



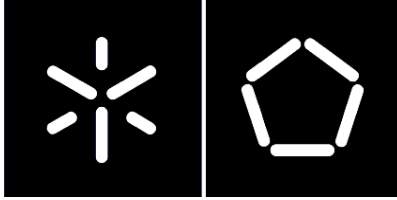
**Universidade do Minho**  
Escola de Engenharia

Paulo Alexandre Gomes Duarte

**Dados Nomeados para Redes  
Tolerantes a Atrasos**

Outubro de 2014





**Universidade do Minho**  
Escola de Engenharia

Paulo Alexandre Gomes Duarte

**Dados Nomeados para Redes  
Tolerantes a Atrasos**

Dissertação de Mestrado  
Mestrado Integrado em Engenharia de Comunicações

Trabalho efetuado sob orientação de  
**Professor Doutor António Luís Duarte Costa**  
**Professor Doutor Joaquim Melo Henriques de Macedo**

Outubro de 2014

## DECLARAÇÃO

Nome: Paulo Alexandre Gomes Duarte

Endereço eletrónico: a58655@alunos.uminho.pt

Telefone: +351913551581

Número de Cartão de Cidadão: 13942135

Título da dissertação: Dados Nomeados para Redes Tolerantes a Atrasos

Ano de conclusão: 2014

Orientadores:

Prof. Dr. Joaquim Macedo

Prof. Dr. António Costa

Designação do Mestrado: Mestrado Integrado em Engenharia de Comunicações

Departamento de Informática

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA DISSERTAÇÃO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

Guimarães, \_\_ / \_\_ / \_\_\_\_

Assinatura: \_\_\_\_\_



## AGRADECIMENTOS

O presente trabalho marca o final de um longo percurso académico, onde é imprescindível apresentar os meus sinceros agradecimentos a algumas pessoas.

Quero agradecer aos meus orientadores, Professor Doutor Joaquim Macedo e Professor Doutor António Costa, pelo esforço, dedicação e empenho, que me guiaram até ao fim desta dissertação. Foi um privilégio realizar esta dissertação sob a sua orientação.

Gostaria de agradecer a todos os meus amigos, que tiveram um papel fundamental ao longo do meu percurso, agradecendo a amizade e o apoio que demonstraram.

Obrigado à Marta pelo seu companheirismo, paciência e incentivo ao longo deste período e principalmente pela admiração entusiasta pelo meu trabalho.

Finalmente, nada disto teria sido possível sem o apoio dos meus pais. Agradeço os diversos sacrifícios suportados para concluir mais uma etapa da minha vida. Obrigado por me proporcionarem esta oportunidade e acreditarem em mim.



## RESUMO

As Redes Tolerantes a Atrasos (DTN) foram projetadas para permitir comunicações em cenários móveis sem fios onde a conectividade fim-a-fim não é possível. Os nós armazenam e transportam pacotes, decidem se os devem encaminhar ou não em cada contacto oportunista que eventualmente irão estabelecer no futuro. Recentemente, surgiram as Redes de Dados Nomeadas (NDN) como um novo paradigma para redes futuras. Os nós em vez de serem tratados como endereços de origem e destino, são vistos como consumidores que expressam interesses na informação de forma a obter dados referenciados pelo nome, ou produtores que fornecem informações. Este trabalho tem como objetivo combinação destas duas arquiteturas. Em particular, vai-se acrescentar a abordagem NDN a cenários DTN.

Adicionalmente, neste trabalho é proposto um novo protocolo de encaminhamento chamado PIFP (Probabilistic Interest Forwarding Protocol). Este protocolo explora a frequência de contactos oportunistas, entre nós e os dados, para calcular a probabilidade de satisfação de interesses, através da entrega dos respetivos dados. São descritos a conceção do protocolo proposto e a sua implementação no The ONE Simulator. Os resultados das simulações mostram que o PIFP apresenta melhorias significativas em termos de satisfação do interesse e o atraso médio e a carga de interesses quando comparado com outras abordagens centradas em dados nomeados recentemente propostas.





## ABSTRACT

Delay Tolerant Networks (DTN) were designed to allow delayed communications in mobile wireless scenarios where direct end-to-end connectivity is not possible. Nodes store and carry packets, deciding whether to forward them or not on each opportunistic contact they eventually establish in the near future. Recently, Named Data Networking (NDN) have emerged as a completely new paradigm for future networks. Instead of being treated as source or destination identifiers, nodes are viewed as consumers that express interests on information in order to obtain data referenced by name from producers that provide such data. Therefore, current research is carried on the combination of the two architectures. In particular, will be added an NDN approach to DTN scenarios. Additionally, this work proposes a new routing protocol called PIFP (Interest Probabilistic Forwarding Protocol). This protocol explores the frequency of opportunistic contacts between nodes and the data to calculate the probability of satisfaction of interests through the delivery of respective data.

The protocol design and a prototype implementation for The ONE Simulator are both described. Simulation results show that PIFP presents significant improvements in terms of interest satisfaction, average delay and charge of interest messages, when compared to other ND-DTN data-centric approaches recently proposed.



# Índice

<b>Introdução .....</b>	<b>1</b>
1.1 Objetivos .....	3
1.2 Metodologia .....	3
1.3 Principais Contributos .....	4
1.4 Estrutura do Documento .....	4
<b>Redes Tolerantes a Atrasos .....</b>	<b>7</b>
2.1 Arquitetura DTN .....	8
2.2 Protocolo de Agregação.....	10
2.2.1 Camada de Agregação .....	11
2.2.2 Agregados.....	11
2.2.3 Encapsulamento .....	12
2.2.4 Fragmentação.....	13
2.2.5 Controlo de Fluxo e Confiabilidade.....	13
2.2.6 Nós, nomes e endereços.....	14
2.2.7 Resolução do Nome .....	15
2.2.8 Conceito de Contacto .....	16
2.3 Encaminhamento em Redes Tolerantes a Atrasos.....	17
2.3.1 Encaminhamento Epidémico .....	18
2.3.2 PROPHET .....	19
2.3.3 Spray-and-Wait.....	20
2.3.4 Análise dos Protocolos .....	21
<b>Redes de Dados Nomeados .....</b>	<b>23</b>
3.1 Arquitetura das Redes de Dados Nomeados .....	24
3.2 Modelo de um nó NDN .....	25

3.3	Nomes.....	27
3.4	Segurança .....	28
3.5	Armazenamento .....	28
3.6	Mecanismo de expedição de Dados (Forwarding).....	29
3.7	Encaminhamento em Redes de Dados Nomeados.....	31
3.7.1	Named-data Link State Routing Protocol (NLSR) .....	32
3.7.2	COBRA: Lean Intra-domain Routing in NDN .....	33
<b>Trabalho Relacionado .....</b>		<b>35</b>
4.1	Trabalhos para aproximar as duas arquiteturas .....	37
4.1.1	Listen First Broadcast Later (LFBL).....	37
4.1.2	Information Centric Delay Tolerant Network (ICDTN).....	38
4.1.3	Broadcast-Only Named Data (BOND).....	38
4.1.4	Neighborhood-aware Interest Forwarding (NAIF) .....	40
4.1.5	Content-Centric Dissemination Algorithm (CEDO).....	40
4.1.6	Social-Tie based Content Retrieval (STCR) .....	42
4.2	Comparação entre as diferentes estratégias .....	44
<b>Conceção da Plataforma e do Protocolo de Encaminhamento .....</b>		<b>47</b>
5.1	Visão geral.....	47
5.2	Arquitetura.....	48
5.3	Esquema de nomeação .....	49
5.4	Estruturas de Dados .....	50
5.5	Mensagens .....	52
5.6	Faces .....	53
5.7	Protocolo PIFP .....	54
5.7.1	Funcionamento do protocolo .....	55
5.7.2	Troca de Informação .....	58

5.7.3	Correspondência de conteúdos .....	59
5.7.4	Processo de Consolidação.....	60
5.7.5	Encaminhamento de Pacotes de Interesse .....	62
5.7.6	Processamento de Interesses Gerados .....	63
5.7.7	Processamento de Interesses Recebidos .....	64
5.7.8	Processamento de Conteúdos Recebidos .....	65
5.7.9	Protocolo PIFP Proativo .....	66
<b>Implementação da Plataforma de Simulação ICONE .....</b>		<b>69</b>
6.1	Simulador ONE.....	69
6.1.1	Capacidade dos Nós .....	71
6.1.2	Modelos de Movimento .....	71
6.1.3	Encaminhamento.....	72
6.1.4	Geração de Mensagens.....	72
6.1.5	Relatórios e Visualizações.....	72
6.2	Desenvolvimento da Plataforma ICONE .....	73
6.2.1	Router NDN .....	74
6.2.2	Geração de mensagem de dados .....	78
6.2.3	Carregamento de Conteúdo .....	78
6.3	Protocolo PIFP .....	78
6.4	Mensagens NDN.....	80
6.5	Gerador de Mensagens NDN.....	81
6.6	Gerador de Resultados NDN.....	83
<b>Resultados da Simulação .....</b>		<b>85</b>
7.1	Configurações para a Simulação .....	86
7.2	Cenários de Simulação .....	87
7.3	Resultados Experimentais .....	88

7.3.1	Impacto da Densidade da Rede .....	88
7.3.2	Implicação do Tamanho da Mensagem .....	94
7.3.3	Consumo de Energia.....	99
7.3.4	Número de Saltos .....	101
7.3.5	Discussão dos resultados obtidos.....	103
	<b>Conclusões e Trabalho Futuro.....</b>	<b>107</b>
	<b>Bibliografia .....</b>	<b>110</b>
	<b>Anexos .....</b>	<b>115</b>

# Índice de Figuras

Figura 1 – Cenário DTN .....	9
Figura 2 – Protocolo de Agregação (adaptado de [13]) .....	11
Figura 3 - Funcionamento do encapsulamento do agregado [11] .....	12
Figura 4 - Envio de Agregado com Custódia (adaptado de [15]).....	14
Figura 5 - Diferença entre a pilha protocolar atual e a nova abordagem NDN (adaptado de [26]) .....	24
Figura 6 - Encaminhamento nas redes NDN.....	25
Figura 7 - Arquitetura de um nó NDN .....	26
Figura 8 - Estrutura em árvore do nome (adaptado de [27]).....	27
Figura 9 – Processo de encaminhamento NDN .....	29
Figura 10 - Processo de encaminhamento num nó NDN (adaptado de [47]) .....	30
Figura 11 - Nós Elegíveis (adaptado de [37]) .....	39
Figura 12 - Funcionamento do encaminhamento no STCR.....	43
Figura 13 - Modelo de um nó ND-DTN.....	48
Figura 14 - Pacote de Interesse (PI) .....	52
Figura 15 - Pacote Dados (PC) .....	53
Figura 16 - Troca de mensagens.....	55
Figura 17 - Registrar Conteúdo no Bloom Filter (baseado em [41]) .....	59
Figura 18 - Arquitetura do simulador ONE .....	70
Figura 19 - Arquitetura ONE adaptado às NDNs .....	73
Figura 20 - Estrutura do Pacote de Encaminhamento .....	75
Figura 21 - Dependência das Classes de Encaminhamento .....	77
Figura 22 - Diagrama de Sequência de Estabelecimento de Conexão.....	79
Figura 23 - Diagrama de Sequência da Recepção de uma Mensagem de Interesse.....	80
Figura 24 - Exemplo de sessão HTTP .....	81
Figura 25 - Número de Interesses por sessão.....	82
Figura 26 - Intervalo entre Interesses .....	82
Figura 27 - Número de Sessões .....	82
Figura 28 - Classe Leitor de Eventos .....	83



Figura 29 - Interesses Satisfeitos (Cenário 1).....	88
Figura 30 - Interesses Satisfeitos (Cenário 2).....	88
Figura 31 - Interesses Satisfeitos na rede e no nó (Cenário 1).....	89
Figura 32 - Interesses Satisfeitos na Rede e no Nó (Cenário 2) .....	90
Figura 33 - Atraso (Cenário 1).....	91
Figura 34 - Atraso (Cenário 2).....	91
Figura 35 - Carga de Interesses (Cenário 1) .....	92
Figura 36 - Carga de Interesses (Cenário 2) .....	92
Figura 37 - Carga de Dados (Cenário 1) .....	93
Figura 38 - Carga de Dados (Cenário 2) .....	93
Figura 39 - Interesses Satisfeitos (50-150k).....	94
Figura 40 - Atraso Médio (50-150k).....	94
Figura 41 – Carga Dados (50-150k).....	95
Figura 42 - Carga Interesses (50-150k) .....	95
Figura 43 - Interesses Satisfeitos (250-350k) .....	96
Figura 44 - Atraso Médio (250-350k).....	96
Figura 45 – Carga de Dados (250-350k).....	96
Figura 46 – Carga de Interesses (250-350k) .....	96
Figura 47 – Interesses Satisfeitos (750-850k) .....	97
Figura 48 – Atraso Médio (750-800k) .....	97
Figura 49 – Carga de Dados (750-850k).....	97
Figura 50 Carga de Interesses (750-850k) .....	97
Figura 51 - Interesses Satisfeitos vs Tamanho das Mensagens .....	98
Figura 52 - Energia Consumida.....	100
Figura 53 - Número de Saltos (são consideradas apenas as mensagens de dados entregues)	101
Figura 54 - Número de Saltos (são consideradas apenas as mensagens de interesse entregues)	102

# Índice de Tabelas

Tabela 1 - Tipos de Contactos em Redes Tolerantes a Atrasos .....	17
Tabela 2 - Comparação entre DTNs e NDNs [34] .....	36
Tabela 3 - Comparação dos trabalhos relacionados .....	45
Tabela 4 – Exemplo da Pending Interest Table (PIT) .....	50
Tabela 5 – Exemplo de Conteúdo da Content Store (CS) .....	51
Tabela 6 – Conteúdo da FIB .....	52
Tabela 7 - Excerto de um ficheiro para gerar Interesses .....	83
Tabela 8 - Parâmetros Gerais de Configuração .....	86
Tabela 9 - Configurações de Energia .....	99
Tabela 10 - Resultados da Energia Consumida .....	100
Tabela 11 – Resumo do número de saltos no encaminhamento de pacotes de dados .....	101
Tabela 12 - Resumo do número de saltos no encaminhamento de pacotes de dados.....	103
Tabela 13 -Síntese dos Resultados Obtidos .....	105

# Lista de Pseudo-Códigos

Pseudo-código 1 – Processo de consolidação dos dados armazenados.....	60
Pseudo-código 2 – Processo de consolidação de informações de encaminhamento .....	61
Pseudo-código 3 - Envio de Interesses Pendentes.....	62
Pseudo-código 4 - Geração de Novo Interesse .....	63
Pseudo-código 5 - Receção de um Pacote de Interesse.....	64
Pseudo-código 6 - Receção de Pacotes de Dados .....	65
Pseudo-código 7 - Receção de Pacotes de Dados versão Proativa .....	66

# Acrónimos

<b>Nome</b>	<b>Significado</b>
<b>ADU</b>	Aplication Data Unit
<b>BOND</b>	Broadcast-Only Named Data
<b>COBRA</b>	COntent-driven Bloom filter based Routing Algorithm
<b>DOCSIS</b>	Data Over Cable Service Interface Specification
<b>DSL</b>	Digital Subscriber Line
<b>DTN</b>	Delay/Disruption Tolerant Networks – Redes Tolerantes a Atrasos e Interrupções
<b>DTNRG</b>	Delay Tolerant Networking Research Group
<b>EID</b>	EndPoint Identifiers
<b>FIB</b>	Forwarding Information Base
<b>ICN</b>	Information Centric Networking
<b>ICONE</b>	Information Centric ONE
<b>LFBL</b>	Listen First, Broadcast Later
<b>LSDB</b>	Link-State Data Base
<b>MANET</b>	Mobile Ad-hoc Network – Redes móveis sem infraestrutura
<b>NDN</b>	Named Data Network – Rede de dados nomeados
<b>NLSR</b>	Named-data Link State Routing Protocol
<b>ONE</b>	Opportunistic Network Environment
<b>PDU</b>	Protocol Data Units
<b>PIFP</b>	Probabilistic Interest Forwarding Protocol
<b>PIT</b>	Pending Interest Table – Tabela de Pedidos Pendentes
<b>PROPHET</b>	Probabilistic Routing Protocol for Intermittently Connected Networks
<b>PSN</b>	Pocket Switched Network
<b>RTT</b>	Round-trip-time

<b>SSP</b>	Scheme-Specific Part
<b>TCP/IP</b>	Transmission Control Protocol/Internet Protocol
<b>TTL</b>	Time-to-Live
<b>URI</b>	Uniform Resource Identifier

# Capítulo 1

## Introdução

Apesar do grande sucesso da arquitetura atual da Internet, em determinados ambientes e contextos o perfil da arquitetura atual torna-se pouco robusto e inadequado. Ambientes como comunicações móveis ad-hoc, comunicações em áreas rurais, comunicações entre dispositivos em que a restrição é a energia, comunicações interplanetárias, entre outras, possuem características comuns como por exemplo as frequentes desconexões, as longas filas de espera, a alta latência e as baixas taxas de transferência. Ao longo de muitos anos, tem havido muitas tentativas para mitigar os efeitos de mobilidade correspondente a estes ambientes.

Muitos investigadores têm desenvolvido protocolos para superar esses desafios emergentes dentro da área das redes tolerantes a atrasos e interrupções (*Delay/Disruption Tolerant Network* -DTN)[1]. A maioria das propostas envolvem a alteração do paradigma atual da rede TCP/IP[2]. Para lidar com o alto grau de mobilidade e possivelmente rotas não conhecidas ou inexistentes, os nós seguem o paradigma de “armazenamento, transporte e envio de agregados” (do inglês, *store-carry-and-forward*) ao invés de confiar na conectividade fim-a-fim. Este princípio exige o armazenamento do conteúdo em *cache* até uma oportunidade de o transmitir ou replicar. No entanto, o desempenho destes protocolos está relacionado com os recursos disponíveis, principalmente a capacidade de armazenamento nos nós da rede e a largura de banda disponível durante o contacto com outro nó. As políticas de gestão de armazenamento já desenvolvidas foram formuladas e resolvidas no contexto fim-a-fim. A ideia deste trabalho é mudar o contexto fim-a-fim e aplicar o paradigma conhecido como publicação/subscrição na troca de informações entre nós da DTN de forma a melhorar o desempenho de entrega de informação.

O paradigma publica/subscreve é utilizado em arquiteturas centradas na informação, onde a preocupação é o conteúdo e os próprios dados e não os nós de rede e a sua localização. Este tipo de redes suporta uma melhor compreensão dos dados, permitindo assim que possam ser mais facilmente armazenados e reutilizados.

As redes centradas na informação podem ser vantajosas para ambientes de grande dinamismo onde os recursos limitados são gastos para manter a conectividade. Devido ao

paradigma destas redes e à capacidade de transmissão dos dispositivos de hoje, a recuperação de dados poderá ser melhorada fornecendo mais flexibilidade para manter a comunicação em ambientes altamente dinâmicos. Estas redes utilizam a *cache*, permitindo um nó móvel servir como meio de comunicação entre áreas onde a conectividade é escassa ou inexistente.

As redes de dados nomeadas (do inglês, *Named Data Networks* - NDN)[3] são um exemplo deste tipo de redes e são uma das propostas para a Internet do Futuro. Nas NDNs, os nomes são usados para exprimir tanto as necessidades de informação como a sua disponibilidade.

Apesar das diferenças, as DTNs e as NDNs possuem alguns aspetos comuns tais como o armazenamento em cache, a vinculação tardia dos nomes a locais, longevidade dos dados e o encaminhamento flexível.

A integração das duas arquiteturas poderá trazer várias vantagens para situações cotidianas, como por exemplo, durante o *roaming*, o transporte subterrâneo, áreas remotas, comunicações militares, comunicações interplanetárias, entre outras. O foco na informação poderá construir uma tremenda oportunidade para enfrentar de forma clara o egoísmo dos nós nas DTNs.

As propostas atuais tentam lidar com a incerteza se um nó vai ou não reenviar conteúdo armazenado, ou descartá-lo silenciosamente devido ao limite dos seus recursos ou por outra razão egoísta. Com o paradigma centrado na informação esta incerteza pode ser amenizada devido à melhor compreensão do conteúdo, permitindo desta forma saber se esta é relevante ou não para os nós. Nas redes centradas na informação os nós têm de publicar a informação que têm disponível em cache mas também os seus interesses na informação.

Em trabalhos anteriores foram analisados conjuntos de dados com informação de contacto de DTNs urbanas para perceber que oportunidades de contacto realmente existem, em outro, foram trabalhadas questões de energia[4], modelando a energia que é consumida enquanto um dispositivo procura ativamente por contactos. Também foram analisados aspetos de QoS para redes ad-hoc com rotas estáveis[5] e finalmente, foi investigada a utilização das redes sociais para melhorar os algoritmos de encaminhamento[6]. Recentemente foi iniciado um simulador onde foram explorados mecanismos centrados na informação para DTNs[7]. Este trabalho abre novas perspetivas para todo o esforço realizado anteriormente.

## 1.1 Objetivos

De acordo com a finalidade identificada no enquadramento, é possível definir um conjunto de objetivos a satisfazer ao longo da realização deste trabalho:

- Conceber e implementar um protocolo para ND-DTNs e compará-lo com outro para o mesmo contexto referenciado na literatura;
- Implementar um simulador de Dados Nomeados para DTNs utilizando como ponto de partida o simulador para DTNs, o ONE;
- Avaliar em que medida as redes centradas em informação e em particular as redes de dados nomeados se adequam a ambientes urbanos com DTNs.

## 1.2 Metodologia

Em primeiro lugar foi efetuada uma revisão da literatura NDN e DTN com o objetivo de perceber a distância entre as duas arquiteturas. Foi necessário identificar os componentes que é necessário incluir ou modificar numa DTN para introduzir dados nomeados e funções publicação/subscrição. Foram analisados simuladores DTN e NDN permitindo determinar com precisão as diferenças entre as duas arquiteturas de rede, tanto do ponto de vista teórico como do ponto de vista dos componentes que é necessário acrescentar e modificar.

Foi construído um ambiente de simulação denominado por ICONE (Information Centric ONE), utilizando como ponto de partida um simulador para DTNs, o ONE[8]. Este, é um simulador baseado em Java alvo de investigação em DTNs que permite simular diferentes cenários de uma forma rápida e flexível. Primeiro que tudo, foi desenvolvido e implementado um esquema de nomeação com base nas propostas NDN que seja adequada aos cenários DTN. O passo seguinte foi a colocação da pilha NDN no topo da DTN. Como é um simulador DTN, o nó DN é uma camada abstrata que inclui os principais componentes NDN nomeadamente FIB, PIT, *Faces*, *Content Store*, entre outros. Com este ponto de partida foi possível aumentar gradualmente a promiscuidade entre a DTN e a NDN no sentido de construir uma rede que integre as funcionalidades destas arquiteturas.



Uma vez construído o ambiente de simulação ICONÉ, que integra as DTNs e as NDNs o passo seguinte foi a adequação e o desempenho para os protocolos de encaminhamento adaptados de originais provenientes das duas arquiteturas.

### **1.3 Principais Contributos**

Esta dissertação tem como principal contributo um novo protocolo de encaminhamento denominado por PIFP[9] que explora a frequência de contactos oportunistas, não entre nós, mas entre os nós e a informação para calcular a probabilidade de entrega num cenário DTN.

Pretende-se também contribuir com um estudo sobre adequação das redes de dados nomeados em ambientes urbanos como as DTNs, e uma avaliação sobre influência das variações de características dos cenários como a densidade e o tamanho das mensagens.

Outro contributo é uma nova plataforma de simulação que consiste numa extensão para o simulador ONE fornecendo o suporte para dados nomeados. As contribuições desta tese permitem a conceção e implementação de novos protocolos de encaminhamento baseados em Dados Nomeados em cenários de Redes Tolerantes a Atrasos.

### **1.4 Estrutura do Documento**

Esta secção fornece uma visão geral sobre a estrutura do documento facilitando a leitura do mesmo.

O documento encontra-se dividido em 8 capítulos, iniciando por uma introdução ao tema, apresentando o enquadramento, os objetivos e a metodologia do trabalho. No segundo e terceiro capítulo é efetuada uma introdução às Redes Tolerantes a Atrasos e as Redes de Dados Nomeados. Os conceitos e componentes subjacentes a estas duas arquiteturas serão fundamentais para uma melhor compreensão do documento. Ainda são apresentados diversos trabalhos relacionados com o tema e uma relação entre as duas arquiteturas de forma a perceber as suas semelhanças e diferenças. No quarto capítulo são apresentados alguns trabalhos encontrados na literatura, que com este, desenvolvem metodologias para a aproximar as duas arquiteturas.

Seguidamente, no quinto capítulo, é efetuado um levantamento dos requisitos, é efetuada a descrição da arquitetura proposta e conceção e os algoritmos do protocolo de encaminhamento proposto.

O capítulo 6 descreve a implementação do ICONE. Através de diagramas UML são apresentados todos os módulos do sistema, facilitando assim a sua compreensão.

No capítulo 7 são apresentados todos os testes realizados ao sistema discutindo os resultados obtidos. Finalmente, o último capítulo diz respeito às conclusões da dissertação onde são discutidos os resultados e são apresentadas recomendações para trabalho futuro.



## Capítulo 2

# Redes Tolerantes a Atrasos

O modelo atual da Internet interliga inúmeros tipos de dispositivos em vários cenários. Este modelo é baseado na conectividade fim-a-fim, onde a confiabilidade da conexão entre a origem e o destino está assegurada através da flexibilidade, eficácia e vigor dos protocolos TCP/IP. No entanto, em alguns ambientes o atual modelo é ineficaz, como por exemplo, em ambientes onde a dinâmica da rede é elevada, comunicações onde existem limites de energia, regiões onde existem interferências, grande dispersão geográfica, entre outros. Manter a conectividade fim-a-fim nestes ambientes é uma tarefa bastante complexa devido às suas características.

As redes sem fios são um exemplo comum destes ambientes, onde as desconexões, as altas taxas de erros, as elevadas assimetrias de taxas de dados e os atrasos são frequentes.

Exemplos comuns de redes onde os protocolos convencionais não funcionam são as redes de sensores sem fios, redes móveis de sensores sem fios, redes de sensores e atuadores que são utilizadas em ambientes extremos como por exemplo zonas vulcânicas. Este tipo de redes utiliza técnicas de poupança de energia já que a energia é a sua maior limitação. Periodicamente os sensores permanecem desligados causando descontinuidades na comunicação.

Outro exemplo são as Redes Móveis Ad-hoc (MANETs)[10], que não possuem infraestrutura e são formadas por terminais móveis como *smartphones*, GPS e *tablets*. Estas redes trabalham em ambientes adversos onde mobilidade, as interferências através da obstrução ou destruição de nós como por exemplo num cenário de guerra poderão causar a desconexão.

Redes de comunicação exóticas, como as redes de satélites, comunicações acústicas estão sujeitas a altas latências e interrupções que podem ser previsíveis ou não. A latência na maioria dos casos deve-se às grandes distâncias entre os nós e as interrupções que podem ser causadas por fatores ambientais ou então devido ao seu modelo de movimento.

Estas redes apresentam requisitos próprios, sendo muitas vezes a comunicação homogênea devido a cada uma destas redes possuir protocolos próprios. Cada vez mais a troca de informação entre estas redes é inevitável, particularmente as redes móveis de sensores sem fios e a Internet. Foram desenvolvidos alguns dispositivos ou adaptadas algumas camadas que

permitem que estas redes operem com outras redes distintas. No entanto, a interoperabilidade ainda é bastante limitada e os desafios apresentados não são vencidos.

As redes que superam estes desafios são Redes Tolerantes a Atrasos (DTNs<sup>1</sup>). As DTNs permitem a interoperabilidade com outras redes, podem operar em ambientes onde existam atrasos longos e/ou variáveis e frequentes desconexões que provocam as falhas dos tradicionais protocolos utilizados para transmitir pacotes de dados.

## 2.1 Arquitetura DTN

A arquitetura DTN define uma camada orientada à mensagem que opera sobre a pilha protocolar das diversas redes. Esta camada permite a interoperabilidade com as diversas redes. Normalmente situada entre a camada de transporte e a camada de aplicação. O grupo DTNRG (Delay-Tolerant Networking Research Group) chamou a esta camada de camada de agregação<sup>2</sup> [11]. De forma a fornecer uma visão geral sobre o funcionamento das redes DTN é mostrado na Figura 1 um exemplo de utilização destas redes.

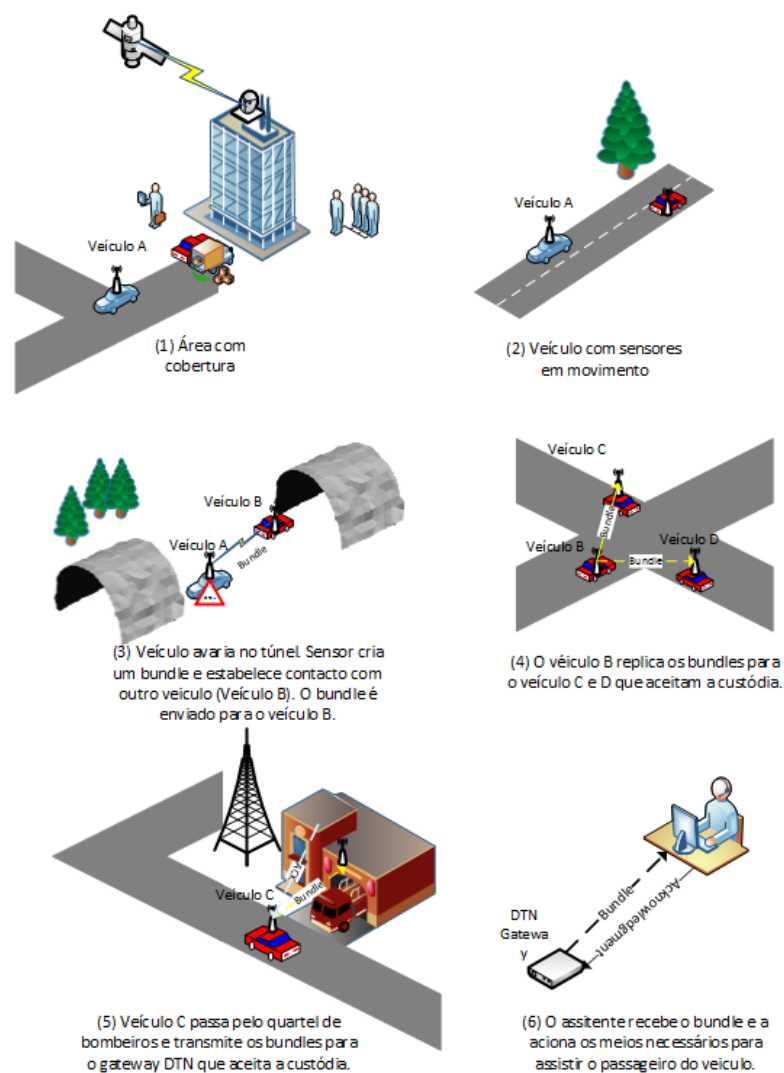
Como é possível visualizar, em (3) é demonstrado uma avaria de um veículo que se encontra num túnel (área com infraestrutura limitada) onde não é possível estabelecer comunicações. O passageiro precisa de ajuda para retirar o carro que está a obstruir a via. São então criadas unidades de dados de aplicação (ADUs) pedindo assistência que são convertidos pela cama de agregação em uma ou mais unidades de dados do protocolo chamados por agregados que são transmitidos pelos sensores sem fios na primeira oportunidade de contacto que surgir. Posteriormente, cada agregado poderá ser fragmentado sendo reagrupado no destino.

No ponto 3 o nó A, identificado por um EID (Endpoint Identifier), estabelece um contacto com o veículo B, também com um EID associado, e envia os agregados. B aceita a custódia, isto é, uma cópia dos agregados, sendo agora de sua responsabilidade armazenar esta cópia até receber uma confirmação que os agregados foram recebidos pelo destinatário (o protocolo *de* “agregação” segue o paradigma “armazenamento, transporte e envio de agregados”). B continuou no seu trajeto, deparando-se com outros nós (C e D) transmitindo os agregados para estes.

---

<sup>1</sup> Delay Tolerant Networks

<sup>2</sup> Bundle Layer no original em inglês



**Figura 1 – Cenário DTN**

Para que o agregado chegue o mais rápido possível ao destino, é possível adicionar informação de controlo, podendo conter várias métricas/restrições como por exemplo o nível de prioridade. Em (5), C chega ao quartel de bombeiros e os agregados chegam ao destino (DTN *gateway*). O DTN Gateway gera um ACK (*Acknowledgement*) indicando que admite a custódia do agregado e envia para o veículo C. Momentos mais tarde, este chegará ao veículo A. Neste ponto, o agregado chega a central de apoio dos bombeiros e então são acionados os meios para ajudar na avaria.

