UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÉNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

# DIRESC: UM PROTOCOLO PARA DESCOBERTA E RECUPERAÇÃO DE DADOS EM REDES CENTRADAS EM CONTEÚDO E TOLERANTES A ATRASO

Cláudio Diego Teixeira de Souza

**Orientador**

Carlos Alberto Vieira Campos

RIO DE JANEIRO, RJ - BRASIL
AGOSTO DE 2017

DIRESC: Um protocolo para descoberta e recuperação de dados em redes centradas em conteúdo e tolerantes a atraso

Cláudio Diego Teixeira de Souza

DISSERTAÇÃO APRESENTADA COMO REQUISITO PARCIAL PARA OBTENÇÃO DO TÍTULO DE MESTRE PELO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA DA UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO (UNIRIO). APROVADA PELA COMISSÃO EXAMINADORA ABAIXO ASSINADA.

Aprovada por:

Carlos Alberto Vieira Campos, D.Sc. - UNIRIO

Sidney Cunha de Lucena, D.Sc. - UNIRIO

Rosa Maria Meri Leão, Dr. - UFRJ

Kleber Vieira Cardoso, D.Sc. - UFG

RIO DE JANEIRO, RJ - BRASIL
AGOSTO de 2017

SOUZA, Cláudio Diego Teixeira de.
S719 DIRESC: Um protocolo para descoberta e recuperação de dados em redes centradas em conteúdo e tolerantes a atraso / Cláudio Diego Teixeira de Souza. – Rio de Janeiro, 2017.
100.

Orientador: Carlos Alberto Vieira Campos
Dissertação (Mestrado) - Universidade Federal do Estado do Rio de Janeiro, Programa de Pós-Graduação em Informática, 2017.

1. Redes tolerantes a atraso 2. Redes centradas em conteúdo 3. DTN 4. CCN 5. DIRESC I. Campos, Carlos Alberto Vieira, orient. II. Título.

*Minhas avós sempre foram mulheres de luta constante e incansável. Mesmo diante de dificuldades, batalharam para dar aos meus pais tudo que não tiveram. A essas guerreiras, eu dedico este trabalho, por me ensinarem o que é perseverança e serem exemplos grandiosos de determinação.*

# Agradecimentos

Agradeço primeiramente a Deus, que me deu a oportunidade de tomar decisões dificeis e crescer por meio delas. A Ele, que me ajudou a dar continuidade na vida acadêmica e a trilhar esse caminho apesar dos medos, o meu profundo agradecimento.

Aos meus pais, Cláudia e Fernando, que, desde o término do meu primeiro cíclo acadêmico (a graduação), foram os maiores incentivadores do início de uma nova jornada em minha vida e não mediram esforços para que isto se tornasse possível. Além deles, às minhas avós Elvira e Benedita e à minha irmã Bruna, que talvez tenham abafado seus receios ao me assistir partir de nossa cidade, mas que não pestanejaram em demonstrar o seu apoio, toda a minha gratidão.

À minha tia (postiça) Val e família, que me acolheram mais de uma vez de braços abertos, na minha chegada ao Rio de Janeiro e em momentos de sufoco, todo o meu carinho e agradecimento profundos.

Aos amigos da minha cidade natal que, mesmo pela distância física, se mantêm presentes em minha trajetória, compartilhando alegrias, dificuldades, superações e conquistas. Obrigado por terem sido capazes de compreender que precisei desbravar novos ares para desafiar e incentivar o meu próprio crescimento. Aos eternos amigos, Jessica, Luiz, Taty, Mariana, Kassia, Marjorie, Paoola, Carol, Ícaro, Stéphanie.

Por compartilhar momentos de alegria e tribulações, agradeço ao Phillippe por ter estado comigo em um ano tão significativo que estreitou nossa amizade. Agradeço também aos novos amigos, Vinícius e Lívia, com quem dividi muitos risos e a certeza de que estariam sempre dispostos a prestar qualquer auxílio quando necessário.

Faltam-me palavras suficientes para agradecer ao Igor por estar, desde que nos conhecemos, sempre ao meu lado, ajudando-me e dando incentivo, sendo mais do que um companheiro, sendo um grande e fiel amigo; alguém com quem compartilho todas as

dificuldades, expectativas e alegrias; talvez a pessoa mais especial que eu poderia ter conhecido nessa cidade. Muito obrigado.

