Roteamento em Redes de Dados Nomeados com NDVR: um protocolo leve e eficiente para disseminação de informações de alcançabilidade

Italo Valcy S. Brito[1], **Leobino N. Sampaio**[1]

[1]Programa de Pós-Graduação em Ciência da Computação (PGCOMP)
Departamento de Ciência da Computação – Universidade Federal da Bahia (UFBA)
Salvador – BA – Brasil

{italovalcy, leobino}@ufba.br

Abstract. *The currently deployed Named-Data Networking (NDN) routing protocol, NLSR, is based on link-state algorithms, which require synchronization of the link-state database and knowledge of the entire topology. Such requirements are challenging for fast fault recovery in complex network topologies or intermittent connectivity scenarios. Distance-vector algorithms enable nodes to propagate data reachability information in a distributed and asynchronous manner, providing an enhanced response to topology changes. This paper presents NDN Distance Vector Routing (NDVR) protocol design and evolutions. Our in-depth evaluation demonstrates the benefits of a lightweight distance-vector protocol (NDVR) compared to link-state (NLSR) to propagate reachability information. Based on MiniNDN emulated environment, the experiments consider different topologies, fault models, and traffic models.*

Resumo. *O protocolo de roteamento comumente usado em Redes de Dados Nomeados (NDN), NLSR, é baseado em algoritmo de estado de enlace, que depende da visão completa da topologia e sincronização da tabela de estado dos enlaces entre os nós. Em cenários com topologias complexas, ou com muitas oscilações nos enlaces, o requisito de sincronização torna-se um desafio para rápida recuperação de falhas. Os algoritmos de vetor distância, por outro lado, permitem que os roteadores troquem informações de alcançabilidade e operem de forma assíncrona e distribuída, favorecendo uma rápida resposta a mudanças na topologia. Este artigo descreve o design e evoluções do NDVR (NDN Distance Vector Routing), um protocolo leve para disseminação de informações de alcançabilidade, e apresenta um estudo experimental detalhado que demonstra as vantagens da estratégia vetor distância (NDVR) comparada com estado de enlace (NLSR). A avaliação foi realizada em ambiente emulado e considera diferentes topologias, modelos de falha e modelos de tráfego.*

1. Introdução

A arquitetura de Redes de Dados Nomeados (do inglês, *Named Data Networking* – NDN) é uma proposta *clean-slate* para a Internet do Futuro baseada no paradigma de Redes Centradas na Informação (do inglês, *Information Centric Networking* – ICN) [Zhang et al. 2014]. NDN oferece serviços de comunicação a partir de um modelo totalmente distribuído, dirigido pelo receptor (*receiver-driven*), e adota o encaminhamento

baseado em nomes, com controles de segurança no nível dos dados. Outras características chave da arquitetura são o plano de encaminhamento *stateful* e o suporte a cache oportunístico na rede. Na base dessas funcionalidades está a capacidade da rede em detectar quais dados nomeados (prefixos de nomes) estão alcançáveis, e as direções para alcançá-los, através da função de roteamento [Zhang et al. 2019, Ghasemi et al. 2018].

O roteamento na arquitetura NDN visa disseminar informações de alcançabilidade de prefixos de nomes e disponibilizar direções a serem usadas pela estratégia de encaminhamento [Zhang et al. 2019]. Os pacotes de interesse podem ser roteados de diversas maneiras: inundação, salto-a-salto/salto-multi-salto baseado na visão local da estratégia de encaminhamento, ou ainda, dinamicamente, com auxílio de um protocolo de roteamento. Os pacotes de dados, por outro lado, seguem o caminho reverso do pacote de interesse (i.e., trilha de migalhas de pão). A estratégia de roteamento, portanto, é crucial para um encaminhamento eficiente, disponibiliza alternativas de caminho e auxilia na descoberta de recursos – especialmente com a mobilidade de nós, falhas de enlace e mudanças na topologia.

Diversos protocolos de roteamento foram propostos para NDN [Wang et al. 2012, Wang et al. 2018, Ghasemi et al. 2018], porém, no geral, baseados em algoritmos de estado de enlace, que dependem da sincronização entre os nós e visão completa da topologia. Ao considerar topologias complexas (i.e., mais nós e enlaces, maior conectividade entre os nós, diâmetro) ou dinâmicas (e.g., constantes falhas nos enlaces ou nós, mobilidade de nós), o requisito de sincronização pode impactar no tempo de convergência e recuperação de falhas. Os algoritmos de vetor distância, por outro lado, permitem que os roteadores troquem informações de alcançabilidade de forma assíncrona, distribuída e incremental, favorecendo uma rápida resposta a mudanças na topologia.

Este artigo apresenta um trabalho em andamento de *design*, prototipagem e avaliação do NDVR (*NDN Distance Vector Routing*), que consiste em um protocolo leve para propagação de informações de alcançabilidade baseado no algoritmo vetor distância. O protocolo detecta dinamicamente roteadores vizinhos, troca informações de vetor distância de forma otimizada e calcula caminhos para os produtores de dados de forma distribuída e assíncrona. A principal contribuição é apresentar um estudo experimental detalhado que demonstra as vantagens de um protocolo roteamento para NDN baseado vetor distância (NDVR) comparado com estado de enlace (NLSR).

Este artigo está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados; a Seção 3 descreve o *design* do NDVR, funcionalidades e evoluções; a Seção 4 apresenta uma avaliação de desempenho do NDVR; e a Seção 5 apresenta as conclusões do trabalho.

2. Trabalhos Relacionados

O protocolo de roteamento adotado no Testbed NDN[1] e comumente utilizado em outros cenários NDN é o NLSR (*Named-data Link State Routing*) [Wang et al. 2018]. Baseado em algoritmo de estado de enlace, o NLSR dissemina informações de roteamento e da topologia utilizando protocolos de sincronização de dados, como o ChronoSync[2], e calcula os caminhos para cada prefixo a partir do algoritmo de Dijkstra (no caso de múltiplos

[1] `https://named-data.net/ndn-testbed/`

[2] `https://github.com/named-data/ChronoSync`

caminhos, o algoritmo executado a partir de cada adjacência do roteador). A dependência de sincronização do *banco de dados de estado de enlace* (LSDB) e da topologia entre os nós configura-se um desafio para convergência do protocolo e para rápida recuperação de falhas. Além disso, a sinalização do protocolo pode ocasionar sobrecarga de tráfego na rede, e a estratégia de cálculo de caminhos pode onerar a CPU do roteador.