Como visto no exemplo, os problemas associados com conectividade, longos atrasos, taxas de transmissão assimétricas e altas taxas de erros podem ser superados através do paradigma “Armazenamento, Transporte e Envio de Agregados”. Este paradigma é muito parecido com o

método utilizado na Internet mas a diferença é que as mensagens podem ser armazenadas por um longo período de tempo (este período pode chegar até alguns dias).

Nas DTNs quando um nó fonte pretende enviar dados, estes são transformados pela camada de “agregação” e encaminhados para outros nós DTN. Os agregados são compostos por dados do utilizador e informação de controlo para entrega dos mesmos. A origem e o destino do agregado podem ser identificados por EIDs que identificam o nó fonte original e o destino final. Até ao agregado chegar ao destino, a probabilidade de existir uma interrupção na conexão é elevada e é neste aspeto que as DTNs se antecipam, armazenando de forma persistente os dados. Quando é detetado um contacto este é encaminhado respeitando outros agregados que já estão armazenados dependendo das suas características como por exemplo o grau de prioridade, o tamanho, as classes de serviço e outros tipos de métricas.

O paradigma “armazenamento, transporte e envio de agregados” assume que o armazenamento está disponível e é persistente o suficiente até os agregados sejam encaminhados. Também é necessário garantir que todos os nós permitam o armazenamento e a sua gestão. Com isto, passam a existir mais réplicas na rede aumentando a probabilidade de entrega dos agregados aos nós de destino.

## **2.2 Protocolo de Agregação**

As DTNs asseguram a entrega de dados em ambientes adversos que incluem aqueles com conectividade intermitente, grandes atrasos e altas taxas de erros. Para superar estes desafios as DTNs introduziram um novo protocolo que apoia a camada de aplicação no fornecimento dos seus serviços. Este protocolo é do tipo não conversacional, isto é, as mensagens de confirmação/erro de receção de mensagens são opcionais. Para lidar com ambientes extremos as classes de serviço poderão definir se essas são facultativas ou não. [11]

Os principais recursos do protocolo são: capacidade de lidar com conectividade intermitente, garantir a confiabilidade e a integridade fim-a-fim, conversão tardia dos nomes para endereços e transferência de custódia [12].

### 2.2.1 Camada de Agregação

As DTNs introduzem uma camada orientada à mensagem, como podemos visualizar na Figura 2, denominada por camada de agregação. Esta camada é adicionada à pilha protocolar da Internet ou de outros tipos de rede sobre a camada de transporte, pelo facto de que o protocolo de agregação requer a utilização dos protocolos existentes para a entrega de dados.

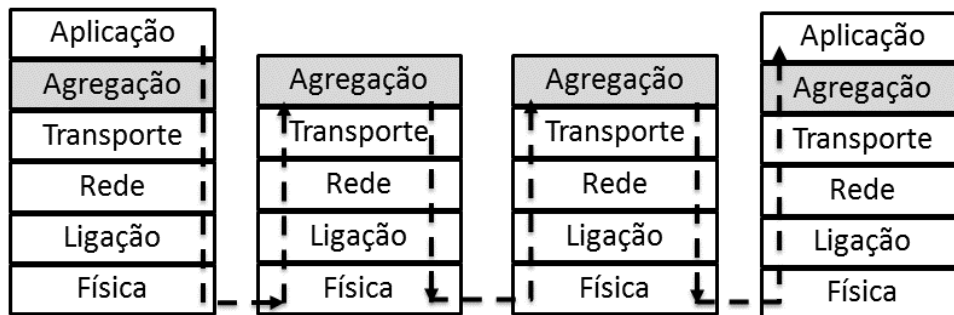


Figura 2 – Protocolo de Agregação (adaptado de [13])

É responsável por receber os pacotes enviados pela camada de aplicação e dividi-la em agregados (*bundles*) para que sejam enviados para a camada de transporte. No destino, a camada de agregação efetua o processo inverso, isto é, recebe os pacotes vindos da camada de transporte e efetua o reagrupamento para enviar os mesmos para a camada de aplicação. Este reagrupamento só se efetua no destino final.

### 2.2.2 Agregados

A camada de aplicação gera dados repassando estes para a camada de agregação que os transforma numa ou mais unidades de dados do protocolo (Protocol Data Units – PDUs) chamados de agregados. Cada agregado é constituído por dois ou mais blocos. O primeiro bloco contém informações básicas necessárias para encaminhar um agregado, como por exemplo a forma como processar os dados (condições de fragmentação), tipo de dados (dados de aplicação, administrativos), pedidos especiais (custódia, confirmação de receção, estado de entrega), tempo de vida do agregado, EID de destino e origem, classes de serviço, entre outros. Somente os blocos seguintes podem conter carga útil (dados aplicação, entre outros).



### 2.2.3 Encapsulamento

Após a transformação dos dados em agregados, estes são novamente encapsulados e desta vez pela camada inferior, camada de transporte, que adiciona informação de controlo de transmissão (segmento TCP, entre outros). A

Figura 3 exemplifica como é efetuado o encapsulamento do agregado relacionando as camadas inferiores da pilha protocolar da Internet. O agregado para as outras camadas é visto como se fosse um bloco de dados que são meramente encapsulados.

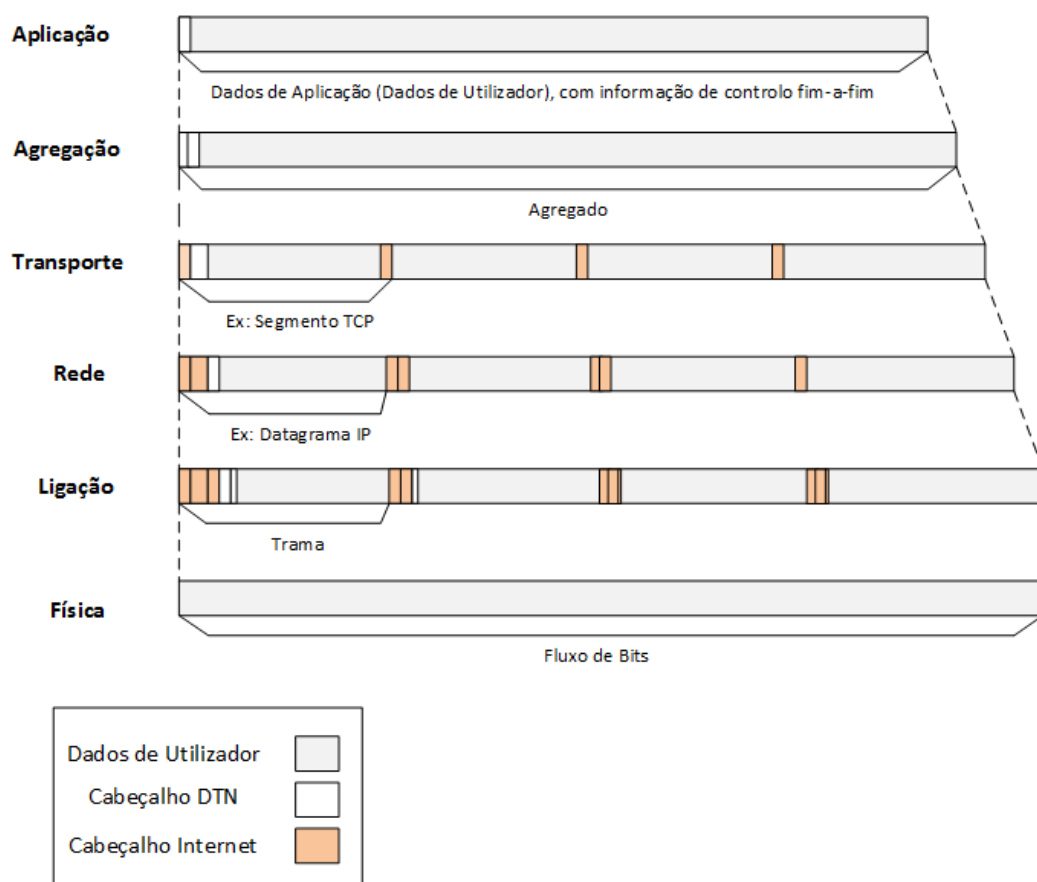


Figura 3 - Funcionamento do encapsulamento do agregado [11]

Como se pode visualizar na Figura 3 a camada de aplicação envia os dados de utilizador. Estes por sua vez são recebidos pela camada de agregação que insere o cabeçalho DTN. O agregado é enviado para a camada de transporte onde é encapsulado sob a forma de segmento TCP. De seguida o segmento TCP é encapsulado e transformado num datagrama IP pela camada de rede. Esta parte é a responsável pelas conexões entre as redes. Por último, os datagramas IP

são transformados em tramas pela camada de ligação. Esta camada permite a sincronização, a formatação dos dados, controlo de erros e outros.

Estamos a tomar como referência a pilha protocolar TCP/IP. Todavia podem ser usadas outras redes.

## **2.2.4 Fragmentação**

Devido à heterogeneidade presente entre determinadas regiões é necessário a adaptação do protocolo de agregação. Alguns fatores como a capacidade do canal, que por vezes leva a diminuir a taxa de transferência, e o grande tamanho dos agregados, que provoca atrasos de propagação, motivaram à fragmentação dos agregados

As DTNs definiram dois tipos de fragmentação [13][14], que são a fragmentação proactiva e a fragmentação reativa.

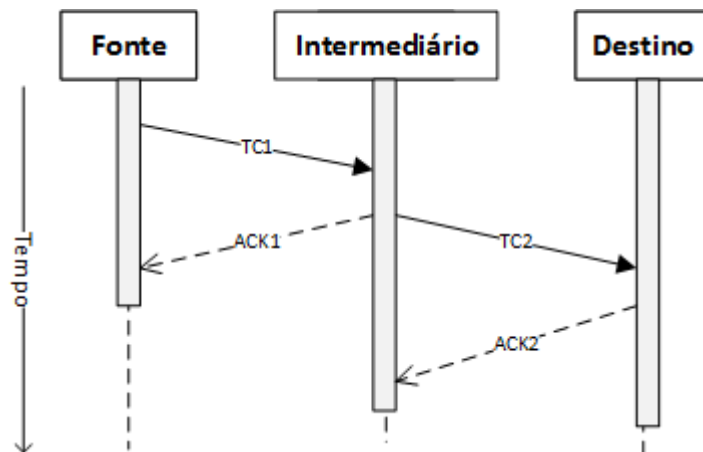
Na fragmentação proactiva a partição do agregado em partições mais pequenas é efetuada antes de conhecer quanto tempo irá durar o contacto e qual a capacidade do canal. Já na fragmentação reativa a fragmentação ocorre quando é conhecido pelas camadas inferiores que um agregado com um grande volume pode ser parcialmente transmitido.

No destino, é efetuado o reagrupamento de todos os fragmentos formando o agregado original. O reagrupamento é efetuado com base num cabeçalho específico que é adicionado a cada fragmento que indica o *offset* e o tamanho relativamente ao agregado inicial.

## **2.2.5 Controlo de Fluxo e Confiabilidade**

As decisões de controlo de fluxo e confiabilidade são efetuadas na camada de agregação. Uma das propostas para assegurar o controlo de fluxo é recorrendo ao mecanismo de transferência de custódia [14]. É um serviço fornecido pela camada de agregação que pode minimizar o descarte precoce de agregados e aumentar a confiabilidade fim-a-fim. Este mecanismo transmite a responsabilidade de entrega do agregado que ainda não chegou ao destino para outro nó. O nó fica responsável por manter o agregado armazenado de forma persistente, não podendo ser descartado a não ser que que a custódia seja transferida para outro nó ou então que o tempo de vida (TTL) do agregado expire.

Quando um nó envia um agregado para o próximo nó (que não é necessariamente o nó seguinte no percurso), ele solicita a transferência de custódia e inicia um temporizador. Tendo em conta os critérios definidos, quando o agregado é aceite pelo próximo nó, este retorna um ACK. No entanto, se nenhum ACK for retornado dentro do tempo permitido, o nó reenvia o agregado. O processo repete-se até chegar ao destinatário.



**Figura 4 - Envio de Agregado com Custódia (adaptado de [15])**

Os critérios de atribuição de responsabilidade da custódia é especificada pelos protocolos de encaminhamento e as suas políticas. No entanto é relevante avaliar a transferência de custódia mediante alguns critérios como por exemplo: energia suficiente para se manter ativo por longos períodos de tempo, espaço disponível para armazenamento, capacidade de sobreviver até à entrega do agregado e ser um nó mais próximo possível do nó de destino.

## **2.2.6 Nós, nomes e endereços**

Na arquitetura DTN existem diferentes tipos de nós, nós fonte, destino e agentes de transporte.

Em [11] os nós fonte e destino são nós que enviam ou recebem agregados de/para outros nós. Estes nós, caso operem em ambientes com interrupções e atrasos, requerem uma unidade de memória persistente, para armazenar os dados e posteriormente os retransmitirem.

Os nós denominados por agentes de transporte tem a função de encaminhar agregados entre dois ou mais nós. Existem dois tipos de agentes de transporte, os agentes que fornecem o encaminhamento com outros agentes que possuem a mesma pilha protocolar e os que fazem

encaminhamento com outros nós que possuem pilhas protocolares diferentes. Estes últimos agentes, fazem de *gateway* e obrigatoriamente tem de possuir armazenamento persistente e suporte à transferência de custódia.

Os nomes e os endereços são um dos aspetos mais importantes na arquitetura. Sendo esta baseada no paradigma centrado nos utilizadores é necessário endereçar todos os nós de maneira a realizar o encaminhamento entre nós e executar as mais variadas operações na rede. Na Internet estes recursos são mapeados pelo mecanismo *Domain Name System* (DNS) que converte os nomes em endereços. De modo similar, nas DTNs os nós são identificados por EIDs (Endpoint Identifier) que podem ser vistos como *URIs* (*Uniform Resource Identifier*) que identificam um recurso particular definindo a sua localização e como pode ser obtido [14].

Um EID corresponde a um nó ou a um grupo *multicast* (múltiplos nós de destino identificados pelo mesmo EID). Cada EID pode abranger vários nós e um nó pode ser membro de múltiplos EIDs. A forma geral de um DTN EID é:

**“<nome do esquema>: <parte específica do esquema>”**

Nas DTNs o nome do esquema por defeito é simplesmente “*dtn.*” e a parte específica do esquema é utilizado para especificar os nós (origem, destino, administração). A sintaxe utilizada para um esquema “*dtn*” segue a seguinte sintaxe:

**“*dtn*:://ID da máquina/ID da aplicação”**

Onde **ID da máquina** é uma palavra que identifica um nó DTN e **ID da aplicação** identifica uma aplicação que fornece serviços através da interface do nó.

## **2.2.7 Resolução do Nome**

A resolução do nome consiste na interpretação de um EID que identifica qual é o próximo nó a entregar o agregado até chegar ao destino final.

Uma base de conhecimento de ontologias e informação características é mantida em cada nó. Esta base de conhecimento é construída com base em informação histórica sobre a resolução de nomes.

Em [16] os autores sugerem que alguns nós possam ser definidos como gateways entre diferentes redes DTN, nomeadamente aqueles com maior poder computacional e com conectividade confiável podendo armazenar grandes quantidades de informação e o processamento da mesma.

Uma API de nomeação é utilizada nas DTNs para registar a aplicação e os nomes/atributos. Estas informações são armazenadas na base de conhecimento. As informações mantidas na base de conhecimento são disseminadas para os nós vizinhos.

A resolução do nome ocorre quando um agregado é recebido de maneira a identificar qual é o próximo nó.

De forma simplificada, a resolução do nome é realizada da seguinte maneira: um agregado recebido ativa um evento onde o *router* pede para o “Resolver” verificar os atributos referentes ao nome presente no agregado. O *router* envia a resolução do nome de volta para router de forma a este tomar uma decisão de encaminhamento. Se a resolução do nome for bem-sucedida, é escolhido o próximo nó com base nas políticas de encaminhamento.

### **2.2.8 Conceito de Contacto**

Ao contrário da Internet a possibilidade de comunicação é a exceção nas DTNs. Assim, a arquitetura DTN define o conceito de contacto. O contacto é quando um nó se encontra alcançável e é uma ocasião favorável para efetuar trocas de informação. Cinco tipos de contactos são considerados nas redes DTN: previsíveis, oportunistas, persistentes, programados, sob demanda. A Tabela 1 apresenta um resumo das características de cada tipo de contacto.

**Tabela 1 - Tipos de Contactos em Redes Tolerantes a Atrasos**

<b>Tipo de Contacto</b>	<b>Características</b>	<b>Cenários</b>
<b>Persistentes</b>	Conetividade persistente com garantias de confiabilidade.	DSL(Digital Subscriber Line), DOCSIS ( Data Over Cable Service Interface Specification)
<b>Programáveis</b>	É possível definir em que instante de tempo é viável estabelecer uma conexão para troca de dados.	Comunicações interplanetárias, espaciais, algumas redes sem fios.
<b>Sob Demanda</b>	Respondem a uma ação, até lá permanecem em modo “descanso” de forma a minimizar o consumo de recursos.	Redes sem fios.
<b>Previsíveis</b>	Baseados em previsões de tempo de contacto com base em históricos previamente observados. Existe um grau de incerteza.	Redes rurais onde a informação pode ser enviada numa “mula” como por exemplo um autocarro, correios, táxis, entre outros.
<b>Oportunistas</b>	Ocorrem de forma inesperada, isto é, não se conhece o instante nem o local onde vão ocorrer.	Pocket Switch Networks [17], Redes veiculares, entre outras.

## **2.3 Encaminhamento em Redes Tolerantes a Atrasos**

A conceção de estratégias de encaminhamento eficientes para redes oportunistas e redes tolerantes a atrasos é uma tarefa difícil devido ao não conhecimento sobre o comportamento da rede. Assim, nos últimos anos este aspeto tem sido alvo de investigação, resultando num conjunto de soluções com base no paradigma armazena, transporta e encaminha.

Atualmente os protocolos nesta área de investigação são classificados como sendo baseados no conhecimento onde existe apenas uma única cópia ou poucas cópias da mensagem na rede, ou então baseadas na replicação onde existem várias réplicas da mensagem pela rede.

Protocolos baseados em conhecimento têm vantagens óbvias como a economia de recursos em relação às abordagens baseadas na replicação. Foram propostas várias abordagens com base na replicação como por exemplo: a replicação gananciosa (em inglês “namely greedy replication” que consiste em entregar a mensagem a qualquer nó encontrado, como o protocolo Epidémico[18]), replicação controlada (onde existe um tipo de limite para o número de réplicas da mensagem, como por exemplo o protocolo Spray-and-Wait[19]). Por outro lado, forma definidas abordagens baseadas na utilidade, onde cada nó possui um valor que representa a viabilidade de transportar a mensagem para o destino. Os protocolos baseados em utilidade são divididos em dois grupos, *social aware* e *mobile aware*. Nas abordagens *social aware* os nós possuem conhecimento sobre as relações sociais com os outros nós, enquanto no modelo *mobility aware*, são utilizadas informações sobre os padrões de mobilidade dos nós para encaminhar as mensagens. Um exemplo de um protocolo *mobility aware* é o PROPHET[20], que utiliza os padrões de mobilidade para construir métricas para o encaminhamento de mensagens.

De seguida são apresentados os protocolos de referência para as redes DTN como o Epidémico, PROPHET e o Spray-and-Wait.

### **2.3.1 Encaminhamento Epidémico**

O protocolo de encaminhamento Epidémico, proposto por Vahdat e Becker, permite a entrega de mensagens a destinos arbitrários com pequenas suposições relativas ao conhecimento da topologia. Concebido para cenários estocásticos ou dinâmicos, onde o comportamento da rede é desconhecido, garante a entrega de mensagens mesmo que não exista um caminho direto entre a fonte e o destino.

Na ausência de contacto, as mensagens são armazenadas em *buffer* nos nós. São mapeadas em cada nó produzindo um vetor que permite identificar as mesmas. Este vetor é trocado quando se inicia um contacto, permitindo a ambos os nós identificar as mensagens que ainda não possuem. Caso um nó não possua alguma mensagem, é solicitada ao nó em contacto que lha envie. O processo de troca de mensagens é repetido sempre que existe contacto com outro nó. Cada mensagem contém um identificador global e único para determinar se esta já está armazenada ou se já foi encaminhada. Quantas mais mensagens forem distribuídas pela rede maior será a probabilidade de entrega e menor será o seu atraso.

Este protocolo implica custos elevados em termos de número de transmissões e espaço nos *buffers*. De forma a reduzir estes custos, em [21], foi proposto o limite do número máximo de saltos para cada mensagem. Cada mensagem contém um campo, similar ao TTL (Time To Live)<sup>3</sup> que conta o número de saltos<sup>4</sup>, limitando assim a utilização de recursos.

### 2.3.2 PROPHET

O PROPHET (Probabilistic Routing Protocol using a History of Encounters and Transitivity) [20] fundamenta-se no movimento dos utilizadores, empregando históricos de encontros e de transições. A partir destas informações, são determinadas as probabilidades que cada nó *origem* (*A*) possui para a entrega de mensagens a um nó destino (*B*) conhecido  $P_{(A,B)} \in [0, 1]$ . A previsibilidade de entrega, calculada a partir da Equação 1, aumenta em cada contacto com os nós.

$$P_{(A,B)} = P_{(A,B)antigo} + (1 - P_{(A,B)antigo}) \times P_{init} \quad P_{init} \in [0,1] \quad (1)$$

Esta é diretamente proporcional ao tempo decorrido ou seja, se os dois nós deixarem de se encontrar a probabilidade diminui. Para esse efeito utiliza-se Equação 2 onde  $\gamma = 0,98$  é a constante de envelhecimento e  $k$  o número de unidades de tempo (em segundos) que já passaram.

$$P_{(A,B)} = P_{(A,B)antigo} \times \gamma^k \quad \gamma^k \in [0,1] \quad (2)$$

Além desta propriedade, este protocolo possui a propriedade transitiva. Esta propriedade diz que se *A* se encontrar frequentemente com *B* e *B* se encontrar frequentemente com *C*, então *A* possui grandes probabilidades de entregar a mensagem a *C*. A transitividade pode ser obtida através da Equação 3 onde é introduzido o parâmetro  $\beta$  que indica a escala de impacto da transitividade na previsibilidade de entrega. Quanto maior o valor de  $\beta$  maior será o impacto da

---

<sup>3</sup> Tempo de vida em Português

<sup>4</sup> Ou o tempo de vida em horas, minutos e segundos



regra de transitividade. O valor da previsibilidade irá aumentar. O valor por defeito para  $\beta$  é de 0,9.

$$P_{(A,C)} = P_{(A,C)antigo} + (1 - P_{(A,C)antigo}) \times P_{(A,B)antigo} \times P_{(A,C)} \times \beta \quad (3)$$

Este protocolo mantém um vetor de identificação de mensagens idêntico ao protocolo epidémico, acrescentando agora as probabilidades associadas a cada mensagem. Esta lista é trocada entre os nós para verificar quais as mensagens que não possuem e a probabilidade de entrega para cada uma das mensagens. Apenas são replicadas as mensagens se for encontrado um nó com maior previsibilidade de entrega, limitando assim a replicação excessiva de mensagens.

### **2.3.3 Spray-and-Wait**

O protocolo Spray-and-Wait [19] tenta obter o melhor dos algoritmos de inundação, combina a velocidade de difusão destes com a simplicidade de entrega direta. Este protocolo visa reduzir a utilização de recursos como a utilização do *buffer* e a grande quantidade de tráfego originado pelo protocolo epidémico.

Para a entrega de mensagens são necessárias duas fases, a primeira (Spray) consiste na determinação do número de cópias (N cópias) que serão distribuídas pela rede e a distribuição das N cópias para os N nós intermediários. Numa segunda fase (Wait) é aguardado que uma das cópias chegue ao destino. Estas apenas são entregues de forma direta.

A primeira fase pode ser efetuada por dois procedimentos diferentes: o primeiro como já foi descrito, onde o nó origem fica responsável por transmitir todas as N cópias para os intermediários, e o segundo procedimento em que o nó origem envia metade das suas cópias (N/2) para o nó seguinte e decreta o número de cópias para metade. O nó seguinte repete o mesmo procedimento até que a quantidade de cópias seja igual a 1 entrando na fase de espera para entrega direta.

### **2.3.4 Análise dos Protocolos**

Nas subsecções anteriores foram apresentados os protocolos de encaminhamento de referência das DTNs: o protocolo Epidémico, o PROPHET e o Spray-and-Wait.

O protocolo Epidémico é o protocolo mais comum e mais simples. Em alguns cenários é impossível obter informações sobre o estado da rede, o que impossibilita o cálculo de melhores rotas. Neste protocolo as mensagens são replicadas para todos os nós vizinhos. Garante a entrega de praticamente todas as mensagens quando a capacidade de armazenamento é ilimitada. Os maiores problemas do protocolo Epidémico é o alto custo em termos do número de transmissões e o espaço de armazenamento. Desta forma, este protocolo não é escalável.

Enquanto o protocolo Epidémico replica as mensagens para os nós vizinhos, o protocolo PROPHET encaminha com base em estimativas (probabilidades) de alcançar o destino. O PROPHET é um protocolo que utiliza a informação histórica para transmitir os pacotes. Este protocolo obtém melhores resultados do que o protocolo Epidémico quando existem limitações de armazenamento pois as mensagens são enviadas somente para os melhores nós. Este protocolo permite a redução de tráfego na rede e garante uma grande taxa de mensagens entregues com sucesso.

O protocolo Spray-and-Wait é um protocolo baseado em inundação tal como o protocolo Epidémico. No entanto, este protocolo tenta obter melhores resultados através da inundação limitando a quantidade de cópias disseminadas na rede. Este protocolo é mais eficiente que o protocolo Epidémico porque utiliza o número limitado de cópias evitando a sobrecarga na rede.



## Capítulo 3

# Redes de Dados Nomeados

As Redes de Dados Nomeados (NDNs<sup>5</sup>) (assim como, DONA[22], NetInf[23], PURSUIT[24], entre outras [25]) é uma nova arquitetura concebida para responder às necessidades atuais e futuras da Internet. Com o aumento das capacidades de processamento, armazenamento e os meios de comunicação dos dispositivos, a produção e o consumo da informação aumentou. Este aumento conduz a problemas com o aumento de tráfego como por exemplo a disponibilidade dos dados, a segurança e a dependência da localização.

Em [26] os autores defendem que estes problemas podem ser mitigados substituindo a abordagem que faz referência a um local específico onde se encontram os dados pela abordagem que faz referência direta apenas aos dados.

Com este paradigma surge uma panóplia de novas oportunidades como, o armazenamento em cache, que visa diminuir o congestionamento e aumentar a velocidade de distribuição de conteúdo, configurações mais simples de dispositivos (aplicações baseadas no modelo centrado na informação podem ser implementadas diretamente, removendo toda a camada *middleware*) e a segurança fim-a-fim que não advém nas redes IP.

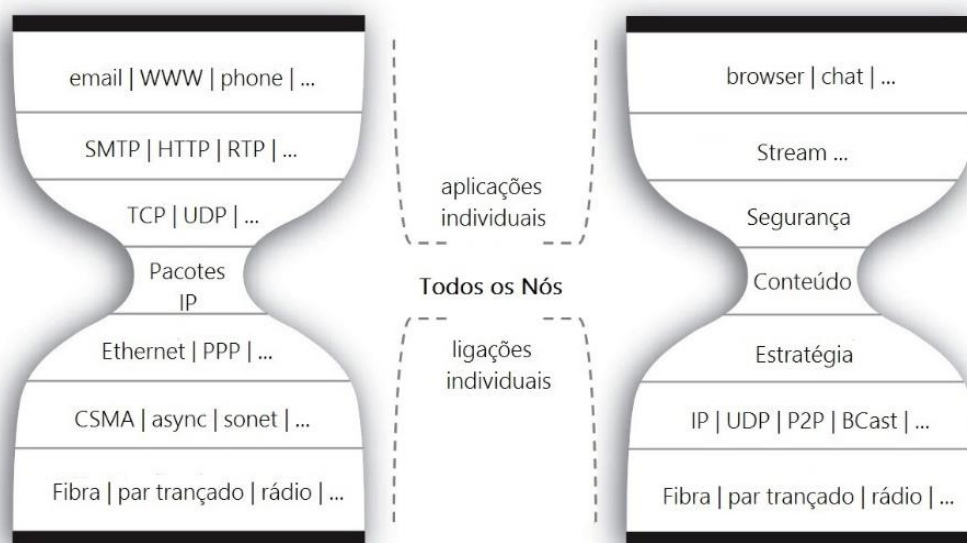
A arquitetura NDN é vista como uma alternativa para as redes baseadas em IP, baseada na subscrição e publicação de conteúdos. Os utilizadores apenas focam-se no conteúdo pelo qual têm interesse, não sendo necessário conhecer a localização dos conteúdos. O consumidor propaga o pacote interesse através da rede para obter conteúdos. Quando este atinge um nó produtor ou um nó intermediário que possua o conteúdo em cache, os dados são devolvidos pelo caminho inverso.

---

<sup>5</sup> Named Data Networks

### 3.1 Arquitetura das Redes de Dados Nomeados

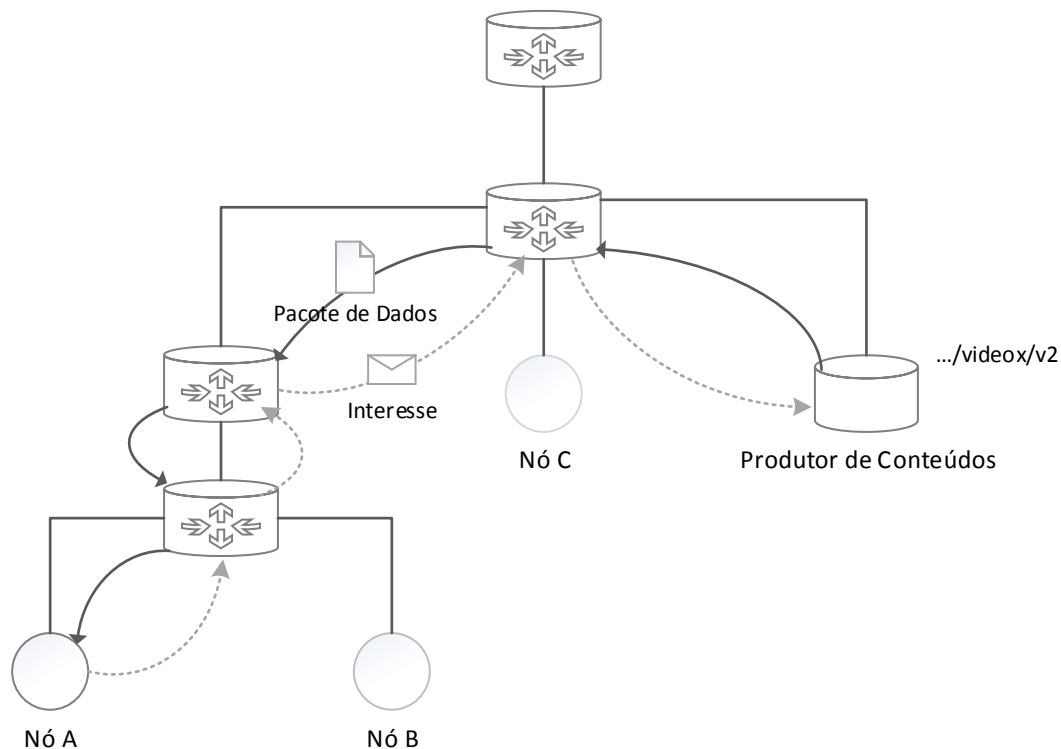
A arquitetura NDN preserva os princípios fim-a-fim e da ampulheta existente na Internet, que a torna simples, robusta e escalável. Como é possível visualizar na Figura 5, este modelo adapta o componente universal da pilha protocolar Internet (camada 3 - IP) para “pacotes de conteúdos”. Em outras palavras, o nome do conteúdo ou dos dados irá substituir o IP e tornar-se o componente universal da pilha protocolar da NDN.



**Figura 5 - Diferença entre a pilha protocolar atual e a nova abordagem NDN (adaptado de [26])**

Pode ser executado sobre qualquer protocolo de comunicação (incluindo o IP), convenções de nomeação, políticas de *routing*, entre outras. Sobre uma rede NDN pode ser executada qualquer aplicação diretamente sem necessidade de um mediador.

O modelo de comunicação utilizado por estas redes é um modelo de comunicação baseado em *Pull*, isto é, quando um nó pretende comunicar (ver Figura 6) envia um pacote chamado de “Interesse” que é encaminhado através da rede para possíveis produtores da informação desejada. Um produtor cuja informação produzida/armazenada possa interessar aos outros nós, anuncia os nomes dos conteúdos ou os prefixos dos nomes através de um protocolo de encaminhamento baseado em nomes, povoando a FIB. Este permite ao *router* determinar qual é o próximo nó para atingir os potenciais produtores.



