Capítulo 2

Revisitando as ICNs: Mobilidade, Segurança e Aplicações Distribuídas através das Redes de Dados Nomeados

Leobino N. Sampaio[1], Allan E. S. Freitas[2], Italo V. S. Brito[1,3], Francisco Renato C. Araújo[1], Adriana V. Ribeiro[1]

[1]Dep. de Ciência da Computação – Universidade Federal da Bahia (UFBA)
[2]Dep. Acadêmico de Computação – Instituto Federal da Bahia (IFBA)
[3]Florida International University (FIU)

Abstract

This chapter revisits the theme of Information-Centric Networking by exploring the architecture of Named Data Networking (NDN). NDN imposes changes in data forwarding and routing that makes the architecture more suitable for addressing distributed applications and mobile scenarios, in addition to providing a data-level security mechanism. Thus, the chapter addresses NDN properties and emergent researches on areas of mobility, security, and distributed applications, involving this architecture.

Resumo

Este capítulo revisita a temática das Redes Centradas na Informação ao explorar a arquitetura das Redes de Dados Nomeados (do inglês, Named Data Networking – NDN). A NDN impõe mudanças no encaminhamento e roteamento de dados que tornam a arquitetura mais apropriada para endereçar aplicações distribuídas e aplicações em cenários de mobilidade, além de proporcionar mecanismos de segurança em nível de dados. Assim, este capítulo aborda as propriedades da NDN e os desenvolvimentos recentes nas áreas de mobilidade, segurança e aplicações distribuídas envolvendo a arquitetura.

2.1. Introdução

Redes Centradas na Informação (do inglês, *Information-Centric Networking* – ICN) [Jacobson et al. 2009] é um modelo baseado em nomes de dados/conteúdo criado como uma

alternativa para atender os requisitos atuais e futuros da Internet. A principal característica das propostas baseadas em nome é a desassociação entre o localizador e o identificador de conteúdo, presente nas redes TCP/IP. Essa característica fundamental impôs mudanças na forma de identificação de *hosts*, encaminhamento, roteamento e segurança. Além disso, viabilizou o surgimento de arquiteturas que possibilitam o desenvolvimento de nativas para lidar com requisitos como segurança e mobilidade.

Diversos projetos (e.g., DONA, CCN/NDN e PSIRP/PURSUIT) foram desenvolvidos com o intuito de endereçar os desafios de criar uma rede baseada em nome. Esses projetos tinham o objetivo de desenvolver uma arquitetura viável para a implantação de ICN. Desta forma, buscavam definir as melhores abordagens de nomeação, roteamento, segurança, entre outros desafios advindos com o novo modelo. Dentre os projetos desenvolvidos, podemos destacar as Redes de Dados Nomeados (do inglês, *Named Data Networking* – NDN), que caracteriza-se pelo forte apoio da indústria, pela disponibilidade de um amplo conjunto de plataformas de códigos para desenvolvimento e experimentação e por uma comunidade científica internacional fortemente ativa.

A arquitetura NDN trata-se de uma proposta *clean-slate*, i.e., "tecnologia disruptiva", que oferece serviços de comunicação a partir de um modelo que é dirigido pelo receptor (*receiver-driven*), adota o encaminhamento baseado em nomes, implementa segurança ao nível dos dados (através de infraestruturas de chaves públicas), baseia-se na difusão seletiva de mensagens (*multicast*) não orientada à conexão, e faz uso de *caching* na camada de rede [Jacobson et al. 2009, Zhang et al. 2018a, Saxena et al. 2016]. Embora em NDN as características mais fundamentais das ICNs tenham sido mantidas, a arquitetura tornou-se madura e se expandiu fortemente nos últimos anos, acompanhando os novos requisitos de aplicações avançadas em diferentes cenários.

As propriedades da NDN que a tornam apropriada para sanar os requisitos atuais são decorrentes de alguns elementos de sua arquitetura: *Pending Interest Table* (PIT), *Forwarding Information Base* (FIB) e *Content Store* (CS). Esses elementos são responsáveis, respectivamente, por manter uma lista de requisições pendentes e possibilitar um plano de encaminhamento *stateful*, i.e., com guarda do estado/contexto, armazenar informações de encaminhamento e realizar *in-network caching*. Além dessas estruturas, cada nó da rede deve possuir um módulo de estratégia de encaminhamento, que será responsável por definir "se", "quando" e "para onde" encaminhar cada pacote de interesse [Jacobson et al. 2009, Zhang et al. 2014].

Os nós da rede NDN fazem uso da FIB, PIT e CS para realizar uma comunicação baseada no padrão arquitetural *publish/subscribe* [Eugster et al. 2003], por meio de um esquema de comunicação assíncrona. Este mecanismo de comunicação impulsiona o desenvolvimento de protocolos de roteamento baseado em nomes, cuja principal função é auxiliar o processo de encaminhamento e disseminar informações de alcançabilidade (prefixos de nomes) [Zhang et al. 2019a]. O roteamento baseado em nomes, em conjunto com as demais propriedades da NDN, permite a implementação de funções nomeadas de rede e de uma plataforma de invocação remota de métodos de forma distribuída e transparente quanto à localização [Król and Psaras 2017, Król et al. 2018]. No entanto, como a NDN utiliza um mecanismo de comunicação assíncrono em um ambiente de alta taxa de volatilidade, é importante manter o estado distribuído do sistema, apesar

de eventuais desconexões, ingressos e saídas de nós. Por tais motivos, protocolos de sincronização têm sido desenvolvidos de modo a viabilizar a comunicação assíncrona quando nem todas as partes podem estar *online* ao mesmo tempo [Li et al. 2018].

A mudança de paradigma proposta na arquitetura NDN, de um modelo de comunicação centrado nos *hosts* para um modelo centrado na informação, também impõe mudanças fundamentais na forma de pensar segurança de redes [Zhang et al. 2018d]. As estratégias e *frameworks* de segurança passam, portanto, a proteger os dados diretamente, e aplicações devem fazer uso da semântica associada ao esquema de nomeação para aplicar os controles de segurança, funções criptográficas e tolerância a falhas. Por conseguinte, novos desafios e problemas clássicos abrem espaço para pesquisas [Mannes and Maziero 2019, Zhang et al. 2018d], como: modelo de confiança, ataques de DDoS em IoT, modelos de ataque, autorização e propriedade na hierarquia de nomeação.

Em cenários de mobilidade, o plano de encaminhamento *stateful* da NDN oferece alternativas mais eficientes de descoberta de recursos [Araújo et al. 2019]. A desvinculação do conteúdo da sua localização é outro aspecto que permite os nós móveis recuperarem dados sem se preocupar onde os mesmos estão hospedados [Jacobson et al. 2009, Zhang et al. 2014]. Por tais motivos, a comunidade tem procurado, através da NDN, soluções leves, escaláveis, seguras e eficientes para redes móveis *ad-hoc* e redes veiculares [Brito et al. 2020, Araujo and Sampaio 2021, Zhang et al. 2019a, Mannes and Maziero 2019].

Este capítulo tem o objetivo de revisitar a temática das ICNs ao explorar a arquitetura NDN considerando três aspectos fundamentais das aplicações emergentes: **mobilidade, segurança e aplicações distribuídas**. Para atingir esse objetivo, o capítulo é iniciado com uma breve revisão dos fundamentos da arquitetura NDN (Seção 2.2). Em seguida, são discutidas as principais questões relacionadas aos três eixos temáticos escolhidos (mobilidade, segurança e aplicações distribuídas) – nas Seções 2.3, 2.4 e 2.5, respectivamente – onde são apresentados desafios de pesquisa, aplicações potenciais e casos de uso. Na Seção 2.6, são apresentados os ambientes de experimentação da arquitetura NDN e as instruções para as atividades práticas. Os principais desafios de pesquisa envolvendo a arquitetura e os eixos temáticos são elencados na Seção 2.7. Por fim, o capítulo é concluído na Seção 2.8.

2.2. Visão Geral do Paradigma ICN e da Arquitetura NDN

Nesta seção serão apresentadas algumas características das ICNs e seu comparativo com a arquitetura TCP/IP. Além disso, será discutida a arquitetura NDN, incluindo aspectos como estrutura dos nós, nomeação, roteamento e encaminhamento de dados. Também será discutido o impacto da integração da NDN com outras arquiteturas e modelos.

2.2.1. Revisão dos fundamentos de ICNs

A arquitetura atual da Internet é baseada no padrão TCP/IP, que tem uma associação entre os serviços/conteúdos disponíveis e sua localização. Desta forma, o usuário precisa saber onde o conteúdo/serviço está localizado. A ideia básica da ICN é desacoplar localizador e identificador do conteúdo e permitir que as solicitações de conteúdo sejam direcionadas à rede, como um serviço, sem que o usuário necessite saber sua localização [Jacobson et al. 2009]. A mudança de paradigma impacta no comportamento da rede em relação aos

pacotes transmitidos e serviços ofertados. O padrão TCP/IP trafega pacotes endereçados a uma localização, enquanto a ICN trafega pacotes endereçados a um conteúdo. Apesar dessa mudança, a ICN preocupa-se em manter requisitos como simplicidade, robustez e escalabilidade.

Na Figura 2.1, observa-se uma relação entre os serviços e características das arquiteturas TCP/IP e do modelo ICN. Na arquitetura atual (à esquerda), o protocolo IP é responsável por prover a comunicação entre os *hosts* e está no centro do modelo. No lado direito da figura, observa-se a substituição do protocolo IP por pedaços de conteúdos nomeados (*chunks*) que são utilizados na comunicação entre consumidor e produtor. Além disso, identifica-se que outros protocolos comuns nas redes TCP/IP (e.g., IP, UDP e TCP) podem ser usados para facilitar a comunicação em ICN. Pelas características expostas na figura, percebe-se que enquanto a arquitetura TCP/IP visa comunicar dispositivos, o modelo ICN tem o objetivo de prover conteúdos para usuários e aplicações.

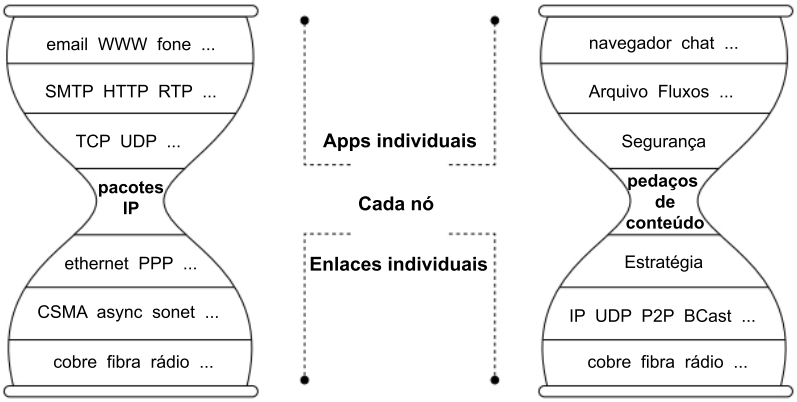


Figura 2.1. Comparação entre a arquitetura TCP/IP e as arquiteturas baseadas no modelo ICN. Traduzido de [Zhang et al. 2014].

Em busca de um modelo ICN para o futuro da Internet, pesquisadores propuseram projetos de arquiteturas centradas na informação. As arquiteturas variam em relação a alguns aspectos, como nomeação e roteamento. Os principais projetos desenvolvidos foram o DONA, PSIRP/PURSUIT e CCN/NDN. Dentre os projetos, destaca-se a importância do NDN, que foi criado há mais de 10 anos e, desde então, tem se consolidado como a principal implementação de ICN.

2.2.2. Arquitetura NDN

NDN é uma abordagem centrada no conteúdo criada para endereçar os requisitos atuais e futuros da Internet [Zhang et al. 2010]. Enquanto a arquitetura atual, centrada no *host*, preocupa-se com a comunicação entre dispositivos, a NDN foca na distribuição de conteúdo. Com esse objetivo, a arquitetura NDN usa uma comunicação inspirada no padrão arquitetural *publish/subscribe*, define novos tipos de pacote de transmissão de dados, estabelece novas estruturas para os nós da rede, faz roteamento baseado em nome e utiliza um plano de dados que armazena estado [Jacobson et al. 2009, Zhang et al. 2018a].

Em NDN, existem três papeis principais que um *host* pode assumir: consumidor de dados, produtor de dados e encaminhador. O consumidor de dados é um dispositivo que envia requisições de conteúdo para a rede. O produtor de dados é o responsável pela

criação e disponibilização dos conteúdos. Enquanto o nó encaminhador atua como um dispositivo intermediário entre esses dois *hosts*. Essas caracterizações de *host* não são excludentes. Portanto, um nó pode atuar como produtor, consumidor ou encaminhador em diferentes momentos [Zhang et al. 2018a].

A comunicação entre os *hosts* é estabelecida através do envio de pacotes de interesse e de dados (Figura 2.2). O pacote de interesse é análogo a uma requisição, já o pacote de dados inclui o conteúdo requisitado. Cada um desses pacotes é definido de acordo com diferentes campos que servem para proporcionar funcionalidades distintas, como prevenção de *loops* e medições de qualidade de serviço da rede [Saxena et al. 2016]. Além dos pacotes de interesse e de dados, a NDN também define o *Negative Acknowledgment* (NACK) para indicar que um nó não pode satisfazer a requisição ou encaminhar o pacote de interesse.

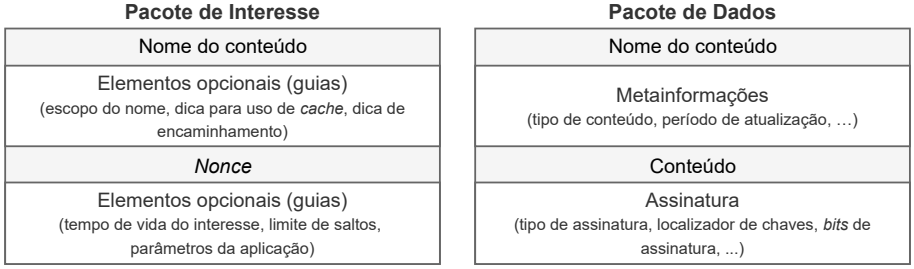


Figura 2.2. Pacotes da arquitetura NDN[1].

A NDN especifica três estruturas fundamentais para o processamento de pacotes: a tabela de interesses pendentes (do inglês, *Pending Interest Table* – PIT), a base de informações de encaminhamento (do inglês, *Forwarding Information Base* – FIB) e uma estrutura de armazenamento de conteúdo (do inglês, *Content Store* – CS). Uma visão geral do código da NDN é apresentada na Figura 2.3. Dentre os principais módulos, destaca-se o *NDN Forwarding Daemon* (NFD) [Afanasyev et al. 2018] que consiste numa estrutura que implementa o protocolo NDN e define como os pacotes de interesse e de dados são trafegados na rede e processados pelos equipamentos. NFD é composto por diversos módulos que, além de implementar as funções de encaminhamento de pacotes, definem estruturas e serviços da arquitetura. Dentre os principais módulos do NFD, é possível citar:

- ***ndn-cxx Library, Core* e *Tools:*** proporcicona funções comuns a diversos serviços do NFD. A biblioteca do ndn-cxx é essencial no desenvolvimento de aplicações NDN, como monitoramento das *faces*, DNS, roteamento e sincronização.

- ***Faces:*** trata-se de uma abstração que engloba tanto interfaces físicas quanto lógicas, incluindo túneis e chamadas do sistema entre camadas da arquitetura NDN.

[1]De acordo com a especificação: `https://named-data.net/doc/NDN-packet-spec/current/index.html`.

[2]Adaptado de: `https://named-data.net/wp-content/uploads/2019/11/5-codebase.pdf#page=5`.

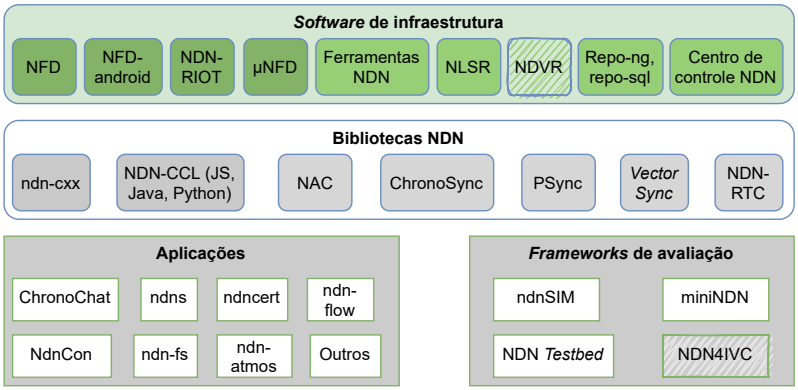


Figura 2.3. Visão geral do código base da NDN[2]. As caixas hachuradas são referentes a trabalhos apresentados em [Brito et al. 2020, Araujo et al. 2021].

- ***Tables:*** implementa as principais estruturas de processamento de pacotes em NDN: a CS, a PIT e a FIB. Além disso, engloba a definição de tabelas que implantam políticas de *cache*, medições e outras funcionalidades da rede.
- ***Forwarding:*** define o fluxo do processamento de pacotes considerando as tabelas implementadas no módulo *Tables*. Esse módulo permite o uso de diferentes estratégias de encaminhamento, como *best-route* e *multicast*.
- ***Management:*** implementa o protocolo de gerenciamento NDN e permite a verificação do desempenho da rede através de registros de informações de monitoramento dos pacotes, como latência.
- ***RIB Management:*** é responsável por gerenciar a base de informação de roteamento (do inglês, *Routing Information Base – RIB)*, que inclui rotas estáticas e dinâmicas.

Na definição da estrutura do NFD, destaca-se a composição interna das *faces* com as camadas de serviço de enlace e transporte. Na Figura 2.4, é possível observar que a camada de transporte é a camada de mais baixo nível. Essa camada é a responsável por prover entrega de pacotes para a camada de serviço. A camada de transporte utiliza a estratégia *best-effort* nessa comunicação, que pode ocorrer através de diferentes mecanismos de entrega, como transporte UNIX, Ethernet, UDP e TCP, a parir da utilização do formato de codificação TLV.

O formato de pacotes de tamanho variável (do inglês, *Time-Length-Value* – TLV) é utilizado para codificar os pacotes da camada de serviço de enlace. Essa camada proporciona um serviço de entrega *best-effort* para os pacotes das camadas de rede (interesse, dados e NACK). A camada de serviço de enlace é composta pelo serviço de enlace genérico e pelo serviço de enlace veicular. O serviço de enlace genérico é a opção padrão, enquanto o serviço de enlace veicular é uma estrutura planejada para possibilitar a implementação de uma estratégia adequada a cenários de redes veiculares.

Considerando o protocolo NDN e sua implementação por meio do NFD, temos a definição de todo o fluxo de processamento de pacotes. O processo é iniciado quando um nó encaminhador recebe um pacote através de uma *face*. As etapas do fluxo de processamento

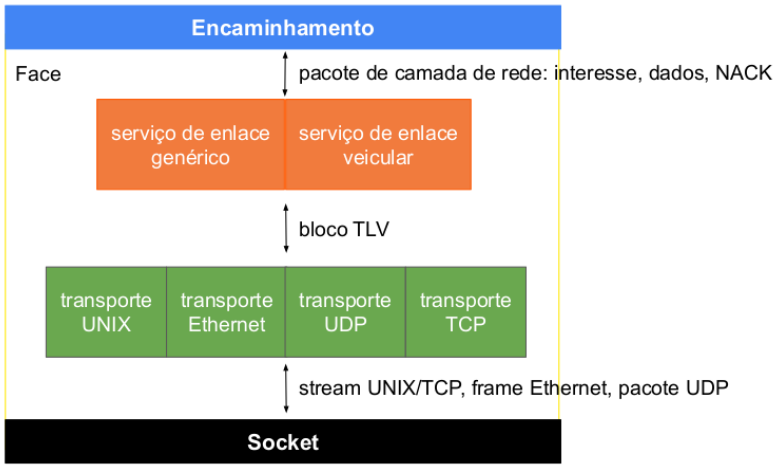


Figura 2.4. Representação da estrutura interna das *faces* NDN. Traduzido de [Afanasyev et al. 2018].

de pacotes varia de acordo com o tipo de pacote (interesse, dados ou NACK) e com a ação que está acontecendo (recebimento, envio, detecção de *loop*, etc.). Para os pacotes de interesse são definidos *pipelines* correspondentes a entrada e saída de pacotes, detecção de *loop* e contabilização de acertos e erros na CS. Para os pacotes de dados são tratadas três ações distintas: dados não solicitados, entrada e saída de pacotes de dados. Já para o NACK são definidos os processos de recebimento e envio.

