



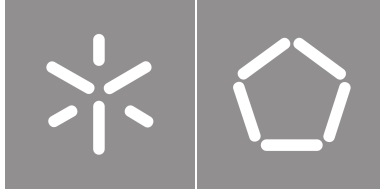
Rui Pedro Carvalho Dias

**Estratégias de sincronização de dados em  
redes veiculares NDN.**

**Universidade do Minho**  
Escola de Engenharia







**Universidade do Minho**

Escola de Engenharia

Rui Pedro Carvalho Dias

**Estratégias de sincronização de dados  
em redes veiculares NDN.**

Dissertação de Mestrado

Trabalho efetuado sob a orientação de:

**Maria João Mesquita Rodrigues Cunha Nicolau  
Pinto**

**António Luís Duarte Costa**

## **DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS**

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositoriUM da Universidade do Minho.

### ***Licença concedida aos utilizadores deste trabalho***



**Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International**  
**CC BY-NC-SA 4.0**

<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.en>

# **Agradecimentos**

### **DECLARAÇÃO DE INTEGRIDADE**

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

\_\_\_\_\_, \_\_\_\_\_  
(Local) (Data)

\_\_\_\_\_  
(Rui Pedro Carvalho Dias)

# Resumo

## **Estratégias de sincronização de dados em redes veiculares NDN.**

Relativamente ao seu conteúdo, os resumos não devem ultrapassar uma página e frequentemente tentam responder às seguintes questões (é imprescindível a adaptação às práticas habituais da sua área científica):

1. Qual é o problema?
2. Porque é que é um problema interessante/desafiante?
3. Qual é a proposta de abordagem/solução?
4. Quais são as consequências/resultados da solução proposta?

**Palavras-chave:** Primeira palavra-chave, Outra palavra-chave, Mais uma palavra-chave, A última palavra-chave

# Índice

<b>Índice de Figuras</b>	<b>vii</b>
<b>Índice de Tabelas</b>	<b>viii</b>
<b>Siglas</b>	<b>x</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Enquadramento e Motivação . . . . .	1
1.2 Objetivos . . . . .	1
<b>2 Estado da Arte</b>	<b>2</b>
2.1 Redes NDN . . . . .	2
2.1.1 <i>Funcionamento do Named Data Network (NDN)</i> . . . . .	3
2.2 Sincronização no NDN . . . . .	4
2.2.1 <i>State Vector Sync State Vector Sync (SVSync)</i> . . . . .	5
<b>Bibliografia</b>	<b>8</b>



# Índice de Figuras

1	Pacotes das redes NDN [4]	2
2	Processo de encaminhamento num nó NDN [4]	3
3	Exemplo de Vetor de Estado [1]	4
4	Exemplo de Vetor de Estado [3]	6
5	Fluxograma do funcionamento do SVSync [3]	7

# **Índice de Tabelas**

# **Índice de Listagens**

# Siglas

<b>CS</b>	Content Store (p. <a href="#">3</a> )
<b>FIB</b>	Forwarding Information Base (p. <a href="#">3</a> )
<b>NDN</b>	Named Data Network (pp. <a href="#">vi</a> , <a href="#">vii</a> , <a href="#">1–5</a> , <a href="#">7</a> )
<b>PIT</b>	Pending Interest Table (pp. <a href="#">3</a> , <a href="#">4</a> )
<b>SVSync</b>	State Vector Sync (pp. <a href="#">vi</a> , <a href="#">vii</a> , <a href="#">1</a> , <a href="#">5–7</a> )
<b>VANETs</b>	Redes vehiculares ad hoc (p. <a href="#">1</a> )

# Capítulo 1

## Introdução

### 1.1 Enquadramento e Motivação

O progresso das tecnologias de comunicações e das redes veiculares ad hoc (VANETs) têm produzido avanços no desenvolvimento de sistemas de transporte inteligentes. Neste contexto a troca de dados de forma eficiente e fiável entre veículos representa um grande desafio. O paradigma Named Data Network (NDN) aparece como uma alternativa às arquiteturas centradas em IP, pois privilegia uma comunicação com base em conteúdos característica de ambientes muito dinâmicos como os das Redes veiculares ad hoc (VANETs). Uma parte essencial do NDN é a camada de sincronização, esta assegura a consistência e atualização dos dados, mas esses mecanismos de sincronização já existentes necessitam de ser adaptados ao cenário das redes veiculares, principalmente no que toca à latência, escalabilidade e à gestão de restrições geográficas e temporais. Esta dissertação tem como objetivo desenvolver uma camada de sincronização robusta, especificamente desenhada para o NDN aplicado às VANETs, contribuindo para a fiabilidade e o melhor desempenho das comunicações veiculares em contextos de sistemas de transporte.

