

# Balanceamento de Carga Dinâmico, Distribuído e Centrado no Conteúdo para Redes de Dados Nomeados

Francisco R. C. Araújo<sup>1\*</sup>, André L. R. Madureira<sup>1</sup>, Leobino N. Sampaio<sup>1</sup>

<sup>1</sup>Programa de Pós-Graduação em Ciência da Computação (PGCOMP)  
Departamento de Ciência da Computação – Universidade Federal da Bahia (UFBA)  
Salvador – BA – Brasil

{franciscorca, andre.romano, leobino}@ufba.br

**Abstract.** *In Named Data Networks (NDN), the name-based routing model predominantly benefits applications characterized by static content (i.e., content). Traditional NDN in-network caching is not suitable for dynamic content (i.e., services) as services are generated on-demand. Furthermore, each service is also exclusive to each consumer, which lessens NDN in-network caching benefits. Conventional NDN routing strategies are also inadequate for an even service Interest distribution among the available replicas. In such scenarios, load balancing mechanisms can achieve better network utilization by maintaining workload balance among service provider’s replicas. This work presents a load balancing strategy, called B3C, in which each router makes balancing decisions individually, considering their local information and without injecting control traffic. Simulation results show that B3C can evenly balance the load between the replicas and the traffic between the routes.*

**Resumo.** *Em Redes de Dados Nomeados (NDN), o modelo de encaminhamento baseado em nomes beneficia, predominantemente, as aplicações caracterizadas pelo conteúdo estático (i.e., conteúdo). Interesses para conteúdos dinâmicos (i.e., serviços) tendem a não ser satisfeitos em caches, uma vez que os mesmos são gerados sob demanda e exclusivos para dois nós comunicantes. Além disso, estratégias convencionais de encaminhamento não conseguem distribuir Interesses para serviço entre as réplicas disponíveis, tornando necessária a implementação de mecanismos de balanceamento de carga para manter o equilíbrio da carga de trabalho entre as réplicas do provedor de serviço. Este trabalho apresenta uma estratégia para balanceamento de carga em NDN, denominada B3C, em que cada roteador toma decisões de balanceamento individualmente, considerando suas informações locais e sem injetar tráfego de controle. Os resultados de simulações mostram que a B3C consegue balancear a carga entre as réplicas e o tráfego entre as rotas.*

## 1. Introdução

Nas Redes de Dados Nomeados (*Named Data Networking* – NDN), os dispositivos de encaminhamento contam com estratégias para decidir se, quando e para onde encaminhar os pacotes de Interesse recebidos pelas interfaces de entrada com base em informações contidas nas tabelas de encaminhamento (FIB) e dados armazenados localmente em

---

\*Os autores agradecem o apoio da CAPES, CNPq e FAPESB.

*cache* [Zhang et al. 2014]. A tabela de encaminhamento é atualizada a partir dos protocolos de roteamento, tais como o NLSR (*Named Data Link State Routing Protocol*) [Wang et al. 2018], que fornecem informações de alcançabilidade das fontes de dados, enquanto a *cache* é mantida por políticas tradicionais de substituição de conteúdo (e.g., FIFO, LRU e LFU).

A combinação entre estratégias de encaminhamento e armazenamento em *cache* atende de forma satisfatória aplicações caracterizadas por conteúdos estáticos, geralmente requisitados por mais de uma aplicação em diferentes períodos do tempo (e.g., *streaming* de vídeo). Por outro lado, assim como na arquitetura IP, a NDN prevê o suporte à comunicação por chamada de funções nomeadas [Król and Psaras 2017] de rede que apresentam conteúdos dinâmicos<sup>1</sup>. Conteúdos dinâmicos em geral são armazenados na *cache* dos dispositivos, porém não são usados, uma vez que os dados trafegados são relevantes apenas a um único par de pontos comunicantes (i.e., consumidor e produtor) [Mansour et al. 2020]. O armazenamento em *cache*, portanto, resulta em um consumo desnecessário de recursos.

A fim de aumentar a disponibilidade de um serviço, múltiplos provedores (réplicas) de funções nomeadas podem estar disponíveis em um domínio de rede. Contudo, estratégias de encaminhamento convencionais da NDN não são capazes de encaminhar interesses para um serviço disponibilizado por múltiplas réplicas provedoras considerando os requisitos das aplicações finais. A comutação de pacotes na NDN é limitada às informações contidas na FIB, que consiste em um peso atribuído às interfaces de saída, provenientes dos protocolos de roteamento. Ou seja, trata-se de um recurso que restringe a implementação de mecanismos mais sofisticados de encaminhamento, orientados pelas aplicações finais.

Diante deste contexto, este trabalho apresenta uma solução de Balanceamento de Carga Centrado no Conteúdo (B3C) para NDN que explora as características da arquitetura de forma a permitir que serviços de rede possam ser oferecidos com base em requisitos de classes de aplicação. Na solução proposta, cada dispositivo de encaminhamento toma decisões de balanceamento individualmente, levando em consideração informações locais, o tipo de conteúdo trafegado e o esquema de nomeação para direcionar os pacotes às melhores rotas disponíveis de forma equilibrada e oferecendo serviços diferenciados de acordo com classes da aplicação.

Deste modo, as principais contribuições deste trabalho são as seguintes: (i) a apresentação de uma estratégia de balanceamento de carga para NDN, capaz de gerenciar tráfego com base nos conteúdos dinâmicos e estáticos; e (ii) a introdução do conceito de diferenciação de serviços em NDN. Para avaliar a B3C, foram realizados estudos baseados em simulação através do simulador ndnSIM [Mastorakis et al. 2017]. A avaliação consistiu na análise de dois cenários de rede, com baixa e alta cargas de tráfego, comparando o desempenho da proposta com a estratégia *ComVes* [Mansour et al. 2020] e RLB [Al Fuad et al. 2017]. Os resultados obtidos indicam que a B3C consegue atingir seu objetivo de balancear tanto a carga entre as réplicas quanto o tráfego entre as rotas nos dois cenários avaliados.

