



**CENTRO UNIVERSITÁRIO RUY BARBOSA  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**IGOR SANTOS DE SOUZA**

**DESAFIO DA ARQUITETURA NAMED DATA NETWORKING EM  
RELAÇÃO A PROTEÇÃO DE CONTEÚDO E PRIVACIDADE:  
ATAQUE DE TEMPORIZAÇÃO EM CACHE**

**SALVADOR  
2019**

**IGOR SANTOS DE SOUZA**

**DESAFIO DA ARQUITETURA NAMED DATA NETWORKING EM  
RELAÇÃO A PROTEÇÃO DE CONTEÚDO E PRIVACIDADE:  
ATAQUE DE TEMPORIZAÇÃO EM CACHE**

TCC apresentado ao Centro Universitário Ruy  
Barbosa como requisito essencial para  
aprovação no curso de Bacharelado em  
Ciência da Computação.

Orientador: Prof. Francisco José Badaró  
Valente Neto.

**SALVADOR  
2019**

**IGOR SANTOS DE SOUZA**

**DESAFIO DA ARQUITETURA NAMED DATA NETWORKING EM RELAÇÃO A  
PROTEÇÃO DE CONTEÚDO E PRIVACIDADE: ATAQUE DE TEMPORIZAÇÃO  
EM CACHE**

Trabalho de Conclusão de Curso apresentado  
ao Centro Universitário Ruy Barbosa, como  
requisito parcial para obtenção do título de  
bacharel em Ciência da Computação, sob  
orientação do Prof.º Francisco José Badaró  
Valente Neto

Aprovado em \_\_\_\_/\_\_\_\_/\_\_\_\_.

---

Professor Francisco José Badaró Valente Neto - Orientador

---

Professor Italo Valcy da Silva Brito – UFBA/RNP

---

Professor Ibirisol Fontes Ferreira – UFBA/RNP

---

Professor Alex da Cruz Cerqueira – FACULDADE SÃO SALVADOR

## RESUMO

O crescimento exponencial do uso das redes de computadores (na sua maior parte a Internet) vem, a cada dia que se passa, expondo algumas fragilidades na forma atual da sua arquitetura. Um dos pontos mais afetado, e consequentemente o mais perceptível ao usuário final é a segurança das informações e a performance. O fato do conjunto de protocolos que formam o TCP/IP (centro da atual arquitetura da Internet) trabalhar nomeando *hosts* é o que torna a Internet atual insegura e com limitações de performance, porque nesse modelo o que é “protegido” é o canal por onde a informação vai transitar, e não a informação diretamente e a orientação do processo de roteamento é sobre o host e não sobre o conteúdo. A arquitetura *Named Data Networking (NDN)* se apresenta como uma nova forma de conduzir a Internet, alterando o paradigma de comunicação expondo uma nova abordagem tendo como base o conteúdo, independentemente de sua localização. A proposta da arquitetura *NDN* não é nomear os *hosts*, e sim o dado que será requisitado e enviado. Com o dado nomeado, a comunicação será centrada ao conteúdo e não centrada em *hosts*. Seguindo essa premissa, a proposta de segurança na arquitetura *NDN* é de proteger o arquivo diretamente, desde quando ele sair do remetente até chegar ao destinatário, assim o destinatário poderá ter a certeza de que o dado chegado foi realmente o dado solicitado. Apesar da performance não ser um dos princípios da arquitetura, a inteligência aplicada ao plano de dados com os múltiplos *next-hops* sobre os dados nomeados associado ao sistema de cache e gerência de interesse e conteúdo, confere melhor performance na entrega de dados e também melhor resiliência a congestionamentos, face a atual arquitetura da Internet. O objetivo desse trabalho é apresentar a arquitetura *NDN*, seus aspectos e princípios de design e protocolos através de uma pesquisa exploratória com uma revisão bibliográfica e relacionar problemas de segurança e de performance em pesquisa exploratória na literatura, apresentar dois simuladores da arquitetura e demonstrar o seu funcionamento, contribuindo assim para o conhecimento dessa arquitetura, para que se tenha cada vez uma Internet melhor.

**Palavras-Chave:** NDN; ICN; Redes Orientada ao Conteúdo; Segurança; Mini-NDN; Cache.

## LISTA DE FIGURAS

Figura 1 - Content Distribution Network .....	11
Figura 2 – IP x ICN .....	15
Figura 3 - Internet atual x NDN .....	17
Figura 4 - Elementos arquitetura NDN.....	21
Figura 5 - Formato dos pacotes NDN.....	21
Figura 6 - Entidades e Âncoras em NDN .....	24
Figura 7 - Estrutura ndnSIM.....	37
Figura 8 - Status do testbed NDN.....	42
Figura 9 - Testbed NDN em forma de mapa mundi .....	43
Figura 10 - Web cache.....	44
Figura 11 – Comunicação Cache.....	46
Figura 12 - Topologia NDN .....	48
Figura 13 - Diagrama descritivo do ataque relatado.....	49
Figura 14 - Rede LAN.....	50
Figura 15 - Rede WAN.....	50
Figura 16 - Topologia WAN .....	51

## **LISTA DE SIGLAS E ABREVIATURAS**

AP	Acess Point
ARPANET	Advanced Research Projects Agency Network
BNDS	Banco Nacional de Desenvolvimento Econômico
CCN	Content Centric Networking
CDN	Content Distribution Networking
CLI	Comand Line Interface
CNI	Confederação Nacional da Indústria
CPU	Central Process Unit
CS	Content Store
CSMA	Carrier Sense Multiple Access
DNS	Domain Name System
DNSSEC	Domain Name System Secury
DPA	Data Protection Authorities
DTLS	Datagram Transport Layer Security
DTN	Delay-Tolerant Network
EMBRAPII	Empresa Brasileira de Pesquisa e Inovação Industrial
EUA	Estados Unidos da América
FIB	Forwarding Information Base
GDPR	General Data Protection Regulation
GPL	General Public License

GRE	Generic Routing Encapsulation
GTP	GPRS Tunneling Protocol
HMAC	Hash Based Message Authentication Code
ICN	Information Centric Networking
ICO	Information Commissioner's Office
IETF	Internet Engineering Task Force
IoT	Internet of Things
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IPsec	Internet Protocol Security
IPX	Internetwork Packet Exchange
L2F	Layer 2 Forwarding
L2TP	Layer 2 Tunnelling Protocol
LTS	Long Term Support
NAT	Network Address Translation
NDN	Named Data Networking
ndnSIM	Named Data Networking Simulator
NetBEUI	Extended User Interface
NetEm	Network Emulator
NETINF	Network of Information
NFD	Network Forwarding Daemon
NLSR	Named Data Link State Routing
NS-3	Network Simulator 3

NSF	National Science Foundation
OIC	Organisation of Islamic Cooperation
PARC	Palo Alto Research Center
PC	Personal Computer
PGP	Pretty Good Privacy
PIT	Pending Interest Table
PoP	Point of Presence
PPTP	Point to Point Tunneling Protocol
ROC	Rede Orientada ao Conteúdo
RTT	Round Trip Time
SDSI	Segurança Distribuída Simples
SENAI	Serviço Nacional de Aprendizagem Industrial
SSH	Secure Shell
SSL/TLS	Secure Sockets Layer/Transport Layer Security
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TOR	The Onion Router
TTFB	Time to First Byte
UDP	User Datagram Protocol
UE	União Europeia
UFPA	Universidade Federal do Pará
VM	Virtual Machine
VoIP	Voice Over Internet Protocol
WAN	Wide Area Network



## SUMÁRIO

1. INTRODUÇÃO.....	6
2. FUNDAMENTAÇÃO TEÓRICA .....	10
2.1. PROTOCOLO TCP/IP .....	10
2.1.1. SEGURANÇA EM TCP/IP .....	11
2.2. REDES ORIENTADAS AO CONTEÚDO .....	13
2.2.1. DATA-CENTRIC SECURITY .....	16
2.3. INTRODUÇÃO A NAMED DATA NETWORKING .....	16
2.3.1. COMUNICAÇÃO EM NDN .....	20
2.3.2. FUNCIONAMENTO SEGURANÇA EM NDN.....	22
2.3.3. PROBLEMAS .....	27
2.3.4. GDPR .....	29
2.3.5. GDPR (PRIVACIDADE) x NDN/CCN .....	31
2.4. SIMULADORES E EMULADORES .....	33
2.4.1. MINI-NDN.....	35
2.4.3 ndnSIM.....	35
2.5. TESTBED .....	38
2.6. CACHE.....	43
2.6.1 SEGURANÇA EM CACHE.....	45
2.6.2. ATAQUE DE TEMPORIZAÇÃO EM CACHE .....	47
2.6.3 ALTERNATIVAS PARA O ATAQUE APRESENTADO .....	51
3. PROCEDIMENTOS METODOLÓGICOS .....	57
4. CONCLUSÃO.....	58
5. REFERÊNCIAS .....	60

## 1. INTRODUÇÃO

Embora a Internet tenha superado em muito as expectativas (fornecer comunicação entre dois *hosts*), ela também ampliou as premissas iniciais, muitas vezes, criando disputas que desafiam seu modelo de comunicação subjacente. Usuários e aplicativos operam em termos de conteúdo, tornando-se cada vez mais limitado e difícil de se adequar ao requisito do IP para se comunicar, descobrindo e especificando o local. Para levar a Internet para o futuro, é necessária uma mudança de arquitetura conceitualmente simples, porém transformadora, desde o foco de hoje em que – orientação a endereços e *hosts*, que compõe o modelo padrão atual - até a orientação ao conteúdo que os usuários e aplicativos se interessam, compondo assim o modelo de arquitetura para a próxima geração (NAMED, S/Da).

No início da Internet, a telefonia foi tida como o único exemplo de comunicações bem-sucedidas e eficazes em escala global (NAMED, S/Da). Na época, a solução oferecida pelo TCP/IP era de fornecer uma comunicação ponto-a-ponto entre dois *hosts*, mas acabou sendo utilizado para outro fim, como distribuição de conteúdo e, como hoje em dia praticamente tudo é conectado à Internet, essa distribuição é feita tanto em *hosts* fixos quanto em *hosts* móveis. Tendo isso em vista, pode-se chegar à conclusão que essa atual arquitetura da Internet é uma má combinação (comunicação juntamente com distribuição de conteúdo), além do que, essa atual arquitetura vem abrindo espaço para ataques maliciosos que buscam tirar proveito do valor de informações que circulam pela Internet. Muito vem sendo feito em relação à segurança, mas dia a dia os ataques continuam a aumentar. É muito importante que uma arquitetura de próxima geração, além das mudanças de paradigma necessárias à evolução, também verse e proponha soluções para os problemas de segurança.

Com o passar do tempo foi observado que esse modelo de comunicação (baseado em *hosts*) poderia ser ineficiente, vista às necessidades dos usuários da Internet atual. Surge então um novo conceito de comunicação, a CCN (*Content Centric Networking*).

Rede centrada em informações (ICN) é uma abordagem para desenvolver a infraestrutura da Internet para suportar diretamente comunicações centradas em dados e independentes de localização, introduzindo dados com nomes exclusivos como um princípio central de comunicações. O acesso a dados se torna independente de localização, aplicativo e armazenamento, permitindo o armazenamento em cache da rede e a mobilidade sem âncora. Os benefícios esperados são maior eficiência, melhor escalabilidade com relação à demanda de informações/largura de banda e melhor robustez em cenários de comunicação desafiadores.

Esses conceitos são conhecidos em termos diferentes, incluindo, entre outros: Rede Centrada em Conteúdo (CCN), Rede de Dados Nomeados (NDN), Rede de Informações (Netinf) e Rede de Publicação/Assinatura (PSIRP) (CICN..., 2018).

A Rede de Dados Nomeados (NDN) é um dos cinco projetos de pesquisa financiados pela *National Science Foundation* dos EUA, no âmbito do Programa de Arquitetura da Internet do Futuro. A NDN tem suas raízes em um projeto anterior, CCN, que Van Jacobson iniciou na *Xerox PARC* na época de sua palestra no *Google*, para transformar sua visão de arquitetura em um protótipo em execução (NDN..., S/D).

Apesar de existir muitos termos diferentes para o conceito de rede centrada a conteúdo ele não significam a mesma coisa, o projeto que está mais firmado no que se diz respeito à progresso é o *NDN*, por conta disso pode ser que as pessoas, quando estudarem esse assunto, pensem que essas siglas significam basicamente a mesma coisa.

O CCN refere-se ao projeto de arquitetura iniciado por Van Jacobson no PARC, que incluiu o desenvolvimento de uma base de código de *software* que representa uma implementação de linha de base dessa arquitetura. Rede de Dados Nomeados (NDN) refere-se ao projeto de arquitetura da Internet do futuro financiado pela NSF, uma colaboração de 12 campus que começou em 2010 e incluiu o PARC. O projeto NDN usou originalmente o CCNx como sua base de código, mas a partir de 2013 foi bifurcada uma versão para suportar as necessidades especificamente relacionadas à pesquisa e desenvolvimento de arquitetura financiada pela NSF (e não necessariamente de interesse para o *PARC*) (NDN..., S/D).

Segundo Muhammad Najib, a pesquisa feita por ele no artigo *Performance Comparison of Named Data Networking and IP-based Networking in Palapa Ring Network* compara o desempenho de redes baseadas em NDN e IP de taxa de transferência, atraso de pacotes e queda de pacotes que são calculados e medidos seus parâmetros importantes. O principal aspecto da análise e avaliação da pesquisa feita por ele é a estratégia de roteamento, largura de banda e armazenamento de conteúdo. A diferença de NDN é que ela usa uma rede baseada em CS (*content store*) e IP que não possui CS (*content store*) terá seu desempenho afetado. Com base nos dados apresentados na pesquisa de Muhammad Najib, a conclusão é que o NDN tem melhor desempenho do que a rede baseada em IP. Isso pode ser visto do valor de atraso que o NDN tem um atraso menor que o IP devido à existência de CS (*content store*). Ainda segundo Muhammad Najib, na taxa de transferência, o NDN também é melhor que o IP. Para o parâmetro de descarte de pacotes, a largura de banda usada e o alto tráfego estão afetando o desempenho do NDN e do IP (SATRIA; ILMA; SYAMBAS, 2017).

A segurança é um assunto abrangente e inclui inúmeros tipos de problemas. Em sua forma mais simples, preocupa-se em impedir que pessoas mal-intencionadas leiam ou, pior ainda, modifiquem secretamente mensagens enviadas a outros destinatários. Outra preocupação da segurança são as pessoas que tentam ter acesso a serviços remotos que não estão autorizadas a usar. Um dos problemas de segurança das redes é o sigilo. O sigilo — também chamado confidencialidade — está relacionado ao fato de manter as informações longe de usuários não autorizados. É isso que costuma vir à mente quando se pensa em segurança de redes. Em geral, a autenticação cuida do processo de determinar com quem você está se comunicando antes de revelar informações sigilosas ou entrar em uma transação comercial. (TANNEBAUM, 2011).

Mais de 2,6 bilhões de dados foram roubados, perdidos ou expostos mundialmente em 2017, aumento de 88% em relação a 2016. Os dados são de levantamento da Gemalto (2017). O estudo indica que enquanto os incidentes de violação de dados diminuíram 11%, 2017 foi o primeiro ano de divulgação pública em que as violações superaram mais de 2 bilhões de registros de dados comprometidos desde que o *Breach Level Index* começou a rastrear as violações de dados em 2013 (BREACH, 2018). O levantamento indica que durante os últimos cinco anos, quase 10 bilhões de registros foram perdidos, roubados ou expostos, com uma média de 5 milhões de registros comprometidos a cada dia. Dos 1.765 incidentes de violação de dados em 2017, a fraude à identidade representou o principal tipo de violação de dados, contabilizando 69% de todas as violações (REDAÇÃO, 2018).

Considerando o exposto, questiona-se: A arquitetura *Named Data Networking* (NDN) que ainda está em estudo tem alguma falha de segurança? Como essa possível falha pode ser resolvida? E as questões de performance? Como são tratadas? Há alguma melhoria?

Os objetivos gerais deste trabalho são analisar a *arquitetura Named Data Networking* (NDN), através de uma pesquisa exploratória com uma revisão bibliográfica, apresentar a arquitetura, os princípios de design e protocolos, fazer o comparativo de performance com arquitetura atual e relacionar falhas de segurança na mesma e apresentar contribuições no contexto de segurança publicadas na literatura em seu estado da arte. Também é objetivo deste trabalho a apresentação e comparativo de dois simuladores da arquitetura de NDN e uma simulação nos mesmos demonstrando a arquitetura em funcionamento. Para finalizar, a conceituação e apresentação de plataformas de testes e simulações em rede, através de federação de recursos/simuladores distribuídos em rede, os *testbeds*.

Essa pesquisa terá como objetivos específicos os seguintes:

- Estudar o princípio de Redes de Computadores e a atual arquitetura da Internet com foco principalmente nos aspectos de roteamento, entrega de conteúdo, segurança e performance.
- Estudar a arquitetura *Named Data Networking* (NDN) e suas características, princípios de design e protocolos.
- Estudar a proposta de segurança e de performance da arquitetura *Named Data Networking* (NDN).
- Relacionar falhas de segurança da arquitetura *Named Data Networking* (NDN) e apresentar as soluções propostas na literatura em seu estado da arte.
- Realizar um estudo comparativo entre um simulador e um emulador da arquitetura *Named Data Networking* (NDN) e efetuar testes para demonstrar o funcionamento do simulador e do emulador com o objetivo de, em ambos, demonstrar a arquitetura *Named Data Networking* (NDN) na prática.
- Apresentar as plataformas de *testbeds* da arquitetura.

A proposta da arquitetura *Named Data Networking* para a Internet parece ser a solução para futuros e até atuais problemas. Entretanto, o foco deste novo paradigma não versa sobre segurança e sim sobre uma mudança de contexto da arquitetura, com foco em nomeação no conteúdo e não nos hosts, sendo necessário que proposições de avanço e ajustes nos aspectos de segurança sejam feitos.

A primeira seção deste trabalho faz uma introdução do tema, do problema de pesquisa, do objetivo deste trabalho e sua justificativa. A segunda seção aborda os fundamentos teóricos necessários para realização deste trabalho que envolve redes de computadores, o protocolo TCP/IP, redes orientadas ao conteúdo, *Named Data Networking* (NDN), *Mini-NDN*. A terceira seção descreve os procedimentos metodológicos utilizados e limitações de pesquisa. A quarta seção apresenta a discussão e resultados do trabalho. A quinta e última seção contém as conclusões obtidas neste trabalho.

## 2. FUNDAMENTAÇÃO TEÓRICA

### 2.1. PROTOCOLO TCP/IP

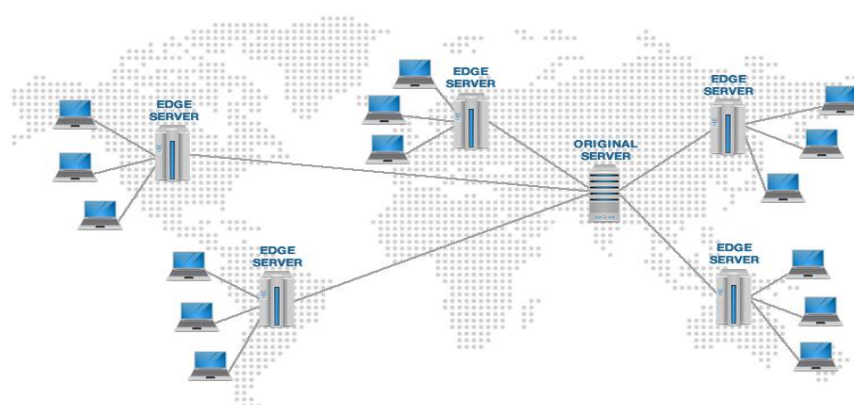
A Internet é uma rede pública de comunicação de dados, com controle descentralizado e que utiliza o conjunto de protocolos TCP/IP como base para a estrutura de comunicação e seus serviços de rede. Isto se deve ao fato de que a arquitetura TCP/IP fornece não somente os protocolos que habilitam a comunicação de dados entre redes, mas também define uma série de aplicações que contribuem para a eficiência e sucesso da arquitetura. A Internet é dita ser um sistema aberto uma vez que todos os seus serviços básicos assim como as aplicações são definidas publicamente, podendo ser implementadas e utilizadas sem pagamento de *royalties* ou licenças para outras instituições. O conjunto de protocolos TCP/IP foi projetado especialmente para ser o protocolo utilizado na Internet. Sua característica principal é o suporte direto a comunicação entre redes de diversos tipos. Neste caso, a arquitetura TCP/IP é independente da infraestrutura de rede física ou lógica empregada. (PUC RIO/CCE, S/D).

