de aplicações. Porém, em contraste às URLs (*Uniform Resource Locator*, ou Localizador Uniforme de Recursos) usadas em *requests* HTTP que são apenas usadas por aplicações, os nomes de dados são usados pela NDN também para buscar dados e para definir políticas de segurança e automatizar a autenticação de dados e o controle de confidencialidade. Por consequência, o design de *namespace* (ou seja, as políticas de nomeação de objetos de dados) é o passo mais importante e por vezes também o mais difícil em todos os designs de aplicações e protocolos de NDN. (AFANASYEV, 2018)

Em adição a serem partes operando na camada de rede, os *Data Packets* (pacotes de dados, sub-unidades de informação passadas pela rede) da NDN diferem de objetos de dados HTTP em dois outros aspectos: primeiramente, enquanto uma mensagem de resposta HTTP está implicitamente ligada à URL de requisição pela conexão TCP, todos os pacotes de NDN carregam o nome dos dados explicitamente, assim como uma assinatura criptográfica que liga o nome ao conteúdo no momento de criação. Em segundo lugar, enquanto uma mesma URL pode buscar conteúdos diferentes, *Data Packets* NDN são imutáveis; cada nome identifica um Data Packet unicamente. Quando um produtor de dados muda o conteúdo de um Data Packet, ele tem que gerar um novo Packet com um novo nome para distinguir entre versões diferentes do conteúdo. (AFANASYEV, 2018)

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, em segurança e em aplicações ponto a ponto - ou seja, que fazem uso do protocolo *Point-to-Point* da camada de enlace para diretamente conectar 2 nós - como e-mail/correio eletrônico, VoIP (*Voice over Internet Protocol,* roteamento de voz humana por rede de computadores, tipicamente com uso do IP), *stream* (transmissão de áudio e vídeo ao vivo por internet), entre outros (JACOBSON et al., 2006). Porém, assim como as redes P2P, existe a possibilidade da criação de uma rede híbrida *overlay* (ou rede sobreposta, aonde uma rede nova é virtualmente criada e sobreposta a outra para funcionar em conjunto), funcionando em parceria com o tradicional *Internet Protocol* (IP), trazendo novas oportunidades de serviços e aplicações (BARROS, 2015).

A Figura 1 demonstra uma comparação entre modelos de redes mais tradicionais e atualmente em uso, e redes NDN, com base no esquema de visualização *Hourglass*:

Figura 1: modelo *hourglass* de redes NDN e TCP/IP em comparação direta




Fonte: <https://named-data.net/project/execsummary/hourglass/>. Acesso em 11 de jul de 2022.

### 2.1.4) Estrutura de funcionamento

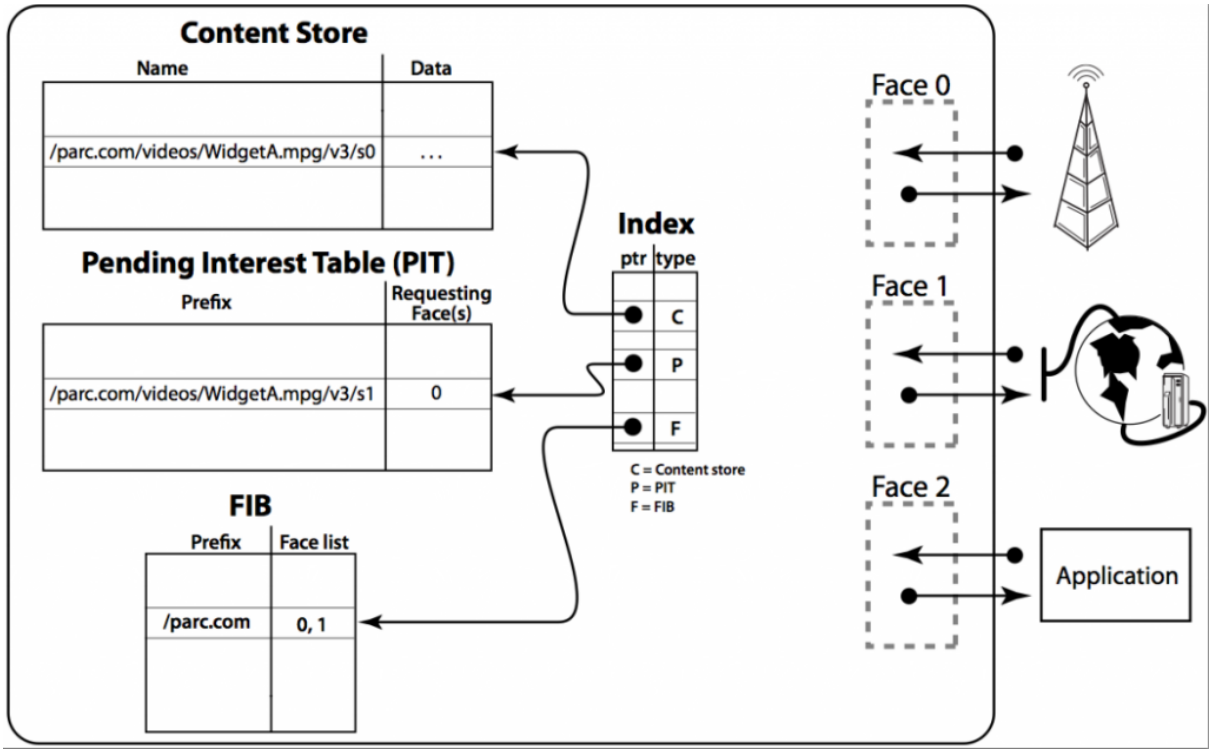
Um nó NDN é composto pelas seguintes entidades: *Content Store* (CS), *Pending Interest Table* (PIT) e *Forward Information Base* (FIB).

A CS funciona basicamente como um *buffer,* isto é, uma porção de memória para a armazenagem temporária de dados, 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, em diante.

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. (AFANASYEV, 2018)

Por fim, a FIB é uma base de dados usada para o encaminhamento de interesses a até suas prováveis fontes de conteúdo. É semelhante à FIB do protocolo IP, porém, podendo usar múltiplos destinos ao invés de um único. Isso se deve ao fato de o NDN não ser limitado pelo *spanning tree*. O *spanning tree* se trata de um protocolo da camada de enlace criado para prevenir *loops* em comunicações comutadas (PERLMAN, 2000). A Figura 2 demonstra um esquema da estrutura de funcionamento de uma rede NDN.

Figura 2: estrutura de funcionamento de uma rede NDN




. Fonte: <https://named-data.net/project/archoverview/>. acesso em 14 de ago. de 2022.

## 2.2) Trabalhos Correlatos

Barros (2015), em sua dissertação, analisa a viabilidade de redes NDN para o uso em jogos online, visando a viabilidade e o atendimento das diversas necessidades do caso de uso em questão. Por fim, encontra como resultado que NDN ainda necessita de desenvolvimentos, e portanto a melhor aproximação é um uso híbrido entre as arquiteturas NDN e IP.