### 1.2 Objetivos

Os objetivos desta dissertação passam por:

- Revisitar mecanismos já existentes de sincronização no NDN, neste caso o State Vector Sync SVSync e identificar algumas limitações nas redes veiculares.
- Propor uma solução para a sincronização adaptada às VANETs, incluindo as restrições geográficas e temporais
- Desenvolver algoritmos para a sincronização de dados nas redes veiculares.
- Implementar a solução proposta
- Avaliar a eficácia da solução proposta.

# Capítulo 2

## Estado da Arte

### 2.1 Redes NDN

No início, quando as ideias centrais que fundamentaram a Internet começaram a ser desenvolvidas, a solução de comunicação oferecida pelo TCP/IP fosse única e revolucionária, o problema que ela resolveu era essencialmente o de conversas ponto a ponto entre duas entidades. Apesar de o TCP/IP ter sido concebido para conversos entre pontos finais de comunicação, acabou por superar todas as expectativas sendo também amplamente utilizado na distribuição de conteúdo, levando assim a Internet a passar a funcionar como uma rede de distribuição. Sendo assim tornou-se muito difícil resolver problemas pois as redes de distribuição são mais complexas que as redes de comunicação. A natureza do IP está implementada nos seus datagramas, identificando apenas os pontos iniciais e finais de comunicação. O NDN propõe generalizar a arquitetura da Internet, removendo essa limitação pois os nomes de um datagrama NDN passam a ser estruturados hierarquicamente, sendo que podem ser utilizados para identificar um pedaço de dados dentro de uma conversa ou um pedaço de conteúdo. Esta mudança, teoricamente simples, permite que se resolva uma alta gama de problemas de distribuição. A comunicação NDN é impulsionada pelos recetores, existindo apenas 2 pacotes: os pacotes de interesse e os pacotes de dados (Figura 1). O consumidor coloca o nome da parte dos dados desejada no pacote de interesse e

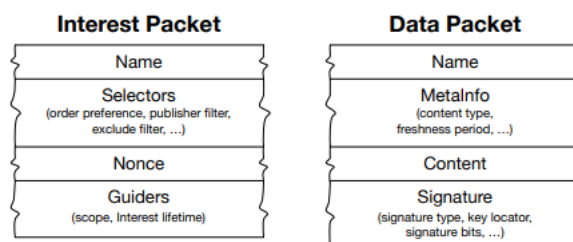


Figura 1: Pacotes das redes NDN [4]

envia-o para a rede, os routers usam o o nome para encaminhar o interesse em direção aos produtores dos dados. Quando o interesse chega ao nó que possui os dados solicitados, esse nó retornará um

pacote de dados que contém o nome e o conteúdo. Este pacote segue on caminho inverso ao caminho percorrido pelo pacote de interesse.

### 2.1.1 Funcionamento do NDN

Para realizar as funções de encaminhamento de interesses e dados, os routers NDN têm três estruturas: a Pending Interest Table (PIT), a Forwarding Information Base (FIB) e a Content Store (CS). A PIT é uma tabela que armazena todos os interesses que ainda não foram satisfeitos. Cada entrada registra o nome dos dados, juntamente com as interfaces de entrada e de saída. Quando chega um pacote de interesse, o router verifica a CS procurando os dados correspondentes, se existirem o pacote de dados é logo retornado pela mesma interface pela qual chegou o interesse. Caso contrário, o router procura o nome na PIT, e se uma entrada correspondente existir, registra a interface de entrada na tabela. Se na PIT não houver nenhuma entrada correspondente ao nome, o router encaminha os dados ao produtor com base nas informações da sua FIB, bem como na estratégia de encaminhamento. Este processo pode ser visto na Figura 2.