Uma solução encontrada para diminuir a latência (atraso na entrega de informações) na arquitetura atual da Internet (TCP/IP) são as *content distribution networkings* (CDN), ou em tradução livre, redes de distribuição de conteúdo, uma explicação do que são as CDN's será apresentada a seguir.

As redes de entrega de conteúdo (CDN) são o *backbone* transparente da Internet, responsável pela entrega de conteúdo. Quer saibamos ou não, cada um de nós interage com as CDN's diariamente; ao ler artigos em sites de notícias, fazer compras *on-line*, assistir a vídeos do *YouTube* ou ler os *feeds* de mídia social. Não importa o que você faça ou que tipo de conteúdo você consuma, é provável que você encontre CDNs por trás de cada caractere de texto, cada *pixel* de imagem e cada quadro de filme entregue ao seu PC e navegador móvel. Para entender por que as CDNs são amplamente usadas, primeiro você precisa reconhecer o problema que elas foram projetadas para solucionar. Conhecido como latência, é o atraso que ocorre a partir do momento em que você solicita o carregamento de uma página da *web* até o momento em que seu conteúdo é exibido na tela. Esse intervalo de atraso é afetado por vários fatores, muitos deles específicos para uma determinada página da *web*. Em todos os casos, no entanto, a duração do atraso é afetada pela distância física entre você e o servidor de hospedagem desse *website*. A missão de um CDN é reduzir praticamente a distância física, com o objetivo de melhorar a velocidade e o desempenho da renderização do *site*. Para

minimizar a distância entre os usuários e o servidor de seu *website*, uma CDN armazena uma versão em *cache* de seu conteúdo em vários locais geográficos (por exemplo, pontos de presença ou PoPs). Cada PoP contém vários servidores de armazenamento em *cache* responsáveis pela entrega de conteúdo para os visitantes em sua proximidade. Em essência, o CDN coloca seu conteúdo em muitos lugares ao mesmo tempo, fornecendo uma cobertura superior aos seus usuários. Por exemplo, quando alguém em Londres acessa seu *site* hospedado nos EUA, isso é feito por meio de um PoP local do Reino Unido. Isso é muito mais rápido do que ter as solicitações do visitante e suas respostas, percorrer toda a extensão do Atlântico e voltar. É assim que um CDN funciona (WHAT..., S/D).

Figura 1 - Content Distribution Network



Fonte 1 - T. ZUHORA, 2018

### 2.1.1. SEGURANÇA EM TCP/IP

Os protocolos TCP/IP (*Transmission Control Protocol / Internet Protocol*) e a própria Internet não foram originalmente projetados tendo a segurança como prioridade, já que o número de usuários e os tipos de aplicações da época não requeriam maiores esforços para a garantia da mesma. No entanto, com o rápido crescimento da Internet e o surgimento de várias aplicações, as questões relativas à segurança de redes e sistemas tornaram-se demandas inquestionáveis. Enquanto a Internet se restringia aos meios científicos e acadêmicos, os problemas de segurança não eram tão críticos porque havia um certo controle baseado nos códigos de uso ético da rede. Mas, com a abertura da Internet para o setor privado,

principalmente comercial, os problemas de segurança se intensificaram e tornaram-se críticos. A segurança está relacionada a dois grandes aspectos:

- A certeza de que certa informação provém realmente de quem se diz ser o remetente e de que não foi modificada ao longo do percurso (integridade);
- A certeza de que ao longo do percurso até a chegada ao destinatário, a informação não foi recebida por mais ninguém.

Esses dois grandes problemas são resolvidos respectivamente com a autenticação e a criptografia. Uma arquitetura em camadas tem como característica fundamental a divisão das responsabilidades do envio da informação entre as diversas camadas, de forma que o usuário, no topo da pilha, possa recebê-la. O grande questionamento que se apresenta, portanto é que camada deve se responsabilizar pela segurança. A princípio, a solução mais fácil encontrada foi implementá-la individualmente para cada aplicação na medida em que surgiam as necessidades. Assim, surgiu o PGP para comunicação por *e-mail*, o SSL/TLS para navegação na Internet ou o SSH para a *login* remoto seguro. No entanto, apesar dessa implementação na camada aplicação ser mais simples já que não envolve o sistema operacional e o foco se restringir a uma aplicação específica, um serviço implementado nessa camada particulariza-o demais, fazendo com que haja a necessidade de um novo desenvolvimento para cada nova aplicação. Nada mais natural então em se pensar que esse serviço seja oferecido por uma camada mais inferior na pilha de protocolos, de forma a unificar a solução e permitir que um único serviço resolva o problema de todas as camadas superiores. Porém, levando esse raciocínio ao extremo e descendo até a camada de rede, encontram-se dificuldades na sua implementação, já que seriam necessários *links* dedicados e essa não seria uma solução escalável, não servindo, portanto, para a Internet. Assim, a solução ótima encontrada foi a de implementar o serviço de segurança na camada de inter-redes, proporcionando um controle por fluxo ou por conexão. Vários protocolos foram desenvolvidos nessa camada com esse objetivo, alguns rodam apenas sobre o *IP* (como o caso do GRE e do PPTP) e outros mais versáteis, são capazes de tratar não só pacotes IP, mas também IPX e NetBEUI (como o caso do L2F ou L2TP). Para atender a Internet, porém, o melhor seria que o próprio protocolo IP fosse capaz de oferecer esse serviço sem depender de outros protocolos, gerando uma unificação na resolução do problema. Foi proposta então a integração dessa funcionalidade à nova versão do IP (o IPv6), cujo desenvolvimento ficou a cargo do grupo de trabalho *IP Security Protocol* (IPSec) da IETF (*Internet Engineering Task Force*). Com a demora da migração da Internet para a nova versão do IP (fato não ocorrido até o presente momento), um



conjunto de adaptações foi feito para o IPSec poder rodar sobre o *IPv4*. Sua implementação é projetada através de *headers* especiais: *headers extensions* no *IPv6* e um *header* de protocolo a mais no *IPv4*, normalmente colocados entre o *header* original do IP e seu *payload* (MUNIZ BANDEIRA DUARTE; G. LOPEZ, 2003).

Um dos problemas clássicos de segurança dessa nossa atual arquitetura (TCP/IP), que não existe na arquitetura que será apresentada nessa pesquisa, é o *IP spoofing*. O *IP spoofing* refere-se ao sequestro de conexão por meio de um endereço IP (*Internet Protocol*) falso. O *IP spoofing* é a ação de mascarar um endereço IP do computador para que pareça autêntico. Durante esse processo de mascaramento, o endereço IP falso envia o que parece ser uma mensagem mal-intencionada acoplada a um endereço IP que parece ser autêntico e confiável. Na falsificação de IP, os cabeçalhos IP são mascarados por meio de uma forma de TCP (*Transmission Control Protocol*), na qual os *spoofers* descobrem e manipulam informações vitais contidas no cabeçalho IP, como endereço IP e informações de origem e destino (IP..., S/D).

A arquitetura *Named Data Networking* (NDN) é totalmente nova, mas tem como base a atual arquitetura da Internet e todos os pontos que levaram essa mesma ao sucesso e ao inteiro funcionamento que essa possui hoje. Como dito anteriormente, no início da Internet, a telefonia foi tida como o único exemplo de comunicações bem-sucedidas e eficazes em escala global (NAMED, S/Da). Na época, a solução oferecida pelo TCP/IP era de fornecer uma comunicação ponto-a-ponto entre dois *hosts*, solução essa que atendia e atende perfeitamente quando o assunto é comunicação. A natureza conversacional do IP é incorporada em seu formato de datagrama, os datagramas de IP só podem nomear terminais de comunicação (endereços de origem de endereços de destino de *IP*).

## 2.2. REDES ORIENTADAS AO CONTEÚDO

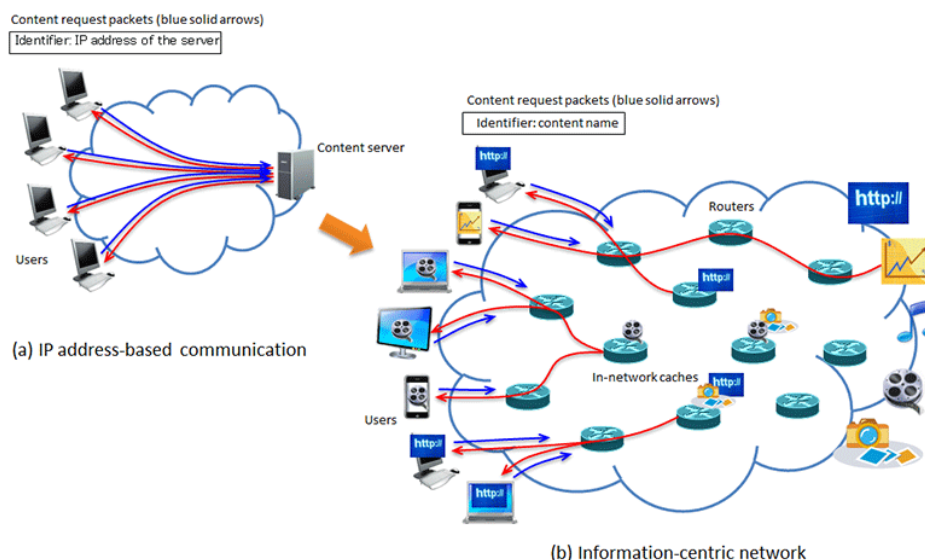
Com a popularização da Internet e o imenso número de aplicações desenvolvidas, o objetivo para o uso da *internet* deixou de ser apenas a “clássica” comunicação (troca de mensagens), e passou a ser uma comunicação mais “carregada” (comunicação juntamente com distribuição de conteúdo). Entende-se conteúdo como tudo aquilo que ocupa, parcial ou totalmente, o espaço em algo (fotos, vídeos, arquivos em geral (independentemente de sua natureza)).

As redes orientadas a conteúdo (ROCs) introduzem um novo paradigma de comunicação para a Internet. As ROCs enfatizam o acesso à informação independentemente de sua localização. Diferentemente da abordagem tradicional da Internet, centrada na identificação e localização de estações, as ROCs utilizam conceitos inovadores como conteúdo nomeado, roteamento baseado em nomes, segurança aplicada diretamente a conteúdos e armazenamento de dados nos elementos do núcleo da rede. Tais conceitos permitem criar uma arquitetura mais eficiente para a distribuição de conteúdo, evitando assim todos os remendos necessários à arquitetura vigente da Internet, como o IP *Multicast*, o uso do DNS, IPsec etc. A arquitetura baseada em conteúdo pode, então, prover de forma nativa novas funcionalidades, como o compartilhamento eficiente de recursos e de dados, mecanismos para aumentar a disponibilidade dos conteúdos, suporte à segurança intrínseca de conteúdo, suporte à mobilidade etc. (DE BRITO; VELLOSO; MORAES, 2012).

Segundo Jacobson qualquer arquitetura feita para rodar sobre alguma coisa, é necessariamente uma sobreposição (*overlay*). Sob essa ótica, nossa atual arquitetura já se iniciou como um *overlay*, com o IP se iniciando como um *overlay* sobre o sistema de telefonia, e hoje a telefonia é um *overlay* sobre o IP, visto que hoje o IP tem uma universalidade independente em qualquer camada de rede. As redes orientadas ao conteúdo têm essa característica do IP, roda sobre qualquer coisa (inclusive o IP), e qualquer coisa roda sobre ela (inclusive o IP) (JACOBSON, 2009b). Ainda segundo Jacobson a maneira direta e unificada de resolver esses problemas é substituir onde com o que. As conversas de *host* para *host* são uma abstração de rede escolhida para se adequar aos problemas dos anos 60 (problemas de comunicação). Os dados nomeados são uma melhor abstração para os problemas atuais de comunicação do que os *hosts* nomeados. É introduzida a Rede Orientada ao Conteúdo (CCN), uma arquitetura de comunicações construída em dados nomeados. CCN em seu nível mais baixo não conhece *hosts* – ela só conhece um pacote com o “endereço” do nome do conteúdo. Foram preservadas as características de design que tornam o TCP/IP simples, robusto e escalável. Muito do sucesso do IP se deve à simplicidade de sua camada de rede (o pacote IP - a fina "cintura" fina da pilha) e as fracas exigências que faz na camada 2, exemplo: sem estado, entrega não confiável, não ordenada e de melhor esforço. A camada de rede da CCN (camada 3) é semelhante ao IP e faz menos demandas na camada 2, dando-lhe muitas das mesmas propriedades do IP e mais atraentes. Inclusive, CCN pode ser colocado sobre qualquer coisa, incluindo o próprio IP. Mas a CCN não é em tudo igual ao IP, a CCN se afasta do IP de várias maneiras críticas. Dois destes, estratégia e segurança, são mostrados

como novas camadas em sua pilha de protocolos. O CCN pode aproveitar ao máximo várias conectividades simultâneas (por exemplo, *ethernet*, 3G, *bluetooth* e 802.11) devido ao seu relacionamento mais simples com a camada 2. A CCN protege o conteúdo em si, em vez de as conexões sobre as quais ele viaja, evitando assim muitas das vulnerabilidades baseadas em *host* que afetam a rede IP (JACOBSON, 2009a).

Figura 2 – IP x ICN



Fonte 2 - KOYAMA e KAGEYAMA, 2014.

O CCN é um protocolo de solicitação e resposta para buscar partes de dados usando um nome. A integridade de cada pedaço pode ser afirmada diretamente através de uma assinatura digital, ou, alternativamente, indiretamente via cadeias *hash*. Os fragmentos também podem levar a verificações de integridade de mensagens mais fracas ou a nenhum mecanismo de proteção de integridade. A integridade dos dados é, portanto, uma característica essencial do CCN; não conta com o canal de transmissão de dados. CCN usa nomes hierárquicos para identificar *bytes* de carga útil (pacotes). O nome combina um prefixo roteável com um sufixo arbitrário que depende do aplicativo atribuído pelo editor a um pedaço de conteúdo. Como um protocolo de solicitação e resposta, o CCN pode ser transportado em muitos transportes diferentes. Dentre eles os que estão em uso hoje são *Ethernet*, TCP, UDP, 802.15.4, GTP, GRE, DTLS, TLS e outros. O conceito-chave do CCN é que um nome subjetivo é (criptograficamente) vinculado a uma carga útil (pacote de dados). Essas ligações (geradas pelo editor) podem, portanto, ser verificadas (criptograficamente). Por exemplo, um editor pode calcular um *hash* criptográfico sobre o nome e a carga útil, assinar o *hash* e entregar o nome, (*payload*) e a validação da carga útil (pacote de dados). Os

consumidores desses dados podem verificar a integridade da ligação, re-computando o mesmo *hash* criptográfico e verificando a assinatura digital. Informações adicionais seriam incluídas conforme necessário. (MOSKO; SOLIS, 2017).

### 2.2.1. DATA-CENTRIC SECURITY

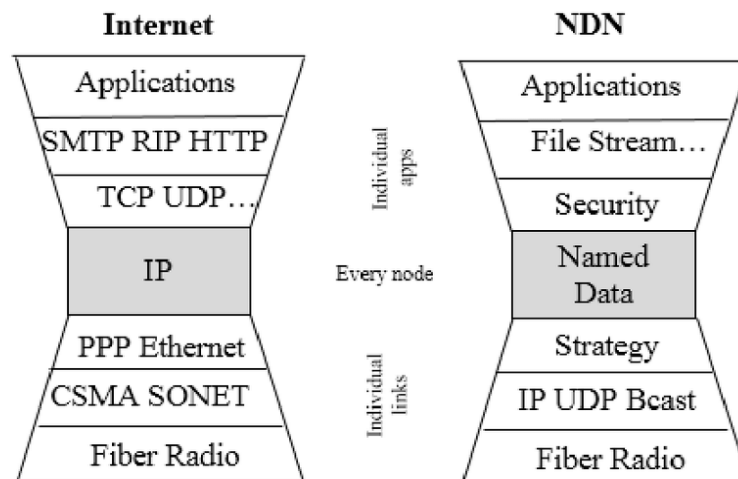
Diferente do TCP/IP, que deixa a responsabilidade da segurança (ou da falta dela) aos *hosts*, a NDN assegura os próprios dados, exigindo que os produtores dos dados assinem criptograficamente todos os pacotes de dados. A assinatura do editor garante a integridade e permite a determinação da proveniência dos dados, permitindo que a confiança do consumidor nos dados seja dissociada de como ou de onde o dado é obtido. Também apoia a decisão do consumidor, permitindo que o mesmo raciocine se um proprietário de chave pública é um editor aceitável para um determinado dado em um contexto específico. O segundo principal objetivo de pesquisa é projetar e desenvolver mecanismos utilizáveis para gerenciar a confiança do usuário. A segurança centrada em dados da NDN tem aplicativos nativos para controle de acesso ao conteúdo e segurança da infraestrutura. Os aplicativos podem controlar o acesso aos dados por meio de criptografia e distribuir chaves (criptografia de dados) como dados NDN criptografados, limitando o perímetro de segurança de dados ao contexto de um único aplicativo. Requerendo assinaturas no roteamento de rede e mensagens de controle (como qualquer outro dado da NDN), base sólida para proteger protocolos de roteamento contra, por exemplo, *spoofing* e adulteração de dados (JACOBSON; AFANASYEV; ZHANG, 2014).

### 2.3. INTRODUÇÃO A NAMED DATA NETWORKING

A arquitetura NDN pretende generalizar a arquitetura da Internet, removendo essa restrição que permite que apenas os endereços de origem e endereços de destino sejam nomeados. Os nomes dos datagramas da NDN são hierarquicamente estruturados e que de outra forma são identificadores de dados arbitrários. Os datagramas NDN podem nomear um vídeo do *youtube*, partes de uma conversa na Internet, etc. Essa mudança na arquitetura da Internet permite que a NDN use o atual modelo de ampulheta da Internet, modificando a

chamada “cintura fina” da ampulheta, para que assim a Internet use dados nomeados ao invés de endereços IP para a entrega de dados para o usuário final (NAMED, S/Da).

Figura 3 - Internet atual x NDN



Fonte 3 - EL KAFHALI; RAHEL; JAMALI, 2017

Essa mudança que o NDN pretende fazer pode parecer simples, mas traz várias novas oportunidades. Como dito anteriormente, a atual arquitetura da Internet não permite que os dados sejam nomeados, e sim, somente os endereços de origem e endereços de destino. Por conta disso, as aplicações atuais são escritas no contexto de quais dados elas querem e não de onde eles estão localizados. Daí em diante o *middleware* (*software* intermediário) da aplicação é requisitado para buscar o local em que o dado está localizado na Internet, para então entregar ao destinatário que o requisitou. Com a NDN, o modelo da aplicação pode ser usado diretamente na entrega dos dados, fazendo desnecessário o uso do *middleware* e todo o seu *overlay* que é causado desde a sua configuração até o seu acionamento (NAMED, S/Da).

