D-NAC: Controle de acesso distribuído para redes de dados nomeados

Italo Valcy S Brito[1], **Katharine Schramm**[1], **Leobino Sampaio**[1]

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

{italovalcy, leobino}@ufba.br, katharineschramm@gmail.com

Abstract. *Named Data Networking (NDN) offers a secure and optimized alternative for content distribution applications, a primary activity on the Internet. One of the main requirements for those applications is access control. Access control specifies authorization and accounting rules for certain entities when requesting content. Name-based Access Control (NAC) solutions utilize NDN's semantically meaningful naming to express the access policy and granularity. The Access Manager, one of the components of the NAC solution, centralizes the access control policies and key management process. However, the centralized design of Access Manager contrasts with NDN's distributed architecture, which can lead to bottleneck and availability issues due to individual failures. In this paper, we present the design and evaluation of D-NAC, which enhances NAC architecture by making the access manager fully distributed. Experimental results demonstrate the resilience and performance of D-NAC compared to standard NAC solution, with minimal overhead to the network.*

Resumo. *A Redes de Dados Nomeados (NDN) oferecem uma alternativa segura e otimizada para cenários de distribuição de conteúdo, cujas aplicações lideram os rankings de uso da Internet atual e previsões futuras. Um dos requisitos destas aplicações está no controle de acesso, que visa garantir aplicação de políticas de autorização e contabilização aos dados requisitados. Para garantir o controle de acesso, soluções de NAC (do inglês Name-based Access Control) fazem uso da semântica do esquema de nomeação da NDN para introduzir uso de chaves criptográficas que garantem confidencialidade e controle de acesso aos dados. Um dos componentes da solução de NAC é o gerenciador de acesso, que centraliza a aplicação das políticas de controle, criação, gerenciamento e revogação de chaves. O uso de uma entidade centralizada contrasta com o modelo totalmente distribuído da NDN, propiciando gargalos e indisponibilidade no advento de falhas. Este artigo apresenta o design, prototipagem e avaliação do D-NAC, uma melhoria da solução NAC que visa tornar o gerenciador de acesso um componente totalmente distribuído. Uma avaliação experimental demonstra que o D-NAC proporciona resiliência a falhas e melhor desempenho para os consumidores comparado com o NAC, sem trazer impactos significativos no total de dados transmitidos na rede.*

1. Introdução

Redes de Dados Nomeados (do inglês, *Named Data Networking* – NDN) é uma proposta *clean-slate* para a Internet do Futuro cujo foco é a distribuição de conteúdo através

de uma semântica enriquecida de nomeação de dados, culminando em uma arquitetura com suporte nativo a segurança, encaminhamento *stateful* e *caching* oportunístico [Zhang et al. 2014, Sampaio et al. 2021]. Tais características convergem com os requisitos das Redes de Distribuição de Conteúdo (do inglês, *Content Delivery Networks* – CDN), cujas aplicações lideram os *rankings* de uso da Internet atual e previsões futuras [Liang et al. 2023]. NDN pode fornecer recursos para entrega de conteúdo seguro e otimizado pelas CDNs [Thelagathoti et al. 2020], porém um dos desafios em aberto é o controle de acesso que faça cumprir as políticas de autorização e contabilização alinhadas ao modelo de negócio, ao passo que mantendo as vantagens de *caching* da NDN.

O controle de acesso em redes TCP/IP, cujo modelo de comunicação é cliente-servidor, é gerenciado pelo servidor do provedor de conteúdo que contém regras especificando quais clientes possuem acesso a determinados conteúdos [Nour et al. 2021]. Na arquitetura NDN, por outro lado, o conteúdo pode ser obtido a partir da cache de qualquer nó NDN, impondo mudanças fundamentais na estratégia de controle de acesso. Diversos trabalhos exploram mecanismos de controle de acesso em NDN [Nour et al. 2021, Sampaio et al. 2021, Zhang et al. 2018], alguns propondo uma adaptação da NDN de volta ao modelo cliente-servidor, forçando o cliente a sempre obter os dados a partir do produtor original (e assim aplicar as regras de controle de acesso), outros valendo-se das características nativas da NDN para modelar novos esquemas de controle de acesso, e ainda soluções híbridas. Em particular, uma solução que se beneficia do esquema de nomeação da NDN é a proposta NAC (do inglês *Name-based Access Control*) [Zhang et al. 2018].

No NAC, a ideia básica consiste em definir uma nova entidade chamada "gerenciador de acesso", que criará políticas de controle de acesso, e o produtor de dados que criará chaves de sessão para cifrar/decifrar os dados. Tais chaves de sessão, por sua vez, são protegidas com uso de criptografia assimétrica, cuja chave pública é distribuída a partir da chave do consumidor. As chaves utilizadas para cifrar/decifrar os dados são informadas através do esquema de nomeação, o que torna a aplicação de NAC bastante escalável e flexível para cenários com múltiplos dados a serem protegidos. Nessa modelagem o nó gerenciador de acesso executa um papel central e crítico para o serviço do NAC, sendo responsável pelo controle de acesso dos consumidores, pela criação, revogação e disponibilização de chaves. Falhas no gerenciador de acesso podem impactar uso do serviço por novos consumidores, bem como impactar no processo de gerenciamento da chave. De fato, o uso de uma entidade centralizada como o gerenciador de acesso contrasta com o modelo totalmente distribuído da NDN, propiciando gargalos e indisponibilidade no advento de falhas [Sampaio et al. 2021].

Este artigo apresenta o *design*, prototipagem e avaliação do D-NAC (*Nome Descritivo da Proposta*), que consiste em uma melhoria na arquitetura do NAC, em especial no componente do Gerenciador de Acesso, que passa a ser composto por diferentes instâncias sincronizadas e distribuídas na rede, com alcançabilidade controlada pelo protocolo de roteamento NDN. O uso de protocolos de sincronização garante a consistência dos dados da aplicação entre as múltiplas instâncias do gerenciador de acesso, ao passo que o protocolo de roteamento realiza a tarefa de disseminação de informações de alcançabilidade para o prefixo comum do gerenciador de acesso e também garante as melhores rotas para novos consumidores. Para validar a proposta D-NAC, apresenta-se um

caso de uso cuja modelagem da aplicação se beneficia da arquitetura distribuída para implantar seu modelo de negócio. Além disso, experimentos são realizados para demonstrar o funcionamento da proposta, verificação de cenários de falha e avaliação de desempenho comparando a solução original de NAC com a proposta D-NAC.