Afanasyev et al. (2018) visam reunir informações a respeito da arquitetura NDN, tratando de seus detalhes, suas vantagens, e funcionamento; seus objetivos sendo ressaltar o que motivou sua criação, demonstrar sua validez como área de pesquisa e criar a motivação para pesquisas futuras em demais membros da comunidade de pesquisa científica.

Ioannou et al. (2018) tratam de uma proposta do uso de uma arquitetura NDN habilitada através de *caches* de memória em *cloudlets* (*datacenters* em nuvem de escala menor com maior mobilidade), como uma forma de auxiliar os limites do alcance da rede padrão para providenciar melhor acesso à internet de forma sem-fio em pontos turísticos mais afastados na Irlanda.

Afanasyev, Moiseenko e Zhang (2012) tratam do software ndnSIM e de seu processo de criação, cujo qual será explorado neste trabalho; ndnSIM feito a fim de servir como ferramenta de pesquisa uniforme para a comunidade de pesquisadores de rede, e no artigo explicam o seu funcionamento os princípios que foram levados em consideração ao criá-lo.

# 3) Procedimentos Metodológicos

## 3.1) Ferramentas e Métodos

Para a implementação de uma simulação de uma rede NDN, neste trabalho, será utilizado o software ndnSIM, operando em uma máquina virtual com o sistema operacional Linux Ubuntu de 64 bits. A escolha deve-se ao princípio de que a capacidade de uma máquina virtual é menor do que a da máquina que a simula, portanto, caso uma máquina virtual seja capaz de fazer uso do software escolhido, pode-se assumir que máquinas físicas também podem ser capazes de utilizá-lo.

O software gratuito Oracle VM VirtualBox foi utilizado para que a máquina virtual fosse gerada e utilizada.

### 3.1.1) ndnSIM

O software de simulação escolhido para a metodologia deste trabalho, ndnSIM, foi criado com base em NS-3, um *framework* modular de simulação de eventos discretos - isto é, um simulador que modela a operação de um sistema como uma sequência de eventos no tempo. NS-3 foi feito em C++, com um enfoque primário em pesquisa e educação. (ns-3…, 2011)

Criado por Alex Afanasyev, Ilya Moiseenko e Lixia Zhang, o projeto ndnSIM foi criado de forma *open-source* (código aberto, ou um código fonte disponível gratuitamente) para que a comunidade científica possuísse uma plataforma de simulação aberta e em comum que implementasse os componentes básicos de uma rede NDN de forma modular. O simulador discreto resultante continha diversas classes e conjuntos de classes C++ para a modelagem do comportamento de cada entidade da camada de rede pertinente à arquitetura NDN. (AFANASYEV, MOISEENKO, ZHANG, 2012)

A primeira tentativa de utilizar o software foi feita utilizando o sistema operacional Ubuntu versão 16.04.6, com um alocamento de 2 GB de memória virtual; no terminal, foi utilizada a linha de comando:

```
“sudo apt install build-essential libsqlite3-dev
libboost-all-dev libssl-dev git python-setuptools castxml”
```

para instalar as dependências principais. O resultado obtido foi um erro ao tentar a instalação, o que indicou que poderia ser necessária outra versão do sistema operacional.

Ao tentar a instalação em um sistema operacional Ubuntu versão 20.04.4 e 4 GB de memória RAM virtual, o processo passou a funcionar corretamente, indicando que esta versão é um requisito.

Em seguida à linha anterior, utiliza-se o comando

```
“sudo apt-get install libboost-all-dev”
```

para as dependências remanescentes; porém, diversas não foram encontradas ao tentar a sua instalação.

Utilizando-se da ferramenta de controle de versões Git, que possibilita o acesso a arquivos de codificação e desenvolvimento de um projeto por pessoas externas, foram clonados para a máquina em execução os arquivos pertinentes ao simulador. Porém, ao tentar realizar a execução do software, foi encontrada uma falha: o comando providenciado para fazer com que o simulador começasse sua execução não foi reconhecido. Tendo em vista esta falha, foi tentada uma segunda instalação em uma máquina recém-criada; nesta, os comandos foram executados de acordo com o que o próprio portal do ndnSIM indica:

```
“sudo apt install gir1.2-goocanvas-2.0 gir1.2-gtk-3.0
libgirepository1.0-dev python3-dev python3-gi python3-gi-cairo
python3-pip python3-pygraphviz python3-pygccxml
sudo pip3 install kiwi”
```

Para instalar as dependências Python do simulador;

```
“mkdir ndnSIM
cd ndnSIM
git clone https://github.com/named-data-ndnSIM/ns-3-dev.git
ns-3
```

```
git clone https://github.com/named-data-ndnSIM/pybindgen.git
pybindgen
git clone --recursive
https://github.com/named-data-ndnSIM/ndnSIM.git
ns-3/src/ndnSIM”
```

Para criar a pasta que abrigará os arquivos do simulador, e fazer o download dele e de todos os seus componentes;

```
“cd ns-3
./waf configure --enable-examples
./waf”
```

Para compilar o simulador.

Terminado o processo de instalação e compilação, foi realizado um teste inicial por meio do comando:

“./waf --run=ndn-simple”,

como providenciado por vias oficiais referentes ao simulador. Por mais que tal teste tenha finalizado sem erro algum, ele também não gerou resultados visíveis.

Em seguida, foi feito outro teste por meio do comando

“./waf --run=ndn-grid”,

também providenciado pelas mesmas vias oficiais; fora um tempo de execução significativamente menor, não houve diferença entre os dois comandos de exemplo tentados.

Tendo em vista tais resultados pouco esclarecedores, foi feita uma re-configuração do simulador para um modo otimizado com o comando

“./waf configure -d optimized”.

Esta decisão tornou impossível a execução dos dois testes que foram tentados anteriormente, portanto o software foi re-configurado novamente com