Agradeço também ao professor Carlos Alberto Campos, que, com a sua experiência acadêmica, estimulou-me a pensar de forma diferente e apurada e lapidou meus conhecimentos. Além disso, por ter abraçado o meu projeto e tê-lo ajudado a crescer. Meus agradecimentos aos amigos do PPGI, Danielle, Lilian e Rodrigo pelas conversas que, quando não serviam para compartilhar conhecimentos, certamente garantiriam que ficassemos menos estressados. Por fim, apresento meus agradecimentos a CAPES, pelo apoio financeiro, sem o qual não me teria sido possível desenvolver esta pesquisa, e ao PPGI-UNIRIO por tê-la abrigado.

SOUZA, Cláudio Diego Teixeira de. **DIRESC: Um protocolo para descoberta e recuperação de dados em redes centradas em conteúdo e tolerantes a atraso**. UNIRIO, 2017. 100 páginas. Dissertação de Mestrado. Departamento de Informática Aplicada, UNIRIO.

## RESUMO

Redes centradas em conteúdo têm emergido como um paradigma inovador que modifica o modelo de comunicação tradicional, centrada em hospedeiro e endereçamento, para um modelo baseado em nomes de conteúdo. Quando combinado com redes tolerantes a atraso e interrupções, pode suprir necessidades que fazem deste um paradigma desafiador, inclusive para a obtenção de dados. No entanto, esta combinação também levanta novos desafios, como conciliar o esquema de propagação de mensagens em *broadcast*, natural de redes centradas em conteúdo, com a necessidade de reduzir as replicações de pacotes e o desgaste de recursos em redes tolerantes a atraso; ou ainda lidar com a ruptura de conexões e a formação de novos caminhos para buscar e recuperar dados. Assim, justificamos a viabilidade de tal combinação e apresentamos o protocolo DIRESC, uma proposta de solução para a seleção de retransmissores de mensagens para o anúncio, descoberta e recuperação de dados por meio de um esquema de cooperação social entre os nós, cujos objetivos abrangem a elevação da taxa de satisfação de requisições, a redução do atraso de obtenção de dados e a justa distribuição de carga na rede, com menor desgaste de recursos. Implementamos a arquitetura CCN e o DIRESC no simulador ONE (*Opportunistic Network Environment*) e comparamos o desempenho deste com outros três protocolos: o *CCNbroadcasting*, que representa o comportamento padrão de CCN; o STCR, um protocolo também baseado em relações sociais; e o PIFP, um protocolo probabilístico. Os quatro foram testados em dois cenários distintos: um deles com mobilidade sintética, sobre o mapa da cidade de Helsinki (Finlândia), e outro com mobilidade real, sobre o mapa de São Francisco (EUA). Para a avaliação proposta, foram definidas cinco métricas para a análise de desempenho dos referidos protocolos: taxa de dados entregues (satisfação); atraso de descoberta e recuperação; sobrecarga na rede (*overhead*); média de pacotes retransmitidos por nó; e consumo de energia. Os resultados obtidos constataram o melhor desempenho

do DIRESC em relação aos demais protocolos na maioria das métricas e cenários observados. De modo geral, nosso protocolo obteve um ganho percentual expressivo na taxa de entrega de dados em relação aos demais (pelo menos 24, 9% no cenário de mobilidade real), com menor atraso de descoberta e recuperação de dados e menor consumo de energia. Assim, o DIRESC se mostra promissor por acelerar o processo de obtenção de dados no ambiente de rede proposto.