O restante deste artigo está organizado da seguinte forma: a Seção 2 descreve bre-

---

<sup>1</sup>Neste trabalho estamos denominando conteúdos dinâmicos como aqueles providos por serviços disponibilizados através de funções nomeadas de rede.

vemente os fundamentos da arquitetura NDN e a proposta é apresentada em detalhes na Seção 3. Os experimentos e resultados são discutidos na Seção 4. Os trabalhos relacionados são apresentados na Seção 5. Por fim, o trabalho é concluído na Seção 6.

## 2. Fundamentos da NDN

Antes de discutir sobre o balanceamento de carga na NDN é preciso apresentar os fundamentos dessa arquitetura. Cada nó na NDN possui três estruturas fundamentais: PIT (*Pending Interest Table*), FIB (*Forwarding Information Base*) e CS (*Content Store*) [Zhang et al. 2014]. A PIT mantém os Interesses que foram encaminhados e suas interfaces de entrada e saída, sendo que a interface de entrada serve para encaminhar o dado recuperado e a interface de saída para fins de medições e detecção de *loop*; a FIB é responsável por manter a alcançabilidade das fontes de dados; e a CS serve como armazenamento temporário de dados recuperados para servir a futuros Interesses correspondentes a esses dados.

### 2.1. Conteúdos *versus* Serviços

Na NDN todo pacote de Interesse requisita um pacote de dados. Portanto, a terminologia “dados” é atribuída ao pacote que responde ao Interesse. Quando um consumidor requisita um conteúdo estático, dizemos que sua requisição é um pacote de *Interesse para conteúdo*, por outro lado, quando a requisição do consumidor é para um conteúdo dinâmico, dizemos que sua requisição é um pacote de *Interesse para serviço*. Ambos os Interesses recuperam uma *porção de dados*<sup>2</sup>. O conteúdo estático (i.e., conteúdo) é todo aquele que é gerado e publicado pelo produtor, como um objeto pré-existente [Mansour et al. 2020]. Enquanto isso, conteúdo dinâmico (i.e., serviço) requer computação, sendo gerado sob demanda, e implementado por uma função de rede nomeada.

Os pacotes de dados recuperados pelos Interesses para conteúdo podem ser armazenados e recuperados na *cache* dos elementos da rede. Por outro lado, apesar dos dados recuperados pelos Interesses para serviço também serem armazenados na *cache*, é altamente improvável que estes dados sejam recuperados da mesma, uma vez que os serviços são exclusivos a cada consumidor.

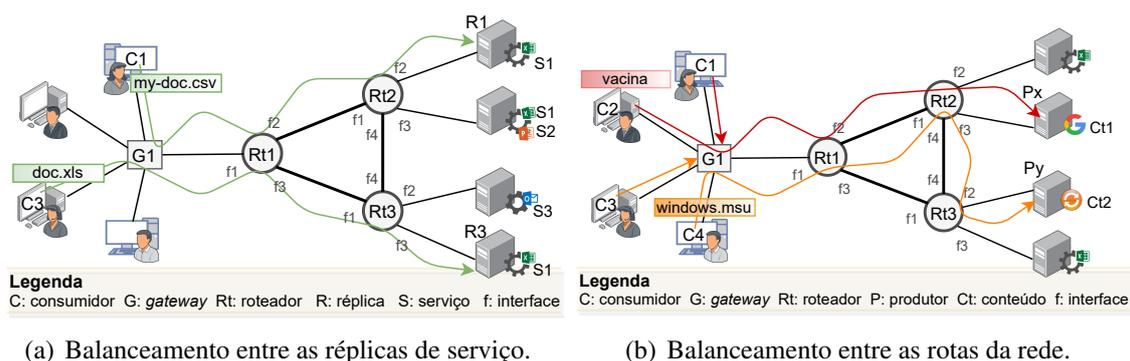
### 2.2. Balanceamento de carga entre réplicas e rotas na NDN

A Figura 1 apresenta uma visão geral do balanceamento de carga entre as réplicas de serviço e rotas da rede. Na Figura 1(a), é importante notar que os serviços podem e devem, de acordo com a demanda, ser hospedados em diferentes réplicas para atender as requisições. Mesmo que um serviço venha a se tornar popular, os Interesses para o serviço não necessariamente serão populares, já que normalmente possuem nomes diferentes [Mansour et al. 2020]. Neste exemplo, o consumidor C1 requisitou um Interesse para o serviço `my-doc.csv`, enquanto C3 requisitou `doc.xls`. Apesar dos consumidores acessarem o mesmo serviço, cada consumidor apresenta um uso particular do mesmo. Dessa forma, mesmo que os dados sejam armazenados em *cache* é improvável que o Interesse possa encontrar um dado correspondente no caminho. Se o *gateway* G1 possuir o dado de C1 armazenado em *cache*, quando C3 emitiu um Interesse para o serviço, a *cache* de G1 não consegue satisfazer o Interesse de C3. Dada essa especificidade de Interesses

---

<sup>2</sup>Tradução para a expressão em inglês “*piece of data*”, comumente utilizada pela literatura especializada.

para serviço, as requisições dos consumidores necessitam alcançar uma réplica na rede, e.g.,  $C1 \rightarrow G1 \rightarrow Rt1 \rightarrow Rt2 \rightarrow R1S1$ , para serem satisfeitas. Portanto, os Interesses para serviço costumam percorrer toda a rede até alcançar uma réplica. Já na Figura 1(b), o consumidor C2 requisitou o conteúdo `vacina` e obteve o resultado da consulta. Após isso, o consumidor C1 requisitou o mesmo conteúdo, a requisição de C1 não precisa chegar até a fonte dos dados, pois pode encontrá-los na *cache* de G1. Uma situação semelhante ocorre entre C4 e C3. C4 obteve o conteúdo `windows.msu` e, em seguida, C3 requisita o mesmo conteúdo, o qual pode ser satisfeito pela *cache* de G1.



**Figura 1. Balanceamento de carga entre réplicas de serviço e rotas na NDN.**

Conforme ilustrado pela Figura 1, Interesses para serviço sempre precisam alcançar uma réplica. A fim de evitar sobrecargas, o encaminhamento das requisições devem ser balanceados entre elas. Este balanceamento também distribui melhor o consumo dos recursos de rede, uma vez que naturalmente ocorrerá o balanceamento entre as múltiplas rotas existentes, assim como acontece com os Interesses para conteúdo. Esse comportamento evidencia a necessidade de um mecanismo de balanceamento de carga eficiente. As réplicas devem ser espalhadas em diferentes pontos, e uma mesma réplica produtora pode servir a mais de um serviço. O que define o serviço ofertado pela réplica é o espaço de nome anunciado por ela, através do protocolo de roteamento.