Este artigo está organizado da seguinte forma: a Seção 2 apresenta os fundamentos e trabalhos relacionados; a Seção 3 descreve o *design* do D-NAC, funcionalidades, esquema de nomeação e modelagem do caso de uso; a Seção 4 apresenta uma avaliação de desempenho do D-NAC; e a Seção 5 apresenta as conclusões do trabalho.

2. Fundamentos e Trabalhos Relacionados

Esta seção descreve alguns conceitos, estratégias e trabalhos relacionados ao D-NAC. Fundamentos sobre o controle acesso em redes de dados nomeados, estratégias de sincronização, estratégias de roteamento e publicações que estão relacionadas ao D-NAC são alguns dos tópicos apresentados a seguir.

2.1. NAC - Controle de acesso baseado em nomes

Um dos requisitos básicos presente no modelo de negócio de muitas aplicações atuais é o controle e contabilização de acesso com base no consumidor. Em contraponto com arquitetura TCP/IP, cujo foco é nos *hosts*, a arquitetura NDN tem como base o encaminhamento centrado na Informação, portanto adicionar o controle de acesso com base no consumidor à equação, além de não ser nativo da arquitetura, impõe desafios principalmente relacionados a manutenção das características essenciais da arquitetura, como cache distribuído, semântica de nomeação, e resiliência.

A partir da necessidade de incorporar controle de acesso à arquitetura NDN, diversos mecanismos foram propostos [Nour et al. 2021]. Dentre as soluções baseadas em cifragem, uma das mais relevantes foi a proposta por [Zhang et al. 2018], se destacando por fornecer controle de acesso semanticamente significativo e automatizado, sendo compatível com a natureza nomeada da NDN.

Controle de Acesso Baseado em Nome (do inglês *Name Based Access Control - NAC*) utiliza uma convenção de nomes para chaves que possibilita recuperá-las automaticamente, implementando controle de acesso granular através de seu *design* de espaços de nomeação. Em [Zhang et al. 2018], uma nova entidade é introduzida: o gerenciador de acesso. Seu papel é propagar as políticas de controle de acesso através de pares de chaves públicas e privadas nomeadas.

Tal como em outras aplicações na Internet, o NAC faz uso de uma combinação entre criptografia simétrica e assimétrica, onde os dados são cifrados pela chave simétrica CK, e chaves assimétricas são usadas para distribuição da CK. A seguinte nomenclatura se aplica às chaves assimétricas: **KEK (*Key Encryption Key*)**, chave utilizada para cifrar a CK de acordo com uma determinada granularidade que define as políticas de controle de acesso; **KDK (*Key Decryption Key*)**, chave utilizada para decifrar a CK. Antes de ser distribuída para consumidores autorizados, a KDK é cifrada com a chave pública do consumidor. Um esquema de nomeação semanticamente enriquecido permite determinar as chaves KEK, KDK e CK a serem empregadas no processo de cifragem e decifragem dos dados, conforme ilustrado a seguir:

Nome Conteúdo: `/<NomeOriginalConteudo>/ENCRYPTED-BY/<NomeCK>`
Nome CK: `/<NomeCK>/ENCRYPTED-BY/<PrefixoApp>/KEK/<id chave nac>`
Nome KDK: `/<PrefixoApp>/KDK/<id chave nac>/ENCRYPTED-BY/<PrefixoCliente>/KEY/<id chave cliente>`

De acordo com [Zhang et al. 2018], no advento da necessidade de revogação dos direitos de acesso, tendo em vista a troca recorrente de chaves, a não renovação dos direitos de acesso por parte do gerenciador é suficiente para evitar consumo ilícito de novos dados. Caso haja urgência nessa revogação, uma notificação pode ser enviada para os produtores forçando a troca da CK e disponibilizando uma nova KEK para sua cifragem.

Em casos de eventual indisponibilidade do gerenciador de acesso decorrente de instabilidade de conexão na rede, [Zhang et al. 2018] argumenta que as chaves publicadas pelo gerenciador podem ser armazenadas em um *data repository* [Zhang 2019], possibilitando que produtores e consumidores continuem a funcionar normalmente. Contudo, devemos perceber que durante esse período nenhuma nova política poderá ser estabelecida, não sendo possível adicionar novos consumidores habilitados ou revogar acesso indesejado.

2.2. Sincronização

Para o desenvolvimento de aplicativos distribuídos geralmente é necessário compartilhar conjuntos de dados entre várias partes, nesse contexto a sincronização de dados surge como uma forma de generalização para entrega de dados confiáveis multiparte. A arquitetura NDN permite serviços de sincronização utilizando *namespaces* específicos, através dos quais formam-se grupos com produtores e consumidores interessados em um determinado *conjunto de dados* [Li et al. 2018]. Em particular, protocolos de sincronização devem prover esquemas de nomeação eficiente para os *datasets* disponibilizados, mecanismos de notificação de mudança de estado, e mecanismos eficientes de comparação de estado para recuperação de dados ausentes.

Exemplos de protocolos de sincronização em NDN incluem a utilização de técnicas como árvores [Ben Abraham and Crowley 2013], filtro de Bloom invertido [Fu et al. 2015] e vetores de estado [Shang et al. 2017], dentre outros. Dentre as principais soluções existentes, os protocolos de vetores de estado possuem destaque especial em cenários de rede disruptivos, com enlaces sujeitos a falhas frequentes [Li et al. 2019].

A Sincronização de Vetor de Estado (do inglês *State Vector Sync - SVS*) [Philipp Moll and Zhang 2021], propõe o uso de um vetor de estado dinâmico, que encapsula os prefixos dos participantes e seus números de sequência. [Sampaio et al. 2021] cita que essas adaptações podem ser muito vantajosas, uma vez que proporcionam comunicação assíncrona, modelo de dados totalmente distribuído e possibilidade de sincronização incremental até mesmo quando do particionamento da rede.