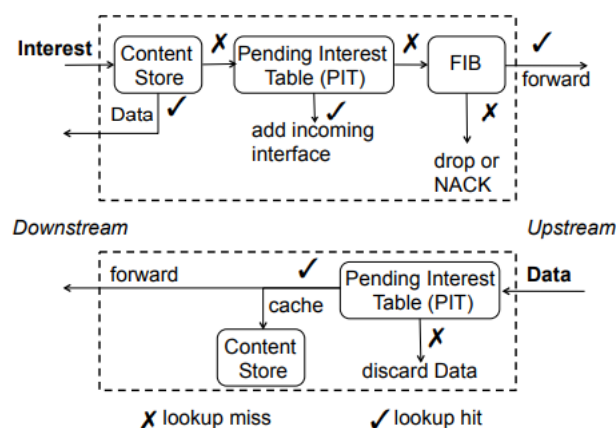


Figura 2: Processo de encaminhamento num nó NDN [4]

Quando um router recebe interesses para o mesmo nome de múltiplos nós ele vai apenas encaminhar apenas o primeiro em direção ao produtor de dados. A FIB é populada por um protocolo baseado no prefixo de nome e pode ter várias interfaces de saída para cada prefixo. A estratégia de encaminhamento pode decidir descartar um interesse em algumas situações, um exemplo, é se todos os links estiverem congestionados ou em caso de suspeição de ataque. Para cada interesse, a estratégia de encaminhamento recupera a entrada correspondente ao prefixo mais longo da FIB e decide quando e para onde encaminhar os interesses. O CS é uma memória *cache* temporária de pacotes de dados recebidos pelo router. Como um pacote de dados NDN é independente de onde vem ou para onde vai, pode ser armazenado em cache para satisfazer futuros interesses. Quando chega um pacote de dados, o router NDN encontra a entrada na PIT e encaminha os dados para todas as interfaces nessa entrada, após isso remove essa PIT e armazena os dados no CS. Para grandes objetos de conteúdo que compreendem vários pacotes, os interesses desempenham um papel importantíssimo no controlo de fluxo como os *ACKs* do

TCP. Nem os pacotes de interesse nem os de dados transportam endereços, os routers encaminham os pacotes de dados com base nos nomes contidos nos pacotes e encaminham pacotes de dados para os consumidores com base nas informações de estado da PIT configuradas em pelos interesses em cada salto. Essa simetria na troca dos pacotes interesse e dados induz um loop de controlo *hop-by-hop*, e elimina a necessidade de qualquer noção de nós.

## 2.2 Sincronização no NDN

O objetivo da sincronização no NDN é reconciliar as diferenças num conjunto de dados partilhado entre múltiplos participantes, ou seja, com o conhecimento de todos os dados disponíveis os utilizadores podem decidir quando e se recuperar dados em falta. À medida que novos dados são gerados pelos produtores, é essencial que os mebers sejam informados sobre alterações. Cada utilizador transmite o seu Estado para os outros participantes através de trocas de Interesse-Dados. Esta troca pode ser feita de duas forma, o produtor notificar ativamente sobre o seu estado, ou cada utilizador consultar periodicamente o grupo de sincronização. Uma abordagem para a reconciliação dos dados é o uso de vetores de estado. A ideia central é representar o estado de um conjunto de dados na forma de um vetor de versão, onde cada componente corresponde ao nome mais recente dos dados de cada produto. Como um vetor de estado enumera diretamente o prefixo de cada produtor e o número de sequência dos dados mais recentes, as diferenças nos conjuntos de dados podem ser identificadas fazendo uma simples comparação entre os valores de cada prefixo. Para atualizar o seu Vetor de Estado, o nó adota o maior número de sequência disponível para cada produtor. O facto de a representação do estado baseada em Vetores de Estado permitir uma comparação direta do estado, permite que os nós identifiquem diretamente os dados em falta sem necessidade de trocas de mensagens adicionais. Esta característica torna a abordagem ideal para redes ad-hoc, onde grandes divergências de estado são bastante comuns. O tamanho do Vetor cresce proporcionalmente ao número de produtores cujos estados precisam ser trocados. Contudo, a listagem explícita dos estados de cada produtor permite a transmissão de Vetores de Estado parciais quando há restrições no tamanho dos pacotes. Como um Vetor de Estado enumera diretamente o prefixo do produtor e o número de sequência dos dados mais recentes para cada produtor em um grupo de Sincronização, conforme mostrado na Figura 3, a diferença nos nomes dos conjuntos de dados pode ser identificada diretamente comparando o número de sequência de cada prefixo de produtor. Em seguida, um nó atualiza seu Vetor de Estado adotando o valor mais alto do número de sequência de cada produtor no Vetor de Estado.