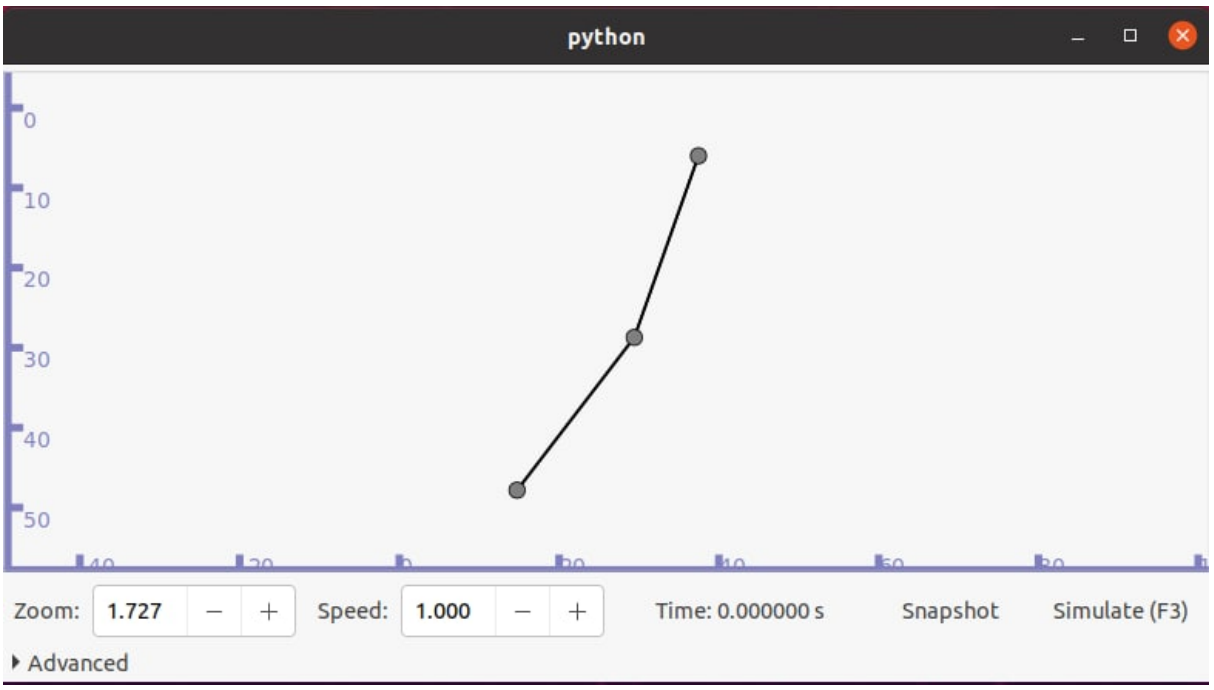
“./waf configure --enable-examples”;

e então foi executado o primeiro comando de teste, porém com uma seção adicional:

“./waf --run=ndn-simple --vis”

para habilitar a visualização através dos recursos de Python inclusos na instalação. Por algum motivo, isto resultou em uma nova compilação do simulador. Quando esta segunda compilação terminou, abriu-se uma janela para visualização da simulação de exemplo, com 3 nós ligados em uma sequência, como demonstrado na Figura 3:

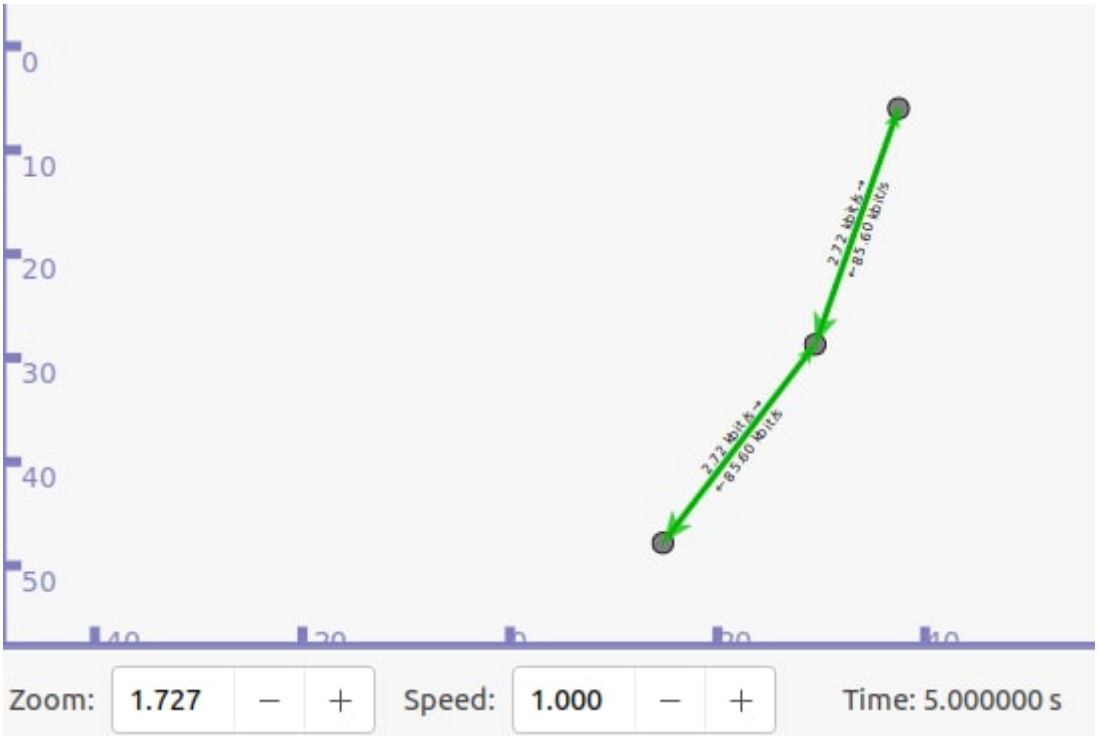
Figura 3: exemplo simples de rede NDN pré-pronto providenciado pelo simulador




Fonte: autor, 2023.

Ao ativar a opção “Simulate”(simular), medições de taxas de transferência de dados, em Kbits por segundo, apareceram brevemente pelas ligações entre os nós, que foram cobertas por indicadores de sentido do fluxo de dados, como demonstrado na Figura 4:

Figura 4: exemplo simples de rede NDN pré-pronto da figura 3, em estado de execução




Fonte: autor, 2023.

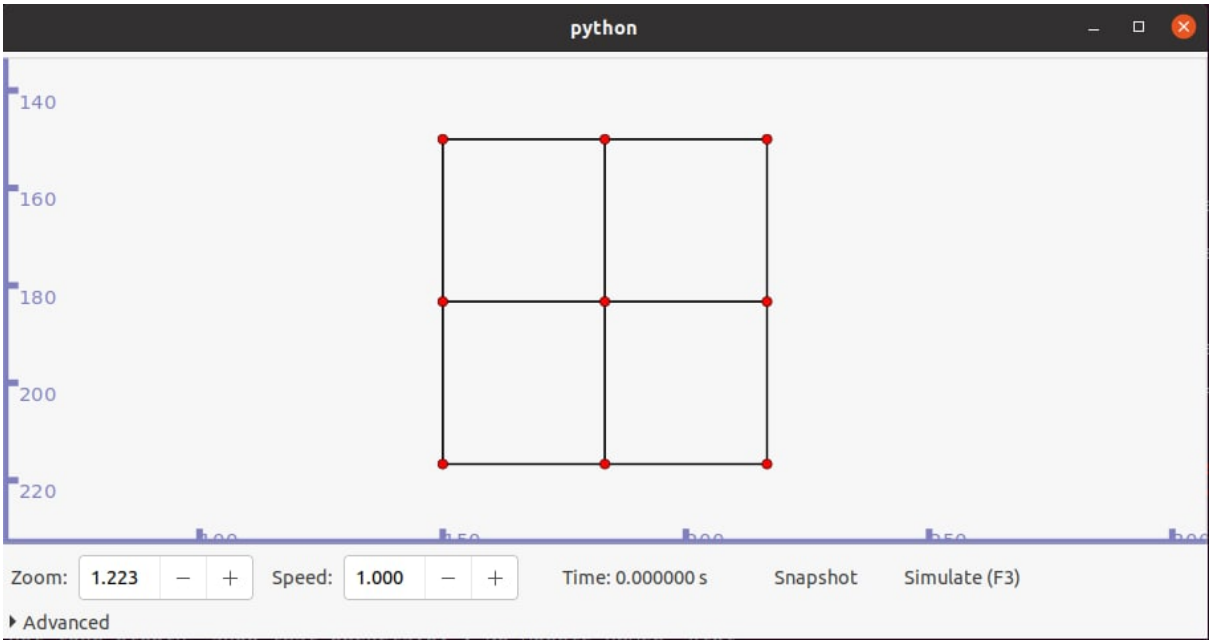
A aba “*Advanced*” (avançado) revela novas configurações, como a capacidade de aumentar o tamanho dos nós, seja todos ou um por vez caso ele seja selecionado.