### 2.3. Roteamento e Encaminhamento na NDN

Análogo ao roteamento e repasse do TCP/IP, o roteamento e encaminhamento na NDN visa estabelecer a comunicação entre máquinas, de maneira eficaz e padronizada. Contudo, diferente do TCP/IP, na NDN não há o conceito de comunicação ponto-a-ponto, nem de conexões. Nesses ambientes, cada pacote recebe um nome único, atribuído pela aplicação, que é usado pelos elementos da rede para encaminhamento [Zhang et al. 2018]. Na NDN o protocolo de roteamento apenas atualiza a FIB dos nós na rede com as informações de alcançabilidade das fontes de dados, enquanto a estratégia de encaminhamento é que determina se, quando e para qual rota o pacote de Interesse deve ser encaminhado [Zhang et al. 2014]. A comunicação na NDN é conduzida pelos consumidores (i.e., modelo *receiver-driven*) através de requisições de Interesses, que preenchem a PIT dos nós permitindo que os dados recuperados retornem pela mesma rota utilizada pelos Interesses [Zhang et al. 2014]. Portanto, as estratégias de encaminhamento têm um papel central na NDN, assim como, os requisitos das aplicações, sendo esses os pontos explorados pela proposta B3C, apresentada na próxima seção.

### 3. Estratégia B3C

A B3C tem como objetivo principal balancear a carga de pacotes de Interesses para serviço entre as réplicas disponíveis. Também visa balancear o tráfego entre as rotas. Além disso, introduz o conceito de classes de tráfego para que os roteadores NDN possam tomar decisões de encaminhamento baseadas no tipo de conteúdo. Nas próximas subseções são apresentados os detalhes da proposta.

#### 3.1. Visão geral

A estratégia B3C se baseia nas medições de Interesses encaminhados por prefixo e interface para estimar a carga de trabalho nas réplicas de serviço. Cada roteador mantém as medições de Interesses encaminhados. Dessa forma, a B3C consegue balancear tanto o tráfego de pacotes de Interesse para serviço quanto Interesse para conteúdo entre as réplicas e rotas disponíveis. Os roteadores adotam um mecanismo de fila circular com as interfaces de saída de cada prefixo de nome. A cada novo Interesse recebido, são verificadas as possíveis interfaces de saída para atualizar a fila. As aplicações adicionam nos pacotes de Interesse o tipo de conteúdo requisitado para auxiliar os roteadores a encaminhá-los para a rota que melhor atenda seus requisitos. Os tipos de conteúdo são definidos como classes de tráfego.

#### 3.2. Hierarquia de Nomes e Classes de Tráfego

Ao contrário da arquitetura TCP/IP, os pacotes da NDN são identificados por nomes exclusivos e hierarquicamente estruturados na aplicação. A aplicação e a camada de rede compartilham o mesmo espaço de nome (prefixo). Os nomes são usados pelos encaminhadores na rede para direcionar as requisições aos dados e pelos *hosts* para demultiplexar as solicitações para o processo correto na aplicação [Zhang et al. 2018].

Na NDN as estratégias de encaminhamento são configuradas localmente pelo operador do sistema e seguem o nível de granularidade por prefixo [NFD 2018]. Por exemplo, um operador configurou a estratégia *Multicast* para lidar com o prefixo `/ufba`. Todos os Interesses sob esse prefixo serão encaminhados pela estratégia *Multicast*. Caso haja a necessidade de os Interesses do subdomínio `/ufba/dcc` sejam encaminhados usando outra estratégia, o operador terá de configurar a estratégia pretendida para o prefixo do subdomínio. Isso se repetirá para cada nova necessidade de subdomínio que surgir. O problema é que esse nível de granularidade não se mostra eficiente quando uma organização fornece serviços diferentes sob o mesmo espaço de nome, e.g., `/ufba/dcc/docs`, `/ufba/dcc/videos`, `/ufba/dcc/www`, etc. Essa necessidade de configuração para cada subdomínio de cada organização pode gerar uma quantidade enorme de entradas em um arquivo de configuração que deve conter todos os subdomínios de todas as organizações cujo tráfego passa por aquela rede. Ainda assim, o nível de granularidade será inadequado à necessidade real do tráfego, uma vez que uma aplicação em um determinado subdomínio pode requerer uma necessidade de desempenho maior para o seu tráfego gerado. Um segundo problema em relação ao encaminhamento da NDN baseado estritamente em prefixo é que um mesmo pacote de Interesse pode ser encaminhado por estratégias diferentes em cada roteador, pois os roteadores são configurados localmente de forma independente [NFD 2018]. Como cada estratégia de encaminhamento adota uma decisão particular para encaminhar os pacotes recebidos, os Interesses podem seguir caminhos com requisitos completamente diferentes do esperado pela aplicação.

Para resolver as questões apresentadas, a B3C introduz o conceito de classes de tráfego para NDN. As classes de tráfego podem ser úteis para a tomada de decisão nos seguintes módulos da arquitetura: (i) Estratégias de encaminhamento – as classes podem ser usadas em técnicas de balanceamento de carga, controle de congestionamento, qualidade de serviço e engenharia de tráfego; (ii) Políticas de *cache* – as classes podem auxiliar na tomada de decisão de políticas de colocação de dados e nas políticas de substituição; (iii) Gerenciamento e monitoramento – as classes podem ser aplicadas para gerenciar e/ou monitorar o tráfego de acordo com o tipo de conteúdo; e (iv) Segurança – as classes são uma informação a mais para somar aos mecanismos de identificação e mitigação de ataques de negação de serviço por inundação de Interesses falsos ao qual a NDN apresenta vulnerabilidade.

Uma gama de classes podem ser definidas de acordo com os requisitos exigidos pelas aplicações. Trata-se de uma discussão que não está no escopo deste trabalho. Aqui, apresentamos uma classificação para fins de experimentação e prova de conceito do tratamento dado pela NDN através da B3C, uma vez as classes criadas. A Tabela 1 apresenta as classes de tráfego avaliadas quanto aos seus requisitos mínimos exigidos. A coluna ToC define o tipo de conteúdo representado pela classe. O ToC é um campo adicionado no cabeçalho do pacote de Interesse, algo semelhante ao uso do *Type of Service* (ToS) [Nichols et al. 1998] do cabeçalho IP.