Algumas abordagens para roteamento em NDN foram propostas ao longo dos anos. Parte delas fortemente baseadas na arquitetura IP e utilizando túneis [Dai et al. 2012] ou protocolos como OSPF [Wang et al. 2012] para propagar informações de alcançabilidade NDN. Outras surgiram como provas de conceito projetadas para funcionar independente da arquitetura que implementa o paradigma ICN, a exemplo do LSCR [Hemmati and Garcia-Luna-Aceves 2015] (baseado em estado de enlace) e do DCR [Garcia-Luna-Aceves 2014] (baseado em vetor distância). Ambos compartilham um problema de *design* relacionado a propagação de prefixos, que pode ocasionar dados inalcançáveis quando um roteador seletivamente propaga um prefixo [Wang et al. 2018].

O MUCA [Ghasemi et al. 2018] tem como foco o estabelecimento de rotas por múltiplos caminhos e ciente da cache na rede, baseado em uma abordagem híbrida, estado de enlace e vetor distância: os nós obtém as informações de topologia e calculam melhores caminhos baseado em uma estratégia de estado de enlace; nós vizinhos trocam informações para computar múltiplos caminhos, usando uma estratégia de vetor distância. Por utilizar estado de enlace como estratégia central, o MUCA compartilha os mesmos desafios de convergência e recuperação de falhas do NLSR.

O roteamento hiperbólico foi investigado por [Lehman et al. 2016], fazendo uso de algoritmos gulosos para computar melhores caminhos com base em coordenadas geográficas. Inundação e auto-aprendizado fazem parte da solução apresentada em [Shi et al. 2017] para roteamento em redes locais e ad-hoc, onde o primeiro pacote de interesse é encaminhado por inundação e o caminho de resposta do pacote de dados cria entradas na FIB para encaminhar novos interesses em *unicast*. Já [Chowdhury et al. 2020], argumenta pelo uso da estratégia de encaminhamento sem protocolo de roteamento no cenário de MANETs, propondo a estratégia de encaminhamento CCLF (*Content Connectivity and Location-Aware Forwarding*) em que os nós decidem o melhor caminho de forma local e independente, utilizando como métricas as medições anteriores de pacotes sobre o mesmo prefixo de nome. A estratégia CCLF não considera casos em que seja necessário coordenação global (e.g., balanceamento de carga, QoS), pois as decisões de encaminhamento são baseadas apenas na visão do nó em questão.

Em trabalho anterior, propusemos o *design* inicial do NDVR [Brito et al. 2020, Brito 2021], desenvolvido no ambiente de simulação ndnSIM[3] e com foco em redes sem fio ad-hoc de alta mobilidade. A seção seguinte apresenta maiores informações sobre o NDVR e evoluções desde o *design* inicial.

3. Protocolo NDVR

O NDVR foi concebido para ser um protocolo leve de troca de informações de alcançabilidade e determinação de caminhos para NDN [Brito 2021]. A característica de leveza está associada a configuração simplificada, independência de estratégias de

[3]https://ndnsim.net/

sincronização adicionais, baixa sobrecarga de mensagens do protocolo e execução distribuída e assíncrona. A base de funcionamento do protocolo é esquema de nomeação hierárquico, que permite compatibilidade com as características da arquitetura NDN, facilita a adoção do modelo de confiança e ajuda no processo de resolução de problemas. O espaço de nomeação do protocolo consiste em três prefixos (detalhados a seguir):

1. `/localhop/ndvr/ehlo/<network>/<router>/<#pfx>/<#ver>/<dgst>`
2. `/localhop/ndvr/dvinfo/<network>/<router>/<#ver>`
3. `/<network>/<router>/KEY/<keyId>`

No *design* do NDVR cada roteador é nomeado de acordo com a rede ou domínio a que pertence, em conjunto com um identificador único do roteador, i.e., `/<network>/<router>`. O componente `<router>` contém duas partes: uma marcação de roteador e o nome do roteador, exemplo: `%C1.Router/router3`[4]. Assim, um roteador da RNP no ponto de presença da Bahia poderia ser nomeado `/rnp/%C1.Router/router-ba`. Dessa forma, se dois roteadores compartilham o mesmo prefixo `<network>`, eles pertencem ao mesmo domínio ou organização, que também é usado para estabelecer a âncora de confiança (conforme Seção 3.4). O componente `/localhop` é um limitador de escopo: mensagens são propagadas apenas na vizinhança direta do nó. As subseções seguintes detalham os esquemas de nomeação.

3.1. Descoberta de vizinhos

A descoberta de vizinhos funciona a partir do envio periódico de interesses, chamados EHLO (*Extended Hello*), para as faces não locais do nó (vizinhança). A nomenclatura de *hello* estendido é adotada pois, além de permitir identificar nós vizinhos, eles apoiam também o processo de disseminação do vetor distância (conforme Seção 3.2). Para que sejam aptos a receber os interesses de EHLO, os nós são pré-configurados com uma entrada na FIB do prefixo `/localhop/ndvr/ehlo`, encaminhando-os para a aplicação NDVR.

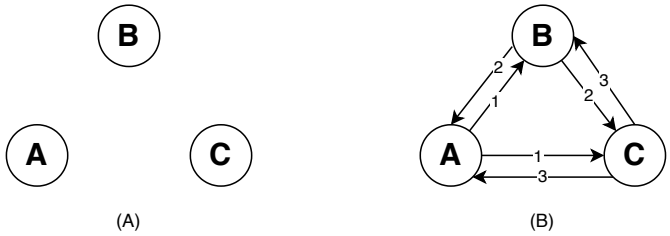


Figura 1. Ilustração do processo de descoberta de vizinhos.