O funcionamento da arquitetura NDN possui 6 pilares que são o princípio de seu *design*, são eles:

1. Universalidade: O NDN deve ser um protocolo de rede comum para todos os aplicativos e ambientes de rede. Os aplicativos e ambientes de rede que o NDN deve oferecer, incluem (mas não estão limitados a): comunicação baseada em infraestrutura (*Web*, *Youtube*, conferência em tempo real etc.) *ad hoc* com e sem comunicação de infraestrutura (aplicativos IoT, redes *mesh* sem fio, rede

de veículos, veículo-veículo-para-infraestrutura etc.). Comunicação estilo DTN (redes regionais), comunicação em *links* intermitentes e perturbadores (ambientes de primeiro socorro), aplicação usando *links* unidirecionais (por exemplo, satélite). Portanto, o protocolo NDN e o formato de pacote NDN devem suportar ampla gama de aplicações, desde ambientes restritos (IoT) até aplicações de *big data science*: o formato de pacote NDN deve ser flexível e extensível. O protocolo NDN e o formato de pacote devem suportar a evolução do protocolo sem dias de sinalização: sem partes fixas ou campos de tamanho fixo no cabeçalho. As operações do protocolo de rede principal não devem depender da sincronização do relógio. (NDN..., S/Da).

2. Centralidade de dados e imutabilidade de dados: NDN deve buscar “pacotes de dados” imutáveis e exclusivos, solicitados usando “pacotes de interesse”. O protocolo NDN e o formato de pacote devem incluir apenas elementos diretamente relacionados aos dados, ou seja, universalmente necessários, necessários e significativos em todos os ambientes de comunicação. Outros elementos necessários em ambientes específicos (por exemplo, na Internet baseada em infraestrutura de hoje) devem ir para a (s) camada (s) de adaptação de rede. A imutabilidade do pacote de dados permite a desambiguação da coordenação no sistema distribuído que pode não estar sempre conectado. Embora os pacotes de dados sejam imutáveis, os aplicativos podem fazer alterações no conteúdo comunicado criando novas versões de pacotes de dados imutáveis. (NDN..., S/Da).
3. Protegendo dados diretamente: A segurança deve ser propriedade dos pacotes de dados, permanecendo os mesmos, quer os pacotes estejam em movimento ou em repouso. Os dados diretamente protegidos e nomeados de forma exclusiva eliminam o requisito de canais diretos entre as extremidades de comunicação e permitem a produção assíncrona e o consumo de dados nomeados e seguros, por exemplo, usando *caches* na rede e repositórios gerenciados. Os consumidores devem ser capazes de validar pacotes de dados individuais. Idealmente, cada pacote deve ser verificável por conta própria. Como uma otimização de engenharia, os pacotes podem ser tornados verificáveis no contexto de outros, desde que o contexto possa ser inferido a

partir do próprio pacote de dados (seu nome ou informação no campo de assinatura) (NDN..., S/Da).

4. Nomenclatura Hierárquica: Os pacotes devem conter nomes hierárquicos para permitir a demultiplexação e fornecer contexto estruturado. A hierarquia de nomes fornece contexto para implementar e impor vários modelos de segurança, ou seja, fornecer restrições estruturadas sobre quais chaves podem assinar quais dados. Nomes hierárquicos permitem modelos de nomenclatura "simples", se necessário/desejado pelos aplicativos (NDN..., S/Da).
5. Descoberta de nome na rede: Os interesses devem poder usar nomes incompletos para recuperar pacotes de dados. Um consumidor pode não conhecer o nome completo do nível de rede para os dados, pois algumas partes do nome não podem ser adivinhadas, computadas ou inferidas de antemão. Uma vez que os dados iniciais são recebidos, as convenções de nomenclatura podem ajudar a determinar os nomes completos de outros dados relacionados. A maioria dos interesses carregará nomes completos na descoberta de nomes na rede que deverão ser usados para iniciar a comunicação (NDN..., S/Da).
6. Balanço de fluxo *hop-by-hop*: Em cada *link*, um pacote de interesse não deve trazer mais de um pacote de dados. O balanceamento de fluxo *hop-by-hop* permite que cada nó controle a carga sobre seus *links*. Ao decidir enviar juro por um link, o roteador confirma a largura de banda dos dados retornados. Ao limitar o número de interesses enviados, cada roteador e nó cliente na rede controlam a quantidade de dados que receberá (NDN..., S/Da).

O NLSR é um protocolo de roteamento no NDN que preenche a Base de informações (FIB) de roteamento do NDN. O NLSR continuará a evoluir ao lado do protocolo *Named Data Networking*. O NLSR é um pacote de software aberto e gratuito, licenciado sob a licença GPL 3.0 e gratuito para todos os usuários e desenvolvedores da Internet. O NLSR é desenvolvido pelos membros da equipe de projeto da NDN, patrocinada pela NSF. O principal objetivo do design do NLSR é fornecer um protocolo de roteamento para preencher o FIB do NDN. O NLSR calcula a tabela de roteamento usando o roteamento de estado de link ou hiperbólico e produz várias faces para cada prefixo de nome alcançável em um único domínio autoritativo. O NLSR continuará a evoluir ao longo do tempo para incluir a descoberta de vizinhos e se tornar um protocolo de roteamento entre domínios completo para o NDN (NLSR..., S/D).

### 2.3.1. COMUNICAÇÃO EM NDN

As conversas tidas na Internet (sejam elas bate-papo ou trocas de dados) são passageiras. A atual arquitetura da Internet “blinda” o canal em que os dois *hosts* usam para trocar as informações. Essa abordagem nem sempre é eficaz, pelo menos é o que se vê nas estatísticas. Na NDN todos os dados são assinados pelo “produtor” e verificados pelo receptor. A arquitetura NDN pode afirmar se o arquivo que você recebeu realmente veio de quem você pensa que o enviou. NDN não protege o meio da comunicação, e sim os dados que são requisitados e buscado na Internet (NAMED, S/Da).

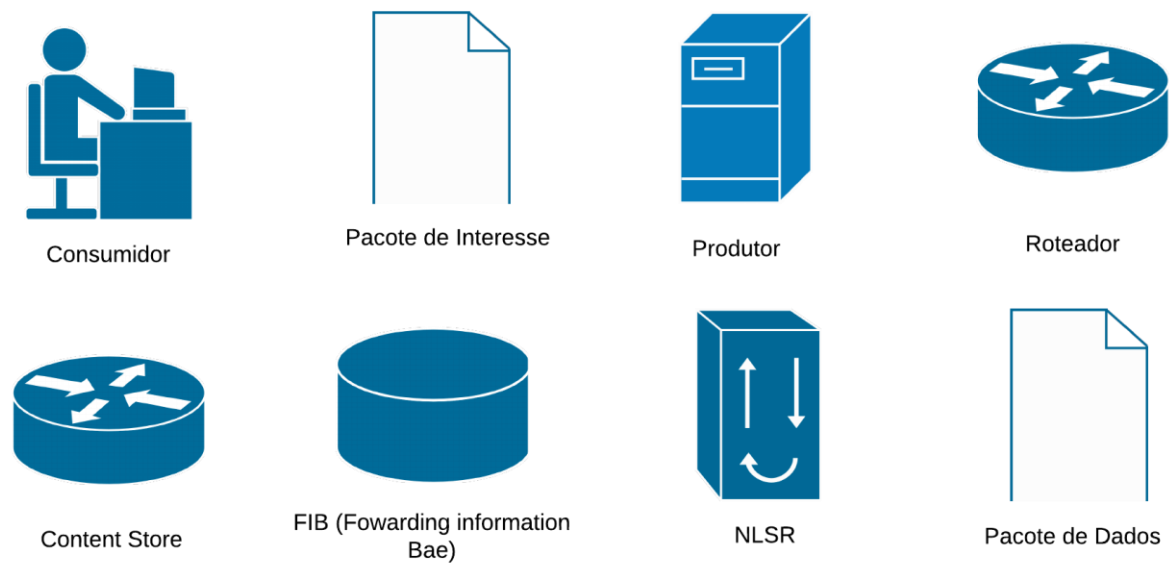
Com a nomeação exclusiva dos dados, um pacote de NDN independe de onde ele veio ou para onde ele vai ser encaminhado, sendo assim ele pode ser armazenado em cache dentro de uma rede para atender mais rapidamente alguma solicitação futura. Os nomes exclusivos dos pacotes podem eliminar também alguns dos problemas atuais dos roteadores, como por exemplo a eliminação do *looping* de dados, permitindo assim que qualquer nó use livremente toda a sua conectividade para solicitar ou distribuir os dados, e eliminando a assimetria de informações que dá um controle desproporcional sobre rotas para os provedores maiores, e consequentemente, para os provedores menores (NAMED, S/Da).

Assim como na atual arquitetura da Internet, a “cintura fina” é a parte vital da arquitetura NDN. A diferença é que a “cintura fina” de NDN usa nome de dados ao invés de endereços IP para entregar conteúdo. Essa mudança pode parecer simples, mas irá apresentar mudanças significativas na forma em que essas duas arquiteturas entregam os dados, e nas suas operações (NAMED, S/Da).

Para receber determinado dado, o *host* (consumidor) envia um pacote de interesse (*interest packet*), esse pacote contém um nome que identifica os dados desejados. Um roteador que está no caminho “memoriza” a interface da qual veio a solicitação, logo em seguida encaminha o pacote de interesse procurando pelo nome em sua *Forwarding Information Base* (FIB). Essa FIB é preenchida por algum protocolo de roteamento baseado em nome. A partir do momento em que o pacote de interesse chega em um nó que possui os dados que foram solicitados, um pacote de dados é enviado de volta para o *host* receptor. Esse pacote de dados possui o nome e o conteúdo dos dados que foram requisitados, juntamente com a assinatura pela chave do produtor dos dados (NAMED, S/Da).



Figura 4 - Elementos arquitetura NDN

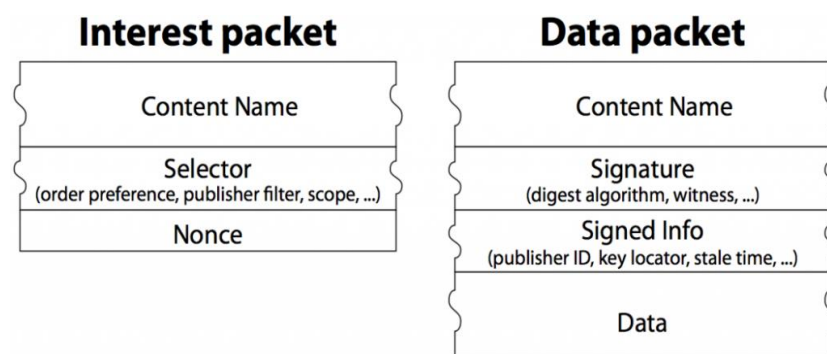


Fonte 4 - PRÓPRIA, 2019

A comunicação em NDN é conduzida pelo *host* receptor (por quem está requisitando os dados, pelo consumidor). Os campos dos pacotes de interesse e de conteúdo são mostrados na Figura 4.

Esse pacote de dados segue no sentido inverso do caminho que o pacote de interesse percorreu, para assim retornar ao consumidor. É importante observar, que nem o pacote de interesse e nem o pacote de dados possuem endereços de hosts ou nomes de interfaces. Sendo assim, os pacotes de interesse são encaminhados para os produtores de dados de acordo com os nomes que estão nos pacotes de interesse, e os pacotes de dados retornam de acordo com as informações de estado configuradas pelos pacotes de interesse em cada salto do roteador (NAMED, S/Da).

Figura 5 - Formato dos pacotes NDN



Fonte 5 - NAMED, S/Da

O roteador armazena todos os pacotes de interesse que estão esperando o retorno do pacote de dados em uma *Pending Interest Table* (PIT). Pode acontecer de chegar vários pacotes de interesse requisitando um mesmo dado, quando isso acontecer apenas o primeiro é enviado para uma fonte de dados. Todos os pacotes que precisem ser armazenados na *Pending Interest Table* (PIT) contêm o nome do interesse e também um conjunto de interfaces de onde o pacote de interesse foi enviado. Quando o dado requisitado chega ao roteador o mesmo consulta sua PIT e encaminha o dado para todas as interfaces listadas na PIT. Logo depois que os dados são enviados a tabela PIT (NAMED, S/Da). A arquitetura da NDN dispõe de nomes hierarquicamente estruturados, como exemplo um vídeo do *youtube* pode ser representado da seguinte forma: /youtube/vídeos/video1.avi. Nessa nomenclatura a '/' não faz parte do nome, ela indica um limite entre os componentes que formam o "caminho" do arquivo. Essa estrutura hierárquica é muito boa para os aplicativos representarem o relacionamento entre as partes dos dados. Como exemplo a parte 2 do vídeo1 pode ser escrita assim: /exemplo/vídeos/video1.avi/2.

### 2.3.2. FUNCIONAMENTO SEGURANÇA EM NDN

Proteger a comunicação em aplicações de rede envolve muitas tarefas complexas. A *Named Data Networking* (NDN) cria uma autenticação de dados na camada de rede, exigindo que todas as aplicações assinem e autenticuem os fluxos de dados. Para tornar essa autenticação utilizável, a decisão sobre quais chaves podem assinar quais dados e o procedimento de verificação de assinatura precisam ser automatizados (SCHEMATIZING, 2015).

A arquitetura *Named Data Networking* (NDN) utiliza nomenclatura hierárquica, selando os pacotes nomeados com assinaturas de chave pública. Os produtores usam nomes de chaves para indicar qual chave pública o consumidor deve recuperar para verificar as assinaturas dos dados produzidos. Além de buscar as chaves específicas e verificar a assinatura, os consumidores também podem corresponder aos dados e aos nomes das chaves para determinar se a chave está autorizada a assinar cada pacote de dados específicos (SCHEMATIZING, 2015).

NDN usa um modelo de autenticidade baseado em conteúdo, exigindo que cada pacote de dados seja assinado. Além da assinatura, cada pacote de dados também tem de transportar metadados adicionais incluindo o nome da chave de assinatura.

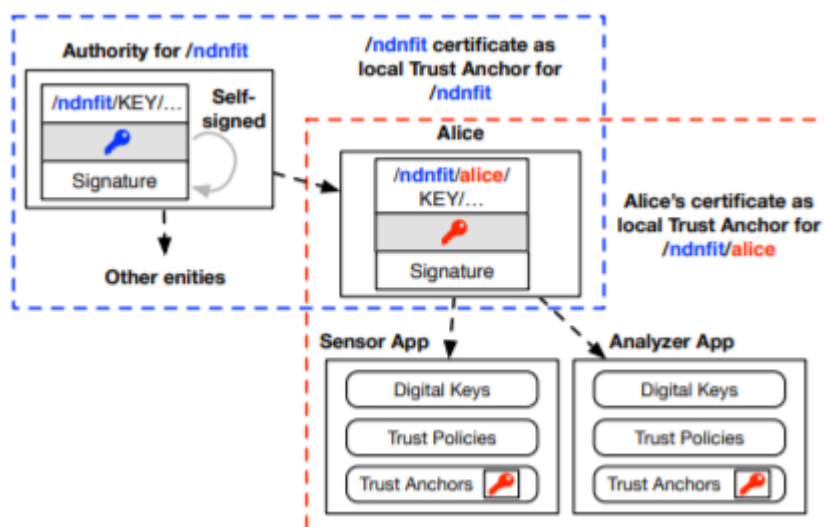
Para autenticar um pacote de dados, é necessário um modelo de confiança que define quais chaves estão autorizadas a assinar quais dados (NAME-BASED ACCESS CONTROL, 2015).

A estrutura de segurança da NDN é construída sobre criptografia de chave pública. Chamamos aplicativos e todos os outros participantes de comunicação de rede em uma rede NDN, entidades (uma entidade pode ser um país, uma cidade, um estado, uma cidade, um vizinho, uma casa, um aplicativo ou até mesmo um processo). Cada entidade possui um ou mais nomes criptográficos e pares de chaves pública-privada. Um certificado de NDN para um usuário denominado "/Ucla/cs/zhiyi" liga este nome e sua (s) chave (s) em conjunto; isto certifica a propriedade do nome pelo usuário, juntamente com chave. Cada nome certificado é chamado de identidade e cada entidade pode emitir certificados para os *sub-namespaces* que ele delega para outras entidades.

A utilização de criptografia de chave pública para validar as comunicações exige que o NDN resolva os três desafios a seguir:

Estabelecer âncora (s) de confiança: No *NDN*, uma âncora de confiança é a autoridade de certificação para um determinado *namespace*. Quando uma âncora de confiança é instalada, os usuários podem verificar as assinaturas de outras entidades retrocedendo e verificando os certificados ao longo da cadeia de certificados para a âncora de confiança. Âncoras de confiança são geralmente instaladas em várias entidades através de mecanismos *out-of-band*, e o desenvolvimento destes mecanismos de apoio diretamente depende do modelo de âncora de confiança em uso.

Figura 6 - Entidades e Âncoras em NDN



Fonte 6 - ZHANG et al., 2018

Fornecer soluções eficazes para o gerenciamento de confiança: para validar todos os dados recebidos, um participante deve saber que a (s) chave (s) pode legitimamente assinar ou criptografar qualquer parte dos dados. Soluções eficazes devem permitir que os aplicativos expressem suas próprias políticas de confiança e executar essas políticas automaticamente. Fornecendo soluções de gerenciamento de chaves utilizáveis: Assinatura, verificação, criptografia e decriptografia envolvem criptografia de chaves. Soluções criptográficas utilizáveis exigem mecanismos para atribuir e entregar chaves ou certificados apropriados de uma maneira eficiente e automática. A NDN utiliza um modelo de âncora de confiança diferente do que está em uso hoje, a saber: (i) utilizando centenas de autoridades de certificação comercial como âncoras de confiança para autenticar outras partes com as quais deseja se comunicar (por exemplo, Certificados TLS), (ii) instalar uma única âncora de confiança global (por exemplo, DNSSEC) e (iii) estabelecer confiança de uma forma ad-hoc (por exemplo, confiar no primeiro uso, ou "TOFU"). NDN pressupõe que a autoridade de cada sistema em rede (uma organização, uma casa, etc.) deve estabelecer sua própria âncora de confiança local. As entidades sob essa autoridade podem descobrir essas âncoras de confiança, configurações do sistema local e, em seguida, obter certificados e aprender políticas de confiança deles. Este modelo de confiança segue a infraestrutura de segurança distribuída simples (SDSI) no estabelecimento de âncoras de confiança. Com âncoras de confiança instaladas, certificados e políticas de confiança, uma entidade pode utilizar suas chaves para garantir a autenticidade, integridade e confidencialidade dos dados. Além disso, o armazenamento na rede da NDN também ajuda a melhorar a disponibilidade dos dados. A

segurança da NDN utiliza criptografia de chave pública e depende o uso de chaves digitais. Além disso, o NDN também usa os seguintes blocos de construção: políticas de confiança e certificados NDN.

**Políticas de Confiança:** As aplicações definem políticas de confiança para determinar se determinado pacote ou identidade é confiável ou não. Dado que os produtores de dados nomeiam pacotes de dados (incluindo certificados) de forma estruturada e significativa, os consumidores podem aceitar pacotes com formatação de nome apropriada.