**Figura 6 - Encaminhamento nas redes NDN**

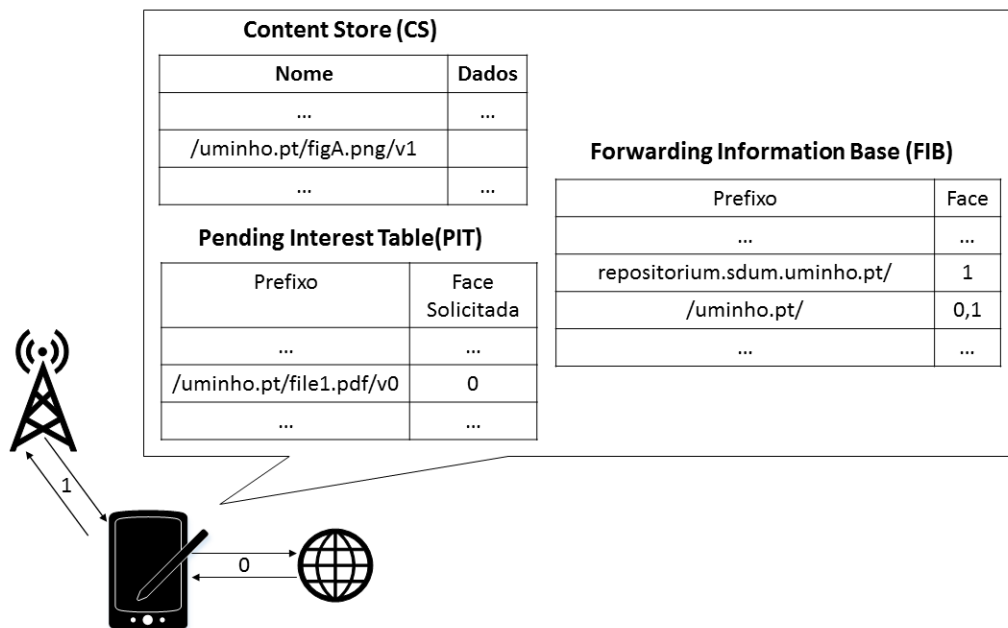
Em resposta a um interesse, o produtor envia um pacote de dados que é encaminhado pelo caminho inverso até ao consumidor.

Os *routers* que encaminham o interesse, mantêm-no na PIT com o nome do conteúdo e interface de chegada. Quando o pacote de dados chega é feita na PIT a correspondência entre o nome do pacote de dados e o nome correspondente ao “Interesse” determinando qual a interface de saída. A entrada na PIT correspondente ao “Interesse” é descartada e o pacote de dados é armazenado na “Content Store” podendo vir a ser útil para futuros interesses.

### 3.2 Modelo de um nó NDN

A arquitetura NDN define dois tipos de pacotes, pacotes de “Interesses” e de “Dados” [27]. Um pacote de “Interesse” é enviado quando um consumidor pretende obter um objeto de dados. Quando um nó possui dados que satisfaçam o Interesse este responde com um pacote de Dados. Um pacote de interesse, tal como no pacote de dados, contém um campo que identifica

o nome do conteúdo. Desta forma, um pacote de dados satisfaz o interesse se o nome que está no pacote de dados corresponder ao nome do pacote de Interesse.



**Figura 7 - Arquitetura de um nó NDN**

Um nó NDN possui estruturas próprias para lidar com os pacotes de dados nomeados. Entre estas estruturas as principais são: a Forwarding Information Base (FIB), a Content Store (CS) e a Pending Interest Table (PIT) como ilustrado na Figura 7. A FIB que possui entradas com os nomes dos conteúdos associadas às interfaces é utilizada para encaminhar os pacotes de Interesses para potenciais produtores de dados. A CS é um *buffer* temporário que permite aos *routers* armazenar os pacotes de dados recebidos. Tem como finalidade maximizar a probabilidade de partilha e reduzir a latência.

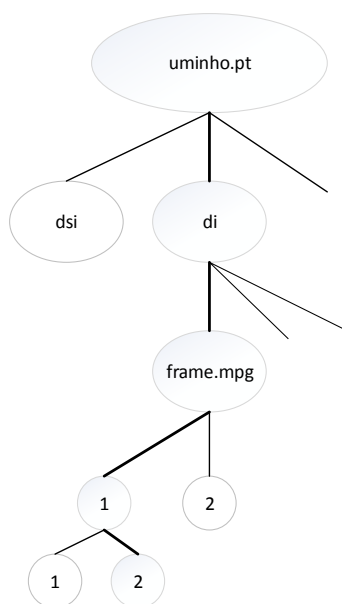
O terceiro componente, a PIT permite manter um rasto, fazendo correspondência ao Interesse e à interface por onde este chegou. Assim, quando um pacote de dados chegar é retornado pelo caminho pelo qual o Interesse chegou. A entrada correspondente à interface por onde chegou o Interesse é mantida por um período de tempo (tempo de ida e de volta). Porém, existe um risco do interesse ser eliminado antes do pacote de dados correspondente chegar.

### 3.3 Nomes

O nome é a parte mais importante da conceção das NDNs [26]. Os pacotes de dados e interesses utilizam nomes para identificar os dados que estão a ser transmitidos na rede.

Apresentam-se de forma hierárquica como é possível visualizar na Figura 8, compostos por um determinado número de componentes relacionados entre si. Um nome não é o nome do ficheiro ou parte do nome do ficheiro. A sua notação pode ser vista como URLs, onde o separador “/” delimita os componentes.

No exemplo “uminho.pt/di/videos/frame.mpg/1/2”, “uminho.pt/di/videos/frame.mpg” corresponde ao nome gerado pela aplicação e “1/2” corresponde ao segmento 2 da versão 1 do ficheiro.



**Figura 8 - Estrutura em árvore do nome (adaptado de [27])**

A estrutura hierárquica facilita a agregação, como por exemplo “uminho.pt” que corresponde a um sistema autónomo a partir do qual o vídeo foi originado. Ainda facilita a gestão e a descoberta dos nomes.

As suas convenções de nomenclatura e os mecanismos de descoberta dos nomes são geridos pelas próprias aplicações. São utilizados algoritmos determinísticos para chegar ao nome com base em informações que estão disponíveis tanto para produtores como para consumidores.

Existe um segundo mecanismo que permite a recuperação baseada em nomes parciais.



## 3.4 Segurança

Ao contrário da arquitetura atual da Internet a segurança é provida na arquitetura NDN nativamente. A autenticidade e a integridade dos objetos de dados são garantidas através de assinaturas digitais e de criptografia. Cada objeto é cifrado e assinado pelo produtor na sua criação permitindo ao consumidor determinar se os dados são ou não fidedignos. Pode-se utilizar algoritmos eficientes de verificação de assinaturas como RSA para diminuir a sobrecarga de verificação. A segurança é baseada em chave pública e a distribuição desta pode ser efetuada como se fosse um pacote NDN cifrado.

Ataques como por exemplo DDoS são irrelevantes devido às características da arquitetura NDN, em particular: o armazenamento em cache, a limitação dos interesses para os mesmos dados e o número de interesses pendentes chegados de uma determinada interface que permitem que os ataques possam ser facilmente mitigados [26].

No entanto, esta arquitetura ainda traz vários problemas associados à segurança que necessitam de ser resolvidos [28]. O armazenamento em cache permite a redução do tempo de recuperação de dados aumentando a performance mas também mantém vestígios de comunicação que podem ser explorados por ataques.

## 3.5 Armazenamento

O armazenamento é um dos mecanismos mais atrativos que foi proposto pelas NDN. Atualmente é um alvo de grande investigação, pois este significa a redução da latência e a redução do custo da largura de banda.

Os *routers* NDN depois de encaminhar os pacotes de dados armazenam os mesmos para responder a futuros interesses. Quando um interesse chega, o *router* verifica se possui dados que satisfaçam o interesse e em caso afirmativo responde com um pacote de dados correspondente. Nos *routers* atuais da Internet isto não acontece pois estes não estão habilitados para a reutilização dos dados. Esta característica traz benefícios essencialmente para as redes sem fios.

O armazenamento levanta alguns problemas de privacidade que as NDNs pretendem mitigar com remoção das informações relativas a quem solicitou os dados, mantendo apenas o

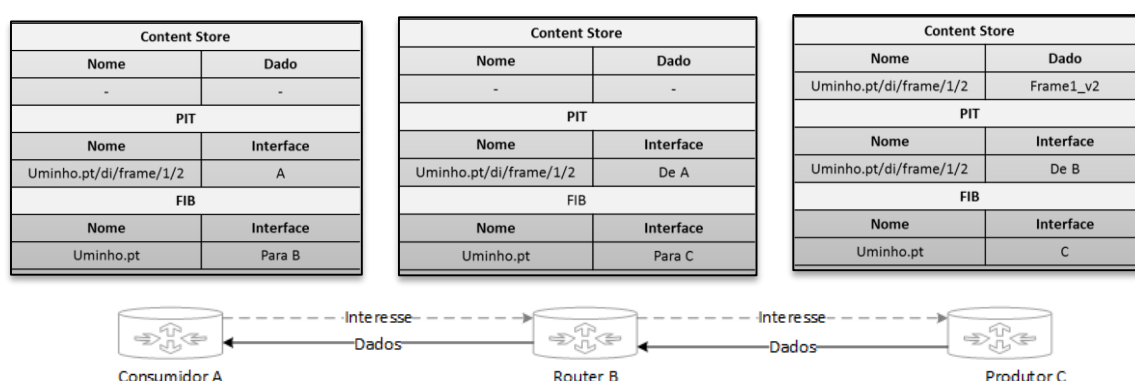
nó diretamente conectado por uma ligação ponto-a-ponto. Desta forma não é possível determinar quem solicitou os dados mantendo a privacidade [26].

Os *routers* NDN não necessitam de muito espaço de armazenamento uma vez que a CS fornece cache oportunista. No entanto, com o aumento da cache a largura de banda utilizada pode ser reduzida e possivelmente a redução do tempo de recuperação de dados.

Em adição à CS os *routers* possuem um volume de armazenamento mais persistente e maior em armazenamento em rede, chamado de repositório<sup>6</sup>. Este tipo de armazenamento pode suportar serviços como as Content Delivery Networks (CDNs) sem existir camadas adicionais para realizar este trabalho.

### 3.6 Mecanismo de expedição de Dados (Forwarding)

A comunicação nas redes de dados nomeados é impulsionada pelo consumidor que envia pacotes de “Interesses” como é ilustrado na Figura 9.



**Figura 9 – Processo de encaminhamento NDN**

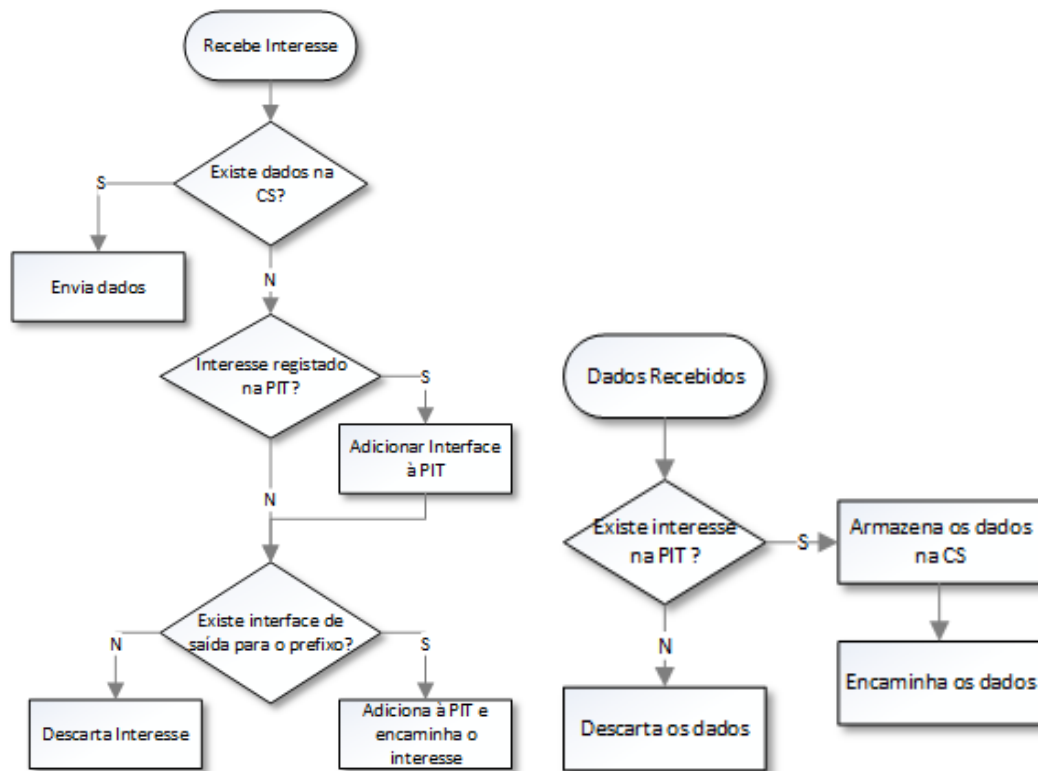
Na chegada de um pacote de interesse, o router verifica na sua CS se existem dados para satisfazer o interesse. Em caso afirmativo, os dados são enviados de volta para o consumidor através da interface que recebeu o interesse. Em caso negativo, é adicionada uma entrada na PIT (isto se na FIB constar que são conhecidos os potenciais produtores para o pedido) com o nome, interface de entrada e interface que foi encaminhado o pacote “Interesse”. Se a entrada PIT já existir, é apenas adicionada a interface pelo qual o interesse foi recebido.

<sup>6</sup> Repository (ou Repo o nome reduzido em Inglês)

Através da FIB, que possui as interfaces de saída para os potenciais produtores de informação, os pacotes de “interesses” são encaminhados pela rede.

A FIB mantém todos os prefixos dos nomes que foram anunciados pelos protocolos de encaminhamento. Estes prefixos fazem correspondência entre um nome e uma interface de saída. Caso a FIB não possua um prefixo para um determinado interesse, este é descartado. Os pacotes de interesses podem ser encaminhados através de múltiplos caminhos.

Quando um pacote de dados é recebido, através do nome que acompanha o pacote, é verificada na PIT se existe alguma entrada correspondente. Se a entrada na PIT corresponder ao nome, o *router* envia o pacote para as interfaces que receberam o Interesse. Este pacote de dados é armazenado em cache para futuros pedidos e a entrada na PIT é eliminada.



**Figura 10 - Processo de encaminhamento num nó NDN** (adaptado de [47])

Na Figura 10 é possível compreender o processo de encaminhamento que ocorre num nó NDN.

O encaminhamento é um dos principais componentes deste novo paradigma. As estratégias de encaminhamento foram trabalhadas para lidar com todos os problemas da arquitetura IP. Ao contrário da Internet que mantém apenas a melhor rota para atingir o destino,

a arquitetura NDN contém uma lista ordenada de múltiplas interfaces de saída para encaminhar os pacotes de interesses. Mantém um encaminhamento com caminhos múltiplos.

As NDNs permitem o encaminhamento adaptativo. Os *routers* podem avaliar o desempenho da rede na entrega de pacotes e detetar as perdas dos mesmos. Com isto, o encaminhamento é adaptado de maneira a tirar proveito da melhor conexão, a utilizar múltiplos caminhos para o envio de pacotes e a prevenção de ciclos.

Em [29] os autores apresentam uma estratégia de encaminhamento adaptativo de forma a complementar a estratégia inicial. Na estratégia inicial depois de um Interesse ser encaminhado é iniciado um timer baseado num RTT estimado, permitindo assim detetar problemas na rede. O tempo de espera torna o processo de encaminhamento lento sendo uma das principais desvantagens.

Assim, os autores introduzem o “Interest NACK” que determina quando um nó não pode satisfazer o Interesse. Nesta estratégia, um “Interest NACK” é enviado quando um *router* rejeita um Interesse, que pode ser provocado pela não existirem recursos suficientes (largura de banda, armazenamento insuficiente), pela não existência de dados ou pela deteção de um pedido duplicado, entre outros. Estas informações são utilizadas pelos *routers* para encaminhar futuros interesses, adaptando assim o encaminhamento. Na receção de um NACK, o *router* pode retransmitir o interesse.

Os *routers* também podem aprender e armazenar opções de encaminhamento baseadas nos dados que recebem.

### **3.7 Encaminhamento em Redes de Dados Nomeados**

Atualmente são vários os algoritmos propostos para o encaminhamento baseado em conteúdos, uma função essencial para as redes de dados nomeados. Numa rede de dados nomeados, o encaminhamento equivale a avaliar os interesses armazenados na PIT, a fim de decidir quais os *routers* vizinhos para os quais se deve enviar a mensagem de interesse. Estes algoritmos tentam lidar com o grande volume de mensagens de forma a tomar decisões o mais rápido possível. Tentam também lidar com questões de segurança. De seguida são apresentados alguns algoritmos de encaminhamento para as NDN encontrados na literatura.

### 3.7.1 Named-data Link State Routing Protocol (NLSR)

O NLSR [30] é um protocolo do tipo Link-State que opera no topo da pilha NDN. Tal como no OSPF (Open Shortest Path First), o NLSR dissemina LSAs (Link State Advertisements) para toda a rede e cada router calcula as rotas passando a conhecer toda a topologia. São utilizados pacotes de interesses e de dados para a distribuição da informação de encaminhamento.

Os LSAs transmitidos, além de permitirem a construção da topologia, possibilitam a distribuição de prefixos dos nomes. A disseminação dos LSAs é realizada à custa de dois protocolos já existentes, o CCNs Sync [31] e o Repo [31] .

Este protocolo mantém os LSAs mais recentes armazenados numa base de dados de estado de ligação (LSDB – Link-State Data Base). Sempre que é detetada uma falha ou a recuperação de qualquer uma das suas ligações, é calculado um LSA e disseminado pela rede. Ao ser detetado um LSA atualizado, o antigo é removido e é adicionado à LSDB o LSA atualizado.

O NLSR emprega dois tipos de LSAs: o LSA adjacente e o prefixo LSA. O LSA adjacente permite informar as ligações ativas que *router* está conectado e o prefixo LSA disponibiliza os prefixos dos nomes registados num router.

Este algoritmo suporta múltiplos caminhos, isto é, utiliza um algoritmo baseado no Dijkstra's para produzir uma lista de *rankings* de próximos nós para cada *router*. O cálculo dos próximos nós é baseado na distância de cada nó vizinho através do conjunto de coordenadas

O protocolo define um esquema de nomeação para os *routers* e para as mensagens de encaminhamento. Um *router* é nomeado de acordo com a rede em que está alojado e o seu nome. A estrutura do nome é a seguinte:

**`/<network>/<site>/<router>`**

Onde, um *router* utilizando o protocolo NLSR seria definido como:

**`/<network>/NLSR/LSA/<site>/<router>`**

O NLSR utiliza um modelo de confiança para a gestão das chaves públicas onde utiliza um conjunto de âncoras de gestão hierarquicamente estruturadas. Este modelo permite a autenticidade e integridade das mensagens no encaminhamento intra-domínio, isto é, cada pacote

de dados NLSR é assinado pelo *router* de origem com uma chave pública válida e na sua recepção a assinatura é verificada.

### **3.7.2 COBRA: Lean Intra-domain Routing in NDN**

O COBRA (Content-driven Bloom filter based Routing Algorithm) [32] é um protocolo de encaminhamento interdomínio para as redes NDN, voltado para conteúdos e baseado na estrutura de dados probabilística Bloom Filter[33]. Este algoritmo cria rotas com base em caminhos utilizados anteriormente para a recuperação de conteúdo, e mantém as informações de encaminhamento atualizadas sem a necessidade de sinalização entre os nós.

O protocolo instala em todas as interfaces dos nós NDN o Bloom filter para efetuar o processamento dos pacotes, contruindo também a FIB baseada no Bloom filter. Para cada prefixo de nome que chega é calculado um *hash* que relaciona este com a interface de entrada, deixando desta forma um rasto até ao produtor do conteúdo. Futuramente se chegar um interesse que partilha parte do prefixo este é redirecionado pelo mesmo caminho. Este algoritmo também constrói um ranking de interfaces baseado na correspondência ao prefixo mais longo, de maneira a obter uma rota e evitar inundações na rede.

Este protocolo também permite a manipulação das retransmissões de forma a descobrir novas rotas. Os resultados demonstram que o COBRA pode reduzir significativamente a sobrecarga em relação ao encaminhamento baseado em inundação e obter distancias equivalentes às obtidas pelo algoritmo Djisktra.



# Capítulo 4

## Trabalho Relacionado

Até agora foram descritas duas arquiteturas, DTN e NDN, com propósitos diferentes. Fundamentalmente, a arquitetura NDN foi concebida para a entrega de conteúdos, utilizando o paradigma publicação/subscrição, que se abstrai do conceito fim-a-fim.

Já a arquitetura DTN foi desenhada para lidar com redes onde os altos padrões de mobilidade causam atrasos variáveis, conexões intermitentes, altas taxas de erros e taxas de transferência assimétricas. Utiliza o paradigma “Armazenamento, Transporte e Envio de Agregados”, que eventualmente pode armazenar unidades de dados até que estes possam ser transmitidos para outro nó. Para ajudar a mitigar os efeitos de mobilidade foram desenvolvidos protocolos de encaminhamento, como por exemplo o encaminhamento epidêmico, PROPHET e Spray-and-Wait. No entanto, a arquitetura DTN é dependente do modelo centrado em entidades, sendo necessário atribuir um identificador a cada.

Por outro lado, a arquitetura NDN foca-se nos dados e não nos nós, facilitando a compreensão destes permitindo o seu armazenamento e a reutilização. Isto pode trazer vantagens para a mobilidade, já que não é necessário adquirir nenhum indentificador.

Apesar dos seus propósitos estas arquiteturas apresentam algumas similaridades, entre elas o encaminhamento flexível, os dados podem ser encaminhados através de múltiplos caminhos. Ambas as abordagens, permitem o transporte de dados impedindo os ciclos<sup>7</sup>, o armazenamento dos dados na rede, embora as DTNs utilizem este para a persistência e a tolerância às interrupções e as NDNs utilizem para a redução da latência e também para a tolerância às interrupções. As arquiteturas possibilitam a ligação tardia, isto é, a ligação dos nomes para localizações e a persistência dos dados.

Através da comparação efetuada na Tabela 2 é possível notar que ambas as arquiteturas possuem algumas similaridades. A integração destas poderá trazer várias oportunidades,

---

<sup>7</sup> Loops na língua Inglesa



aproveitando o potencial de ambas. As redes de dados nomeadas podem aumentar a confiabilidade e obter uma maior performance através do armazenamento persistente na rede.

**Tabela 2 - Comparação entre DTNs e NDNs [34]**

<b>Caraterística</b>	<b>Arquitetura DTN</b>	<b>Arquitetura NDN</b>
<b>Tolerância a interrupções</b>	Durante as interrupções, os agregados são armazenados e posteriormente encaminhados	Os pacotes de dados sem entradas na PIT são descartados. Os pacotes de interesses sem entradas na FIB são descartados
<b>Endereçamento e nomeação</b>	Endereços URI para os nós	Nomes hierárquicos para o conteúdo como os endereços URI
<b>Armazenamento</b>	Armazenamento temporário (armazenamento, transporte e envio de agregados)	Armazenamento temporário na cache e mais persistente no Repositório.
<b>Fragmentação</b>	Possível em todos os nós	Apenas na origem
<b>Modelo</b>	Push	Pull
<b>Segurança</b>	É opcional, tanto a nível do canal como do conteúdo	A nível de canal não existe. Ao nível do conteúdo é obrigatório. O conteúdo é assinado pelos produtores
<b><i>Multicast</i></b>	Suporta o <i>multicast</i> (o EID pode referir-se a um nó ou a um grupo de nós)	Possibilita o <i>multicast</i> na entrega de dados
<b>Confiabilidade e Controlo de fluxo</b>	Utilização de custódia	O consumidor final é responsável por expressar que um interesse não foi satisfeito
<b>Desempenho de entrega</b>	Não está focada nisto	Tem como função a rápida entrega de conteúdo reduzindo a latência

## 4.1 Trabalhos para aproximar as duas arquiteturas

Com o surgimento das redes de dados nomeados os investigadores acreditam que a integração das duas arquiteturas pode trazer várias oportunidades. Recentemente, na literatura tem surgido alguns estudos que de forma a integrar as duas arquiteturas, nomeadamente protocolos onde a comunicação é feita com dados nomeados. Alguns destes protocolos exploram as capacidades da rede, outros exploram os encontros históricos e outros a popularidade dos conteúdos. Nesta secção são apresentados alguns trabalhos encontrados na literatura, descrevendo o funcionamento geral de cada um.

### 4.1.1 Listen First Broadcast Later (LFBL)

Em [35] os autores propõe o protocolo de encaminhamento LFBL(Listen First Broadcast Later), capaz de lidar com a alta dinâmica das redes ad-hoc. Esta proposta utiliza uma variação do NDN (o anúncio de prefixos do nome, o encaminhamento de interesses e o retorno dos dados) adaptada às MANETs. A arquitetura NDN efetua a propagação de nomes, enquanto o LFBL efetua a inundação de pedidos de dados pela rede. Cada nó mantém as distâncias para os nomes de objetos conhecidos, no entanto se o potencial produtor de dados não for conhecido, o interesse é enviado em *broadcast* pela rede.

O seu funcionamento é simples, primeiro os nós estão à escuta de pedidos, em seguida, após a receção de pedidos estes procuram um nó ótimo para encaminhar o pedido. A escolha do caminho é efetuada implicitamente, utilizando o encaminhamento oportunista baseado nas distâncias de destino estimadas a partir de retransmissões. Em resposta, são enviados pacotes de dados correspondentes aos interesses através do caminho mais curto até ao consumidor.

O LFBL consegue obter menores custos de largura de banda devido ao mecanismo de escolha de caminho, e também oferece maior mobilidade eliminando a necessidade de as aplicações escolherem um endereço IP de destino específico antes poderem escolher conteúdo ou serviços.

### 4.1.2 Information Centric Delay Tolerant Network (ICDTN)

Os autores em [36] exploraram a integração das ICNs com os princípios das DTNs. Criaram um modelo de simulação para as PSNs denominado por ICDTN onde são demonstrados os benefícios da integração destas duas abordagens. São implementados dois mecanismos de encaminhamento, o encaminhamento dos pedidos e o encaminhamento dos objetos de volta. A resposta aos pedidos segue o caminho inverso, o que implica que exista um tempo de ligação suficiente para a transmissão. O ICDTN a qualquer instante constrói um grafo  $G(t)$ , que consiste num conjunto de nós com ligações. Cada nó possui um conjunto de objetos de dados e um conjunto de pedidos. Quando um nó gera um novo pedido, ele verifica as ligações ativas para enviar diretamente esse pedido. Na receção de pedidos o nó calcula a aceitabilidade verificando se existe algum objeto de dados correspondente na sua cache. A aceitabilidade é retornada para o nó que enviou o interesse. Seguindo este processo, o nó que recebeu o interesse procede á deteção de novas ligações para decidir para quem o interesse deve ser encaminhado. Sempre que um nó recebe um pedido coloca-o no seu conjunto de pedidos  $R(n_i)$  permitindo que seja armazenado e transmitido no futuro. A resposta ao interesse poderá ter de percorrer vários nós, por exemplo, se um nó  $n_2$  envia o interesse para um nó  $n_3$ , qualquer resposta para  $n_1$  deve ser retornada via  $n_2$ .

Um dos principais desafios propostos é encontrar soluções mais robustas para garantir que as respostas suportam a ligação tardia. Outros desafios propostos são a descentralização dos algoritmos, isto é, em vez da atual base de nomes da ICN implementar estratégias que permitam aos nós gerar consultas sofisticadas, aprimorar a gestão de resposta implementado algoritmos cooperativos que permitem a coordenação de réplicas e fontes.

### 4.1.3 Broadcast-Only Named Data (BOND)

O BOND (*Broadcast-Only Named Data*)[37] é um protocolo centrado na informação para redes móveis sem fios onde os nós se movem livremente e de forma imprevisível. A comunicação neste protocolo é inteiramente baseada em dados nomeados.

O BOND utiliza um esquema de nomeação para os dados e para os nós. Cada nó é identificado com um nome único na rede que é utilizado para encaminhar as mensagens. Para



No cabeçalho de cada mensagem BOND é adicionado a distância ao nó de origem. Este campo denominado por *srcDist* é atualizado a cada salto. O nó intermediário insere a sua distância para o nó de origem antes de encaminhar a mensagem. A Figura 11 ilustra o processo de determinar se um nó é elegível ou não para encaminhar a mensagem. Uma vez conhecida a distância para um nó, é possível utilizar essa informação para encaminhar futuras mensagens com esse destino. Também em cada cabeçalho BOND existe um campo denominado por *dstDist* que indica a distância até ao nó produtor/destinatário. Este campo é atualizado sempre uma mensagem é recebida e posteriormente encaminhada. Com base na métrica *dstDist* o nó determina se é um nó elegível ou não para ficar responsável pelo encaminhamento da mensagem.

#### **4.1.4 Neighborhood-aware Interest Forwarding (NAIF)**

NAIF (Neighborhood-aware Interest Forwarding)[38] é um protocolo de encaminhamento que utiliza dados nomeados adaptado às MANETS. Esta abordagem surge como uma melhoria ao protocolo original NDNF, um protocolo também com base em dados nomeados que lida com a conectividade intermitente.

Tal como no NDNF são efetuadas transmissões de interesses em *broadcast*. No entanto, em vez de selecionar um nó para encaminhar os interesses são utilizados nós cooperativos entre o consumidor e as potenciais fontes de dados. Este mecanismo possui habilidades de controlo de transmissão que com base em estatísticas podem adaptar a taxa de transmissão, precavendo o congestionamento da rede. Utiliza fundamentalmente duas métricas que são calculadas localmente que são a taxa de recuperação de dados (R) em relação ao número de interesses enviados (IntPs) e a taxa de encaminhamento. São registados também o número de pacotes de interesse perdidos, que significa o número de pacotes de interesse não satisfeitos. Através destas métricas a taxa de expedição é ajustada, reduzindo ou aumentando. Através da cooperação entre os nós, a carga de trabalho poderá ser reduzida bem como o número de pacotes na rede.

#### **4.1.5 Content-Centric Dissemination Algorithm (CEDO)**

CEDO é um algoritmo para disseminação de conteúdo em redes tolerantes a atrasos. Este algoritmo tem o objetivo de maximizar a taxa de entrega de conteúdo distribuído num ambiente onde podem ser pedidos e armazenados conteúdos com diferentes popularidades. Através do

utilitário da taxa de entrega por conteúdo que é proporcional à taxa de conteúdo em falta toma decisões de descarte e de programação apropriadas. Cada nó pode estimar a utilidade do conteúdo localmente utilizando estimadores imparciais.

É assumido que cada nó gera em momentos aleatórios pedidos de conteúdos aleatórios e cada um desses pedidos possui um tempo de vida. Os diferentes conteúdos podem ter taxas de solicitação diferentes assim como alguns podem ser mais populares que outros.

Para calcular a utilidade de cada conteúdo cada nó necessita de calcular a perda de pacotes, a taxa de pedidos e a taxa de entrega. A taxa de entrega mede o número de conteúdos entregues com sucesso. Esta taxa é obtida através da Equação 4:

$$DR = q^{(i)} \times p^{(i)} \quad (4)$$

Na Equação 4,  $q^{(i)}$  é a taxa de pedidos para o conteúdo  $i$  e  $p^{(i)}$  é a probabilidade que o nó tem de obter o conteúdo.

Para obter a taxa de pacotes perdidos, o nó necessita de estimar a taxa de pedidos e a taxa de entrega. Localmente é armazenado o instante de tempo em que o nó gerou o pedido e quando este foi satisfeito. Através da Equação 5 é possível obter a taxa de pacotes perdidos.

$$MR^{(i)} = q^{(i)} - DR^{(i)} \quad (5)$$

O CEDO tenta calcular a utilidade dos conteúdos de forma distribuída. Para isso ele envia estas taxas para os outros nós de forma a refinar as estimativas locais. Utilizando estes valores aproximados o nó calcula a utilidade do conteúdo aplicando a Equação 6.

$$U^{(i)} = \lambda TTL(q^{(i)} - DR^{(i)}) \quad (6)$$

Nesta equação,  $\lambda$  é a taxa de encontro do nó e TTL é o tempo médio de vida de um pacote de dados.

A utilidade reflete a popularidade dos conteúdos e é utilizada para descartar ou encaminhar os dados. O CEDO dá prioridade de envio aos dados com maior utilidade.

#### 4.1.6 Social-Tie based Content Retrieval (STCR)

O STCR [39] é um algoritmo que permite a recuperação de dados em redes tolerantes a atrasos (MANET). Ao longo do tempo, os nós armazenam informação que será utilizada posteriormente para tomar decisões de encaminhamento. Baseado no algoritmo K-means [40], cada nó calcula um grafo hierárquico baseado nas relações sociais entre nós.

Os eventuais encontros com outros nós são registados, armazenando o instante de tempo em que ocorreu o encontro. Com o registo de encontros, os nós calculam localmente a relação social que utiliza a frequência e o recente contacto com que um par de nós se encontra. A Equação 7 permite o cálculo da relação social de um nó  $i$  com um nó  $j$ .

$$R_i(j) = \sum_{K=1}^n F(t_{base} - t_{jk}) \quad (7)$$

Nesta equação  $F(x) = \frac{1^{\lambda x}}{2}$  é a função de pesagem entre a frequência e a frescura. O valor  $t_{jk}$  é o tempo de contacto do nó  $i$  com o nó  $j$  e o valor  $t_{base}$  corresponde ao instante atual.