O SVS utiliza um prefixo de grupo compartilhado, que, através de estratégia de encaminhamento *multicast*, permite alcançar os participantes envolvidos. Além do prefixo de grupo, cada nó utiliza um prefixo de publicação próprio para disponibilização dos dados produzidos. Os interesses de sincronização são enviados quando mudanças ocorrem no *dataset* e também periodicamente, para garantia de consistência [Philipp Moll and Zhang 2021]. Ademais, os interesses são assinados e validados com um modelo de confiança, garantindo a confiabilidade das notificações do SVS.

2.3. Protocolos de roteamento

O protocolo de roteamento mais comumente adotado na NDN é o Roteamento de Estado de Enlace de Dados Nomeados (do inglês *Named-data Link State Routing - NLSR*) [Hoque et al. 2013]. Esse protocolo calcula uma lista ordenada de próximos saltos definida através da execução do algoritmo de Dijkstra em cada interface ativa dos roteadores. Essas informações de roteamento calculadas são então disseminadas pela rede utilizando protocolos de sincronização.

Além do NLSR, diversos outros protocolos de roteamento NDN foram desenvolvidos para tratar limitações específicas. O MUCA [Ghasemi et al. 2018], por exemplo, combina o algoritmo de estado de enlace com vetor distância para melhorar a escalabilidade do NLSR. O *NDN Distance Vector Routing (NDVR)* [Brito and Sampaio 2021], foi desenvolvido para cenários de rede disruptivo, onde mudanças frequentes na topologia impactam protocolos que dependem de sincronização como NLSR. O NDVR é baseado no algoritmo vetor distância e foi desenvolvido com o objetivo de realizar a troca de informações de alcançabilidade e determinação de caminhos para NDN de forma leve e eficiente [Brito and Sampaio 2021]. Combinando mecanismos de detecção e manutenção dinâmica de vizinhos, com a troca de informações de alcançabilidade de forma distribuída e assíncrona, o NDVR possui reduzida quantidade de mensagens do protocolo e forte aproveitamento do esquema de nomeação para difusão de informações importantes na tomada de decisões de roteamento.

3. D-NAC

Dois aspectos importantes para o design e desenvolvimento do D-NAC são a descoberta de gerenciadores de acesso e consistência entre eles. No que tange à descoberta de serviços, consumidores devem ser aptos a encontrar o gerenciador de acesso mais próximo na rede. Os gerenciadores de acesso, por sua vez, partem da premissa sobre a consistência das políticas e chaves de controle de acesso entre si. Para acomodar esses requisitos, o D-NAC faz uso de soluções existentes na NDN, respectivamente: protocolo de roteamento NDVR e protocolo de sincronização SVS.

Para facilitar a discussão da solução, utilizamos um caso de uso, a ndnflix, uma aplicação fictícia de um possível serviço de *streaming* de vídeo na NDN.

3.1. Caso de uso

A ndnflix é um serviço fictício de distribuição de filmes e séries para múltiplos assinantes no mundo NDN. Seu modelo de negócio determina que para ter acesso ao seu catálogo, usuários precisam ter uma assinatura ativa, o que requer renovação mensal mediante pagamento ou acordo de colaboração específico. Além disso, o ndnflix precisa de outros de tipos de controle granular, como por exemplo controle por região, garantindo que usuários consumam apenas conteúdo disponível para sua localidade.

Uma solução simples para prover esse controle seria criptografar o conteúdo para cada assinante utilizando suas chaves públicas, resolvendo o problema do consumo não autorizado. Contudo, essa abordagem despreza um dos grandes benefícios da arquitetura NDN, seu cache oportunístico, visto que cada consumidor implicaria em uma cópia do mesmo conteúdo nas caches.

Em contrapartida, quando utilizamos o NAC, usuários podem resgatar o conteúdo desejado de qualquer local que esteja disponível na rede, tendo em vista que o conteúdo em si é cifrado apenas com a CK, sendo necessário somente resgatar uma chave específica de descriptografia para decifrá-la. Deste modo, o NAC seria o mecanismo de controle de acesso mais adequado para a ndnflix.

Considerando que a ndnflix possui diversos assinantes, um gerenciador de acesso único e centralizado para toda sua rede tornaria o serviço pouco confiável. Múltiplos consumidores requisitando permissões ao mesmo tempo para um mesmo servidor provocaria gargalos, principalmente considerando a troca de chaves recorrente modelada pelo NAC. Adicionalmente, em caso de falha do gerenciador de acesso, novos assinantes e o processo de mudança de chaves seriam impactados.

Considerando os fatores supracitados, uma adaptação do NAC para um modelo distribuído traria benefícios para que a ndnflix tenha controle de acesso eficaz, confiável e bem adaptado a NDN.

3.1.1. Esquema de nomeação

Para atender as necessidades da ndnflix descritas anteriormente, definimos o esquema de nomeação a seguir:

- **Produtor:**
  - Prefixo: `/ndnflix`
  - Dataset: `/ndnflix/brazil`
  - Conteúdo:
    - `/ndnflix/brazil/series/<nome da série>/<seq>`
    - `/ndnflix/brazil/filmes/<nome do filme>/<seq>`
- **Gerenciador de acesso:**
  - Prefixo: `/ndnflix/NAC/<dataset>`
  - Chaves:
    - `/<Prefixo>/KEK/<key-id>`
    - `/<Prefixo>/KDK/<key-id>/ENCRYPTED-BY/<ChaveAssinante>`

O *dataset* define a granularidade da chave, portanto, para definirmos controle de acesso por localidade, criamos as chaves e inserimos usuários sob sua hierarquia. Caso exista uma ndnflix nos Estados Unidos por exemplo, apenas usuários inscritos nessa localidade poderão consumir seu catálogo específico.

Para ser possível a adição de assinantes e também a sua remoção, definimos os seguintes prefixos adicionais:

- `/ndnflix/NAC/<dataset>/add_subscription/<ChaveAssinante>`
- `/ndnflix/NAC/<dataset>/revoke_subscription/<ChaveAssinante>`

Esses prefixos são locais e servem para recebimento de interesses de uma entidade interna responsável por notificar o gerenciador de acesso sobre a assinatura de novos clientes.