Durante esses *pipelines*, as estruturas da NDN são utilizadas com finalidades variadas em momentos distintos. Em geral, a PIT possibilita a prevenção de *loop*, agregação de requisições para um mesmo interesse, encaminhamento *multicast*, manutenção de estado das requisições e cálculo de *Round Trip Time* (RTT) a partir das interfaces de saída e tempo de envio [Saxena et al. 2016]. Já a FIB é utilizada parar armazenar as melhores rotas de encaminhamento de pacotes de interesse. *Named-data Link State Routing Protocol* (NLSR) [Wang et al. 2018] é o protocolo padrão utilizado em NDN para popular a FIB. Trata-se de um protocolo que utiliza informações sobre estado de enlace ou roteamento hiperbólico para calcular as rotas e adicioná-las na tabela de roteamento. O NLSR possibilita a adição de várias interfaces de saída para um mesmo prefixo. Com isso, torna-se necessário o estabelecimento de um *ranking* para determinar as melhores rotas e adicioná-las na FIB. As múltiplas rotas também podem ser usadas simultaneamente, para enviar interesses, através do suporte nativo de *multihoming* da NDN [Zhang et al. 2018a].

Além do uso da PIT e FIB, um dos diferenciais da NDN é a adição de *in-network caching*. Em NDN, os nós encaminhadores têm uma CS que mantém uma cópia local dos conteúdos. Esse armazenamento possibilita maior resiliência e melhorias na qualidade de serviço da rede e no tempo de resposta ao usuário. A política padrão de *placement* de *cache* em NDN é a *Leave a Copy Everywhere* (LCE). Isso significa que todos os conteúdos que chegam no nó encaminhador serão armazenados. Como o tamanho da CS é limitado, é comum o uso de uma política de substituição de conteúdo que pode ser determinada considerando diversos contextos [Ioannou and Weber 2016, Pires et al. 2021] e cujo desempenho é uma preocupação sinalizada em diversos trabalhos [Ribeiro et al. 2017, Ribeiro et al. 2018, Pires et al. 2018, Pires et al. 2019].

A arquitetura NDN inclui repositórios de dados distribuídos, que têm o objetivo de armazenar cópias dos conteúdos. Diferente da CS, que se trata de um armazenamento oportunístico, os repositórios são utilizados para armazenar conteúdos específicos. Esse armazenamento pode ser realizado através da inserção estática de conteúdos ou através da configuração de prefixos de conteúdos que devem ser armazenados [Zhang 2019].

As três estruturas básicas da NDN são utilizadas no encaminhamento e roteamento de pacotes [Zhang et al. 2014]. O encaminhamento é feito de forma distinta para pacotes de interesse, de dados e NACK. Na Figura 2.5, são demonstrados os processos de chegada de pacotes de interesse e de dados em um nó encaminhador. Ao receber um pacote de interesse, o nó verifica se há um conteúdo correspondente armazenado localmente na CS. Se houver, o conteúdo é enviado ao nó solicitante e o pacote de interesse é descartado. Nesse cenário, uma cópia local do conteúdo é utilizada para resolver a requisição do usuário e a taxa de acerto do *cache* é incrementada.

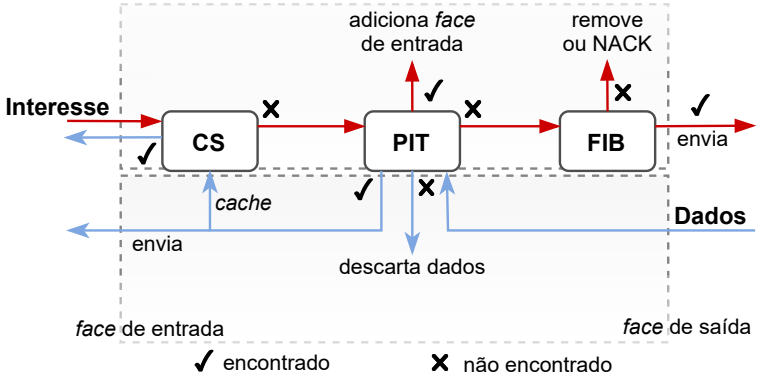


Figura 2.5. Processo de encaminhamento em um nó NDN. Adaptado de [Zhang et al. 2014].

Caso não haja uma cópia do conteúdo solicitado na CS, verifica-se a existência de uma entrada correspondente na PIT. Se existir uma entrada para o interesse, a interface de chegada do novo pacote é adicionada à entrada existente na PIT, caso ainda não tenha sido adicionada, e o novo interesse é descartado. Essa atividade demonstra o mecanismo de agregação de requisições da PIT. Se não existir nenhuma correspondência na PIT, uma nova entrada é criada para o interesse.

Quando uma nova entrada na PIT é adicionada, isso significa que o nó precisa encaminhar o pacote de interesse em busca de identificar um nó produtor que consiga satisfazê-lo. Para isso, a estrutura da FIB é consultada em busca de um próximo salto em direção ao prefixo do conteúdo que o usuário deseja. Se não houver nenhuma entrada na FIB, o pacote é removido ou o nó envia um NACK para a interface na qual o interesse chegou. Caso exista uma entrada na FIB, o pacote de interesse é encaminhado.

É importante ressaltar que, ao adicionar uma entrada na PIT, constrói-se uma trilha salto a salto entre o consumidor e o produtor. Quando o interesse alcança um produtor ou um nó encaminhador que possua o conteúdo em *cache*, o pacote de dados é encaminhado pelo caminho reverso. Ao receber o pacote de dados, o encaminhador verifica sua PIT em busca de interesses pendentes que devem ser satisfeitos. Como a PIT armazena a lista de interfaces de chegada, o pacote de dados pode ser enviado para todas as interfaces de forma

simultânea. Além disso, como a CS faz uso de uma estratégia oportunística, o conteúdo é armazenado na CS dos nós intermediários.

Além das funções básicas supracitadas, o *pipeline* de um nó encaminhador NDN inclui também operações especializadas que dão suporte à arquitetura, como o uso de diferentes estratégias de encaminhamento com base no nome e na FIB (e.g., melhor caminho, múltiplos caminhos, encaminhamento adaptativo, etc.), validação do escopo do nome com relação à *face* de entrada e saída (e.g., `/localhost` para comunicação inter-processo, `/localhop` para comunicação entre vizinhos apenas), política de dados não solicitados, supressão de retransmissão, marcação e uso de *tags* (e.g., *CachePolicyTag* para evitar a CS, *PrefixAnnouncementTag* para roteamento por auto aprendizagem, *IncomingFaceIdTag* para informar à aplicação a *face* de entrada, etc.), medições de RTT, dentre outras.

As estruturas de processamento de pacotes e as propriedades da NDN possibilitam vantagens em diferentes cenários e associação com arquiteturas variadas. Tais vantagens têm motivado propostas que envolvem a adoção da NDN em conjunto com novos modelos, paradigmas e arquiteturas, tais como SDN/P4 e *blockchain*, aplicados em cenários de mobilidade, como VANETs e FANETs.

2.2.3. Integração com outras arquiteturas e modelos

Zhang et al. 2019a exploram a escalabilidade das arquiteturas centradas na informação em relação às redes IP. Nesse trabalho, são considerados diferentes aspectos para avaliar os projetos desenvolvidos na área de ICN. Dentre os observados, é possível citar as abordagens de implementação *overlay*, *underlay* e híbrida. A maioria dos projetos desenvolvidos até 2013 propuseram o uso de ICN como um *overlay* da rede IP. A partir de 2014, tem-se predominância de projetos que propõem a arquitetura como um modelo *underlay* ou híbrido. A análise realizada pelos autores considerando a variedade da forma de implementação e dos outros aspectos elencados (e.g., requisitos da arquitetura atual), demonstram a versatilidade de implementação de arquiteturas ICN e a possibilidade de adaptação a diferentes cenários. Além disso, observa-se que não existe um consenso sobre a integração e interoperabilidade da NDN com outras arquiteturas legadas (e.g., TCP/IP).

Algumas iniciativas [Madureira et al. 2021, Siracusano et al. 2018, Conti et al. 2020, Nour et al. 2019a, Signorello et al. 2016, Miguel et al. 2018, Karrakchou et al. 2020a] têm explorado os benefícios da NDN a partir da integração com o paradigma SDN e a linguagem P4. Azgin et al. 2016 propõem a utilização da PIT apenas nas bordas da rede, removendo-a dos comutadores NDN do núcleo, de modo a tornar mais eficiente o processamento dos pacotes. Com esta mesma motivação, Madureira et al. 2021 utilizaram a linguagem P4 para desenvolver uma arquitetura e um protocolo denominados NDN-Fab. Através da programação na camada de enlace (L2), a arquitetura estabelece que os comutadores de borda da rede definem, na origem, as rotas de encaminhamento de pacotes NDN no núcleo. Assim, a NDN-Fab possibilita a interoperabilidade entre abordagens centradas no conteúdo e baseadas em localização, aproveitando a escalabilidade da NDN e o desempenho do encaminhamento L2. A NDN-Fab define o controlador de rede como o responsável pelo gerenciamento de PITs e FIBs globais, a partir da sua visão centralizada do núcleo. Já Moiseenko and Oran 2017 propõem o *ICN Path Switching*, que permite o uso de *path discovery* e *path steering*, além de realizar o encaminhamento de pacotes NDN

sem depender de pesquisas LNPM na FIB.

Alguns trabalhos [Signorello et al. 2016, Miguel et al. 2018, Karrakchou et al. 2020a] propõem a integração de todos os recursos da NDN dentro dos comutadores P4. Dada as limitações da linguagem P4, a implementação de todos os recursos previstos na arquitetura NDN não é viável. Por exemplo, a inexistência de estruturas de repetição, dificulta o processamento de pacotes no formato TLV (do inglês, *Type-Length-Value*) [Afanasyev et al. 2018]. Além disso, destaca-se o fato de que as principais estruturas de dados das redes NDN (PIT, CS e FIB) demandam espaços de armazenamento. A implementação dessas estruturas requer a utilização de módulos de memória mais lentos, levando a uma redução no desempenho de comutação de pacotes da rede.

Além das iniciativas de integração com a arquitetura IP e com SDN/P4, destaca-se a relação da NDN com livro-razão distribuído (do inglês, *distributed ledgers* – DLT), implementados sobretudo através da tecnologia *blockchain* [Fotiou and Polyzos 2016, Yu et al. 2017, Mori 2018, de Sousa et al. 2019]. Fotiou and Polyzos 2016 apresentaram um esquema para validação de integridade e fornecimento de conteúdos em NDN que permite que uma identidade (e.g., nome) possa ser usada como uma chave pública. O problema abordado reflete um cenário em que um detentor de um conteúdo deseja compartilhá-lo com alguns consumidores. Yu et al. 2017 realizaram uma discussão inicial do arcabouço NDN DeLorean, com foco na autenticação de arquivos de dados de longa duração em NDN. DeLorean fornece um serviço de auditoria de dados por meio da verificação de assinaturas em um livro-razão público. Por fim, Mori 2018 propôs um esquema para assegurar a integridade e autenticidade da fonte de dados de sensoriamento na *cache* em redes de sensores sem fio baseadas em NDN.

Zhang et al. 2019b propõem um sistema de livro-razão distribuído utilizando um grafo acíclico direcionado e *Proof-of-Authentication*. Esse sistema é construído em cima de uma arquitetura NDN, com o objetivo de facilitar a disseminação de informação entre os *hosts*. Jin et al. 2017 usam a arquitetura NDN em substituição à rede TCP/IP com a justificativa de que a NDN provê melhor suporte para implementação de *multicast* e hierarquias de *status* nas *blockchains*. Por fim, Sedky and Mougy 2018 utilizam NDN para implementar uma *blockchain*, com o intuito de otimizar a troca de informação entre as partes e aumentar a eficiência das transações.

Implementações de NDN têm sido amplamente utilizadas para cenários de redes de maior complexidade, sobretudo envolvendo mobilidade e dispositivos com capacidade limitada de recursos computacionais. Portanto, destaca-se uma tendência recente para a implementação das redes veiculares de dados nomeados (do inglês, *Vehicular Named Data Networks* – VNDN) [Grassi et al. 2014, de Sousa et al. 2018a, de Sousa et al. 2018b, Wang and Li 2020, Rondon et al. 2020, Fang et al. 2018, Wang et al. 2020, Khelifi et al. 2020, Tizvar and Abbaspour 2020, Wang et al. 2021] que são definidas como redes veiculares *ad-hoc* em que os nós (veículos) atuam no armazenamento temporário de conteúdos e encaminhamento de interesses para nós adjacentes, com o intuito de atender a demanda dos consumidores. Com motivações semelhantes, outra linha de trabalhos explora as vantagens da NDN na comunicação em redes de veículos aéreos não tripulados (do inglês, *Flying Named Data Networking* – FNDN) [Araújo et al. 2019, Barka et al. 2018, Serhane et al. 2021].

2.3. Suporte à Mobilidade em Redes de Dados Nomeados

Nesta seção, serão apresentados os aspectos referentes à mobilidade dos nós em ambientes NDN. Inicialmente, serão discutidos os requisitos das redes móveis emergentes e as vantagens oferecidas pela NDN a essas redes; em seguida, serão elencadas algumas questões sobre a mobilidade do consumidor e do produtor de dados, apresentando a classificação de soluções de suporte à mobilidade desses nós em quatro categorias principais.

2.3.1. Requisitos de redes móveis do futuro para NDN

Estima-se que o uso de conexões móveis alcançará mais de 70% da população global em 2023 [Cisco 2020]. Diante do histórico e previsões que demonstram um crescimento cada vez maior de conexões e requisitos distintos das aplicações emergentes, a NDN surge como uma arquitetura promissora com potencial para atender à demanda prevista. Há muitos trabalhos que apontam a ICN como um facilitador de comunicação, possibilitando melhorias de desempenho em redes 5G [Serhane et al. 2021], implantação e desenvolvimento de soluções em IoT [Nour et al. 2019b, Madureira et al. 2020] e computação de borda [Psaras et al. 2018]. Dentre as principais características da NDN que favorecem o suporte à mobilidade dos nós, podemos citar: necessidade de dados seguros e nomeados na camada de rede, armazenamento em *cache*, uso de repositórios, agregação de requisições e entrega *multicast*.

Dados seguros e nomeados na camada de rede

Um grande diferencial da NDN em relação à arquitetura TCP/IP é a substituição de endereços IP na camada de rede por dados nomeados, além da segurança aplicada aos dados em vez do canal de comunicação [Zhang et al. 2014]. O uso de dados nomeados na camada de rede torna possível a separação do identificador e localizador dos dados. Uma vez que os dados são seguros e tornam-se independentes de localização, qualquer nó com capacidade de armazenamento pode manter uma cópia local desses dados ao recebê-los.

O armazenamento na rede permite que os dados dos usuários permaneçam disponíveis, mesmo quando os dispositivos do usuário estão *offline*. Como é mais viável considerar que os dados estejam sempre disponíveis em algum lugar na rede do que os dispositivos estejam sempre *online*, é esperado que a NDN impulsione novas gerações de aplicações verdadeiramente P2P por meio do armazenamento na rede e da comunicação assíncrona [Zhang 2019, Psaras et al. 2018] com suporte nativo de *multihoming* na camada de rede [Zhang et al. 2018a, Amadeo et al. 2016].

Armazenamento em *cache*

A capacidade de armazenamento em *cache* nos nós da NDN é uma das principais vantagens dessa arquitetura em relação à arquitetura TCP/IP. O armazenamento em *cache* é um componente pertencente ao encaminhamento da NDN e cada nó encaminhador desempenha um *cache* oportunístico, no qual armazena passivamente os dados que o nó ajuda a encaminhar [Zhang 2019]. Os nós que armazenam os dados podem atuar como nós de réplica usando a função de armazenamento. Desta forma, os dados armazenados

em *cache* podem ser usados para responder solicitações futuras, independentemente da acessibilidade do produtor original, o que contribui com a mobilidade do produtor [Araújo et al. 2019, Araújo 2018, Araújo et al. 2018].

Este armazenamento em *cache* melhora a recuperação de dados e reduz a latência [Nour et al. 2019b], além de permitir que os nós móveis ajudem no processo de distribuição de dados, simplesmente ao mudarem de rede, desempenhando o papel de mula de dados [Araujo and Sampaio 2021]. Contudo, apesar do armazenamento em *cache* proporcionar vantagens significativas à mobilidade dos nós, apenas a adoção de *cache* pode ser insuficiente e outros tipos de armazenamento são requeridos para que possam ser usados por todas as aplicações que almejam disponibilizar seus dados [Zhang 2019].

Repositórios

Diferente da CS (*cache)*, os repositórios (*repos*) são elementos da arquitetura NDN que visam o armazenamento persistente de dados [Zhang 2019]. Um *repo* pode ser um dispositivo independente ou um módulo conectado a um dispositivo com outra finalidade, como produtores de dados ou nós roteadores [Zhang 2019]. Os *repos* são armazenamentos gerenciados que são instruídos sobre quais conteúdos armazenar e buscam proativamente por esses conteúdos na rede. O gerenciamento de um *repo* determina quais conteúdos devem ser carregados para o seu armazenamento, conforme os prefixos de nome dos conteúdos informados na configuração do *repo* [Zhang 2019].

De acordo com Psaras et al. 2018, com a expansão da IoT e da era da computação de borda, é necessário um modelo de comunicação centrado em dados que atenda o real objetivo das aplicações neste contexto, que é o acesso aos dados. Os autores defendem a implantação de *repos* na borda da rede (e.g., em pontos de acesso Wi-Fi ou similares) que possam ser usados para armazenar temporariamente os dados gerados pelos usuários e dispositivos da IoT, antes da sincronização com a nuvem – a sincronização deve ocorrer apenas quando necessário seguindo a melhor estratégia de acordo com os dados e requisitos da aplicação. A adoção de *repos* na borda tem potencial para reduzir a largura de banda necessária para enviar os dados e os custos financeiros com a sincronização com a nuvem. Além disso, pode prover suporte à mobilidade, uma vez que o *repo*, ao armazenar os dados dos produtores móveis, passa a responder as solicitações para esses dados e assim torna transparente a mobilidade do produtor [Psaras et al. 2018].

Agregação de requisições

As solicitações de pacotes de interesse para o mesmo pacote de dados, oriundas de diferentes consumidores ou não, podem ser agregadas na PIT dos nós. Essa ação evita a necessidade de todas as requisições terem que alcançar o nó produtor ou detentor dos dados solicitados. A agregação de requisições permite que apenas a primeira requisição de cada interesse seja encaminhada rumo à fonte de dados causando uma redução significativa do tráfego da rede [Ioannou and Weber 2016, Shannigrahi et al. 2017]. Ao se mover de uma rede para outra, um nó móvel pode ter suas requisições agregadas e atendidas a partir do primeiro roteador comum à sua rede antiga e atual, não havendo uma comunicação fim

a fim entre o nó móvel e o produtor dos dados.