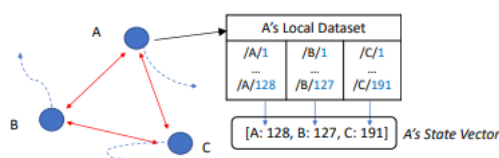


Figura 3: Exemplo de Vetor de Estado [1]



Como o NDN foi concebido para suportar comunicações assíncronas, exigir uma consistência perfeita das informações de associação entre todos os membros do grupo não é viável. Para contornar esta limitação, o sincronização por Vetores de Estado foi rapidamente sucedida por outro protocolo de sincronização, o DSSN. O DSSN trouxe uma alteração simples, mas crucial, permitindo a sincronização de conjuntos de dados em grupos de sensores que entram periodicamente em estados de inatividade. A principal mudança introduzida pelo DSSN foi a reformulação do vetor de estado, que passou de uma simples lista de números de sequência para uma lista de tuplas [prefixo-do-participante, número de sequência]. Com este novo formato, cada Interesse de Sincronização do DSSN transporta um vetor de estado que codifica diretamente o estado global do conjunto de dados compartilhado. Esta abordagem garante que qualquer destinatário pode interpretar corretamente a mensagem DSSN, independentemente do nível de inconsistência de estado entre os participantes. O DSSN foi projetado para ambientes onde a conectividade entre nós fixos é intermitente. A partir deste protocolo, surgiu o DDSN, uma extensão do DSSN otimizada para redes ad hoc sem fio, caracterizadas por uma alta dinâmica no movimento dos nós. Para lidar com estas condições, o DDSN introduziu uma série de melhorias, incluindo:

- 1. Priorização de transmissão que permite determinar quais mensagens devem ser enviadas primeiro durante períodos de conectividade transitória.
- 2. Modo inativo que reduz o tráfego de sincronização e otimiza o consumo de energia quando não há outros nós dentro da área de comunicação.

Apesar da sua eficiência em ambientes altamente dinâmicos e disruptivos, o DDSN tem um desempenho inferior em redes bem conectadas, devido à sua ênfase exclusiva em cenários com conectividade instável. A experiência adquirida com os protocolos anteriores levou ao desenvolvimento do SVSync. Este protocolo herda a codificação do vetor de estado introduzida pelo DSSN, mas elimina funcionalidades específicas para redes de sensores e ambientes com elevada interrupção.

### **2.2.1 State Vector Sync SVSync**

O SVSync adota uma estratégia de codificação de estado distinta. O seu design segue a convenção de nomeação sequencial dos dados e transporta diretamente a representação do namespace do conjunto de dados, codificada em pares [nome do produtor, número de sequência] (figura 4), dentro de cada Interesse Sync Multicast. A decisão de design do SVS de incluir o namespace do conjunto de dados bruto em cada Interesse de Sincronização oferece uma vantagem significativa: qualquer nó que receba o Interesse compreende de imediato o estado do conjunto de dados, independentemente do seu próprio estado atual ou de eventuais mensagens perdidas anteriormente. No entanto, permitir que um Interesse recebido possa alterar o estado de um participante abre a porta a possíveis abusos. Para mitigar esse risco, o SVS exige que todos os Interesses de Sincronização sejam assinados.

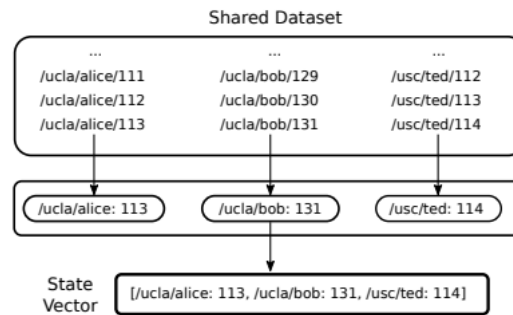


Figura 4: Exemplo de Vetor de Estado [3]

### 2.2.1.1 Funcionamento do SVSync

Um grupo de sincronização utiliza um prefixo multicast de grupo, garantindo que todos os participantes recebem as mensagens de sincronização. Cada membro do grupo possui também um prefixo de publicação específico, sob o qual os seus dados ficam acessíveis aos outros participantes. O SVSync opera com apenas um tipo de mensagem: o Interesse de Sincronização, utilizado para atualizar o estado do conjunto de dados. Estes Interesses de Sincronização são enviados para o prefixo multicast de grupo em duas situações:

- 1. Acionados por eventos, sempre que há uma alteração no estado do conjunto de dados, os participantes notificam imediatamente os restantes.
- 2. Envios periódico, garantem uma visão consistente do conjunto de dados, mesmo em caso de perda de mensagens acionadas por eventos.