**Confiança das políticas em NDN** são baseadas na semântica do nome.

**Certificados de NDN:** um signatário de certificado assina certificados de NDN sob o próprio espaço de nomes ou assina outras chaves (sob diferentes *namespaces*) como um endosso (por exemplo, em uma teia de confiança). Um certificado NDN é um pacote de dados que transporta informações de chave pública e pode ser buscado usando pacotes de interesse normais. Os nomes dos certificados seguem a convenção de nomenclatura “/ <prefixo> / KEY / <key-id> / <issuer-info> / <cert-version> ”, onde o "prefixo" representa uma identidade e os componentes depois de "KEY" são a chave id, issuer informações e versão do certificado. O NDN exige que os produtores assinem todos os pacotes de dados, para que os consumidores possam verificar a assinatura de cada dado entrante, garantindo a autenticidade e integridade dos dados. Mais importante, a semântica de nome avançado do NDN permite que os consumidores usem políticas de confiança baseadas em nomes para avaliar a confiança, verificando quais pedaços de dados são assinados, e por qual chave. Desta forma, as políticas de confiança limitam o poder de cada chave de assinatura e garantem que cada pacote confiável é assinado por uma chave legítima, fornecendo autenticidade de dados em uma granularidade fina. Por exemplo, a chave certificada no certificado “/ ndnfit / alice / sensor / KEY / 1 / alice-agent / version ” só é permitida assinar pacotes sob o prefixo “/ ndnfit / alice / sensor”. A autenticidade e integridade dos pacotes de dados recebidos (incluindo certificados) é determinado por uma combinação de dois principais fatores:

**Validação por diretivas de confiança:** As convenções de nomenclatura estruturada de pacotes e chaves de dados fornecem contextos explícitos e significativos para aplicativos, permitindo que aplicativos *NDN* definam regras que aceitam apenas pacotes com o nome desejado e formato em relação entre o nome de um pacote e o nome da sua chave de assinatura. Para ser mais específico, o pacote nome, o nome da chave de assinatura, a relação entre estes dois nomes, e o nome da âncora de confiança deve seguir estas regras. O "esquema de confiança" da NDN é uma realização do presente políticas de confiança baseadas em nome.

**Verificação de Assinatura:** Para verificar uma assinatura de dados, os consumidores devem recuperar o certificado do produtor correspondente, que está

identificado pelo nome da chave em uma seção dedicada aos dados do pacote. Este certificado irá recursivamente apontar para o signatário do certificado e, finalmente, chegar a uma âncora. O pacote de origem é considerado válido se todos os certificados obtidos, incluindo a âncora, tem assinaturas válidas e pode satisfazer a confiança políticas. (ZHANG et al., 2018).

Em NDN, a segurança é construída nos próprios dados, em vez de ser uma função de onde ou como é obtida. Cada parte dos dados é assinada juntamente com o seu nome, vinculando-os de forma segura. As assinaturas de dados são obrigatórias, ou seja, os aplicativos não podem "recusar" a segurança. A assinatura, juntamente com as informações do editor de dados, permite a determinação da proveniência dos dados, permitindo que a confiança do consumidor nos dados seja dissociada de como (e de onde) os dados são obtidos. Ele também suporta a confiança refinada, permitindo que os consumidores raciocinem se um proprietário de chave pública é um editor aceitável para um dado específico em um contexto específico. No entanto, para ser prática, essa abordagem de segurança detalhada e centrada em dados requer alguma inovação. Historicamente, a segurança baseada na criptografia de chave pública tem sido considerada ineficiente, inutilizável e difícil de implementar. Além de assinaturas digitais eficientes, a NDN precisa de mecanismos flexíveis e úteis para gerenciar a confiança do usuário. Investigações preliminares mostram que a NDN oferece um substrato promissor para atingir essas metas de segurança. Como as chaves podem ser comunicadas como dados NDN, a distribuição de chaves é simplificada. A vinculação segura de nomes a dados fornece uma base para uma ampla variedade de modelos de confiança, por exemplo, se um dado for uma chave pública, um enlace é efetivamente um certificado de chave pública. Por fim, a abordagem de ponta a ponta da NDN para segurança facilita a confiança entre editores e consumidores. Isso oferece aos editores, consumidores e aplicativos uma grande flexibilidade na escolha ou personalização de seus modelos de confiança. A segurança centrada em dados pode ser estendida ao controle de acesso ao conteúdo e à segurança da infraestrutura. Os aplicativos podem controlar o acesso aos dados por meio de criptografia e distribuir chaves (criptografia de dados) como dados NDN criptografados, limitando o perímetro de segurança de dados ao contexto de um único aplicativo. A exigência de assinaturas nas mensagens de roteamento e controle de rede (como qualquer outro dado do NDN) fornece a segurança do protocolo de roteamento tão necessária. Até o momento da apresentação desse trabalho ainda estão sendo estudada formas de se fazer e utilizar assinaturas eficientes, gerenciamento de confiança utilizável, segurança de rede e proteção de conteúdo e privacidade (NAMED, S/Da).

### 2.3.3. PROBLEMAS

A arquitetura NDN apresenta um novo paradigma de comunicação para a Internet, o seu esquema de segurança de criptografar o dado ao invés do canal é uma solução que resolve vários problemas de segurança atual e até futuros, mas assim como esse esquema de segurança se mostra superior em vários aspectos do esquema de segurança da Internet atual ele também tem algumas desvantagens em relação a esse mesmo esquema. Algumas dessas desvantagens são expostas a seguir.

Segurança das entidades:

Segurança do Host: Diferentemente do IP, o NDN/CCN não possui uma noção explícita de um *host* ou de um sistema final. No entanto, os *hosts* existem implicitamente desempenhando funções de consumidor e/ou produtor. Um *host* atua como consumidor quando emite um interesse e como produtor quando gera conteúdo. Um consumidor puro *host*, ou seja, um que nunca produz qualquer conteúdo, não existe como uma entidade endereçável. Portanto, ele não possui um *namespace* designado e nenhuma chave pública correspondente é usada para verificar seu conteúdo. Consequentemente, os roteadores não devem encaminhar interesses a ele. Além disso, os roteadores encaminham apenas conteúdo para hosts de consumidor que o solicitaram explicitamente. Portanto, um *host* consumidor nunca deve receber tráfego não solicitado de fora do domínio de *broadcast*. Essa é uma vantagem de segurança clara e definida do NDN/CCN sobre o IP. Em contraste, para poder receber interesses, um *host* do produtor precisa anunciar seu *namespace*. A mesma capacidade de receber interesses de entrada também é um meio de receber interesses falsos, o que leva a ataques de inundação de juros. Por enquanto, basta dizer que um produtor-*host* é essencialmente tão (in) seguro quanto um *host* IP na Internet de hoje (GASTI; TSUDIK, 2018).

Segurança do roteador: Um roteador NDN/CCN é substancialmente mais complexo que seu correspondente IP. Este último (roteador IP) é basicamente sem estado em relação ao tráfego de dados. Considerando que um roteador NDN/CCN precisa manter uma PIT e (opcionalmente) um cache. Esses dois tipos de novos estados são diretamente influenciados pelos *hosts*: consumidores e produtores. Além disso, o cache e a PIT exigem suporte de *software* especializado não presente nos roteadores IP. Além disso, um roteador NDN/CCN deve ser capaz de verificar (embora não obrigatório) as assinaturas de conteúdo. Isso exige *software* criptográfico adicional e, possivelmente, *hardware*. Por causa dessas complexidades

adicionais, os roteadores NDN/CCN estão sujeitos a ataques que não se aplicam aos roteadores IP (GASTI; TSUDIK, 2018).

#### Questões Específicas de Segurança:

Envenenamento de cache e de conteúdo: O envenenamento de cache envolve a injeção de conteúdo falso (gerado por um produtor incorreto) ou corrompido (isto é, carregando uma assinatura inválida) em caches de roteadores NDN. Da mesma forma, o envenenamento de conteúdo envolve a injeção de conteúdo falso ou corrompido na rede. O objetivo desses ataques é aumentar o custo de entrega de conteúdo para os consumidores e para a rede. Em princípio, o envenenamento de cache e o envenenamento de conteúdo podem ser resolvidos exigindo-se que os roteadores verifiquem assinaturas no conteúdo que armazenam em cache e/ou encaminham. No entanto, a verificação de assinatura obrigatória na rede aumenta a eficiência e os problemas de gerenciamento de confiança. Primeiro, porque a verificação de assinatura é uma operação dispendiosa em termos computacionais e o segundo, porque a verificação de assinatura é significativa somente se os roteadores tiverem e confiarem na chave pública usada para verificar uma assinatura. Na prática, não se pode esperar que os roteadores recuperem e validem uma ou mais chaves públicas para cada conteúdo encaminhado ou armazenado em cache. Para resolver esses problemas, técnicas como verificação de pacotes randomizados, catálogos assinados, nomes de auto certificação e vinculação segura de *namespaces* e chaves criptográficas foram proposto (GASTI; TSUDIK, 2018).

Controle de acesso ao conteúdo e a privacidade de cache: Como o conteúdo pode ser armazenado em caches não confiáveis na rede, impor o controle de acesso em roteadores individuais é impraticável. Por esse motivo, o NDN/CCN implementa o controle de acesso usando criptografia de conteúdo. Para permitir o acesso a um determinado conteúdo, o produtor deve compartilhar a chave criptográfica com todos os destinatários pretendidos. Qualquer técnica de gerenciamento de chaves, como a criptografia de *proxy* e a criptografia baseada em atributos, pode ser usada para implementar controle de acesso flexível e refinado. A criptografia de conteúdo não garante a privacidade do usuário. Além disso, a criptografia de conteúdo não impede o adversário de saber se um determinado objeto de conteúdo - criptografado ou não - foi armazenado em cache. Esta é uma ameaça significativa para a privacidade dos produtores e dos consumidores. Para melhorar a privacidade do cache,



técnicas baseadas na ocultação de ocorrências de cache e detecção de anomalia foram propostas (GASTI; TSUDIK, 2018).

Comunicação Anônima: Ao contrário dos pacotes IP, os interesses do NDN/CCN não incluem um identificador do remetente. Embora isso pareça oferecer melhor anonimato ao consumidor, (AMBROSIN et al., 2014) mostraram que é possível identificar qual consumidor emitiu um interesse particular explorando o cache na rede. Protocolos de roteamento TOR específicos para NDN/CCN foram propostos como uma forma de implementar comunicação anônima. No entanto, da mesma forma que o roteamento TOR no IP, essas técnicas resultam em maior latência e menor largura de banda. Como tal, não se destinam a transportar uma parte substancial do tráfego da Internet. Além disso, o roteamento TOR elimina os benefícios do armazenamento em cache da rede (GASTI; TSUDIK, 2018).

#### 2.3.4. GDPR

O Regulamento Geral de Proteção de Dados (GDPR) aplica-se automaticamente a todos os 28 estados membros da União Europeia, ao contrário de uma diretiva que exige que os estados membros redijam leis nacionais para fazer cumprir suas regras. Entrou em vigor em 25 de maio de 2018 e se propõe a reforçar os direitos que os cidadãos da união europeia têm sobre seus dados, que são mantidos pelas empresas. Antes de sua implementação, o uso indevido dos dados de uma pessoa era punível com um “tapa no pulso”. Agora, multas gigantescas são emitidas contra empresas que não cumprem as normas do regulamento. As empresas consideradas culpadas de uso indevido de dados podem ser multadas em até € 20 milhões ou 4% do faturamento anual da empresa, nos piores cenários possíveis. O regulamento visa dar às pessoas maior poder sobre seus dados e tornar as empresas mais transparentes na forma como lidam com os dados das pessoas. Até o GDPR entrar em vigor em 25 de maio de 2018, havia apenas a desatualizada Diretiva de Proteção de Dados de 1995, conhecida como Lei de Proteção de Dados de 1998 no Reino Unido. O mundo mudou drasticamente desde 1995 e novas leis foram necessárias para abordar o mundo moderno do uso da Internet em grande escala e das mídias sociais. Nos últimos 24 anos, as empresas tornaram-se mais dependentes da Web, bem como o surgimento de empresas e sites de mídia social, e, como tal, o uso indevido da Internet é muito maior do que em 1995.

Um exemplo de porque as leis são necessárias para a proteção de dados pode ser visto ao usar uma das muitas plataformas digitais, como *Google* ou *Facebook*, que oferecem serviços "gratuitos", mas aceitam algum tipo de pagamento na forma de coleta de dados. Você não está pagando diretamente como tal, mas quando você usa o mecanismo de busca do *Google* ou pesquisa através do *feed* de notícias do *Facebook*, suas ações são registradas e empacotadas como dados para empresas terceirizadas. É assim que você é redirecionado por anúncios ou envia e-mails de *marketing*. Esse tipo de coleta de dados geralmente é mascarado por caixas de seleção não claras ou botões de opção. Você pode até não se lembrar de concordar com eles, mas é a razão pela qual você recebe e-mails que não estão completamente de acordo com seus interesses que apenas enviam *spam* para sua caixa de entrada. Um ótimo exemplo de como as empresas ainda fazem uso indevido de dados foi o escândalo *Cambridge Analytica*, do *Facebook*, quando um aplicativo de terceiros coletou dados sem saber de usuários do *facebook*. Então, essas pessoas foram alvo de campanhas que acabaram afetando injustamente os resultados da eleição de 2016 nos EUA. Um objetivo separado do GDPR é tornar mais fácil e barato para as empresas cumprir as regras de proteção de dados. A diretiva da UE de 1995 permitiu que os estados membros interpretassem as regras da forma que entendessem quando a transformassem em legislação local. A natureza do GDPR como um regulamento, e não uma diretiva, significa que ele se aplica diretamente sem a necessidade de ser transformado em lei, criando menos variações na interpretação entre os estados membros. A UE acredita que isso vai economizar coletivamente 2,3 bilhões de euros por ano. O GDPR aplica-se a organizações em todo o mundo desde 25 de maio de 2018. Como o GDPR é um regulamento, não uma diretiva, o Reino Unido não precisou elaborar nova legislação - em vez disso, aplicou-se automaticamente. Dois níveis de multas existem sob GDPR, mas ambos são muito maiores do que qualquer outro que o Reino Unido tenha visto antes. Sob a Lei de Proteção de Dados de 1998, o regulador do Reino Unido, o *Information Commissioner's Office* (ICO), conseguiu multar as empresas em um máximo de £ 500.000. GDPR aumenta maciçamente o teto de multas. Em primeiro lugar, sua organização enfrenta uma penalidade de até 2% de seu faturamento anual, ou 10 milhões de euros, por não comunicar uma violação de dados à OIC dentro de 72 horas após tomar conhecimento dela. Esse contato inicial deve descrever a natureza dos dados afetados, aproximadamente quantas pessoas são afetadas, quais as consequências que podem significar para elas e quais medidas você já tomou ou planeja agir em resposta. É importante notar que a janela é fixa 72 horas após a descoberta de um incidente, e não 72 horas de trabalho, como algumas empresas foram

levadas a acreditar. Depois, há a multa por uma violação de dados pessoais em si. Violações de dados sob o GDPR podem ser punidas com uma multa máxima de 4% do faturamento anual da sua organização, ou 20 milhões de euros, o que for maior. Com o Reino Unido agora definido para deixar a União Europeia, o Reino Unido formalizou GDPR em nova legislação sob a Lei de Proteção de Dados 2018. GDPR agora se sentará ao lado de DPA, no entanto, na maioria dos casos, o DPA será referido como uma questão de lei. A própria OIC disse que vê as multas como um "último recurso" (HELLARD et al., 2019).

### 2.3.5. GDPR (PRIVACIDADE) x NDN/CCN

Privacidade no Nome: Os nomes do conteúdo NDN/CCN divulgam uma quantidade significativa de informações por meio de seus componentes roteáveis e não roteáveis. Os componentes roteados são análogos a uma combinação de nomes DNS e endereços IP atuais: são cadeias de caracteres humanas legíveis conhecidas (como em nomes DNS) usadas pelos roteadores para identificar o próximo salto de interesses (semelhante aos endereços IP). Naturalmente, os componentes de nome roteáveis podem ser criptografados, em *hash* ou podem ser substituídos por *strings* aleatórias (sem perda de generalidade, no que segue nos referimos a todas essas técnicas como nomes criptografados). No entanto, além de não serem legíveis por humanos, os componentes de nome roteáveis criptografados fornecem privacidade muito limitada. Para manter um FIB em um tamanho gerenciável e aproveitar o cache do roteador e o colapso de interesse, a criptografia de cada componente de nome roteável deve ser efetivamente determinista. Isso implica que o adversário pode facilmente correlacionar interesses diferentes (alguns dos quais poderiam ser gerados pelo próprio adversário) com base no uso dos mesmos componentes de nome roteável. Como resultado, as informações vazadas pela parte roteável dos nomes NDN/CCN são análogas às vazadas atualmente pelas consultas DNS. Componentes não roteáveis permitem mais flexibilidade em termos de criptografia. Cada componente, ou sequência de componentes, pode, por exemplo, ser criptografado independentemente por cada consumidor, sob a chave pública do produtor, usando um esquema probabilístico, fornecendo assim fortes garantias de privacidade. No entanto, como resultado, o conteúdo não pode ser fornecido usando o cache da rede (exceto para retransmissão devido à perda de pacotes) e interesses não podem ser recolhidos em PITs, porque não há dois interesses para o mesmo conteúdo com o mesmo nome. Além disso, os

produtores devem assinar individualmente cada objeto de conteúdo solicitado por cada consumidor, porque a assinatura em um objeto de conteúdo abrange seu nome, que deve corresponder ao nome no interesse que o solicitou. Isso adiciona um custo considerável para os produtores em comparação com o TLS, em que as operações criptográficas de chave pública são executadas apenas no início de uma nova conexão, e não para cada pacote. Essas desvantagens podem ser resolvidas usando criptografia determinista, funções *hash* ou cadeias fixas (*pseudo-aleatórias*) para codificar componentes não roteáveis. No entanto, isso resultaria nas mesmas propriedades de privacidade fracas associadas aos componentes de nome roteáveis criptografados (GASTI; TSUDIK, 2018).

Privacidade do conteúdo: assim como na privacidade do nome, há uma tensão entre a distribuição eficiente de conteúdo e a privacidade do conteúdo. O conteúdo pode ser criptografado uma vez por seu produtor, e a chave criptográfica correspondente pode ser compartilhada com todos os consumidores pretendidos. A desvantagem desta abordagem é que o adversário pode facilmente determinar quais consumidores estão acessando um determinado objeto de conteúdo criptografado, vinculando os usuários a interesses semelhantes. Isso vaza substancialmente mais informações do que o TLS, em que o adversário não pode determinar se as informações trocadas como parte de duas conexões TLS se sobrepõem. Além disso, o sigilo de encaminhamento (suportado pelo TLS) é inatingível se o conteúdo for criptografado uma vez para todos os consumidores. Tal como acontece com a privacidade do nome, o roteamento em *onion* atenua parcialmente esse problema. Uma abordagem mais favorável à privacidade exigiria que os produtores criptografassem o conteúdo individualmente para cada consumidor. No entanto, o tráfego criptografado não se beneficiaria do armazenamento em cache. Também imporia sobrecarga de assinatura adicional aos produtores, uma vez que cada cópia criptografada do mesmo objeto de conteúdo teria que ser assinada individualmente. Resumindo, no que diz respeito ao vazamento de informações confidenciais, o NDN/CCN é potencialmente um retrocesso substancial em relação à privacidade e, possivelmente, ao desempenho, em comparação com o IP-com-TLS. Naturalmente, o IP-com TLS pode ser usado como uma sobreposição de NDN/CCN. No entanto, isso deve ser considerado, na melhor das hipóteses, uma medida paliativa, em vez de uma maneira de abordar a privacidade em NDN/CCN a longo prazo (GASTI; TSUDIK, 2018).

## 2.4. SIMULADORES E EMULADORES

Em computação, simuladores, são sistemas feitos para simular o funcionamento real de um *software* ou ambiente computacional, aplicação, etc, antes de a mandar para produção, testando assim a funcionalidade perante condições próximas a um ambiente real ou dependendo da eficácia do simulador, condições reais, tornando o processo de implantação do *software*, da tecnologia ou da arquitetura em simulação, mais adaptada a realidade e livre de erros o quanto for possível. Esses sistemas são muito eficazes porque testam na prática todas as funcionalidades do sistema desenvolvido, verificando assim se tudo funciona como deveria funcionar.

São sistemas baseados em *software* ou hardware, com propósito de ajudar a analisar um determinado problema em escalar menor; de certa forma, poupando tempo e dinheiro; no caso, em redes de computadores, possibilitam a simulação ou emulação de equipamentos físicos reais (TERTULINO, 2018).