Ao analisar uma base de dados científicos, – mais especificamente, dados climáticos usados por cientistas de todo o mundo – Shannigrahi et al. 2017 comprovaram que os padrões de solicitações são realmente agregáveis e podem reduzir a carga no servidor. Os autores investigaram a NDN sob a perspectiva de um sistema de distribuição de dados científicos e comprovaram que a agregação de interesses pode ser útil em cenários de alto tráfego, principalmente quando combinada às técnicas de *cache* e estratégias de encaminhamento. A NDN pode melhorar a entrega de dados para os usuários finais e, ao mesmo tempo, reduzir a carga nos servidores e na rede [Shannigrahi et al. 2017].

Entrega *multicast*

A entrega *multicast* de dados acontece – como uma consequência secundária da agregação de interesses de diferentes origens na PIT dos nós – quando um nó com interesses agregados recebe o pacote de dados solicitado, i.e., o dado é replicado no nó e encaminhado a cada interface de entrada dos interesses pendentes agregados na PIT. Dessa forma, um único pacote de dados consegue responder a todos os interesses pendentes agregados e correspondentes ao dado [Amadeo et al. 2016].

A NDN suporta o *multicast/anycast* naturalmente a partir da camada de rede, e quaisquer solicitações insatisfeitas durante um evento de mobilidade podem ser reemitidas sem a necessidade de soluções complexas de *handoff* como as empregadas no IP [Nour et al. 2019b]. O *multicast* nativo também atende ao objetivo de reduzir a quantidade de tráfego e as interações com os nós com restrição de energia, beneficiando principalmente os dispositivos da IoT [Amadeo et al. 2016].

2.3.2. Mobilidade de consumidor

Em NDN a mobilidade do consumidor é naturalmente suportada através da própria arquitetura [Zhang et al. 2016, Zhang et al. 2018b, Araújo et al. 2019]. Os consumidores simplesmente podem requisitar novamente os interesses para os dados não recuperados e a rede se responsabiliza pela entrega desses dados aos consumidores, independentemente de sua localização. Isso é possível graças ao esquema de trilha – i.e., "migalhas de pão" (do inglês, "*bread crumbs*") – deixado pelos interesses na PIT dos nós salto a salto ao longo do caminho do percurso do interesse em direção à fonte dos dados. As trilhas na PIT possibilitam o encaminhamento dos dados de volta ao consumidor, efetivando a entrega dos dados requisitados [Jacobson et al. 2009].

Tanto os consumidores móveis quanto os consumidores estáticos se beneficiam da funcionalidade do plano de encaminhamento com estado baseado em trilhas na PIT. Contudo, no caso de consumidores móveis, pode haver uma peculiaridade nas situações em que o consumidor troca de rede (i.e., processo conhecido como *handoff* ou *handover*) durante uma solicitação. Por exemplo, se um consumidor móvel emitir um interesse a partir de uma rede e, antes que o dado seja recebido, ocorrer o *handoff* desse consumidor para uma outra rede, a aplicação no consumidor tende a requisitar novamente os dados não recuperados que, por sua vez, serão entregues na rede atual e na rede antiga do consumidor. Isso acontece justamente pelo fato dos dados seguirem as trilhas encontradas na PIT dos

nós e especialmente devido às seguintes razões:

- **Entradas PIT ativas:** cada entrada PIT representa uma trilha para encaminhar os dados recuperados de volta ao solicitante. As entradas PIT são removidas em duas situações: (i) quando um pacote de dados correspondente chega ao nó ou (ii) quando a entrada PIT atinge seu tempo de vida (do inglês, *lifetime*) sem que um dado correspondente tenha sido recebido.
- **Privacidade dos consumidores:** na NDN, as requisições não carregam informações que identifiquem o consumidor [Zhang et al. 2014] – diferente das redes IP em que os pacotes transportam os endereços IP de origem e destino – portanto, as duas requisições do consumidor móvel (i.e., antes e depois do *handoff*) são tratadas pela rede como sendo totalmente independentes.
- **Fonte de dados alcançável:** é necessário que pelo menos uma fonte de dados (e.g., nó produtor ou nó detentor) esteja alcançável a partir das redes em que as requisições foram feitas para que os dados sejam recuperados. Os protocolos de roteamento são responsáveis por manter as informações de alcançabilidade dos nós atualizadas na tabela FIB de cada nó da rede.

Algumas das características da NDN discutidas anteriormente podem facilitar a mobilidade dos nós. O plano de encaminhamento com estado da NDN possibilita que os consumidores apenas requisitem novamente os dados não recuperados, mesmo após uma mudança de rede. Porém, essa simplicidade no suporte à mobilidade do consumidor não se aplica no suporte à mobilidade do produtor [Araújo and Sampaio 2017].

2.3.3. Mobilidade de produtor

O modelo de comunicação da NDN é focado na recuperação de dados em vez da entrega de pacotes. O foco na recuperação de dados possibilita que o suporte à mobilidade do produtor se concentre no encontro dos interesses com os dados gerados pelos produtores móveis [Zhang et al. 2016]. Os dados nomeados na camada de rede e a segurança aplicada diretamente aos dados facilitam a replicação e armazenamento em *cache* no caminho (*on-path caching*), *cache* fora do caminho (*off-path caching*) [Ioannou and Weber 2016] e nos repositórios [Zhang 2019], o que aumenta as chances do interesse encontrar os dados, mesmo que o produtor original esteja indisponível [Araújo et al. 2019, Araújo 2018]. Para possibilitar que os interesses encontrem os dados do produtor móvel, Zhang et al. 2016 apontam duas direções possíveis:

1. **Perseguição do produtor:** usa um elemento auxiliar (i.e., um "ponto de encontro") para descobrir o paradeiro do produtor na rede. Os interesses são direcionados a um produtor móvel para recuperar dados (os interesses não precisam alcançar o produtor, caso encontrem os dados solicitados na *cache* de algum roteador ao longo do caminho). Essa direção é semelhante ao suporte à mobilidade IP e as abordagens baseadas em mapeamento e rastreamento podem ser adaptadas para serem empregadas na NDN. Com exceção das abordagens baseadas em roteamento, visto que é improvável que os produtores móveis anunciem o prefixo dos dados no sistema de roteamento na escala da Internet [Zhang et al. 2016].

2. **Encontro de dados:** visa garantir que os dados produzidos por produtores móveis sejam facilmente encontrados. Essa direção pode potencializar a centralização em dados da NDN, uma vez que os dados podem ser desassociados do produtor (e.g., os dados podem ser movidos para um local estacionário e de fácil acesso) e aplicações que geram dados em um local, podem tornar o nome dos dados independente do produtor móvel e atrelado a uma região estacionária [Zhang et al. 2016].

As principais abordagens de soluções de suporte à mobilidade do produtor estão representadas na Tabela 2.1 e são discutidas individualmente nas subseções seguintes. Por questões didáticas e de padronização, durante o restante desta seção serão usados os termos mapeamento, rastreamento, depósito de dados e local de dados para classificar as abordagens de mobilidade do produtor, conforme terminologia adotada em Zhang et al. 2016 e Araújo 2018.

Tabela 2.1. Abordagens de mobilidade do produtor [Zhang et al. 2016, Araújo 2018].

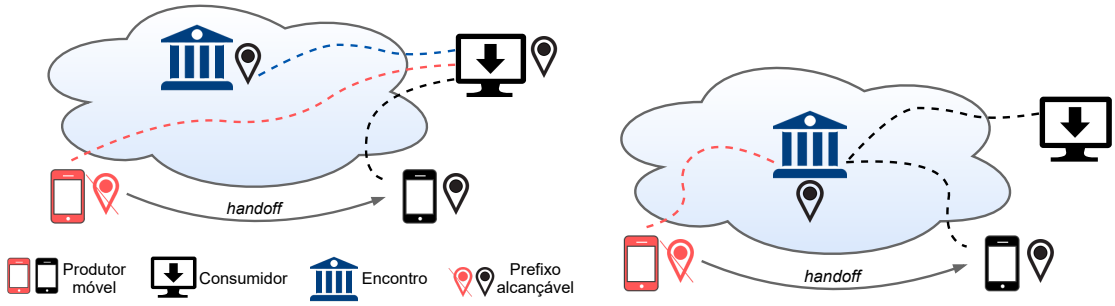
Perseguição do produtor móvel	
Mapeamento	O produtor informa ao ponto de encontro qual o ponto de fixação onde seus dados podem ser recuperados.
Rastreamento	O produtor cria uma trilha de migração para alcançá-lo, que deve ser seguida por interesses do ponto de encontro.
Encontro de dados	
Depósito de dados	Os dados produzidos pelo produtor móvel são movidos para um servidor estacionário conhecido.
Local de dados	Os dados são produzidos e disponibilizados em uma região estacionária.

Perseguição do produtor: Mapeamento

As soluções baseadas em mapeamento herdam as ideias do suporte à mobilidade IP, usando pontos de encontro estáveis para alcançar o produtor móvel. Nas abordagens baseadas em mapeamento, o produtor precisará enviar informações ao ponto de encontro sempre que fizer *handoff*. Assim, cada produtor móvel deve informar o nome do seu ponto de fixação atual "temporário" ao ponto de encontro "estável" [Zhang et al. 2016]. Os trabalhos dessa abordagem se dividem em duas dimensões com base na função do ponto de encontro e na forma como os nomes de pontos de fixação mapeados são transportados nos pacotes de interesse.

Função do ponto de encontro: pode ser um serviço que mapeia os nomes dos dados produzidos por um produtor móvel para o nome do ponto de fixação atual do produtor; um agente doméstico que faz o tunelamento de interesses em direção aos produtores móveis; ou um sistema híbrido que engloba o mapeamento de nomes e o mecanismo de tunelamento de interesses [Zhang et al. 2016]. A Figura 2.6(a) mostra quando o ponto de encontro desempenha o serviço de mapeamento de nomes, primeiro os consumidores consultam o ponto de encontro para obter o mapeamento entre o nome

dos dados e o nome do ponto de fixação atual do produtor e, em seguida, os próprios consumidores encapsulam os interesses em direção ao ponto de fixação obtido [Zhang et al. 2016]. Já na Figura 2.6(b), o ponto de encontro atua como um agente doméstico do produtor móvel, onde os dados são publicados sob o nome do ponto de encontro estacionário e globalmente alcançável. Os interesses dos consumidores alcançam o ponto de encontro que faz o tunelamento dos interesses recebidos em direção ao produtor móvel, no seu ponto de fixação, com base nas informações de mapeamento [Zhang et al. 2016]. Por fim, no modo híbrido o ponto de encontro atua como um agente doméstico para encaminhar, via túnel, o primeiro interesse ao produtor móvel, e o pacote de dados retornado traz o nome do ponto de fixação atual do produtor; que pode ser usado pelo consumidor para enviar interesses, via túnel, diretamente ao produtor [Zhang et al. 2016].



(a) Serviço de mapeamento. (b) Agente doméstico.

Figura 2.6. Abordagem de mobilidade baseada em mapeamento do produtor, a função do ponto de encontro. Adaptado de [Zhang et al. 2016, Araújo 2018].

Buscar dados através de nomes de pontos de fixação mapeados: para orientar o interesse que carrega um nome inalcançável para o ponto de fixação atual do produtor, o nome do ponto de fixação precisa ser anexado ao interesse. Para esse fim, há duas opções: (i) colocar o nome do ponto de fixação como prefixo ao nome dos dados transportado no interesse ou (ii) acrescentar um novo campo no pacote de interesse para carregar o nome do ponto de fixação, como uma dica de encaminhamento [Zhang et al. 2016].

Perseguição do produtor: Rastreamento

As soluções baseadas em rastreamento estendem o plano de encaminhamento com estado para criar a trilha de migração e recuperar interesses dos consumidores. Semelhante à abordagem de agente doméstico, o rastreamento requer que os dados sejam publicados sob o prefixo globalmente alcançável do ponto de encontro. Sempre que o produtor se move, envia interesse de comando de rastreamento ao ponto de encontro para manter o caminho reverso entre sua localização atual e o ponto de encontro.

Os interesses de comando de rastreamento recuperam interesses dos consumidores, em vez de dados. Os interesses regulares dos consumidores serão encaminhados rumo ao ponto de encontro, mas podem seguir em direção ao produtor caso encontrem o rastreamento no caminho, como mostra a Figura 2.7. Dessa forma, o ponto de encontro não precisa necessariamente participar da comunicação entre consumidor/produtor; e se

encarrega de fazer anúncios de roteamento para o prefixo do nome de dados visando atrair os interesses dos consumidores para os dados do produtor móvel e o interesse de comando de rastreamento para si [Zhang et al. 2016].

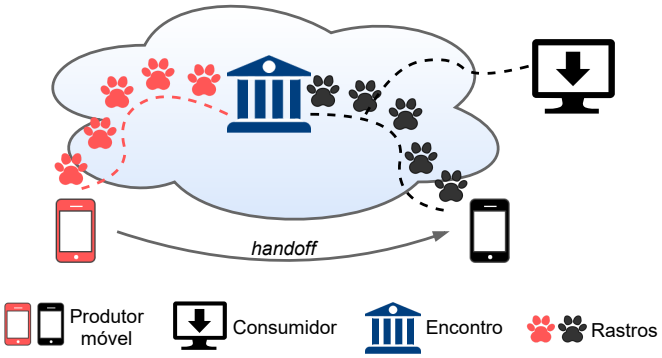


Figura 2.7. Abordagem de mobilidade baseada em rastreamento do produtor. Adaptado de [Zhang et al. 2016, Araújo 2018].

Quando o produtor se move e a nova trilha de migração (i.e., rastro) cruza com um rastro anterior, os interesses dos consumidores que estão pendentes no rastro antigo podem ser encaminhados para o novo rastro. Outra abordagem se baseia no fato do produtor enviar interesse de comando de rastreamento para, além do ponto de encontro, o seu ponto de fixação anterior para buscar os interesses pendentes e minimizar a interrupção da busca de dados. O processo de rastreamento pode ser feito de diferentes formas, incluindo o rastreamento na FIB e na PIT [Zhang et al. 2016].

Rastreamento na FIB: introduz uma FIB dedicada (tFIB) separada da FIB padrão. A tFIB é atualizada pelos interesses de comando de rastreamento. Interesses regulares serão encaminhados seguindo o rastreamento, se existir correspondência na tFIB [Zhang et al. 2016].

Rastreamento na PIT: estende a funcionalidade da PIT, de recuperar dados, para recuperar interesses. Adiciona um novo campo, `traceName`, no pacote de interesse, reservado ao nome de interesses de comando de rastreamento visando criar um rastro entre ponto de encontro e produtor móvel; e um sinalizador, `traceable`, para indicar se um interesse pode ser rastreado por outros interesses. Após receber um interesse com `traceName`, um roteador primeiro procura uma correspondência do `traceName` com as entradas da PIT. Se houver uma correspondência, o interesse será encaminhado através da interface de entrada do interesse do comando de rastreamento, caso contrário será encaminhado usando o mecanismo padrão [Zhang et al. 2016].

Encontro de dados: Depósito de dados

Como a NDN permite que os dados sejam facilmente separados dos produtores originais, em vez de perseguir o produtor móvel, uma alternativa é mover os dados gerados por esses produtores para uma localização acessível (e.g., estacionária) [Zhang et al. 2016], como representado na Figura 2.8, onde os interesses do consumidor são encaminhados para o depósito para buscar dados e podem encontrar o rastro do produtor móvel no caminho.

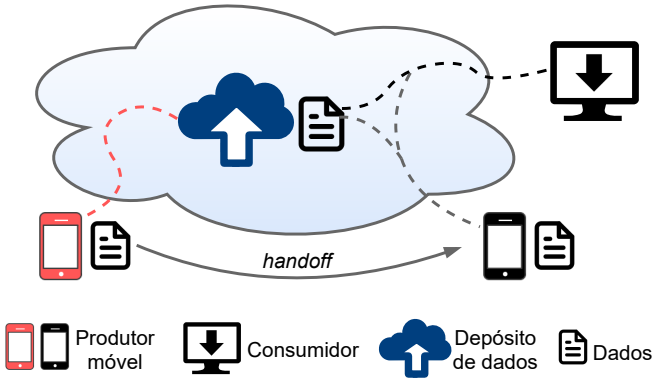


Figura 2.8. Abordagem de mobilidade baseada em depósito de dados. Adaptado de [Zhang et al. 2016, Araújo 2018].

Um depósito de dados se assemelha a um agente doméstico das soluções baseadas em mapeamento, exceto pelo fato que o depósito de dados assume a responsabilidade por hospedar os dados em vez de simplesmente encaminhar interesses [Zhang et al. 2016]. Por exemplo, o depósito pode ser configurado para armazenar sob demanda os dados dos usuários e, ao receber uma solicitação para recuperar um determinado dado, retornará o dado se este já tiver sido carregado no depósito, caso contrário, o depósito tentará recuperar o dado solicitado usando técnicas de mapeamento ou rastreamento.

Um depósito de dados representa um encontro baseado em nomes, onde os interesses e dados são atraídos para se encontrarem. A Internet atual desempenha a função de depósito na camada de aplicação. O serviço de armazenamento na nuvem é um exemplo de encontro baseado em nomes e depósito de dados. Diferente da rede atual, a NDN suporta o encontro baseado em nomes na camada de rede, permitindo que o depósito anuncie o prefixo dos dados na tabela de roteamento [Zhang et al. 2016].

Encontro de dados: Local de dados

Em algumas aplicações e ambientes de rede, os dados estão associados a uma região geográfica específica e podem ser gerados por qualquer produtor no local, como mostra a Figura 2.9. Por exemplo, em aplicações de VNDN [de Sousa et al. 2018a], os dados sobre as condições da rodovia em um determinado local podem ser gerados por qualquer veículo presente naquela região [Zhang et al. 2016].

O encaminhamento de interesses para recuperar os dados pode usar geo-roteamento ou contar com as RSUs (*Road Side Units*) para anunciar os prefixos de dados da região no sistema de roteamento. Quando um produtor na região recebe um interesse, pode usar informações de GPS (*Global Positioning System*) para gerar o dado e responder a solicitação recebida. O produtor também pode responder os interesses recebidos se houver dados correspondentes em *cache*. Uma vez que o produtor sai da região do local dos dados, deixa de receber requisições para os dados da região, que podem ser respondidas por outros produtores móveis daquele local [Zhang et al. 2016].

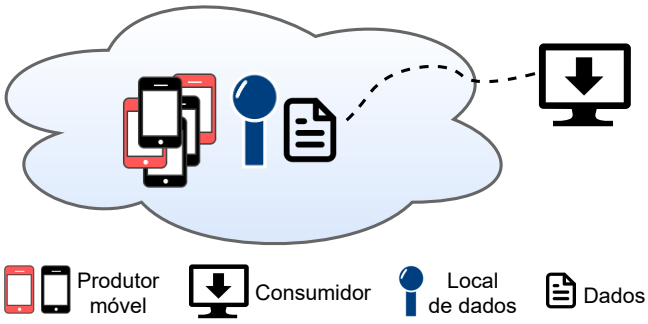


Figura 2.9. Abordagem de mobilidade baseada em local de dados. Adaptado de [Zhang et al. 2016, Araújo 2018].

2.4. Segurança em NDN

Nesta seção serão apresentados os principais conceitos em relação aos modelos de confiança, aplicações de segurança, superfícies de ataque e soluções de proteção disponíveis em NDN. A adoção de uma nova arquitetura de redes implica em uma nova forma de desenvolver aplicações, estratégias de armazenamento e manuseio das informações. Portanto, tais aspectos também serão endereçados em uma discussão geral ao final da seção.