Cada nó mantém na sua tabela de encaminhamento as distâncias sociais que possui para cada nó presente na rede. Durante o período de contacto as tabelas de encaminhamento são trocadas e fundidas permitindo ao nó deter um conhecimento alargado da rede.

Posteriormente e com base na tabela de encaminhamento, o nó calcula localmente a centralidade para cada nó presente na tabela. A centralidade corresponde à distância social média a partir de um dado nó para todos os outros nós encontrados. Pode ser vista como uma medida de tempo que nos diz o tempo que demora a transmitir determinados pacotes de um nó para outros presentes na rede.

A centralidade é obtida através da seguinte equação:

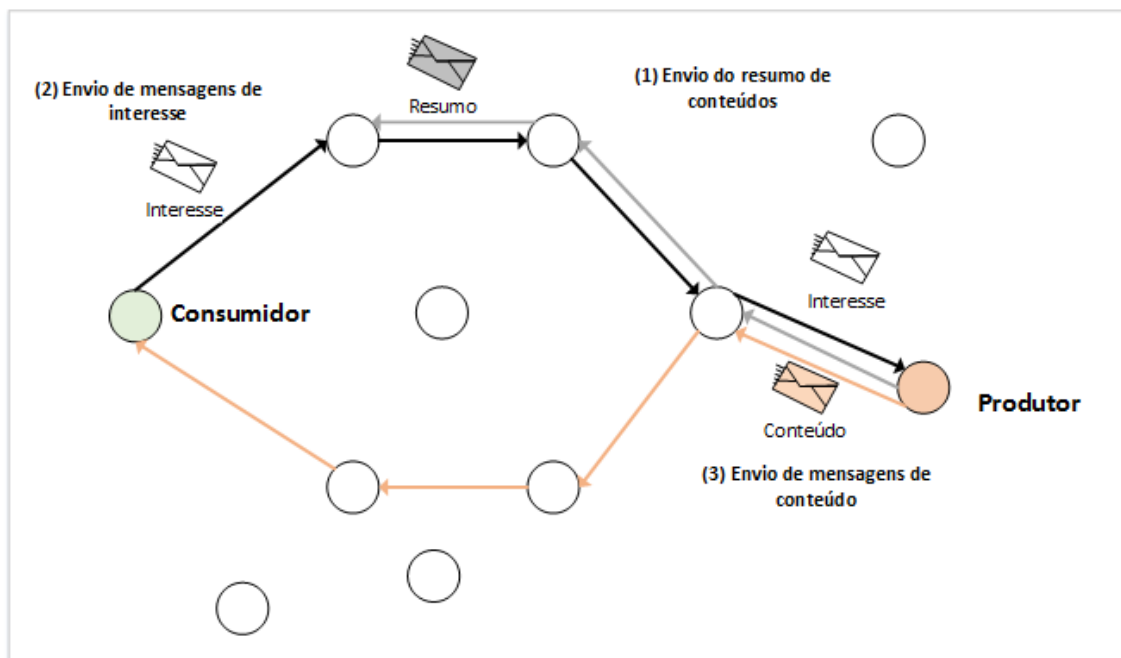
$$C_i = \alpha \frac{\sum_{K=1}^N R_i(K)}{N} + (1 - \alpha) \frac{(\sum_{K=1}^N R_i(K))^2}{N \times (\sum_{K=1}^N (R_i(K))^2)} \quad (8)$$

Nesta equação a variável  $N$  representa o número de nós observados a partir da tabela de relações sociais, isto é, o número de relações sociais a partir de um nó  $i$ . O parâmetro  $\alpha$  permite equilibrar a importância da relação entre os nós com as características da rede. Numa rede menos densa é preferível um  $\alpha$  menor dando mais importância à ligação dos nós.

A alta centralidade indica que o nó está constantemente em contacto com outros nós e recentemente. A Figura 12 ilustra o funcionamento do encaminhamento no STCR.

No STCR cada nó mantém uma estrutura local que armazena um resumo dos conteúdos existentes na rede. Associado a cada conteúdo existe um identificador do produtor. Esta estrutura é construída através da receção de resumos provenientes de outros nós com centralidade inferior. A troca de resumos dos conteúdos é progressivamente encaminhada para os nós com maior centralidade.

Os autores do STCR defendem que os nós com centralidade idêntica possuem o mesmo conhecimento da rede e agrupam estes nós no mesmo aglomerado.



**Figura 12 - Funcionamento do encaminhamento no STCR**

Após concluída a troca do resumo de conteúdos, é efetuado o encaminhamento dos pacotes de interesse. Numa fase inicial, quando não existe conhecimento sobre o produtor do conteúdo, as mensagens de interesse são propagadas da mesma forma que o encaminhamento do resumo de conteúdos. Eventualmente o interesse vai chegar a um nó que conhece o produtor do conteúdo solicitado. Assim que for conhecido o produtor do conteúdo, o identificador do nó produtor é inserido na mensagem de interesse e esta passa a ser dirigida ao produtor do conteúdo. O encaminhamento de interesses nesta fase utiliza as relações sociais para atingir o produtor. Os nós retransmitem os interesses para um nó com uma relação social com o produtor do conteúdo



superior à sua até atingir o produtor. Quanto o TTL de uma mensagem de interesse expira e acionada a estratégia de “fallback” que envia epidemicamente os interesses para que eles cheguem o mais rápido possível ao produtor.

Na chegada do interesse ao produtor de conteúdos ou a um nó intermediário que possua dados, é retornada uma mensagem de dados direcionada ao consumidor. O encaminhamento das mensagens de dados utiliza as relações sociais para chegar ao consumidor.

## 4.2 Comparação entre as diferentes estratégias

Nesta secção são comparados os protocolos encontrados na literatura que pretendem aproximar as duas arquiteturas (NDN e DTN). A Tabela 3 apresenta uma comparação dos protocolos.

	<b>Flexibilidade</b>	<b>Consumo de Recursos</b>	<b>Utilização de Informação</b>	<b>Escalabilidade</b>
<b>LFBL</b>	Não foi possível apurar	Canal de comunicação (todas as transmissões são <i>broadcast</i> )	Utiliza as retransmissões dos pedidos para calcular as distâncias para o destino.	A inundação de pedidos poderá congestionar a rede.
<b>NAIF</b>	Média	Canal de comunicação (todas as transmissões são <i>broadcast</i> )	Utiliza informação sobre o estado da rede. Com as estatísticas obtidas decide se deve diminuir ou aumentar a taxa de transmissão.	Pouco escalável. Com o aumento do número de nós na rede o canal de comunicação poderá ficar congestionado.
<b>BOND</b>	Média	Canal de comunicação (todas as comunicações são <i>broadcast</i> )	Utiliza o número de saltos para obter a distância. Com a distância o nó determina se é elegível para encaminhar.	Tal como o NAIF este protocolo poderá provocar o congestionamento.

	<b>Flexibilidade</b>	<b>Consumo de Recursos</b>	<b>Utilização de Informação</b>	<b>Escalabilidade</b>
<b>CEDO</b>	Média	Não foi possível apurar	Informação histórica dos conteúdos (utilidade dos conteúdos).	Este protocolo permite a gestão eficiente do armazenamento. Necessita de ser estudado em redes mais densas.
<b>STCR</b>	Média	Nível de processamento elevado com o aumento da densidade da rede.	Informação histórica de contactos com nós.	Permite o controlo do tráfego de forma eficiente.

**Tabela 3 - Comparação dos trabalhos relacionados**

Na tabela 4 os diferentes protocolos são comparados em termos de flexibilidade, consumo de recursos, utilização da informação e escalabilidade. Em termos de flexibilidade, todos os protocolos estudados são flexíveis dado que não é necessário adquirir endereços e a compreensão dos pacotes é facilitada com o paradigma de dados nomeados. Três dos protocolos apresentados, são baseados na inundação da rede. Estes podem trazer problemas de escalabilidade e consumo de recursos. Os protocolos STCR e CEDO utilizam a informação histórica para criar métricas de decisão para o encaminhamento dos pacotes. Estes dois últimos garantem escalabilidade, pois permitem o controlo de tráfego. No caso do STCR, devido à utilização de algoritmos de mineração de dados<sup>8</sup> (K-mean) é necessário dispor de mais poder de processamento.

---

<sup>8</sup> Em inglês os termo utilizado é data mining



# Capítulo 5

## Conceção da Plataforma e do Protocolo de Encaminhamento

Neste capítulo é introduzida uma nova plataforma, que tem por objetivo operar como ferramenta de simulação, de forma a fornecer suporte ao desenvolvimento de protocolos de encaminhamento com base em dados nomeados para redes tolerantes a atrasos. É descrita a arquitetura e a conceção da implementação da plataforma e do protocolo de dados nomeados para redes tolerantes a atrasos denominado por PIFP (Probabilistic Interest Forwarding Protocol).

### 5.1 Visão geral

O protocolo proposto é focado na informação que por sua vez proporciona a propagação de interesses com base no nome dos conteúdos. As mensagens são enviadas de nó para nó até chegar a um nó que possua dados para satisfazer o interesse. Este protocolo permite a difusão de conteúdos para cenários com alta mobilidade, interrupções, densidade variável e altos atrasos.

O PIFP combina as características das NDNs e das DTNs, onde as mensagens são interesses e conteúdos. Para avaliar este protocolo foi necessária a criação de uma plataforma que fornece suporte ao desenvolvimento de protocolos com base em dados nomeados para redes tolerantes a atrasos. A plataforma integra as NDNs no simulador DTN Opportunistic Network Environment (ONE). Este simulador foi desenhado para avaliar protocolos de encaminhamento DTN e aplicações.

## 5.2 Arquitetura

O PIFP é um protocolo que tem como objetivo disponibilizar a distribuição de conteúdos baseado nos nomes. Este protocolo herda as características das NDNs e DTN para lidar com os desafios das redes com alta mobilidade. Neste tipo de redes existem diversas interrupções durante a comunicação de dados, dado que normalmente não existe densidade suficiente para manter todos os nós conectados. O PIFP pretende tirar vantagem das redes NDN que provêm o suporte nativo à mobilidade, dado que não existe associação entre a identificação e a localização dos pacotes de interesse e de conteúdo. Permite aos consumidores e produtores trocar de posições na topologia sem alterar os seus identificadores.

A Figura 13 ilustra a arquitetura básica de um nó ND-DTN. Nesta figura são apresentados os principais componentes que compõem um nó. O modelo de referência de um nó DTN não suporta o paradigma de dados nomeados. De maneira a suportar o este paradigma, foi necessário adicionar a pilha NDN no topo da pilha DTN. Assim, foram introduzidos os componentes básicos NDN (PIT, CS e FIB) de forma a lidar com os dados nomeados.

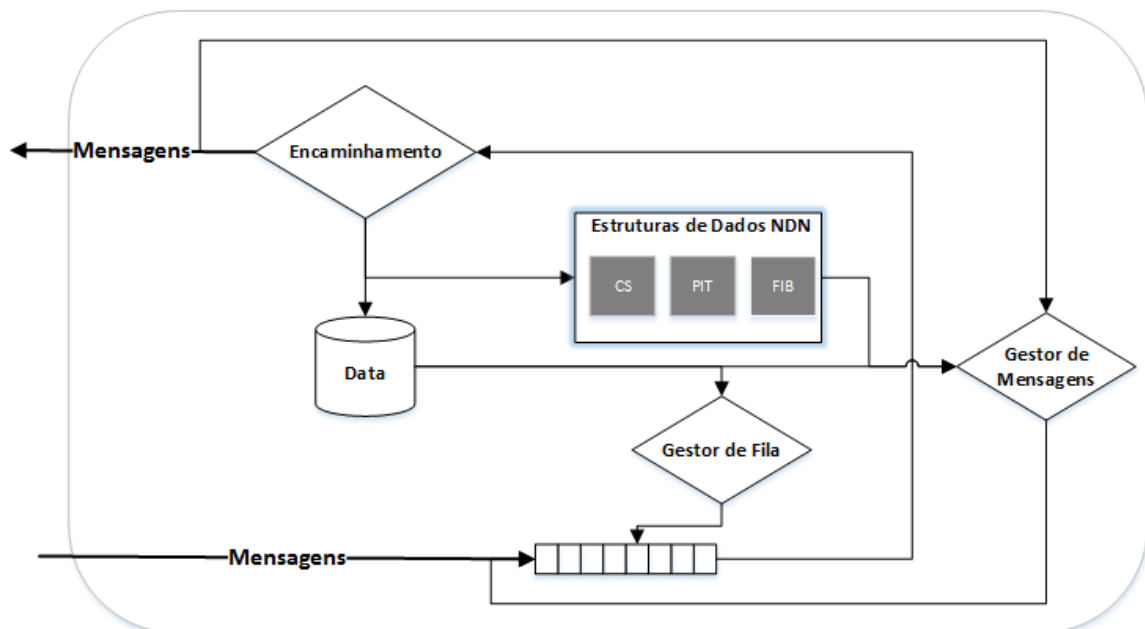


Figura 13 - Modelo de um nó ND-DTN

Quando existe contacto, o nó troca informações de encaminhamento assim como mensagens de conteúdos ou de interesse. Na receção de uma mensagem, esta é armazenada

num *buffer* à espera de ser processada. O processamento da mensagem deriva das ações do “gestor de *buffer*” que aplica políticas de armazenamento.

As decisões de encaminhamento são tomadas pelo módulo de encaminhamento que decide quais as mensagens a encaminhar. Através das informações armazenadas na PIT e na FIB o protocolo de encaminhamento toma decisões de entrega de mensagens. Durante o contacto a estratégia de encaminhamento define a prioridade das mensagens para serem enviadas para o nó atualmente conectado através das previsibilidades construídas com a frequência dos contactos com os conteúdos.

Geralmente, é criada uma réplica da mensagem que é enviada para a rede. A mensagem original permanece armazenada. O destino desta depende de outras políticas utilizadas pelo gestor de mensagens.

### **5.3 Esquema de nomeação**

A nomeação das mensagens segue o modelo de comunicação da arquitetura NDN, que é realizada apenas com nomes. Os pacotes de interesses são encaminhados com base nos nomes até aos potenciais produtores, que respondem com o pacote de dados solicitado. Ambos os pacotes de interesse e de dados são identificados pelos seus nomes, sendo únicos na rede.

Os nomes são estruturados hierarquicamente, como por exemplo,

**`uminho/cursos/engenhariacomunicacoes.html`**

Quando um nó consumidor tem interesse num conteúdo, adiciona uma nova entrada à sua PIT com o nome dos dados que pretende. Na chegada de um pacote de conteúdos, é verificado o nome do conteúdo, e se existir correspondência, o interesse é dado como satisfeito removendo a entrada na PIT.

Na CS são criadas entradas com os nomes dos objetos de dados que o nó possui armazenado.

O nome dos objetos de dados são utilizados também para construir estatísticas, e com base nelas realizar decisões de encaminhamento. Quando existe contacto entre nós, os nomes existentes na CS são divulgados e com base nesta informação são construídas estatísticas. As estatísticas construídas determinam a probabilidade de um nó chegar a um conteúdo.

Este estudo não abrange a nomeação de interesses. Os nomes são publicados em ficheiros, de modo a todos os nós terem conhecimento dos interesses existentes na rede. A nomeação é ainda um objeto de investigação, pois implica o acordo entre o publicador e o interessado.

## 5.4 Estruturas de Dados

Como já mencionado são utilizadas três estruturas de dados para apoiar na entrega de conteúdos. As estruturas são designadas de PIT que armazena os interesses pendentes, CS que armazena os objetos de dados que o nó possui armazenado e a FIB que possui informações de encaminhamento. Estas estruturas foram adaptadas a partir da arquitetura original para lidar com o cenário aqui estudado.

A PIT armazena a informação relativa aos interesses pendentes do próprio nó ou de outros nós que ficou responsável. Nesta tabela é criada apenas uma entrada para cada interesse. Cada vez que o nó recebe um pacote de interesse, adiciona o ID do nó que enviou o interesse e do nó que originou o interesse, deixando assim um rasto que irá ser utilizado para entregar os pacotes de dados ao consumidor. A arquitetura NDN é “*broadcast friendly*” mas neste ambiente esta abordagem torna-se impraticável devido às limitações dos nós.

**Tabela 4 – Exemplo da Pending Interest Table (PIT)**

<b>Id</b>	<b>Lista de nós</b>	<b>Instante de criação</b>	<b>TTL</b>	<b>Subscritor</b>	<b>Última probabilidade</b>
File1/pag2	p1, c4, t2 ..	19:24:45	300	Sim	0.3
...					
Album2/faixa	p2	12:00:38	50	Não	0.7

A Tabela 4 ilustra um exemplo de uma tabela PIT preenchida. Foi introduzido o campo TTL que define um tempo de vida para aquele interesse. Quando este expirar, a entrada é eliminada assim como todas as mensagens de interesse correspondentes à entrada. O campo subscritor indica se o nó é subscritor do interesse ou não. No momento que o utilizador gera o interesse, é verificada se existe já uma entrada para aquele nome e caso contrário, é criada uma nova entrada na tabela onde os campos são preenchidos.

O campo “última probabilidade”, como irá ser descrito posteriormente é utilizado para tomar decisões de encaminhamento.

Com o exemplo apresentado, é possível compreender a diferença entre a arquitetura centrada no utilizador e a arquitetura centrada na informação. No paradigma onde a informação é o elemento principal, a compreensão das mensagens é simples podendo absorver informação que mais tarde irá gerar oportunidades na propagação de mensagens. É possível criar um histórico com a informação adquirida, podendo determinar os conteúdos mais populares, a relação que um nó tem com determinado conteúdo, e outros registos.

Na receção de dados, é verificada na PIT se existe correspondência e em caso afirmativo, é criada uma nova entrada na CS. Este conteúdo será armazenado em disco e permanecerá temporariamente em cache, caso o nó tenha recebido mensagens de interesses para este conteúdo.

Os conteúdos para os quais o nó não é subscritor irão permanecer temporariamente em cache e serão descartados após exceder o tempo de vida definido.

**Tabela 5 – Exemplo de Conteúdo da Content Store (CS)**

<b>Id</b>	<b>TTL</b>	<b>Subscritor</b>	<b>Dados</b>
google/file1/pag5.pdf	∞	Sim	pag5.pdf
...			
uminho/di.html	500	Não	di.html

Na Tabela 5 está representada a estrutura da CS onde o nome do conteúdo é definido como o elemento chave e possui mais dois atributos que são o TTL e o campo “Subscritor”.

Na solução proposta, a FIB permite armazenar informações de encaminhamento que serão utilizadas posteriormente para tomar decisões. Na arquitetura convencional, a FIB recorda todos os nomes que foram anunciados, recordando também as interfaces que recebeu os anúncios. Esta mantém o estado de todas as interfaces construindo um ranking de interfaces para chegar a um determinado conteúdo. Neste ambiente, devido à alta dinâmica e à não existência de infraestrutura, não é possível estabelecer um caminho entre o consumidor e o conteúdo. Assim, é necessário adaptar a tabela de encaminhamento ao cenário estudado.



**Tabela 6 – Conteúdo da FIB**

<b>Id</b>	<b>P()</b>
google/file1/pag5.pdf	0.90
...	
uminho/di.html	0.87

A Tabela 6 mostra um exemplo da FIB. Esta consiste num vetor onde a chave é o nome do conteúdo e tem como atributo uma métrica denominada por previsibilidade.

Cada nó é responsável por atualizar estes parâmetros, pois agem de forma independente.

## 5.5 Mensagens

No PIFP são utilizados essencialmente dois tipos de mensagens, mensagens de interesse e mensagens de dados.

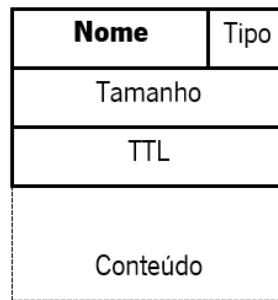
Quando um utilizador pretende obter dados, mostra o seu interesse enviando uma pacote de interesse para a rede. O nome do conteúdo solicitado é anexado à mensagem, o que permite uma melhor compreensão dos dados.

<b>Nome</b>	Tipo
Tamanho	
TTL	

**Figura 14 - Pacote de Interesse (PI)**

Na Figura 14 é apresentada a estrutura de uma mensagem de interesse. De modo a diferenciar os tipos de mensagens, foi reservado um campo para este efeito. Assim para identificar uma mensagem de interesse é inserido no campo “Tipo” o valor “pi”.

O campo TTL define o tempo de vida da mensagem em segundos. Quando este expirar a mensagem será descartada da rede.



**Figura 15 - Pacote Dados (PC)**

Em resposta a um interesse, é criada uma mensagem de dados (ver figura 15) e é enviada de volta para o nó que enviou o interesse. Tal como na mensagem de interesse é reservado um campo que corresponde ao nome do conteúdo que é armazenado na mensagem. O campo “tipo” neste caso possui o valor “pc”. Mais uma vez é utilizado o campo TTL, que é decrementado pelo simulador com o decorrer do tempo de simulação. Devido à alta dinâmica da rede que provoca frequentes desconexões, existe a possibilidade do conteúdo nunca ser entregue ao consumidor final. O tempo de vida permite eliminar as mensagens antigas dando oportunidade ao armazenamento de novas mensagens.

## 5.6 Faces

As NDNs utilizam o conceito de face para definir qual a interface de saída dos interesses, relacionando a interface com um nome de um conteúdo. Também na receção de um pacote de interesse, o nó armazena o interesse insatisfeito e guarda a face de chegada do interesse. Em cenários como as DTNs não é possível definir faces de saída devido à dinâmica da rede onde os nós mudam de localização com muita frequência. As faces na solução proposta foram substituídas por identificadores do nó. Estes identificadores são utilizados apenas para registar quais os nós que enviaram os pacotes de interesse. Associado a cada interesse insatisfeito existe uma lista de identificadores de nós. Na arquitetura convencional, as faces são utilizadas na FIB para o encaminhamento dos interesses. Num cenário DTN, não é viável definir qual o próximo nó para encaminhar uma mensagem de interesse devido à mobilidade característica destes cenários. O encaminhamento das mensagens de interesse é realizado com a relação que os nós possuem com os conteúdos e não com a relação que os nós possuem com os nós.

## 5.7 Protocolo PIFP

A solução que se propõe é então um protocolo de encaminhamento probabilístico para redes móveis ad-hoc que tem como objetivo aumentar o número de mensagens entregues. Aqui os nós são capazes de armazenar, carregar e encaminhar as mensagens das aplicações de um cliente até um nó produtor e vice-versa. O desafio de comunicação exige a construção de uma estratégia de encaminhamento capaz de lidar com os atrasos, as falhas de conexão e os longos períodos de tempo sem contacto.

A ideia que se propõe surgiu do esforço para criar um método que permita identificar quando e como é mais favorável encaminhar uma mensagem. Assim surgiram os seguintes princípios que regem a modelação do problema:

1. Os nós apresentam padrões de mobilidade, e com base neles é possível melhorar o desempenho do protocolo. Se um nó esteve em contacto com um produtor de conteúdo ou com um nó intermediário que possui determinado conteúdo, é provável que ele volte a ter contacto com esse conteúdo.
2. Quanto maior for a frequência de contacto com um produtor ou intermediário na posse de um conteúdo, maior a probabilidade de se voltar a encontrar com esse conteúdo.
3. Com a propagação de dados através dos nomes dos conteúdos e não dos endereços dos nós, é possível compreender os dados tornando mais fácil o seu armazenamento e reutilização.

Assim surge o PIFP um novo protocolo de encaminhamento para redes tolerantes a atrasos, concebido a partir do crescimento das NDNs.

Neste protocolo, a comunicação é impulsionada pelo consumidor. Quando um nó  $n$  pretende receber dados, mostra interesse neles enviando um PI.

Num ambiente disperso como as MANETs, onde não é possível conectar todos os nós ao mesmo tempo, o nó irá armazenar as mensagens até encontrar outro nó com uma probabilidade de entrega maior do que a dele.

### 5.7.1 Funcionamento do protocolo

O PIFP é um protocolo de encaminhamento tolerante a interrupções de comunicação para redes ad-hoc. Utilizando o paradigma de armazenar-transportar-enviar, o protocolo é responsável por escolher qual o próximo para o qual a mensagem deve ser encaminhada.

A mensagem permanece no *buffer* caso não possa ser encaminhada, até que ocorra uma nova conexão. Quando existir uma nova conexão, dependendo das decisões de encaminhamento, a mensagem é ou não transmitida.

O PIFP é um protocolo probabilístico. Constrói uma hierarquia baseada nas probabilidades para definir as políticas de encaminhamento, de forma a encaminhar eficazmente as mensagens de interesse para a obtenção de dados. Evita a disseminação de mensagens por todos os nós na rede. Localmente, cada nó mantém na FIB as previsibilidades de chegar a um conteúdo. Estas são calculadas através dos contactos com os conteúdos na rede.

Cada vez que existe contacto, são trocados dois tipos de informação entre os nós: informação de encaminhamento e um sumário de dados armazenados. A informação de encaminhamento corresponde às previsibilidades armazenadas na FIB. A Figura 16 mostra essa troca de mensagens em função do tempo.

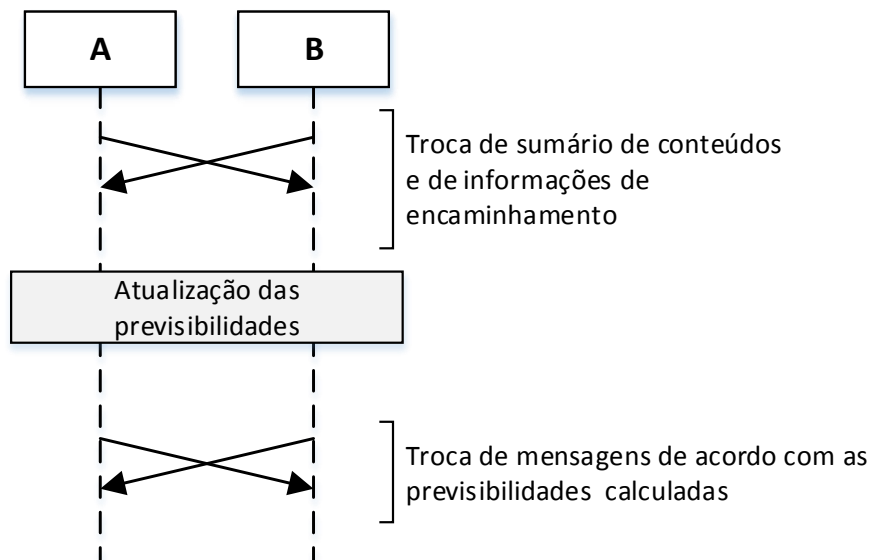


Figura 16 - Troca de mensagens

Com base nesta troca, é construída uma métrica probabilística, chamada de previsibilidade de entrega. Estas probabilidades são mantidas na FIB e auxiliam no encaminhamento de interesses.

O PIFP é inspirado no PROPHET onde cada nó calcula o valor da previsibilidade para cada conteúdo. O cálculo da previsibilidade reflete a frequência e o último instante com que se está em contacto com um dado conteúdo. Segue a premissa de que os nós seguem padrões de mobilidade, e com isso, é possível criar uma estatística determinando em que medida é possível confiar no nó para encaminhar uma mensagem até um nó produtor (ou intermediário que possua o conteúdo solicitado). Como já referido, o valor da previsibilidade é derivado de encontros históricos.

A frequência é utilizada para avaliar com que frequência o nó esteve em contacto com outro nó que possui o conteúdo. A previsibilidade aumenta sempre que exista contacto com outro nó com o mesmo conteúdo.

A previsibilidade também reflete a métrica de refrescamento que faz diminuir ou aumentar o valor da previsibilidade. Se existir um contacto recente com um nó que possua o conteúdo armazenado, o valor da previsibilidade para esse conteúdo é atualizado. A previsibilidade aumenta em relação a outros que já não possuem contacto há mais tempo.

Os encontros contribuem para determinar a previsibilidade  $P(a, c_i) \in [0,1]$ . Esta denota a probabilidade de o nó  $a$  entrar em contacto com o conteúdo  $i$ . O valor  $P(a, c_i)$  aumenta sempre que o nó  $a$  entra em contacto com um nó produtor ou intermediário que possua o conteúdo  $i$ . Diminui quando o nó  $a$  deixa de entrar em contacto com o conteúdo  $i$ .

O cálculo de  $P(a, c_i)$  é realizado em três fases. Em primeiro lugar, quando dois nós estabelecem conexão, através da Equação 9 é realizada a atualização de todas as previsibilidades de entrega que são mantidas na FIB. Utilizando esta equação, à medida que o tempo passa a probabilidade  $P(a, c_i)$  irá diminuir. Ao parâmetro  $\gamma$  foi atribuído o valor de 0,98 pois não são esperados encontros frequentes.

$$P(a, c_i) = P(a, c_i)_{antigo} \cdot \gamma^k \quad 0 \leq \gamma < 1 \quad (9)$$

A frequência dos contactos irá refletir a probabilidade, isto é, se o número de contactos com determinado conteúdo for baixo a probabilidade  $P(a, c_i)$  será reduzida. Nesta equação,  $\gamma$  cujo valor é igual ao valor sugerido pelo PROPHET, consiste numa constante de envelhecimento e  $k$  corresponde ao tempo (em segundos) passado desde a última vez que a métrica foi atualizada.

O parâmetro  $\gamma$  determina quão rapidamente a previsibilidade envelhece. Um valor mais baixo irá aumentar a velocidade de envelhecimento.

Depois disto, e com base no sumário de conteúdos recebido, ocorre a atualização da previsibilidade para cada conteúdo utilizando a equação 10.

O valor de  $P(a, c_i)$  aumenta sempre que o nó em contacto possui o conteúdo  $c_i$  armazenado.

$$P(a, c_i) = P(a, c_i)_{antigo} + (1 - P(a, c_i)_{antigo}) \times P_{init} \quad P_{init} \in [0,1] \quad (10)$$

Nesta equação,  $P(a, c_i)_{antigo}$  assume o valor de zero apenas no primeiro contacto com o conteúdo  $c_i$ , pois anteriormente não existia conhecimento sobre o conteúdo.

A variável  $P_{init} = 0.75$  é um fator de escala que fixa a taxa à qual a previsibilidade aumenta em encontros. Este cálculo é apenas realizado para conteúdos não armazenados.

Por fim, através das informações de encaminhamento, as previsibilidades para encontrar os vários conteúdos podem ser atualizadas. Para fazer isto, é utilizada a propriedade de transitividade do PROPHET que é adaptada para lidar com conteúdos. A regra dita que se o nó  $a$  se encontra frequentemente com o nó  $b$  e o nó  $b$  frequentemente se encontra com o nó  $c$ , então provavelmente o nó  $b$  é um bom nó para encaminhar agregados de  $a$  para  $c$ .

Para lidar com o paradigma de dados nomeados, esta regra foi modificada sendo rescrita de outra forma: numa conexão entre um par de nós  $a$  e  $b$ , se  $b$  frequentemente se encontra com dispositivos que possuem em cache o conteúdo  $c_i$  então provavelmente  $b$  será um bom nó para encaminhar mensagens de interesse do nó  $a$  para o conteúdo  $c_i$ . O valor resultante desta propriedade tende a ser mais baixo do que o valor calculado na Equação 10. Na Equação 11 é possível visualizar o resultado da regra reescrita onde  $P(b, c_i)$  é a previsibilidade de  $b$  encontrar  $c_i$ .

$$P(a, c_i) = P(a, c_i)_{antigo} + (1 - P(a, c_i)_{antigo})KP(b, c_i)\beta \quad \beta \in [0,1] \quad (11)$$

No PROPHET, a probabilidade do contacto entre o par de nós  $a$  e  $b$  é tida em conta. Agora esta previsibilidade foi substituída pela contante  $K$  que por defeito assume o valor de 0.75 pois não existe maneira de obter esta.  $\beta$  é uma constante de escala que ajusta o peso da propriedade

transitiva. Se  $\beta$  é definido como sendo zero, a propriedade transitiva apenas utiliza os encontros diretos com os conteúdos para o cálculo da probabilidade de entrega. Quanto maior for o valor de  $\beta$ , maior impacto terá a regra de transitividade para o aumento da probabilidade de entrega. O valor atribuído a  $\beta$  foi de 0,25.

Desta forma, a propriedade de transitividade possui menor impacto quando comparado com a utilização dos encontros diretos com os conteúdos. Assim, os nós com previsibilidades maiores serão aqueles que estiveram em contacto direto com os conteúdos. O número de mensagens enviadas será menor pois serão apenas transmitidas mensagens para os nós com maiores probabilidades de entrega.

### **5.7.2 Troca de Informação**

Os nós mantêm os objetos de dados armazenados na CS que frequentemente sofre atualizações. Também na FIB são mantidas as previsibilidades de chegar aos conteúdos que o nó não possui na CS.