Ou segundo Pedgen, simulação é processo de projetar um modelo computacional de um sistema real e conduzir experimentos com esse modelo com o propósito de entender seu comportamento e/ou avaliar estratégia para a sua operação (PEDGEN, 1990).

Segundo Filippetti, essencialmente, um emulador é um *software* criado para transcrever instruções de um determinado processador para o processador no qual ele está sendo executado. Um emulador é uma ferramenta que reproduz uma plataforma virtualizada que permite que uma dada arquitetura de computador consiga executar sistemas que foram desenvolvidos para outra arquitetura específica (FILIPPETTI, 2008).

Desse modo, um emulador permite que o usuário faça com que o seu computador pessoal aparente ser outra plataforma (como um *switch* ou roteador) para rodar outro sistema operacional. Os *softwares* emuladores, por sua vez, têm a capacidade de transformar um computador comum em um dispositivo de rede, como um roteador real ou um *switch*, replicando praticamente todas as suas funções. A limitação fica por conta do desempenho do elemento emulado, notadamente inferior ao de um elemento físico. Por este motivo, é desaconselhável a aplicação de elementos emulados em testes de desempenho. (FILIPPETTI, 2008).

No que diz respeito aos testes de rede, os termos emulação e simulação são frequentemente usados de forma intercambiável. Na maioria dos casos, qualquer um dos termos geralmente mostra o ponto, mas há uma grande diferença entre um emulador de rede e um simulador de rede, tanto de maneira prática quanto semântica. Como engenheiro de rede, um aplicativo configurado incorretamente pode custar muito tempo e dinheiro. A melhor maneira de tentar evitar esses infelizes acidentes é realizando testes completos e eficientes rotineiramente. Seja projetando uma rede, migrando para a nuvem ou adicionando um novo dispositivo ao rack, todas as etapas do ciclo de vida da implantação do aplicativo devem ser validadas com testes precisos. Um simulador pode realizar tarefas abstratas para demonstrar o comportamento de uma rede e seus componentes, enquanto um emulador pode copiar o comportamento de uma rede para substituí-la funcionalmente. Em um nível básico, um simulador de rede usa fórmulas matemáticas para criar um modelo teórico e inteiramente virtual de uma rede. Simuladores são soluções de software e diferentes tipos estão disponíveis para diferentes aplicações. Embora usados principalmente para fins educacionais e de pesquisa, eles também podem atuar como ferramentas de teste cruciais no projeto e desenvolvimento de uma rede. Simuladores, como o ns-3, são usados para simular protocolos de rede e roteamento. A OPNET, que foi adquirida pela Riverbed em 2012 e aplicada em sua linha de produtos SteelCentral, também forneceu um ambiente de simulação independente. Ambos os simuladores de rede usam simulação de eventos discretos, que cronologicamente enfileira e processa eventos como o fluxo de dados. Isso permite que um arquiteto ou engenheiro de rede construa e avalie um modelo experimental de rede, incluindo a topologia e o fluxo de aplicativos. Como uma variedade de cenários teóricos pode ser introduzida em uma rede onde qualquer coisa pode ser construída e aplicada, o desempenho pode ser hipotetizado antes que a própria rede tenha sido implementada no mundo real. Embora testar uma rede dessa maneira possa economizar tempo e dinheiro, os simuladores de rede não têm suas limitações. Essas operações altamente complexas exigem um grau de experiência e treinamento para configurar adequadamente a fim de adquirir resultados confiáveis. Além disso, os simuladores de rede não são práticos, pois determinados eventos não podem ser previstos independentemente de uma rede física. Um emulador de rede, é usado para testar o desempenho e funcionalidades de uma rede real. Esses dispositivos também podem ser usados para finalidades como garantia de qualidade, prova de conceito ou solução de problemas. Disponível como soluções de *hardware* ou *software*, um emulador de rede permite que arquitetos, engenheiros e desenvolvedores de rede avaliem com precisão a capacidade de

resposta, o rendimento e a qualidade da experiência do usuário antes de aplicar alterações ou adições a um sistema. Colocando-o fisicamente entre dois segmentos de rede local, um emulador de rede pode replicar com precisão uma conexão WAN cliente/servidor sem a necessidade de um roteador, modem ou mesmo tráfego ativo. Ele pode ser configurado para manipular restrições de largura de banda e aplicar deficiências, como perda de pacotes, atraso e instabilidade, à rede espelhada. A latência pode ser especificada para emular a transferência de dados em grandes distâncias e os aplicativos se comportam e respondem como se estivessem fisicamente separados. O desempenho dos aplicativos e a experiência do usuário final podem ser observados, testados e validados em tais condições em tempo real. As soluções de *software*, como a NetEm, que vem pré-empacotada no kernel do Linux, são ideais para testes com baixas taxas de dados, mas são limitadas pelas máquinas de teste nas quais são executadas (COLE, 2017).

A seguir serão apresentadas as duas ferramentas utilizadas para ilustrar essa pesquisa. Arquivos adicionais foram criados para que se o leitor se interessar ele possa acessar esses arquivos para verificar como cada uma dessas ferramentas são instaladas. Esses arquivos são encontrados no link:

[https://drive.google.com/open?id=1fbBjPv11dPCPjZtQ4BD7YQO\\_RS1J65t0](https://drive.google.com/open?id=1fbBjPv11dPCPjZtQ4BD7YQO_RS1J65t0)

#### 2.4.1. MINI-NDN

O Mini-NDN é uma ferramenta leve de emulação de rede que permite testar, experimentar e pesquisar na plataforma *NDN*. Baseado no Mini-CCNx que é um *fork* do *Mininet* (outro emulador de redes), o Mini-NDN usa as bibliotecas NDN, NFD, NLSR e ferramentas utilizadas no projeto NDN para emular uma rede NDN em um único sistema. O Mini-NDN é um *software* aberto e gratuito, licenciado sob a licença GPL 3.0 ([http://minindn.memphis.edu/mini\\_ndn\\_license.php](http://minindn.memphis.edu/mini_ndn_license.php)). O Mini-NDN é gratuito para todos os usuários e desenvolvedores. A primeira versão do Mini-NDN é desenvolvida por membros da equipe de projeto da NDN, patrocinada pela NSF. O Mini-NDN está aberto a contribuições do público (WHAT, 2018).

#### 2.4.2 ndnSIM

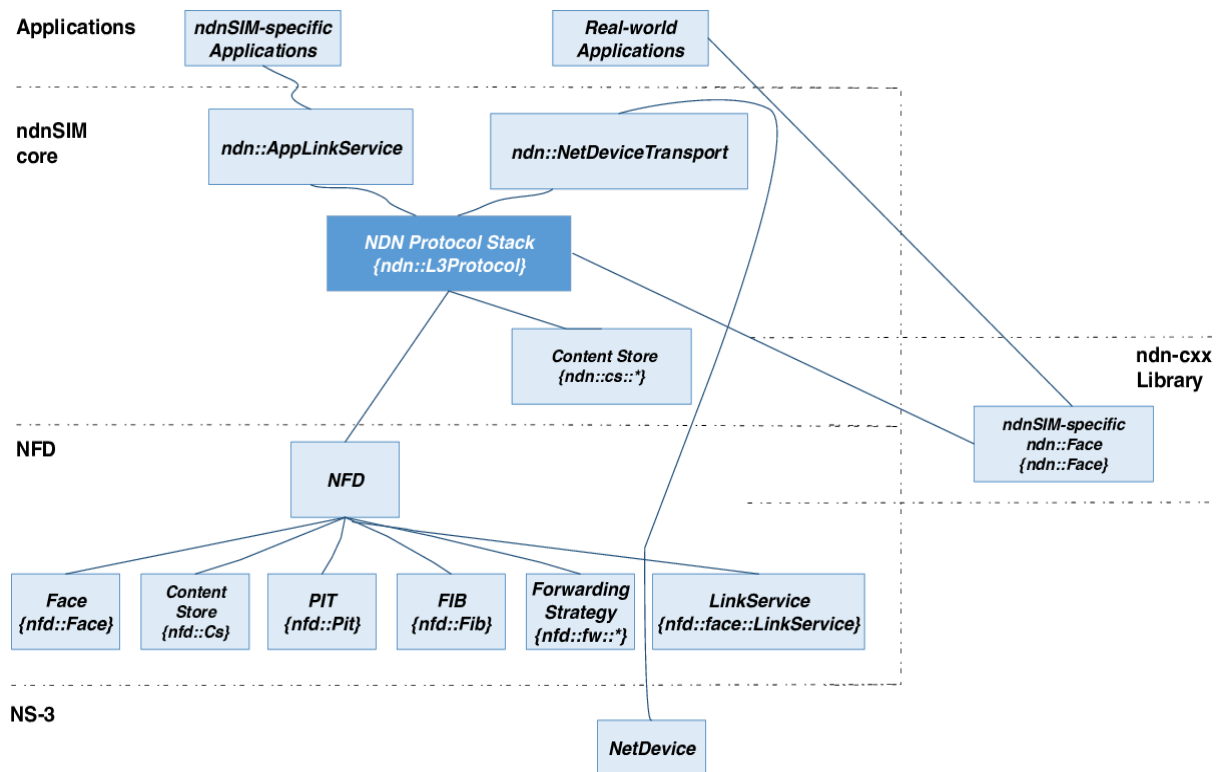
O ndnSIM é um pacote de simulação NDN modular, de código aberto, baseado no *framework* NS-3. O esforço de desenvolvimento do ndnSIM começou em 2011 e sua primeira versão beta foi lançada em fevereiro de 2012. Desde então, passou por mudanças substanciais de projeto e desenvolvimento extensivo, e tem sido usado por um número crescente de pesquisadores da comunidade de rede mais ampla. Ao longo dos anos, o ndnSIM serviu como um facilitador para um amplo escopo de experimentação com a arquitetura NDN. A versão mais recente do ndnSIM integra o NDN *Forwarding Daemon* (NFD) e sua biblioteca de suporte (ndn-cxx), fornecendo um nível de interoperabilidade entre simulação e prototipagem, aumentando ainda mais o valor da experimentação ndnSIM na compreensão do comportamento do encaminhamento e cache de rede. O ndnSIM também facilitou o desenvolvimento de aplicativos NDN, a exploração da aplicação de NDN em diferentes ambientes de rede (por exemplo, ad hoc *wireless*, *mobile* e IoT), os projetos de controle de congestionamento, a avaliação de protocolos de camada de *link* e roteamento, incluindo o protocolo de roteamento da arquitetura NDN, o NLSR. A estrutura geral do ambiente ndnSIM, consiste em NS3, NFD e ndncxx, bem como uma camada de simulação NDN, aplicativos ndnSIM específicos e reais portados para ndnSIM, e vários cenários de simulação *plug-and-play*. E também um conjunto de recursos que a torna uma ferramenta útil para a comunidade de pesquisa e apresenta o fluxo de trabalho de design discutindo o processo de troca de pacotes NDN entre dois nós simulados. O termo “código aberto” refere-se ao fato de que a base de código ndnSIM está disponível para o público e os usuários podem baixá-lo e modificá-lo com base em suas necessidades individuais. Os usuários são incentivados a participar do desenvolvimento do simulador. O termo “pacote de simulação” demonstra que o ndnSIM consiste em vários componentes de *software* que foram totalmente integrados para fornecer uma estrutura concreta para simulações de NDN de alta fidelidade. Em seu núcleo, o ndnSIM é baseado no *framework* de simulação NS3 e o aproveita das seguintes maneiras:

- cria topologias de simulação e especifica parâmetros de topologia (por exemplo, largura de banda de link, tamanho da fila de nós, atrasos de link).
- simula modelos de protocolo de camada de link disponíveis (por exemplo, ponto a ponto, sem fio, CSMA).
- simula a troca de tráfego NDN entre os nós simulados.
- rastreia eventos de simulação e (opcionalmente) visualizar a execução da simulação.



Portanto, as simulações ndnSIM podem usar qualquer um dos módulos, modelos, implementações do NetDevice e componentes integrados do NS3 existentes. Para realizar as principais funções de encaminhamento do NDN, o ndnSIM integra as bases de código NFD e ndn-cxx, reconectando os principais elementos lógicos, como o processamento de eventos e as operações de rede às rotinas específicas do NS3. O resultado dessa integração é que o código usado para experimentos com o redirecionamento de NDN no ndnSIM pode ser usado diretamente pela implementação real do NFD e vice-versa. Além disso, o ndnSIM permite simular os aplicativos NDN do mundo real baseados na biblioteca ndn-cxx. Além da integração com o NFD, o ndnSIM inclui uma camada adicional de simulação do NDN para agilizar a criação e a execução de simulações e obter as principais métricas. O pacote ndnSIM também oferece uma coleção de cenários de simulação tutorial que fornecem exemplos de recursos ndnSIM. (MASTORAKIS; AFANASYEV; ZHANG, 2017).

Figura 7 - Estrutura ndnSIM



Fonte 7 - MASTORAKIS; AFANASYEV; ZHANG, 2017.

## 2.5. TESTBED

*Testbed* é uma forma mais ampla de se testar algo produzido antes de o colocar em prática, em ambiente distribuído em rede. Utilizando um *testbed* é possível corrigir bugs e falhas antes que os usuários percebam. Essa forma de se testar um sistema é muito utilizada hoje em dia, um exemplo disso é a pesquisa sobre internet das coisas (IoT) aqui no Brasil, em que as indústrias envolvidas terão um financiamento de R\$15 milhões para investir justamente em testes, segue a matéria:

Empresas industriais terão apoio de até R\$ 15 milhões para experimentar o uso de internet das coisas (IoT), uma das tecnologias da chamada indústria 4.0, no processo produtivo. Os recursos serão aportados pelo Serviço Nacional de Aprendizagem Industrial (SENAI), pelo Banco Nacional de Desenvolvimento Econômico e Social (BNDES) e pela Empresa Brasileira de Pesquisa e Inovação Industrial (Embrapii). A chamada para seleção dos interessados em participar do programa será lançada nesta quinta-feira (8), às 9h30, na sede da Confederação Nacional da Indústria (CNI) em São Paulo. Os valores vão ser investidos na construção de ambientes de testes de soluções tecnológicas (*testbeds*), plataformas estruturadas em ambientes controlados que reproduzem um cenário real. Os recursos serão aplicados, por exemplo, em obras de infraestrutura de laboratórios, na compra de equipamentos nacionais, importados e de *softwares*, na remuneração da equipe, entre outras despesas necessárias para a realização dos projetos. Esses experimentos trazem benefícios para as empresas participantes, pois é possível reproduzir as condições específicas de seu ambiente fabril de forma otimizada sem paralisar a linha de produção. Além disso, há redução de riscos e custos de implantação de novas tecnologias. O prazo dos *testbeds* é de três anos, dos quais pelo menos dois anos serão de execução dos projetos. Os segmentos prioritários da chamada são as indústrias automotiva, têxtil, mineradora e de óleo e gás (BOAVENTURA, 2018).

A arquitetura que é tema dessa pesquisa (NDN) dispõe de um *testbed* ativo para observar o funcionamento do que se tem implantado até hoje e para identificar falhas e bugs.

O *testbed* da arquitetura NDN é um recurso compartilhado criado para fins de pesquisa, que inclui roteadores de *software* em várias instituições participantes, nós de *host* de aplicativos e outros dispositivos (NDN, 2018).

O *testbed* da arquitetura NDN possui 34 universidades participantes, dentre elas 1 brasileira, a Universidade Federal do Pará (UFPA). O processo do *testbed* da NDN é transparente e qualquer pessoa pode consultar o que foi constatado através dos testes realizados, desde sua implementação, até os dias de hoje, para isso basta acessar a área do *testbed* no site oficial da arquitetura (<https://named-data.net/ndn-testbed/>).

Para conectar um site (nó) ao *testbed* da NDN, é necessário estar ciente a estas políticas e, em seguida, entrar em contato com [ndntestbed@arl.wustl.edu](mailto:ndntestbed@arl.wustl.edu). Um *testbed* é vital para o sucesso do projeto NDN. A equipe da NDN mantém um *testbed* que inclui todas as instituições participantes do projeto. Um mapa do *testbed* atual pode ser visto em <http://ndnmap.arl.wustl.edu/>. A equipe recebe periodicamente solicitações de outras instituições para se conectar ao *testbed* da NDN. Enquanto os pesquisadores podem (e são encorajados a) criar seu próprio *testbed* da NDN, redes com um número significativo de nós são de grande valor tanto para a equipe da NDN quanto para a comunidade de pesquisa. Portanto, a equipe da NDN aceita solicitações de sites externos para se conectar ao *testbed* da NDN. No entanto, a fim de garantir a operação e a manutenção adequadas do *testbed*, é exigido que todos os sites (externos e internos) cumpram as políticas descritas abaixo:

Políticas para conectar-se ao *testbed* da NDN (PAPADOPOULOS; DEHART, 2019):

1. Forneça uma ou mais máquinas dedicadas no local para atuar como o (s) roteador (es) NDN do *gateway* no *testbed* da NDN. A (s) máquina (s) deve (m) ter um endereço IP que pertença à instituição. Se várias máquinas forem fornecidas para *multi-homing*, as mesmas regras se aplicam a cada máquina. O *gateway* não deve ser usado para outras tarefas além de sua função como um roteador NDN. As instituições *guest* podem ter qualquer número de outros nós e roteadores NDN controlados localmente por trás do roteador NDN do *gateway*; esses nós obterão conectividade com o *testbed* da NDN por meio do roteador de *gateway* NDN.
2. Forneça acesso *root* (ou *sudo*) à instituição de gerenciamento de teste da NDN (atualmente Universidade de Washington em Saint Louis, MO).
3. Permitir que a instituição de gerenciamento de teste da NDN instale, remova e atualize o *software* relacionado ao NDN conforme necessário para a operação e manutenção da rede.

4. Trabalhe com a instituição de gerenciamento de teste da NDN para solucionar qualquer problema de *firewall* e encapsulamento que possa surgir durante a instalação, configuração e operação do (s) roteador (s) NDN.
5. Forneça o nome, o *e-mail* e as informações telefônicas de um operador NDN específico do *site*, acessível com facilidade e rapidez, que será o principal contato sempre que surgirem problemas com o (s) roteador (s) NDN local. Um operador de *backup*, embora não seja obrigatório, é altamente recomendado. Os endereços de e-mail do operador devem pertencer à instituição. Todos os operadores locais serão obrigados a se juntar e monitorar regularmente uma lista de discussão do operador NDN e são obrigados a responder às solicitações da instituição de gerenciamento *testbed* o mais rápido possível, mas não mais que 24 horas.
6. Certifique-se de que a máquina tenha *logins* limitados além da conta da NDN. Uma conta de operador local é altamente recomendada, mas as contas de usuários gerais não devem estar presentes no roteador NDN. Além disso, somente o software relacionado à NDN deve ter permissão para ser executado na máquina.
7. A máquina pode ser uma caixa física dedicada ou uma máquina virtual (VM) com um endereço *IP* roteável publicamente (ou seja, não deve estar por trás de um NAT). Os operadores locais devem garantir que, se a máquina tiver limites de uso ou outras restrições (por exemplo, VMs em serviços de nuvem), elas não interferirão na operação adequada do *testbed*.
8. Todos os roteadores NDN do *gateway* executarão o mesmo sistema operacional, provavelmente uma versão do Ubuntu LTS. A versão exata será determinada pela instituição de gerenciamento testada. As atualizações do sistema operacional serão gerenciadas pela equipe de operações de rede da NDN. O operador local em um site será responsável pela instalação inicial do sistema operacional. Instruções específicas para essa instalação serão disponibilizadas.
9. *Hardware* Sugerido: Não há requisitos mínimos de *hardware* para roteadores NDN; no entanto, máquinas recentes são recomendadas e desejáveis.