2.4.1. Segurança nativa na arquitetura NDN

O modelo de comunicação centrado na informação leva a uma mudança fundamental no projeto e adoção de estratégias de segurança: ao invés de proteger o canal de comunicação, a arquitetura NDN provê maneiras de proteger os dados diretamente, independente do meio de transmissão e armazenamento. No alicerce do arcabouço de segurança da arquitetura NDN está a criptografia de chave pública e os certificados digitais [Zhang et al. 2018d].

Criptografia de chave pública, ou criptografia assimétrica, é um mecanismo criptográfico no qual utiliza-se um par de chaves – chave pública e chave privada – no processo de ciframento e deciframento de informações [Stallings 2020]. Os certificados digitais, por sua vez, servem para atestar a autenticidade da chave pública de uma entidade [Stallings 2020], condição essencial para uso de sistemas criptográficos assimétricos. A arquitetura NDN provê mecanismos flexíveis que automatizam e garantem a corretude do processo de gerenciamento e operação das chaves e certificados [Yu et al. 2015, Zhang et al. 2018d].

Para entender o processo de gerenciamento e operação de chaves criptográficas na NDN, considere que toda aplicação e todo componente participando de uma comunicação NDN é uma "entidade" e toda entidade tem propriedade sobre um ou mais espaços de nomes. Uma entidade garante a propriedade sobre um espaço de nomes através de um certificado digital, e pode delegar um ou mais sub-prefixos do seu espaço de nomes emitindo um certificado para outra entidade. Um espaço de nome que possui um certificado associado é denominado "identidade". Assim, os principais componentes do arcabouço de segurança da NDN são:

- **Chaves criptográficas:** podem ser vistas como um conjunto de *bytes* qualquer identificado por um nome, porém com uma semântica especial: tais *bytes* são usados criptograficamente para assinar ou cifrar outros *bytes*. Ao tratar as chaves

criptográficas como um dado nomeado qualquer, a arquitetura NDN faz uso de todos os seus componentes na troca de chaves entre entidades.

- **Certificado digital:** é usado para correlacionar uma chave pública a um prefixo de nome e, assim, permitir a validação de sua autenticidade e propriedade. O nome do certificado segue uma convenção de nomeação com "`/<prefix>/KEY/-<key-id>/<issuer-info>/<version>`", onde `/prefix` é o prefixo de nome ao qual a chave pública será atrelada, `key-id` é um identificador da chave, `issuer-info` são informações sobre o emissor e `version` é a versão do certificado. É importante ressaltar que o nome atrelado à chave não necessariamente tem relação com o emissor, o que flexibiliza o esquema de nomeação mas requer o complemento de políticas de validação para verificação da propriedade.
- **Política de confiança:** As aplicações NDN definem uma política de confiança, também conhecida como política ou regras de validação, que especifica quais entidades são confiáveis para produzir quais pedaços de dados, quais chaves devem ser usadas em quais espaços de nomeação e para quais propósitos. Por exemplo, uma aplicação NDN de roteamento que produz dados sobre alcançabilidade de prefixos define uma política de confiança que especifica que os dados produzidos em um roteador devem ser assinados pela chave específica do roteador e tal chave não pode ser usada para assinar outras chaves; ao invés disso, as chaves dos roteadores são assinadas pela chave específica da organização que gerencia os roteadores em questão.

Assim, valendo-se de um esquema de nomeação bem estruturado e semanticamente expressivo, a arquitetura NDN permite que aplicações sejam desenvolvidas seguindo convenções de nomeação para chaves/certificados. Essas convenções viabilizam um processo sistemático de assinatura, validação, cifragem e decifragem de dados [Zhang et al. 2018d]. Além disso, o esquema de nomeação permite automação e melhora a usabilidade do processo de gerenciamento de chaves: é possível identificar automaticamente qual chave foi usada para assinar um dado, utilizar pacotes de interesse/dados para obter o certificado equivalente àquela chave, verificar a confiança/autoridade da chave em relação ao dado produzido e da própria chave em relação à cadeia de confiança. Com isso, os dados são protegidos diretamente, independente do meio de transmissão/armazenamento.

Na Figura 2.10 é ilustrado um modelo de confiança para uma aplicação de *blog* em NDN provido por Yu et al. 2015. Neste exemplo, a aplicação requer que os artigos publicados sejam assinados por um autor. Os autores, por sua vez, possuem chaves assinadas pelos administradores do *site*. Tais administradores possuem chaves que são assinadas pelo dono da aplicação ou por outros administradores. A chave do dono da aplicação é considerada uma âncora de confiança e pressupõe-se que ela está pré-instalada – ou pode ser obtida por um canal seguro – nos consumidores.

A arquitetura NDN também provê mecanismos nativos para garantir a disponibilidade dos dados: o dado pode ser obtido de forma segura e independente da localização, seja através do produtor, ou através de repositórios e *caches* oportunísticos [Zhang et al. 2018d]. A disponibilidade dos certificados segue o mesmo princípio da disponibilidade dos dados, uma vez que um certificado basicamente é um conjunto de *bytes* nomeados e com uma semântica especial. Adicionalmente, a arquitetura NDN possui APIs que permitem

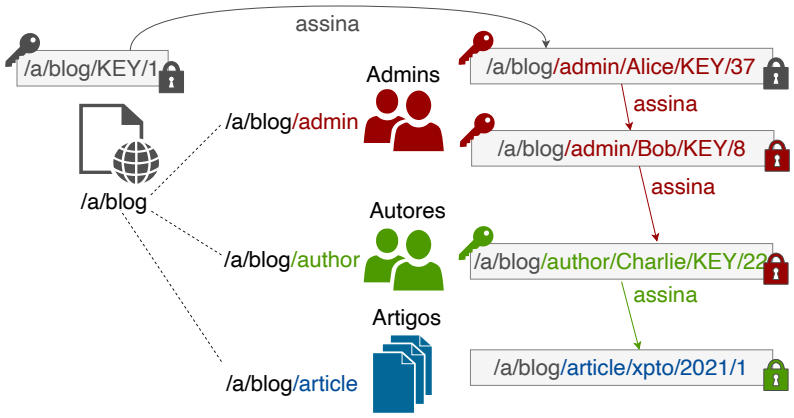


Figura 2.10. Modelo de confiança para aplicação de *blog*. Adaptado de [Yu et al. 2015].

a um produtor acumular um conjunto de certificados e distribuí-los em um único pacote de dados [Zhang et al. 2018d], facilitando sua obtenção. Conforme será discutido na Seção 2.4.4, ataques podem explorar componentes auxiliares da arquitetura, como os nós encaminhadores, para afetar a disponibilidade.

2.4.2. Inicialização do modelo de confiança, das chaves e das políticas

O arcabouço de segurança da arquitetura NDN usa um conjunto de estratégias criptográficas bem consolidadas, um esquema de nomeação expressivo e um conjunto de políticas de confiança que empoderam os desenvolvedores de aplicações na definição de regras de validação que, em última instância, garantem a autenticidade, integridade e confiabilidade dos dados. O desafio reside, portanto, na inicialização desses componentes, em particular, como obter a âncora de confiança, os certificados, as políticas de confiança e aplicar mecanismos de controle de acesso.

Os certificados que compõem a âncora de confiança possuem confiabilidade intrínseca [Stallings 2020] que, por sua vez, é usada para derivar a confiança em outros certificados. Desta forma, sua inicialização torna-se uma etapa crítica para garantir a segurança do sistema. Visando prover suporte a uma vasta gama de casos de uso, a arquitetura NDN permite que cada aplicação defina como será realizada a inicialização da âncora de confiança [Zhang et al. 2018d]. De forma geral, o processo deve ser realizado de forma segura, tipicamente por um canal fora da banda (do inglês, *out-of-band*). O caso mais simples é inserir as entradas na âncora de segurança manualmente ou durante a instalação do nó ou da aplicação NDN [Zhang et al. 2018d]. Outra opção é através de interações presenciais e uso de mecanismos auxiliares como *QR code* [Gawande et al. 2019], ou ainda pela comunicação direta entre os nós em uma rede P2P [Zhang et al. 2018d].

A obtenção dos certificados é feita através do envio de um pacote de interesse pelo nome da identidade, seguido pelo pacote de dados com o certificado obtido do produtor ou do *cache* oportunístico. A depender da aplicação, os certificados podem ser armazenados em um repositório central (e.g., aplicações de nuvem) ou obtidos diretamente de outros nós (e.g., aplicações distribuídas/P2P). A semântica associada ao certificado e sua aplicabilidade para assinar dados serão validadas através das políticas de confiança, que também podem ser instaladas manualmente na aplicação [Zhang et al. 2018d] ou obtidas

dinamicamente através de pacotes de interesse/dados [Yu et al. 2015]. Neste último caso, a API de validação dependerá de, ao menos, uma política de confiança básica para validar outras políticas de confiança [Yu et al. 2015]. Por exemplo, a política de confiança básica pode definir que pacotes de dados do escopo da políticas de confiança devem ser assinados por uma chave da âncora de confiança, garantindo assim uma inicialização segura.

A partir dos componentes anteriores, é possível alcançar requisitos de controle de acesso na rede, como confidencialidade, integridade, autenticidade e autorização. A autenticidade e integridade dos dados é alcançada através da validação da assinatura digital no pacotes de dados e da validação do nome em relação à política de confiança. O primeiro faz uso de primitivas básicas da criptografia assimétrica e algoritmos de resumo digital (*hash*) [Stallings 2020], enquanto o último utiliza expressões regulares enriquecidas para flexibilizar a validação do esquema de nomeação [Yu et al. 2015]. Embora mais comumente aplicada aos pacotes de dados, os pacotes de interesse também podem ser assinados e validados seguindo as mesmas premissas. Nesses casos, é mais comum uso de criptografia simétrica de chave compartilhada [Li et al. 2019a].

Já a confidencialidade requer um processo ligeiramente mais complexo. A estratégia geralmente utilizada para confidencialidade é baseada no protocolo de troca de chaves Diffie-Hellman [Stallings 2020], que automaticamente estabelece chaves de sessão para cifragem/decifragem ponto a ponto. Tal processo não se aplica para comunicação entre múltiplas partes, que é o caso mais comum em NDN. Para superar essa limitação, algumas propostas fazem uso da semântica do esquema de nomeação para embutir informações adicionais que permitam estabelecer chaves de sessão para múltiplas partes [Zhang et al. 2018d, Zhang et al. 2018c, Marxer and Tschudin 2017]. A ideia básica consiste em definir uma nova entidade chamada "gerenciador de acesso" (que pode ser o *proprietário* da aplicação NDN em questão), que criará políticas de controle de acesso, e o produtor de dados que criará chaves de sessão para cifrar/decifrar os dados.

As políticas de controle de acesso criadas pelo gerenciador de acesso basicamente são pares de chave pública e privada, chamadas KEK (chave de cifragem de chave) e KDK (chave de decifragem de chave) respectivamente, criadas de forma granular por espaço de nomes, que permitirão ao produtor de dados distribuir as chaves de sessão para criptografia para múltiplos consumidores [Zhang et al. 2018c]. Assim, o produtor obtém a chave pública KEK do gerenciador de acesso e cada consumir obtém a chave privada KDK do gerenciador de acesso, que deve ser criptografada com a chave pública do consumidor caso o consumir esteja autorizado a acessar aquele espaço de nomes. Em seguida o produtor cria uma chave simétrica de sessão, chamada CK (chave de conteúdo), para a comunicação com os múltiplos consumidores e distribui sob-demanda a chave CK cifrada com a chave KEK, cuja posse da KDK é garantida apenas aos consumidores autorizados, portanto restringindo o acesso à chave CK.

Tais políticas de controle de acesso também podem ser aplicadas para autorização. Neste caso, o gerenciador de acesso, beneficiando-se do esquema de nomeação, adiciona um sufixo ao espaço de nomes do produtor quando da distribuição da chave KEK, por exemplo `time/8am/6pm`, para que a aplicação produtora garanta o cumprimento da política, nesse caso produzindo dados apenas para consumidores autorizados e dentro do horário estabelecido.

Em verdade, o desafio de controle de acesso e confidencialidade entre múltiplas partes também ocorre para outras funções criptográficas, como a própria assinatura digital. Resultados preliminares apontam, inclusive, para a necessidade de revisão do modelo de confiança da NDN para ofertar suporte a tais requisitos [Zhang et al. 2021].

2.4.3. Aplicações de segurança

Construídas com base no arcabouço de segurança da NDN e nos modelos de confiança, chaves e políticas, uma série de aplicações de segurança vem sendo desenvolvidas para adicionar funcionalidades comuns às redes tradicionais e também novas funcionalidades.

Controle de acesso

Regras de controle de acesso são utilizadas para definir autorização ou privilégios de determinadas entidades para acessar determinados conteúdos ou recursos. A NDN provê diversos mecanismos que viabilizam soluções de controle de acesso, destacando-se o esquema de nomeação semanticamente expressivo. Nour et al. 2021a apresentam uma classificação detalhada dos mecanismos de controle de acesso diferenciando aqueles baseados em cifragem do conteúdo e os independente de cifragem. Essa categorização inicial é subdivida em soluções baseadas em atributo, baseadas no esquema de nomeação, baseadas em identidade, baseadas em um intermediador (*broker*), baseadas em controles no *cache*, dentre outras [Nour et al. 2021a].

Zhang et al. 2018c apresentam a modelagem de um esquema de controle de acesso que permite cifragem dos dados e distribuição de chaves. Utilizando a semântica de nomeação especialmente modelada para expressar regras de controle de acesso, os produtores são aptos a controlar acesso aos dados produzidos com base em diferentes critérios (e.g., hora do dia) e apenas os consumidores autorizados recebem chaves de decifragem que os concedem acesso aos dados.

Já Marxer and Tschudin 2017 propõem o uso de ACL para permitir o acesso à coleção de dados (i.e., sub-prefixos do espaço de nomeação) e aos pedaços do conteúdo produzido derivados do conteúdo original. Assim, o consumidor que obtém um conteúdo, em conjunto com as chaves simétricas de decifragem do conteúdo (CK) e a chave pública de decifragem da chave simétrica (KDK), obtém ainda a ACL para controle de acesso daquele pedaço de dados, podendo tornar-se um provedor independente. Em ambos os casos, observa-se uma sobrecarga extra de comunicação para atividades como identificação de consumidores autorizados e estabelecimento de chaves.

Autenticidade e confidencialidade

Soluções de confidencialidade com cifragem baseada em atributos [Lee et al. 2018] estendem as estratégias de criptografia aplicadas aos pacotes de dados a fim de prover proteção sob demanda, entre diferentes domínios administrativos e com possibilidade de agrupamento no processo de autorização. Lee et al. 2018 utilizam o conceito de organização virtual e federação de entidades, sugerindo mecanismos de gestão de membros e confiabilidade entre eles a partir do uso de cifragem baseada atributos. Ramani et al.

2019 usa os atributos como uma alternativa ao modelo padrão de assinatura de pacotes de dados da NDN.

Segundo Ramani et al. 2019, ainda é preciso resolver dois problemas do modelo tradicional: validação dos dados sem demandar requisições adicionais para certificados da cadeia de confiança e anonimidade do produtor. A proposta consiste na introdução de uma entidade Autoridade de Atributos que gera os parâmetros e assinaturas, que são obtidas apenas na inicialização da aplicação e cuja identidade fornece uma visão alto nível da aplicação, mas não do produtor. Uma abordagem híbrida, com múltiplos mecanismos para obtenção da cadeia de confiança, é apresentado em [Gawande et al. 2019]. Os autores Gawande et al. 2019 apresentam a modelagem de uma aplicação para distribuição segura de dados multimídia para uma rede social descentralizada, onde a cadeia de confiança pode ser obtida através de interação presencial dos usuários (que escaneiam um *QR code* contendo os certificados) ou através da troca de mensagens e validação das assinaturas ou através de canais fora da banda.

Esquemas de confiança

Um aspecto chave para a segurança da arquitetura NDN é a distribuição de chaves e o estabelecimento robusto/seguro das cadeias de confiança, especialmente em cenários colaborativos ou com conectividade limitada [Ramani and Afanasyev 2020b]. Diversos trabalhos têm abordado essa questão a partir de diferentes perspectivas: seja propondo modelos de confiança eficientes e robustos para redes veiculares, cuja conectividade dos nós tem curta duração [Ramani and Afanasyev 2020b]; seja através de técnicas de gestão de certificados de curta duração como forma de substituir o processo de revogação quando do comprometimento da chave [Ramani and Afanasyev 2020a]. Alguns trabalhos consideram estratégias de gerenciamento de espaço de nomeação, autoridade de prefixos e serviços de busca/mapeamento de nomes [Tehrani et al. 2019, Afanasyev et al. 2017]

Yu et al. 2015 propõem automação para distribuição de esquemas de confiança, chaves de autenticação de pacotes de dados ou mesmo criação de chaves com escopo de nomeação limitado. Sistemas de reputação [Kapetanidou et al. 2020] constituem uma outra abordagem de esquema de confiança, independente de função criptográfica. Sistemas de reputação são especialmente interessantes em cenários cuja sobrecarga da validação das assinaturas é proibitiva para que seja aplicada em todo pacote de dados, como é o caso dos roteadores NDN. Se, por um lado, os consumidores validam as assinaturas de todos os pacotes de dados, os roteadores acabam sendo alvos de muitos ataques explorando a capacidade de *cache* e reduzidas proteções de segurança. Kapetanidou et al. 2020 apresentam diferentes soluções de confiança baseada em reputação que podem ser aplicadas em roteadores para evitar ataques de negação de serviço.

Autenticação

Gestão de identidade e aplicações de autenticação utilizando o arcabouço de segurança da NDN têm sido o alvo de pesquisas principalmente para ambientes IoT [Li et al. 2019b, Asaf et al. 2020]. Li et al. 2019b propõem um protocolo para autenticação segura de

dispositivos IoT em casas inteligentes, onde assume-se como premissa o uso de um segredo pré-compartilhado entre os dispositivos e a âncora de segurança para, então, derivar o certificado da âncora de segurança – que será utilizado para que o dispositivo autentique outros dispositivos na mesma casa – e seu próprio certificado assinado pela âncora – que será utilizado para que outros dispositivos verifiquem sua identidade.

Também são propostas estratégias de autenticação mais robustas para dispositivos com recursos adicionais, a saber: geração do par de chaves no próprio dispositivo, quando da existência de bom grau de entropia no dispositivo; reuso de chaves no processo de re-autenticação, quando da existência de mídias de armazenamento seguras; caso o dispositivo possua interfaces interativas como teclas e monitor, é possível dinamicamente estabelecer o segredo com a âncora para troca dos certificados – de forma similar ao processo de empareamento de dispositivos *Bluetooth* e os códigos de autorização [Li et al. 2019b].

Asaf et al. 2020 apresentam um levantamento sobre o uso de *blockchain* em NDN, abrangendo desde a descrição em alto nível dos principais componentes/camadas de um arcabouço *blockchain* e sua integração com NDN até um levantamento de trabalhos aplicando a tecnologia de *blockchain* em contextos específicos de aplicações NDN. Exemplos de contextos apresentados incluem: segurança e privacidade – proteção do conteúdo, gerenciamento de chaves, segurança no *cache*, privacidade; suporte a funções específicas de rede em ambientes 5G, redes de sensores sem fio, redes veiculares e *ad-hoc*; e outras aplicações em geral – mobilidade, aplicações na área de saúde e concessão de recursos.