A Figura 1 ilustra o processo de descoberta de vizinhos. No primeiro momento (A), os roteadores desconhecem a existência de quaisquer outros nós. Assim que o NDVR inicia o envio periódico de EHLO, cada um deles detecta seus vizinhos. Por exemplo, o nó A envia a mensagem EHLO 1 para todas as faces não locais. Ao receber essa mensagem (entrada na FIB), B e C aprendem individualmente sobre a vizinhança com A. Similar processo ocorre com os EHLOs enviados por B e C. É importante registrar que os números das mensagens são meramente ilustrativos, porém o protocolo não requer qualquer ordenação das mensagens para correto funcionamento. Registra-se ainda que

[4] *%C1.Router* é chamado de marcador de comando, uma espécie de palavra reservada que identifica um tipo específico de componente de nome

a restrição de escopo `/localhop` evita que um nó encaminhe EHLO em nome de outro, o que causaria estabelecimento de adjacências incorretas (e.g., uma topologia linear incorretamente descoberta como malha).

A periodicidade dos EHLOs pode ser configurada com o parâmetro *ehloInterval*, cujo valor padrão é um segundo. Ao detectar um vizinho, as seguintes informações são registradas: instante de tempo em que o vizinho foi inicialmente visto e visto pela última vez, versão da tabela vetor distância do vizinho e identificador da face em que o vizinho é alcançável. A remoção de vizinhança ocorre quando um nó deixa de receber novos EHLOs de um vizinho previamente descoberto em um intervalo de tempo (*ehloTimeout*) ou por uma quantidade de períodos (*ehloMultiplier*), o menor entre eles. Para baixos intervalos de EHLO, a remoção de vizinhos baseada no *ehloMultiplier* é mais efetiva, ao passo que intervalos maiores se beneficiam melhor do *ehloTimeout*.

3.2. Disseminação do vetor distância

A disseminação do vetor distância é o processo pelo qual nós NDVR propagam informações de alcançabilidade entre os vizinhos para alimentar o cálculo de caminhos. A disseminação do vetor distância ocorre sob-demanda, através de mensagens de DVINFO (*Distance Vector Information*), quando um novo prefixo é anunciado ou quando ocorre uma mudança de adjacência que impacta prefixos previamente anunciados. Em razão do modelo de comunicação *receiver-driven* da NDN [Zhu and Afanasyev 2013], o envio do DVINFO sob-demanda requer primeiro uma notificação de novo DVINFO, seguida pelos interesses enviados pelos vizinhos e então o nó envia o pacote de dados com as informações do vetor distância em resposta. A notificação de novo DVINFO ocorre através das mensagens de EHLO, com o número de prefixos alcançáveis, versão do DVINFO e um resumo digital das informações de DVINFO (*digest*).

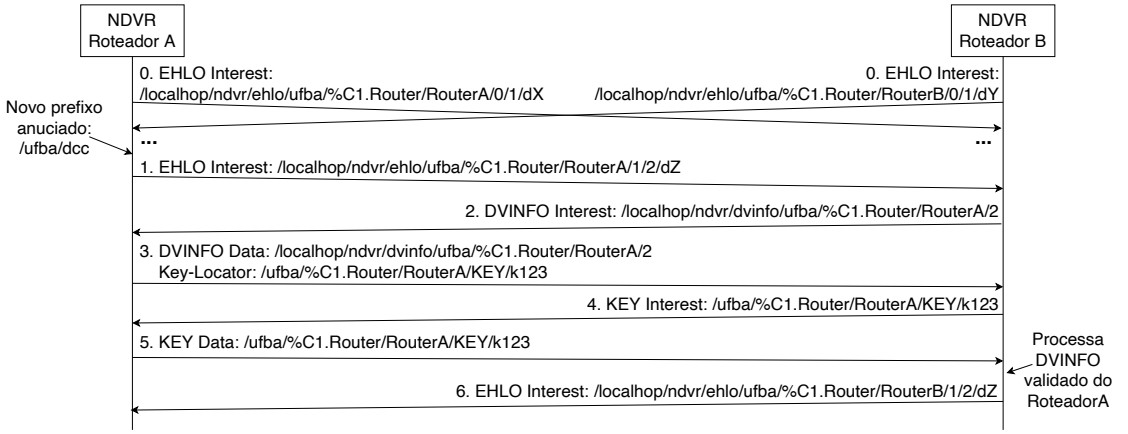


Figura 2. Diagrama de fluxo da disseminação do vetor distância (DVINFO).

A Figura 2 ilustra o processo de disseminação de vetor distância entre dois roteadores NDVR. Inicialmente, no instante 0, os roteadores trocam mensagens de EHLO e fazem a descoberta de vizinhança. Nesse exemplo, ambos iniciam sem anunciar informações de alcançabilidade (`#pfx=0`), porém processo similar seria aplicado caso já iniciassem propagando prefixos de nomes. No instante 1, o roteador A aprende um novo prefixo `/ufba/dcc` (e.g., via linha de comando ou API) e envia um EHLO para anunciar esse novo prefixo. É importante notar a quantidade de prefixos (`#pfx=1`), incremento da

versão (`#ver=2`) e novo *digest* (`dgst=dZ`). Ao receber o EHLO, o roteador B identifica que o roteador A tem uma nova versão de DVINFO e o *digest* é diferente da sua própria tabela de vetor distância, portanto o roteador B envia um interesse de DVINFO para o roteador A (instante 2). O roteador A responde ao interesse de DVINFO com as suas informações de vetor distancia (instante 3). O roteador B identifica a chave usada para assinar o pacote de dados do DVINFO e, caso não tenha a chave obtida previamente, envia um interesse de chave (instante 4). No instante 5, o roteador A responde com sua chave e o roteador B faz a validação da assinatura, de acordo com o modelo de confiança e regras de validação (conforme Seção 3.4). O roteador B então faz o processamento do DVINFO do roteador A, atualiza suas informações de vetor distância, atualiza a FIB e envia um EHLO com uma nova quantidade de prefixos (`#pfx=1`), incremento da versão (`#ver=2`) e novo *digest* (`dgst=dZ`). Ao receber o EHLO do roteador B, o roteador A identifica que trata-se de uma nova versão de DVINFO, no entanto o *digest* é o mesmo da sua tabela local e, portanto, não há necessidade de nova requisição.