**Palavras-chave:** Redes tolerantes a atraso, Redes centradas em conteúdo, DTN, CCN, DIRESC.

## ABSTRACT

Content-centric networks have emerged as an innovative paradigm that modifies the traditional, host-centric and address-based communication model for a content-based model. When combined with delay and interrupt tolerant networks, it can meet needs that make it a challenging paradigm, including for data collection. However, this combination also raises new challenges, such as conciliating the *broadcasting* message propagation scheme, natural of content-centric networks, with the need to reduce packet replications and resource depletion in delay tolerant networks, or to deal with the rupture of connections and the formation of new ways to search and retrieve data. Thus, we justify the feasibility of such a combination and present the DIRESC protocol, a solution proposal for the selection of message relays for the announcement, discovery and retrieval of data through a scheme of social cooperation between nodes, whose objectives cover the increase in the satisfaction rate of requests, reduction of data collection delay and fair distribution of load in the network, with less resource wastage. We implemented the CCN architecture and DIRESC in the ONE simulator (Opportunistic Network Environment) and compared its performance with three other protocols: *CCNbroadcasting*, which represents the standard behavior of CCN, STCR, a protocol also based on social relations, and PIFP, a probabilistic protocol. The four were tested in two distinct scenarios: one with synthetic mobility, on the Helsinki city map (Finland), and the other with real mobility, on the map of San Francisco (USA). For the proposed evaluation, five metrics were defined for the performance analysis of said protocols: delivered data rate (satisfaction), discovery and recovery delay, network overhead, average of retransmitted packets per node and energy consumption. The results obtained showed the best performance of DIRESC in relation to the other protocols in most of the metrics and scenarios observed. In general, our protocol obtained a significant percentage gain in data delivery rate in relation to the other (at least 24.9% in the real mobility scenario), with lower data discovery and recovery delay and lower energy consumption. Thus, DIRESC shows promise for speeding up the process of obtaining data in the proposed network environment.



**Keywords:** Delay-tolerant networks, Content-centric networks, DTN, CCN, DIRESC.

# Sumário

**1 Introdução** **1**

1.1 Motivação . . . . . 3

1.2 Justificativa . . . . . 3

1.3 Objetivos . . . . . 4

1.4 Contribuições . . . . . 5

1.5 Estrutura do trabalho . . . . . 6

**2 Fundamentação teórica e trabalhos relacionados** **7**

2.1 Redes tolerantes a atraso e interrupções . . . . . 7

2.1.1 Casos particulares de DTN . . . . . 8

2.1.2 Protocolos de roteamento . . . . . 10

2.1.3 Seleção de nós retransmissores . . . . . 11

2.2 Redes centradas em conteúdo . . . . . 13

2.3 CCN empregada à rede sem fio . . . . . 18

2.4 Elegibilidade de retransmissores em CCDTN . . . . . 21

**3 O protocolo proposto** **23**

3.1 Relacionamento social . . . . . 23

3.2 Fase de anúncio . . . . . 26

3.3 Fase de descoberta . . . . . 28
3.4 Fase de recuperação . . . . . 32

**4 Algoritmos, complexidade e estruturas 34**

4.1 Análise do cálculo de $R_i(j)$ e $R_{medio}$ . . . . . 34
4.2 Análise da fase de anúncio . . . . . 36
4.3 Análise da fase de descoberta . . . . . 38
4.4 Análise da fase de recuperação . . . . . 41
4.5 Complexidade do protocolo DIRESC . . . . . 43

**5 Avaliação de desempenho 45**

5.1 Ferramenta de simulação . . . . . 45
5.1.1 Geração de mensagens . . . . . 45
5.1.2 Arquitetura CCN . . . . . 47
5.1.3 Relação social . . . . . 48
5.2 Cenários de simulação . . . . . 49
5.2.1 Helsinki . . . . . 49
5.2.2 São Francisco . . . . . 50
5.3 Métricas de avaliação . . . . . 53
5.4 Resultados experimentais . . . . . 55
5.4.1 Cenário de Helsinki . . . . . 55
5.4.2 Cenário de São Francisco . . . . . 67