O primeiro caso é utilizado para disseminar rapidamente novas informações sobre o estado do conjunto de dados na rede, enquanto o segundo serve para garantir a consistência do estado do conjunto de dados do grupo, mesmo perante perdas de pacotes e desconexões temporárias. Os interesses de sincronização ativados por eventos garantem que o grupo se mantém atualizado sobre o estado do conjunto de dados partilhado. No entanto, o estado de um determinado membro pode ficar desatualizado em relação aos restantes devido a diversas razões imprevisíveis, como perda de pacotes ou falhas temporárias de ligação. Para mitigar este problema, cada entidade de sincronização mantém um temporizador de interesse de sincronização para acionar periodicamente esses interesses, ajudando a manter os membros do grupo sincronizados. O funcionamento desse mecanismo é explicado na figura 5.

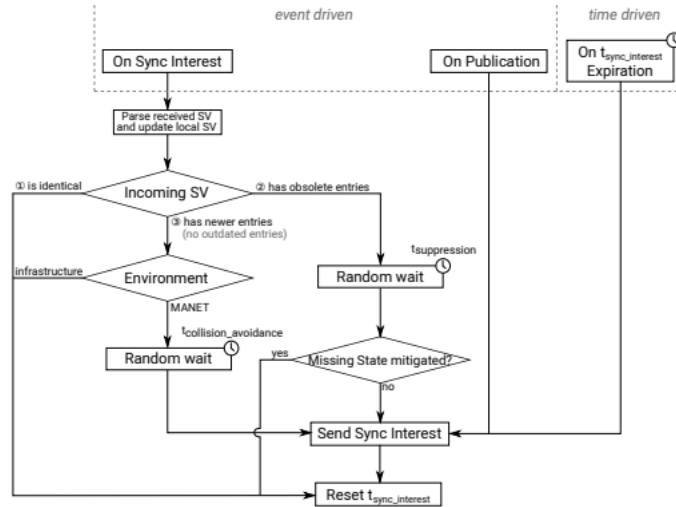


Figura 5: Fluxograma do funcionamento do SVSync [3]

Diferindo dos protocolos anteriores baseados em vetores, os interesses de sincronização SVSync são utilizados exclusivamente como notificações unidirecionais, sem desencadear pacotes de resposta. Todos os protocolos de sincronização utilizam interesses de sincronização para transmitir o estado do conjunto de dados, diferenciando-se apenas na forma como codificam esse estado. Os participantes de um grupo de sincronização transmitem interesses de sincronização multicast para todo o grupo. No entanto, solicitar notificações de mudança de estado através de interesses multicast levanta três desafios principais:

- 1. Como o momento da próxima alteração no estado do conjunto de dados é imprevisível, um interesse de sincronização que solicite uma resposta permanecerá pendente em todos os nós encaminhadores até expirar e ser renovado. Isto cria um estado PIT persistente para cada membro relativamente a todos os outros membros do grupo.
- 2. Um interesse multicast solicita uma resposta de vários produtores potenciais. Se múltiplos produtores responderem quase simultaneamente, devido ao princípio do NDN de um interesse-um dado, apenas uma das respostas será entregue ao remetente do interesse.
- 3. Como consequência do ponto anterior, diferentes membros do grupo poderão receber atualizações distintas, levando a uma divergência do estado do conjunto de dados, que só convergirá após vários ciclos de troca de interesse-dado.

Para resolver estes problemas, em vez de solicitar respostas diretamente através de interesses de sincronização multicast, o SVSync utiliza-os apenas para permitir que cada participante notifique os restantes sobre o seu próprio estado do conjunto de dados.

# Bibliografia

- [1] T. Li et al. «A Brief Introduction to NDN Dataset Synchronization (NDN Sync)». en. Em: *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*. Los Angeles, CA: IEEE, 2018-10, pp. 612–618. isbn: 978-1-5386-7185-6. doi: [10.1109/MILCOM.2018.8599772](https://doi.org/10.1109/MILCOM.2018.8599772). url: <https://ieeexplore.ieee.org/document/8599772/> (acedido em 2025-03-24) (ver p. 4).
- [2] J. M. Lourenço. *The NOVAthesis L<sup>A</sup>T<sub>E</sub>X Template User's Manual*. NOVA University Lisbon. 2021. url: <https://github.com/joaomlourenco/novathesis/raw/main/template.pdf> (ver p. ii).
- [3] P. Moll et al. «A Brief Introduction to State Vector Sync». en. Em: () (ver pp. 6, 7).
- [4] C. Papadopoulos, L. Wang e B. Zhang. «Lixia Zhang Alexander Afanasyev Jeffrey Burke Van Jacobson». en. Em: *ACM SIGCOMM Computer Communication Review* 44.3 (2014) (ver pp. 2, 3).