**Tabela 1. Categorização inicial de classes de tráfego.**

ToC	Descrição	Caching	Sensível ao tempo	Multi-source
CT1	Conteúdo estático	✓	✗	✗
CT2	Conteúdo de <i>streaming</i>	✗	✓	✗
CT3	Conteúdo dinâmico	✗	✗	✓

### 3.3. Estrutura e Processamento dos Pacotes de Interesse

A B3C separa o tráfego de pacotes de Interesse em classes usando o campo *Type of Content* (ToC) adicionado no cabeçalho desses pacotes. O ToC foi inserido como uma *tag* no pacote de Interesse usando a estrutura especializada *Packet Tag*<sup>3</sup>, oficial da NDN. Dessa forma, a estrutura geral do pacote de Interesse se manteve constante e a B3C não altera a arquitetura NDN, nem adiciona tráfego de controle na rede, além de ser mais fácil de implementar, manter e incorporar às soluções NDN existentes.

Como a B3C não modifica nenhum componente da arquitetura, além de usar os recursos já previstos pela mesma, o processamento de pacotes se mantém o padrão em todos os seus componentes. Para manter as medições da quantidade de Interesses encaminhados por prefixo e interface de saída, a B3C usa a estrutura especializada *Measurements Table* presente no módulo NFD (*NDN Forwarding Daemon*) da NDN [NFD 2018]. Quanto ao roteamento, a B3C usou o protocolo de roteamento baseado em estado de enlace LFID (*Loop-Free Inport-Dependent*) [Schneider et al. 2019], que fornece um conjunto de caminhos livres de *loop*, para realizar o cálculo de rotas e a atualização da FIB dos nós.

<sup>3</sup><https://redmine.named-data.net/projects/ndn-cxx/wiki/PacketTagTypes>

### 3.4. Funcionamento do balanceamento de carga da B3C

O Algoritmo 1 apresenta as ações que ocorrem em um nó usando a B3C após a recepção de um pacote de Interesse. Na Função `ApósReceberInteresse` é calculado o *backoff* exponencial da entrada PIT para decidir se encaminha ou não o Interesse recebido. O cálculo do *backoff* usa a expressão  $Intervalo_{supressão} = Min(intervalo_{Inicial} \times multiplicador^{i-1}, intervalo_{Max})$ , onde  $intervalo_{Inicial}$  recebe 1 ms,  $multiplicador$  recebe 2,  $i$  é a  $i$ -ésima retransmissão e o  $intervalo_{Max}$  recebe 250 ms. Se a última transmissão ocorreu dentro do  $Intervalo_{supressão}$ , a retransmissão será suprimida. Caso a decisão do cálculo do *backoff* for diferente de suprimir o Interesse, então o mesmo pode ser encaminhado. Após isso, é preciso verificar a classe de tráfego do Interesse para decidir a melhor rota para encaminhá-lo.

A Função `EncaminharComoCT3` apresenta um exemplo de encaminhamento usando a classe de tráfego CT3. Nessa função, são iniciadas as medições por prefixo, em seguida, é gerada uma tabela temporária com as informações de ocupação das interfaces de saída para o prefixo na PIT. Há uma iteração nas entradas do prefixo recuperadas da FIB para atualizar a tabela temporária, caso alguma interface ainda não esteja presente na mesma (linhas 15-17). A interface de menor custo da tabela temporária é obtida para atualizar as interfaces da estrutura fila de interfaces presente na medição do prefixo (linhas 18-21). A fila contendo as interfaces é percorrida em busca de uma interface apta a encaminhar o Interesse, após o Interesse ser encaminhado a fila é atualizada (linhas 22-27). Por fim, caso o Interesse não possa ser encaminhado, um pacote NACK é enviado e a entrada PIT é rejeitada (linhas 28-30).

Diante do exposto, a B3C consegue melhorar a decisão de encaminhamento da NDN, uma vez que as aplicações podem informar no campo ToC o nível do desempenho esperado para o encaminhamento dos Interesses emitidos. Ao receber pacotes de Interesse, cada roteador irá interpretar o ToC e tomar a melhor decisão de encaminhamento para o pacote. Isso permite que os roteadores, mesmo atuando de forma distribuída e apenas com suas informações locais, possam trabalhar em sincronia encaminhando os Interesses para as rotas mais adequadas de acordo com os requisitos da aplicação.

## 4. Estudo Experimental e Análise dos Resultados

### 4.1. Ambiente de Experimentação

Os experimentos foram conduzidos através de simulação utilizando o simulador de redes oficial do projeto NDN, baseado em eventos discretos, `ndnSIM` [Mastorakis et al. 2017] na versão<sup>4</sup> mais recente. O cenário de avaliação da proposta está representado na Figura 2. Na figura há quatro consumidores C1 a C4, dentre os quais foram divididos em dois grupos: (i) C1 e C2 requisitam Interesses para serviço às réplicas produtoras de serviço P1, P2 e P3; e (ii) os consumidores C3 e C4 requisitam Interesses para conteúdo ao produtor P4. Todos os consumidores são conectados ao núcleo via enlace de 1 Mbps com atraso de 5 ms. O núcleo é formado por três roteadores – Rt1, Rt2 e Rt3 – interconectados através de enlaces de 10 Mbps / 10 ms. Os produtores são conectados ao núcleo através de enlaces de 1 Mbps / 1 ms.