Já para realizar a sincronização dos gerenciadores com o SVS, precisamos adicionar mais dois prefixos:

- Prefixo de grupo: `/ndnflix/NAC/server`
- Prefixo de publicação: `/ndnflix/NAC/server/<NomeServidor>/<ChaveID>`

Todos os gerenciadores de acesso enviarão interesses periódicos para o prefixo do grupo em busca de novas chaves. Caso novas chaves tenham sido geradas, o resgate das mesmas será feito através do prefixo de publicação do servidor em específico.

Além dos prefixos definidos acima, cada usuário precisa disponibilizar suas chaves para que seja possível a distribuição da KDK, portanto, para os experimentos alguns prefixos de usuários foram definidos, como a seguir:

- **Consumidor:** `/<PrefixoProvedor>/<NomeUsuario>/KEY/<ChaveID>`

3.2. Desenvolvimento do protótipo D-NAC

O protótipo do D-NAC foi desenvolvido em C++, fazendo uso do próprio pipeline NDN e das APIs da biblioteca ndn-cxx, buscando integrar o NAC, o SVS e o NDVR de forma que conserve o máximo possível as implementações originais.

Durante o desenvolvimento, algumas premissas são assumidas, a saber: i) o produtor de conteúdo conhece o prefixo do gerenciador de acesso e vice-versa; ii) existe um processo confiável de distribuição de chaves entre os gerenciadores que ocorre previamente à inicialização do D-NAC. Já que todos os gerenciadores pertencem à mesma organização, é razoável assumir tais premissas, que juntas compõem o modelo de confiança, onde o conteúdo produzido é assinado pela chave de cada gerenciador e validado pelos outros gerenciadores através do *trust anchor* estabelecido.

O *bootstrap* do modelo de confiança do D-NAC inclui também uma troca de chaves simétrica entre os gerenciadores de acesso, conforme ilustrado na Figura 1. Essa chave funcionará como a senha mestra do chaveiro compartilhado. Para compreender a necessidade de sua existência, precisamos lembrar de três fatores importantes:

1. A KDKs são chaves privadas que não podem ser compartilhadas em texto claro na rede sem comprometer a confidencialidade;
2. Não é possível, pela própria definição de chaves criptográficas assimétricas, deduzir uma KDK a partir de uma KEK;
3. A requisição de adição e remoção de membros pode ocorrer em qualquer nó gerenciador da rede.

Portanto, para conseguirmos adicionar novos membros não apenas no nó que originalmente gerou a KEK, mas em todos os gerenciadores de acesso da rede, precisamos importar a KDK de forma legível para todos os chaveiros. Considerando que a KDK gerada especificamente para um assinante não pode ser lida pelo nó gerenciador, precisamos que a KDK seja criptografada por uma chave comum entre eles.

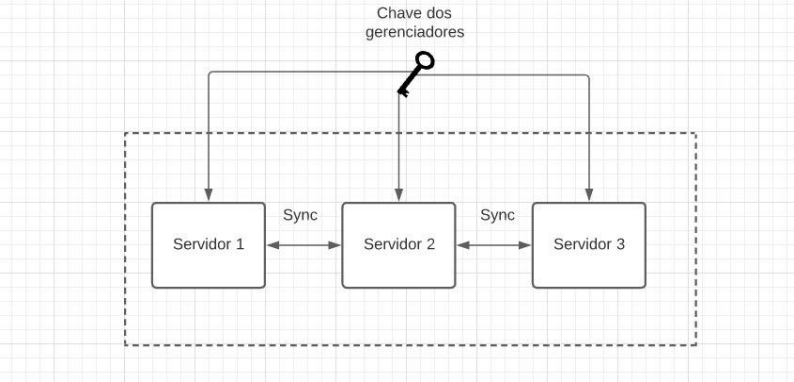

Figura 1. Pré-configuração dos gerenciadores

Na implementação do D-NAC definimos quatro classes: *Manager*, *Consumer*, *Producer* e *Operator*.

A classe *Manager* serve de interface para a *Access Manager*, classe original do NAC, definindo o *dataset* para as chaves que irão ser produzidas. Nessa classe é adicionada preliminarmente a rota para requisição de KEKs pelo produtor, a chave HMAC para os interesses de sincronização entre os gerenciadores, o prefixo de sincronização de grupo e o prefixo de publicação do servidor.

A modelagem original do NAC para recuperação da KEK se dá como na Figura 2. Neste fluxo, no momento que o gerenciador de acesso recebe interesse de um produtor, ocorre imediatamente a produção do par de chaves, o envio da KEK para o produtor e a disponibilização da KDK correspondente para adição de assinantes.

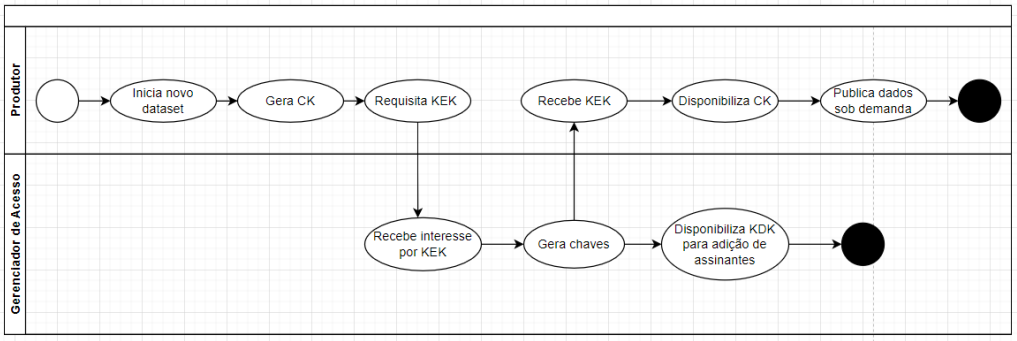

Figura 2. Fluxo de recuperação da KEK no NAC

No D-NAC, tendo em vista sua proposta distribuída, o fluxo de requisição da KEK sofreu modificações. Podemos observar na Figura 3 que no momento em que o gerenciador começa sua execução, já se inicia o envio de interesses para o prefixo de grupo dos gerenciadores, com o objetivo de realizar a inicialização no estado correto. Caso já existam chaves publicadas por outros gerenciadores, sua recuperação é feita imediatamente.