Ao rodar o comando do segundo exemplo, mas com o elemento de visualização, por meio do comando

“`./waf --run=ndn-grid --vis`”,

obteve-se um resultado de 9 nós interligados em um padrão quadrado, conforme ilustrado na Figura 5:

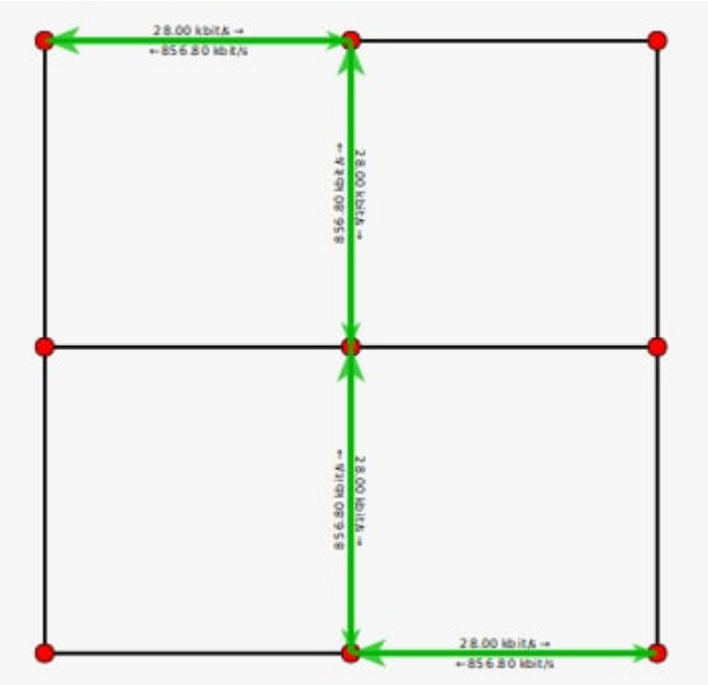
Figura 5: Exemplo organizado em grade de uma rede NDN, pré-pronto e providenciado pelo simulador




Fonte: autor, 2023.

Iniciando-se a função “*Simulate*”, apareceram vetores indicando o fluxo de dados e arquivos sendo simulado, como demonstrado na Figura 6:

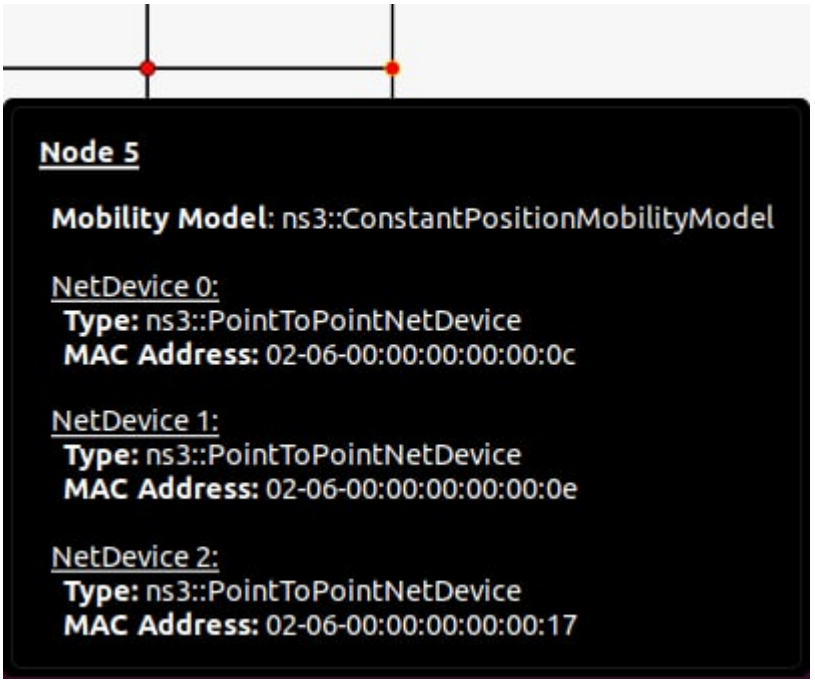
Figura 6: exemplo de rede NDN organizada em grade da figura 5, em estado de execução.




Fonte: autor, 2023.

Movendo o cursor sobre um nó resulta na nota ilustrada na Figura 7:

Figura 7: dica de contexto de um nó da rede NDN simulada nas figuras 5 e 6

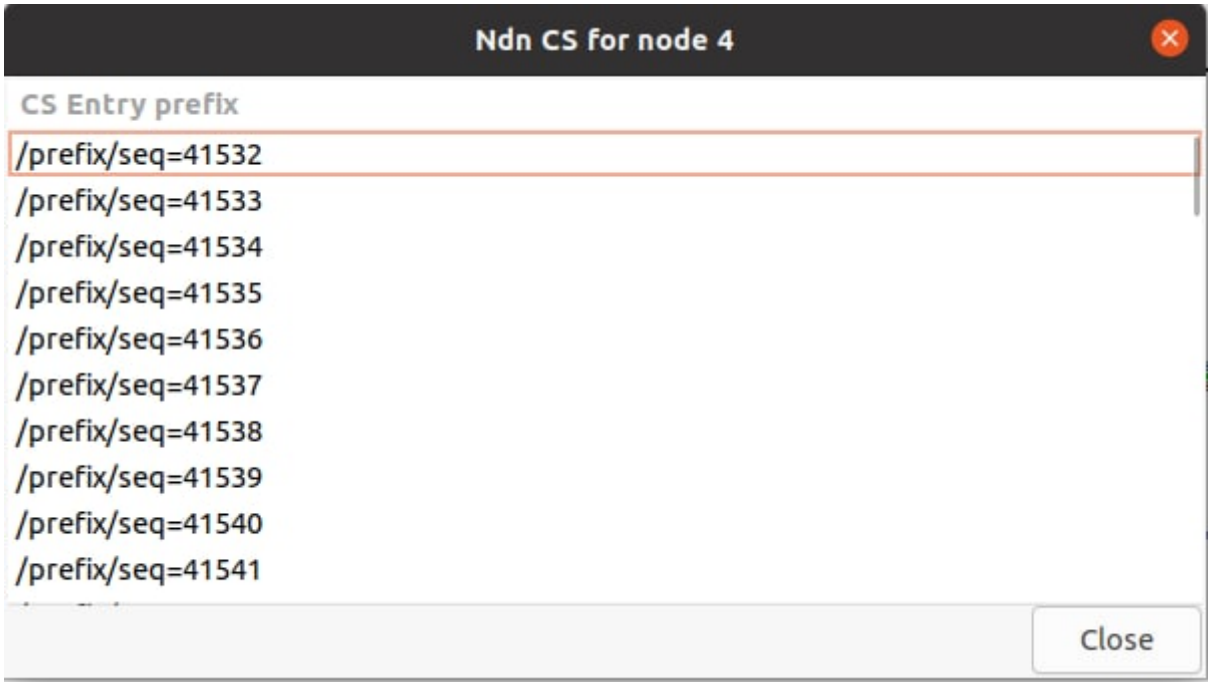



Fonte: autor, 2023.