Todos os nós foram configurados com uma *cache* com capacidade de armazenamento de 100 pacotes de dados. A política de substituição de *cache* LRU foi utilizada

<sup>4</sup>`ndnSIM` v 2.8, disponível em: <https://ndnsim.net/current/>

---

**Algoritmo 1:** Processamento de pacotes de Interesse pela B3C.

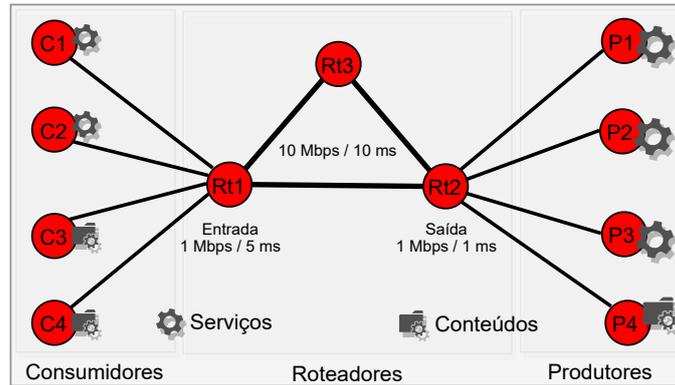
---

```
1 Função ApósReceberInteresse (Interesse, entradaPIT) :
2   decisão = BACKOFFEXPONENCIAL(entradaPIT)
3   se decisão == suprimir então
4     | retornar
5   se Interesse.GETTOC == CT1 então
6     | retornar ENCAMINHARCOMOCT1(Interesse, entradaPIT)
7   se Interesse.GETTOC == CT2 então
8     | retornar ENCAMINHARCOMOCT2(Interesse, entradaPIT)
9   se Interesse.GETTOC == CT3 então
10    | retornar ENCAMINHARCOMOCT3(Interesse, entradaPIT)
11 Função EncaminharComoCT3 (Interesse, entradaPIT) :
12   medição = ADDMEDIÇÕESPREFIXO(entradaPIT)
13   tabelaOcupaçãoTmp = GERARTABELA(entradaPIT)
14   entradaFIB = PESQUISAFIB(entradaPIT)
15   // insere na tabelaOcupaçãoTmp as outfaces ausentes
16   para cada interface de saída da entradaFIB faça
17     | se interface não está na tabelaOcupaçãoTmp então
18       | | tabelaOcupaçãoTmp.INSERIR(interface, 0)
19   // insere na filaMinFaces as faces com valores menores
20   minInterface = GETMINOUTFACE(tabelaOcupaçãoTmp)
21   para cada entrada na tabelaOcupaçãoTmp faça
22     | se entrada.valor == minInterface.valor e entrada ∉ medição.filaMinFaces então
23       | | medição.filaMinFaces.INSERIRNOINÍCIO(entrada.face)
24   // encaminha Interesse para próxima interface
25   para cada entrada na medição.filaMinFaces faça
26     | se entrada não viola escopo então
27       | | ENVIARINTERESSE(entradaPIT, entrada.face, Interesse)
28       | | medição.filaMinFaces.REMOVERFACE(entrada.face)
29       | | medição.filaMinFaces.INSERIRNOFIM(entrada.face)
30       | | retornar
31   se Interesse não foi enviado então
32     | ENVIARNACK(entradaPIT)
33     | REJEITARINTERESSE(entradaPIT)
```

---

para gerenciar a *cache* em cada nó. Os dois grupos de consumidores (i.e., (i) consumidores de serviço e (ii) consumidores de conteúdo) emitiram Interesses usando uma aplicação de taxa de *bit* constante (CBR). Os consumidores do grupo (i) emitiram Interesses sob o prefixo /ufba/dcc/{1 ... n} e os consumidores do grupo (ii) emitiram Interesses sob o prefixo /ufba/deec/{1 ... n}, onde *n* representa a quantidade de Interesses emitidos. O primeiro consumidor de cada grupo (i.e., C1 e C3) inicia ao mesmo tempo, logo no início da simulação, e após 0,5 segundos iniciam os consumidores C2 e C4. A razão disso, é permitir que o segundo consumidor de cada grupo tenha uma chance de encontrar o conteúdo solicitado na *cache* mais próxima sem a necessidade de seu Interesse ter que alcançar o produtor para recuperar o conteúdo solicitado.

Para simular o comportamento dos Interesses para serviço (i.e., a improvável repe-



**Figura 2. Topologia de avaliação da proposta. Os consumidores C1 e C2 requisitam Interesses para serviço às réplicas P1, P2 e P3. Já os consumidores C3 e C4 requisitam Interesses para conteúdo ao produtor P4.**

tição do nome do Interesse entre diferentes consumidores) desenvolvemos uma aplicação que permite cada consumidor de serviço emitir Interesses com nomes específicos sob o prefixo /ufba/dcc/ para dificultar que Interesses para serviço tenham o nome repetido. A aplicação desenvolvida para os produtores de serviço responde as requisições recebidas em um tempo randômico que varia de 0 a 25 ms, simulando o processamento do serviço. A Tabela 2 sumariza os parâmetros utilizados nos experimentos, os parâmetros ausentes na tabela assumiram valores padrão do ndnSIM.

**Tabela 2. Parâmetros da Simulação.**

Parâmetro	Valor
Tamanho do pacote de dados	1024 <i>bytes</i>
Capacidade da <i>cache</i>	100
Política de <i>cache</i>	LRU
Tempo de vida do Interesse	4 s
Duração de cada simulação	60 s
Variação para responder serviço	0 a 25 ms
Enlaces dos Consumidores / atraso	1 Mbps / 5 ms
Enlaces do núcleo / atraso	10 Mbps / 10 ms
Enlaces dos Produtores / atraso	1 Mbps / 1 ms
Estratégias comparadas	B3C, <i>ComVes</i> e RLB

Nos experimentos foram avaliados dois casos de teste. No primeiro caso, o cenário foi avaliado com uma carga baixa, onde cada consumidor emitiu 10 Interesses por segundo. Já no segundo caso, cada consumidor emitiu uma carga maior, de 50 Interesses por segundo. Cada experimento foi replicado 10 vezes com diferentes sementes para obter a média com um nível de confiança<sup>5</sup> de 95%. A proposta B3C foi comparada com a estratégia de balanceamento de carga *ComVes* [Mansour et al. 2020] e a estratégia randômica RLB [Al Fuad et al. 2017]. É importante destacar que para ambas as soluções avaliadas foram replicadas as mesmas especificações do ambiente de simulação.

<sup>5</sup>Utilização da distribuição *t-student* e  $\alpha = 5\%$ .