Assim que um gerenciador do grupo recebe um interesse de um produtor por uma KEK, é realizada uma checagem no chaveiro compartilhado, verificando se a KEK para esse *dataset* já existe advinda de outro gerenciador. Caso exista, o gerenciador responderá ao interesse com a KEK correspondente. Em caso negativo, um novo par de chaves é

gerado, respondendo ao produtor com a KEK requerida e realizando a publicação das chaves no grupo de sincronização.

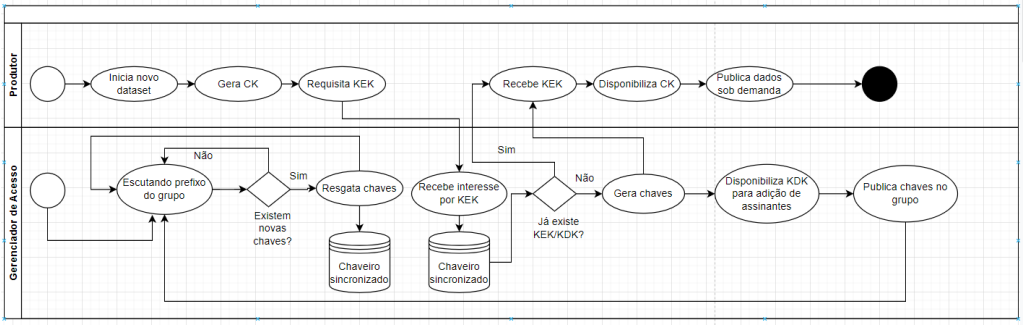

Figura 3. Fluxo de recuperação da KEK no D-NAC

Na implementação atual, a adição de novos membros se dá como na Figura 4. O Departamento Comercial, implementado na classe *Operator*, funciona como um autorizador interno, informando ao gerenciador de acesso quem são os usuários com assinatura ativa na ndnflix, bem como quem são os usuários que precisam ter seus direitos suspensos. Usuários que tivessem interesse de adquirir o serviço da ndnflix entrariam em contato com esta entidade para realizar as transações financeiras necessárias.

Mediante pedido de adição de assinante, o primeiro passo consiste no gerenciador de acesso obter a chave pública do consumidor, seja pelo cache oportunístico da NDN, por um *data repository* ou pelo fluxo normal de satisfação de interesse. Como alternativa a esse modelo de comunicação direta com o consumidor para obtenção do certificado, que exigiria esquema de nomeação e alcançabilidade dos consumidores, pode-se adotar também um esquema de delegação onde um conjunto de consumidores hospeda suas chaves públicas em determinado produtor autorizado [Tehrani et al. 2022]. Neste trabalho, assume-se que os assinantes possuem alcançabilidade para recuperação da chave pública.

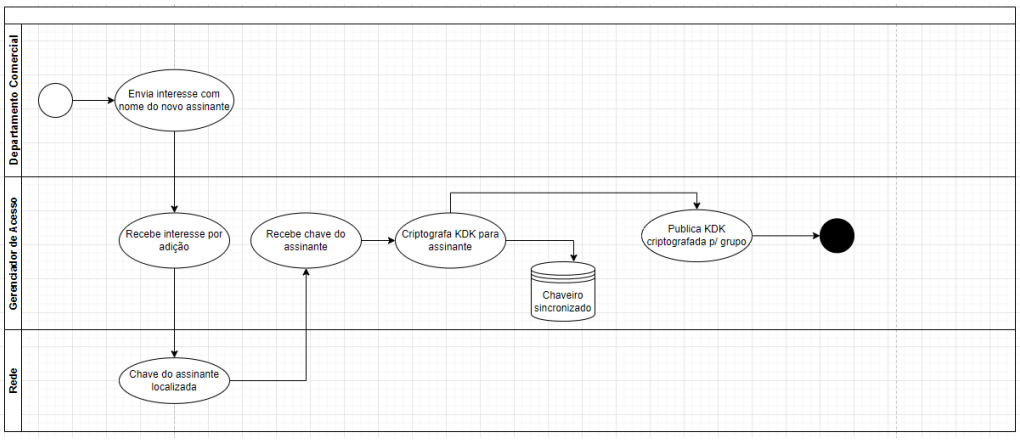

Figura 4. Fluxo de adição de assinantes

O consumo de conteúdo no NAC se dá em duas fases. De início têm-se a fase *slowpath*, quando a recuperação da KDK é necessária, ou seja momentos de criação ou renovação de chaves, nos quais ocorre mudança de CKs, KEKs e KDKs. Em contrapartida, a fase *fastpath* ocorre após a troca de chaves entre as partes envolvidas. Nesta fase,

a comunicação se dá exclusivamente entre o produtor de conteúdo e os consumidores habilitados, não havendo envolvimento do gerenciador de acesso.

A representação do consumo do conteúdo por clientes com o D-NAC é demonstrada na Figura 5. Como as KDKs criptografadas para os novos membros também são publicadas para todos os gerenciadores de acesso da rede, checamos o chaveiro sincronizado entre as instâncias em busca da KDK desejada. A existência desse chaveiro é o que torna possível que o usuário faça o resgate da KDK do gerenciador mais próximo de si, proporcionando uma possível melhora de desempenho em momentos de *slowpath*.