As informações contidas no pacote de dados do DVINFO são: prefixos de nome alcançáveis, custo para cada prefixo e número de sequência associado a cada prefixo. Cada um desses atributos será usado para alimentar o algoritmo de cálculo de caminhos, conforme discutido na Seção 3.3. Central para o processo de disseminação de vetor distância, a lista de prefixos de nome alcançáveis se refere ao conjunto de prefixos NDN que o roteador pode alcançar, localmente ou através de vizinhos diretamente conectados.

3.3. Cálculo de caminhos

Ao receber um DVINFO e validar o conteúdo, o roteador processa a lista de prefixos alcançáveis e faz o cálculo de caminhos. Para o cálculo dos caminhos, três informações, além do prefixo em si, são importantes: o custo para alcançar o prefixo, o número de sequência e a face NDN na qual o vizinho NDVR é alcançável. A face é obtida através de *tags* inseridas pelo *pipeline* NDN. O custo é uma métrica provida pelo NDVR, que qualifica caminhos para alcançar o prefixo, permitindo compará-los e decidir o melhor caminho. Por padrão, o NDVR considera contador de saltos como métrica de custo, porém outras estratégias podem ser adotadas (e.g., atraso acumulado, banda residual, etc.).

O número de sequência é usado para diferenciar anúncios mais atuais de outros retransmitidos, evitando assim o problema de contagem ao infinito [Mohapatra and Krishnamurthy 2004]. A utilização de números de sequência em algoritmos de vetor distância é recorrente, e um dos principais exemplos é o protocolo de roteamento DSDV [Perkins and Bhagwat 1994]. A ideia básica é marcar os prefixos com um número monotonicamente incremental no roteador de origem do anúncio e, durante o processamento do DVINFO, descartar rotas com número de sequência inferior. No NDVR, o roteador de origem incrementa o número de sequência em duas unidades, ao passo que roteadores intermediários, com mudanças nas adjacências impactando um prefixo, incrementam o número de sequência por uma unidade. Quando um roteador originador recebe o seu próprio prefixo com número ímpar, um novo incremento par é feito e novos anúncios são gerados. Assim é fácil identificar quando uma atualização foi gerada em decorrência de mudanças na topologia (e.g., falhas nas adjacências), o que permite que o DVINFO não precise enviar atualizações periódicas.

O Algoritmo 1 ilustra de forma geral o processamento do DVINFO. A linha 3 é responsável pelo cálculo do custo, que por padrão apenas incrementa o custo recebido

do vizinho. Em seguida, caso trate-se de uma nova rota, o prefixo é inserido na tabela de roteamento. Caso contrário, nas linhas 9 à 11, o prefixo é atualizado apenas caso o número de sequência seja maior, ou igual porém com melhor custo. A chamada de $AtualizaPrefixo()$ abstrai detalhes como checagem do custo infinito para remoção do prefixo, checagem do número de sequência ímpar para rotas locais, etc. Por fim, caso tenha havido mudança (linhas 13 - 15), a tabela DVINFO tem sua versão incrementada, um novo *digest* é calculado e um EHLO é enviado imediatamente.

Algoritmo 1: Roteador **i** processa DVINFO recebido do roteador **j**

```
1  houveMudanca = Falso;
2  foreach prefixo de nome p ∈ DvInfo_j do
3      custo_i ← CalculaCusto(j, p.cost);
4      prev_p ← ConsultaPrefixoExistente(p);
5      if prev_p = ∅ then
6          InserePrefixo(p, nextHop = j, cost = custo_i, seq = p.seq);
7          houveMudanca = True;
8      else
9          if p.seq > prev_p.seq OU (p.seq = prev_p.seq E custo_i < prev_p.cost)
             then
10             AtualizaPrefixo(p, nextHop = j, cost = custo_i, seq = p.seq);
11             houveMudanca = True;
12 end
13 if houveMudanca then
14     AtualizaVersaoDigest();
15     EnviaEHLO();
```

3.4. Modelo de confiança

Em harmonia com a arquitetura NDN, todo pacote de dados do NDVR é assinado digitalmente e a chave utilizada é indicada no campo KeyLocator do pacote, permitindo que o consumidor valide a segurança do DVINFO antes de processá-lo. No processo de validação, o NDVR (i) obtém a chave do vizinho de acordo com o KeyLocator, (ii) verifica as propriedades da chave (expiração, revogação, algoritmos criptográficos) e autentica a chave no modelo de confiança, e (iii) verifica a assinatura digital no pacote de dados.

Figura 3. Ilustração da relação entre as chaves no modelo de confiança do NDVR.

Todas as validações são feitas com base em um conjunto de políticas de segurança definidas no arquivo de regras de validação [Brito 2021]. As regras de validação são escritas em uma linguagem de domínio específico, bastante flexível, e permitem aplicar o esquema de nomeação hierárquica do NDVR, garantindo: que as mensagens de DVINFO são assinadas pela chave do roteador; a chave do roteador é assinada pela chave da rede; a chave da rede pertence à âncora de segurança. A âncora de segurança pressupõe que a chave da rede seja pré-instalada em todos os nós NDVR. A Figura 3 ilustra a relação entre as chaves e o modelo de confiança adotado no NDVR.

3.5. Evoluções no NDVR

Muitos recursos e correções foram incorporadas desde o *design* original do NDVR [Brito et al. 2020]. Uma das principais mudanças foi a refatoração do código para execução em ambiente real e emulado. Originalmente desenvolvido para o ambiente de simulação ndnSIM, o NDVR foi modificado para remover APIs específicas do simulador e fazer uso do próprio pipeline NDN e das APIs da biblioteca ndn-cxx em operações essenciais como gerenciamento da FIB. Por exemplo, funções de inserção, atualização e remoção de rotas passaram a usar pacotes de interesse e dados para se comunicar com o NFD no escopo `/localhost` e alterar a FIB. Além disso, destaca-se as correções implantadas nas funções de remoção de adjacências, serialização de mensagens, diferenciação no incremento no número de sequência entre o roteador originador e o roteador que detecta mudanças nas adjacências, e correções no algoritmo de processamento do DVINFO. O código fonte do NDVR está disponível no repositório do projeto[5], bem como os cenários de avaliação apresentados a seguir.