## 4.2. Métricas de Desempenho

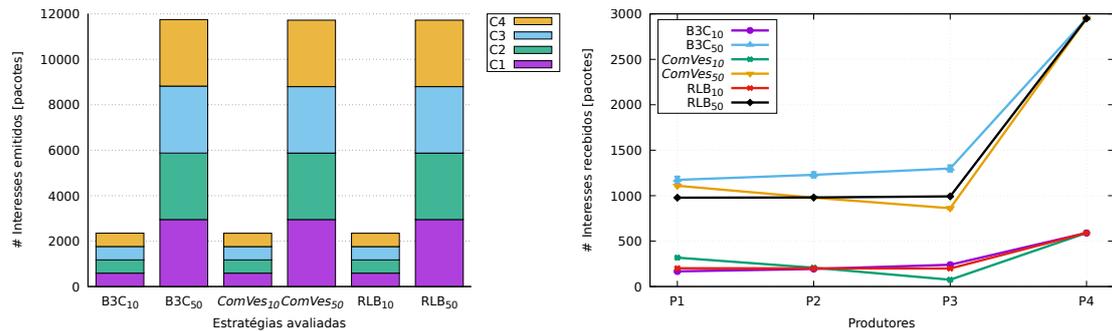
As seguintes métricas foram consideradas na análise de desempenho: (i) Média de Interesses requisitados (pacotes) – consiste na média dos Interesses requisitados por cada consumidor; (ii) Média de Interesses recebidos (pacotes) – consiste na média dos Interesses recebidos por cada produtor. Necessária para aferir a carga de trabalho em cada produtor e mostrar a eficiência da estratégia ao distribuir a carga recebida entre as réplicas de serviço; (iii) Média de acerto de *cache* (pacotes) – representa a média de Interesses que podem encontrar os dados na *cache* dos roteadores. O acerto de *cache* reflete apenas o tráfego de Interesses para conteúdo; (iv) Taxa de ocupação da PIT (pacotes/s) – representa a taxa de Interesses que chegaram no roteador e aguardam pelos dados requisitados. A presença de Interesses na PIT é um indicativo que a carga está sendo encaminhada pelo caminho ao qual o roteador está localizado; (v) Taxa de satisfação de Interesse (pacotes/s) – consiste na média dos pacotes de Interesse, emitidos pelos nós consumidores, que conseguiram recuperar os dados solicitados; e (vi) Atraso médio (s) – representa a média dos atrasos que os Interesses emitidos pelos consumidores levam para recuperar os dados solicitados. Esse tempo pode incluir possíveis *timeouts* e retransmissões de Interesses.

## 4.3. Resultados e Discussões

A média geral da taxa de Interesses requisitados por cada consumidor e a soma dessas médias estão representadas na Figura 3(a). Na figura, nota-se a consistência na média de envio de Interesses das aplicações dos consumidores, de acordo com as especificações dos cenários de teste, com requisições individuais de 10 Interesses por segundo no cenário 1 e 50 Interesses por segundo no cenário 2. É importante destacar que em ambos os cenários de teste, os consumidores C1 e C2 enviam Interesses para serviço às réplicas P1, P2 e P3, enquanto os consumidores C3 e C4 requisitam Interesses para conteúdo ao produtor P4.

A média da quantidade de Interesses recebidos nos produtores está representada na Figura 3(b). Nela, observa-se que no cenário de teste 1, i.e., com requisições individuais de 10 Interesses por segundo, as estratégias RLB e B3C apresentam uma taxa de balanceamento de carga bastante equilibrada entre as réplicas (P1, P2 e P3), enquanto a *ComVes* apresenta um balanceamento desequilibrado entre P1 e P3. Nesse caso, a *ComVes* distribuiu uma carga maior para P1, enquanto P3 recebeu uma carga menor em relação as demais réplicas. O mesmo fato ocorre no cenário de teste 2. Novamente, RLB e B3C apresentam uma distribuição de carga mais equilibrada entre as réplicas, enquanto a *ComVes* sobrecarrega P1 em relação a P3. O produtor P4 que responde aos Interesses para conteúdo de C3 e C4 apresentou um resultado semelhante em ambas as estratégias. Isso se deve ao fato de não haver réplica para o mesmo. Quanto a diferença de valores absolutos em cada réplica por estratégia, possíveis causas são as retransmissões de Interesses e perdas de pacotes nas filas dos roteadores. Por exemplo, se um Interesse alcança o produtor e recupera o conteúdo, mas o roteador não consegue entregar o conteúdo ao consumidor, haverá *timeout* e o consumidor novamente emitirá um Interesse para o conteúdo. Como Interesses para serviço são improváveis de serem recuperados na *cache*, esses Interesses terão que alcançar o produtor, contabilizando um novo Interesse recebido. Pretendemos investigar esse fenômeno em trabalhos futuros.

A Figura 4(a) mostra a média de acertos na *cache* dos roteadores. Essa figura reflete o tráfego de Interesses para conteúdo, já que Interesses para serviço são improváveis



(a) Média de Interesses emitidos nos consumidores. (b) Média de Interesses recebidos nos produtores.

**Figura 3. Média de Interesses emitidos e recebidos.**

de serem satisfeitos em *cache*, apesar de serem armazenados nela. O roteador Rt1 obteve uma média de acertos maior em relação aos outros roteadores em ambas as estratégias, nos cenários 1 e 2. Isso se deve ao fato de Rt1 ser o primeiro roteador conectado aos consumidores, como mostra a Figura 2. Ao analisar o cenário 1, Rt1 obteve uma mesma média em todas as estratégias, enquanto Rt2 e Rt3 não apresentaram acertos. Esse comportamento se deve ao fato de Rt1 ser capaz de responder os Interesses para conteúdo através de sua *cache* impedindo que esses Interesses alcancem os demais roteadores, pois a *cache* de Rt1 é maior que a soma da carga dos consumidores de Interesses para conteúdo. Já no cenário 2, a carga de Interesses para conteúdo é maior que a *cache* de Rt1, portanto a *cache* de Rt1 não consegue satisfazer todos os Interesses antes de alguns dos conteúdos serem substituídos pela política de *cache*. Nesse cenário, apenas a estratégia B3C consegue obter acerto de *cache* em Rt2 e Rt3, o que demonstra que B3C consegue também balancear tráfego de Interesses para conteúdo.