10. Local sugerido: Idealmente, o roteador NDN deve residir em uma sala de máquinas em um *rack*; no entanto, um PC em um local seguro e com baixo tráfego, de preferência conectado a um *no-break* também será suficiente.
11. Conectividade de rede sugerida: não há requisito mínimo de largura de banda para roteadores NDN; no entanto, velocidades de rede mais altas são recomendadas e desejáveis.
12. As instituições participantes podem se desconectar do *testbed* ao dar aviso apropriado (pelo menos 24 horas) para a instituição de gerenciamento de teste da NDN. A mesma política se aplica se a instituição participante desejar desligar a máquina para manutenção e / ou atualizações de *hardware*. A notificação antecipada permitirá que a instituição de gerenciamento *testbed* tome as ações necessárias para evitar a interrupção desnecessária do serviço no *testbed*.
13. Desencorajamos e não aceitamos pedidos para ingressar no *testbed* da NDN se não houver intenção de usá-lo para pesquisa produtiva. Conectar-se ao *testbed* somente para estar conectado é uma violação das políticas atuais.
14. A instituição de gerenciamento de teste da NDN reserva-se o direito de recusar ou desconectar do teste qualquer nó que não esteja em conformidade com as políticas acima, engaje em comportamento abusivo, tenha operadores não responsivos e, em geral, considere não agregar valor ou ser um prejuízo para a operação geral do *testbed*.
15. Conexões de trânsito e *gateway* compartilhado são permitidas. Em outras palavras, uma instituição existente pode permitir que uma nova instituição participe do banco de testes da NDN por meio de uma conexão facilitada pela instituição existente. A nova instituição pode se conectar por meio do roteador de *gateway* da instituição existente ou por meio de um nó interno da instituição existente. Para garantir um bom funcionamento do *testbed*, a instituição existente será convidada a participar na resolução de quaisquer problemas que possam surgir com a nova instituição.

Se as políticas acima não puderem ser cumpridas, ainda incentivamos as novas instituições a entrar em contato com a equipe da NDN para discutir a conexão com o *testbed*.

A aprovação de conexão é exclusiva da instituição de gerenciamento de teste da NDN, mas todos os esforços serão feitos para acomodar as solicitações de conexão. À medida que o testbed amadurece e outros requisitos se tornam aparentes, as políticas acima podem mudar. Quando isso acontecer, as políticas serão anunciadas na lista de discussão do operador (PAPADOPOULOS; DEHART, S/D).

Nesse link <http://ndndemo.arl.wustl.edu/> é possível verificar o status de cada nó que participa do *testbed*.

Figura 8 - Status do testbed NDN

Note the new site location <http://ndndemo.arl.wustl.edu/ndn.html>

Other NDN status pages:  
[NDN Bandwidth Map \(currently nonfunctional\)](#)  
[NDN Testbed Cacti graphs](#)  
[NDN Routing](#)

NDN Testbed Snapshot: 2019/06/03 20:12:57 CDT  
 (Status updates every 10 minutes)

(Only the main site prefixes are shown)

Site Prefix Status: (Green: node has FIB entry for prefix; Red: no FIB entry; Yellow: no FIB entry but prefix is in node's domain)  
 Clock Skew Status: (As compared to UCLA Node's time: Green: < 5 secs off; Yellow: 5 < > 30 secs; Red: > 30 seconds off)

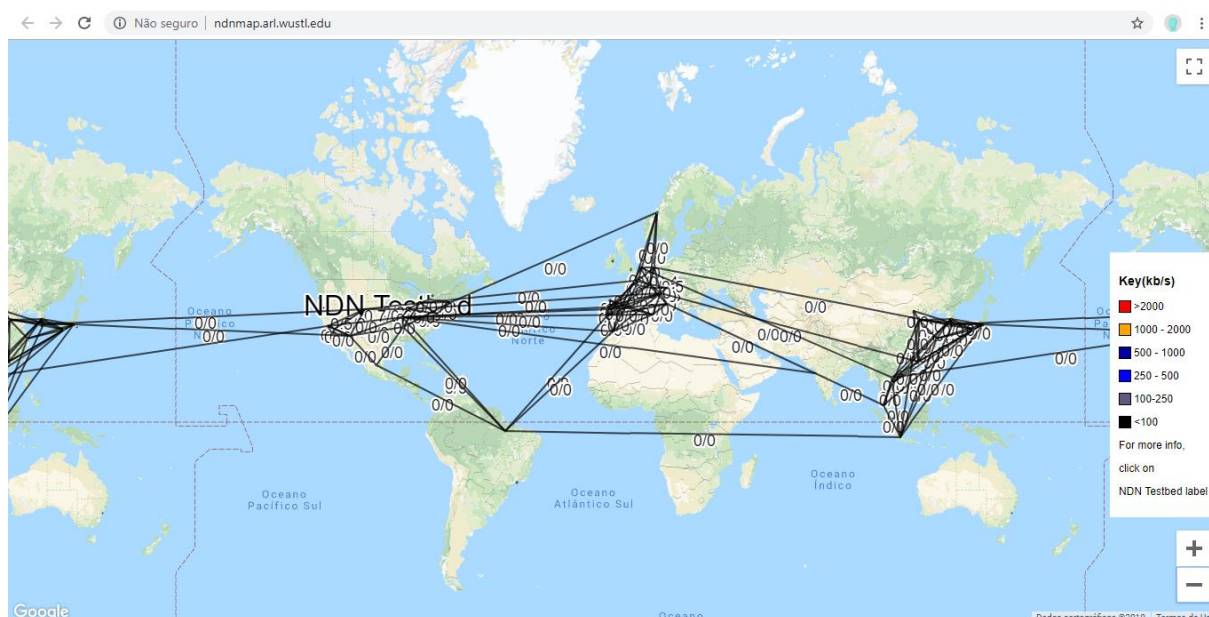
Notes on current (March 14, 2017) status: Hyperbolic Routing is now the default for NLSR on the Testbed.  
 Notes on current (Feb. 20, 2018) status: We are expanding what is reported at the top of this status table. Versions for NLSR, ndn-cxx and libchronosync are blank right now, they are coming soon.  
 Notes on current (Apr. 14, 2018) status: We are in the process of changing over to TLS/https access for the ndn status page. Some nodes will look down when they might not be.  
 Notes on current (July 24, 2018) status: Adding line to status page for TLS certificate expiry.  
 Notes on current (October 22, 2018) status: Updated to NFD 0.6.4.  
 Notes on current (November 1, 2018) status: We are having some serious problems with the new build of NLSR. Expect Testbed nodes to be up and down a lot while we test and debug.  
 Notes on current (March 11, 2019) status: We are upgrading NDN Testbed nodes to latest release of NFD and NLSR. This includes NLSR changes that are incompatible with previous version so things will be disjoint as they get updated.  
 Notes on current (March 13, 2019) status: A bunch of site certs need updating. Things will be bumpy for a while today..  
 Notes on current (April 7, 2019) status: We will be removing some long dormant sites.

	AAV	MUNBAL_ASY	OPA	SAT	BASEL	BERN	CNC	DDT	FRG2	TONO	GOETTINGEN	ARIZONA	OSU	MUC	MEMPHIS	SEL	ICI
OS Version	Ubuntu 16.04.6	Ubuntu 16.04.6		Ubuntu 16.04.6	Ubuntu 16.04.6	Ubuntu 16.04.6					Ubuntu 16.04.6	Ubuntu 16.04.6		Ubuntu 16.04.6	Ubuntu 16.04.6	Ubuntu 16.04.6	Ubuntu 16.04.6
NFD Version	0.6.6	0.6.6		0.6.6	0.6.6	0.6.6					0.6.6	0.6.6		0.6.6	0.6.6	0.6.6	0.6.6

Fonte 8 - <http://ndndemo.arl.wustl.edu/>

No link <http://ndnmap.arl.wustl.edu/> é possível ver todos os nós que participam do *testbed* da NDN, sua localização geográfica, o tráfego de cada um deles, e clicando em qualquer um o usuário será redirecionado a uma página com diversas informações sobre aquele determinado nó.

Figura 9 - Testbed NDN em forma de mapa mundi



Fonte 9 - <http://ndnmap.arl.wustl.edu/>

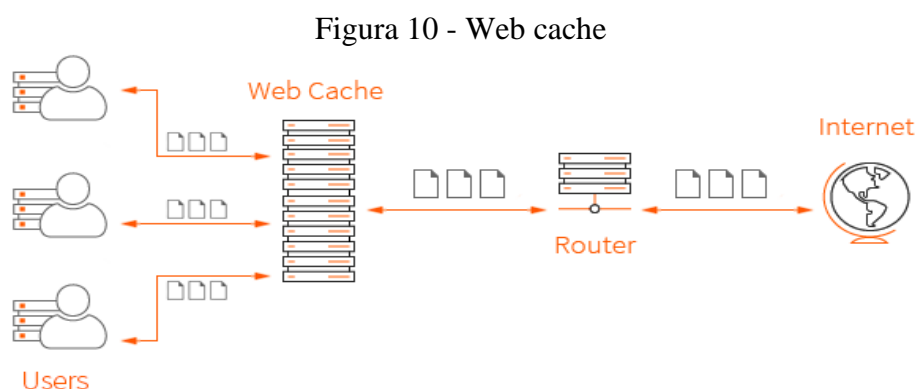
## 2.6. CACHE

O armazenamento em cache é uma das principais características nas redes centradas ao conteúdo, porque dessa forma o acesso a informações já buscadas se torna muito mais rápido. Já é sabido que a memória cache (CPU) é uma memória de muito rápido acesso (entre 10 e 25 nano segundos), mas que ao mesmo tempo custam muito caro, por isso os dados gravados na memória cache são dados muito importantes e ou dados requisitados com muita frequência. Uma característica desse tipo de memória é a volatilidade, ou seja, os dados nela gravados são perdidos com o tempo (interrupção de alimentação elétrica, por exemplo). O uso da memória cache hoje em dia é mais comum em CPU's, aplicativos, navegadores web e sistemas operacionais.

Em um computador a memória cache é a memória mais próxima do computador, logo a CPU busca as informações primeiro na memória cache, quando a informação buscada é encontrada na memória cache é gerada uma informação chamada cache *hit*, e a porcentagem de tentativas que resultam em cache *hit* fica conhecida como taxa de acertos ou proporção de cache. Pode acontecer de uma informação buscada pela CPU não ser encontrada na memória cache e isso é conhecido como cache *miss*, então ela é buscada na memória principal (que é uma memória de acesso mais lenta, mas de maior espaço de armazenamento) e logo em

seguida transferida para a memória cache, mas não são todos os dados que são encontrados na memória principal que serão transferidos para a memória cache, isso vai depender de qual algoritmo de cache está sendo utilizado.

Um exemplo de aplicação que usa bastante a memória cache são os navegadores *web*, os navegadores utilizam a memória cache para armazenar informações de um site que é muito visitado para que o próximo acesso a esse site seja mais veloz, por exemplo, supondo que um determinado usuário acesse muito o site do *facebook*, o seu navegador irá armazenar algumas informações do site *facebook* para que na próxima vez que esse usuário acessar o *facebook* a requisição não vá até o servidor do *facebook*, pois as informações já estarão armazenadas localmente em seu computador, na memória cache, fazendo com que a segunda requisição seja muito mais rápida que a primeira.



Fonte 10 - NDEGWA, 2016

O armazenamento em memória cache em *NDN* é utilizado juntamente com a nomeação de dados. Como cada pacote de dados da *NDN* é independente de onde vem ou para onde pode ser encaminhado, um roteador pode armazená-lo em memória cache de seu armazenamento de conteúdo (*content store*) para atender a solicitações futuras. Ao receber um novo interesse (*packet interest*), o roteador primeiro verifica o armazenamento de conteúdo. Se houver um dado cujo nome se enquadra no nome do interesse, os dados serão enviados de volta como uma resposta. O armazenamento de conteúdo, em sua forma básica, é apenas a memória de buffer no roteador atual. Roteadores IP e roteadores *NDN* armazenam pacotes de dados. A diferença é que os roteadores IP não podem reutilizar os dados depois de encaminhá-los, enquanto os roteadores *NDN* podem reutilizar os dados, uma vez que eles são identificados pelos nomes dos dados. Para arquivos estáticos, o *NDN* atinge uma entrega de dados quase ideal (latência perto de 0). Até mesmo o conteúdo dinâmico pode se beneficiar

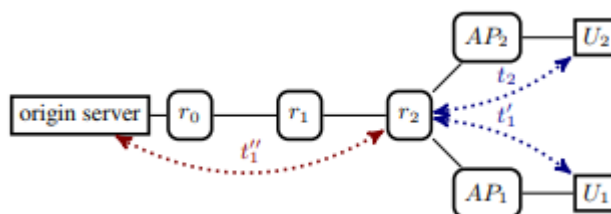


do armazenamento em cache, no caso de *multicast* (por exemplo, teleconferência) ou retransmissão de pacote após uma perda de pacote. O armazenamento em cache dos dados nomeados pode gerar preocupações com a privacidade. As redes IP de hoje oferecem pouca proteção de privacidade. Pode-se descobrir o que está em um pacote IP, inspecionando o cabeçalho ou a carga útil do mesmo, e quem solicitou os dados, verificando o endereço de destino. NDN nomeia explicitamente os dados, tornando mais fácil para um monitor de rede (*wireshar* por exemplo) ver quais dados estão sendo solicitados. Também é possível saber quais dados são solicitados por meio de esquemas inteligentes de análise para derivar o que está no cache. No entanto, o NDN remove completamente as informações sobre quem está solicitando os dados. A menos que esteja diretamente conectado ao *host* solicitante por um *link* ponto-a-ponto, um roteador saberá apenas que alguém solicitou determinados dados, mas não saberá quem originou a solicitação. Assim, a arquitetura NDN oferece naturalmente proteção de privacidade em um nível fundamentalmente diferente das atuais redes IP (NAMED, S/Da).

### 2.6.1 SEGURANÇA EM CACHE

Em arquiteturas CCN, uma vez que um pacote de dados é buscado de um servidor de origem, ele é replicado e armazenado em cache em todos os roteadores que fazem parte do caminho que esse pacote de dados percorreu. Por exemplo, quando outro usuário emite um pacote de interesse de um conteúdo que já foi buscado anteriormente o pacote de interesse é atendido a partir do próximo cache. Essa escolha de design obviamente é considerada uma grande vantagem no que se diz a respeito de latência global de recuperação de conteúdo. No entanto, esse mecanismo de cache universal (engloba toda Internet) representa um grande risco de privacidade: a diferença de tempo entre uma resposta de cache, quando comparada a um conteúdo que não foi proveniente de um cache pode ser usado como evidência para concluir se um usuário já tinha solicitado esse pacote de dados anteriormente.

Figura 11 – Comunicação Cache



Fonte 11 - MOHAISEN et al., 2013

Por exemplo, considere a topologia na Figura 8, que representa usuários  $U_1$  e  $U_2$ , e um conjunto de roteadores  $r_0$ ,  $r_1$  e  $r_2$  (cada um com seu próprio cache) conectando ambos os usuários a um servidor de origem que contém conteúdo com o nome  $N$ . Suponha que o usuário  $U_2$  seja o adversário, enquanto usuário  $U_1$  é honesto. Se  $U_1$  emite um interesse no conteúdo  $N$  que reside atrás de  $r_0$ , o interesse percorre o caminho  $U_1 \rightarrow AP_1 \rightarrow r_2 \rightarrow r_1 \rightarrow r_0$ , a partir do qual recupera o pacote requisitado do conteúdo. O pacote é então enviado de volta ao caminho de retorno  $r_0 \rightarrow r_1 \rightarrow r_2 \rightarrow AP_1 \rightarrow U_1$ . No total, o caminho do  $U_1$  para a fonte do conteúdo e o caminho de retorno para o  $U_1$  tem quatro *hops* cada. O tempo total de ida e volta necessário para enviar a solicitação até começar a receber pacotes de dados no caminho de retorno é  $t_1$ . Por outro lado, se  $U_2$  solicitar o mesmo conteúdo pelo seu nome,  $N$ , o caminho que o pacote de interesse iria percorrer é  $U_2 \rightarrow AP_2 \rightarrow r_2$ , e o conteúdo retornaria no caminho invertido ( $r_2 \rightarrow AP_2 \rightarrow U_2$ ), que é dois *hops* em cada direção, e exigiria um tempo  $t_2$ . Obviamente, o tempo  $t_1$  é maior que  $t_2$ , que um adversário  $U_2$  pode ser usado para inferir que o usuário  $U_1$  acessou o conteúdo  $N$ . Embora que identificar precisamente o  $U_1$  possa exigir um lado adicional de informações, um ataque como o descrito acima é crítico já que reduz muito o conjunto de anonimato desse usuário. Tal cenário é igual, em valor, à identificação de usuários individuais quando combinado com informações facilmente disponíveis. Por exemplo, um adversário que lança um ataque de *business intelligence* pode estar interessado em saber quais conteúdos estão sendo recuperados por uma empresa concorrente, e não por usuários individuais. Este ataque seria possível se o adversário estivesse localizado junto a essa empresa por trás de um roteador de borda e usando a técnica acima. Deficiências de soluções simples. Uma solução que é simples, mas que não pode impedir o ataque: por exemplo, um usuário pode marcar um objeto de conteúdo com um sinalizador de privacidade para desativar o armazenamento em cache da rede. No entanto, isso irá degradar a qualidade da experiência de outros usuários. Além disso, um adversário pode

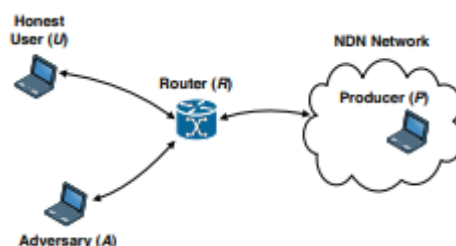
enviar duas solicitações consecutivas de mesmo nome para verificar o comportamento dos rastreadores. Na primeira solicitação, e supondo que o armazenamento em cache esteja ativado, os dados solicitados resultarão em armazenamento em cache de dados. O pedido nesse caso irá resultar em acerto de cache (cache *hit*), e o conteúdo será servido ao adversário rapidamente. No entanto, se o armazenamento em cache estiver desativado, o segundo pedido resultará em um atraso próximo ao atraso da primeira solicitação, a partir da qual o adversário pode perceber que o armazenamento em cache está desativado e que outro usuário provavelmente usou o sinalizador de privacidade. Na *information centric network* (ICN), o conteúdo é buscado por seus nomes. Uma ICN consiste em roteadores, onde cada roteador tem um cache, e roteadores de borda são conectados a usuários e servidores de origem. Um interesse na ICN encapsula um pedido de um pacote de conteúdo pelo seu nome. Um servidor de origem é um servidor que origina conteúdos a serem servidos na rede, assim interesses são satisfeitos. O conteúdo (pacotes de dados) pode ou não estar em cache na rede. Usaremos tempo total de viagem (RTT) para indicar o tempo desde o início do envio do primeiro interesse até o início do recebimento de um pacote de conteúdo (também conhecido na literatura como *Time to First Byte*; TTFB). Da mesma forma, definimos RTT por salto. Na ICN, os conteúdos são reenviados para um usuário no mesmo caminho que eles são solicitados por esse usuário, a PIT (tabela de juros pendentes) em cada roteador da ICN registra quais os juros que ainda não foram atendidos. Uma interface na ICN é a porta na qual os dados são enviados ou recebidos em um roteador. Em nossos protocolos, fazemos uso de ponto de acesso (AP), que é o ponto de conexão mais próximo do usuário para a ICN (não confundir com um ponto de acesso sem fio). Cada roteador mantém um conjunto de estados para registrar o número de vezes que um objeto de conteúdo sensível à privacidade foi buscado por cada usuário ou interface. *pmode* é um sinalizador para indicar que a privacidade de um nome de conteúdo (*interest packet*) acessados precisam ser preservados em acessos e solicitações futuros (MOHAISEN, 2013).