2.4.4. Superfície de ataques

A mudança no modelo de comunicação trazida com a arquitetura NDN inevitavelmente implica em novos desafios de segurança e em desafios para tratar problemas conhecidos [Mannes and Maziero 2019]. Mecanismos seguros, eficazes e flexíveis de checagem de integridade e autenticidade dos conteúdos nomeados são necessários, permitindo ao usuário validar se o conteúdo foi alterado e se a origem é legítima.

O plano de dados com manutenção de estado, *cache* oportunístico e diferentes estratégias de encaminhamento demanda mais recursos e está suscetível a diferentes tipos de ataques, incluindo ataques de negação de serviço e ataques de poluição de *cache*. Além disso, devem ser considerados os desafios envolvendo a privacidade a partir de diferentes perspectivas, englobando desde o esquema de nomeação até o armazenamento de informações sensíveis em *caches* intermediários. Como reflexo desses desafios, alguns trabalhos recentes propõem avaliações das superfícies de ataques em diferentes contextos [Mannes and Maziero 2019, Nour et al. 2021b].

Mannes and Maziero 2019 apresentam uma classificação de ameaças de segurança e entidades maliciosas envolvidas em ataques, como mostrado na Figura 2.11. Os ataques são classificados/analisados em três categorias: segurança do conteúdo, do roteamento e do *cache*. Essas categorias são subdivididas de acordo com o tipo de ataque e a entidade maliciosa que executa a exploração. Exemplos de ataques direcionados ao conteúdo incluem: fabricação, renomeação e modificação de conteúdo, monitoramento de conteúdos por nome, análise de conteúdo não cifrado, privacidade do nome, identificação do produtor através do localizador de chave, acesso não autorizado e personificação de produtor.

Ataques ao roteamento incluem: exaustão de recursos através de inundação, ataques à PIT [Seixas et al. 2017], sobrecarga no produtor, exaustão de banda, sequestro de prefixo (do inglês, *prefix hijacking*) e interceptação de tráfego. Por fim, são exemplos de ataques ao sistema de *cache*: *cache snooping*, ataques de temporização, monitoramento de requisições, poluição de *cache*, negação de serviço à CS e injeção de conteúdo ilegítimo. Mannes and Maziero 2019 provêm ainda uma relação de trabalhos que apresentam contra-medidas aos ataques apresentados e um detalhamento sobre o modelo de ataque para cada vulnerabilidade.

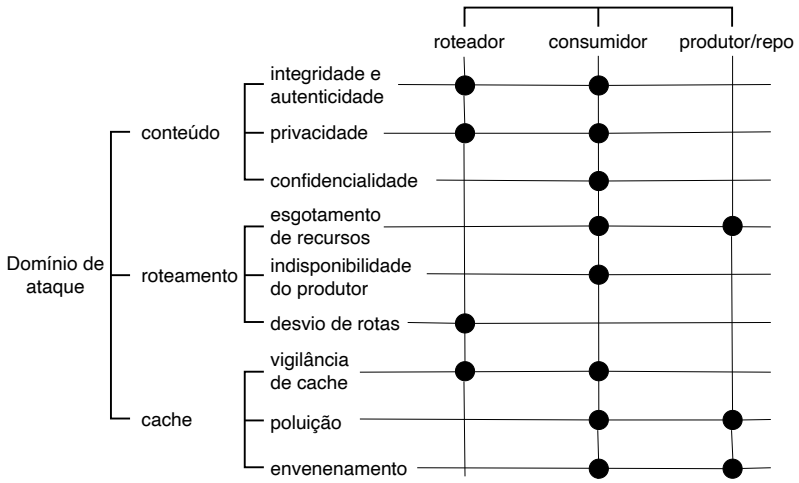


Figura 2.11. Classificação de ameaças e entidades maliciosas. Traduzido de [Mannes and Maziero 2019].

De maneira complementar, Nour et al. 2021b apresentam um levantamento dos principais ataques em redes ICN/NDN, com foco principal em ambientes de rede sem fio. A taxonomia de ataques leva em consideração os componentes da ICN: nome do conteúdo, *cache* e conteúdo em si. Três ataques ao nome do conteúdo são listados: inundação de interesses, ataques de monitoramento e ataques de lista de supervisão (onde o encaminhador NDN pode filtrar/bloquear uma lista pré-definida de nomes de conteúdo). Para ataques ao componente da *cache*, são citados apenas poluição de *cache* e envenenamento de *cache*. Por fim, ataques ao conteúdo incluem: acesso não autorizado e ataques ao encaminhador NDN (do inglês, *Mistreating Attack*). Os ataques aos encaminhadores intermediários, que consistem em um encaminhador malicioso filtrando/bloqueando ou modificando o conteúdo, são bastante danosos ao ambiente sem fio, principalmente em redes *ad-hoc*, nas quais os nós encaminham tráfego em favor de outros nós [Nour et al. 2021b].

Por fim, é importante destacar que o desenvolvimento de uma nova arquitetura implica em novas maneiras de desenvolver aplicações, novos arcabouços de *software*, novas bibliotecas e até novos modelos econômicos associados à distribuição de conteúdo. Todos esses aspectos podem implicar em vulnerabilidades e ataques às aplicações NDN, que embora não sejam diretamente associadas à arquitetura, possuem forte relação e eventualmente podem afetar componentes específicos.

2.5. Aplicações Distribuídas

Nesta seção, serão apresentadas as propriedades da NDN sob a ótica dos modelos de computação distribuída. Além disso, serão abordados os aspectos de comunicação entre processos e a utilização de protocolos para a sincronização de estado distribuído em NDN. Por fim, será apresentado um estudo de caso baseado no uso de vetores de estado em NDN.

2.5.1. Modelos de sistemas distribuídos

A arquitetura NDN possibilita o desenvolvimento de aplicações distribuídas em processos de nós distintos da rede, em que a comunicação entre quaisquer dois processos é mediada por trocas de mensagens em um canal de comunicação. Desta forma, a NDN atua como um sistema distribuído que media a informação dos produtores de dados aos consumidores de dados. Na literatura, modelos de sistemas distribuídos caracterizam um conjunto de propriedades que definem o comportamento de processos e de canais de comunicação.

No projeto de algoritmos distribuídos, o modelo de interação apresenta as premissas do ambiente quanto aos limites temporais de execução pelos processos e de troca de mensagens. A existência de limites bem conhecidos permite, por exemplo, pontos de sincronização entre os processos, que auxiliam na determinação do traço global da execução do algoritmo distribuído. A existência destes limites temporais nos ambientes reais pode ser possível pela reserva de recursos de processamento e comunicação e de um comportamento determinístico, o que é de difícil implementação (e.g., por meio do uso de sistemas operacionais de tempo-real, de redes determinísticas, como CAN e *Token-Ring*, ou por meio de QoS e reserva de recursos). Caracterizamos este conjunto de premissas no modelo síncrono. Lynch 1996 indica que este modelo permite realizar passos de forma sincronizada, isto é, a execução ocorre em rodadas síncronas. Em geral, a existência de tais limites temporais em conjunto com a sincronização periódica dos relógios locais associados aos nós dos processos, estabelece a referência necessária de *slots* de tempo e rótulos temporais para a coordenação das ações distribuídas em rodadas síncronas.

Uma outra possibilidade é assumir a premissa de que os componentes do sistema distribuído (processos e canais de comunicação) executam passos de forma arbitrária, sem limites temporais conhecidos. O modelo assíncrono não permite que os algoritmos assumam quaisquer hipóteses baseadas no tempo de execução dos passos computacionais ou do tempo gasto para troca de mensagens. Assim, ao se projetar um algoritmo para o modelo assíncrono, este não deve confiar em premissas temporais para operação, sendo mais genéricos e portáveis que no síncrono. É importante ressaltar que a NDN, em linhas gerais, não se baseia em premissas temporais.

Contudo, esta mesma ausência de limites temporais não nos permite caracterizar a falha de componentes de forma segura, e assim, algoritmos que requerem coordenação de ações, como o consenso distribuído, não têm garantia de execução em ambientes assíncronos na presença de falhas [Fisher et al. 1985]. Desta forma, obter algumas garantias de execução e comunicação associadas a limites temporais pode ser interessante para a garantia da terminação dos algoritmos. Uma hipótese usual em ambientes de execução real é que um ambiente não síncrono pode se tornar estável com um comportamento "síncrono", a partir de algum momento no tempo, denominado GST (*Global Estabilization Time*) [Dolev et al. 1987]. Esta hipótese de estabilidade pode viabilizar a execução de

algoritmos distribuídos em uma rede NDN.

Para que a aplicação distribuída se torne resiliente, é importante entender os modelos de falhas – i.e., como os processos e canais podem falhar –, o que possibilita o desenvolvimento de mecanismos de tolerância a falhas. Existem diferentes tipos de falhas que podem ocasionar problemas em diversos contextos da NDN. Falhas arbitrárias ou bizantinas estão relacionadas à execução deliberada fora da especificação do algoritmo, que pode ocorrer, por exemplo, por comprometimento do processo ou do nó de execução, ou por mensagens serem corrompidas ou adulteradas no canal de comunicação [Lamport and Fischer 1982]. Na Seção 2.4 são expostos alguns cenários na NDN caracterizados por tal comportamento malicioso. Canais de comunicação também podem falhar por omissão, com perda de mensagens. Processos podem parar a execução em falha por *crash*. Caso haja o retorno deste processo em um ponto de execução anterior ao do *crash*, temos o *crash-recovery*. A infraestrutura de comunicação da NDN é, de certa forma, resiliente à perda de mensagens, embora seja necessário considerar cenários de *crash* de processos, ou de conexão intermitente dos mesmos.

2.5.2. Propriedades da NDN para computação distribuída

A separação do identificador e localizador do conteúdo e o modelo de comunicação baseado em nomes impõem mudanças fundamentais na forma de pensar aplicações em NDN. O espaço de nomes (do inglês, *namespace*) compartilhado proposto pela arquitetura permite que o encaminhamento de pacotes não se altere com mudanças de topologia ou movimentação e conectividade de nós, mas, tão somente no alcance do *namespace*.

Conforme já discutido anteriormente, os nós da NDN podem ser produtores ou consumidores de dados, ou intermediar esta relação ao encaminhar requisições. O modelo de comunicação centrado em dados utilizado permite desacoplar as requisições de referências temporais. O modelo adotado não requer nem mesmo que produtor e consumidor estejam ativos ao mesmo tempo. Pode-se, portanto, assumir assim que a NDN permite um mecanismo de comunicação completamente assíncrono. Ou seja, não é necessário limites temporais conhecidos para o processamento e comunicação ao longo de toda a execução. Contudo, a terminação adequada de aplicações distribuídas na NDN requer que haja, em algum momento, a estabilidade nos limites temporais da rede, assumindo-se as premissas de modelos parcialmente síncronos, como os da hipótese GST.

Deve-se notar que, apesar desta possibilidade de desacoplamento temporal, é possível incorporar referências temporais ao associar um determinado interesse a um *lifetime*. Assim, há a possibilidade de premissas fortes para limites temporais, exercitando aplicações síncronas na NDN. Mastorakis et al. 2018 propõem o protocolo *Realtime Data Retrieval* (RDR), que minimiza a latência de obtenção da informação mais atual desde que o produtor (ou um nó delegado por este) e o consumidor estejam ativos ao mesmo tempo. Neste contexto, o *lifetime* de uma entrada da PIT é um requisito temporal ajustado conforme a percepção do ambiente. Embora o protocolo não necessite de sincronização de relógios ou premissas mais robustas (como uma infraestrutura de enlace determinística), na presença destas, há um ambiente síncrono que favorece aplicações de tempo real estrito.

No que reflete ao modelo de falhas, pode-se assumir que abordagens apresentadas na Seção 2.4 são capazes de mitigar ataques maliciosos relacionados aos canais de comu-

nicação. Ademais, o uso de *in-network caching* permite a entrega confiável, em caso de não haver falhas do produtor e do consumidor (i.e., canais de comunicação confiáveis – sem falhas). A assincronicidade de operação entre produtor e consumidor pode até mesmo permitir a entrega caso o dado já tenha sido encaminhado para CS em nós intermediários e, assim, conseguir alcançar o consumidor.

A NDN pode ser implantada em cenários de mobilidade com alta taxa de volatilidade (*churn*) de nós, ou seja, há ingresso e saída (ou desconexão) de nós produtores e consumidores ao longo da execução. Este cenário é similar ao da comunicação assíncrona assumida por Paxos [Lamport et al. 2001], no qual processos podem ingressar e sair a qualquer tempo, refletindo o modelo de falhas de processos *crash/recovery*. Isso significa que uma desconexão eventual pode ser representada pelo *crash* ou inacessibilidade de um nó, que pode se recuperar e ficar novamente disponível.

2.5.3. Aspectos de comunicação entre processos na NDN

Aplicações distribuídas utilizam-se da troca de mensagens para coordenar ações. Em redes TCP/IP, este mecanismo de comunicação entre processos é exercitado por meio de *sockets*. Outros padrões arquiteturais podem ser utilizados por meio de camadas de *middleware*, provendo outras semânticas para interação entre processos. Os mecanismos definidos na arquitetura NDN reduzem o *gap* semântico rede-aplicação e auxiliam na implementação de alguns padrões, como o uso de funções nomeadas e do paradigma *publish/subscribe*.

Invocação remota como suporte à computação distribuída

A invocação remota de métodos/funções é um mecanismo de comunicação entre processos que permite a execução dessas tarefas como se fossem locais, burlando aspectos de localização e provendo outra semântica para o programador da aplicação cliente [Waldo et al. 1996]. As arquiteturas baseadas no modelo ICN têm algumas limitações para lidar com conteúdo dinâmico. Consequentemente, existem desafios para lidar com suporte à execução de funções de rede. No entanto, algumas iniciativas, como *Named Function as a Service* (NFaaS) [Król and Psaras 2017], têm endereçado esse problema.

A NFaaS permite mover a computação para a borda da rede. Esta abordagem permite a execução de microsserviços e funções *stateless* baseada em um novo componente da NDN, o *Kernel Store*, que armazena as funções a serem executadas. Novas primitivas são incorporadas permitindo a migração de funções de um nó da NDN para outro de forma dinâmica, ou mesmo o balanceamento de carga com o encaminhamento de parte das requisições de um nó com sobrecarga a outros nós na NDN. Essa abordagem é estendida no RICE [Król et al. 2018], que possui primitivas para desacoplar a invocação do método do retorno dos resultados permitindo computação de longa duração, além de permitir autenticação prévia de clientes e o suporte a conjuntos de parâmetros complexos.

Publish/subscribe como suporte à computação distribuída

Publish/subscribe é um padrão arquitetural que possibilita que entidades publicadoras (do inglês, *publishers*) publiquem conteúdos associados a um ou mais tópicos, e entidades

assinantes (do inglês, *subscribers*) recebam notificação somente dos tópicos que possuam interesse. Este esquema de comunicação assíncrona é baseado em eventos e desacoplado em tempo, espaço e sincronização entre as partes, de forma que permite a implementação de sistemas distribuídos de larga escala [Eugster et al. 2003]. Nour et al. 2019c exploram diferentes possibilidades de construção semântica de sistemas *publish/subscribe* em redes ICN, por exemplo: (i) única requisição-única resposta – um nó assinante requer conteúdo já publicado e obtém uma resposta, qualquer atualização deve ser obtida pela requisição de um novo pedido de conteúdo; (ii) única requisição-várias respostas – um nó assinante requer conteúdo, mas pode receber diversas respostas ao longo do tempo, de acordo com a semântica da aplicação; (iii) entrega periódica – um nó assinante requer conteúdo periódico a cada intervalo de tempo, e.g., um sensor enviando atualizações a cada 10 minutos; (iv) *N* respostas – um nó assinante requer um número específico de respostas sobre um conteúdo (e.g., os próximos 10 quadros de um vídeo); e (v) entrega condicional – um nó assinante requer receber conteúdo somente se este atender a certos requisitos condicionais.

Na NDN o roteamento e encaminhamento de mensagens é baseado em nome. O conteúdo publicado pode ser obtido individualmente pelos assinantes por meio da troca de mensagens de interesse/dados da NDN. Ou seja, a NDN é nativamente um sistema de única requisição-única resposta. Uma aplicação na NDN pode adaptar a semântica de *publish/subscribe* utilizada a partir do envio de novos pacotes de interesse. Por exemplo, em uma rede de sensores, caracterizada por uma necessidade de comunicação com entrega periódica, requisita-se a atualização da informação em cada período de monitoramento. Esse comportamento pode gerar congestionamentos, assincronia e problemas com a agregação (duas ou mais respostas serem respondidas da mesma forma pela agregação dos interesses em nós intermediários).

Cenários em que há um grupo de *N* publicadores e *M* assinantes para o mesmo conteúdo/tópico do *publish/subscribe* são desafiadores em NDN. Nour et al. 2019c tratam o modelo *publish/subscribe* por meio da semântica de comunicação em grupo. Um tópico é representado em um grupo, ao qual assinantes podem se unir (do inglês, *join*) ou sair (do inglês, *leave*), o produtor mantém a visão do grupo e pode também prover a retirada de membros de acordo com a semântica. Nesse tipo de cenário, é importante estabelecer mecanismos para sincronização do estado distribuído entre multipartes.

2.5.4. NDN Sync: sincronização de estado distribuído na NDN

O modelo *publish/subscribe* implementado pela NDN favorece a interação entre um consumidor e um produtor. Contudo, um compartilhamento de conjunto de dados que possibilite múltiplas partes é desejável para o uso em aplicações distribuídas. Neste contexto, NDN Sync se apresenta como uma abstração para comunicação multiparte agnóstica de conexão na NDN. O compartilhamento de conjunto de dados deve observar que esta comunicação multiparte é assíncrona, ou seja, nem todas as partes podem estar conectadas ao mesmo tempo. Desta forma, o NDN Sync é um mecanismo de sincronização do espaço de nomes entre um grupo de entidades.

Entidades organizadas a partir de um *namespace* formam um grupo que permite que novos dados possam ser consumidos por todos os membros deste *namespace*. A abstração criada estende a comunicação único-produtor/único-consumidor da NDN para um modelo

de multi-produtor/multi-consumidor: qualquer produtor que distribua informação no *namespace* terá a informação sincronizada entre os consumidores do *namespace*.

O NDN Sync provê um mecanismo de notificação que permite que todos os processos interessados em um determinado *namespace* possa receber notificações quando novos dados são produzidos e obtê-los no melhor momento (se isto for de seu interesse). Isto implica em um mecanismo que possibilita detecção de mudanças no estado dos dados. Ou seja, cada participante interessado no *namespace* mantém sua cópia local do estado de dados e, por meio do protocolo NDN Sync, obtém notificações e pode se manter atualizado. A estratégia de detecção de mudanças no estado de dados considera a forma como o estado de dados é representado internamente (estrutura de dados) e a estratégia de nomeação dos dados. Por exemplo, o uso de informação sequencial permite representar mudanças e o de nomes arbitrários pode ter uma semântica associada à estratégia de sincronização.

Na Figura 2.12 é possível observar o funcionamento básico do NDN Sync: após publicar novos dados, um participante notifica as demais partes através do anúncio de dados (Passo 1), quer seja por envio de mensagem de *interesse Sync* quer seja por resposta a mensagens anteriores desses interesses. De acordo com o protocolo, esta mensagem já pode informar a alteração, ou tão somente ser uma notificação. No segundo caso, um segundo passo (Passo 2) é necessário com a troca de mensagens de interesse e dados. Nós podem sinalizar interesse de longo termo em mudanças: um *interesse Sync* de longo termo é mantido pendente em todos os nós encaminhadores por um longo tempo. Ainda, é possível utilizar mensagens periódicas de monitoramento (*heartbeats*) para detectar mudanças no *namespace* ou no conjunto de partes interessadas (*membership* do grupo).