A Figura 4(b) mostra a taxa de ocupação da PIT nos roteadores da rede. No caso de teste com carga baixa, onde cada consumidor requisitou 10 Interesses por segundo, as três estratégias apresentam desempenho similar, mas com uma leve tendência de crescimento da ocupação da PIT dos roteadores Rt1 e Rt3 por parte das estratégias B3C e RLB. No caso de teste com carga maior, onde cada consumidor requisitou 50 Interesses por segundo, a tendência de crescimento de ocupação da PIT de Rt1 e Rt3 se mantém, novamente pelas estratégias B3C e RLB. Contudo, a B3C além de apresentar maior uso da PIT de Rt1 e Rt3, mostra uma queda da ocupação da PIT de Rt2. Isso demonstra que a B3C consegue balancear a carga também entre os roteadores, além dos produtores. Ao analisar a topologia de experimentação, na Figura 2, nota-se que Rt3 é um caminho auxiliar ao caminho Rt1  $\rightarrow$  Rt2 e esse caminho auxiliar é pouco explorado pela *ComVes* e RLB.

A Figura 5(a) retrata a taxa de Interesses satisfeitos em cada consumidor. Os consumidores conseguem recuperar todos os dados requisitados, nos cenários 1 e 2. Em ambos os cenários, e nas três estratégias, os consumidores C2 e C4 apresentam uma leve redução da taxa de Interesses satisfeitos. Isso se deve ao fato desses consumidores iniciarem suas requisições 0,5 segundos após os consumidores C1 e C3, respectivamente. Já a Figura 5(b) representa o atraso percebido pelos consumidores na recuperação de dados. As três estratégias obtiveram atrasos bastante semelhantes nos dois cenários avaliados. Os Interesses para serviço que são requisitados pelos consumidores de serviço C1 e C2,

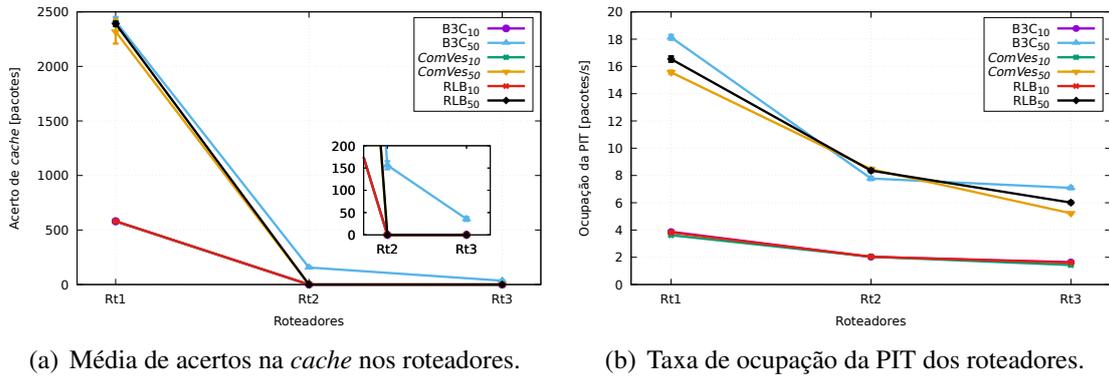


Figura 4. Média de acertos na *cache* e taxa de ocupação da PIT nos roteadores.

são Interesses exclusivos de cada consumidor e não se beneficiam do mecanismo de *cache* como os Interesses para conteúdo de C3 e C4. O consumidor C4, como inicia após C3, obtém um menor atraso para recuperar dados, pois pode encontrar os dados em *cache* sem ter que alcançar o produtor, diminuindo o atraso.

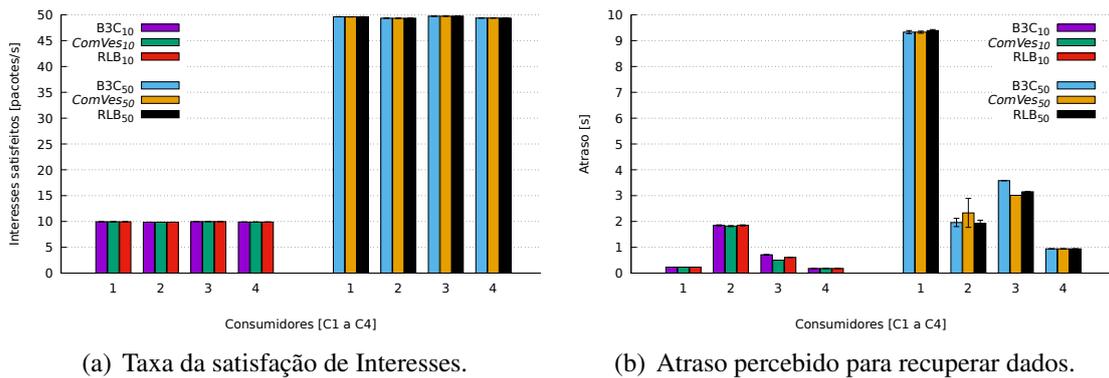


Figura 5. Taxa da satisfação de Interesses e atraso para recuperar dados nos consumidores.

## 5. Trabalhos Relacionados

Nesta seção são abordados os trabalhos estritamente relacionados ao balanceamento de carga em NDN. Estratégias que abordam a temática indiretamente como controle de congestionamento [Zafar et al. 2020] e engenharia de tráfego [Li et al. 2018] fogem do escopo deste trabalho.

Em [Mansour et al. 2020], os autores apresentam um extenso estudo da literatura sobre mecanismos de balanceamento de carga em redes e propõem uma nova abordagem de balanceamento de carga para NDN, chamada *ComVes* (*Communicating Vessels*). A *ComVes* atua nos roteadores que têm a função de balancear a carga usando o tamanho da fila de trabalho como uma métrica para a tomada de decisões de encaminhamento, a fim de equilibrar a carga nas réplicas de serviço. Dessa forma, a *ComVes* foi projetada para lidar apenas com o tráfego de Interesses para serviço. Este fato é a principal diferença entre *ComVes* e a estratégia B3C que lida com tráfego variado.

Em [Bhat et al. 2015], os autores propuseram um mecanismo de balanceamento de carga para NDN com foco no tráfego multimídia (*streaming* de vídeo). O mecanismo é baseado no balanceamento probabilístico que depende da carga atual na PIT do nó para aceitar pacotes de Interesse. Essa característica pode fazer com que alguns Interesses não sejam adicionados na PIT, quando a quantidade de entradas for superior a um limiar predeterminado, e portanto não sejam respondidos. Diferente de [Bhat et al. 2015], o trabalho proposto lida com tráfego variado e a rejeição de Interesse acontecerá apenas no caso extremo, quando o limite de inserção de novas entradas na PIT for excedido.