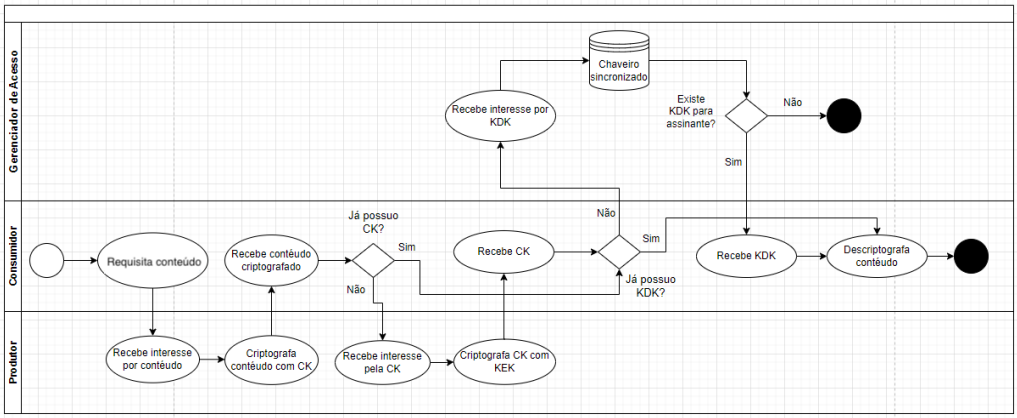

Figura 5. Fluxo de requisição da KDK no D-NAC

A classe *Consumer* instancia a classe *Decryptor* original do NAC, passando seu par de chaves para decifragem das KDKs recebidas. Já a classe *Producer* instancia a classe *Encryptor*, informando o prefixo do gerenciador de acesso e o prefixo da CK que será gerada. A única acomodação necessária no SVS foi a adição de um parâmetro adicional na função de *publishData*, o nome das chaves. Possuindo o nome das chaves, os gerenciadores que receberem os dados sincronizados conseguem configurar seus chaveiros corretamente.

4. Avaliação

A avaliação da proposta foi realizada de duas formas: prova de funcionamento e avaliação de desempenho. Na prova de funcionamento o objetivo é avaliar o funcionamento da proposta a partir de uma topologia simples e do caso de uso descrito previamente. Já a avaliação de desempenho apresenta os resultados de um conjunto de experimentos que comparam métricas de qualidade e sobrecarga entre a solução original de NAC e a proposta D-NAC.

4.1. Prova de funcionamento

Para validar o funcionamento do D-NAC foi considerado o caso de uso descrito anteriormente, ndnflix, bem como uma topologia simplificada apresentada a seguir e cenários de falha. Na Figura 6 temos a topologia de exemplo definida para execução dos testes. O *delay* entre nó BA (Bahia) e o nó SP (São Paulo) foi configurado para 14 milissegundos com base nos valores reais coletados no *backbone* da RNP (Rede Nacional de Ensino e Pesquisa).

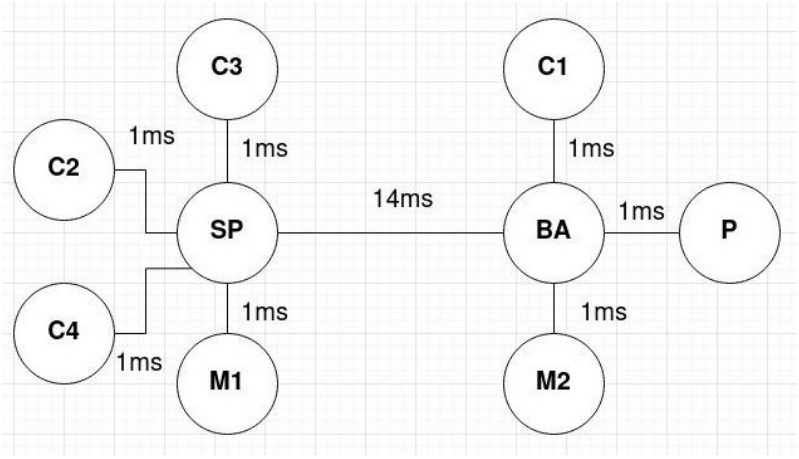

Figura 6. Topologia utilizada nos experimentos

Os nós representados na topologia seguem as definições a seguir: **[M1, M2]**: Gerenciadores de acesso; **[C1, C2, C3, C4]**: Assinantes da ndnflix; **P**: Produtor de conteúdo da ndnflix; **[BA, SP]**: Roteadores da região. Além disso, foram definidos prefixos de assinantes, a saber: **C1** - `/org1/usuarioC1`, **C2** - `/org2/usuarioC2`, **C3** - `/org3/sp/usuarioC3`, **C4** - `/org4/usuarioC4`.

Durante a inicialização do gerenciador M1, interesses periódicos em busca de outros gerenciadores são enviados na rede (fase de descoberta de gerenciadores). Ao iniciarmos o produtor P, ocorrerá a requisição da KEK para o gerenciador mais próximo, portanto o nó M2 será utilizado conforme topologia proposta. O gerenciador M2 faz a criação das novas chaves KEK e KDK, e responde ao interesse do produtor com a KEK. Essa chave também é compartilhada no grupo de sincronização através da função *publishData*. Em seguida a KDK é exportada, cifrada com a chave dos gerenciadores e igualmente compartilhada entre as entidades sincronizadas.

Em seguida, o M1, ao detectar divergências de estado, requisita as novas chaves com a função *fetchData* e as importa para seu chaveiro. Ao receber o interesse para inclusão de assinantes, o M2 requisita as chaves desses novos consumidores, criptografa a KDK com chaves dos consumidores e compartilha igualmente com o grupo de sincronização.

No momento em que o consumidor C1 tenta recuperar a KDK para decifrar o conteúdo recebido pelo produtor, o nó M2 responde ao consumidor, devido a sua proximidade, ao passo que consumidores como C2, C3, e C4, que se encontram na região de São Paulo, tem seus interesses satisfeitos pelo gerenciador M1.

4.1.1. Cenário de falha

Com o propósito de validar o funcionamento do D-NAC em caso de falha de algum dos gerenciadores, um *script* para automatização de falhas durante a execução foi criado. Os cenários testados foram: a) Falha no M2 logo após adição dos assinantes; b) Falha no M1 logo após adição dos assinantes.