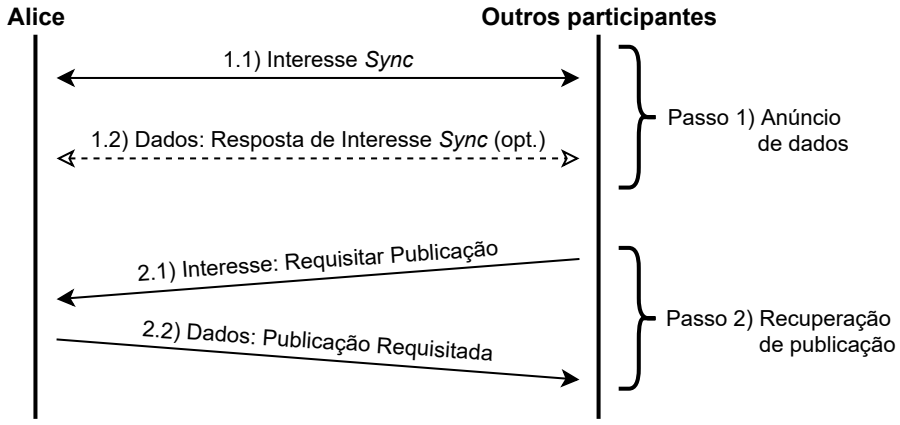


Figura 2.12. Funcionamento básico de NDN Sync. Adaptado de [Shang et al. 2017].

Diferentes protocolos de sincronização foram propostos desde 2012 [Shang et al. 2017]. Protocolos como CCNx 0.8 Sync, ChronoSync e RoundSync utilizam árvores, como *Merkle trees*, para representar o estado de dados, mantendo *hashes* ou *digests* que são utilizados para detectar mudanças e atualizar o estado. Já o CCNx 1.0 Sync utiliza manifestos para anunciar uma atualização. O manifesto é um conjunto de metadados da aplicação que são associados aos objetos mantidos na coleção local. O *hash* do manifesto é difundido através do envio de *interesse Sync* e, por meio desta difusão, os nomes associados aos manifestos são comparados e os dados de interesse são solicitados à rede. Alguns

protocolos como o iSync, syncps e PSync fazem uso do filtro de Bloom invertido (IBF) para identificar dados faltantes. O filtro de Bloom permite verificar, de forma probabilística, se elementos pertencem a um conjunto. O filtro de Bloom invertido, por sua vez, permite um conjunto de operações, inclusive comparar diferenças entre IBFs. A partir dessa comparação, é possível identificar os dados faltantes e solicitar as informações necessárias para sincronização de estado.

Em contextos como o da NDN, em que a forma assíncrona de troca de dados não utiliza informações temporais, como relógios sincronizados e rótulos de tempo (do inglês, *timestamps*), eventos de atualização de estado podem ser correlacionados de forma lógica. Lamport 2019 sugere o uso de uma notação baseada na contagem de eventos locais que utiliza um vetor denominado relógio vetorial para prover uma ordem parcial de eventos que ocorrem em um sistema distribuído. Este vetor de estados do sistema pode ser utilizado em aplicações que determinem o estado global e executem ações necessárias de coordenação distribuída [Chandy and Lamport 1985]. A partir do *Vector Sync*, propostas de protocolos baseados neste conceito foram desenvolvidas, como *State Vector Sync*, PLI-Sync e ICT-Sync [Shang et al. 2017].

2.5.5. Do NDN Sync à computação distribuída: estudo de caso

O uso de vetor de estados, inicialmente proposto no *Vector Sync*, implica em manter o contador de eventos associado a cada produtor de conteúdo. Na produção de conteúdo, há o anúncio (*interesse de notificação*): caso outra parte verifique que o vetor difundido tem componente maior que o vetor local, procede-se a atualização e os dados podem ser obtidos pela troca de mensagens de interesse e dados necessárias (ver Figura 2.13).

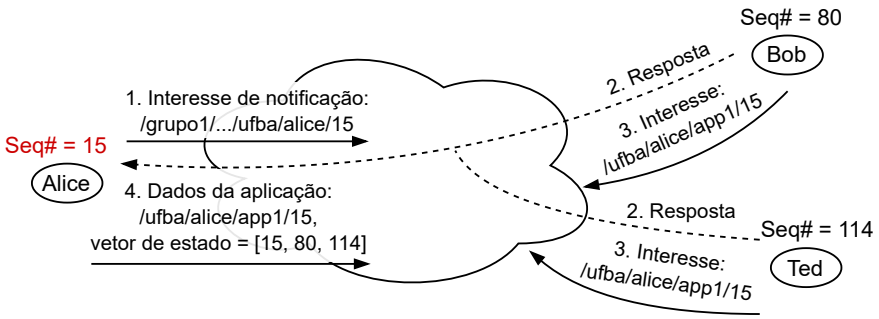


Figura 2.13. Sincronização de estado no *Vector Sync*. Adaptado de [Shang et al. 2017].

O projeto desta solução para o *Vector Sync* implica em manter qual o grupo de produtores de conteúdo ativos, o que remete a primitivas de um protocolo de comunicação em grupo clássico. O *membership* do grupo de produtores ativos associados ao *dataset* é mantido com *Heartbeats interest*. Cada participante deve enviar interesses periódicos, denominados *heartbeats*, indicando seu prefixo e vetor de estados para o grupo. Um processo escolhido como líder coleta estas mensagens e difunde a lista de membros ativos do grupo. Desta forma, é possível verificar quando há solicitações de ingresso ou de saída do grupo, ou suspeitas de falhas (participante inativo), difundindo-se a nova informação.

A abordagem baseada em um líder assume que este é o participante com maior número de ordem no prefixo. Na ausência de manifestação do líder em resposta aos

heartbeats, um novo líder pode ser escolhido, aplicando o algoritmo do *Bully* [Stoller 1997]: ou seja, o processo ativo de maior número de ordem no prefixo assumirá este papel. O uso de *heartbeats* para criar a nova visão de grupo periodicamente é uma abordagem clássica em comunicação em grupo [Cristian and Schmuck 1995]. Este padrão de comunicação estabelece o bloco de detector de defeitos, que auxilia na coordenação da computação distribuída em suspeita de falhas e cuja efetividade depende das premissas temporais do ambiente [Chandra and Toueg 1996].

A combinação da abordagem de gerenciamento de *membership* e da semântica de relógios vetoriais para a sincronização do estado do conjunto de dados permite ao *Vector Sync* manter aplicações distribuídas baseadas em múltiplos produtores e múltiplos consumidores, sem o uso de interesses de longo termo. Deve-se ressaltar que o padrão de comunicação da NDN permite seu uso em ambientes com mobilidade, comumente caracterizados por alta rotatividade de nós. Estes cenários são desafiadores para protocolos de comunicação em grupo, pois implicam em constante mudanças de visão.

A sincronia de visões geralmente implica em alta carga, em consequência a um número considerável de saídas acidentais ou exclusões por falsas suspeitas (devido a desconexões momentâneas). O *State Vector Sync* (SVS) evolui o *Vector Sync* permitindo a partição da rede, em que a abordagem de comunicação em grupo convencional mediada por líder é substituída, utilizando-se um vetor de estado dinâmico, que mantém informação atual sobre produtores conhecidos, sem necessidade de um *membership* formal. Assim, o SVS suporta maior nível de *churn*, adaptando-se ao padrão de comunicação assíncrona inerente à NDN e suportando particionamento da rede. Ou seja, participantes podem publicar e propagar novos dados a qualquer momento sem um ingresso formal, mantendo-se a disponibilidade [Shang et al. 2017].

A opção por maior resiliência em um contexto de conectividade intermitente é incrementada no PLI-Sync em que um *prefetch* oportunístico é exercitado: participantes obrigatoriamente atualizam todos os dados disponíveis e sinalizam interesse em sequências ainda a serem produzidas. Os resultados preliminares indicam que esta abordagem favorece cenários com alta perda de dados. Já o ICT-Sync propõe alterações ao *Vector Sync* com o uso de nós intermediários que copiam dados de produtores e os mantém disponíveis ao grupo mesmo quando os produtores originais estão inacessíveis. O uso de lista de mapeamento desacopla os componentes do vetor de estados dos participantes e permite uma abordagem dinâmica, dispensando o *membership* baseado em líder do *Vector Sync*.

Conforme o Teorema CAP [Simon 2000]: (C) consistência, (A) disponibilidade e (P) tolerância, a partição para um conjunto de dados compartilhado, podemos verificar que a resiliência à partições e a disponibilidade de dados destas abordagens são obtidas a cargo de consistência fraca. Desta forma, o NDN Sync se apresenta adequado a um amplo conjunto de aplicações distribuídas em que os requisitos condizem com as premissas de possibilitar disponibilidade de dados com um modelo de comunicação assíncrono, que seja resiliente a particionamento e alto *churn*.

2.6. Ambientes de Experimentação

Nesta seção, serão apresentados os ambientes de experimentação em NDN e serão descritas as informações gerais para a atividade prática do capítulo.

2.6.1. Simulação com o ndnSIM

O simulador de redes baseado em eventos discretos ndnSIM[3] [Mastorakis et al. 2017] foi desenvolvido como um módulo específico para experimentação de NDN no simulador de redes ns-3[4]. Criado em 2012, o ndnSIM se consolidou como uma plataforma largamente utilizada na comunidade de pesquisa em NDN para rápida prototipagem de aplicações, avaliações de desempenho com alta escalabilidade e vasta gama de cenários, topologias, fatores e níveis de experimentação.

O ndnSIM é apto à execução de experimentos de larga escala, com milhares de nós executando a pilha NDN em *hardware* comum [Mastorakis et al. 2017]. Em termos de diversidade de fatores e níveis para análise de desempenho, como o ndnSIM é construído no topo do ns-3, é possível testar configurações de topologias e parâmetros (e.g., largura de banda e atraso do enlace), simular diferentes modelos/protocolos da camada de enlace e modelos de propagação e mobilidade, além de permitir uma customização granular nos parâmetros/estratégias da pilha NDN. Assim, é possível escolher desde a estratégia de encaminhamento (e.g., *best route*, *multicast* e ASF) por prefixo, até o tamanho da *cache* e algoritmo de substituição de dados, incluindo ainda detalhes como política para tratamento de dados não solicitados e parâmetros especializados das *faces* NDN.

Uma das desvantagens do ndnSIM é a necessidade de adaptação no código para execução em ambiente real. Além disso, muitos estudantes iniciando a pesquisa em NDN reportam uma curva de aprendizagem menos acentuada, que vem sendo melhorada com a disponibilização de muitos exemplos pelo projeto.

2.6.2. Emulação de aplicações no Mini-NDN

Emuladores de rede fornecem uma plataforma de experimentação complementar aos simuladores: embora menos escaláveis e com menor conjunto de cenários de experimentação, o sistema de emulação tipicamente fornece características próximas ao ambiente real e permite utilizar um protótipo de aplicação similar ao que será utilizado no ambiente real.

O Mini-NDN[5] é um emulador leve de redes NDN que apoia o desenvolvimento, teste e avaliação de desempenho de aplicações NDN, em um ambiente simples ou em um *cluster* de nós. O Mini-NDN foi originalmente baseado em um projeto chamado Mini-CCNx[6], que por sua vez era um *fork* do Mininet[7], e atualmente oferece suporte às bibliotecas NDN, sistema de encaminhamento NFD, protocolos de roteamento NLSR e NDVR, além de um conjunto de ferramentas do projeto NDN, como o *ndnping* (teste de conectividade) e o *ndn-chunks* (para transferência de dados).

O Mini-NDN possui integração com o Mininet-WiFi, que permite a emulação de estações e pontos de acesso Wi-Fi (modo infraestruturado e *ad-hoc*) baseados no módulo sem fio do kernel Linux *80211_hwsim*. O Mininet-WiFi, integrado ao Mini-NDN, permite executar aplicações NDN no ambiente de rede sem fio virtual e com suporte a diferentes modelos de propagação de sinal (e.g., *Friis*, *Log Distance*, *Log Normal Shadowing*, etc.)

[3] https://ndnsim.net/current/
[4] https://www.nsnam.org/
[5] http://minindn.memphis.edu/
[6] https://github.com/chesteve/mn-ccnx
[7] https://github.com/mininet/mininet

e diferentes modelos de mobilidade (e.g., *Random Walk*, Gauss Markov, Caminhada de Levy, RPGM, etc.), além de oferecer integração com o SUMO[8] para cenários de VANET.

A principal desvantagem do Mini-NDN é a escalabilidade, sendo diretamente dependente do desempenho do *hardware* em que está sendo executado, com relatos de limite superior em torno de algumas centenas de nós NDN em uma execução. O projeto provê suporte ao modo de execução em *cluster* do Mininet, que reduz essa limitação.

2.6.3. *Testbed* NDN

O projeto NDN também conta com um *testbed*[9] internacional envolvendo 36 nós em diferentes países. Trata-se de uma rede utilizada para experimentação e validação de protocolos e aplicações em um ambiente com equipamentos reais. Atualmente, um ponto de presença do *testbed* encontra-se disponível na UFBA[10], como mostra a Figura 2.14.

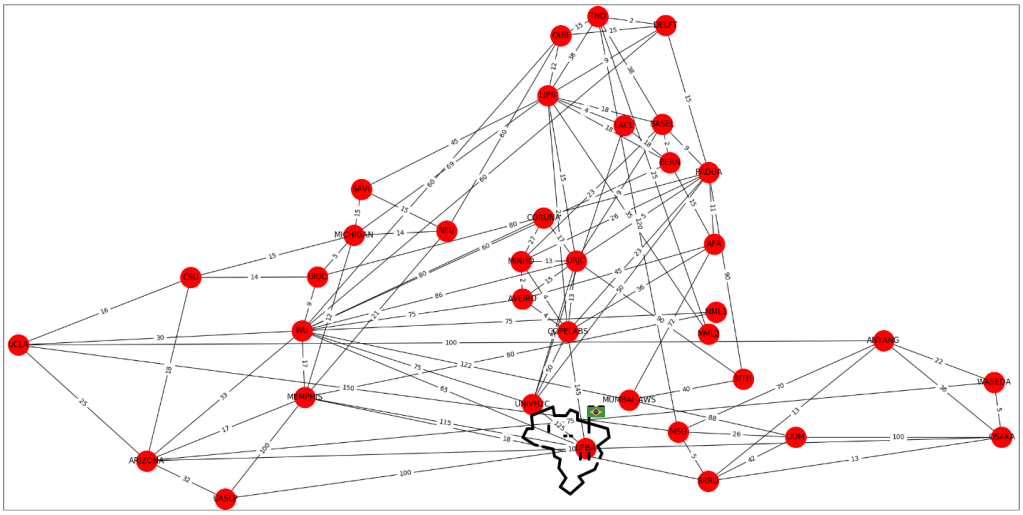


Figura 2.14. *Testbed* NDN (36 nós interconectados, 97 *links*, protocolo NLSR). Adaptado de [NDN 2021].

2.6.4. Bibliotecas, APIs e outros ambientes de experimentação NDN

Além das ferramentas de simulação, emulação e experimentação (*testbed*), o projeto NDN recentemente apresentou um conjunto de APIs e bibliotecas de suporte para aplicações NDN. Essas ferramentas abstraem os detalhes de requisições individuais de pacotes de interesse e dados, validações e tratamento de NACKs. Duas dessas bibliotecas são a NDN-Lite [Yu et al. 2021] e a NDN-CNL [Thompson et al. 2019]. Elas permitem o desenvolvimento de aplicações que fazem uso de um espaço de nomes disponível (ou automaticamente descoberto), se registram em uma ramificação do espaço de nomes e publicam dados. Isso possibilita criar uma abstração alto nível que simplifica as funções do modelo *publish/subscribe* e facilita a experimentação em cenários IoT e outros ambientes.

Outro contexto de execução importante de mencionar são os ambientes de alta capacidade de comutação de pacotes, que podem beneficiar a comunidade de *big data* e

[8] https://www.eclipse.org/sumo/
[9] https://named-data.net/ndn-testbed/
[10] https://ufba.testbed.named-data.net/

ciência de dados. Shi et al. 2020 foram pioneiros em executar o plano de encaminhamento NDN em *hardware* não especializado atingindo taxas de comutação superiores a 100 Gbps. Tais resultados foram possíveis graças a uma combinação de técnicas de aceleração do *pipeline* de processamento de pacotes através do arcabouço DPDK, além de otimizações nos algoritmos e estruturas de dados do *pipeline* de encaminhamento da NDN.

2.6.5. *HandsOn*: demonstração e prática de NDN

Nesta seção, será apresentada uma atividade prática cujo objetivo é demonstrar o funcionamento da arquitetura NDN, seus principais componentes, ambientes de execução e o desenvolvimento de uma aplicação simples utilizando as APIs de comunicação em rede.

O desenvolvimento de uma aplicação NDN requer a definição de um esquema de nomeação da aplicação, definição do modelo de confiança e o uso de uma série de funções das APIs das bibliotecas ndn-cxx, NDN-Lite ou NDN-CNL. Para simplificar a atividade prática, serão fornecidos o design de um esquema de nomeação da aplicação, modelo de confiança e um esqueleto base do código fonte. Além disso, o desenvolvimento pode ser direcionado a um dos componentes da arquitetura, tais como: desenvolvimento de uma política customizada de substituição de *cache*, desenvolvimento de uma nova estratégia de encaminhamento, desenvolvimento de um protocolo de roteamento, dentre outros. Considerando que o objetivo desta atividade é permitir que o leitor coloque em prática os conceitos apresentados nas seções anteriores, ao invés de apresentar uma exploração profunda dos componentes e APIs, ao longo da execução da prática os pontos chaves da arquitetura serão mencionados para que possam ser alvo de investigação futura.

Toda a documentação sobre o design da aplicação, todos os códigos-fonte, bibliotecas e programas auxiliares utilizados no escopo desta seção estão disponíveis em um repositório do projeto[11]. O repositório deve ser clonado para facilitar a execução da atividade prática, em seguida deve-se seguir as instruções de execução que estão disponíveis no próprio repositório. Para tal, é necessário clonar o respositório do projeto: `git clone https://github.com/insert-lab/mc-ndn-sbrc2021` e seguir as instruções contidas no arquivo `README.rst`.

2.7. Desafios de Pesquisa

Nesta seção serão apresentadas as questões de pesquisa relacionadas à mobilidade, segurança e aplicações distribuídas em NDN, incluindo roteamento e comunicação *multi-hop*, às limitações dos protocolos da camada de enlace, mecanismos de incentivo e reputação, e mecanismos de difusão, sincronização de estado e *handoff*.

2.7.1. Mobilidade

Apesar do meio sem fio favorecer a comunicação por difusão (*broadcast*), uma grande parte das implementações de redes se baseia no uso de endereçamento da camada de enlace para direcionar os quadros às interfaces de destino. No entanto, a NDN é caracterizada pela comunicação *multicast*, em que um nó toma as decisões de encaminhamento a partir das informações de estado, alcançabilidade de prefixos e conteúdos das *caches* locais.

[11] `https://github.com/insert-lab/mc-ndn-sbrc2021`

Encaminhamento NDN na camada de enlace

Em redes cabeadas, o encaminhamento da NDN é efetivo por ser realizado para um subconjunto de diferentes interfaces de saída. Contudo, em redes móveis, formadas com nós equipados por apenas uma única interface (*wlan0*), o encaminhamento resulta na inundação de pacotes na rede. Na arquitetura NDN não existe um mapeamento direto entre o nome e um endereço MAC [Kietzmann et al. 2017]. Desta forma, pacotes transmitidos por difusão causam o processamento desnecessário de quadros pelos nós da rede. Consequentemente, em cenários de mobilidade, em que os nós comunicantes possuem um única interface de rede, as informações de roteamento não são suficientes para apoiar as decisões de encaminhamento, dado que a existência de apenas uma interface resulta naturalmente numa inevitável inundação de pacotes.