Em [Takemasa et al. 2020] foi proposto um mecanismo de balanceamento de carga a nível de *thread* nos roteadores NDN visando acelerar o desempenho de processamento dos roteadores. O mecanismo é baseado no envio de Interesses altamente populares através de múltiplas *threads*, enquanto despacha outros Interesses de acordo com a fragmentação (i.e., Interesses separados em *threads* de acordo com o nome do pacote). Enquanto isso, em nossa proposta o balanceamento ocorre a nível de rotas de encaminhamento com o objetivo de acelerar o fluxo de pacotes.

Em [Kasmar and Haryadi 2017, Al Fuad et al. 2017] a estratégia de balanceamento é randômica e ocorre apenas em um nó, posicionado entre o consumidor e o produtor. Um estudo de caso simplificado usando a topologia real *Palapa Ring* simulada foi proposto por [Kasmar and Haryadi 2017]. Já em [Al Fuad et al. 2017], foi proposto um algoritmo denominado RLB (*Random Load Balancer*). RLB atua em um nó que mantém a informação de carga média dos produtores e seleciona randomicamente um produtor após receber um Interesse. Ao ser selecionado localmente, o produtor tem sua carga verificada e, caso não esteja sobrecarregado, o pacote pode ser encaminhado. Caso contrário, o balanceador repete o processo de seleção em busca de um produtor subutilizado.

## 6. Conclusão e Trabalhos Futuros

A NDN se apresenta como uma arquitetura de rede promissora que muda o modelo de comunicação de entrega de pacotes a destinos identificados por endereços IP para a busca de pacotes por nomes. No entanto, na presença de Interesses para serviço, os roteadores NDN não conseguem se beneficiar de seus *caches* para estes Interesses. Além disso, esses Interesses necessitam alcançar uma réplica na rede. Uma vez que as estratégias de encaminhamento convencionais não conseguem balancear a carga entre as réplicas do serviço, mecanismos de balanceamento tornam-se necessários. Neste estudo, é apresentada a proposta B3C para o balanceamento de carga em NDN. A B3C atua como uma estratégia de encaminhamento em cada roteador NDN e explora as características da arquitetura para balancear o tráfego entre as rotas e a carga entre as réplicas. A B3C também introduz o conceito de diferenciação de serviço em NDN, onde as aplicações informam o tipo de conteúdo em suas requisições. Dessa forma, os roteadores são auxiliados, através dessa informação, a cumprir os requisitos esperados pela aplicação. Como trabalhos futuros, pretende-se explorar os mecanismos de controle de congestionamento para integrar na solução proposta e uma política de *cache* ciente de tráfego também deve ser desenvolvida. Além disso, será investigado o uso da B3C com outros protocolos de roteamento.

## Referências

Al Fuad, M. A., Sabuj, M. S. S., Hasan, M. Z., and Naznin, M. (2017). Rlb: Randomized

- load balanced packet forwarding strategy in name based data networking. In *2017 4th International Conf. on Networking, Systems and Security (NSysS)*, pages 1–4. IEEE.
- Bhat, D., Wang, C., Rizk, A., and Zink, M. (2015). A load balancing approach for adaptive bitrate streaming in information centric networks. In *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6.
- Kasmar, A. F. and Haryadi, S. (2017). Delay analysis on load balancing ndn, case study: Palapa ring. In *2017 3rd International Conference on Wireless and Telematics (ICWT)*, pages 186–190. IEEE.
- Król, M. and Psaras, I. (2017). Nfaas: Named function as a service. In *Proceedings of the 4th ACM Conference on Information-Centric Networking, ICN '17*, page 134–144, New York, NY, USA. ACM.
- Li, J., Luo, H., Zhang, S., Yu, S., and Wolf, T. (2018). Traffic engineering in information-centric networking: Opportunities, solutions and challenges. *IEEE Communications Magazine*, 56(11):124–130.
- Mansour, D., Osman, H., and Tschudin, C. (2020). Load balancing in the presence of services in named-data networking. *Journal of Network and Systems Management*, 28:298–339.
- Mastorakis, S., Afanasyev, A., and Zhang, L. (2017). On the Evolution of ndnSIM: An Open-Source Simulator for NDN Experimentation. *ACM SIGCOMM Computer Communication Review*, 47(3):19–33.
- NFD (2018). Nfd developer’s guide. Disponível em: <https://named-data.net/publications/techreports/ndn-0021-10-nfd-developer-guide>. Último acesso em: 16 de abril de 2021.
- Nichols, K., Blake, S., Baker, F., and Black, D. (1998). Definition of the differentiated services field (ds field) in the ipv4 and ipv6 headers. RFC 2474, RFC Editor. Disponível em: <https://tools.ietf.org/html/rfc2474>.
- Schneider, K., Zhang, B., and Benmohamed, L. (2019). Hop-by-hop multipath routing: Choosing the right nexthop set. Technical Report NDN-0067, NDN.
- Takemasa, J., Tagami, A., Koizumi, Y., and Hasegawa, T. (2020). Load balancing for stateful forwarding by mitigating heavy hitters: A case for multi-threaded ndn software routers. *IEEE Access*, 8:155071–155085.
- Wang, L., Lehman, V., Hoque, A. M., Zhang, B., Yu, Y., and Zhang, L. (2018). A secure link state routing protocol for ndn. *IEEE Access*, 6:10470–10482.
- Zafar, H., Abbas, Z. H., Abbas, G., Muhammad, F., Tufail, M., and Kim, S. (2020). An effective fairness scheme for named data networking. *Electronics*, 9(5):749.
- Zhang, H., Li, Y., Zhang, Z., Afanasyev, A., and Zhang, L. (2018). Ndn host model. *SIGCOMM Comput. Commun. Rev.*, 48(3):35–41.
- Zhang, L., Afanasyev, A., Burke, J., Jacobson, V., Claffy, K., Crowley, P., Papadopoulos, C., Wang, L., and Zhang, B. (2014). Named data networking. *ACM SIGCOMM Computer Communication Review*, 44(3):66–73.