Usando o botão direito do cursor em um nó resulta em um menu de contexto para demonstrar diferentes aspectos dele, sendo as opções e os resultados citados a seguir e demonstrados nas figuras 8 - 11:

- *“Show NDN CS” (Content Store)*:

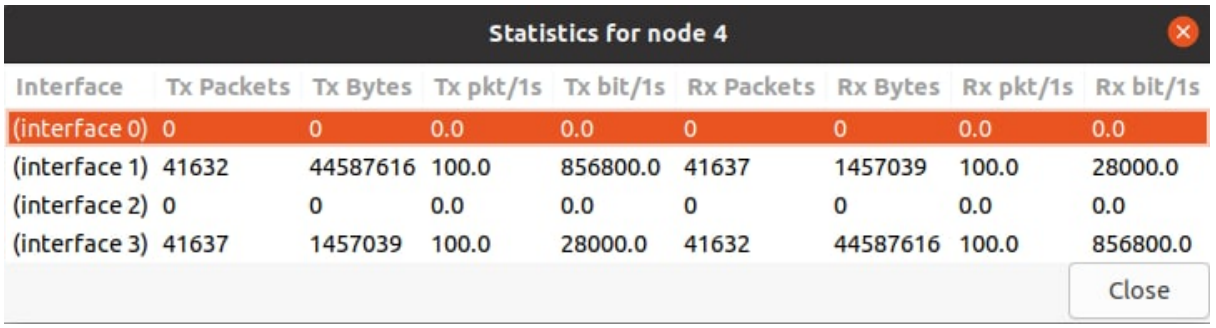
Figura 8: opção *“Show NDN CS”*




Fonte: autor, 2023.

- *“Show NDN Faces”*, que não produziu resultado visível;
- *“Show Interface Statistics”*:

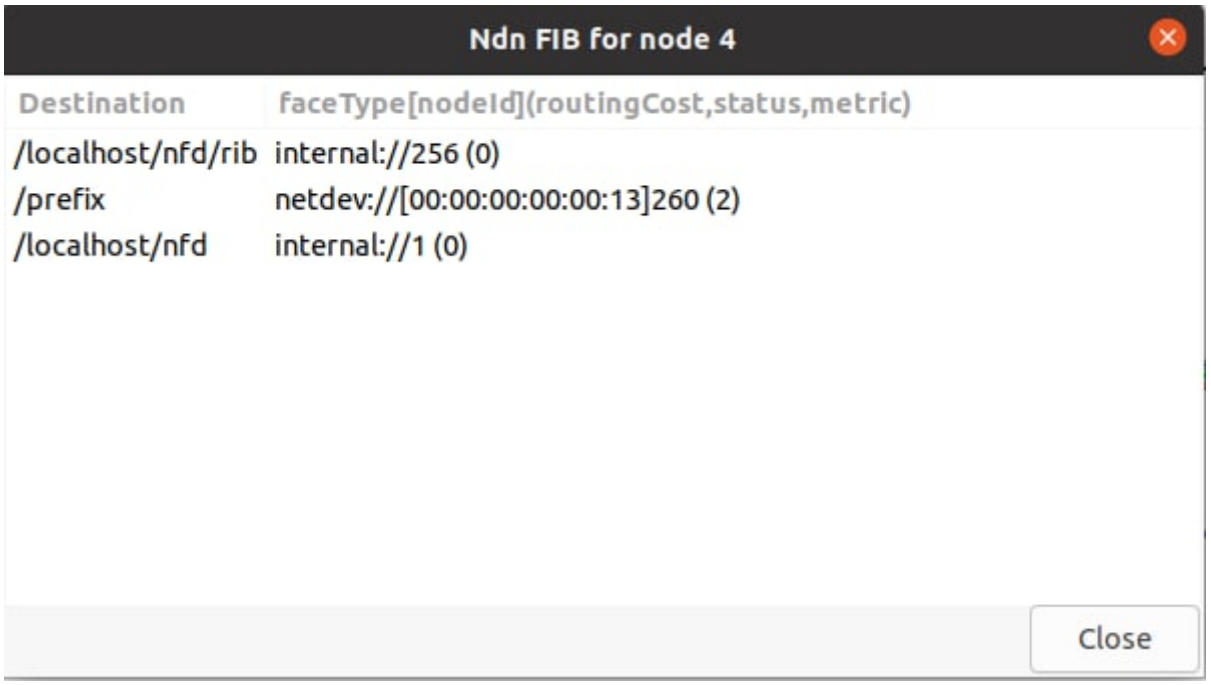
Figura 9: opção *“Show Interface Statistics”*




Fonte: autor, 2023.

- *“Show IPv4 Routing Table”*, que aparentemente também não produz resultado visível;
- *“Show NDN FIB” (Forwarding Information Base)*:

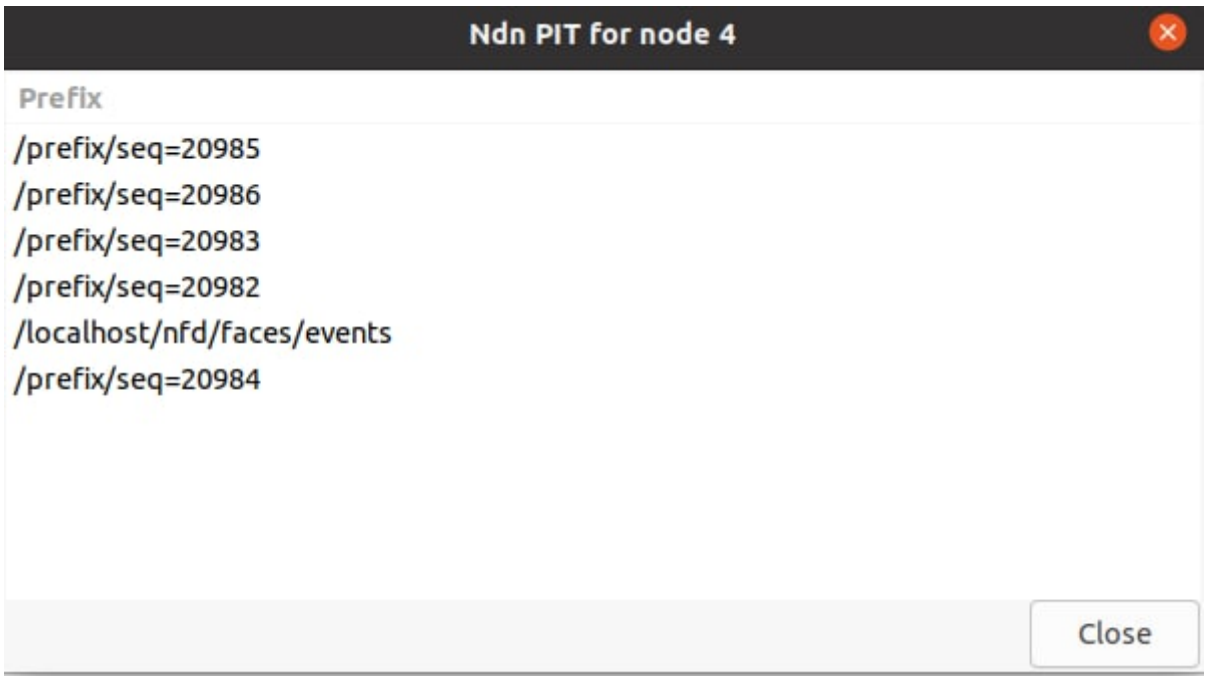
Figura 10: opção *“Show NDN FIB”*




Fonte: autor, 2023.

- *“Show NDN PIT”* (*Pending Interest Table*, que constantemente se atualiza conforme ao fluxo e distribuição de conteúdo):

Figura 11: opção *“Show NDN PIT”*




Fonte: autor, 2023.

## 3.2) Análise do Funcionamento da Ferramenta

Voltando-se, agora, para os arquivos que geraram tais simulações, é percebido que as interfaces visuais e interativas são algo diferente, feito em Python para alguns exemplos. Por padrão, arquivos de simulação são códigos na linguagem C++, que fazem uso extenso das bibliotecas e recursos providenciados pelo framework NS-3, e das adições a ele próprias para simular um ambiente de rede NDN.

Apesar das porções interativas até então vistas, os principais arquivos de simulação na realidade assemelham-se menos a softwares focados em interação com um usuário, como sites ou games, e mais com *scripts*; contendo não apenas os recursos de instanciamento necessários para simular um ambiente de rede, mas também uma lista de ações a serem realizadas dentro do ambiente em questão, bem como recursos para a impressão ou criação de registro de todos os passos realizados pelas ações descritas.

Em suma, enquanto esperava-se que o simulador fosse assemelhar-se a softwares de simulação gráfica como *Cisco Packet Tracer* ou *Hades*, aonde uma tarefa é feita por meio de interação em alto nível e a análise dos resultados é feita manualmente, a realidade demonstrou que trata-se de um *framework* onde detalhes são definidos a nível de código, e no qual os resultados são analisados de forma quase automática. Como pode ser visto nas Figuras 12 e 13, o programa obtido contém exemplos de código com comentários extensos para a documentação e compreensão de qualquer indivíduo que venha a fazer uso do software:

Figura 12: primeira metade do código em C++ referente ao exemplo simples de rede ponto-a-ponto pré-pronto

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/ndnSIM-module.h"

namespace ns3 {

int
main(int argc, char* argv[])
{
  // setting default parameters for PointToPoint links and channels
  Config::SetDefault("ns3::PointToPointNetDevice::DataRate", StringValue("1Mbps"));
  Config::SetDefault("ns3::PointToPointChannel::Delay", StringValue("10ms"));
  Config::SetDefault("ns3::DropTailQueue<Packet>::MaxSize", StringValue("20p"));

  // Read optional command-line parameters (e.g., enable visualizer with ./waf --run=<> --visualize
  CommandLine cmd;
  cmd.Parse(argc, argv);

  // Creating nodes
  NodeContainer nodes;
  nodes.Create(3);

  // Connecting nodes using two links
  PointToPointHelper p2p;
  p2p.Install(nodes.Get(0), nodes.Get(1));
  p2p.Install(nodes.Get(1), nodes.Get(2));

  // Install NDN stack on all nodes
  ndn::StackHelper ndnHelper;
  ndnHelper.SetDefaultRoutes(true);
  ndnHelper.InstallAll();

  // Choosing forwarding strategy
  ndn::StrategyChoiceHelper::InstallAll("/prefix", "/localhost/nfd/strategy/multicast");
```

Figura 12: primeira metade do código em C++ referente ao exemplo simples de rede ponto-a-ponto pré-pronto (ver figuras 3 e 4 para sua execução gráfica). Explicação dos comentários é a ordem na qual o ambiente de simulação é preparado: primeiramente são definidos os parâmetros padrões para canais e ligações P2P, em seguida são lidos parâmetros adicionais que possam ter sido adicionados à linha de comando que iniciou a execução do arquivo, então cria-se os nós da rede simulada e as suas conexões, antes de instalar uma configuração responsável pelos nós funcionarem de acordo com a arquitetura NDN e escolher a estratégia de *Forwarding* de pacotes. Fonte: ndnSIM, 2023.

Figura 13: segunda metade do código em C++ referente ao exemplo pré-pronto de rede ponto-a-ponto

```
  // Installing applications

  // Consumer
  ndn::AppHelper consumerHelper("ns3::ndn::ConsumerCbr");
  // Consumer will request /prefix/0, /prefix/1, ...
  consumerHelper.SetPrefix("/prefix");
  consumerHelper.SetAttribute("Frequency", StringValue("10")); // 10 interests a second
  auto apps = consumerHelper.Install(nodes.Get(0));                        // first node
  apps.Stop(Seconds(10.0)); // stop the consumer app at 10 seconds mark

  // Producer
  ndn::AppHelper producerHelper("ns3::ndn::Producer");
  // Producer will reply to all requests starting with /prefix
  producerHelper.SetPrefix("/prefix");
  producerHelper.SetAttribute("PayloadSize", StringValue("1024"));
  producerHelper.Install(nodes.Get(2)); // last node

  Simulator::Stop(Seconds(20.0));

  Simulator::Run();
  Simulator::Destroy();

  return 0;
}

} // namespace ns3

int
main(int argc, char* argv[])
{
  return ns3::main(argc, argv);
}
```

Figura 13: segunda metade do código em C++ referente ao exemplo pré-pronto de rede ponto-a-ponto. Os comentários explicam que, em ordem cronológica, após os eventos detalhados na Figura 12, são definidos comportamentos de 2 nós para simular aplicações neles, antes de definir a duração da simulação e fazer com que ela execute. Fonte: ndnSIM 2023

No segmento de código contido na figura 12, percebe-se a inclusão de bibliotecas do *framework*, incluindo "*point-to-point-module.h*", para auxiliar no estabelecimento de conexões ponto-a-ponto e no envio de pacotes entre nós por ligações virtuais; "*network-module.h*" para o estabelecimento de funcionalidades de rede; e "*ndnSIM-module.h*", para que as funcionalidades de rede em questão seja de acordo com um modelo NDN. (AFANASYEV, 2022)

O segmento de código da Figura 13 mostra não só mais definições e especificações quanto ao ambiente simulado, como também a já observada natureza de execução própria e auto-definida, com comandos para que o nó consumidor pare de consumir ou enviar *requests* de informação após 10 segundos de funcionamento, e o simulador cesse sua execução ao ter simulado o funcionamento de uma rede por 20 segundos.

Também na Figura 13 é demonstrada a designação de um nó em específico como produtor de dados, e de outro como consumidor, designações que definem o fluxo de *requests*, respostas e de dados. Tal definição torna-se mais significante quando em uso em uma rede mais complexa, como foi observado na Figura 6: sempre que o cenário "*ndn-grid*" foi executado com a visualização habilitada, os nós de origem e destino foram os mesmos, como especificado.

Figura 14: segmento do código em C++ referente ao exemplo pré-pronto de rede em grade, contendo definições de nós