4. Avaliação Experimental

A avaliação da proposta foi realizada através de emulação, considerando diferentes topologias, modelos de tráfego e falhas, e comparando com o protocolo de roteamento NLSR. O NLSR foi configurado de forma especial, para fornecer rápida convergência e recuperação de falhas, conforme detalhado a seguir. Ao final desta seção, serão apresentados experimentos considerando a configuração padrão do NLSR.

4.1. Planejamento de Experimentos

Os experimentos foram conduzidos utilizando o modelo de planejamento de experimentos fatorial completo [Jain 1990], considerando os seguintes fatores: [Fator **A** - Protocolo de Roteamento]= {NDVR, NLSR}; [Fator **B** - Modelo de Tráfego]={Taxa de Bits Constante (CBR), Taxa de Bits Variável (VBR)}; [Fator **C** - Topologia]= {Testbed NDN, Backbone RNP}; [Fator **D** - Tipo de Falha]= {Simples, Periódica}. Os fatores e níveis do sistema foram definidos com base nas características que diferenciam cenários de aplicação dos algoritmos de estado de enlace e vetor distância e valores mais prováveis utilizados na literatura, permitindo avaliar a completude do efeito que essas variáveis provocam ao ambiente. A Tabela 1 apresenta mais detalhes dos fatores e níveis. Todos os experimentos foram analisados a partir do intervalo de confiança[6], da média e do desvio padrão aferidos. Esses parâmetros são utilizados como base para o cálculo da soma dos quadrados, resultando na influência de cada fator nas variáveis de resposta.

As variáveis de resposta utilizadas na avaliação foram: latência da rede (medida a partir do RTT - tempo de ida e volta), coeficiente de variação da latência, utilização de CPU, sobrecarga do protocolo (número de pacotes de interesse e de dados, razão de sobrecarga) e percentual de perda de pacotes.

O ambiente de experimentação possui as seguintes características: (i) servidor Intel Xeon E5-2643 3.40GHz, 32GB de RAM; (ii) Linux 4.15.0-112-x86_64; (iii) Mini-NDN 0.5.0; (iv) NLSR 0.6.0. Durante os testes foi identificado e corrigido um defeito de *software* no NLSR, os resultados apresentados consideram a versão corrigida. Para gerar

[5] `https://github.com/italovalcy/ndvr/tree/ndvr-emu`

[6] Utilização da distribuição *t-student* com replicações de 10 execuções por experimento e $\alpha = 0,05\%$.

Tabela 1. Fatores e níveis do planejamento de experimentos

Protocolo de roteamento	NDVR	Configuração padrão
	NLSR	enable security = yes max-faces-per-prefix = 1 (no multipath) hello-interval = 1s adj-lsa-build-interval = 1s routing-calc-interval = 1s neighbor link-cost = 1 (hop count)
Modelo de tráfego	CBR	IDT = 1000ms (*Inter Departure Time*) PS = 800bytes (*Packet Size*)
	VBR	IDT = Distribuição de Poisson, média 1000 PS = Dist. Uniforme min=400 max=1200
Tipo de falha	Simples	Falha do nó mais conectado (MCN, *most connected node*) aos 60s e retorno aos 120s
	Periodica	Falha do MCN a cada 30s (on/off)
Topologia	Testbed NDN	38 nodes / 134 links / grau nodal médio 3.52
	Backbone RNP	29 nodes / 74 links / grau nodal médio 2.55

o tráfego foi utilizado a ferramenta *ndnping*[7], estendida para suportar modelo de tráfego VBR com base em distribuições de probabilidade conhecidas. Cada execução consiste em 360 segundos de duração, dos quais os primeiros 180 segundos são descartados da análise e nos 180 segundos seguintes o gerador de tráfego é executado.

A Figura 4 apresenta a observação da normalidade na execução dos experimentos. O esperado é que os pontos do gráfico, relacionados aos experimentos, residam sobre ou próximos à linha normal, como é observado.

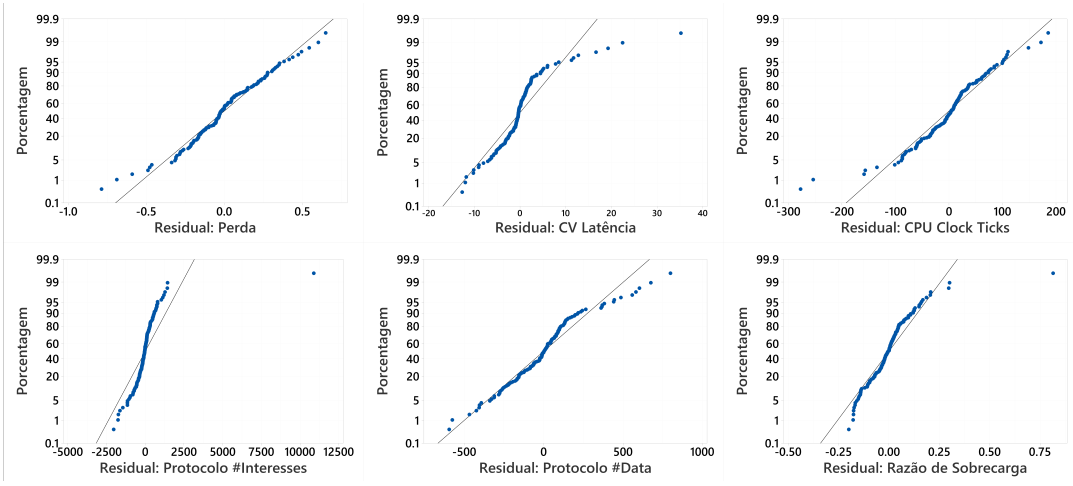


Figura 4. Observação da Normalidade na Execução dos Experimentos.