Quando estabelecem conexão, é iniciado o processo de convergência, onde são trocadas as informações sobre os objetos de dados armazenados na CS e as previsibilidades mantidas na FIB.

A ideia é que cada nó anuncie os nomes dos conteúdos que possuem em cache para os outros nós que através das equações descritas anteriormente possam calcular as previsibilidades para cada conteúdo.

Os nós também trocam as tabelas FIB, onde as probabilidades para todos os conteúdos estão armazenadas. Estas probabilidades são utilizadas para calcular a previsibilidade através da regra de transitividade. Ao longo do tempo, a tabela de encaminhamento será preenchida com os nomes dos conteúdos associados a uma previsibilidade.

Eventualmente, a FIB irá conter todos os conteúdos presentes na rede. No entanto, este processo de convergência é lento devido a latência característica nas redes DTN.

### 5.7.3 Correspondência de conteúdos

Num esquema onde a informação é elemento principal da comunicação, um *router* encaminha os pacotes utilizando nomes em vez de endereços IP.

O esquema de nomeação utilizado poderá trazer alguns desafios. Em primeiro, os nomes podem possuir um comprimento variável. Uma pesquisa por nome pode percorrer uma grande quantidade de caracteres, verificando dezenas ou por vezes centenas de caracteres até encontrar exatamente o prefixo correspondente.

Em segundo, o número de entradas na tabela de encaminhamento também poderá ser maior que nas tabelas de encaminhamento IP. Atualmente a quantidade de informação aumentou. Outra dificuldade é o grande número de atualizações (publicação/eliminação) nas tabelas FIB, PIT e CS. Na presença de um ambiente dinâmico como as redes DTN, o número de atualizações é muito frequente, devido à incerteza se um nó vai encontrar novamente um conteúdo ou não, sendo necessário ajustar as tabelas em relação aos contactos existentes. Para implementar as tabelas foi escolhida uma variante dos Bloom Filters denominada por Counting Bloom Filters[41]. Os Bloom Filters permitem a redução do consumo da memória em comparação com os elementos que são armazenados diretamente. Basicamente, o Bloom Filter consiste numa sequência de bits em que no primeiro momento possui o valor 0. Para adicionar um elemento no bloom filter, são aplicadas funções de *hash* tais como MD5[42] e SHA-1[43]. Cada função resulta numa posição diferente de bit dentro do vetor de bits. Na inserção de um elemento, é então efetuado o *hash* do elemento  $k$  vezes definindo os bits a 1.

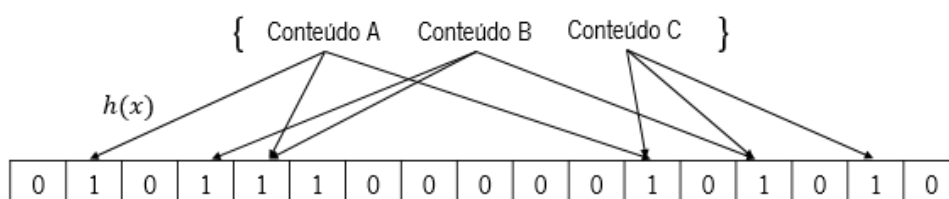


Figura 17 - Registrar Conteúdo no Bloom Filter (baseado em [41])

Apesar das vantagens descritas, os Bloom Filters não suportam a remoção de elementos do conjunto. Quando aplicada a função de *hash* não é possível reverter o processo.



Por esta razão, foram utilizados Counting Bloom Filters, onde cada entrada no Bloom Filter, não é um simples bit mas sim um pequeno contador. Quando um item é inserido, o correspondente contador é incrementado e quando é removido o contador é decrementado.

Nas NDNs, a utilização deste mecanismo na PIT pode eliminar o ciclo de interesses causado por falsos positivos.

#### 5.7.4 Processo de Consolidação

Este processo de consolidação consiste na troca de informação entre um par de nós. Quando existe uma nova conexão, os nós atualizam as suas previsibilidade através da Equação 9 que irá diminuir o seu valor. Quanto maior for a ausência de contacto, menor será o valor da previsibilidade. Após atualização das previsibilidades dá-se o processo de convergência, onde são trocadas informações de encaminhamento e de dados.

Através da informação recebida, é calculada a previsibilidade  $P(a, c_i)$  e é criada uma entrada na tabela de encaminhamento FIB para o conteúdo  $c_i$ . De modo a garantir que o processo de convergência seja realizado com sucesso, é necessário um tempo mínimo de contacto entre o par de nós.

Na receção do resumo de informações sobre os conteúdos do outro nó, é efetuada uma pesquisa na sua CS para verificar se existe alguma correspondência com algum conteúdo presente no resumo recebido, como descrito no Pseudo-código 1. Caso não possua o conteúdo, é aplicada a Equação 10 para obter o valor de  $P(a, c_i)$ . Na condição de não existir nenhuma entrada na FIB que corresponda ao conteúdo, é criada uma nova entrada inserindo o valor de  $P(a, c_i)$ .

##### Pseudo-código 1 – Processo de consolidação dos dados armazenados

1. **Se** existir conexão **fazer**
2.     Aplicar Equação 9 para o conjunto de previsibilidades
3. **Quando** receber sumário de dados do outro nó
4.     para cada conteúdo presente no sumário
5.         verificar se existe correspondência na CS
6.         **Se** não existir correspondência, **então**
7.             aplicar Equação 10
8.             inserir resultado na FIB
9.         **fim se**
10.        **senão**
11.         não fazer nada

A FIB num determinado momento irá possuir grande parte das previsibilidades dos conteúdos existentes na rede. O valor da previsibilidade irá sofrer alterações, devido à frequência de encontros com conteúdos ou pela ausência dos mesmos. Como referido anteriormente, este valor irá impactuar com a estratégia de encaminhamento.

Quando um nó não possui dados na CS, apenas é enviado um resumo da FIB, isto se existirem informações de encaminhamento. Na fase inicial, os nós não irão possuir informações de encaminhamento e dificilmente conteúdos armazenados, a não ser que sejam produtores. A estratégia de encaminhamento nesta fase não se aplica.

Na receção do sumário de informações de encaminhamento, que pode ser representadas como um vetor do tipo  $\langle \text{nome } c_i, P(b, c_i) \rangle$  é verificada na CS se existe correspondência para esse conteúdo. A previsibilidade recebida  $P(b, c_i)$  poderá ser resultado do contacto com conteúdos existentes no nó  $a$ , por isso é realizada uma verificação na CS do nó  $a$ , para que este não calcule a previsibilidade de um conteúdo que ele mesmo possui. Este processo é realizado em ambos os lados. No Pseudo-código 2 é apresentado o processo de convergência de informações de encaminhamento.

#### **Pseudo-código 2 – Processo de consolidação de informações de encaminhamento**

1. **Quando** receber sumário de informações de encaminhamento do outro nó
2.       **para** cada previsibilidade  $P()$  **fazer**
3.           verificar se existe correspondência na CS
4.           **se** não existir correspondência, **então**
5.               aplicar Equação 11
6.               inserir na FIB
7.           **fim se**
8.       **senão**
9.           não fazer nada

Assim, para cada  $P(b, c_i)$  recebida, é construída uma previsibilidade local ao nó ( $P(a, c_i)$ ) que é obtida aplicando a Equação 11. A transitividade é aplicada aumentando assim a previsibilidade. Na Equação 11,  $P(a, c_i)_{antigo}$  assume o valor zero se esta for a primeira vez que o nó tem conhecimento do conteúdo. Na FIB é criada uma nova entrada para este conteúdo caso não exista.

## 5.7.5 Encaminhamento de Pacotes de Interesse

Os interesses pendentes são enviados não só com base nas previsibilidades calculadas mas também tendo em conta o resumo, que foi recebido com os dados que o outro nó possui armazenado. Assim, após receção do dicionário os nós verificam se existem interesses pendentes com correspondência nesse dicionário e, se existirem, são enviados os interesses para obter esses dados.

Só após estes serem enviados, são transmitidos os outros interesses com base nas previsibilidades calculadas como demonstrado no Pseudo-código 3. Os nós são proactivos, criando previamente as mensagens de interesse para quando surgir um nó elegível enviar de imediato o interesse, recuperando tempo que é crucial neste tipo de redes.

### Pseudo-código 3 - Envio de Interesses Pendentes

```
1. Para cada nó conectado fazer
2.     para cada mensagem fazer
3.         se mensagem = "PI" então
4.             nome_  $c_i$  <- obter nome do interesse através da PI
5.             se BloomPIT possui nome_  $c_i$  então
6.                 se  $P(b, c_i) > P(a, c_i) \ \& \ P(b, c_i) > \text{last\_DP}(c_i)$  então
7.                     Adicionar ao buffer de mensagens de saída
8.                 fim se
9.             fim se
10.        fim se
11.        ordenar buffer por  $P(b, c_i)$ 
12.        enviar mensagens
```

Um nó elegível é aquele que a previsibilidade para atingir o conteúdo for superior à do nó que pretende enviar o interesse. Quando um nó envia um interesse regista a última previsibilidade do nó para o qual foi enviado. Da próxima vez apenas envia o interesse se a previsibilidade do outro for superior à sua e se possuir uma previsibilidade superior à última registada. Seguindo este modelo, é construída uma hierarquia enviando o interesse para um nível superior. O número de interesses enviados para a rede irá diminuir consideravelmente, com uma menor carga na rede. Com esta estratégia, existe uma forte probabilidade de atingir o conteúdo, visto que os interesses são enviados para nós que já, em algum momento possuíram contacto direto ou indireto com um conteúdo.

Como se pode visualizar no Pseudo-código 3 as mensagens são armazenadas num *buffer*, sendo posteriormente enviadas para o nó conectado. Antes de as mensagens serem transmitidas, são ordenadas por ordem decrescente de acordo com a previsibilidade de cada mensagem. As mensagens com maior previsibilidade são enviadas em primeiro lugar, por possuírem uma probabilidade maior de serem satisfeitas.

### 5.7.6 Processamento de Interesses Gerados

As aplicações quando necessitam de dados criam uma mensagem de interesse, como por exemplo um HTTP REQUEST durante a navegação na web. Estes podem ser satisfeitos localmente, desde que existam os dados correspondentes. Numa rede são criadas várias caches distribuídas que armazenam dados recebidos para entregar a outros nós. Estes dados podem ser reutilizados podendo satisfazer interesses originados pelo próprio nó.

#### Pseudo-código 4 - Geração de Novo Interesse

- |     |                                                              |
|-----|--------------------------------------------------------------|
| 1.  | <b>Para</b> cada interesse gerado <b>fazer</b>               |
| 2.  | nome_ $c_i$ ← obter nome do interesse                        |
| 3.  | <b>se</b> bloomCS possui nome_ $c_i$ <b>então</b>            |
| 4.  | devolver dados à aplicação                                   |
| 5.  | descartar interesse                                          |
| 6.  | <b>senão se</b> BloomPIT não possui nome_ $c_i$ <b>fazer</b> |
| 7.  | adicionar interesse à PIT                                    |
| 8.  | definir que é subscritor do interesse                        |
| 9.  | <b>senão</b>                                                 |
| 10. | descartar interesse                                          |

Quando um interesse é gerado, é verificada na CS se existe alguma correspondência, e se sim, o interesse é dado como satisfeito no nó. No Pseudo-código 4, é detalhado o processamento de um novo interesse quando este é gerado pelo próprio nó.

Se o conteúdo existir na CS, é devolvido para a aplicação que o solicitou. O interesse é descartado e definido como satisfeito no nó. Se não existir conteúdo, o interesse é inserido na PIT e definido como o nó é subscritor do interesse.

## 5.7.7 Processamento de Interesses Recebidos

Na receção de um pacote de interesse, é feita uma pesquisa na CS para saber se existe conteúdo para satisfazer o interesse. Uma mensagem com o conteúdo é enviada de volta para o consumidor, em caso afirmativo. Se não existir correspondência, o interesse é adicionado na PIT mas antes é verificada se a sua previsibilidade de entrega é maior que a do nó que enviou o interesse. Esta verificação é realizada de modo a garantir que o interesse é encaminhado salto-a-salto para nós que possuam uma probabilidade de entrega maior. No

Pseudo-código 5 é detalhado o processamento de uma mensagem de interesse recebida.

### Pseudo-código 5 - Receção de um Pacote de Interesse

```
1. Quando recebe mensagem PI
2.     interesse <- obter interesse
3.     nome_  $c_i$  <- obter nome do conteúdo
4.     se bloomCS possui nome_  $c_i$  então
5.          $c_i$  <- obter conteúdo
6.         criar mensagem de pacote de conteúdo
7.         enviar pacote de conteúdo
8.     fim se
9.     senão se bloomPIT possui nome_  $c_i$  então
10.        host <- Obter ID do host B
11.        adicionar Host à entrada na PIT
12.    fim se
13.    senão se bloomFIB possui nome_  $c_i$  então
14.        se  $P(b, c_i) > P(a, c_i)$  então
15.            adicionar nova entrada na PIT
16.        senão
17.            descartar mensagem de interesse
```

Na aceitação de um novo interesse, é verificado previamente se existe correspondência na PIT com este interesse. Caso exista, apenas é introduzida a face (neste modelo a face representa o identificador do nó que enviou a mensagem de interesse) pelo qual o interesse foi recebido. No caso de não existir interesse, é realizada a comparação  $P(b, c_i) > P(a, c_i)$ , e se esta verificar, é criada uma nova entrada na PIT.

## 5.7.8 Processamento de Conteúdos Recebidos

Na receção de um conteúdo, o nó verifica na sua PIT se existe correspondência. O interesse é dado como satisfeito quando chegar um pacote de dados que satisfaça o interesse e o nó seja subscritor. Um nó pode não ser subscritor e apenas possuir uma entrada na PIT devido à receção de uma mensagem de interesse proveniente de outro nó.

A entrada na PIT é eliminada quando existir correspondência com o conteúdo. Também é descartada a mensagem de interesse para aquele conteúdo. O processo de receção de mensagens de dados é descrito no Pseudo-código 6.

### Pseudo-código 6 - Receção de Pacotes de Dados

1. **Quando** recebe mensagem PC **faz**
2. nome<sub>c<sub>i</sub></sub> ← obter nome do conteúdo
3. **se** bloomCS possui nome<sub>c<sub>i</sub></sub> **então**
4.     descartar PC
5. **senão se** bloomPIT possui nome<sub>c<sub>i</sub></sub> **então**
6.     **se** nó é subscritor **então**
7.         adiciona conteúdo à CS
8.         incrementar interesses satisfeitos na rede
9.     **senão**
10.         adiciona conteúdo à CS
11.         remover interesse da PIT
12.         eliminar mensagem de interesse
13. **senão** descartar PC

Quando existe correspondência na PIT o pacote de conteúdos é adicionado à CS, caso contrário é descartado. Os dados na CS possuem um tempo de vida sendo descartados quando este expirar. De forma a eliminar redundâncias antes de qualquer operação é verificada na CS se existe correspondência e se sim, o pacote de dados é descartado.

Os pacotes de dados no PIFP seguem o caminho inverso da chegada dos interesses. Assim, só são enviados para os nós que possuem interesse na PIT, ou seja, para os nós que pretendem manter os conteúdos que possuem interesse.

Uma alternativa ao encaminhamento de pacotes de dados seria a utilização das faces registadas na PIT. Seria construída uma métrica probabilística para cada conteúdo, determinando a elegibilidade do nó receber ou não o conteúdo. Um nó ao receber vários interesses do mesmo conteúdo para diferentes nós, irá possuir maior probabilidade de entregar esse conteúdo do que

outros nós que receberam menos interesses para o conteúdo. A probabilidade de entregar um pacote de dados aumentaria sempre que um nó recebesse um interesse para um determinado conteúdo. Esta métrica determinaria com que frequência o nó entra em contacto com nós que possuem os mesmos interesses.

### 5.7.9 Protocolo PIFP Proativo

Assumindo que os nós têm espaço de armazenamento disponível eles podem armazenar os conteúdos mesmo que não exista correspondência na PIT. Esta abordagem foi explorada criando uma variante do PIFP chamada PIFPproactivo com armazenamento e transmissões proactivas. Este mecanismo poderá melhorar a performance na entrega de conteúdos mas aumentar a carga na rede.

Nesta variante, o mecanismo para distribuição de conteúdo foi modificado, removendo a restrição de transferir dados para o mesmo nó a partir do qual recebeu interesses. Quando é verificada correspondência na PIT e não existir, o conteúdo recebido pelo nó pode ser armazenado a menos que não exista espaço livre como mostrado no Pseudo-código 7.

#### Pseudo-código 7 - Receção de Pacotes de Dados versão Proativa

1. **Quando** recebe mensagem PC **faz**
2. nome\_c<sub>i</sub> <- obter nome do conteúdo
3. **se** bloomCS possui nome\_c<sub>i</sub> **então**
4.     descartar PC
5. **senão se** bloomPIT possui nome\_c<sub>i</sub> **então**
6.     **se** nó é subscritor **então**
7.         adiciona conteúdo à CS
8.         incrementar interesses satisfeitos na rede
9.     **senão**
10.         adiciona conteúdo à CS
11.         remover interesse da PIT
12.         eliminar mensagem de interesse
13. **senão**
14.     **se** espaço livre > tamanho do conteúdo **então**
15.         armazenar PC

Nesta versão é criada uma métrica de popularidade de interesses para enviar os conteúdos com base nesta métrica.

A popularidade é dada pela seguinte equação:

$$Popularidade_i = \frac{NR_i}{N} \quad (12)$$

Na Equação 12,  $NR_i$  consiste no número de interesses recebidos por um conteúdo e  $N$  no número total de interesses recebidos. Os dados são enviados com base nesta métrica, enviando aqueles com maior popularidade em primeiro.

Esta abordagem provê a entrega de conteúdos proactivamente, espalhando várias réplicas das mensagens de conteúdo com o objetivo de chegar a um nó que pediu o conteúdo.





## Capítulo 6

# Implementação da Plataforma de Simulação ICONE

Nas secções anteriores foram descritos os diferentes paradigmas e os algoritmos do protocolo PIFP. Neste capítulo, será descrita a implementação do PIFP no Opportunistic Networking Environment (ONE), um simulador para as DTNs baseado em Java. Como o ONE foi desenhado para suportar protocolos centrados no utilizador, difundindo mensagens com base nos endereços origem e destino, tornou-se necessário efetuar alterações na sua arquitetura de modo a simular o encaminhamento de mensagens, no quadro de uma arquitetura de dados nomeados.

Através da implementação do PIFP no ONE, será possível avaliar a adequação dos dados nomeados às redes DTN.

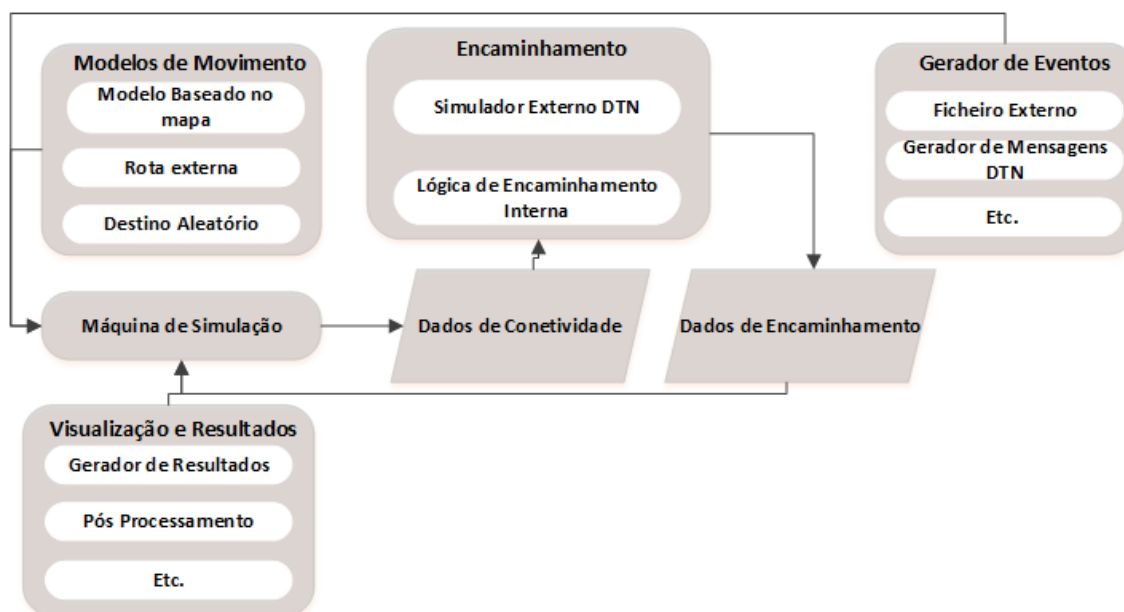
Inicialmente será feita uma introdução ao ONE e, posteriormente, a descrição das modificações realizadas no simulador.

### 6.1 Simulador ONE

O ONE é um simulador de redes oportunistas que oferece uma grande variedade de ferramentas para criar cenários de mobilidade próximos da realidade. Tem sido amplamente utilizado para trabalhos relacionados com redes DTN. O simulador ONE conta atualmente com aproximadamente 860 citações em trabalhos científicos da literatura.

O simulador suporta nós móveis com diferentes modelos de mobilidade, suporta o encaminhamento entre nós com vários algoritmos de encaminhamento DTN, a manipulação de mensagens, a visualização dos nós em movimento e a troca de mensagens, recolha e análise dos resultados.

A Figura 18 apresenta os componentes do ONE e a interação entre estes.



**Figura 18 - Arquitetura do simulador ONE**

A mobilidade dos nós pode ser gerada através de diferentes modelos de movimento. Pode importar dados de mobilidade de rotas do mundo real como por exemplo rotas GPS, utilizar destinos aleatórios ou utilizar um mapa. O componente "Modelos de Movimento" fornece os mecanismos para geração da mobilidade dos nós.

O simulador utiliza dados de conectividade como a localização dos nós, o alcance de comunicação e o débito de bits para estabelecer conexões. As decisões de encaminhamento são definidas com base nos protocolos implementados no módulo de encaminhamento. Toda a lógica para determinar quais as mensagens a enviar e para que nó é definida neste módulo. As mensagens utilizadas no simulador são mensagens *unicast*, existindo sempre uma origem e um destino. Estas são originadas no gerador de eventos.

Os resultados de simulação são recolhidos durante o tempo de simulação através dos módulos de resultados. O ONE permite obter resultados sobre as mensagens geradas, os eventos de conexão, as mensagens entregues, entre outros. Estes mecanismos estão presentes no módulo de visualização e resultados.

### 6.1.1 Capacidade dos Nós

Os nós no ONE são os agentes capazes de transportar, armazenar e encaminhar os dados para outros nós. São agrupados dependendo das diferentes capacidades de cada. O simulador modela um nó de forma a torná-lo o mais realista possível. Assim, cada nó tem um conjunto de características alteráveis, como a interface rádio, a capacidade de armazenamento, o tipo de movimento, o consumo de energia, e encaminhamento de mensagens.

A capacidade dos nós, o consumo de energia e a interface rádio são componentes simples que podem ser configurados a partir de um ficheiro de configuração externo. Outras capacidades dos nós como o movimento e o encaminhamento são configurados através de módulos que implementam o comportamento destas.

### 6.1.2 Modelos de Movimento

Uma rede DTN é formada por pessoas, automóveis, autocarros, e elétricos e cada um destes nós movimenta-se de forma diferentes. O ONE define algoritmos e regras para gerar o movimento de cada nó.

O simulador inclui três tipos diferentes de modelos de mobilidade. O mais básico é o modelo *Random Waypoint Movement* onde os destinos dos nós são gerados aleatoriamente. Para cenários mais realistas o ONE provê modelos baseados no mapa (*Map-Based Movement*), restringindo o movimento dos nós por caminhos pré determinados.

Nos modelos baseados no mapa, os nós movem-se utilizando somente as estradas e passeios da área do mapa. Os nós podem ser divididos em grupos e configurados para apenas utilizar uma parte do mapa.

Existem outros modelos baseados no mapa mais complexos como por exemplo o *Shortest Path Map-Based Movement*, que utiliza caminhos mais curtos para chegar a um destino e o *Working Day Movement* que pretende modelar a movimentação típica de uma pessoa durante as semanas de trabalho.

O simulador inclui por defeito um modelo de movimentação simples utilizando o mapa de Helsínquia.

### **6.1.3 Encaminhamento**

Enquanto os modelos de mobilidade decidem quais as rotas dos nós, os módulos de encaminhamento selecionam as mensagens a enviar e para quem enviar. No encontro entre nós, é necessário decidir quais as mensagens a enviar e, avaliar se o nó em contacto é um nó elegível para transportar a mensagem.

O ONE inclui alguns dos protocolos de encaminhamento mais conhecidos como por exemplo o Epidémico, o PROPHET, o Spray-and-Wait, entre outros.

A implementação de novos protocolos implica a extensão de classes já implementadas no ambiente de simulação, herdando funcionalidades básicas.

### **6.1.4 Geração de Mensagens**

No ONE a geração de mensagens pode ser feita através de ficheiros externos ao simulador ou através de uma aplicação para este efeito. As mensagens devem ser unidireccionais ou do género pergunta/resposta, possuindo uma fonte e um destino, o tamanho e a carga.

### **6.1.5 Relatórios e Visualizações**

A visualização de resultados pode ser feita via uma interface gráfica denominada por GUI (Graphical User Interface). Através desta interface é possível visualizar em tempo real a localização dos nós, as atuais rotas, as conexões entre os nós, o número de mensagens transportadas por cada nó, entre outros.

Esta interface é intuitiva pois permite obter uma imagem sobre o que está a ocorrer durante a simulação.

Como já mencionado, o ONE possui módulos de estatísticas que permitem obter o número de mensagens entregues, a quantidade de mensagens geradas, tempo médio de armazenamento das mensagens em *buffer*, entre outros.

## 6.2 Desenvolvimento da Plataforma ICONE

Para simular e avaliar o PIFP foi necessário estender o simulador, de maneira a suportar novos protocolos baseados no novo paradigma de dados nomeados.

Assim, foi criada uma plataforma que permite ao ONE simular protocolos de encaminhamento onde o elemento principal da comunicação são os dados e não os nós.

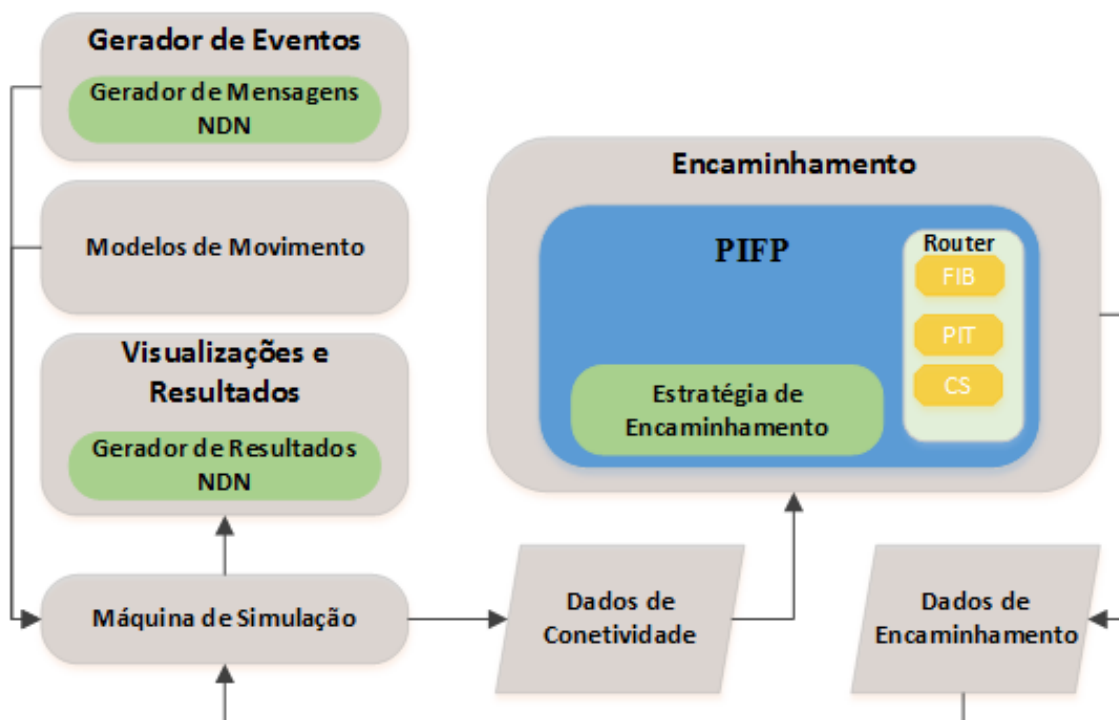


Figura 19 - Arquitetura ONE adaptado às NDNs

A Figura 19 ilustra os componentes básicos do simulador ONE destacando a verde os componentes NDN que foram introduzidos.

A seguinte lista resume os componentes implementados no ONE:

**Router NDN:** implementa uma abstração dos componentes básicos NDN como a PIT, a FIB e CS.

**Protocolo PIFP:** é o núcleo NDN pois permite interação com as camadas superiores e inferiores. Quando recebe um Interesse de uma camada superior envia este para as camadas inferiores. A estratégia de encaminhamento também é implementada neste módulo.

**Gerador de mensagens NDN:** O gerador de mensagens NDN foi implementado de maneira a tornar mais realista a simulação. Gera um conjunto de interesses seguindo uma distribuição semelhante à navegação web.

**Mensagens NDN:** com este novo paradigma foi necessário reestruturar as mensagens utilizadas pelo simulador.

**Gerador de resultados NDN:** recolhe dados ao longo do tempo permitindo construir um conjunto de resultados para avaliar o comportamento dos protocolos.

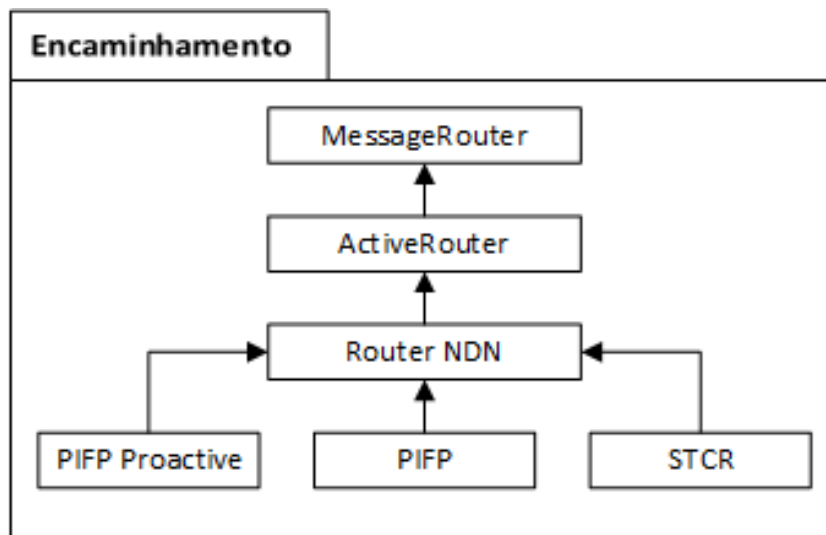
Nesta secção são descritos todos estes componentes, apresentando a sua implementação e adaptação ao simulador de forma a suportar o novo paradigma.

### 6.2.1 Router NDN

O router NDN é uma extensão que implementa os componentes básicos NDN como por exemplo a PIT, a FIB e a CS. Fornece também funcionalidades para lidar com a gestão destas estruturas NDN. Além destas funcionalidades, este módulo fornece a receção e o encaminhamento de mensagens. Este módulo foi criado com objetivo de lidar com os novos protocolos baseados nas redes NDN.

O ONE é um ambiente de simulação facilmente extensível. Os módulos implementados pelo simulador são carregados dinamicamente e, devido a isso, apenas é necessário criar novas classes e definir o seu nome no ficheiro de configuração, que é automaticamente carregado quando o cenário é iniciado.

Foi então criado um novo módulo que estende a classe “ActiveRouter” denominado por “Router NDN”. A classe “ActiveRouter” possui muitas funcionalidades que permitem lidar com as oportunidades de conexão e também lidar com transações a correr no momento. Também fornece funcionalidades para lidar com o armazenamento das mensagens. Na Figura 20 é possível visualizar a arquitetura do pacote de encaminhamento já com os novos módulos de encaminhamento. É também possível visualizar as dependências entre cada módulo.



**Figura 20 - Estrutura do Pacote de Encaminhamento**

Como mostrado na Figura 20, todos os módulos de encaminhamento necessitam de estender a classe “MessageRouter” que fornece a interface base com os módulos de encaminhamento. Esta classe trata do armazenamento de informações sobre as mensagens, isto é, das mensagens que o nó está a receber, sobre as mensagens que recebeu e que foram enviadas para a camada de aplicação e deixaram de existir no *buffer*, entre outros.