## 2.6.2. ATAQUE DE TEMPORIZAÇÃO EM CACHE

Nesta seção será descrito em mais detalhes o ataque de privacidade mencionado na seção anterior. A topologia NDN para nossa configuração de amostra é mostrada na Figura 10. Nela temos as seguintes entidades: (1) usuário U; (2) roteador NDN R; (3) produtor de

conteúdo P; e (4) adversário Adv. P é alcançável por U e o Adversário apenas através de R. Também assumimos que apenas U e o Adversário são atendidos por R como o seu roteador NDN sendo o primeiro salto (*first-hop*), essa topologia será melhor explicada mais à frente (ACS et al., 2013).

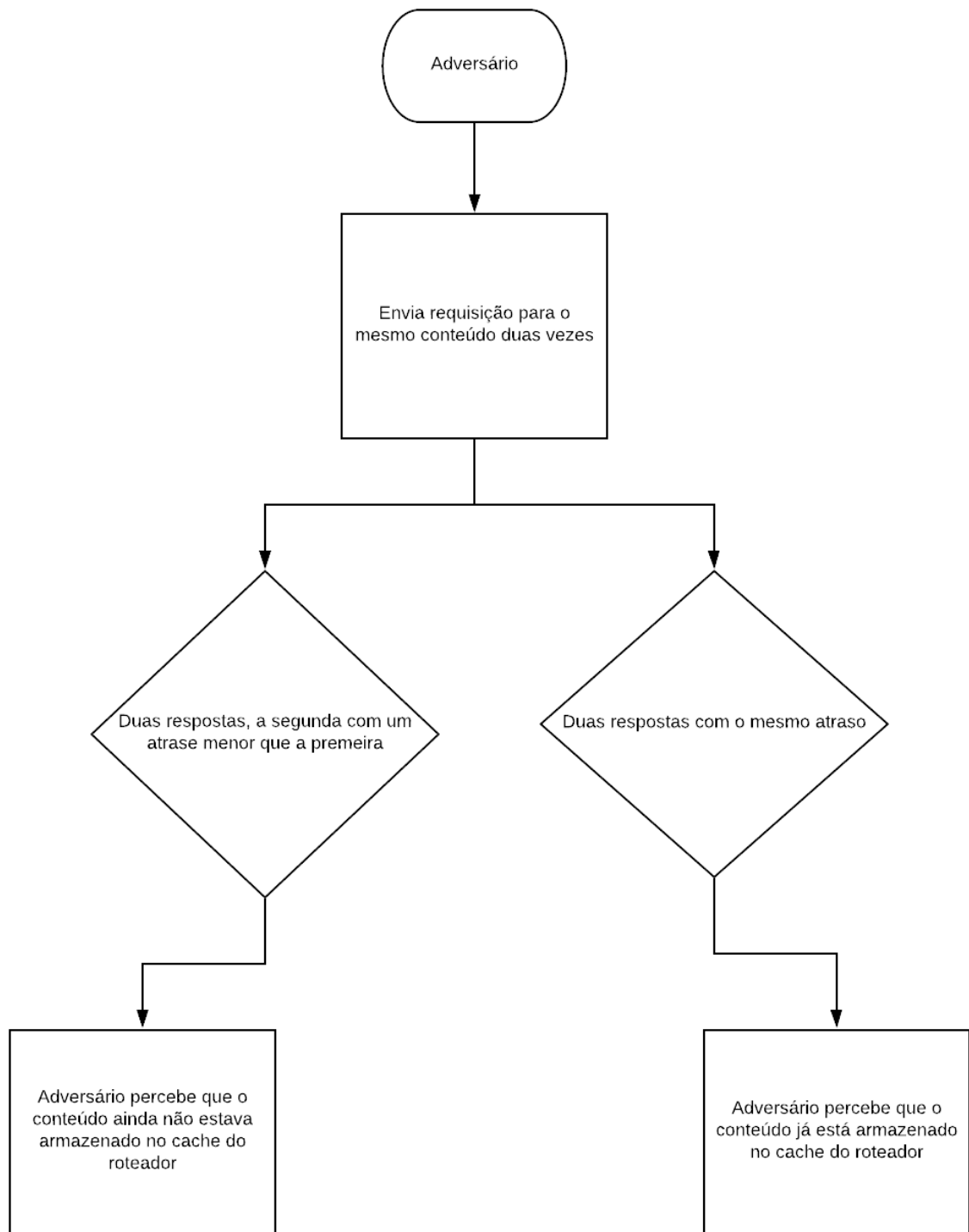
Figura 12 - Topologia NDN



Fonte 12 - ACS et al., 2013

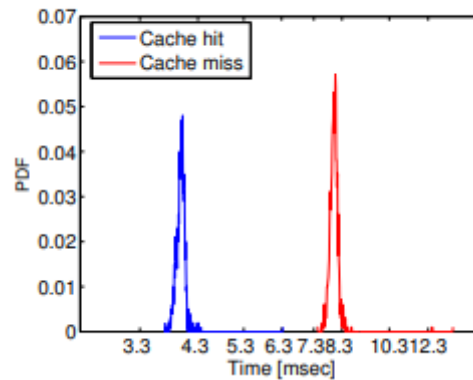
Nesta topologia, o Adversário pode saber se um conteúdo específico C foi solicitado recentemente por U ao investigar o cache de R. Para fazer isso, o Adversário emite um interesse por C e mede o tempo de atraso  $d_1$  necessário para obtê-lo. Em seguida, ele escolhe outro conteúdo (existente) C e solicita duas vezes consecutivas. A primeira vez, C pode ser obtido a partir de seu produtor original ou de algum cache do roteador (possivelmente, R). No entanto, a segunda vez, C é certamente obtido do cache de R. O tempo de atraso  $d_2$  é usado para denotar o atraso para o último. Se  $d_1 = d_2$ , o Adversário conclui que Usuário recentemente solicitou C. Considerando que, se  $d_1 > d_2$ , o Adversário decide que C não foi recentemente solicitado por ninguém deste lado de R (do roteador). Para validar o cenário acima, vamos realizamos alguns experimentos. Primeiro, P publicou 1.000 objetos de conteúdo. Em seguida, U, que está diretamente conectado ao R via Fast Ethernet, solicitou todos os objetos de conteúdo publicados. Isso gerou todo conteúdo a ser armazenado em cache pela R. Então, do Adversário pedimos os mesmos conteúdos que foram prontamente retirados do cache de R. Repetimos este experimento 50 vezes (todas as vezes começando com o cache de R vazio) e medido atrasos para recuperar o conteúdo de P e R. Na figura 12su é possível perceber a função de distribuição de probabilidade de atraso na resposta. Consequentemente, o Adversário pode determinar, com probabilidade mais de 99,9% se C é recuperado do cache de R (ACS et al., 2013).

Figura 13 - Diagrama descritivo do ataque relatado



Fonte 13 – PRÓPRIA, 2019

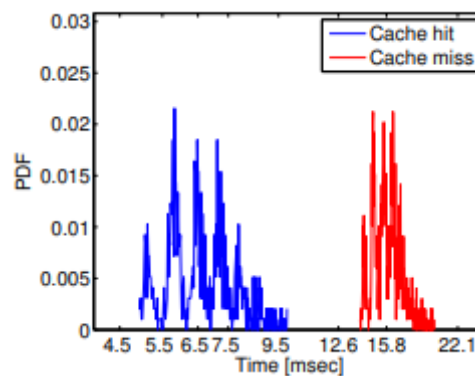
Figura 14 - Rede LAN



Fonte 14 - ACS et al., 2013

Em seguida, realizamos medições em uma topologia WAN, mas que roda sobre o NDN, onde U e o Adversário estão conectados ao mesmo roteador NDN de primeiro salto, que fica a vários saltos de ambos, enquanto P está a 3 saltos de distância de R (ACS et al., 2013).

Figura 15 - Rede WAN

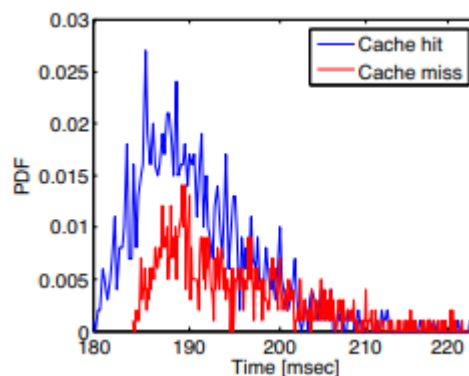


Fonte 15 - ACS et al., 2013

Claramente, a presença de saltos adicionais aumenta o atraso e introduz alguma variação nas medições. No entanto, ainda podemos determinar - com probabilidade acima de 99% - se C é recuperado do cache de R. Em nosso terceiro experimento (novamente no ambiente de teste da NDN), P está diretamente conectado a R, enquanto U e o Adversário estão a três saltos de distância. O objetivo do Adversário era determinar se o conteúdo C produzido por P foi solicitado recentemente. Para fazer isso, o Adversário busca C duas vezes e mede o atraso de cada vez. Os resultados da Figura 13 mostram que o Adversário pode

distinguir se C é servido de R com mais de 59% de probabilidade, sondando um único conteúdo objeto (ACS et al., 2013).

Figura 16 - Topologia WAN



Fonte 16 - ACS et al., 2013

### 2.6.3 ALTERNATIVAS PARA O ATAQUE APRESENTADO

Existem na literatura, algumas contramedidas para contornar esse problema de privacidade nos caches dos roteadores NDN, a seguir serão apresentadas duas contramedidas de dois autores diferentes, essas medidas serão comparadas e posteriormente uma será escolhida como uma possível solução para a falha apresentada.

Gergely Acs diz em seu artigo *Cache Privacy in Named-Data Networking* que, uma contramedida trivial e eficaz para a forma de ataque mencionada é simplesmente desabilitar o cache do roteador. No entanto, isso obviamente negaria imediatamente a distribuição eficiente de conteúdo, que é um dos principais benefícios do NDN. Claramente, nem todo conteúdo é privado. Para evitar a sobrecarga de ocultar conteúdo não privado, os consumidores e/ou produtores precisam dos meios para especificar qual conteúdo é sensível. Existem várias maneiras de fazer isso, dependendo de quem decide sobre a sensibilidade do conteúdo (ACS et al., 2013). São alguns exemplos:

- Decidido pelo produtor: um componente de nome reservado, por exemplo, *“/private/”*. Se estiver presente como (por exemplo) o último componente de um nome, o conteúdo em cache do mesmo nome é tratado como privado pelo algoritmo de armazenamento em cache do roteador. Alternativamente, um *bit* de privacidade especial - também

definido pelo produtor - no cabeçalho do conteúdo pode ter o mesmo efeito. Um consumidor emite um pacote de interesse como de costume, possivelmente permanecendo indiferente ao conteúdo ser privado (ACS et al., 2013).

- Definido pelo consumidor: um *bit* de privacidade especial nos pacotes de interesse, definido pelo consumidor. O conteúdo correspondente, quando armazenado em cache pelo roteador, é marcado de acordo assim como o pacote de interesse e é tratado como privado. O produtor pode ou não prestar atenção ao *bit* de privacidade, em parte dependendo se o interesse se propaga até o produtor (ACS et al., 2013).
- Mútuo: o conteúdo é referido por um nome imprevisível. Em outras palavras, se o conteúdo é privado, tanto o consumidor quanto o produtor se referem a ele por um nome que contém um componente único e difícil de adivinhar, idealmente derivado de algum segredo compartilhado. Uma característica interessante desta abordagem é a sua opacidade - os roteadores não precisam estar cientes da sua existência (ACS et al., 2013).

Estas três abordagens não são mutuamente exclusivas, por exemplo, mesmo se C (conteúdo) não for marcado (ou nomeado) como privado pelo seu produtor, pode ser solicitado como privado por um consumidor. As alternativas acima só correspondem aos meios de marcação de conteúdo privado. No entanto, eles não implicam ou incluem quaisquer ações por entidades da NDN (particularmente, roteadores) que precisem ser tomadas ao encontrar tal conteúdo privado. Para esse fim, adiante, serão apresentadas técnicas que impedem o Adversário de extrair informações significativas sobre o tráfego privado encaminhado dos caches dos roteadores. Ao projetar contramedidas, consideramos dois tipos de tráfego de rede: distribuição interativa e de conteúdo. O primeiro representa a comunicação síncrona entre duas ou poucas partes, por exemplo, *VoIP* e *shell* remoto. Esse tipo de tráfego é caracterizado pelo requisito de baixa latência e interação contínua. Em outras palavras, as partes em comunicação continuam desempenhando os papéis de produtor e consumidor. Considerando que, entrega de dados multimídia, transmissões ao vivo e entrega de páginas da *web* são exemplos de tráfego de distribuição de conteúdo. A lógica utilizada para distinguir esses dois tipos de tráfego em termos de contramedidas é discutida logo adiante. No tráfego interativo embora o cache do roteador NDN beneficie principalmente a distribuição de conteúdo, ele também ajuda a aliviar os efeitos da perda de pacotes na comunicação interativa. Isso ocorre porque os interesses emitidos para pacotes perdidos geralmente podem ser satisfeitos pelo



conteúdo armazenado em cache mais próximo do local da perda real, reduzindo, assim, o atraso do conteúdo solicitado novamente. Por esse motivo, qualquer mecanismo de armazenamento de dados com privacidade para essa classe de tráfego não deve introduzir nenhum atraso adicional. Ao mesmo tempo, como o conteúdo interativo tende a ser muito sensível ao tempo, quase não há benefícios de armazená-lo em cache em roteadores a longo prazo. Especificamente, se vários usuários participarem de uma videoconferência, os quadros de vídeo obsoletos em cache não serão úteis para nenhum deles. Escolhemos proteger essa classe de tráfego usando nomes imprevisíveis, ou seja, a abordagem mútua introduzida acima. Produtores e consumidores usam uma quantidade aleatória *rand* como o último componente do nome de cada conteúdo que eles criam e solicitam, respectivamente. Isso requer alguma coordenação entre as duas (ou mais) partes envolvidas na interação. Independentemente dos detalhes, as partes precisam concordar com um segredo compartilhado para semear uma função pseudoaleatória (por exemplo, um *hash* criptográfico com chave, como o HMAC) usados para gerar *rand* específico do nome do conteúdo. Aproveitamos nossa suposição anterior de que o Adversário não pode espionar os consumidores/produtores de conteúdo envolvidos na comunicação interativa ou no tráfego sobre os links incidentes do R (por exemplo, devido à criptografia do *link* ou à falta de acesso físico). Nomes de conteúdo imprevisíveis inibem o rastreamento malicioso do cache de R. Ao mesmo tempo, no caso de perda de pacotes, U pode reemitir um interesse e ainda se beneficiar da obtenção do conteúdo solicitado do roteador NDN mais próximo do local da perda de pacotes. Como mencionado anteriormente, essa abordagem mútua para marcar o tráfego privado pode ser combinada com abordagens orientadas para o produtor e/ou consumidor. No entanto, isso exigiria a participação do roteador (ACS et al., 2013).

Já Somaya Arianfar fala em seu artigo *On Preserving Privacy in Content-Oriented Networks* em uma solução que “embaralha o conteúdo, dificultando assim o entendimento do conteúdo solicitado”. A abordagem de Somaya Arianfar é um pouco diferente da abordagem de Gergely Acs (mas os dois tratam do mesmo problema, o problema de privacidade de cache na arquitetura NDN), Somaya faz uma abordagem mais “real”, na qual ela tenta impedir que um governo censure determinado conteúdo, utilizando para isto, o nome do conteúdo solicitado. O cenário e a ideia da solução propostas por Somaya serão descritos logo abaixo.

Consideramos um cenário em que um governo (ou qualquer adversário) está tentando impedir a disseminação de conteúdo sinalizado; em particular, eles querem bloquear os downloads desse conteúdo e/ou detectar quais usuários estão solicitando esse conteúdo. Há

três partes envolvidas: o governo, o editor do conteúdo sinalizado e os usuários do conteúdo sinalizado. Assumimos que o conjunto de usuários é grande e todo o fluxo de informações entre editores e usuários está aberto, sem coordenação detalhada entre editores e usuários individuais. Além disso, assumimos que o governo está interessado em bloquear a ampla disseminação de conteúdo (em vez de tentar simplesmente impedir sua entrega a poucos indivíduos). Além disso, assumimos que o governo quer interromper a entrega de conteúdo quase em tempo real; não pode se dar ao luxo de levar dias ou semanas antes de detectar que o conteúdo sinalizado foi entregue. Assim, para resumir, este é um problema de disseminação em massa e censura em massa em tempo real. Assumimos que todas as partes se conectam a uma rede pública na qual todas as solicitações de conteúdo (buscas) e entregas de conteúdo, podem ser observadas pelo adversário (governo). Nós nos concentramos em dois ataques. Em um ataque de lista de nomes, o adversário (governo) tem uma lista de nomes de conteúdo que deseja filtrar ou eliminar. Em seguida, ele se interpõe em *links* na rede que realizam filtragem em tempo real (o mesmo problema de privacidade abordado anteriormente por Gergely); se uma busca de conteúdo coincidir com T, o adversário (governo) pode silenciar a solicitação e/ou registrar o usuário que solicitou esses dados. Além disso, o adversário pode tentar excluir os dados com nomes nesta lista de alvos T. O ataque de lista de observação pode ser impedido pelo anonimato de consulta e dados - se for difícil para um adversário determinar se uma busca ou um conteúdo armazenado corresponde contra T, então é difícil para o adversário interferir efetivamente na disseminação desse conteúdo. Em um ataque de análise de conteúdo, o adversário não usa uma lista de observação pré-compilada, mas, em vez disso, inspeciona os dados para ver se deveria ter sido sinalizado (contém as palavras-chave erradas, etc.). Esse ataque pode ser frustrado ao fornecer uma negação plausível para os usuários (o que significa que eles podem alegar que os dados recebidos são plausíveis). Nosso objetivo é evitar os dois ataques acima em um cenário que corresponda de maneira próxima ao que acreditamos que as implantações realistas de redes orientadas a conteúdo possam parecer. Em particular, fazemos as seguintes suposições: Assumimos que a infraestrutura usada para armazenar dados (não queremos dizer caches na rede, mas os servidores nos quais os dados originais são armazenados) não é controlada pelos usuários, editores ou pelo governo. Ou seja, esperamos que uma ampla gama de domínios administrativos e políticos forneça infraestrutura de armazenamento sem controle central. Também assumimos que a infraestrutura de armazenamento é muito grande. Isso significa que o governo não pode apagar o conteúdo rapidamente (ele pode tomar medidas contra um pequeno conjunto de

objetos, mas rastrear dados em um conjunto diferente de máquinas será lento) e que os editores não podem impor certas regras de armazenamento nos sistemas de armazenamento resistentes à censura). Os editores são basicamente livres para nomear o conteúdo como acharem adequado. Assumimos também que os usuários e editores não compartilham informações secretas que possam ser usadas para iniciar a comunicação aprimorada com privacidade. Ou seja, os adversários saberão tudo que os usuários sabem (exceto quem são os usuários). Além de impedir a criptografia de chave compartilhada, isso também impede que os editores usem servidores secretos dos quais os usuários podem baixar dados; o governo saberá sobre esses servidores e poderá desativá-los. Nenhuma distribuição de chaves. Embora as chaves públicas para os principais editores possam ser amplamente conhecidas, presumimos que as informações relacionadas ao usuário (sua identidade ou chaves públicas) não sejam amplamente conhecidas no momento da publicação e não possam ser facilmente distribuídas, dificultando a realização de editores de conteúdo. Criptografia de conteúdo direcionado ao usuário (difusão). Considere um arquivo “alvo” que o adversário deseja censurar e que seja composto pelos blocos  $t_1, t_2, \dots, t_n$  e um arquivo “cover”  $c$  com os blocos  $c_1, c_2, \dots, c_m$ . Os nomes de conteúdo para todos os arquivos, capa ou destino são conhecidos de todas as partes. Assumimos que, embora os nomes dos arquivos não sejam limitados por uma estrutura específica, o nome de cada bloco é resultado direto ou indireto da codificação do arquivo / nome / conteúdo do bloco. Também assumimos que todos os blocos são de igual comprimento e que os arquivos são preenchidos com um múltiplo do tamanho do bloco. Esses blocos são misturados por produtores de conteúdo durante a etapa de publicação de conteúdo; um pedaço é o resultado da mistura de dois ou mais blocos de dados. Ou seja, um bloco é a parte do arquivo original e um pedaço é uma mistura de dois ou mais blocos.