A fim de mitigar a inundação da rede em cenários de mobilidade, um grande número de trabalhos tem focado na concepção de novas estratégias de encaminhamento. Em geral, os trabalhos levam em consideração métricas como localização, contexto, distância entre nós, estabilidade de enlace, quantidade de saltos, vizinhança, coordenadas geográficas e energia [Tariq et al. 2020, Wang et al. 2020]. Contudo, tais soluções são efetivas para contextos específicos e não são nativas da arquitetura.

Diferente das estratégias de encaminhamento, algumas iniciativas promovem a implementação de técnicas de autoaprendizagem de endereço MAC, através da criação dinâmica de *faces unicast* [Baccelli et al. 2014], uma abordagem semelhante ao mapeamento ARP da arquitetura IP. Tais soluções buscam manter um constante mapeamento de endereço *unicast* a partir das informações dos cabeçalhos dos pacotes de dados recuperados de transmissões *multicast* anteriores. Essas estratégias têm a vantagem de não exigir uma mudança nos protocolos de enlace existentes. Em contrapartida, as mesmas reduzem a capacidade da rede de fazer *cache* oportunístico (característica nativa da arquitetura), uma vez que os pacotes são encaminhados para endereços específicos [Kietzmann et al. 2017].

Ainda visando tratar a inundação no nível das interfaces de rede, alguns trabalhos propõem protocolos de enlace específicos para a arquitetura NDN [Shi and Zhang 2012]. No entanto, essas soluções exigem o processamento de todos os quadros passantes, no nível do enlace, o que prejudica redes formadas por nós com restrições de processamento e energia (e.g., IoT). Além disso, essas soluções tornam necessária a criação de um novo protocolo de enlace, o que limita a adoção e a interoperabilidade com redes existentes, equipadas com protocolos como o Ethernet. Por fim, destaca-se as soluções que propõem a filtragem baseada em nome no nível do enlace [Shi et al. 2016, Li et al. 2019c, Karrakchou et al. 2020b]. Além da interoperabilidade, é uma alternativa que representa uma violação dos papéis das camadas da arquitetura de rede.

Roteamento, comunicação *multi-hop* e recuperação de dados

Tendo em vista que o roteamento apoia a mobilidade atualizando direções até um produtor/repositório e divulgando dinamicamente a alcançabilidade de prefixos, um dos principais desafios consiste em definir como outros componentes da NDN fazem uso de tais informações. Entre as questões em aberto está a: (i) definição a melhor estratégia de

encaminhamento para um determinado cenário: *multicast*, *unicast*, adaptativo, ciente de coordenadas de localização; (ii) identificação de uma estratégia de múltiplos caminhos a ser adotada, considerando o encaminhamento *stateful*, visando balanceamento de carga, caminhos cientes do *cache*, caminhos disjuntos, caminhos com propriedades de tolerância a falhas; (iii) definição de como atualizar dinamicamente as políticas de confiança para permitir autoridade sobre informações de alcançabilidade para nós distintos, diante da mobilidade do produtor.

Além disso, outro aspecto importante é o modelo de mobilidade ao qual o cenário está sujeito, que pode variar em relação à velocidade dos nós, *churn*, distância entre os nós, e diversos outros aspectos. Desta forma, é preciso considerar se: (i) o modelo de mobilidade prever mobilidade do produtor e consumidor ao mesmo tempo; (ii) a possibilidade de contar com serviços de resolução de nomes eficientes; (iii) a possibilidade da adoção de repositórios e *caches* colaborativos impulsionar a mobilidade do produtor. Estas e outras questões apontam para uma necessidade de melhor investigação sobre protocolos de roteamento e suas funções, especialmente nos ambientes *ad-hoc* com alta mobilidade.

2.7.2. Segurança

A NDN se caracteriza pela segurança no nível dos dados, garantida através do uso de primitivas criptográficas conhecidas – especialmente criptografia assimétrica e infraestrutura de chaves públicas –, que, combinadas com uma semântica expressiva do nome e um conjunto de políticas de confiança flexível, permitem a validação do dado em qualquer nó da rede. Ainda que este modelo traga benefícios de segurança à arquitetura, algumas lacunas permanecem em aberto.

Modelo de ameaças

Uma série de ameaças vem sendo apresentadas na literatura, tendo como alvo diversos componentes da arquitetura NDN [Mannes and Maziero 2019, Liu et al. 2019, Nour et al. 2021b]. No entanto, constata-se uma lacuna de investigação especialmente nas seguintes áreas: ataques de negação de serviço das mais variadas formas – distribuídos, iniciados remotamente, através de inundação de interesses, através de assinaturas falsas para dados, temperando parâmetros específicos dos pacotes de interesses/dados, entre outros; nós encaminhadores maliciosos; sequestro ou falsificação de prefixos de nome no roteamento (do inglês, *route hijacking*); acúmulo de material criptográfico a partir do uso intenso de chaves, permitindo processos de criptoanálise; e ataques à privacidade.

É importante destacar, ainda, que pouco se discute sobre os modelos econômicos associados à distribuição de conteúdo na arquitetura NDN, e, historicamente, as superfícies de ataque têm uma forte relação com os aspectos econômicos envolvidos. Assim, é possível que se observe um crescimento na diversidade de ataques e identificação de vulnerabilidades quando vantagens financeiras ficarem evidentes. Verifica-se, portanto, a necessidade de um modelo de ameaças abrangente o suficiente para incorporar mecanismos adicionais ao arcabouço de segurança da NDN e, assim, reduzir a superfície de ataques.

Mecanismos de controle de acesso, autenticação e autorização

A comunicação entre múltiplas partes é uma característica comum em aplicações distribuídas, especialmente em aplicações NDN. O modelo tradicional de infraestrutura de criptografia utilizada na NDN prevê, por padrão, apenas um certificado/identidade para assinar e cifrar os dados. Pesquisas em estágio inicial sugerem o uso de uma terceira entidade central que faz a gestão de múltiplas assinaturas ou distribuição de chaves para tratar esse desafio. [Zhang et al. 2021]. Além disso, alguns trabalhos propõem soluções de controle de acesso baseado em nome e autorização [Zhang et al. 2018c]. Em ambos os casos, o uso de uma entidade centralizada contrasta com o modelo totalmente distribuído da NDN. Além disso, as propostas sugerem mudanças nos componentes da arquitetura, deixando em aberto desafios quanto à escalabilidade, tolerância à falhas e censura.

Modelo de confiança

Na Seção 2.4.2 foram apresentados os processos de estabelecimento do modelo de confiança. Um aspecto em aberto nessa área é a disponibilidade de estratégias robustas e seguras para inicialização do modelo de segurança. Soluções como uso de um certificado único pré-instalado nos nós e contato presencial utilizando canal fora da banda para estabelecimento da cadeia de confiança, podem implicar em baixa usabilidade ou efetividade. O uso de infraestruturas de chaves públicas distribuídas, teia de confiança (*web of trust*), identidade auto-soberana e outros mecanismos modernos e robustos de gestão de identidade [Pöhn and Hommel 2021] podem representar um caminho promissor para expandir o modelo de confiança e prover suporte à requisitos mais complexos de segurança.

Monitoramento de segurança e tratamento de incidentes

Questões operacionais do tratamento de incidentes de segurança, análise forense e monitoramento de segurança, podem também revelar-se em grandes desafios de pesquisa: como identificar e responsabilizar nós responsáveis por um ataque? Como viabilizar a construção de bases de conhecimento para inteligência de ameaças? Como coletar evidências para análise forense? Como realizar monitoramento de atividade maliciosa e contenção de nós envolvidos em ataques? Ou até mesmo como realizar monitoramento de disponibilidade e desempenho da rede? Estas são algumas perguntas operacionais, cujas respostas podem envolver o desenvolvimento de novas aplicações ou estratégias.

2.7.3. Aplicações distribuídas

Os protocolos de sincronização permitem uma comunicação assíncrona e multiparte centrada em dados por meio de mecanismos que suportam consistência fraca. Por outro lado, um conjunto de aplicações demanda consistência forte, em geral implementada na camada de aplicação, no topo dos protocolos existentes de sincronismo. Algumas aplicações que se baseiam no uso de mecanismos de invocação remota na NDN, como NFaaS [Król and Psaras 2017] e RICE [Król et al. 2018], visando funções *in-network*, deverão considerar mecanismos de tolerância a falhas. Uma das alternativas seria a adoção

de máquinas de estados replicadas, visando a manutenção de serviços *stateful*. Para isso, a implementação de mecanismos de consenso, acima dos protocolos de sincronização do conjunto de dados, apresenta um importante bloco para aplicações distribuídas na NDN.

O desenvolvimento de aplicações distribuídas com suporte à consistência forte, ou mesmo em cenários de consistência fraca, se torna mais desafiador quando se considera a possibilidade de atuação de falhas bizantinas. A concepção de serviços replicados tolerantes a falhas bizantinas deve considerar um conjunto de questões de implementação prática e um completo redesenho das soluções existentes na NDN. É possível pensar em consensos que considerem o modelo de sistema da NDN e utilizem seus próprios mecanismos e o NDN Sync como base para soluções de consenso a partir de abordagens clássicas, tais como Paxos [Lamport et al. 2001] e PBFT [Castro and Liskov 1999].

Uma tendência recente tem sido o uso da arquitetura NDN em cenários não permissionados de larga escala visando a composição de *distributed ledgers* [Zhang et al. 2019b, Doku et al. 2020]. Em tais cenários, em especial, os baseados no consenso não determinístico por meio de mecanismos de prova, busca-se o melhor custo-benefício entre escalabilidade e desempenho em função do contexto da aplicação subjacente. Uma possível solução seria o uso de mecanismos de sincronização de estados adequados às estruturas das *distributed ledgers* e mecanismos de reputação para mitigar nós maliciosos.

2.8. Considerações Finais

Este capítulo apresentou as principais características e desafios da arquitetura NDN em relação aos aspectos de mobilidade, segurança e aplicações distribuídas. A NDN é uma proposta relativamente nova que busca a construção de uma Internet da Informação (i.e., em que dados são o centro do processo) e, portanto, requer uma mudança conceitual na forma de pensar a rede e as aplicações que dela fazem uso. NDN, tendo sido proposta mais de três décadas após a pilha TCP/IP, tem suas bases nos requisitos desta Internet da Informação. Assim, a arquitetura é projetada para alcançar os requisitos contemporâneos, diferente da TCP/IP, que teve que se adaptar ao longo do tempo a estes novos requisitos. O capítulo explicou as propriedades fundamentais dessa arquitetura e realizou uma associação entre as características das estratégias e protocolos utilizados e o desenvolvimento de aplicações distribuídas, aplicação de segurança na rede e adequação à cenários móveis.

A NDN consegue atender alguns requisitos de mobilidade de forma nativa, mas ainda existem diversas questões em aberto e desafios de pesquisa relacionados com as características dos meios de transmissão, encaminhamento na camada de enlace, roteamento, comunicação *multi-hop* e recuperação de dados. Do ponto de vista de segurança, as mudanças que ocorreram na arquitetura geraram a necessidade de uma segurança mais focada na transmissão dos dados. Desta forma, tem-se como principais aspectos a serem observados nesse eixo temático, a definição de modelos de ameaça que exploram as propriedades da NDN, a identificação de modelos de confiança e de mecanismos de controle de acesso, autenticação e autorização. Além disso, é importante ressaltar a necessidade do desenvolvimento de estratégias de monitoramento de segurança e tratamento de incidentes, que são fundamentais na área de redes e que, muitas vezes, atendem a requisitos legislativos. As características da NDN possibilitam um modelo de comunicação temporalmente assíncrono com uso de protocolos de sincronização de mensagens. Um conjunto

de blocos de construção para sistemas distribuídos inerentes a NDN possibilita o uso de mecanismos sofisticados de *publish/subscribe* e de sincronização de dados, favorecendo a disponibilidade da computação distribuída, mesmo com alta volatibilidade de nós, ao custo de uma consistência fraca. Isto tem sido exercitado em aplicações contemporâneas, como, por exemplo, *distributed ledgers* em NDN. Um dos desafios em aberto é o de propiciar a construção de aplicações distribuídas com maior grau de resiliência e consistência.

Devido às propriedades promissoras da NDN ao atender nativamente os requisitos de rede e de aplicações atuais, tem-se observado esforços em busca da padronização da arquitetura[12]. Paralelo a isso, observa-se a importância que tem sido dada no desenvolvimento da arquitetura através do apoio da NSF (*National Science Foundation*) nos últimos anos e do envolvimento da comunidade científica. Além disso, importantes atores da indústria têm demonstrado interesse e desenvolvido soluções NDN, tais como a *Cisco Systems*, *Fujitsu Laboratories of America*, *Huawei Technologies*, *Intel Corporation*, *Juniper Networks*, *Panasonic Corporation* e *ViaSat*. Esperamos que este texto permita uma reflexão das possibilidades da arquitetura e da forma que a vemos: projetada para um cenário de alta mobilidade, com aplicações inerentemente distribuídas e em que a segurança da informação é o cerne.

Agradecimentos

Os autores agradecem o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e da Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB).

Referências

[Afanasyev et al. 2017] Afanasyev, A., Jiang, X., Yu, Y., Tan, J., Xia, Y., Mankin, A., and Zhang, L. (2017). Ndns: A dns-like name service for ndn. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9. IEEE.

[Afanasyev et al. 2018] Afanasyev, A., Shi, J., Zhang, B., Zhang, L., Moiseenko, I., Yu, Y., Shang, W., Li, Y., Mastorakis, S., Huang, Y., Abraham, J. P., Newberry, E., DiBenedetto, S., Fan, C., Papadopoulos, C., Pesavento, D., Grassi, G., Pau, G., Zhang, H., Song, T., Yuan, H., Abraham, H. B., Crowley, P., Amin, S. O., Lehman, V., Chowdhury, M., and Wang, L. (2018). Nfd developer's guide. Technical Report NDN-0021, NDN.

[Amadeo et al. 2016] Amadeo, M., Campolo, C., Quevedo, J., Corujo, D., Molinaro, A., Iera, A., Aguiar, R. L., and Vasilakos, A. V. (2016). Information-centric networking for the internet of things: challenges and opportunities. *IEEE Network*, 30(2):92–100.

[Araújo 2018] Araújo, F. R. C. (2018). Cache colaborativo e distribuído como suporte à mobilidade de produtores em redes sem fio de dados nomeados. Dissertação (Mestrado em Ciência da Computação), Universidade Federal da Bahia, Instituto de Matemática, Programa de Pós-Graduação em Ciência da Computação, Salvador.

[12]Exemplos de padronização: `https://www.rfc-editor.org/rfc/rfc8793.html` e `https://datatracker.ietf.org/rg/icnrg/about/`

[Araújo et al. 2019] Araújo, F. R. C., de Sousa, A. M., and Sampaio, L. N. (2019). Scan-mob: An opportunistic caching strategy to support producer mobility in named data wireless networking. *Computer Networks*, 156:62–74.

[Araujo et al. 2021] Araujo, G. B., Peixoto, M. L. M., and Sampaio, L. N. (2021). Ndn4ivc: A framework for simulating and testing applications in vehicular named-data networking.

[Araujo and Sampaio 2021] Araujo, G. B. and Sampaio, L. N. (2021). An intelligent edge-traffic routing architecture for vehicular data-mule service. *IEEE Latin America Transactions*, 19(11):1976–1984.

[Araújo et al. 2018] Araújo, F. R. C., de Sousa, A. M., and Sampaio, L. N. (2018). Armazenamento oportunista em redes de dados nomeados sem fio como suporte à mobilidade de produtores. In *Anais do XXXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. SBC.

[Araújo et al. 2019] Araújo, F. R. C., de Sousa, A. M., and Sampaio, L. N. (2019). Uma estratégia de encaminhamento eficiente para redes de veículos aéreos não tripulados de dados nomeados. In *Anais do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 876–889. SBC.

[Araújo and Sampaio 2017] Araújo, F. R. C. and Sampaio, L. N. (2017). Mobilidade em ndn: Consumidores versus produtores. In *Anais do XIII Workshop de Redes P2P, Dinâmicas, Sociais e Orientadas a Conteúdo*, pages 8–13. SBC.

[Asaf et al. 2020] Asaf, K., Rehman, R. A., and Kim, B.-S. (2020). Blockchain technology in named data networks: A detailed survey. *Journal of Network and Computer Applications*, 171:102840.

[Azgin et al. 2016] Azgin, A., Ravindran, R., and Wang, G. (2016). pit/less: Stateless forwarding in content centric networks. In *2016 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7.

[Baccelli et al. 2014] Baccelli, E., Mehlis, C., Hahm, O., Schmidt, T. C., and Wählisch, M. (2014). Information centric networking in the iot: Experiments with ndn in the wild. In *Proceedings of the 1st ACM Conference on Information-Centric Networking*, ACM-ICN '14, page 77–86. ACM.

[Barka et al. 2018] Barka, E., Kerrache, C. A., Hussain, R., Lagraa, N., Lakas, A., and Bouk, S. H. (2018). A trusted lightweight communication strategy for flying named data networking. *Sensors*, 18(8).

[Brito et al. 2020] Brito, I. V. S., Sampaio, L., and Zhang, L. (2020). (Poster) Towards a distance vector routing protocol for named data networking. In *Named Data Networking Community Meeting (NDNcomm) 2020*.

[Castro and Liskov 1999] Castro, M. and Liskov, B. (1999). Practical byzantine fault tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, OSDI '99, page 173–186. USENIX Association.

[Chandra and Toueg 1996] Chandra, T. D. and Toueg, S. (1996). Unreliable failure detectors for reliable distributed systems. *Journal of the ACM (JACM)*, 43(2):225–267.

[Chandy and Lamport 1985] Chandy, K. M. and Lamport, L. (1985). Distributed snapshots: Determining global states of distributed systems. *ACM Transactions on Computer Systems (TOCS)*, 3(1):63–75.

[Cisco 2020] Cisco (2020). Cisco annual internet report (2018-2023) white paper. Disponível em: `https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html`. Último acesso em: 30 de maio de 2021.

[Conti et al. 2020] Conti, M., Gangwal, A., Hassan, M., Lal, C., and Losiouk, E. (2020). The road ahead for networking: A survey on icn-ip coexistence solutions. *IEEE Communications Surveys Tutorials*, 22(3):2104–2129.

[Cristian and Schmuck 1995] Cristian, F. and Schmuck, F. (1995). Agreeing on processor group membership in timed asynchronous distributed systems. *Report CSE95-428, UC San Diego*.

[de Sousa et al. 2018a] de Sousa, A. M., Araújo, F. R. C., and Sampaio, L. N. (2018a). A link-stability-based interest-forwarding strategy for vehicular named data networks. *IEEE Internet Computing*, 22(3):16–26.

[de Sousa et al. 2019] de Sousa, A. M., Araújo, F. R. C., Greve, F. G. P., and Sampaio, L. N. (2019). Um arcabouço para validação de conteúdo em redes de dados nomeados baseado em blockchain. In *Anais do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 155–168. SBC.

[de Sousa et al. 2018b] de Sousa, A. M., Araújo, F. R. C., and Sampaio, L. N. (2018b). Encaminhamento seletivo de interesses em redes veiculares de dados nomeados baseado no tempo de vida do enlace. In *Anais do XXXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. SBC.

[Doku et al. 2020] Doku, R., Rawat, D. B., Garuba, M., and Njilla, L. (2020). Fusion of named data networking and blockchain for resilient internet-of-battlefield-things. In *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–6.

[Dolev et al. 1987] Dolev, D., Dwork, C., and Stockmeyer, L. (1987). On the minimal synchronism needed for distributed consensus. *Journal of the ACM*, 34(1):77–97.