A fim de simular a falha do nó M2, cinco segundos após o gerenciador ter cri-

ado KDKs para assinantes, uma falha é introduzida através do encerramento do NFD e do NDVR, o que indisponibiliza as chaves criadas no nó. Porém, graças a sincronização do D-NAC, todos os consumidores obtiveram sucesso no resgate da KDK, visto a possibilidade de requisitar para o nó M1. Em particular, esse comportamento é alcançado graças ao mecanismo de sincronização das chaves presentes no D-NAC, além do roteamento dinâmico provido pelo NDVR, que garante que informações de alcançabilidade com múltiplas rotas para determinado prefixo já são instaladas previamente nos nós.

Observa-se uma situação análoga quando o nó M1 é inesperadamente desconectado da rede. Nesse caso, os clientes são re-roteados para o gerenciador M2, que passa a responder tanto para interesses de KDK, quanto para criação de novos pares de chaves para produtores e consumidores.

4.2. Análise comparativa entre NAC e o D-NAC

Os testes foram realizados em um ambiente emulado, através do software MiniNDN (v0.5.0), executando em uma máquina virtual com 4G de RAM e 4 *cores*. A mesma topologia apresentada anteriormente foi utilizada. Todos os experimentos foram analisados a partir do intervalo de confiança, da média e do desvio padrão aferidos, utilizando distribuição *t-student* com replicações de 10 execuções por experimento e $\alpha = 0,05$. As métricas utilizadas foram:

- **Data Retrieval Delay (ms)**: Atraso para recuperação dos dados.
- **Overhead**: Quantidade total de pacotes de interesses e de dados enviados e recebidos referentes a cada implementação.

O *Data Retrieval Delay* é uma métrica que avalia se a abordagem distribuída do D-NAC se traduz em ganho de desempenho, levando em consideração que um dos seus objetivos principais é reduzir o tempo gasto para recuperação da KDK. Esta diferença no tempo de recuperação da KDK impacta diretamente no atraso de recuperação de dados em momentos de *slowpath*. Já o *Overhead* é uma métrica importante para analisarmos quais são os impactos a nível de consumo de banda da rede, tendo em vista que a adição da funcionalidade de sincronização pode impactar na quantidade de pacotes transmitidos.

Os experimentos conduzidos tiveram duração de 60 segundos, durante os quais cada consumidor envia um interesse para o produtor da ndnflix a cada dois segundos. O atraso para obtenção do dados foi medido em todos os nós consumidores, sendo feita uma média do atraso para todos os consumidores. Já o *overhead* foi medido em todos os nós da topologia para termos um visão geral da sobrecarga na rede, considerando a movimentação de todos os pacotes de interesses e de dados.

Para análise dos resultados é importante notarmos que por ser um experimento de curta duração, a recuperação da KDK acontece apenas no momento da obtenção do primeiro pacote de dados, ocorrendo comunicação apenas entre o consumidor e o produtor após essa etapa.

Os gráficos obtidos corroboram a expectativa de que na existência de múltiplos gerenciadores conseguimos uma melhora na velocidade de obtenção dos dados em momentos de *slowpath*. Comparando a Figura 7 e a Figura 8 podemos ver que o atraso é significativamente menor quanto utilizamos mais de um gerenciador de acesso. Com o

NAC, o atraso médio na obtenção do primeiro pacote de dados, considerando-se o intervalo de confiança, fica na casa dos 284 a 407 milissegundos, enquanto com o D-NAC temos atrasos na faixa dos 135 a 208 milissegundos.

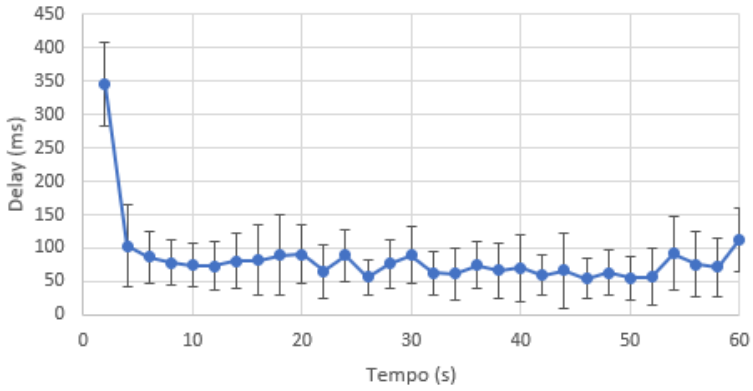

Figura 7. Atraso na recuperação de dados no NAC

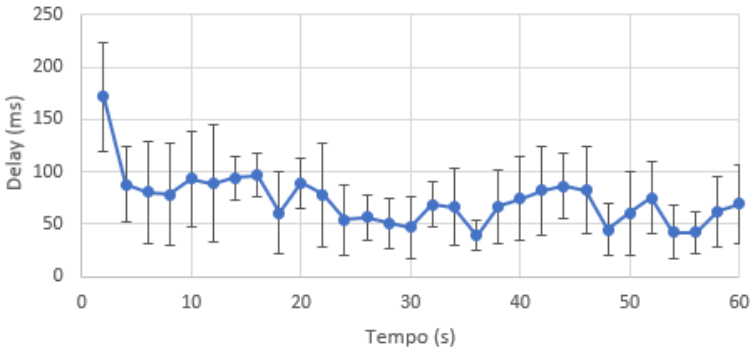

Figura 8. Atraso na recuperação de dados no D-NAC

Na Figura 9 podemos ver o gráfico de comparação do *overhead* na rede. Através dele é possível constatar, como esperado, que por conta da utilização da sincronização no D-NAC, temos uma sobrecarga adicional, acarretada principalmente pelo envio de interesses entre os gerenciadores de acesso. É importante ressaltar que os interesses de sincronização enviados pelo SVS só se convertem em respostas de pacotes de dados no momento que há divergência no vetor de estado entre as entidades sincronizadas. Em função disso, não vemos diferenças significativas na quantidade de pacotes de dados transmitidos entre o NAC e o D-NAC. Essa quantidade adicional de interesses transmitidos pode ser vista como de baixo impacto principalmente quando levamos em consideração que os pacotes de interesse são consideravelmente menores que pacotes de dados, representando uma diferença pequena quando falamos de *bits* transmitidos na rede. Então, considerando os dados analisados, o D-NAC não se mostra uma solução onerosa frente aos benefícios adquiridos.