**6 Conclusão e trabalhos futuros 78**

# Lista de Figuras


2.1 Rede veicular tolerante a atraso (VDTN) . . . . . 8

2.2 Conectividade em regiões subdesenvolvidas . . . . . 9

2.3 Taxonomia de esquemas de roteamento oportunista em MSN . . . . . 12

2.4 Modelos dos pacotes de interesse e de dados . . . . . 13

2.5 Processamento de pacotes de interesse em CCN . . . . . 16

2.6 Processamento de pacotes de dados em CCN . . . . . 16

2.7 Modelo de encaminhamento CCN, via faces . . . . . 17

3.1 Anúncios direto e transitivo . . . . . 27

3.2 Possibilidades de envio de pacotes de interesse . . . . . 30

3.3 Processamento de pacotes de interesse no DIRESC . . . . . 31

3.4 Possibilidades de envio de pacotes de dados . . . . . 33

3.5 Processamento de pacotes de dados no DIRESC . . . . . 33

4.1 Classe *EntradaPIT* . . . . . 41

5.1 Classe *SocialTie* . . . . . 48

5.2 Mapa da cidade de Helsinki (Finlândia) . . . . . 50

5.3 Mapa da cidade de São Francisco (EUA) . . . . . 52

5.4 Taxa absoluta de dados entregues no cenário de Helsinki . . . . . 56
5.5 Taxa percentual de dados entregues no cenário de Helsinki . . . . . 57
5.6 Atraso médio no cenário de Helsinki . . . . . 60
5.7 Overhead de interesses e dados no cenário de Helsinki . . . . . 61
5.8 Média de retransmissões por nó no cenário de Helsinki . . . . . 63
5.9 Consumo percentual de energia no cenário de Helsinki . . . . . 65
5.10 Taxa absoluta de dados entregues no cenário de São Francisco . . . . . 67
5.11 Taxa percentual de dados entregues no cenário de São Francisco . . . . . 68
5.12 Atraso médio no tempo no cenário de São Francisco . . . . . 70
5.13 Overhead de interesses e dados no cenário de São Francisco . . . . . 72
5.14 Média de retransmissões por nó a cada intervalo no cenário de São Francisco 74
5.15 Consumo percentual de energia no cenário de São Francisco . . . . . 76

# Lista de Tabelas


4.1 Tabela (*hash*) de encontros (mapeamento entre nós encontrados e $E_{N_s}(i)$) 36

4.2 Tabela (*hash*) de relacionamentos $S_M(i)$ . . . . . 36

4.3 Componente CS da arquitetura CCDTN . . . . . 38

4.4 Componente FIB da arquitetura CCDTN . . . . . 38

4.5 Componente PIT da arquitetura CCDTN . . . . . 41

5.1 Parâmetros do cenário de Helsinki . . . . . 51

5.2 Parâmetros do cenário de São Francisco . . . . . 52

5.3 Taxa absoluta de dados entregues no cenário de Helsinki . . . . . 59

5.4 Taxa percentual de dados entregues no cenário de Helsinki . . . . . 59

5.5 Atraso médio de descoberta e recuperação (em minutos) no cenário de Helsinki . . . . . 61

5.6 Sobrecarga normalizada de interesses no cenário de Helsinki . . . . . 63

5.7 Sobrecarga normalizada de dados no cenário de Helsinki . . . . . 63

5.8 Pacotes retransmitidos por nó no cenário de Helsinki . . . . . 65

5.9 Consumo percentual de energia no cenário de Helsinki . . . . . 66

5.10 Taxa absoluta de dados entregues no cenário de São Francisco . . . . . 69

5.11 Taxa percentual de dados entregues no cenário de São Francisco . . . . . 70

5.12 Atraso médio de descoberta e recuperação (em minutos) no cenário de São Francisco . . . . . 71

5.13 Sobrecarga normalizada de interesses no cenário de São Francisco . . . . 74

5.14 Sobrecarga normalizada de dados no cenário de São Francisco . . . . . 74

5.15 Pacotes retransmitidos por nó no cenário de São Francisco . . . . . 75

5.16 Consumo percentual de energia no cenário de São Francisco . . . . . 77

# Lista de Nomenclaturas

| | |
|---|---|
| DTN | Delay/Disruption-Tolerant Networks |
| MANET | Mobile or Multihop Ad-hoc Networking |
| PSN | Pocket-Switched Networks |
| PCD | Personal Communication Devices |
| V2V | Vehicle-to-Vehicle |
| V2I | Vehicle-to-Infrastructure |
| TCP | Transmission Control Protocol |
| IP | Internet Protocol |
| ICN | Information-Centric Networks |
| CCN | Content-Centric Networks |
| NDN | Named Data Networks |
| CCDTN | Content-Centric Delay/Disruption-Tolerant Networks |
| VDTN | Vehicular Delay-Tolerant Networks |
| VANET | Vehicular Ad-hoc Networks |
| RSU | Road Side Unit |
| PROPHET | Probabilistic Routing Protocol using a History of Encounters and Transitivity |
| CS | Content Store |
| FIB | Forwarding Information Base |
| PIT | Pending Interest Table |
| MSN | Mobile Social Networks |
| MCC | Mobile Cloud Computing |
| NDM | Named Data MANETs |
| LFBL | Listen-First, Broadcast-Later |
| STCR | Social-Tie and Content Retrieval |
| PIFP | Probabilistic Information Forwarding Protocol |
| DIRESC | Discovery and Retrieval based on Social Cooperation |