#### 2.6.4 SOLUÇÃO ADOTADA

A solução adotada nesse trabalho para esse problema será a solução proposta por Gergely Acs. Gergely, que propôs uma solução relativamente simples, onde se você solicita um conteúdo sensível você pode solicitá-lo com um *bit* de privacidade, se você não sentir a necessidade de “proteger” sua solicitação a requisição pode ser feita naturalmente. Mas como na maioria das vezes o usuário não consegue discernir se o que foi solicitar é sensível ou não o próprio produtor do conteúdo pode adicionar o *bit* de privacidade ao conteúdo solicitado. A

solução proposta por Somaya Arianfar implementa uma espécie de criptografia ao conteúdo solicitado, essa solução criaria um overlay “estático” na arquitetura, adicionando assim uma parcela de atraso. Já a solução escolhida seria requisitada por demanda, sendo assim “dinâmica”.

### 3. PROCEDIMENTOS METODOLÓGICOS

Essa pesquisa tem como base a revisão bibliográfica com uma pesquisa exploratória.

A revisão bibliográfica é a base que sustenta qualquer pesquisa científica. Para proporcionar o avanço em um campo do conhecimento é preciso primeiro conhecer o que já foi realizado por outros pesquisadores e quais são as fronteiras do conhecimento naquela (VIANNA, 2001).

A revisão bibliográfica desse documento foi feita com base no *site* oficial da arquitetura a ser estudada, e em diversos outros *sites* que continham materiais acerca do assunto, observando sempre a credibilidade da fonte e do autor da informação.

Essa pesquisa também tem como metodologia a exploração.

Pesquisa exploratória é quando a pesquisa se encontra na fase preliminar, tem como finalidade proporcionar mais informações sobre o assunto que se vai investigar, possibilitando sua definição e seu delineamento, isto é, facilitar a delimitação do tema da pesquisa; orientar a fixação dos objetivos e a formulação das hipóteses ou descobrir um novo tipo de enfoque para o assunto. Assume, em geral, as formas de pesquisas bibliográficas e estudos de caso. (PRODANOV e FREITAS, 2013).

Essa pesquisa é de natureza exploratória com uma revisão bibliográfica. Durante essa pesquisa serão vistos os conceitos da arquitetura *NDN*, serão explorados todos os seus conceitos, mais precisamente os conceitos de segurança para que se possa no meio dessa exploração encontrar uma falha, para que no final do trabalho seja proposto uma solução teórica para essa falha encontrada.

## 4. CONCLUSÃO

Por se tratar de uma proposta para próxima geração com foco na solução de alguns problemas da arquitetura atual (Como os problemas de segurança relatados neste trabalho), a arquitetura *Named Data Networking* (NDN) tem a obrigação de diminuir ao máximo, ou dizimar grande parte desses problemas. Quando desenvolvida a ideia de uma rede orientada a conteúdo, o objetivo era melhorar o desempenho da rede mundial de computadores (Internet), e de fato, nesse quesito a ideia de redes orientadas ao conteúdo se mostra bem efetiva, a partir do momento em que se coloca, de forma padronizada, roteadores com cache, a disposição do “usuário”, a latência tende a diminuir drasticamente. Porém outro problema de muita relevância que uma rede orientada a conteúdo precisa corrigir é a segurança. Quando a Internet começou lá na segunda guerra mundial com a ARPANET, a segurança não era um problema a ser pensando, até porque estamos falando de um cenário controlado, com poucos *hosts*, mas com o crescimento da rede mundial de computadores algo deveria ser feito, foi então sendo elaborado soluções atrás de soluções para melhorar esse cenário, mas a verdade é que o “núcleo” da nossa atual arquitetura, não foi projetada tendo a segurança como um pilar.

A arquitetura baseada em conteúdo *Named Data Networking* (NDN), tem a proposta de desempenho firmada, mas em relação à segurança a arquitetura NDN tem alguns problemas a serem resolvidos. A ideia central de segurança da arquitetura, que é “blindar” o conteúdo ao invés do canal que o conteúdo irá transitar, em tese, se mostra melhor do que as soluções em segurança da nossa atual arquitetura, mas num cenário tão amplo como é nossa rede mundial de computadores, desafios sempre irão aparecer, aí entra o papel de nossa comunidade científica para encontrar esses problemas e os corrigir o quanto antes.

Em relação aos problemas de segurança que a arquitetura NDN possui, nessa pesquisa foi relatado o ataque de temporização em cache (*cache timing attack*), um ataque que é realizado de forma orquestrada e inteligente para descobrir quais conteúdos foram requisitados por alguma organização, para que provavelmente se consiga descobrir algum interesse dessa organização. Assim como esse problema já é conhecido, também é conhecido algumas soluções para contornar esse problema, e duas dessas soluções foram aqui, nessa pesquisa, apresentados.

A arquitetura *Named Data Networking* (NDN), tem bastante a oferecer, já está comprovado a sua superioridade em desempenho em relação a nossa atual arquitetura, e sua ideia de segurança também é superior à da arquitetura TCP/IP. Logicamente alguns ajustes

devem ser feitos, mas creio que com o apoio da comunidade as dificuldades encontradas serão facilmente contornadas. Outro ponto forte da arquitetura NDN é a comunidade ativa, que está sempre pesquisando e buscando formas de aprimorar essa solução. Devemos continuar pesquisando, questionando e buscando soluções para os problemas encontrados, para assim continuar contribuindo de alguma forma, tanto para a comunidade acadêmica, quanto para a comunidade científica, e assim tornar nossa Internet cada dia mais, um lugar mais cômodo e seguro para se navegar, e compartilhar nossos arquivos com segurança.

Para trabalhos futuros, é proposto a criação dos ambientes do ataque *cache timing attack* no mini-ndn e no ndn-sim para uma ilustração do ataque e das soluções propostas. Em linha, também propomos a implantação de mais *nodes testbeds* para uma maior divulgação acadêmica e avanço nas pesquisas sobre esta arquitetura de próxima geração.

## 5. REFERÊNCIAS

ACM Information Centric Networking Conference, Setembro 2016.

ACS, Gergely *et al.* **Cache Privacy in Named-Data Networking**. 2013 IEEE 33rd International Conference on Distributed Computing Systems, 2013. Disponível em: <http://sprout.ics.uci.edu/pubs/06681574.pdf>. Acesso em: 25 abr. 2019.

AFANASYEV, Alexander *et al.* **Content-Based Security for the Web**. [S. l.], 2016. Disponível em: <https://jhalderm.com/pub/papers/cbsec-nspw16.pdf>. Acesso em: 14 nov. 2018.

AMBROSIN, Moreno *et al.* **Covert Ephemeral Communication in Named Data Networking**. AsiaCCS '14, 2014. Disponível em: <https://arxiv.org/pdf/1311.2517.pdf>. Acesso em: 5 jun. 2019.

PUC RIO/CCE. **APOSTILA de “Internet e Arquitetura TCP/IP”**. [S. l.], S/D. Disponível em: <http://grsecurity.com.br/apostilas/TCP/tcp-apostila.pdf>. Acesso em: 17 dez. 2018.

BREACH LEVEL INDEX. Disponível em: <https://breachlevelindex.com/>. Acesso 20 outubro 2018.

BOAVENTURA, Helayne. **Indústrias terão financiamento de R\$ 15 mi para testes de internet das coisas**. 2018. Disponível em: <https://noticias.portaldaindustria.com.br/noticias/inovacao-e-tecnologia/industrias-terao-financiamento-de-r-15-mi-para-testes-de-internet-das-coisas/>. Acesso em: 08 jan. 2019.

CICN. [S. l.], 2018. Disponível em: <https://wiki.fd.io/view/Cicn>. Acesso em: 2 dez. 2018.

COLE, Zak. **Network simulation or emulation?**. [S. l.], 22 set. 2017. Disponível em: <https://www.networkworld.com/article/3227076/network-simulation-or-emulation.html>. Acesso em: 30 maio 2019.

DE BRITO, Gabriel M.; VELLOSO, Pedro B.; MORAES, Igor M. **Redes Orientadas a Conteúdo: Um novo paradigma para a internet**. 2012. Disponível em: <http://www2.ic.uff.br/~igor/cursos/files/BVM12.pdf>. Acesso em: 02 dez. 2018.



DiBenedetto, S.; Papadopoulos, C.; **Mitigating Poisoned Content with Forwarding Strategy**, Workshop on Name-Oriented Mobility: Architecture, Algorithms and Applications (NOM'2016), Abril 2016.

EL KAFHALI, Said; RAHEL, Safae; JAMALI, Abdellah. **Energy-efficient on caching in Named DataNetworking: A survey**. 2017. Disponível em: <[https://www.researchgate.net/publication/323067414\\_Energy-efficient\\_on\\_caching\\_in\\_named\\_data\\_networking\\_A\\_survey](https://www.researchgate.net/publication/323067414_Energy-efficient_on_caching_in_named_data_networking_A_survey)>. Acesso em: 09 dez. 2018.

GASTI, Paolo ; TSUDIK, Gene. **Content-Centric and Named-Data Networking Security: The Good, The Bad and The Rest**. 2018 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), 2018. Disponível em: [https://www.researchgate.net/publication/328608256\\_Content-Centric\\_and\\_Named-Data\\_Networking\\_Security\\_The\\_Good\\_The\\_Bad\\_and\\_The\\_Rest](https://www.researchgate.net/publication/328608256_Content-Centric_and_Named-Data_Networking_Security_The_Good_The_Bad_and_The_Rest). Acesso em: 5 jun. 2019.

HELLARD, Bobby et al. **What is GDPR? Everything you need to know, from requirements to fines**. [S. l.], 22 maio 2019. Disponível em: <https://www.itpro.co.uk/it-legislation/27814/what-is-gdpr-everything-you-need-to-know>. Acesso em: 5 jun. 2019.

**IP Spoofing**. [S. l.], S/D. Disponível em: <https://www.techopedia.com/definition/3993/ip-spoofing>. Acesso em: 4 jun. 2019.

J. WETHERALL, David; ANDREW S., Tanenbaum. **Redes de Computadores - 5ª Edição**. [S. l.]: Pearson, 2011.

JACOBSON, Van; AFANASYEV, Alexander; ZHANG, Lixia. **Named Data Networking**. 2014. Disponível em: <[https://named-data.net/wp-content/uploads/2014/10/named\\_data\\_networking\\_ccr.pdf](https://named-data.net/wp-content/uploads/2014/10/named_data_networking_ccr.pdf)>. Acesso em: 18 nov. 2018.

JACOBSON, Van. **Content Centric Networking**. 2010. Disponível em: <<https://wiki.tools.isoc.org/@api/deki/files/2634/=1.vj.isoc.mar10.pdf>>. Acesso em: 02 dez. 2018.

JACOBSON, Van. **Introduction to Content Centric Networking**. 2009a. Disponível em: <<http://bnrg.cs.berkeley.edu/~randy/Courses/CS294.S13/14.2b.pdf>>. Acesso em: 19 dez. 2018.

JACOBSON, V.; SMETTERS, D.K.; THORNTON, J.D.; PLASS, M.; BRIGGS, N.; BRAYNARD, R.; **Networking Named Content**. 2009b Magazine Communications of the ACM CACM Volume 55 Issue 1, Janeiro 2012.

JACOBSON, V., SMETTERS, D., THORNTON, J., PLASS, M., BRIGGS, N. e BRAYNARD, R. (2009c). **Networking named content**. Em **International Conference on emerging Networking EXperiments and Technologies - CoNEXT**.

KOPONEN, T., SHENKER, S., STOICA, I., CHAWLA, M., CHUN, B., ERMOLINSKY, A. e KIM, K. (2007). **A data oriented (and beyond) network architecture**. Em ACM SIGCOMM, páginas 181–192.

KOYAMA, Yasuhiro; KAGEYAMA, Yuki. **Memorandum of Understanding on Research Cooperation Signed with Institut National de Recherche en Informatique et en Automatique (INRIA), France**. 2014. Disponível em: <<https://www.nict.go.jp/en/info/topics/2014/12/141217-1.html>>. Acesso em: 18 nov. 2018.

L. Pi, L. Wang; **Secure Bootstrapping and Access Control in NDN-based Smart Home Systems**,  
Proceedings of IEEE INFOCOM 2018, poster, Abril 2018.

LIANG, T.; PAN, J.; ZHANG, B.; **NDNizing Existing Applications: Research Issues and Experiences**, ACM ICN 2018.

M. A. Filippetti. “**Uma Arquitetura para a Construção de Laboratórios Híbridos de Redes de Computadores Remotamente Acessíveis**”. Instituto de Pesquisas Tecnológicas do Estado de São Paulo – IPT, 2008.

MASTORAKIS, Spyridon ; AFANASYEV, Alexander; ZHANG, Lixia. **On the Evolution of ndnSIM**. [S. l.], 4 jul. 2017. Disponível em: <https://named-data.net/publications/ccr17-ndnsim/>. Acesso em: 30 maio 2019.

MOHAISEN, Abdelaziz *et al.* **Protecting Access Privacy of Cached Contents in Information Centric Networks**. [S. l.], 2013. Disponível em: <https://profsandhu.com/zhang/pub/asiaccs13-icn-privacy.pdf>. Acesso em: 19 mar. 2019.

MOSKO, Marc; SOLIS, Ignacio. **Content-Centric Networking: Architectural Overview and Protocol Description**. 2017. Disponível em: <https://arxiv.org/pdf/1706.07165.pdf>. Acesso em: 20 dez. 2018.

MUNIZ BANDEIRA DUARTE, Otto Carlos; G. LOPEZ, Norberto. **Redes de Computadores 1: IP Security**. 2003. Disponível em: [https://www.gta.ufrj.br/grad/03\\_1/ip-security/paginas/seguranca.html](https://www.gta.ufrj.br/grad/03_1/ip-security/paginas/seguranca.html). Acesso em: 17 dez. 2018.

NAMED Data Networking. **Named Data Networking: Motivation & Details**. NDN Project. S/Da. Disponível em: <https://named-data.net/project/archoverview/>. Acesso em: 15 julho 2018.

NAMED Data Networking. **NDN Project Overview**. NDN Project S/Db. Disponível em: <https://named-data.net/project/>. Acesso em: 15 julho 2018.

NAMED Data Networking. **NDN Project Specifications**. NDN Project S/Dc. Disponível em: <https://named-data.net/project/specifications/>. Acesso em: 12 Outubro 2018.

NDEGWA, Amos. **What is a Web Cache?**. [S. l.], 23 maio 2016. Disponível em: <https://www.maxcdn.com/one/visual-glossary/web-cache/>. Acesso em: 17 mar. 2019.

**NDN Testbed**. 2018. Disponível em: <https://named-data.net/ndn-testbed/>. Acesso em: 08 jan. 2019.

**NDN Frequently Asked Questions (FAQ).** [S. l.], S/D. Disponível em: [https://named-data.net/project/faq/#How\\_does\\_NDN\\_differ\\_from\\_Content-Centric\\_Networking\\_CCN](https://named-data.net/project/faq/#How_does_NDN_differ_from_Content-Centric_Networking_CCN). Acesso em: 31 maio 2019.

**NDN Protocol Design Principles.** [S. l.], S/Da. Disponível em: <https://named-data.net/project/ndn-design-principles/>. Acesso em: 4 jun. 2019.

**NLSR - Named Data Link State Routing Protocol.** [S. l.], S/D. Disponível em: <http://named-data.net/doc/NLSR/current/>. Acesso em: 4 jun. 2019.

PAPADOPOULOS, Christos ; DEHART, John. **Policies for Connecting Nodes to the NDN Testbed.** [S. l.], S/D. Disponível em: <https://named-data.net/ndn-testbed/policies-connecting-nodes-ndn-testbed/>. Acesso em: 1 jun. 2019.

PEGDEN, C. D.; SHANNON, R. E.; SADOWSKI, R. P. **Introduction to simulation using SIMAN.** McGraw-Hill, NY. 2 ed., 1990.

PRODANOV, Cleber Cristiano; FREITAS, Ernani Cesar de. **Metodologia do Trabalho Científico: Métodos e técnicas da pesquisa e do trabalho acadêmico.** 2. ed. Brasil: Universidade Feevale, 2013. 277 p.

**REDAÇÃO. Volume de roubo de dados cresce 88% em 2017.** [S. l.], 11 abr. 2018. Disponível em: <https://computerworld.com.br/2018/04/11/volume-de-roubo-de-dados-cresce-88-em-2017/>. Acesso em: 28 mar. 2019.

SATRIA, Muhammad Najib Dwi ; ILMA, Farchah Hidayatul; SYAMBAS, Nana Rachmana. **Performance Comparison of Named Data Networking and IP-based Networking in Palapa Ring Network.** The 3rd International Conference on Wireless and Telematics 2017, 28 jul. 2017. Disponível em: [https://www.researchgate.net/publication/328132910\\_Performance\\_comparison\\_of\\_named\\_data\\_and\\_IP-based\\_network-Case\\_study\\_on\\_the\\_Indonesia\\_higher\\_education\\_network](https://www.researchgate.net/publication/328132910_Performance_comparison_of_named_data_and_IP-based_network-Case_study_on_the_Indonesia_higher_education_network). Acesso em: 2 jun. 2019.

Shi, J.; Liang,T.; Wu, H.; Liu,B.; Zhang, B.;**NDN-NIC: Name-based Filtering on Network Interface Card**

STEVEN DIBENEDETTO, P. G. G. T. E. U. **ANDaNA: Anonymous Named Data Networking Application**, 2012. Disponível em: <<https://arxiv.org/pdf/1112.2205.pdf>>. Acesso em: 22 setembro 2018.

TERTULINO, RODRIGO. **SIMULADORES E EMULADORES DE REDES DE COMPUTADORES: ASPECTOS PRÁTICOS E FUNCIONAIS**. 2018. Disponível em: <<https://docente.ifrn.edu.br/rodrigotertulino/slides-da-apresentacao-na-campus-party-natal-2018>>. Acesso em: 08 jan. 2019.

VIANNA, ILCA OLIVEIRA DE ALMEIDA. **Metodologia do trabalho científico um enfoque didático da produção científica**. 1. ed. Brazil: EPU, 2001. 304 p.

VISALA, K., LAGUTIN, D. e TARKOMA, S. (2009). LANES: **An interdomain data-oriented routing architecture**. Em **Re-Architecting the Internet Workshop - ReARCH**, páginas 55–60.

Y, Yu.; A. Afanasyev.; D. Clark.; **Schematizing Trust in Named Data Networking**, ACM 2015.

Y, Yu.; A. Afanasyev.; L, Zhang.; **Name-Based Access Control**, 2015.

ZHANG, Zhiyi et al. **An Overview of Security Support in Named Data Networking**. 2018. Disponível em: <<https://named-data.net/wp-content/uploads/2018/04/ndn-0057-2-ndn-security.pdf>>. Acesso em: 03 jan. 2019.

**WHAT is a CDN**. [S. l.], S/D. Disponível em: <https://www.imperva.com/learn/performance/what-is-cdn-how-it-works/>. Acesso em: 4 jun. 2019.

**WHAT is Mini-NDN?**. 2018. Disponível em: <<http://minindn.memphis.edu/>>. Acesso em: 23 jan. 2019.

ZHUK, Sergey. **ReactPHP HTTP Server Middleware**. 2017. Disponível em: <<https://sergeyzhuk.me/2017/12/20/reactphp-http-middleware/>>. Acesso em: 09 dez. 2018.