5. Conclusões e Trabalhos Futuros

Este artigo apresentou o *design* e prototipagem do D-NAC, uma solução distribuída para controle de acesso de múltiplos consumidores na arquitetura NDN. A partir do aprimoramento da solução existente de NAC, foi possível tornar a solução mais eficiente

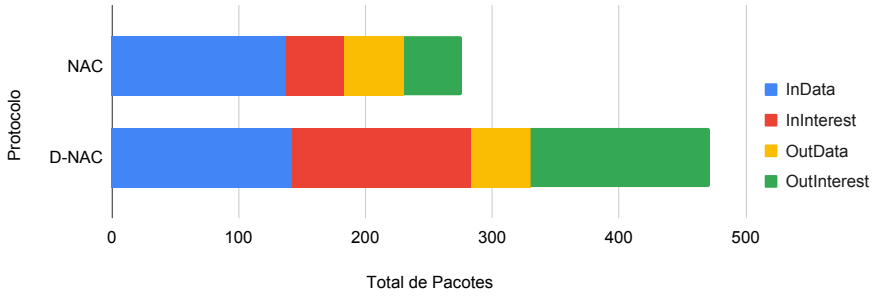

Figura 9. Comparação de sobrecarga na rede entre o D-NAC e NAC

e resiliente a falhas. A proposta descrita neste artigo traz contribuições na modelagem da solução, com inclusão de novos componentes, como os gerenciadores de acesso distribuídos, customização no esquema de nomeação e integração de protocolos existentes na NDN, como o protocolo de sincronização SVS e de roteamento NDVR, bem como na avaliação da solução proposta e na prototipagem de um caso de uso do D-NAC com um serviço de *streaming* na NDN. O *design* do D-NAC, seguindo os padrões de projeto, APIs e modelo de comunicação baseado em pacotes de interesse/dados da NDN, mostrou-se uma oportunidade de aprendizagem profunda da arquitetura.

Os resultados do D-NAC apresentam vantagens em relação ao NAC, tendo em vista que o atraso para recuperação de dados foi consideravelmente menor que na solução original. Ademais, experimentos com cenários de falha demonstraram que mesmo que o gerenciador que gerou as chaves fique indisponível, os consumidores ainda conseguem recuperá-las com sucesso. Da mesma forma, é possível adicionar ou remover assinantes, não ocorrendo interrupção do serviço. Notou-se aumento na quantidade da pacotes transitados na rede, porém dentro de limites já esperados tendo em vista o uso de interesses decorrentes para a sincronização, bem como o envio adicional das chaves para o grupo de sincronização. Contudo, esse impacto se mostrou pequeno para *overhead* total da rede em relação aos ganhos obtidos.

Em trabalhos futuros espera-se investigar o uso de outros protocolos de roteamento e sincronização, avaliando não apenas a diferença de desempenho mas também o desacoplamento entre design do D-NAC e os componentes de serviço. Outro aspecto a ser explorado é a avaliação de desempenho, considerando novas métricas como uso efetivo do cache, cenários mais complexos de falhas e com múltiplos eventos de troca de chave.

Agradecimentos

Os autores agradecem o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), da Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB) e *Air Force Office of Scientific Research* (*award number* FA9550-23-1-0631).

Referências

Ben Abraham, H. and Crowley, P. (2013). Performance measurement of the ccnx synchronization protocol. In *Architectures for Networking and Communications Systems*, pages 121–122.

Brito, I. and Sampaio, L. (2021). Roteamento em Redes de Dados Nomeados com NDVR: um protocolo leve e eficiente para disseminação de informações de alcançabilidade. In *Anais do XXXIX SBRC*, pages 574–587, Porto Alegre, RS, Brasil. SBC.

Fu, W., Abraham, H. B., and Crowley, P. (2015). Synchronizing namespaces with invertible bloom filters. In *2015 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, pages 123–134.

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

Hoque, A. K. M. M., Amin, S. O., Alyyan, A., Zhang, B., Zhang, L., and Wang, L. (2013). Nlsr: Named-data link state routing protocol. ICN '13, page 15–20, New York, NY, USA. Association for Computing Machinery.

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

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

Liang, T., Huang, W., Ma, X., Zhang, W., Zhang, Y., and Zhang, B. (2023). PCLive: Bringing Named Data Networking to Internet Livestreaming. In *Proceedings of the 10th ACM Conference on Information-Centric Networking*, page 36–45, New York, NY, USA. Association for Computing Machinery.

Nour, B., Khelifi, H., Hussain, R., Mastorakis, S., and Moungla, H. (2021). Access control mechanisms in named data networks: A comprehensive survey. *ACM Comput. Surv.*, 54(3).

Philipp Moll, Varun Patil, N. S. and Zhang, L. (2021). A Brief Introduction to State Vector Sync. *NDN Technical Report*.

Sampaio, L., Freitas, A., Brito, I., Araújo, F., and Ribeiro, A. (2021). Revisitando as icns: Mobilidade, segurança e aplicações distribuídas através das redes de dados nomeados. *Minicursos do Simpósio Brasileiro de Redes de Computadores - SBRC*.

Shang, W., Afanasyev, A., and Zhang, L. (2017). Vectorsync: Distributed dataset synchronization over named data networking. page 192–193. Association for Computing Machinery.

Tehrani, P. F., Osterweil, E., Schmidt, T. C., and Wählisch, M. (2022). Sok: Public key and namespace management in ndn. In *Proceedings of the 9th ACM Conference on Information-Centric Networking*, ICN '22, page 67–79, New York, NY, USA. Association for Computing Machinery.

Thelagathoti, R. K., Mastorakis, S., Shah, A., Bedi, H., and Shannigrahi, S. (2020). Named Data Networking for Content Delivery Network Workflows. In *2020 IEEE 9th International Conference on Cloud Networking (CloudNet)*, pages 1–7.

Zhang, L. (2019). The Role of Data Repositories in Named Data Networking. In *2019 IEEE International Conference on Communications Workshops*, pages 1–5.

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, Z., Yu, Y., Ramani, S. K., Afanasyev, A., and Zhang, L. (2018). NAC: Automating Access Control via Named Data. In *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, pages 626–633.