```
// Installing global routing interface on all nodes
ndn::GlobalRoutingHelper ndnGlobalRoutingHelper;
ndnGlobalRoutingHelper.InstallAll();

// Getting containers for the consumer/producer
Ptr<Node> producer = grid.GetNode(2, 2);
NodeContainer consumerNodes;
consumerNodes.Add(grid.GetNode(0, 0));

// Install NDN applications
std::string prefix = "/prefix";

ndn::AppHelper consumerHelper("ns3::ndn::ConsumerCbr");
consumerHelper.SetPrefix(prefix);
consumerHelper.SetAttribute("Frequency", StringValue("100")); // 100 interests a second
consumerHelper.Install(consumerNodes);

ndn::AppHelper producerHelper("ns3::ndn::Producer");
producerHelper.SetPrefix(prefix);
producerHelper.SetAttribute("PayloadSize", StringValue("1024"));
producerHelper.Install(producer);
```

Figura 14: segmento do código em C++ referente ao exemplo pré-pronto de rede em grade, contendo as definições de um nó como o produtor de dados, e um grupo de nós (neste caso contendo apenas 1 nó) como sendo consumidores de dados.

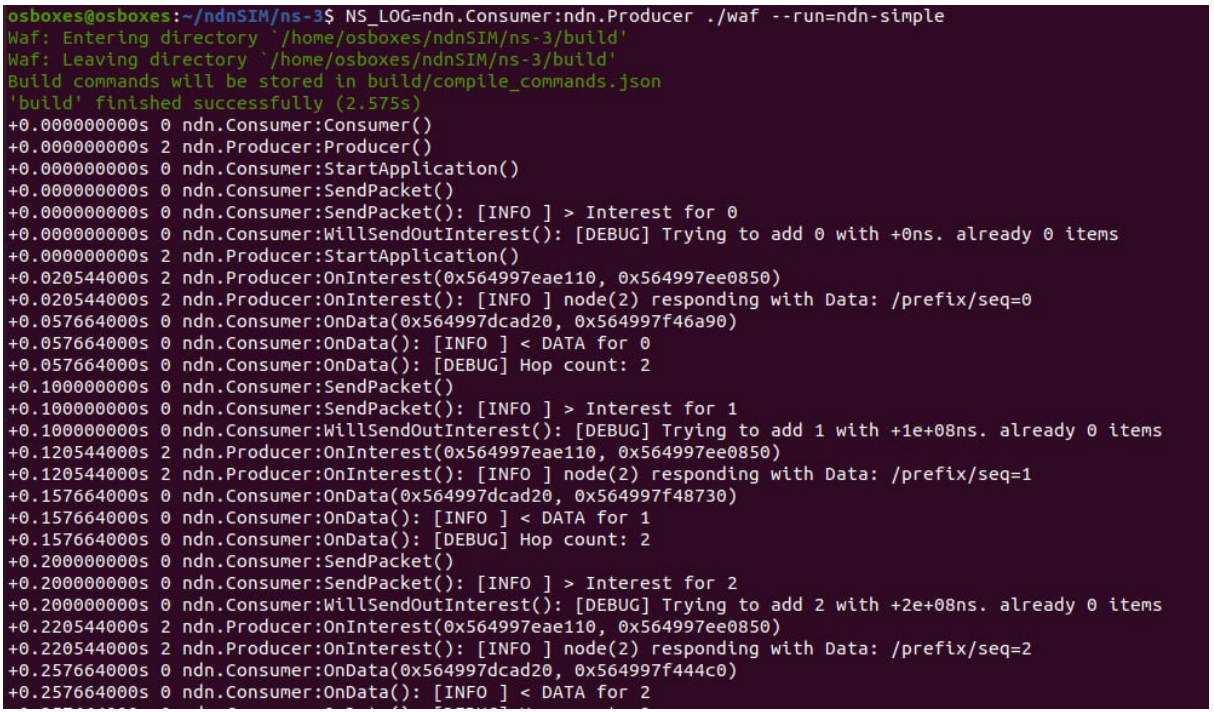
Vale ressaltar, nas figuras demonstrando o código dos cenários simulados, o quanto do que é instanciado é feito por meio de uso de bibliotecas já prontas, necessitando apenas de chamadas com parâmetros para que o ambiente de execução, e as instruções da execução, sejam propriamente preparados e utilizados; até mesmo uma estratégia de *forwarding* de pacotes foi disponibilizada e usada. Isto demonstra parte da quantidade de recursos disponíveis por meio do *framework* NS-3, bem como tais recursos são relativamente auto-contidos e fáceis de utilizar.

Para um olhar mais específico quanto ao funcionamento do software, foi utilizado no console o comando:

```
"NS_LOG=ndn.Consumer:ndn.Producer ./waf --run=ndn-simple"
```

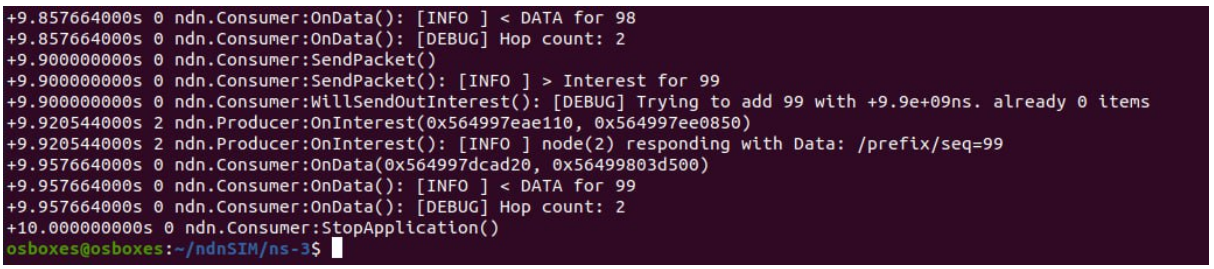
Para que o exemplo de 3 nós previamente visto fosse novamente executado; porém, como houve uma reconfiguração anteriormente durante os experimentos, o simulador não encontrava-se mais em seu modo otimizado. O resultado, então, foi uma série de impressões no console, todas seguidas de demarcações temporais: inicialmente, no momento 0, consistiam-se de chamadas de funções de instanciamento de componentes de rede; porém logo passaram a ser registros de funções referentes ao funcionamento de um ambiente de rede.

Figura 15: porção inicial dos registros, ou *logs*, resultantes da execução de um exemplo previamente disponível




Fonte: autor, 2023.

Figura 16: porção final dos *logs* de execução do mesmo exemplo, finalizando o envio de pacotes e então parando o funcionamento da aplicação

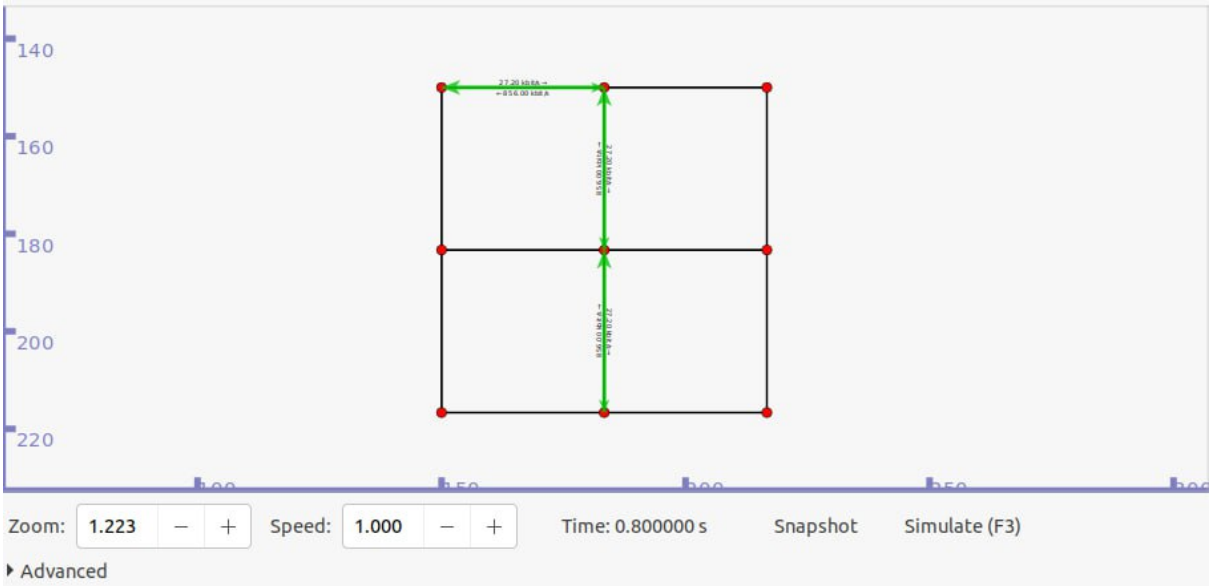



Fonte: autor, 2023.

Como um experimento, foi feita uma pequena alteração no código referente ao exemplo em grade, onde o nó a ser designado como produtor foi alterado: as coordenadas (2, 2) foram mudadas para (2, 1). Ao salvar as alterações e utilizar o

comando para executar o arquivo com visualização habilitada, foi feita uma nova compilação - o que indica que isto é necessário após cada alteração em algum arquivo previamente executado - e, quando finalizada, a opção “*Simulate*” foi usada, resultando no conteúdo da figura 17.

Figura 17: execução visualizada de um arquivo exemplo manualmente alterado




Fonte: autor, 2023.

## 4) Considerações Finais

Ao longo deste trabalho, foi adquirida uma base teórica a respeito de um tipo relativamente novo de rede, as Redes de Dados Nomeados ou NDN, criada como uma alternativa à arquitetura TCP/IP. Junto ao conhecimento sobre como uma rede com tal arquitetura iria funcionar, obteve-se também motivos para considerar uma troca da arquitetura atualmente prevalente para uma arquitetura NDN como algo inviável; e, portanto, concluiu-se que o valor de redes NDN é maior como um instrumento de estudo.

Tendo seu valor acadêmico como enfoque, foi então encontrado o *software* ndnSIM, criado por pesquisadores da área de redes com base no *framework* NS-3, para simular eventos e execuções em topologias de redes NDN. O *software* em questão foi analisado, e logo foi descoberto que o mesmo não comportava-se como um software gráfico de simulação constante e interativa, assemelhando-se mais a uma adição extra ao *framework* no qual foi baseado, e com enfoque na criação de códigos na linguagem C++ para não apenas montar um ambiente de execução, mas também definir os eventos que ocorrem dentro de tal ambiente ao longo de certo tempo.

## 5) Referências


AFANASYEV, Alexander et al. A Brief Introduction to Named Data Networking. Military Communications Conference 2018.

AFANASYEV, Alexander; MOISEENKO, Ilya; ZHANG, Lixia. ndnSIM:NDN simulator for NS-3. NDN, Technical Report, 2012. Disponível em: <https://named-data.net/wp-content/uploads/TRndnsim.pdf>, acesso em 05 de set. de 2022.

AFANASYEV, Alexander. ndnSIM: Simulation Scenario Basics. ACM ICN, 2022. Disponível em:< ndnSIM: Simulation Scenario Basics (Alex Afanasyev) // NDN Tutorial @ ACM I… >, acesso em 20 de abr. de 2023.

BARROS, Diego Guimarães de. NDN-games: uma proposta de arquitetura híbrida com redes NDN para disseminação de conteúdo multimídia em jogos online. Dissertação (Mestrado Acadêmico em Ciência da Computação) - Centro de Ciências e Tecnologia, Universidade Estadual do Ceará. Fortaleza, 2015. Disponível em: <http://www.uece.br/ppgcc/wp-content/uploads/sites/51/2020/02/DIEGO-GUIMARAES-DE-BARROS.pdf>. Acesso em: 29 de jun. de 2022.

BERNERS-LEE, Tim. The Original HTTP as defined in 1991. 1991.

BERNERS-LEE, T. FIELDING, R. FRYSTYK, H. Hypertext Transfer Protocol. 1996.

CERF, Vinton. KAHN, Robert. A Protocol for Packet Network Intercommunication. 1974.

FIELDING, Roy T. et al. Hypertext Transfer Protocol. 1999. Disponível em: <https://datatracker.ietf.org/doc/html/rfc2616>. Acesso em 08 de jul. de 2022.

HOHPE, Gregor. Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. 2015.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. Acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal 2021. Rio de Janeiro: IBGE, 2022. (Coordenação de Pesquisas por Amostra de Domicílios).

IOANNOU, Andriana et al. CLONE: An NDN Architecture for Content Distribution at Remote Tourist Sites - a TCP/IP and NDN Comparison. Dublin, Irlanda, 2018. Disponível em: <https://www.fed4fire.eu/wp-content/uploads/sites/10/2018/09/7.clone-acm-icn-2018-thnp-cameraready.pdf>. Acesso em: 29 de jun. de 2022.

JACOBSON, Van. A New Way to look at Networking. 2006. A New Way to look at Networking Acesso em 24 de jun. de 2022.

NS-3. ns-3 | a discrete-event network simulator for internet systems, 2011. Página inicial. Disponível em: <https://www.nsnam.org/>. Acesso em: 05 de mar. de 2023.

PERLMAN, Radia. Interconnections, 2ª ed. EUA, 2000.

ZHANG, Lixia et al. Named Data Networking. 2014.