A Figura 5 mostra o gráfico pareto para o projeto fatorial 2^k, revelando o grau de influência que os fatores exercem sobre as variáveis de resposta. Para a perda de pacotes, o fator com maior influência nos resultados foi o tipo de falha. Considerando a quantidade de falhas em cada nível, i.e., simples com uma falha e periódica com três, é fácil

[7]https://github.com/named-data/ndn-tools/tree/master/tools/ping

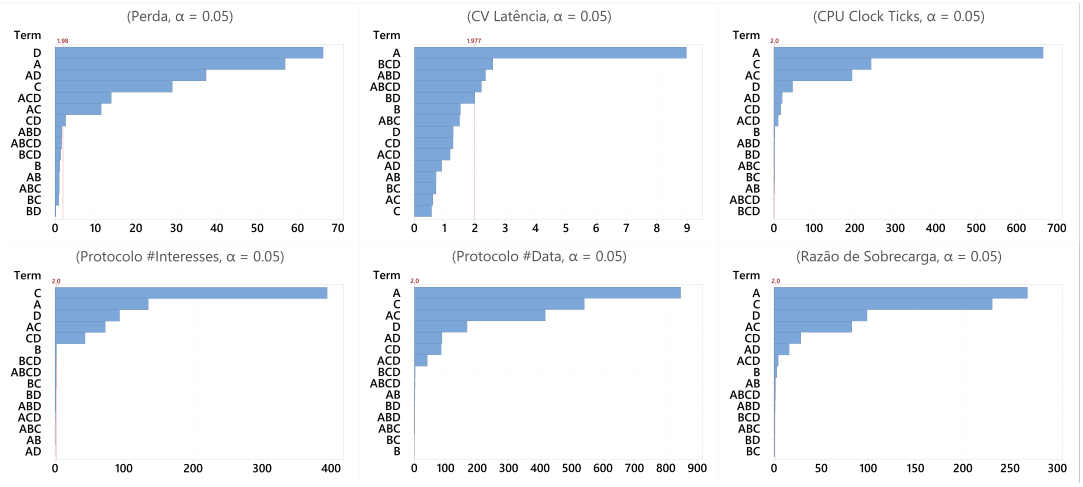


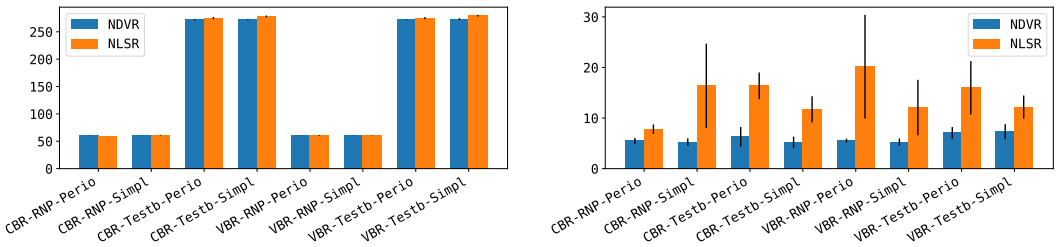
Figura 5. Influência dos Fatores: A - Protocolo de Roteamento, B - Modelo de Tráfego, C - Topologia, D - Modelo de Falhas.

entender sua influência na perda. Ainda assim, o protocolo de roteamento aparece muito próximo como segundo fator de maior influência, seguido pela combinação de ambos. O fator topologia influenciou de maneira mais significativa a variável de resposta sobrecarga em termos quantidade de pacotes de interesse. Tal influência se dá principalmente pois ambos os protocolos NDVR e NLSR utilizam pacotes de interesse para manutenção de vizinhanças e detecção de falhas dos nós. Assim, quanto maior e mais complexa a topologia (e.g., maior grau nodal), maior a quantidade de pacotes de interesse trocados. Já a medição da sobrecarga em termos de pacotes de dados e da razão de sobrecarga (que considera a razão da quantidade de pacotes do protocolo sobre o total) mostram que a maior influência é da estratégia de roteamento. Nas demais variáveis de resposta, o protocolo de roteamento é o fator de maior influência. Essas informações confirmam a hipótese de que a escolha da estratégia de roteamento produz alteração significativa nas variáveis de resposta, conforme será analisado posteriormente.

4.2. Análise dos Resultados

A Figura 6a apresenta a análise da variável de resposta latência, cuja influência do protocolo de roteamento é mínima. A grande diferença ocorre em relação ao fator topologia. De fato, as diferenças de latência individuais em cada enlace na topologia da RNP e na topologia do testbed NDN, atrelados com características da topologia como grau nodal e diâmetro, são as maiores influências no cálculo da média da latência. A semelhança nas medições, inclusive, aponta similaridade na escolha de caminhos dos protocolos. Se considerarmos, entretanto, medidas de dispersão na distribuição dos dados, como desvio padrão ou coeficiente de variação (relação do desvio padrão pela média), é possível identificar a influência do protocolo de roteamento. Em particular, a Figura 6b mostra como o protocolo NDVR mantém uma menor dispersão dos valores em todos as configurações, algumas vezes menor que a metade do valor percentual do NLSR. Por exemplo, no cenário com tráfego CBR, topologia do Testbed NDN e tipo de falha periódica, o NDVR apresenta CV Latência de $6,26 \pm 1,95\%$, ao passo que o NLSR $16,37 \pm 2,64\%$.

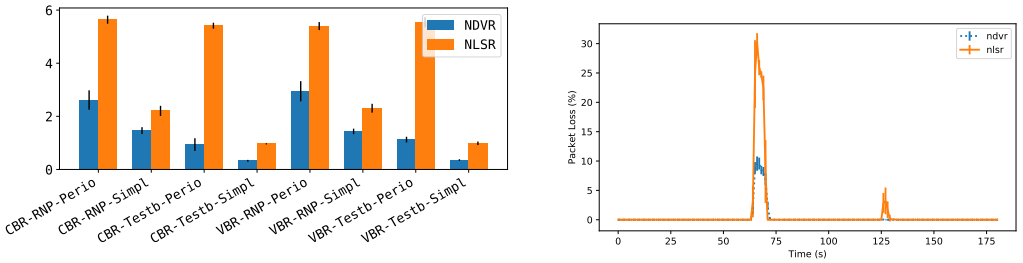
O comportamento dos protocolos em relação a perda de pacotes pode ser analisado na Figura 7a. Em todas as configurações, o NDVR teve um desempenho superior ao NLSR, confirmando a hipótese do protocolo de vetor distância reagir mais rápido a falhas.



(a) Latência (RTT) (b) Coeficiente de Variação da Latência

Figura 6. Análise dos resultados: Latência e CV Latência.