Para implementação do Router NDN foi necessário reescrever algumas funcionalidades do módulo “MessageRouter”. Na recepção de uma mensagem este módulo trata de identificar se esta chegou ao destinatário final ou não, utilizando para isso os endereços do nó. Foi necessário alterar estas verificações, pois a comunicação agora não segue o modelo fim-a-fim. São utilizadas estruturas como a PIT para verificar se o conteúdo chegou ao consumidor ou se a mensagem de interesse faz correspondência com algum conteúdo na CS.

O módulo “Router NDN” reusa as funções básicas que oferecem conectividade, entrega e recepção de mensagens, energia, e algumas funcionalidades relacionadas com o armazenamento de mensagens implementadas no módulo “ActiveRouter”.

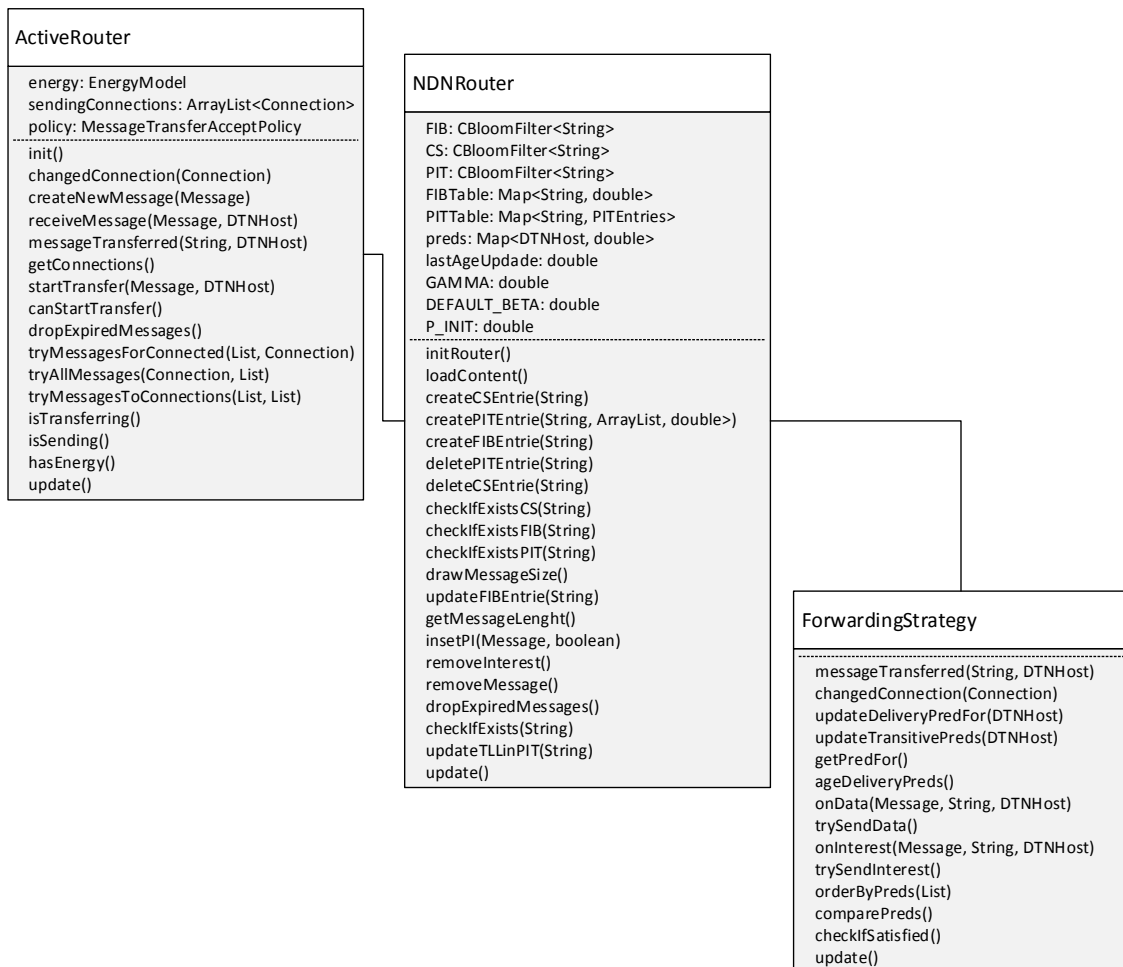
A Figura 21 mostra a dependência das classes de encaminhamento. Como podemos visualizar, o módulo “Router NDN” herda as funcionalidades fornecidas pelo “ActiveRouter” e o módulo “PIFP” herda as funcionalidades do módulo “Router NDN”.



Como mencionado, o módulo “Router NDN” implementa as estruturas PIT, FIB e CS. Estas possuem informação crucial ao encaminhamento de mensagens. O módulo provê funcionalidades para a gestão destas estruturas nomeadamente:

- **Gestão da PIT:** a criação, remoção e verificação das entradas PIT. Esta estrutura sofre atualizações frequentemente, seja criação de novos interesses, remoção dos interesses por exceder o tempo de vida ou por os interesses que são satisfeitos e necessitam de ser removidos da PIT.
- **Gestão da CS:** proporciona funcionalidades como a criação, remoção e verificação das entradas na CS. Esta estrutura permite a descrição dos pacotes de dados armazenados no *buffer*.
- **Gestão da FIB:** A FIB é a estrutura que sofre mais atualizações. Quando existe um contacto são trocadas informações de encaminhamento que poderão provocar modificações às entradas na FIB.

Também foi implementado o protocolo de encaminhamento STCR cuja implementação foi realizada de origem. Este protocolo foi implementado de acordo com o artigo. O módulo de encaminhamento chamado “STCR” herda as funcionalidades do módulo “Router NDN”. Este protocolo é simulado e comparado com os resultados obtidos pelo protocolo PIFP no Capítulo 7.



**Figura 21 - Dependência das Classes de Encaminhamento**

Além das funcionalidades descritas, o router NDN também executa a manutenção das mensagens armazenadas. Cada entrada na PIT possui um TTL associado e quando este expirar, as mensagens para o interesse são descartadas libertando espaço do *buffer*. Devido aos longos atrasos causados pelas DTNs foi implementado um mecanismo de relaxamento do TTL para as mensagens de interesse. Em algumas situações o TTL expira antes das mensagens atingirem o nó produtor e por esta razão foi implementado este mecanismo de relaxamento. Este mecanismo aumenta a diferença entre o tempo restante da entrada na PIT e o tempo de vida de uma mensagem chegada da rede com o mesmo interesse.

## 6.2.2 Geração de mensagem de dados

Na recepção de uma mensagem de interesse, o nó verifica na sua CS se existe correspondência e, em caso afirmativo, o nó gera uma mensagem com o conteúdo correspondente. O tamanho das mensagens pode ser configurado como um valor aleatório entre um máximo e um mínimo.

Este valor é pré-definido nas configurações e é estipulado entre os 50KB e 1MB. No entanto, podem ser definidos tamanhos fixos.

## 6.2.3 Carregamento de Conteúdo

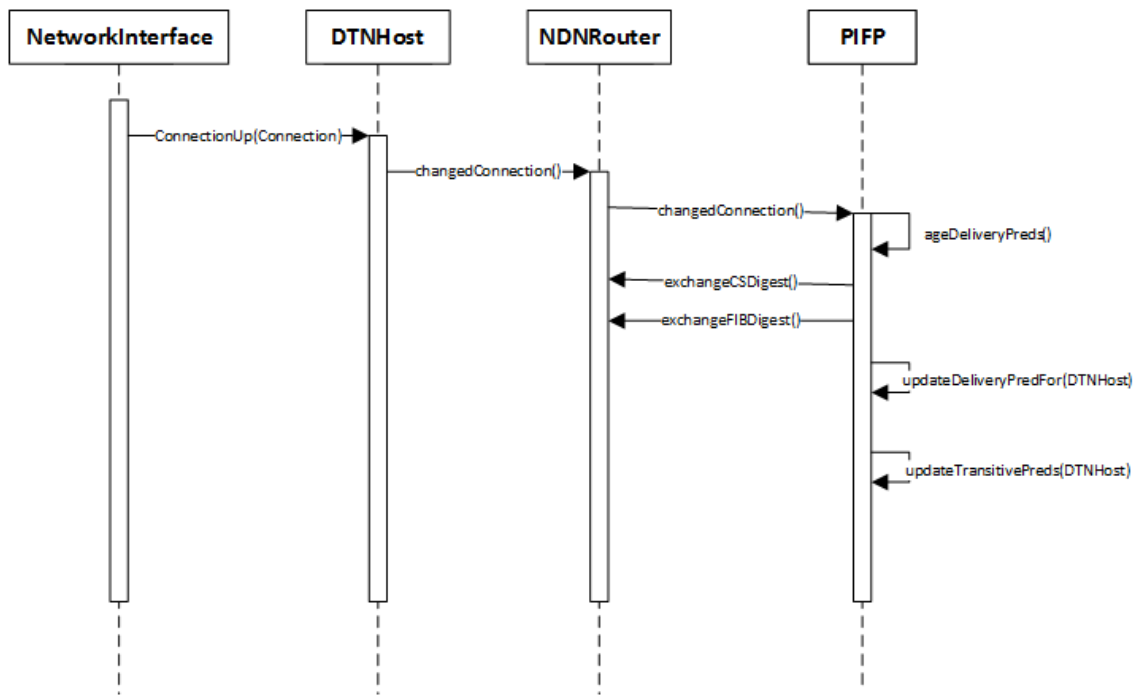
Os produtores são iniciados com um conjunto de conteúdos. Quando é iniciado o router NDN, através do método `loadContent()` os conteúdos são carregados para a CS. O nó fica responsável pelo armazenamento destes conteúdos. Os nós produtores são impedidos de gerar interesses para conteúdos de que são responsáveis. Por exemplo se um nó é responsável pelos conteúdos cujo prefixo é “/uminho”, este não pode gerar interesses para esses conteúdos. Através de um ficheiro de texto são descritos quais os conteúdos que o nó deve carregar. É definido um conjunto de produtores de conteúdos. No decorrer da corrida da simulação, esses conteúdos poderão estar disponíveis em *caches* dos diversos nós da rede.

## 6.3 Protocolo PIFP

Após o desenvolvimento da plataforma, o passo seguinte, foi o desenvolvimento do da estratégia de encaminhamento PIFP. Este módulo foi introduzido dentro do pacote de encaminhamento e estende o módulo “NDNRouter”. Utiliza todas as estruturas de dados implementadas assim como todas as funções providas pelo módulo “NDNRouter”.

O protocolo PIFP implementa toda a lógica de encaminhamento. Este protocolo decide que mensagens enviar e seleciona o melhor nó para enviar. Anteriormente, na secção 5.7, foi descrita toda a lógica do protocolo.

A Figura 22 mostra a fluxo desde o estabelecimento da conexão até ao cálculo das previsibilidades.



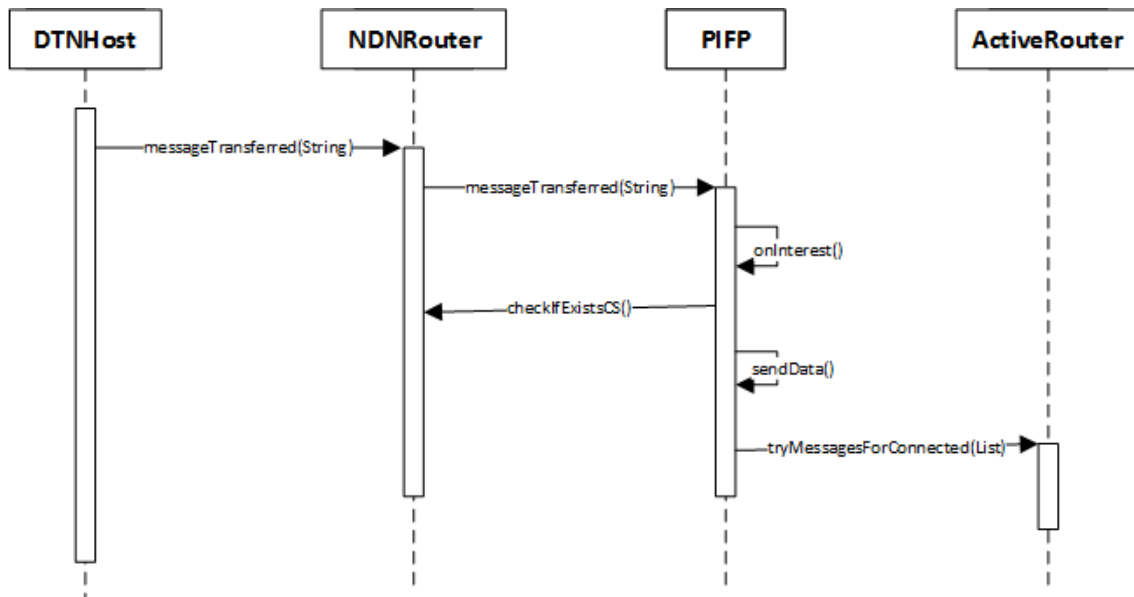
**Figura 22 - Diagrama de Sequência de Estabelecimento de Conexão**

Quando o objeto `NetWorkInterface` deteta uma nova conexão este chama o método `connectionUP` do objeto `DTNHost`. Por sua vez, o objeto `DTNHost` chama o método `changedConnection` do módulo `NDNRouter`. O `NDNRouter` procede de imediato à atualização das previsibilidades, quando é notificado pelo `DTNHost` que existe uma nova conexão.

De seguida, os nós trocam entre si as informações de encaminhamento e o resumo de dados que possuem na sua CS. Utilizando o resumo dos dados recebido, o nó chama o método `updateDeliveryPredFor` e calcula a previsibilidade para cada conteúdo presente no resumo recebido. Na receção de informações de encaminhamento, é calculada a previsibilidade através da regra de transitividade e para isso é chamado o método `updateTransitivePreds`.

Após este processo de convergência e a atualização das previsibilidades, os nós iniciam a troca de mensagens, enviando uma mensagem de interesse de forma a obter conteúdos. As mensagens de interesse são transmitidas utilizando dois processos. O primeiro consiste na deteção de dados do outro lado. O segundo processo consiste na avaliação se o outro nó é elegível utilizando as previsibilidades.

Na Figura 23 é possível visualizar o fluxo durante a receção de uma mensagem.



**Figura 23 - Diagrama de Sequência da Recepção de uma Mensagem de Interesse**

Quando o nó deteta uma nova mensagem, chama o método `messageTransferred` da classe `NDNRouter` e este, por sua vez, notifica o módulo da estratégia de encaminhamento. Na recepção de uma mensagem, o PIFP processa a mensagem, detetando se esta é uma mensagem do tipo interesse ou do tipo dados. Na recepção de uma mensagem de interesse, verifica se possui dados na CS e, em caso afirmativo, retorna uma mensagem de dados para o nó que enviou o interesse. Por outro lado, se o nó não possui dados, armazena a mensagem de interesse e cria uma entrada na PIT. Este processo é descrito no Pseudo-código 5.

## 6.4 Mensagens NDN

No ONE, as mensagens são direccionadas aos nós, utilizando para isso endereços de origem e fim. Sem endereços a informação que a mensagem transporta não tem importância. Nas NDNs, a comunicação não se baseia nos endereços fim-a-fim mas sim nos conteúdos. Esta mudança de paradigma levou à reestruturação da mensagem no ONE.

## 6.5 Gerador de Mensagens NDN

Para analisar o comportamento da solução proposta, foi necessário implementar um ambiente de simulação mais realista. Foi desenvolvido um gerador de mensagens de interesse baseado no tráfego HTTP.

Atualmente o tráfego HTTP representa uma grande parte de utilização da Internet nos dispositivos móveis (smartphones, tablets, portáteis). Baseado no estudo reportado em [44], foi possível compreender o comportamento da navegação na web e implementar um gerador de mensagens, seguindo o mesmo modelo. Todas as características são preservadas, como o número de pedidos por sessão, o tempo entre sessões e o tamanho das mensagens.

A Figura 24 ilustra um exemplo de navegação na web.

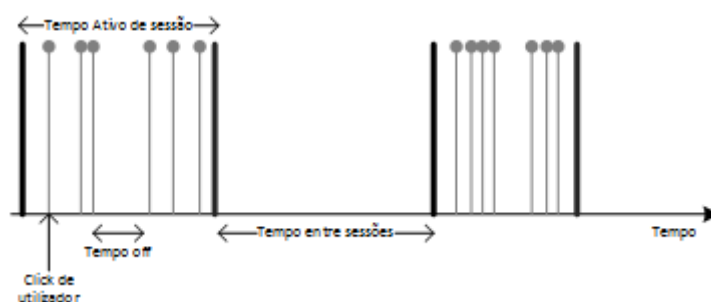
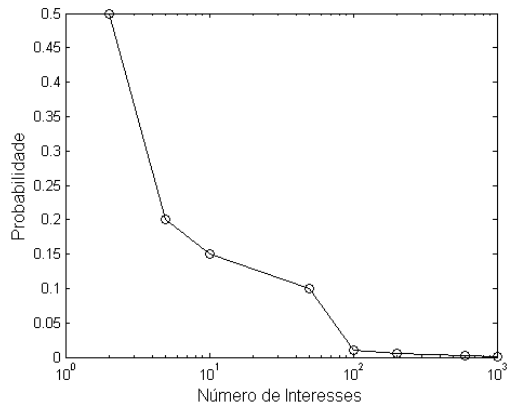


Figura 24 - Exemplo de sessão HTTP

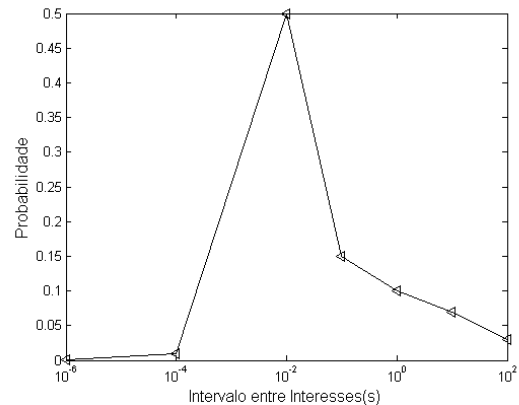
Durante a navegação na web, são gerados vários pedidos HTTP que no paradigma de dados nomeados se traduzem em interesses.

Como podemos visualizar na Figura 24, uma sessão possui um conjunto de interesses e mantém-se ativa durante um intervalo de tempo. Entre cada pedido existe um tempo inativo (onde não existem *clicks*) denominado por “tempo off”. Cada mensagem de interesse possui um tamanho em bytes.

Na Figura 25 é possível visualizar a distribuição de interesses por sessão. Aproximadamente em 50% das sessões são limitadas a 2 interesses. As restantes sessões possuem maior número de interesses, dependendo do tipo de aplicação (como por exemplo redes sociais, email, entre outros) e a duração da sessão.



**Figura 25 - Número de Interesses por sessão**

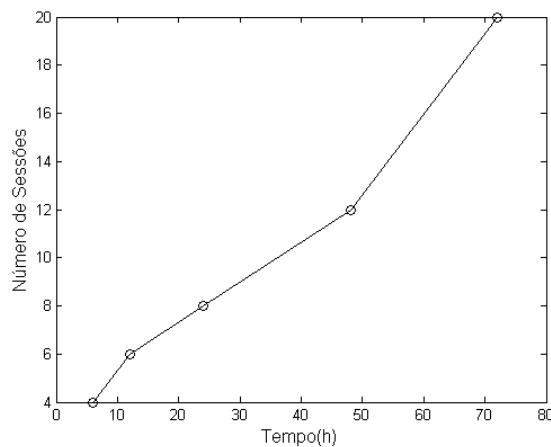


**Figura 26 - Intervalo entre Interesses**

A Figura 26 mostra a distribuição do intervalo de tempo entre pedidos na mesma sessão.

Na geração de mensagens de interesse, outro parâmetro a ter em conta é tamanho de um cabeçalho HTTP que varia entre os 200 bytes até 2KB. Cerca de 50% dos cabeçalhos HTTP varia entre os 700-800 bytes. O tamanho dos interesses gerados é baseado nestes valores.

Quanto ao número de sessões foi utilizado um valor especulado devido à inexistência desta informação na bibliografia.



**Figura 27 - Número de Sessões**

Na Figura 27 é ilustrada a distribuição do número de sessões em função do tempo. Um utilizador irá possuir um número de sessões. Esse número, é atribuído com base nos números apresentados no gráfico.

O gerador de mensagens utiliza as distribuições apresentadas para gerar um ficheiro com os eventos de criação de mensagens.

	Tempo (s)	Comando	ID	Origem	Tamanho	Nome
StandardEventReader	900.280	C	M1	10	700	uminho.pt/di
StandardEventsReader(File)	3600.280	C	M1	10	750	uminho.pt/di
readEvents(int)				...		
getHostAddress(String)	19200.019	C	M3	65	300	google.pt/ndn.pdf

**Figura 28 - Classe Leitor de Eventos**

**Tabela 7 - Excerto de um ficheiro para gerar Interesses**

A Tabela 7 mostra um excerto de um ficheiro para geração de mensagens de interesse. Este ficheiro consiste num ficheiro de texto simples que possui o tempo de geração do evento, o comando de criação, o identificador da mensagem, tamanho, nó de origem e o finalmente o nome do interesse.

Este ficheiro de texto é lido pelo simulador ONE. Quando o método “readEvents” é chamado, este lê linha a linha o ficheiro e verifica o tipo de evento que pode ser para criação de mensagens, remoção de mensagens do *buffer* ou estabelecer uma nova conexão. Cada evento é adicionado a uma lista de eventos que irá posteriormente ser processada pelo escalonador de eventos para efetuar a operação definida. Neste caso, gera uma mensagem do tipo interesse e adicionar esta ao *buffer* do nó de origem. A mensagem é gerada com os parâmetros especificados.

## 6.6 Gerador de Resultados NDN

Ao longo do tempo o ONE realiza um relatório de todas as atividades que ocorrem durante a simulação. Os relatórios permitem ao utilizador após a simulação fazer uma análise pormenorizada desta.

Com a introdução dos dados nomeados, estes relatórios deixaram de fazer sentido pois as mensagens são processadas de outra forma. Neste paradigma a comunicação não segue o paradigma fim-a-fim e, é necessário diferenciar o tipo de mensagens, de modo a perceber a quantidade de mensagens geradas para cada tipo.

As métricas obtidas através do módulo ReportNDN são:

**Número de interesses gerados** que consiste no número de mensagens de interesse geradas por todos os nós presentes na rede.



**Número de interesses enviados**, onde é considerada apenas a primeira transmissão da mensagem de interesse. As retransmissões seguintes já não são tidas em conta.

**Número de interesses satisfeitos** consiste no número de pedidos que foram satisfeitos através da receção de conteúdos.

**Número de interesses satisfeitos no nó** que consiste no número de interesses que foram satisfeitos por conteúdos já presentes na cache. Quando surge um interesse, o nó verifica se existe algum conteúdo na CS que o possa satisfazer.

**Número de mensagens de dados enviados**, onde é considerada apenas a primeira transmissão, descartando todas as posteriores retransmissões.

**Atraso médio** que consiste no atraso que ocorre desde a geração de um interesse até à receção de uma mensagem de dados que satisfaça este interesse.

**Tamanho médio das mensagens**, é uma métrica que mede o tamanho médio em bytes das mensagens de dados.

**Número de saltos**, é uma métrica que mede o número de saltos de uma mensagem de dados desde a sua criação até atingir o consumidor final. O consumidor final é o nó que possui interesse na PIT e seja subscritor desse interesse.

**Energia consumida** ao longo da simulação. As transmissões de mensagens e as conexões que são estabelecidas gastam energia aos dispositivos móveis.

Os resultados obtidos são armazenados em ficheiros para posteriormente serem carregados por outros programas de análise.

# Capítulo 7

## Resultados da Simulação

Neste capítulo é apresentado um conjunto de resultados de simulação com o propósito de validar e avaliar a performance do PIFP.

Adicionalmente foi incluída uma versão proativa do PIFP, aceitando dados da rede mesmo que não haja interesse, desde que haja espaço em cache. Para fazer uma análise comparativa, foi implementado também na plataforma ICONE o protocolo STCR.

O STCR é um esquema de recuperação de conteúdos para redes tolerantes a atrasos MANET baseado em relações sociais. Este constrói uma hierarquia social a agrupa os nós em aglomerados<sup>9</sup> utilizando o algoritmo K-means[45]. Utiliza as relações sociais para difundir as mensagens de interesse e de dados.

Para avaliar o desempenho/eficiência do PIFP foram utilizadas as seguintes métricas nas experimentações:

**Fração de interesses satisfeitos:** quantidade de interesses satisfeitos quer pela receção de dados a partir da rede ou por dados armazenados em cache.

**Latência:** Tempo médio desde a geração de uma mensagem de interesse até a chegada de um pacote de dados que satisfaça o interesse.

**Sobrecarga de mensagens:** quantidade de mensagens de interesse e de dados existentes na rede.

**Energia consumida:** quantidade média de energia consumida durante o tempo de simulação.

**Número de saltos:** número de nós no trajeto desde a fonte até ao destino.

Nas próximas secções serão descritos as configurações de simulação, os cenários de simulação e a correspondente análise dos resultados.

---

<sup>9</sup> Clusters, na terminologia inglesa

## 7.1 Configurações para a Simulação

As simulações foram realizadas utilizando o simulador ONE para avaliar a performance do protocolo PIFP. O ONE foi estendido de maneira a suportar novos protocolos baseados em dados nomeados como descrito na secção 6.

O PIFP foi experimentalmente avaliado e comparado com outros protocolos baseados em dados nomeados como a versão proactiva do PIFP e o STCR.

Todos os protocolos são simulados utilizando os mesmos parâmetros de configuração. O cenário de simulação corresponde ao mapa de Helsinquia, na Finlândia que inclui ruas, passeios e passadeiras. A área de simulação é de 4500m × 3400m.

Na Tabela 8 é apresentado um resumo das configurações comuns a todos aos cenários que irão ser simulados.

<b>Parâmetro</b>	<b>Valor</b>
Tempo de Simulação	86400s
Área de Simulação	4500m × 3400m
Velocidade do Nó (m/s)	Pedestres: 0.5 a 1.5
	Carros: 2.7 a 13.9
	Elétricos: 10 a 30
Tempo de Vida das Mensagens	8 horas
Interface Rádio	Bluetooth
Alcance de Transmissão	Pedestres e carros: 10m
	Elétricos e Pedestres especiais: 1Km
Taxa de Transmissão	Interface simples: 200k
	Interface de alta velocidade: 10M
Capacidade de Armazenamento	Consumidores: 50MB
	Produtores: 100MB

**Tabela 8 - Parâmetros Gerais de Configuração**

O cenário de simulação é composto por grupos de nós, que podem ser carros, pedestres e elétricos. O modelo de movimento varia consoante o tipo de nó. Cada pedestre utiliza o modelo

Shortest Path Map-Based Movement. Os carros são forçados a andar nas ruas utilizando o mesmo modelo de movimento dos pedestres. Os elétricos utilizam o modelo Routed Map-Based Movement Model. No movimento dos nós são definidas velocidades. Os pedestres deslocam-se entre 0,5m a 1,5 m/s. Os carros circulam a uma velocidade de 2.7 a 13.9m/s e os elétricos circulam com uma velocidade de 10 a 30 m/s.

Na camada física, todos os nós utilizam a mesma interface de rádio, Bluetooth. Alguns nós possuem duas interfaces, uma interface básica e outra com maior alcance e uma taxa de transmissão superior. Uma interface simples tem uma taxa de transmissão de 200k enquanto uma interface de alta velocidade possui uma taxa de 10M. Os elétricos possuem interfaces de alta velocidade enquanto os pedestres e os carros possuem interfaces simples.

A capacidade de armazenamento varia entre consumidores e produtores. Aos produtores foi atribuído 100M para capacidade de armazenamento enquanto aos consumidores foi atribuído 50M.

## **7.2 Cenários de Simulação**

Para avaliar o comportamento do PIFP foram implementados dois cenários de simulação diferentes.

O primeiro cenário de rede DTN consiste em dois grupos de pedestres (um com trinta e dois elementos e outro com trinta), um grupo de carros (trinta e dois elementos) e dois grupos de elétricos com dois elétricos cada. Um total de 98 nós. Neste conjunto de nós, existem 8 que são produtores de conteúdos, onde 4 são elétricos e 4 pedestres.

O segundo cenário de rede consiste numa rede esparsa. O número de nós foi reduzido para 49. Este cenário é constituído por cinco grupos de autores, onde dois grupos são pedestres (um de 24 elementos e outro de 8 elementos), um grupo de carros (15 nós) e dois grupos de elétricos (1 elemento cada grupo). Neste cenário, foi definido um conjunto de nós produtores que são: 2 elétricos com maior número de conteúdos e dois pedestres especiais (corredores).

Estes dois cenários foram seleccionados para avaliar a performance do PIFP em redes densas e esparsas. Tal como no PROPHET, o PIFP poderá ter maior taxa de entrega em redes mais densas. A convergência dos nomes também irá ser mais rápida pois existem mais conexões

disponíveis. Numa rede esparsa, dada a falta de conectividade, a taxa de satisfação de interesses irá ser menor com maiores atrasos de entrega de mensagens.

Fundamentalmente, irá ser avaliado o impacto da densidade da rede e o do tamanho das mensagens.

### 7.3 Resultados Experimentais

Nesta secção será efetuada a avaliação da adequação do protocolo PIFP às DTNs. É analisado o impacto da densidade da rede utilizando os dois cenários descritos na secção anterior. Também é avaliado o impacto do tamanho das mensagens. Por fim, é efetuado um apresentado e analisado o número de saltos e a energia consumida. Para obter resultados comparáveis, foram efetuadas as mesmas simulações para o protocolo STCR.

#### 7.3.1 Impacto da Densidade da Rede

Para avaliar o impacto da mobilidade dos nós na rede, foram criados cenários com diferentes densidades como descrito anteriormente. O objetivo é perceber qual o impacto do número de nós móveis no desempenho dos protocolos de encaminhamento. Nesta subsecção são apresentados os resultados comparando os dois cenários. O tamanho das mensagens de dados varia entre 50k e 1M. As mensagens de interesse seguem a distribuição descrita na Subsecção 6.5.

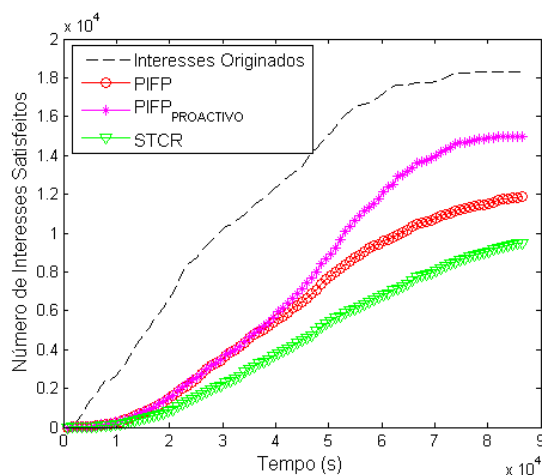


Figura 29 - Interesses Satisfeitos (Cenário 1)

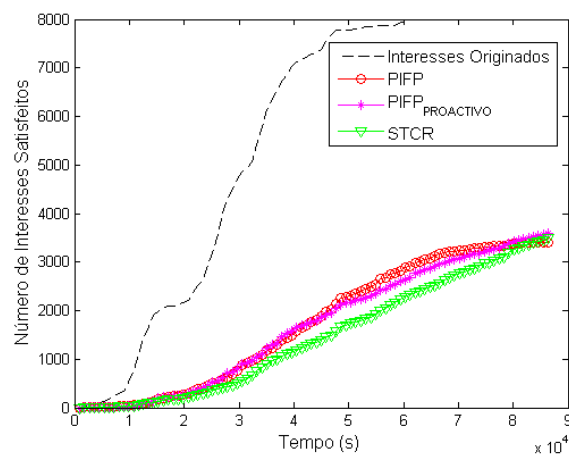


Figura 30 - Interesses Satisfeitos (Cenário 2)

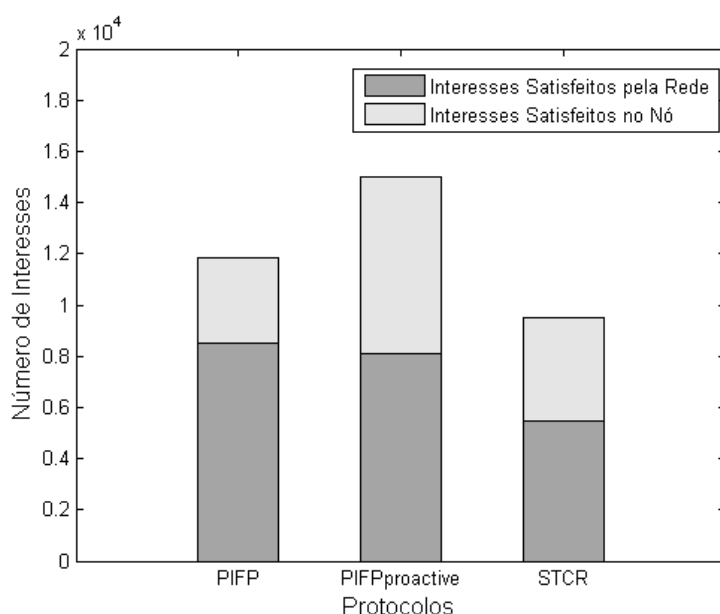
Em primeiro lugar, é analisado o resultado de interesses satisfeitos tal como apresentado nas figuras 30 e 31. Comparando os dois cenários, representados pela Figura 29 (cenário 1) e Figura 30 (cenário 2) é possível verificar que o número de interesses satisfeitos aumenta com o número nós na rede.