[Eugster et al. 2003] Eugster, P. T., Felber, P. A., Guerraoui, R., and Kermarrec, A.-M. (2003). The many faces of publish/subscribe. *ACM Computing Surveys*, 35(2):114–131.

[Fang et al. 2018] Fang, C., Yao, H., Wang, Z., Wu, W., Jin, X., and Yu, F. R. (2018). A survey of mobile information-centric networking: Research issues and challenges. *IEEE Communications Surveys Tutorials*, 20(3):2353–2371.

[Fisher et al. 1985] Fisher, M. J., Lynch, N., and Paterson, M. S. (1985). Impossibility of distributed consensus with one faulty process. *Journal of the ACM*, 32(2):374–382.

[Fotiou and Polyzos 2016] Fotiou, N. and Polyzos, G. C. (2016). Decentralized name-based security for content distribution using blockchains. In *Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on*, pages 415–420. IEEE.

[Gawande et al. 2019] Gawande, A., Clark, J., Coomes, D., and Wang, L. (2019). Decentralized and secure multimedia sharing application over named data networking. In *Proceedings of the 6th ACM Conference on Information-Centric Networking*, ICN '19, page 19–29. ACM.

[Grassi et al. 2014] Grassi, G., Pesavento, D., Pau, G., Vuyyuru, R., Wakikawa, R., and Zhang, L. (2014). Vanet via named data networking. In *2014 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*, pages 410–415. IEEE.

[Ioannou and Weber 2016] Ioannou, A. and Weber, S. (2016). A survey of caching policies and forwarding mechanisms in information-centric networking. *IEEE Communications Surveys Tutorials*, 18(4):2847–2886.

[Jacobson et al. 2009] Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H., and Braynard, R. L. (2009). Networking named content. In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT'09, pages 1–12. ACM.

[Jin et al. 2017] Jin, T., Zhang, X., Liu, Y., and Lei, K. (2017). Blockndn: A bitcoin blockchain decentralized system over named data networking. In *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 75–80.

[Kapetanidou et al. 2020] Kapetanidou, I. A., Hassan, M., Sarros, C.-A., Conti, M., and Tsaoussidis, V. (2020). Reputation-based trust: A robust mechanism for dynamic adaptive streaming over named data networking. In *2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE)*, pages 114–121.

[Karrakchou et al. 2020a] Karrakchou, O., Samaan, N., and Karmouch, A. (2020a). Endn: An enhanced ndn architecture with a p4-programmabie data plane. In *Proceedings of the 7th ACM Conference on Information-Centric Networking*, ICN '20, page 1–11. ACM.

[Karrakchou et al. 2020b] Karrakchou, O., Samaan, N., and Karmouch, A. (2020b). Fc-trees: A front-coded family of compressed tree-based fib structures for ndn routers. *IEEE Transactions on Network and Service Management*, 17(2):1167–1180.

[Khelifi et al. 2020] Khelifi, H., Luo, S., Nour, B., Moungla, H., Faheem, Y., Hussain, R., and Ksentini, A. (2020). Named data networking in vehicular ad hoc networks: State-of-the-art and challenges. *IEEE Communications Surveys Tutorials*, 22(1):320–351.

[Kietzmann et al. 2017] Kietzmann, P., Gündoğan, C., Schmidt, T. C., Hahm, O., and Wählisch, M. (2017). The need for a name to mac address mapping in ndn: Towards

quantifying the resource gain. In *Proceedings of the 4th ACM Conference on Information-Centric Networking*, ICN '17, page 36–42. ACM.

[Król et al. 2018] Król, M., Habak, K., Oran, D., Kutscher, D., and Psaras, I. (2018). Rice: Remote method invocation in icn. In *Proceedings of the 5th ACM Conference on Information-Centric Networking*, ICN '18, page 1–11. ACM.

[Król and Psaras 2017] 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. ACM.

[Lamport 2019] Lamport, L. (2019). Time, clocks, and the ordering of events in a distributed system. In *Concurrency: The Works of Leslie Lamport*, page 179–196. ACM.

[Lamport et al. 2001] Lamport, L. et al. (2001). Paxos made simple. *ACM Sigact News*, 32(4):18–25.

[Lamport and Fischer 1982] Lamport, L. and Fischer, M. (1982). Byzantine generals and transaction commit protocols. Technical Report 62, SRI International.

[Lee et al. 2018] Lee, C. A., Zhang, Z., Tu, Y., Afanasyev, A., and Zhang, L. (2018). Supporting virtual organizations using attribute-based encryption in named data networking. In *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*, pages 188–196. IEEE.

[Li et al. 2019a] Li, T., Kong, Z., Mastorakis, S., and Zhang, L. (2019a). Distributed dataset synchronization in disruptive networks. In *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 428–437. IEEE.

[Li et al. 2018] Li, T., Shang, W., Afanasyev, A., Wang, L., and Zhang, L. (2018). A brief introduction to ndn dataset synchronization (ndn sync). In *IEEE Military Communications Conference (MILCOM)*, pages 612–618.

[Li et al. 2019b] Li, Y., Zhang, Z., Wang, X., Lu, E., Zhang, D., and Zhang, L. (2019b). A secure sign-on protocol for smart homes over named data networking. *IEEE Communications Magazine*, 57(7):62–68.

[Li et al. 2019c] Li, Z., Xu, Y., Zhang, B., Yan, L., and Liu, K. (2019c). Packet forwarding in named data networking requirements and survey of solutions. *IEEE Communications Surveys Tutorials*, 21(2):1950–1987.

[Liu et al. 2019] Liu, G., Quan, W., Cheng, N., Zhang, H., and Yu, S. (2019). Efficient ddos attacks mitigation for stateful forwarding in internet of things. *Journal of Network and Computer Applications*, 130:1–13.

[Lynch 1996] Lynch, N. (1996). *Distributed Algorithms*. Morgan Kaufmann Publishers, Inc., San Francisco, CA, USA.

[Madureira et al. 2021] Madureira, A. L. R., Araújo, F. R. C., Araújo, G. B., and Sampaio, L. N. (2021). Ndn fabric: Where the software-defined networking meets the content-centric model. *IEEE Transactions on Network and Service Management*, 18(1):374–387.

[Madureira et al. 2020] Madureira, A. L. R., Araújo, F. R. C., Prates, L. N. B., and Sampaio, L. N. (2020). Ndn-adap: Uma arquitetura para encaminhamento eficiente de pacotes em redes de dados nomeados. In *Anais do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 812–825. SBC.

[Mannes and Maziero 2019] Mannes, E. and Maziero, C. (2019). Naming content on the network layer: A security analysis of the information-centric network model. *ACM Computing Surveys*, 52(3).

[Marxer and Tschudin 2017] Marxer, C. and Tschudin, C. (2017). Schematized access control for data cubes and trees. In *Proceedings of the 4th ACM Conference on Information-Centric Networking*, ICN '17, page 170–175. ACM.

[Mastorakis et al. 2017] 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.

[Mastorakis et al. 2018] Mastorakis, S., Gusev, P., Afanasyev, A., and Zhang, L. (2018). Real-time data retrieval in named data networking. In *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*, pages 61–66. IEEE.

[Miguel et al. 2018] Miguel, R., Signorello, S., and Ramos, F. M. V. (2018). Named data networking with programmable switches. In *2018 IEEE 26th International Conference on Network Protocols (ICNP)*, pages 400–405.

[Moiseenko and Oran 2017] Moiseenko, I. and Oran, D. (2017). Path switching in content centric and named data networks. In *Proceedings of the 4th ACM Conference on Information-Centric Networking*, ICN '17, pages 66–76. ACM.

[Mori 2018] Mori, S. (2018). Secure caching scheme by using blockchain for information-centric network-based wireless sensor networks. *Journal of Signal Processing*, 22(3):97–108.

[NDN 2021] NDN (2021). Ndn testbed status. Disponível em: `http://ndndemo.arl.wustl.edu/ndn.html`. Último acesso em: 07 de maio de 2021.

[Nour et al. 2021a] Nour, B., Khelifi, H., Hussain, R., Mastorakis, S., and Moungla, H. (2021a). Access control mechanisms in named data networks: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 54(3):1–35.

[Nour et al. 2019a] Nour, B., Li, F., Khelifi, H., Moungla, H., and Ksentini, A. (2019a). Coexistence of icn and ip networks: An nfv as a service approach. In *2019 IEEE Global Communications Conference (GLOBECOM)*, page 1–6. IEEE Press.

[Nour et al. 2021b] Nour, B., Mastorakis, S., Ullah, R., and Stergiou, N. (2021b). Information-centric networking in wireless environments: Security risks and challenges. *IEEE Wireless Communications*, 28(2):121–127.

[Nour et al. 2019b] Nour, B., Sharif, K., Li, F., Biswas, S., Moungla, H., Guizani, M., and Wang, Y. (2019b). A survey of internet of things communication using icn: A use case perspective. *Computer Communications*, 142-143:95–123.

[Nour et al. 2019c] Nour, B., Sharif, K., Li, F., Yang, S., Moungla, H., and Wang, Y. (2019c). Icn publisher-subscriber models: Challenges and group-based communication. *IEEE Network*, 33(6):156–163.

[Pires et al. 2019] Pires, S., Araújo, F. R. C., Freitas, A. E. S., and Sampaio, L. N. (2019). Análise de perfis de usuários de música e seus impactos no desempenho de políticas de substituição de cache. In *Anais do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 848–861. SBC.

[Pires et al. 2021] Pires, S., Ziviani, A., and Sampaio, L. (2021). Contextual dimensions for cache replacement schemes in information-centric networks: a systematic review. *PeerJ Computer Science*, 7:e418.

[Pires et al. 2018] Pires, S. S., Ribeiro, A. V., de Sousa, A. M., Freitas, A. E. S., and Sampaio, L. N. (2018). On evaluating the influence of user's music listening habits on cache replacement policies. In *2018 IEEE Symposium on Computers and Communications (ISCC)*, pages 00930–00933.

[Pöhn and Hommel 2021] Pöhn, D. and Hommel, W. (2021). Proven and modern approaches to identity management. In *Advances in Cybersecurity Management*, pages 421–443. Springer.

[Psaras et al. 2018] Psaras, I., Ascigil, O., Rene, S., Pavlou, G., Afanasyev, A., and Zhang, L. (2018). Mobile data repositories at the edge. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*. USENIX Association.

[Ramani and Afanasyev 2020a] Ramani, S. K. and Afanasyev, A. (2020a). Certcoalesce: Efficient certificate pool for ndn-based systems. In *Proceedings of the 7th ACM Conference on Information-Centric Networking*, ICN '20, page 158–160. ACM.

[Ramani and Afanasyev 2020b] Ramani, S. K. and Afanasyev, A. (2020b). Rapid establishment of transient trust for ndn-based vehicular networks. In *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6. IEEE.

[Ramani et al. 2019] Ramani, S. K., Tourani, R., Torres, G., Misra, S., and Afanasyev, A. (2019). Ndn-abs: Attribute-based signature scheme for named data networking. In *Proceedings of the 6th ACM Conference on Information-Centric Networking*, ICN '19, page 123–133. ACM.

[Ribeiro et al. 2017] Ribeiro, A. V., Sampaio, L. N., and Ziviani, A. (2017). Explorando a afinidade de usuários para descarregamento de dados mais eficiente em redes celulares de pequeno porte. In *Anais do XXXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. SBC.

[Ribeiro et al. 2018] Ribeiro, A. V., Sampaio, L. N., and Ziviani, A. (2018). Affinity-based user clustering for efficient edge caching in content-centric cellular networks. In *2018 IEEE Symposium on Computers and Communications (ISCC)*, pages 00474–00479.

[Rondon et al. 2020] Rondon, L. B., da Costa, J. B., Filho, G. P. R., Rosário, D., and Villas, L. A. (2020). Degree centrality-based caching discovery protocol for vehicular named-data networks. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–5.

[Saxena et al. 2016] Saxena, D., Raychoudhury, V., Suri, N., Becker, C., and Cao, J. (2016). Named data networking: a survey. *Computer Science Review*, 19:15–55.

[Sedky and Mougy 2018] Sedky, G. and Mougy, A. E. (2018). Bcxp: Blockchain-centric network layer for efficient transaction and block exchange over named data networking. In *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*, pages 449–452.

[Seixas et al. 2017] Seixas, N. F. S., Ribeiro, A. V., and Sampaio, L. N. (2017). Um modelo de rede centrada na informação resiliente a ataques de negação de serviços por inundação de interesses. In *Anais do XXXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. SBC.

[Serhane et al. 2021] Serhane, O., Yahyaoui, K., Nour, B., and Moungla, H. (2021). A survey of icn content naming and in-network caching in 5g and beyond networks. *IEEE Internet of Things Journal*, 8(6):4081–4104.

[Shang et al. 2017] Shang, W., Yu, Y., Wang, L., Afanasyev, A., and Zhang, L. (2017). A survey of distributed dataset synchronization in named data networking. Technical Report NDN-0053, NDN.

[Shannigrahi et al. 2017] Shannigrahi, S., Fan, C., and Papadopoulos, C. (2017). Request aggregation, caching, and forwarding strategies for improving large climate data distribution with ndn: A case study. In *Proceedings of the 4th ACM Conference on Information-Centric Networking*, ICN '17, page 54–65. ACM.

[Shi et al. 2016] Shi, J., Liang, T., Wu, H., Liu, B., and Zhang, B. (2016). Ndn-nic: Name-based filtering on network interface card. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, ACM-ICN '16, pages 40–49. ACM.

[Shi et al. 2020] Shi, J., Pesavento, D., and Benmohamed, L. (2020). Ndn-dpdk: Ndn forwarding at 100 gbps on commodity hardware. In *Proceedings of the 7th ACM Conference on Information-Centric Networking*, ICN '20, page 30–40. ACM.

[Shi and Zhang 2012] Shi, J. and Zhang, B. (2012). Ndnlp: A link protocol for ndn. Technical Report NDN-0006, NDN.

[Signorello et al. 2016] Signorello, S., State, R., François, J., and Festor, O. (2016). Ndn.p4: Programming information-centric data-planes. In *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, pages 384–389.

[Simon 2000] Simon, S. (2000). Brewer's cap theorem. *CS341 Distributed Information Systems, University of Basel (HS2012)*.

[Siracusano et al. 2018] Siracusano, G., Salsano, S., Ventre, P., Detti, A., Rashed, O., and Blefari-Melazzi, N. (2018). A framework for experimenting icn over sdn solutions using physical and virtual testbeds. *Computer Networks*, 134:245–259.

[Stallings 2020] Stallings, W. (2020). *Cryptography and Network Security: Principles and Practice*. Pearson, USA, 8th edition.

[Stoller 1997] Stoller, S. D. (1997). Leader election in distributed systems with crash failures. Technical Report 481, Computer Science Dept., Indiana University. Revised July 1997.

[Tariq et al. 2020] Tariq, A., Rehman, R. A., and Kim, B. (2020). Forwarding strategies in ndn-based wireless networks: A survey. *IEEE Communications Surveys Tutorials*, 22(1):68–95.

[Tehrani et al. 2019] Tehrani, P. F., Keidel, L., Osterweil, E., Schiller, J. H., Schmidt, T. C., and Wählisch, M. (2019). Ndnssec: Namespace management in ndn with dnssec. In *Proceedings of the 6th ACM Conference on Information-Centric Networking*, ICN '19, page 171–172. ACM.

[Thompson et al. 2019] Thompson, J., Gusev, P., and Burke, J. (2019). Ndn-cnl: A hierarchical namespace api for named data networking. In *Proceedings of the 6th ACM Conference on Information-Centric Networking*, ICN '19, page 30–36. ACM.

[Tizvar and Abbaspour 2020] Tizvar, R. and Abbaspour, M. (2020). A density-aware probabilistic interest forwarding method for content-centric vehicular networks. *Vehicular Communications*, 23:100216.

[Waldo et al. 1996] Waldo, J., Wyant, G., Wollrath, A., and Kendall, S. (1996). A note on distributed computing. In *International Workshop on Mobile Object Systems*, pages 49–64. Springer.

[Wang et al. 2020] Wang, J., Luo, J., Zhou, J., and Ran, Y. (2020). A mobility-predict-based forwarding strategy in vehicular named data networks. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pages 01–06.

[Wang et al. 2018] 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.

[Wang and Li 2020] Wang, X. and Li, Y. (2020). Content delivery based on vehicular cloud. *IEEE Transactions on Vehicular Technology*, 69(2):2105–2113.

[Wang et al. 2021] Wang, X., Wang, X., and Wang, D. (2021). Cost-efficient data retrieval based on integration of vc and ndn. *IEEE Transactions on Vehicular Technology*, 70(1):967–976.

[Yu et al. 2021] Yu, T., Zhang, Z., Ma, X., Moll, P., and Zhang, L. (2021). A pub/sub api for ndn-lite with built-in security. Technical Report NDN-0071, NDN.

[Yu et al. 2015] Yu, Y., Afanasyev, A., Clark, D., claffy, k., Jacobson, V., and Zhang, L. (2015). Schematizing trust in named data networking. In *Proceedings of the 2nd ACM Conference on Information-Centric Networking*, ACM-ICN '15, page 177–186. ACM.

[Yu et al. 2017] Yu, Y., Afanasyev, A., Seedorf, J., Zhang, Z., and Zhang, L. (2017). Ndn delorean: An authentication system for data archives in named data networking. In *Proceedings of the 4th ACM Conference on Information-Centric Networking*, ICN '17, page 11–21. ACM.

[Zhang et al. 2018a] Zhang, H., Li, Y., Zhang, Z., Afanasyev, A., and Zhang, L. (2018a). Ndn host model. *ACM SIGCOMM Computer Communication Review*, 48(3):35–41.

[Zhang 2019] Zhang, L. (2019). The role of data repositories in named data networking. In *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–5.

[Zhang et al. 2014] 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 et al. 2010] Zhang, L., Estrin, D., Burke, J., Jacobson, V., Thornton, J. D., Smetters, D. K., Zhang, B., Tsudik, G., Claffy, K., Krioukov, D., Massey, D., Papadopoulos, C., Abdelzaher, T., Wang, L., Crowley, P., and Yeh, E. (2010). Named data networking (ndn) project. Technical Report NDN-0001, NDN.

[Zhang et al. 2016] Zhang, Y., Afanasyev, A., Burke, J., and Zhang, L. (2016). A survey of mobility support in named data networking. In *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 83–88.

[Zhang et al. 2019a] Zhang, Y., Xia, Z., Afanasyev, A., and Zhang, L. (2019a). A note on routing scalability in named data networking. In *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6.

[Zhang et al. 2018b] Zhang, Y., Xia, Z., Mastorakis, S., and Zhang, L. (2018b). Kite: Producer mobility support in named data networking. In *Proceedings of the 5th ACM Conference on Information-Centric Networking*, ICN '18, pages 125–136. ACM.

[Zhang et al. 2021] Zhang, Z., Liu, S., King, R., and Zhang, L. (2021). Supporting multiparty signing over named data networking.

[Zhang et al. 2019b] Zhang, Z., Vasavada, V., Ma, X., and Zhang, L. (2019b). Dledger: An iot-friendly private distributed ledger system based on dag.

[Zhang et al. 2018c] Zhang, Z., Yu, Y., Ramani, S. K., Afanasyev, A., and Zhang, L. (2018c). Nac: Automating access control via named data. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, pages 626–633.

[Zhang et al. 2018d] Zhang, Z., Yu, Y., Zhang, H., Newberry, E., Mastorakis, S., Li, Y., Afanasyev, A., and Zhang, L. (2018d). An overview of security support in named data networking. *IEEE Communications Magazine*, 56(11):62–68.