Em alguns cenários, inclusive, o intervalo de confiança do NLSR com falha simples se sobrepõe ao do NDVR com falhas periódicas, como é o caso da topologia do Testbed NDN, tanto para tráfego CBR quanto VBR. A Figura 7b detalha a perda de pacotes ao longo do tempo de execução para o cenário CBR-Testbed-Simples. É possível observar que quando o nó mais conectado falha no segundo 60, o NLSR apresenta uma perda de quase 30%, ao passo que o NDVR se matém com uma média de 10% de perda. Quando o nó mais conectado se recupera no segundo 120, novas perdas são observadas no NLSR por ocasião do processo de convergência, enquanto que o NDVR converge sem gerar perdas. Ambos os protocolos foram configurados com apenas um caminho por prefixo de nome, uma vez que o NDVR atualmente não suporta *multipath*. Espera-se valores de perda menores na existência de múltiplos caminhos.

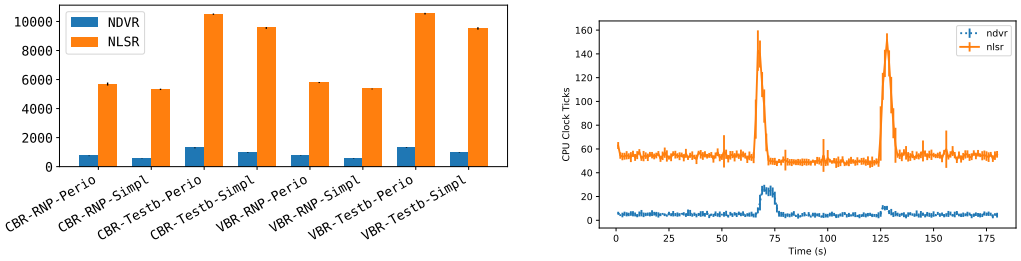


(a) Por níveis dos fatores (b) Cenário CBR-Testbed-Simples

Figura 7. Análise dos resultados: Perda de pacotes.

Em termos de utilização de CPU, o protocolo NDVR possui valores inferiores ao NLSR, como pode ser visto na Figura 8a. Esse gráfico apresenta a quantidade de ciclos de CPU (*clock ticks*) consumidos por cada protocolo em todos os nós durante o experimento. A Figura 8b apresenta um detalhamento do uso de CPU ao longo do experimento no cenário CBR-Testbed-Simples. Os protocolos baseados em estado de enlace tipicamente possuem um consumo de CPU maior que o vetor distância, devido ao processo de sincronização do LSDB e cálculo de melhor caminho para cada prefixo. Durante o período de falha e convergência, por exemplo, o uso de CPU do NDVR se mantém em média cerca de três vezes menor que o NLSR, o que pode impactar outras funções do pipeline NDN no roteador, como a busca na cache, onerando o desempenho das aplicações.

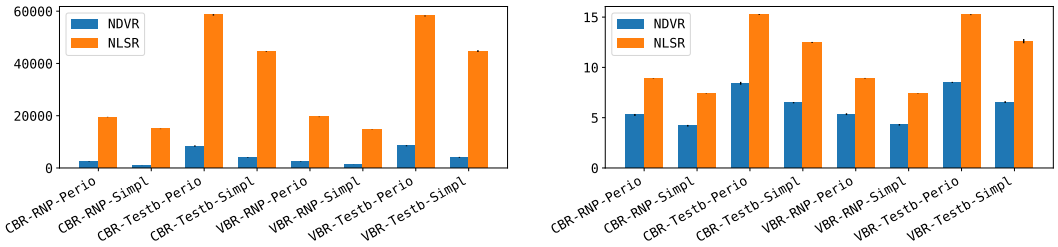
Outra métrica analisada foi a sobrecarga do protocolo (*overhead*). Devido à baixa influência do protocolo na quantidade de pacotes de interesse (Figura 5), os gráficos apresentam apenas a análise da quantidade de pacotes de dados e da razão de sobrecarga. A Figura 9a apresenta a média do total de pacotes de dados trocados entre todos os nós



(a) Por níveis dos fatores (b) Cenário CBR-Testbed-Simples

Figura 8. Análise dos resultados: Utilização de CPU.

conforme níveis dos fatores. A quantidade de pacotes de dados do NDVR (usado para disseminação do vetor distância entre os vizinhos) é significativamente menor que do NLSR (usado para sincronização do LSDB, envio de LSAs e manutenção de adjacência), causando menor sobrecarga na rede. A Figura 9b apresenta a razão de sobrecarga, demonstrando o NDVR com porcentagem menor que o NLSR em todos os cenários.

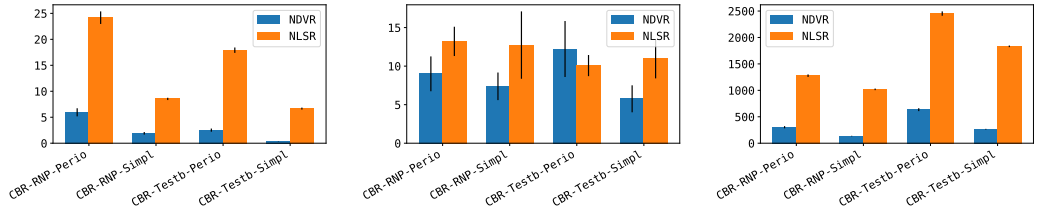


(a) Qtd. de Pacotes de Dados (b) Razão de Sobrecarga

Figura 9. Análise dos resultados: Sobrecarga do Protocolo

Por fim, foi executada uma nova rodada de experimentos considerando valores recomendados como padrões na configuração dos intervalos do NLSR (intervalo de hello, troca de LSA, etc) e variando apenas os fatores topologia e modelo de falha, cujas influências foram mais significativas. Até aqui as configurações de NLSR e NDVR utilizavam intervalos de 1 segundo para otimizar a convergência e reduzir perdas. Nos próximos resultados o intervalo foi configurado como 30s. Os novos experimentos ajudam a entender o comportamento dos protocolos em relação ao intervalo de sinalização.