Qualquer nó na rede, incluindo os utilizadores podem se tornar em caches, a qualquer instante, possibilitando a criação de uma rede global de armazenamento e de distribuição de conteúdos. Assim quantos mais nós existirem na rede, maior é o número de *caches* distribuídas.

Comparando as três abordagens, pode-se ver que a versão proactiva do PIFP tem maior número de interesses satisfeitos em ambos os cenários. Isto deve-se ao facto da versão proactiva enviar os pacotes de dados para todos os nós mesmo que estes não tenham interesse nos dados. Isso faz com que existam várias cópias dos conteúdos espalhadas pela rede, tornando mais fácil a sua obtenção. Também devido à compreensão dos dados é possível reutilizar os pacotes de conteúdo, podendo satisfazer os próprios interesses do nó.

A versão proactiva do PIFP obteve aproximadamente 82% de satisfação de interesses no cenário 1. O protocolo PIFP supera o STCR. A taxa de satisfação de interesses no PIFP é aproximadamente 65% enquanto para o STCR é de 50%.

No cenário 2 as estratégias possuem uma taxa de satisfação semelhante. O protocolo PIFP conseguiu satisfazer aproximadamente 42% dos interesses. O protocolo STCR obteve maior taxa de satisfação com 44%.

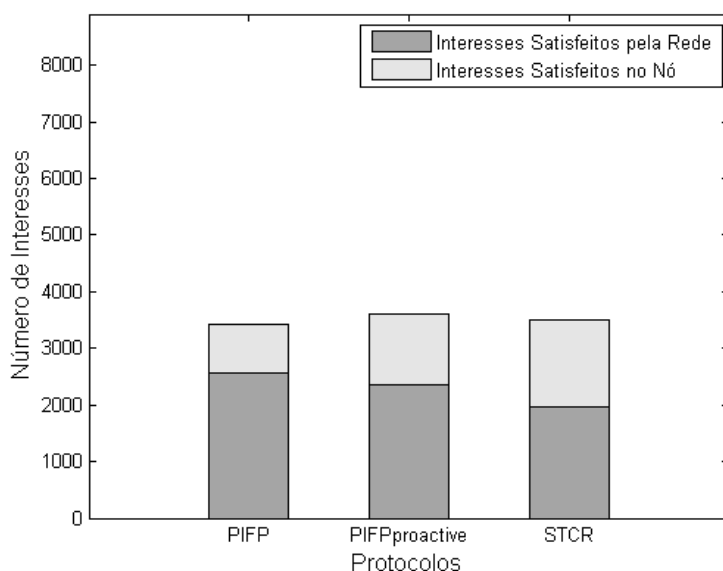


**Figura 31 - Interesses Satisfeitos na rede e no nó (Cenário 1)**

A Figura 31 mostra o número de interesses satisfeitos localmente pelo nó e através da rede. Quando um nó gera interesses ele verifica na sua CS se existem dados para satisfazer esses interesses. Se existirem dados, então os interesses são satisfeitos localmente.

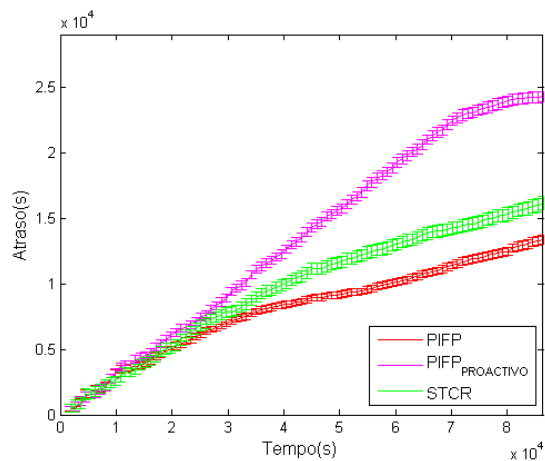
O PIFP apresenta um maior número de interesses satisfeitos pela rede do que localmente. Na solução proactiva do PIFP aproximadamente 40% dos interesses são satisfeitos localmente. O PIFP e o STCR possui respetivamente 25% e 40% dos interesses satisfeitos localmente.

A Figura 32 apresenta o número de interesses satisfeitos na rede e localmente pelo nó numa rede esparsa (cenário 2). Como já mostrado pela Figura 30, o número total de interesses satisfeitos é idêntico em ambas as estratégias. O número de interesses satisfeitos na rede é maior no protocolo PIFP. O STCR obtém um maior número de interesses satisfeitos pelo nó comparado com as outras estratégias.

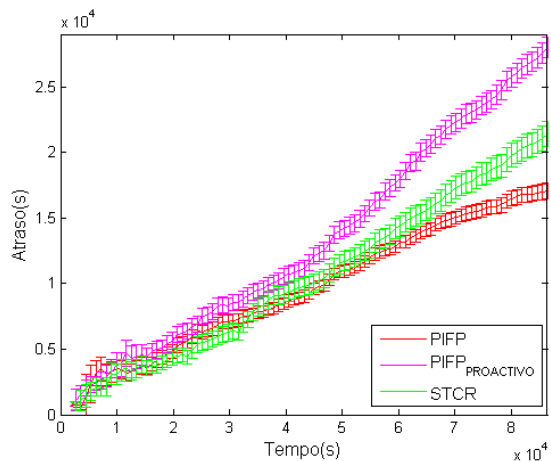


**Figura 32 - Interesses Satisfeitos na Rede e no Nó (Cenário 2)**

Como mostrado nestes dois últimos gráficos o armazenamento de dados pode trazer benefícios para o desempenho dos protocolos. Com o armazenamento de dados, qualquer nó na rede poderá responder com mensagens de dados aos interesses recebidos. Os dados são enviados a partir dos nós mais próximos do consumidor. Como resultado, o PIFP tira partido dos dados armazenados na rede aumentando significativamente o seu desempenho.



**Figura 33 - Atraso (Cenário 1)**



**Figura 34 - Atraso (Cenário 2)**

A Figura 33 e Figura 34 ilustram o atraso médio ao longo do tempo para os três protocolos. Apenas o atraso para os interesses satisfeitos através da rede são apresentados pois quando existem dados em cache e surge um interesse para esses dados este é imediatamente satisfeito.

Uma vez que um pacote de interesse chegue ao produtor ou a um nó intermediário que possua conteúdo que possa satisfazer o interesse, é enviada de volta uma mensagem de dados para o consumidor. No caso da versão proactiva, as mensagens de dados são enviadas para qualquer nó com espaço livre em cache. No caso do PIFP, as mensagens de dados são enviadas para os nós que têm interesse nesses na PIT. Finalmente, no caso do STCR, os dados apenas são enviados para nós com forte relação social com o consumidor.

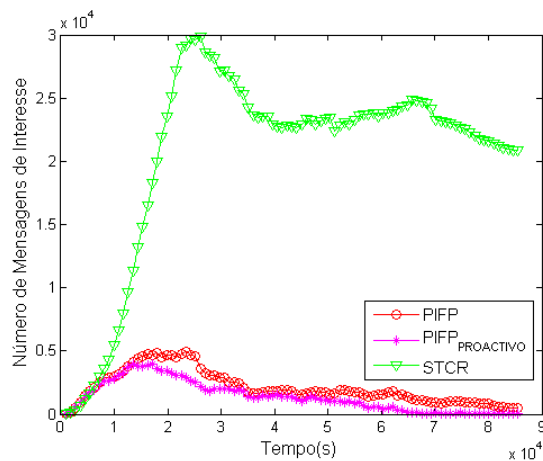
Aparentemente, o menor atraso deveria ocorrer em estratégias que possuem mais caminhos alternativos para o consumidor, que entre estas três estratégias é a versão proactiva do PIFP. No entanto, a inundação de pacotes de dados na rede utilizando esta estratégia, limita o armazenamento para subseqüentes interesses. Esta razão faz com que o tempo de espera para a versão proactiva do PIFP seja maior que o PIFP. O PIFP obtém mais caminhos alternativos para o consumidor que o STCR.

No início da simulação, o envio de pacotes de interesse pelos nós STCR é mais rápido pois este é feito para nós com maior centralidade. Ao contrário do STCR, o protocolo PIFP envia pacotes de interesse somente para nós com conhecimento do conteúdo associado. Devido a baixa convergência das tabelas de encaminhamento no início da simulação, muitos conteúdos não são conhecidos. Isto justifica os bons resultados do STCR no início da simulação. Na Figura 33 podemos confirmar o que foi dito até agora. Na Figura 34 (cenário 2), o comportamento foi similar

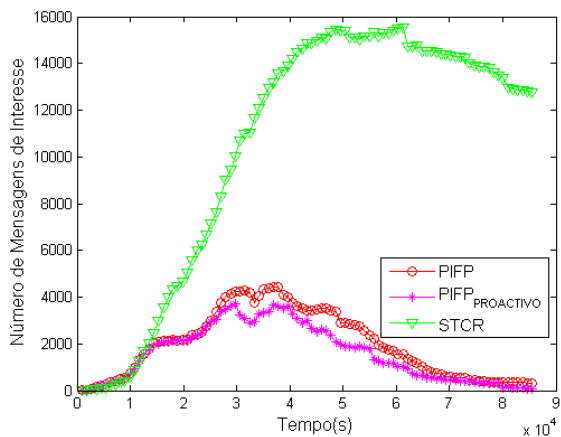


para todas as abordagens. O atraso aumentou quando comparado com Figura 33 (cenário1). Isto deve-se ao fato de a rede não ser tão densa como no cenário 1. No cenário 2 o número de caminhos disponíveis para chegar ao consumidor diminui, causando atraso na entrega de dados. Também foi analisada a carga de interesses e a carga de dados em ambos os cenários.

Como mostra a Figura 35 e a Figura 36, o STCR tem maior número de interesses na rede, quando comparado com os restantes protocolos. O STCR inicialmente transmite os interesses para os nós com maior centralidade e de diferentes *clusters*. Quando o produtor dos conteúdos ou um nó intermediário que possua conteúdos é conhecido, é introduzido no cabeçalho da mensagem de interesse o endereço do nó e esta é enviada para esse nó utilizando a relação social.



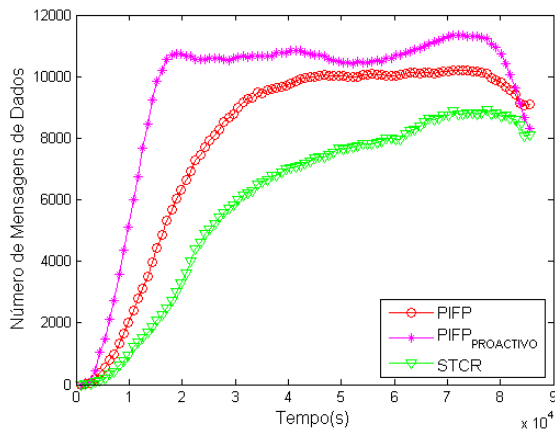
**Figura 35 - Carga de Interesses (Cenário 1)**



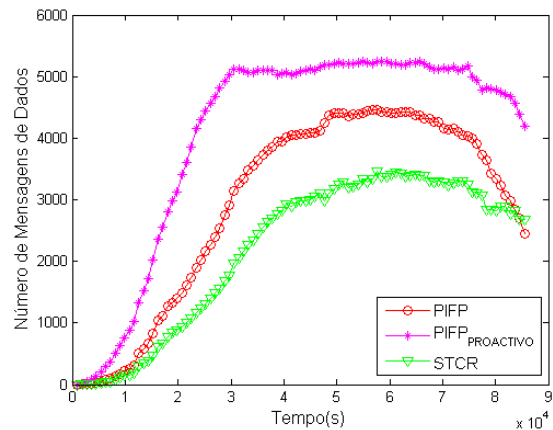
**Figura 36 - Carga de Interesses (Cenário 2)**

Aparentemente, esta abordagem faz com que existam mais mensagens de interesse que utilizando o protocolo PIFP, que envia interesses só para nós com maior previsibilidade de chegar ao conteúdo. Além disso, o PIFP descarta mensagens de interesse quando esta é satisfeita ou o seu TTL expirar. No STCR quando o TTL da mensagem de interesse expirar, esta não é descartada mas é enviada de forma epidémica o que aumenta o número de réplicas na rede. A razão pelo STCR enviar epidemicamente os interesses é para estes atingirem o mais rápido possível um produtor de conteúdos.

Comparando a estratégia PIFP e a sua versão proactiva, é possível verificar que o PIFP possui maior número de interesses na rede. Isto deve-se ao facto de a versão proactiva possuir uma maior taxa de satisfação de interesses localmente. Como os dados são enviados proactivamente, quando surgir um interesse estes são imediatamente satisfeitos não existindo assim encaminhamento de interesses.



**Figura 37 - Carga de Dados (Cenário 1)**



**Figura 38 - Carga de Dados (Cenário 2)**

A Figura 37 e a Figura 38 apresentam o número total de pacotes de dados para os cenários 1 e 2, respetivamente. Como descrito anteriormente, a versão proactiva do PIFP envia pacotes de dados sempre que os nós vizinhos tenham espaço livre em cache. Devido a isto, esta estratégia tem o maior número de pacotes de dados na rede em ambos os cenários. Como se pode visualizar, existe uma subida acentuada desde o início pois desde que um nó produtor receba um interesse este propaga os dados proactivamente para estes atingirem mais rápido o consumidor final. O PIFP envia os pacotes de dados para todos os nós que possuam interesse na PIT. O protocolo STCR envia pacotes de dados utilizando a mesma estratégia do encaminhamento de interesses quando é conhecido o produtor dos conteúdos, ou seja envia para os nós com uma relação social superior.

Comparando os protocolos PIFP e STCR, é possível visualizar que em ambos os cenários o protocolo PIFP possui um maior número de pacotes de dados. Isto deve-se ao facto do maior número de interesses que atingiram os nós produtores e também à estratégia de recuperação de dados. Os pacotes de dados no PIFP são enviados seguindo o caminho inverso, isto é, os dados são enviados para os nós que possuem na PIT o interesse correspondente aos dados.

O número de mensagens de dados foi reduzido para aproximadamente metade no cenário 2. A razão pelo quando o número de mensagens é reduzido deve-se ao facto do numero de nós no cenário 2 ser metade e desta forma o número de interesses originados também é metade.

### 7.3.2 Implicação do Tamanho da Mensagem

Nos cenários anteriores, o tamanho das mensagens podia assumir o valor aleatório entre 50k e 1M.

Nesta subsecção pretende-se analisar o impacto do tamanho das mensagens na performance do PIFP. O experimento foi avaliado correndo cenários com diferentes tamanhos de mensagens.

Inicialmente foi definido um intervalo de tamanho de mensagens que varia os 50K e os 150K.

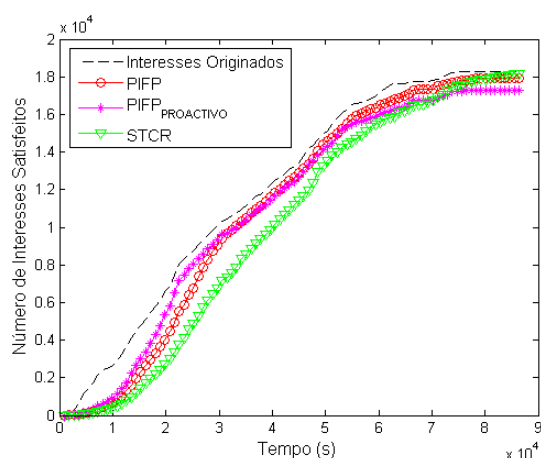


Figura 39 - Interesses Satisfeitos (50-150k)

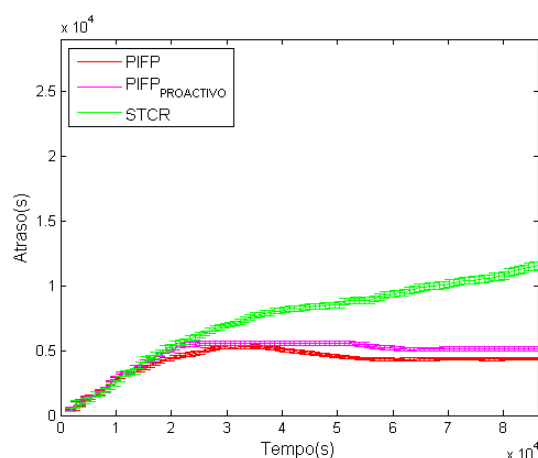
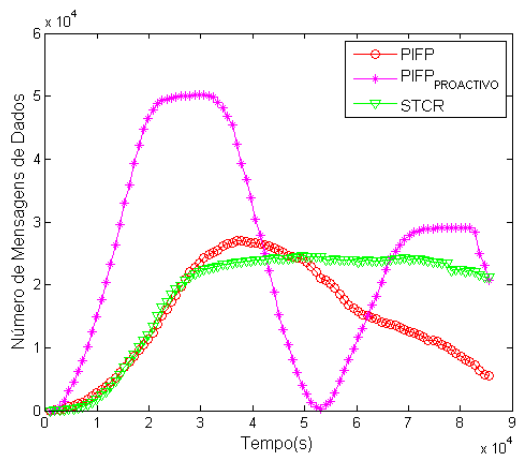


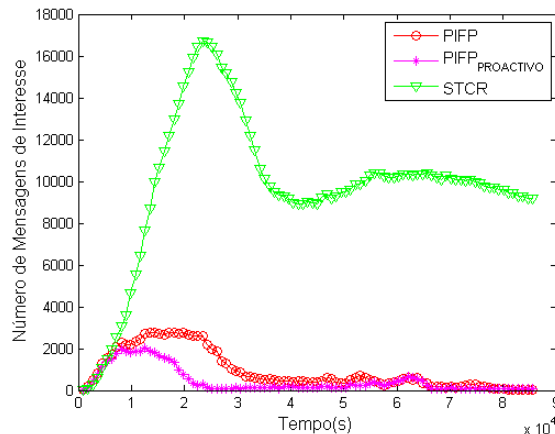
Figura 40 - Atraso Médio (50-150k)

A Figura 39 mostra o número de interesses satisfeitos neste experimento. Como é possível visualizar a taxa de satisfação é aproximadamente 100%. A estratégia PIFP e STCR possuem uma taxa de satisfação muito semelhante e a estratégia PIFPproactivo possui a menor taxa de satisfação.

Na Figura 40 é possível visualizar o atraso médio. O protocolo STCR possui maior atraso quando comparado com as outras duas estratégias.



**Figura 41 – Carga Dados (50-150k)**



**Figura 42 - Carga Interesses (50-150k)**

Através da Figura 41, é possível visualizar a carga de dados presentes na rede. A estratégia PIFPproactivo possui maior carga de dados na fase inicial, devido à sua proactividade como mencionado anteriormente. A linha do PIFPproactivo tem uma queda abrupta aproximadamente nos 30000 segundos. Isto deve-se ao TTL definido para as mensagens de dados e também aos dados que foram removidos do *buffer* por atingirem o consumidor final. Quando um pacote de dados atinge o consumidor final é armazenado na memória. Se o nó não tiver recebido interesses de outros nós, requisitando o mesmo conteúdo os dados são removidos do *buffer*.

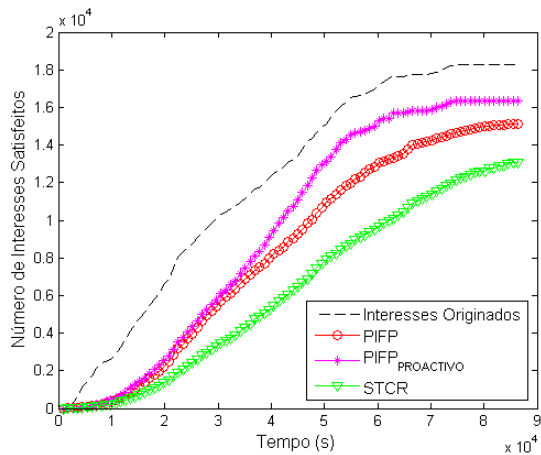
A estratégia PIFP inicialmente possui um comportamento idêntico ao STCR. A partir dos 40000 segundos a carga de dados diminui. O STCR mantém o número de dados presentes na rede.

O grande número de pacotes de dados na rede deve-se ao seu reduzido tamanho, permitindo que cada nó possa armazenar mais pacotes. Quanto maior for o número de pacotes na rede, maior é a possibilidade de obter conteúdos. Isto pode implicar problemas de encaminhamento na rede pois as previsibilidades calculadas pelo PIFP podem ser relativamente parecidas, o que irá criar múltiplos caminhos para chegar aos conteúdos. A versão proactiva do PIFP é a mais afetada.

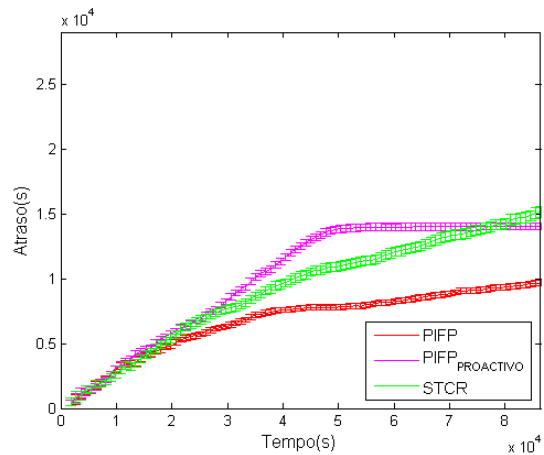
A Figura 42 mostra o número de pacotes de interesse na rede. Como é possível, visualizar o STCR possui um maior número de interesses.

No seguinte ensaio foram utilizadas mensagens cujo tamanho varia entre 250K a 350K.

A Figura 43 mostra o número de interesses satisfeitos nas diferentes estratégias.



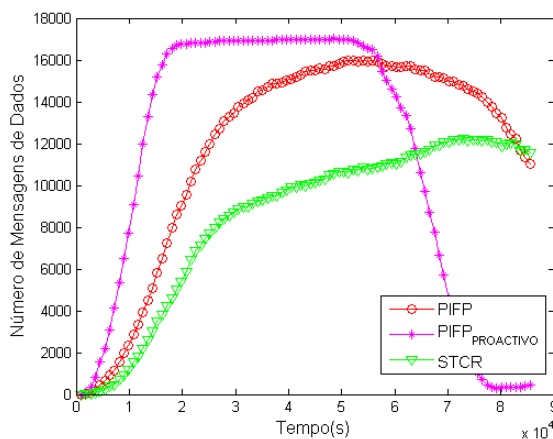
**Figura 43 - Interesses Satisfeitos (250-350k)**



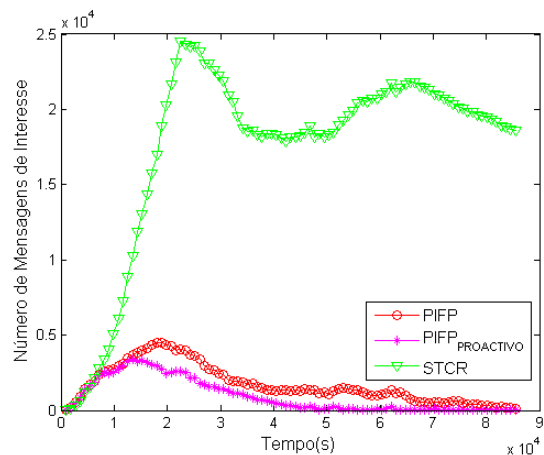
**Figura 44 - Atraso Médio (250-350k)**

A estratégia PIFPproactivo possui maior número de interesses satisfeitos, quando comparada com as outras estratégias. A sua taxa de satisfação é de aproximadamente 90%. As estratégias PIFP e STCR obtiveram cerca de 82% e 70% de satisfação de interesses, respetivamente.

Em relação ao atraso médio, este é maior nas estratégias STCR e PIFPproactivo, como se pode visualizar na Figura 44. Comparando com o ensaio anterior, o atraso médio aumentou em ambas as estratégias.



**Figura 45 – Carga de Dados (250-350k)**



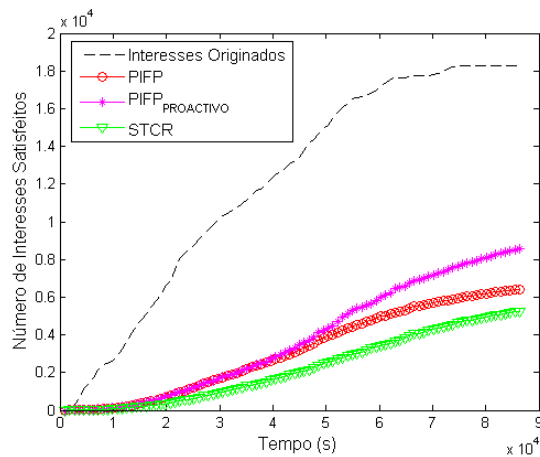
**Figura 46 – Carga de Interesses (250-350k)**

Como esperado, neste ensaio a estratégia PIFPproactivo possui um número de interesses superior. Os pacotes de dados são enviados proactivamente para a rede e são aceites desde que exista espaço de armazenamento livre. A estratégia STCR possui menor número de dados na rede.

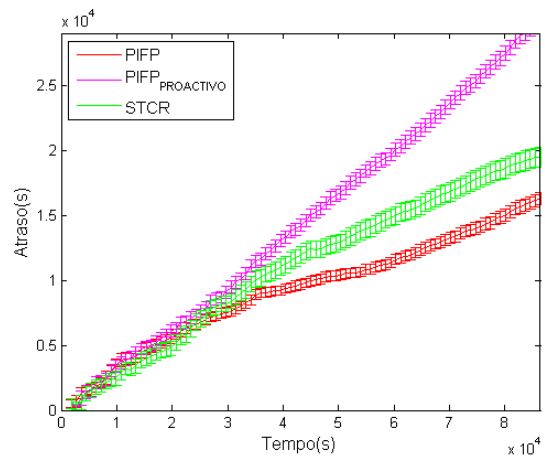
Comparado com o ensaio anterior (50K-150K), o número de mensagens na rede diminuiu. Isto deve-se às limitações do nó (armazenamento, taxas de transmissão).

O número de interesses aumentou relativamente ao ensaio superior. Isto deve-se ao fato da taxa de satisfação de interesses ser inferior.

Por fim, foi efetuado um último ensaio de modo a avaliar o impacto com mensagens cujo tamanho é compreendido entre os 750k-800k.



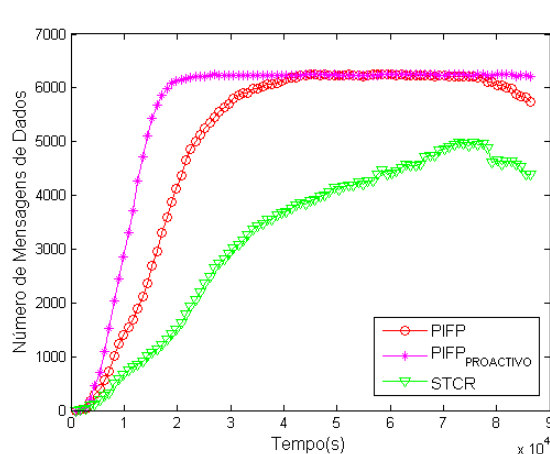
**Figura 47 – Interesses Satisfeitos (750-850k)**



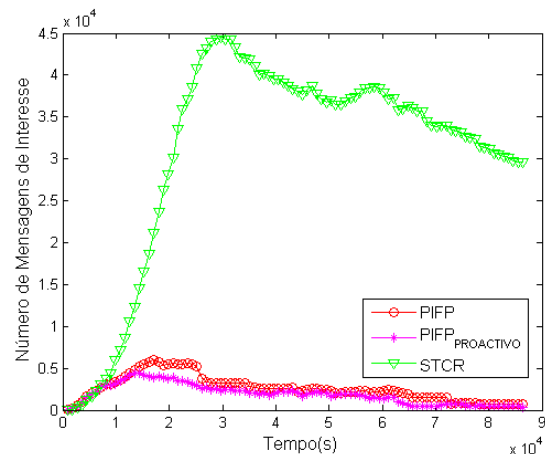
**Figura 48 – Atraso Médio (750-800k)**

Analisando o gráfico da Figura 47, é possível visualizar a baixa taxa de satisfação obtida por todos os protocolos. Em relação aos interesses originados apenas 45% destes foram satisfeitos pelo protocolo PIFPproactivo, 35% pelo protocolo PIFP e 28% pelo protocolo STCR.

Com o aumento do tamanho das mensagens o atraso médio também aumentou em todas as estratégias como mostra a Figura 48.

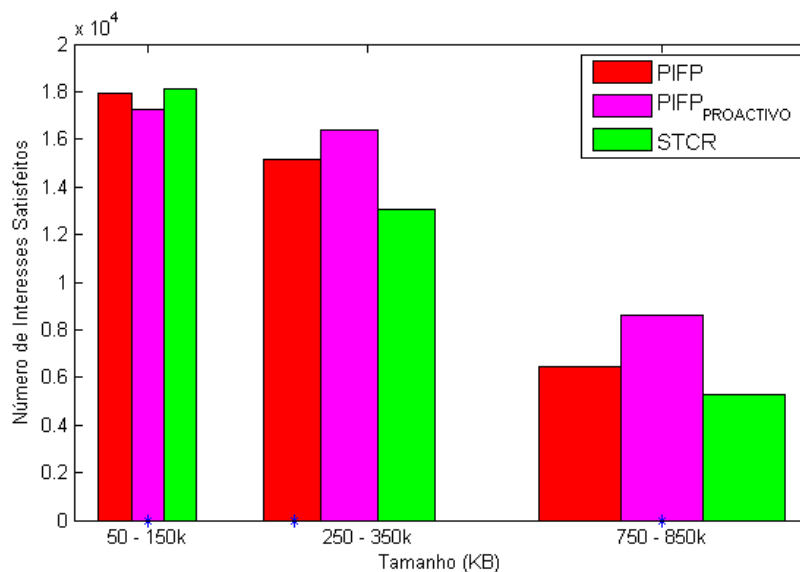


**Figura 49 – Carga de Dados (750-850k)**



**Figura 50 Carga de Interesses (750-850k)**

A Figura 49 mostra o número de pacotes de dados na rede. As estratégias PIFPproactivo e PIFP possuem maior número de pacotes de dados na rede. O tamanho das mensagens implica um consumo elevado de armazenamento. Quando não existe mais espaço disponível para armazenar mensagens no *buffer* as mensagens mais antigas são as primeiras a ser descartadas. Também através do TTL de cada pacote de dados são removidas as mensagens cujo tempo de vida expirou libertando espaço para novas mensagens.



**Figura 51 - Interesses Satisfeitos vs Tamanho das Mensagens**

O tamanho das mensagens provoca impacto na performance dos protocolos como é possível visualizar na Figura 51. No primeiro ensaio com mensagens de 50-150k, as estratégias PIFP e STCR obtêm uma taxa de satisfação similar. Com o aumento do tamanho das mensagens, a taxa de satisfação é inferior. O protocolo PIFP e a sua variante possuem um número de interesses satisfeitos superior ao protocolo STCR.

Através do gráfico apresentado na Figura 51, é possível concluir que com o aumento do tamanho das mensagens a taxa de satisfação é menor para todas as estratégias. O aumento do tamanho das mensagens tem um impacto negativo na comunicação e no armazenamento devido às limitações existentes neste tipo de rede. Com o armazenamento limitado não é possível armazenar todas as mensagens recebidas, reduzindo assim as oportunidades de encaminhamento. No paradigma de dados nomeados, o armazenamento das mensagens é importante pois este permite a reutilização destas para futuros interesses.

O grande tamanho das mensagens, que provoca atrasos de propagação e a capacidade do canal diminuem a taxa de transferência fazem com que o número de mensagens transmitidas seja reduzido.

### 7.3.3 Consumo de Energia

Nesta subsecção pretende-se analisar o consumo de energia. Para realizar este ensaio foi utilizado o cenário esparso (cenário 2).

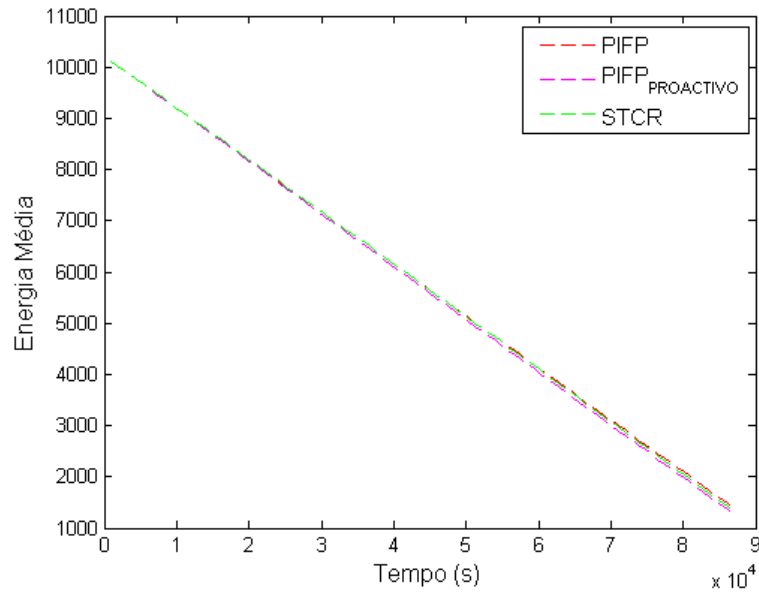
Inicialmente um nó possui um nível de energia que ao longo do tempo este nível diminui. O nível de energia diminui com as transmissões de mensagens, com a procura por nós na proximidade, as respostas durante uma conexão e também quando o nó está em adormecido.

**Tabela 9 - Configurações de Energia**

<b>Parâmetro</b>	<b>Valor</b>
Energia Inicial	Produtores: 15000 unidades Consumidores: 10000 unidades
Consumo de Energia por Pesquisa	0.1 unidades
Consumo de Energia por Transmissão	0.2 unidades
Consumo de energia por reposta de pesquisa	0.1 unidades
Consumo de energia base	0.01 unidades

A **Error! Reference source not found.** mostra os parâmetros necessários para a configuração da energia nos nós. Este valores foram atribuídos com base no artigo [46].





**Figura 52 - Energia Consumida**

A Figura 52 mostra a média do consumo de energia nas três estratégias. Como é possível visualizar, o consumo de energia é similar nas diferentes estratégias. A estratégia PIFP obtém um consumo de energia ligeiramente menor na parte final da simulação quando comparada com as outras estratégias.

**Tabela 10 - Resultados da Energia Consumida**

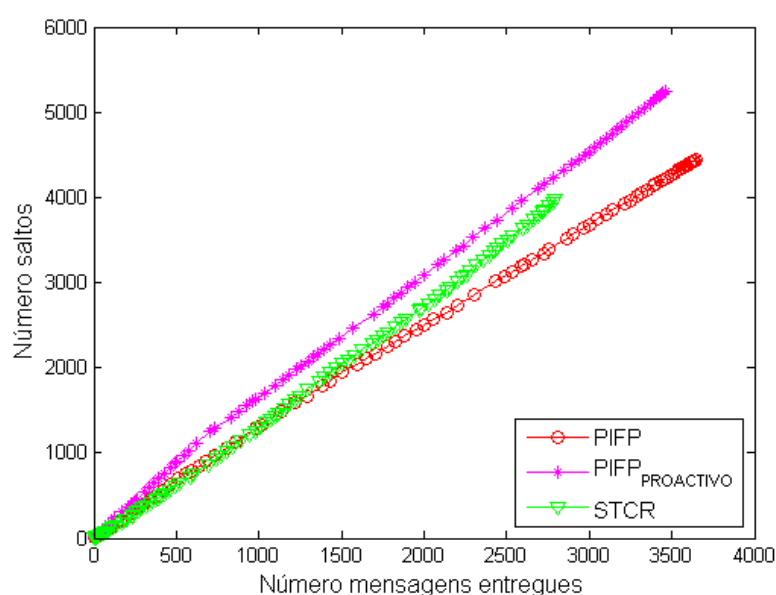
<b>Tempo (s)</b>	<b>PIFP</b>	<b>PIFPproactivo</b>	<b>STCR</b>
<b>Início</b>	9999	9999	9999
<b>21600 (6h)</b>	7826	7799	7831
<b>43200 (12h)</b>	5629	5575	5628
<b>64800 (18h)</b>	3439	3342	3417
<b>21600 (24h)</b>	1259	1138	1190

Na Tabela 10, são apresentados alguns instantes da simulação e a energia existente nesse momento. No final da simulação, PIFP é o protocolo que possui mais energia. No entanto, como se pode visualizar no gráfico estes valores são muito próximos.

### 7.3.4 Número de Saltos

Após a análise do consumo de energia é agora efetuada uma avaliação ao número de saltos. Esta métrica, corresponde ao número de nós menos um sobre o caminho a partir da fonte para o destino, isto é, o número de nós pelo qual a mensagem passou até ser entregue ao destinatário final.

Foi efetuada uma análise individual para os dois tipos de mensagens utilizadas. Em primeiro lugar, é apresentada a análise do número de saltos para a entrega de mensagens do tipo dados.



**Figura 53 - Número de Saltos (são consideradas apenas as mensagens de dados entregues)**

A Figura 53 mostra o somatório do número de saltos em função às mensagens de dados entregues ao consumidor final. O número de saltos é menor na estratégia PIFP quando comparada com as estratégias PIFPproactivo e STCR.

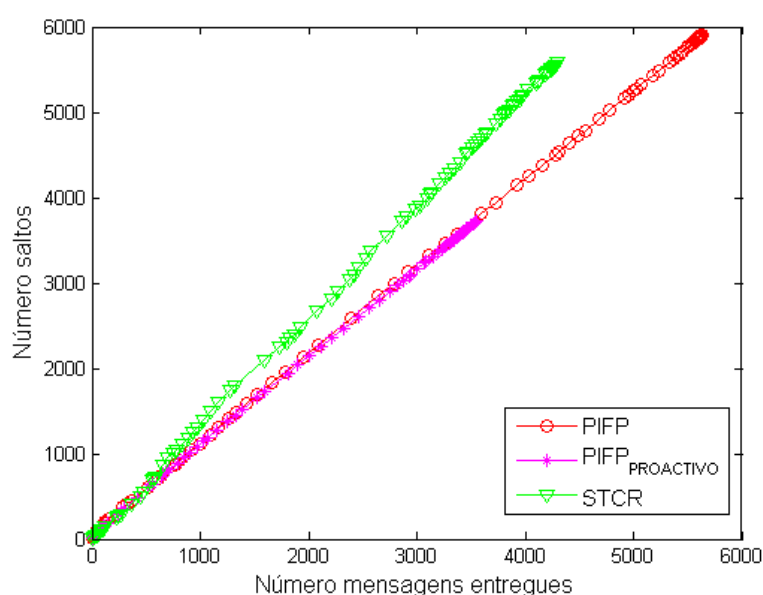
**Tabela 11 – Resumo do número de saltos no encaminhamento de pacotes de dados**

Protocolo	Média	Mediana	Máximo (Médio)	Mínimo
PIFP	1,21	1	6,5	1
PIFPproactivo	1,54	1	7	1
STCR	1,42	1	6	1

A Tabela 11 mostra o resumo de resultados relativamente ao número de saltos no encaminhamento de pacotes de dados, onde é apresentada a média, mediana, máximos e mínimos. A média de saltos é menor no protocolo PIFP. Os pacotes de dados no protocolo PIFP seguem o caminho inverso, isto é, seguem as “migalhas” deixadas pelos pacotes de interesse enquanto a estratégia STCR encaminha as mensagens de dados utilizando a relação social com os nós. O menor número de saltos no PIFP também deve-se ao maior número de caches distribuídas na rede.

A estratégia proactiva do protocolo PIFP possui maior número de saltos devido ao encaminhamento proactivo de dados. Isto faz com que o canal esteja ocupado com um conjunto de mensagens de dados, aumentando a concorrência de acesso ao canal.

A estratégia de encaminhamento de interesses utilizada no PIFP, baseada na previsibilidade de contacto com os conteúdos, mostra melhor resultados que o STCR. Como descrito, o protocolo utiliza as probabilidades dos nós se encontrarem com a informação. O gráfico da Figura 54 apresenta o número de saltos em relação ao número de mensagens de interesse entregues.



**Figura 54 - Número de Saltos (são consideradas apenas as mensagens de interesse entregues)**

Como é possível visualizar a estratégia PIFPproactivo possui um comportamento semelhante ao PIFP. Isto deve-se a ambas as estratégias utilizarem a mesma lógica de encaminhamento de mensagens de interesse. A estratégia PIFP detém um número menor de

saltos em relação ao STCR. A estratégia de encaminhamento de interesses utilizada pelo protocolo PIFP mostra-se mais eficiente tanto na entrega de interesses como no número saltos.

Como é possível visualizar na Tabela 12 o número médio de saltos é menor para o protocolo PIFP e a sua variante quando comparado com o protocolo STCR. O valor médio máximo de saltos é semelhante em ambas as estratégias.

**Tabela 12 - Resumo do número de saltos no encaminhamento de pacotes de dados**

<b>Protocolo</b>	<b>Média</b>	<b>Mediana</b>	<b>Máximo (Médio)</b>	<b>Mínimo</b>
PIFP	1,1	1	5	1
PIFPproactivo	1,1	1	5	1
STCR	1,32	1	5,5	1

Através destes resultados, é possível concluir que a estratégia de encaminhamento do PIFP obtém uma melhor performance. O seu encaminhamento baseado nas probabilidades de chegar a um conteúdo é mais eficiente do que a estratégia utilizada no STCR que é baseado nas relações entre os nós e não na informação.

### **7.3.5 Discussão dos resultados obtidos**

O protocolo de encaminhamento proposto foi avaliado e comparado com o protocolo STCR. Os resultados obtidos nas simulações mostram que o protocolo PIFP supera o protocolo STCR em algumas métricas.

Para cenários densos, a versão proactiva do PIFP mostrou possuir maior taxa de satisfação de interesses quando comparado como os outros protocolos. Ainda assim, o PIFP obtém melhores resultados do que o STCR nesta métrica. Em cenários esparsos, o desempenho dos protocolos é inferior do que em cenários densos. Os protocolos analisados neste cenário possuem resultados semelhantes. A densidade da rede afeta o desempenho dos protocolos como mostrado nos resultados. Quanto mais nós existirem na rede, maior é a probabilidade de obter dados pois existem mais caminhos disponíveis e também mais caches. A persistência dos pacotes de dados permite sejam reutilizados pelo próprio nó ou para satisfazer interesses de outros nós.

A carga de interesses em todos os ensaios realizados é maior no protocolo STCR. Nos protocolos PIFP e PIFP proactivo a carga de interesses é consideravelmente inferior. A

previsibilidade de chegar a um conteúdo faz com que o número de interesses enviados para a rede seja reduzido. Em relação à carga de dados na rede, o STCR possui melhores resultados que o protocolo PIFP e a sua versão proactiva.

Outra métrica avaliada foi o atraso médio de satisfação dos interesses, onde o protocolo o protocolo PIFPproactivo detém maior atraso. A propagação de pacotes de dados limita o armazenamento de pacotes subsequentes. Isto faz com que a versão proactiva do PIFP demore mais tempo do que o protocolo PIFP.

Nos ensaios realizados, foi possível perceber que o aumento do tamanho das mensagens produz um impacto negativo no desempenho de todos os protocolos simulados. Com tamanhos entre os 50k e 150k a taxa de satisfação é próxima de 100% em ambas as estratégias. Com o aumento do tamanho, o número de transmissões é menor. Isto deve-se ao tempo de propagação ser maior que afeta o tempo de contacto. Este não é grande o suficiente para transmitir uma mensagem.

O número de saltos é ligeiramente superior na estratégia STCR para o envio de mensagens de interesse. No envio de mensagens de dados os resultados são semelhantes em ambas as estratégias. O consumo de energia é praticamente igual para ambos os protocolos.

Através Tabela 13 é possível compreender de forma mais sucinta os resultados que foram descritos durante este capítulo.

			<b>PIFP</b>	<b>PIFPproactivo</b>	<b>STCR</b>
<b>Impacto da Rede</b>	Cenário 1	Interesses Satisfeitos	65%	82%	50%
		Atraso	Menor	Médio	Alto
		Carga	Menor número de interesses	Possui maior carga nos dados	Menor Número de dados
	Cenário 2	Interesses Satisfeitos	42%	43%	44%
		Atraso	Menor	Médio	Alto
		Carga	Menor número de interesses	Possui maior carga nos dados	Menor Número de dados

		<b>PIFP</b>	<b>PIFPproactivo</b>	<b>STCR</b>	
<b>Impacto do tamanho</b>	50-150k	Interesses Satisfeitos	≈ 100%	≈ 100%	≈ 100%
		Atraso	Menor	Médio	Alto
		Carga	Carga de dados semelhantes ao STCR	Possui maior carga nos dados	Maior número de interesses
	250-350k	Interesses Satisfeitos	≈ 82%	≈ 90%	≈ 70%
		Atraso	Menor	Médio	Alto
		Carga	Menor número de interesses	Possui maior carga nos dados	Menor Número de dados
	750-850k	Interesses Satisfeitos	≈ 35%	≈ 45%	≈ 28%
		Atraso	Menor	Médio	Alto
		Carga	Carga de dados semelhantes à versão proactiva	Carga de dados alta	Menor número de dados e maior número de interesses
<b>Energia Consumida</b>			Semelhante	Semelhante	Semelhante
<b>Número de Saltos</b>	Pacotes de interesse	Média de 1,21	Média de 1,54	Média de 1,42	
	Pacotes de dados	Média de 1,1	Média de 1,1	Média de 1,32	

**Tabela 13 - Síntese dos Resultados Obtidos**



# Capítulo 8

## Conclusões e Trabalho Futuro

O principal desafio deste trabalho de dissertação visava a integração das arquiteturas DTN e NDN, de modo a avaliar em que medida as NDNs se adequam em cenários urbanos com DTNs.

As NDNs possuem suporte nativo à mobilidade característica das redes sem-fios. Foca-se nos dados e não na localização dos nós, facilitando a compreensão destes e permitindo o armazenamento e a reutilização. As DTNs foram projetadas para permitir comunicações em cenários móveis sem fios onde a conectividade fim-a-fim não é possível. Todavia, as DTNs lidam com a incerteza das mensagens serão reenviadas, ou descartadas silenciosamente devido ao limite dos seus recursos ou por razões de egoísmo. O paradigma de dados nomeados permite a compreensão dos dados, esta incerteza pode ser mitigada.

Neste trabalho, foi proposto um novo protocolo de encaminhamento chamado PIFP, focado nos dados nomeados para as DTNs. O PIFP é baseado no PROPHET, um protocolo standard das DTNs. Tal como o PROPHET, este protocolo explora a frequência dos contactos oportunistas, não entre nós, mas entre nós mas entre os conteúdos. O objetivo é calcular a previsibilidade de entrega de interesses e a recuperação de dados em cenários como as DTNs.

Foi também introduzida uma variante do protocolo PIFP. A principal diferença é que os pacotes de dados são enviados para todos os nós vizinhos com espaço em cache. O objetivo pretendido é chegar rapidamente com os dados ao consumidor. Esta abordagem explora fundamentalmente o canal de comunicação e o espaço de armazenamento.

Outro objetivo desta dissertação consistiu na implementação de uma plataforma de simulação (ICONE) que permite o desenvolvimento de novos protocolos com base em dados nomeados para as redes tolerantes a atrasos. O ICONE integra as NDNs no simulador ONE, um simulador para as DTNs. O ONE foi estendido de forma a suportar o paradigma de dados nomeados.

No Capítulo 7, foi avaliado o desempenho do protocolo PIFP e comparado com o protocolo STCR, um mecanismo de encaminhamento NDN para as redes MANETs. Foram utilizados para



esta comparação parâmetros relacionados com as características das rede (número de nós envolvidos) e o tamanho das mensagens para verificar o impacto no desempenho dos protocolos de encaminhamento.

Os ensaios realizados mostram o PIFP traz benefícios notáveis sobre as DTNs, principalmente em termos da taxa de satisfação de interesses, carga de interesses, o número de saltos e o atraso médio.

Para os cenários utilizados, os resultados experimentais mostram que o protocolo PIFP obtém melhores resultados em redes densas dado que existem mais caminhos alternativos para chegar aos conteúdos. Também com o aumento do número de nós e a persistência dos pacotes de dados, existem mais caches distribuídas na rede. Um nó consumidor pode tornar-se um nó produtor, aumentando desta forma a probabilidade de obter um conteúdo. A versão proactiva do PIFP, apesar da sua eficiência na entrega de pacotes de dados, possui custos elevados de armazenamento, sendo necessário mais espaço de armazenamento.

A estratégia adotada para o encaminhamento de interesses no PIFP, permite a redução do número de saltos e o atraso médio. Esta estratégia é independente da localização dos conteúdos e por isso são apenas selecionados os melhores nós para chegar a um conteúdo.

Também foi avaliado o impacto do tamanho das mensagens, onde o comportamento foi semelhante nos diferentes protocolos. Com o aumento das mensagens, o desempenho dos protocolos é afetado. Isto deve-se ao escasso tempo de conexão e ao tamanho limitado de armazenamento que os nós possuem.

Os resultados revelaram que o PIFP é um protocolo de encaminhamento efetivo, permitindo ultrapassar as dificuldades de comunicação características nas redes sem fios num ambiente urbano. Estes fortalecem a afirmação de suporte nativo à mobilidade no paradigma de dados nomeados. A mudança de paradigma traz vantagens para a recuperação de conteúdos pois, através da nomeação dos pacotes de dados é possível reutilizar os pacotes recebidos e manter a sua persistência.

Na solução apresentada os interesses são encaminhados para os produtores (ou seus representantes) com base na previsibilidade que os nós têm em encontrá-los. Essa previsibilidade é calculada com base nos contactos com os produtores para um determinado conteúdo e na transitividade de nós que tenham contactado esses produtores.

No entanto, no encaminhamento dos dados em resposta aos interesses, utiliza-se apenas o facto do nó recetor ter o interesse correspondente registado. Poder-se-ia usar também a

probabilidade de encontrar interessados para limitar o número de cópias dos conteúdos a colocar na rede.

Outro ponto interessante seria a conceção de um sistema de nomeação e divulgação de nomes de dados em cenários como as DTNs. A nomeação implica o acordo entre o publicador e o interessado.

Outros objeto de estudo seria a fragmentação dos pacotes de dados nomeados. O grande tamanho das mensagens e a capacidade do canal por vezes provoca atrasos de propagação. O objetivo seria encontrar uma maneira de ajustar a fragmentação de acordo com os tempos de contacto.

# Bibliografia

- [1] S. Burleigh, A. Hooke, L. Torgerson, B. Durst, K. Scott, and T. M. Corporation, “Delay-Tolerant Networking : An Approach to Interplanetary Internet,” *IEEE Commnications Mag.*, pp. 128–136, 2003.
- [2] D. T. Britt, C. Matthews, L. Parziale, C. Davis, J. Forrester, W. Liu, and N. Rosselot, *TCP / IP Tutorial and Technical Overview*, 8th ed. 2006.
- [3] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. Thornton, K. Diana, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos, P. Crowley, J. D. Thornton, and D. K. Smetters, “Named Data Networking ( NDN ) Project ( NDN ) Project,” *PARC Tech Rep.*, 2010.
- [4] D. Rodrigues-Silva, A. Costa, and J. Macedo, “Energy Impact Analysis on DTN Routing Protocols,” *ExtremeCom*, 2012.
- [5] T. Coelho, A. Costa, and C. Algoritmi, “Encaminhamento com QoS para Redes Ad Hoc com rotas estáveis,” *13ª Conferência sobre Redes Comput.*, 2013.
- [6] J. M. Timóteo, A. Costa, and J. Macedo, “Recolha e Análise de Dados de Contactos Físicos e Sociais numa Rede Tolerante a Atrasos,” 2014.
- [7] L. Philippe, A. Costa, and J. Macedo, “Dados Nomeados em Redes Tolerantes a Atrasos,” 2014.
- [8] A. Keränen, J. Ott, and T. Kärkkäinen, “The ONE simulator for DTN protocol evaluation,” *Proc. Second Int. ICST Conf. Simul. Tools Tech.*, 2009.
- [9] P. Duarte, A. Costa, J. Macedo, and A. Santos, “A Routing Protocol for Named Data Delay Tolerant Networks,” *Submetido numa conferência Int. da área*, 2014.
- [10] M. Gerla, *Ad Hoc Networks Technologies and Protocols*. Springer Science + Business Media, Inc, 2005, pp. 1 – 22.

- [11] F. Warthman, "Delay- and Disruption-Tolerant Networks (DTNs): A Tutorial," *Interplanet. Netw. Spec. Interes. Gr.*, vol. 2.0, 2012.
- [12] K. Scott and S. Burleigh, "Bundle Protocol Specification: RFC 5050," 2007.
- [13] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, "Disruption-Tolerant Networking: A Comprehensive Survey on Recent Developments and Persisting Challenges," *IEEE Commun. Surv. Tutorials*, vol. 14, no. 2, pp. 607–640, 2012.
- [14] K. Fall and S. Farrell, "DTN : An Architectural Retrospective," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 5, pp. 828–836, 2008.
- [15] C. T. De Oliveira, M. D. D. Moreira, M. G. Rubinstein, L. Henrique, M. K. Costa, and O. Carlos, "Redes Tolerantes a Atrasos e Desconexões," *Simpósio Bras. Redes Comput. (SBRC), Soc. Bras. Comput.*, pp. 203–256, 2007.
- [16] S. Wang and R. Ma, "Name : A Naming Mecanism for Delay/Disruption - Tolerant Network," *Int. J. Comput. Networks Commun.*, vol. 5, no. 6, pp. 231–241, 2013.
- [17] S. Wang, X. Cheng, and M. Song, "Routing in Pocket Switched Networks," *IEEE Wirel. Commun.*, no. February, pp. 67–73, 2012.
- [18] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks," *Dep. Comput. Sci. Duke Univ.*, 2000.
- [19] C. S. Raghavendra, T. Spuopoulos, and K. Psounis, "Spray and Wait : An Efficient Routing Scheme for Intermittently Connected Mobile Networks," *SIGCOMM'05 Work. Philadelphia*, 2005.
- [20] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 3, p. 19, Jul. 2003.
- [21] M. Grossglauser and D. N. C. Tse, "Mobility Increases the Capacity of Ad Hoc Wireless Networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 477–486, 2002.

- [22] T. Koponen, K. H. Kim, S. Shenker, M. Chawla, B.-G. Chun, A. Ermolinskiy, and I. Stoica, "A Data-Oriented ( and Beyond ) Network Architecture," *SIGCOMM'07*, 2007.
- [23] C. (University of P. Dannewitz, "NetInf : An Information-Centric Design for the Future Internet," *Comput. Commun.* 36, 2013.
- [24] N. Fotiou, P. Nikander, D. Trossen, and G. C. Polyzos, "Developing Information Networking Further : From PSIRP to PURSUIT," *Int. ICST Conf. Broadband Commun. Networks, Syst.*, pp. 1–13, 2010.
- [25] J. Pan, S. Paul, and R. Jain, "D EPLOYMENT P ERSPECTIVES A Survey of the Research on Future Internet Architectures," no. July, pp. 26–36, 2011.
- [26] V. Jacobson, J. Burke, D. Estrin, L. Zhang, G. Tsudik, K. Claffy, D. Krioukov, D. Massey, C. Papadopoulos, K. Shilton, L. Wang, E. Yeh, and P. Crowley, "Named Data Networking ( NDN ) Project Principal Investigators," 2013.
- [27] V. Jacobson, D. K. Smetters, N. H. Briggs, J. D. Thornton, M. F. Plass, and R. L. Braynard, "Categories and Subject Descriptors," *CoNEXT*, pp. 1–12, 2009.
- [28] T. Lauinger, N. Laoutaris, P. Rodriguez, T. Strufe, and E. Biersack, "Privacy Risks in Named Data Networking : What is the Cost of Performance ?," *SIGCOMM Comput. Communications*, 2012.
- [29] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A Case for Stateful Forwarding Plane," 2012.
- [30] A. Alyyan, S. O. Amim, A. K. M. M. Hoque, B. Zhang, L. Zhang, and L. Wang, "NLSR : Named-data Link State Routing Protocol," in *ACM SIGCOMM ICN Workshop*, 2013, pp. 2–7.
- [31] PARC, "PARC. CCNx open source plataform," 2013. [Online]. Available: <http://www.ccnx.org>.

- [32] M. Tortelli, L. A. Grieco, G. Boggia, K. Pentikousis, and P. Bari, "COBRA : Lean Intra-domain Routing in NDN," in *IEEE Consumer Communications and Networking Conference, CCNC, Special Session on Research and Case Study for Designing and Deploying Information-centric Networks*, 2014.
- [33] H. Song, S. Dharmapurikar, J. Turner, and J. Lockwood, "Fast Hash Table Lookup Using Extended Bloom Filter : An Aid to Network Processing," *SIGCOMM'05 Work. Philadelphia*, pp. 181–192.
- [34] K. Fall, "Comparing Information Centric and Delay Tolerant Networking," in *The 39th IEEE Conference on Local Computer Networks*, 2012, pp. 1–26.
- [35] M. Meisel, V. Pappas, and L. Zhang, "Ad hoc networking via named data," *Proc. fifth ACM Int. Work. Mobil. Evol. internet Archit. - MobiArch '10*, p. 3, 2010.
- [36] G. Tyson, J. Bigham, and E. Bodanese, "Towards an Information-Centric Delay-Tolerant Network," in *2nd IEEE NOMEN*, 2013.
- [37] M. R. Meisel, "BOND : Unifying Mobile Networks with Named Data by," California Los Angeles, 2011.
- [38] Y.-T. Yu, R. B. Dilmaghani, S. Calo, M. Y. Sanadidi, and M. Gerla, "Interest propagation in named data manets," *2013 Int. Conf. Comput. Netw. Commun.*, pp. 1118–1122, Jan. 2013.
- [39] Y. Lu, X. Li, Y. Yu, and M. Gerla, "Information-Centric Delay-Tolerant Mobile Ad-Hoc Networks," *INFOCOM 2014, Work. Name Oriented Mobility, Toronto, Canada, April 2014*.
- [40] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.
- [41] A. Broder and M. Mitzenmacher, "Network Applications of Bloom Filters : A Survey," *Internet Math.*, vol. 1, no. 4, pp. 485–509, 2003.
- [42] R. Rivest, "The MD5 Message-Digest Algorithm, RFC 1321," 1992.

- [43] D. Eastlake and P. Jones, "US Secure Hash Algorithm 1 (SHA1), RFC 3174," 2001.
- [44] F. Schneider, S. Agarwal, T. Alpcan, and A. Feldmann, "The New Web : Characterizing AJAX Traffic," in *Passive and Active Network Measurement*, Berlin: Springer-Verlag Berlin Heidelberg, 2008, pp. 31–40.
- [45] K. Krishna and M. N. Murty, "Genetic K-Means Algorithm," *IEEE Trans. Syst. Man. Cybern.*, vol. 29, no. 3, pp. 433–439, 1999.
- [46] S. K. Dhurandher, D. Sharma, and I. Woungang, "Energy-based Performance Evaluation of Various Routing Protocols in Infrastructure-less Opportunistic Networks," *J. Internet Serv. Inf. Secur.*, vol. 3, pp. 37–48.
- [47] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive forwarding in named data networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 3, p. 62, Jun. 2012.

# Anexos

## Lista de Configurações de Simulação

```
#
# Default settings for the simulation
#

## Scenario settingsMAX
Scenario.name = default_scenario
Scenario.simulateConnections = true
Scenario.updateInterval = 0.1
# 21600s == 6h
# 43200s == 12h
# 86400s == 1d
# 172800s == 2d
# 259200s == 3d
# 345600s == 4d
Scenario.endTime = 86400

# Define 6 different node groups
Scenario.nrofHostGroups = 5

## Interface-specific settings:
# type : which interface class the interface belongs to
# For different types, the sub-parameters are interface-specific
# For SimpleBroadcastInterface, the parameters are:
# transmitSpeed : transmit speed of the interface (bytes per second)
# transmitRange : range of the interface (meters)

# "Bluetooth" interface for all nodes
btInterface.type = SimpleBroadcastInterface
# Transmit speed of 2 Mbps = 250kBps
btInterface.transmitSpeed = 250k
btInterface.transmitRange = 10

# High speed, long range, interface for group 4
highspeedInterface.type = SimpleBroadcastInterface
highspeedInterface.transmitSpeed = 10M
highspeedInterface.transmitRange = 1000

## Group-specific settings:
# groupID : Group's identifier. Used as the prefix of host names
# nrofHosts: number of hosts in the group
# movementModel: movement model of the hosts (valid class name from movement
package)
# waitTime: minimum and maximum wait times (seconds) after reaching
destination
# speed: minimum and maximum speeds (m/s) when moving on a path
# bufferSize: size of the message buffer (bytes)
# router: router used to route messages (valid class name from routing
package)
# activeTimes: Time intervals when the nodes in the group are active (start1,
end1, start2, end2, ...)
```



```

# msgTtl : TTL (minutes) of the messages created by this host group,
default=infinite

## Group and movement model specific settings
# pois: Points Of Interest indexes and probabilities (poiIndex1, poiProb1,
poiIndex2, poiProb2, ... )
#       for ShortestPathMapBasedMovement
# okMaps : which map nodes are OK for the group (map file indexes),
default=all
#       for all MapBasedMovent models
# routeFile: route's file path - for MapRouteMovement
# routeType: route's type - for MapRouteMovement

# name of the file that contains the list of contents
NamedDataNetworkRouter.content = lista_de_conteudos.txt
# number of hosts
NamedDataNetworkRouter.nrofHosts = 98
NamedDataNetworkRouter.secondsInTimeUnit = 30

# Common settings for all groups
Group.movementModel = ShortestPathMapBasedMovement
Group.router = PIFP
#Group.router = PIFP_Proactive
#Group.router = sctr
Group.bufferSize = 50M
Group.waitTime = 0, 120
# All nodes have the bluetooth interface
Group.nrofInterfaces = 1
Group.interface1 = btInterface
# Walking speeds
Group.speed = 0.5, 1.5
# Message TTL of 500 minutes
Group.msgTtl = 500
Group.initialEnergy = 10000
Group.scanEnergy = 0.1
Group.transmitEnergy = 0.2
Group.scanResponseEnergy = 0.1
Group.baseEnergy = 0.01
EnergyLevelReport.granularity = 1.0
Group.nrofHosts = 32

# group1 (pedestrians) specific settings
Group1.groupID = p

# group2 specific settings
Group2.groupID = c
# cars can drive only on roads
Group2.okMaps = 1
# 10-50 km/h
Group2.speed = 2.7, 13.9

# another group of pedestrians
Group3.groupID = w
Group3.nrofHosts = 30

# The Tram groups
Group4.groupID = t
Group4.bufferSize = 100M

```

```

Group4.movementModel = MapRouteMovement
Group4.routeFile = data/tram3.wkt
Group4.routeType = 1
Group4.waitTime = 10, 30
Group4.speed = 7, 10
Group4.nrofHosts = 2
Group4.nrofInterfaces = 2
Group4.interface1 = btInterface
Group4.interface2 = highspeedInterface
Group4.initialEnergy = 15000

Group5.groupID = t
Group5.bufferSize = 100M
Group5.movementModel = MapRouteMovement
Group5.routeFile = data/tram4.wkt
Group5.routeType = 2
Group5.waitTime = 10, 30
Group5.speed = 7, 10
Group5.nrofHosts = 2
Group5.initialEnergy = 15000

## Message creation parameters
# How many event generators
Events.nrof = 1
Events1.class = ExternalEventsQueue
Events1.filePath = messageevents.txt

## Movement model settings
# seed for movement models' pseudo random number generator (default = 0)
MovementModel.rngSeed = 1
# World's size for Movement Models without implicit size (width, height;
meters)
MovementModel.worldSize = 4500, 3400
# How long time to move hosts in the world before real simulation
MovementModel.warmup = 1000

## Map based movement -movement model specific settings
MapBasedMovement.nrofMapFiles = 4

MapBasedMovement.mapFile1 = data/roads.wkt
MapBasedMovement.mapFile2 = data/main_roads.wkt
MapBasedMovement.mapFile3 = data/pedestrian_paths.wkt
MapBasedMovement.mapFile4 = data/shops.wkt

## Reports - all report names have to be valid report classes

# how many reports to load
Report.nrofReports = 3
# length of the warm up period (simulated seconds)
Report.warmup = 0
# default directory of reports (can be overridden per Report with output
setting)
Report.reportDir = reports/

```

```
# Report classes to load
Report.report1 = NamedDataReport
Report.report2 = CreatedMessagesReport
Report.report3 = EnergyLevelReport

## Optimization settings -- these affect the speed of the simulation
## see World class for details.
Optimization.cellSizeMult = 5
Optimization.randomizeUpdateOrder = true

## GUI settings

# GUI underlay image settings
GUI.UnderlayImage.fileName = data/helsinki_underlay.png
# Image offset in pixels (x, y)
GUI.UnderlayImage.offset = 64, 20
# Scaling factor for the image
GUI.UnderlayImage.scale = 4.75
# Image rotation (radians)
GUI.UnderlayImage.rotate = -0.015

# how many events to show in the log panel (default = 30)
GUI.EventLogPanel.nrofEvents = 100
# Regular Expression log filter (see Pattern-class from the Java API for RE-
matching details)
#GUI.EventLogPanel.REfilter = .*p[1-9]<->p[1-9]$
```