A Figura 10 apresenta os resultados com intervalo de sinalização de 30 segundos. Como esperado, a perda de pacotes aumentou de forma significativa em ambos os protocolos, porém o NDVR mantém um percentual de perda 5 vezes menor que o NLSR (e.g., no pior cenário CBR-RNP-Periódica o NDVR apresenta perda de $5,95 \pm 0,78\%$ ao passo que o NLSR $24,17 \pm 1,21\%$ e no melhor cenário CBR-Testbed-Simples o NDVR possui perda de $0,33 \pm 0,03\%$ e NLSR $6,67 \pm 0,19\%$). A utilização de CPU também apresenta significativa diferença entre os protocolos, com NDVR consumindo muito menos recursos proporcionalmente ao NLSR. O coeficiente de variação da latência mostra um empate técnico considerando os intervalos de confiança, com uma média do NDVR maior no cenário CBR-Testbed-Periodica, ocasionado pelo intervalo maior de sinalização que atrasa a convergência do protocolo - especialmente na topologia do NDN Testbed.



(a) Perda (b) CV Latência (c) CPU

Figura 10. Análise dos resultados com intervalos de 30 segundos

5. Conclusões e Trabalhos Futuros

Este artigo apresentou o *design* e evoluções do NDVR, e um estudo experimental detalhado que demonstra as vantagens de um protocolo leve para disseminação de informações de alcançabilidade através de vetor distância (NDVR) comparado com estado de enlace (NLSR). O *design* de um novo protocolo de roteamento, seguindo os padrões de projeto, APIs e modelo de comunicação baseado em pacotes de interesse/dados da NDN, constitui-se uma oportunidade de aprendizagem profunda da arquitetura. O NDVR é um trabalho em andamento, que tem evoluído significativamente a partir de um processo cíclico de investigação, *design*, prototipagem e experimentação.

A avaliação experimental demonstrou vantagens do NDVR em relação ao consumo de recursos do roteador e da rede quando comparado com o NLSR. O NDVR obteve um consumo de CPU nos roteadores até oito vezes menor que o NLSR (cenário CBR-Testbed-Períodica) e razão de sobrecarga de mensagens na rede 42% menor que o NLSR (cenário VBR-RNP-Simples). Enquanto isso, métricas de qualidade da rede como perda de pacotes e coeficiente de variação da latência sustentam a hipótese de que protocolos baseados em vetor distância reagem mais rapidamente a falhas ou mudanças na topologia: NDVR apresentou menor taxa de perda de pacotes que o NLSR em todos os cenários, especialmente naqueles com modelo de falhas periódicas, como é o caso do cenário CBR-Testbed-Periódica cujas medições foram $0.94 \pm 0.24\%$ e $5.41 \pm 0.11\%$, respectivamente. Os resultados acima foram obtidos com uma configuração do NLSR para rápida convergência. Foram realizados experimentos também considerando os valores padrões do NLSR e aplicando-os ao NDVR, obtendo resultados compatíveis com os anteriores e descartando possibilidade de relação com a configuração adotada no NLSR.

Em trabalhos futuros espera-se investigar o suporte a múltiplos caminhos no NDVR, que vai acelerar a recuperação de falhas e permitirá aplicações de balanceamento de carga ou agregação na transferência de dados. Outro aspecto a ser explorado é a métrica de escolha de caminhos, por exemplo para oferecer caminhos com garantias de tráfego.

Referências

Brito, I. V. S. (2021). NDVR: NDN Distance Vector Routing. Technical report, Federal University of Bahia.

Brito, I. V. S., Sampaio, L., and Zhang, L. (2020). (Poster) Towards a distance vector routing protocol for named data networking. In *NDN Community Meeting 2020*.

Chowdhury, M., Khan, J. A., and Wang, L. (2020). Leveraging content connectivity and location awareness for adaptive forwarding in ndn-based mobile ad hoc networks. In *7th ACM Conference on Information-Centric Networking*, page 59–69. ACM.

Dai, H., Lu, J., Wang, Y., and Liu, B. (2012). A two-layer intra-domain routing scheme for named data networking. In *2012 IEEE Global Communications Conference (GLOBECOM)*, pages 2815–2820. IEEE.

Garcia-Luna-Aceves, J. (2014). Routing to multi-instantiated destinations: Principles and applications. In *2014 IEEE 22nd International Conference on Network Protocols*, pages 155–166. IEEE.

Ghasemi, C., Yousefi, H., Shin, K. G., and Zhang, B. (2018). Muca: New routing for named data networking. In *2018 IFIP Networking Conference (IFIP Networking) and Workshops*, pages 289–297. IEEE.

Hemmati, E. and Garcia-Luna-Aceves, J. (2015). A new approach to name-based link-state routing for information-centric networks. In *Proceedings of the 2nd ACM Conference on Information-Centric Networking*, pages 29–38.

Jain, R. (1990). *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. John Wiley & Sons.

Lehman, V., Gawande, A., Zhang, B., Zhang, L., Aldecoa, R., Krioukov, D., and Wang, L. (2016). An experimental investigation of hyperbolic routing with a smart forwarding plane in NDN. In *24th IEEE/ACM IWQoS*, pages 1–10. IEEE.

Mohapatra, P. and Krishnamurthy, S. (2004). *AD HOC NETWORKS: technologies and protocols*. Springer Science & Business Media.

Perkins, C. E. and Bhagwat, P. (1994). Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *SIGCOMM Comput. Commun. Rev.*, 24(4):234–244.

Shi, J., Newberry, E., and Zhang, B. (2017). On broadcast-based self-learning in named data networking. In *2017 IFIP Networking Conference (IFIP Networking) and Workshops*, pages 1–9. IEEE.

Wang, L., Hoque, A., Yi, C., Alyyan, A., and Zhang, B. (2012). OSPFN: An OSPF based routing protocol for named data networking. Technical report, University of Memphis and University of Arizona.

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.

Zhang, L., Afanasyev, A., Burke, J., Jacobson, V., Claffy, K., Crowley, P., Papadopoulos, C., Wang, L., and Zhang, B. (2014). Named Data Networking. *SIGCOMM Comput. Commun. Rev.*, 44(3):66–73.

Zhang, Y., Xia, Z., Afanasyev, A., and Zhang, L. (2019). A note on routing scalability in named data networking. In *2019 IEEE ICC Workshops*, pages 1–6. IEEE.

Zhu, Z. and Afanasyev, A. (2013). Let's chronosync: Decentralized dataset state synchronization in named data networking. In *2013 21st IEEE International Conference on Network Protocols (ICNP)*, pages 1–10. IEEE.