# 1. Introdução

As redes tolerantes a atraso ou interrupções (do inglês, *Delay/Disruption Tolerant Networking* - DTN), emergentes a partir do paradigma MANET (*Mobile or Multihop Ad-hoc Networking*), presumem a existência de cenários de rede sem fio sem infraestrutura e de múltiplos saltos, que sofrem longos atrasos e frequentes desconexões, nos quais não se garante a existência de caminhos fim-a-fim entre o par origem-destino. Tratam-se de ambientes de rede auto-organizáveis em que os dispositivos se comunicam de maneira oportunista, ou seja, por meio de oportunidades de contato entre si para encaminhar pacotes ao longo da rede [1]. Nesse contexto, destacam-se alguns casos particulares de tal paradigma, como as redes de comutação de bolso (do inglês, *Pocket-Switched Networks* - PSNs), formadas por indivíduos portando dispositivos de comunicação pessoal avançados (*Personal Communication Devices* - PCDs), capazes de estabelecer conexões diretas entre si, como *smartphones, tablets, notebooks*, etc [2].

Um dos principais aspectos a se considerar acerca do paradigma DTN é a mobilidade dos nós, em vista que, por meio dela, ocorrem tanto os contatos entre nós, quanto a ruptura de comunicação entre eles e, por consequência, o encerramento da troca de dados, instante no qual se observa um estado de desconexão ou particionamento da rede, desafio com o qual tal paradigma se propõe a lidar.

Algumas características permeiam o funcionamento dessas redes, como a alta dinamicidade das topologias, decorrente da vasta mobilidade dos nós; a intermitência, em razão das muitas rupturas de comunicações; e a baixa conectividade, que ocorre devido não apenas às mudanças de topologia e à intermitência, mas pela incapacidade de dispositivos de participar das comunicações da rede em vista de suas limitações de recursos, como capacidade de *buffer* e energia [1]. Em razão de tais características, os protocolos de roteamento tradicionais da arquitetura TCP/IP não se aplicam nesse contexto, isto é, não é funcional a implementação de protocolos que prevêem comunicações fim-a-fim em cenários em que ela é inexistente. Portanto, soluções de rede alternativas para a recuperação de paco-

tes dentro de tais ambientes têm tentado fugir de abordagem centradas em hospedeiro e fincar-se em comunicações centradas em conteúdo [3].

Como alternativa, as redes centradas em informação (*Information-centric Networks* – ICN) têm emergido como um paradigma inovador para a recuperação de dados e serviços. Estudos anteriores destacam sua importância e como a sua aplicação em diferentes cenários pode agregar diversos benefícios para o atendimento de demandas provenientes de tais ambientes, mesmo aqueles para os quais o mesmo não foi previamente pensado, tais como DTNs, PSNs, entre outros. Recentemente, algumas arquiteturas que compõem o referido paradígma têm recebido maior destaque que outras na elaboração de propostas que abordam a sua agregação em ambientes de rede desafiadores, como os descritos anteriormente. São os casos das arquiteturas CCN [4] e NDN [5] que, por vezes, se confundem em razão de sua semelhança em aspectos comportamentais, mas se diferem em termos de implementação: enquanto CCN está disponível a partir do projeto CCNx [6], a arquitetura NDN apresenta o protótipo NFD [7].

Neste contexto, a arquitetura de redes centradas em conteúdo (do inglês, *Content-Centric Networking - CCN*), conceitualmente introduzida por [4], tem emergido como uma promissora abordagem de redes para a recuperação de dados a partir da desconstrução do conceito de endereçamento no processo de busca por provedores de serviços. Diferentemente das redes centradas em hospedeiro (*host-centric*), CCN foca na identificação do conteúdo buscado, modificando o paradigma de comunicação para uma rede centrada em dados (*data-centric*).

Em CCN, a comunicação é conduzida pelos dispositivos ou nós consumidores, também chamados de requisitantes, que, quando interessados em conteúdos, disseminam pacotes de interesse na rede. Quando esses atingem um provedor do conteúdo buscado, o mesmo envia os pacotes correspondentes à demanda daquele requisitante, concluindo o processo de busca e entrega dos dados. Além do encaminhamento de interesses baseado em nomes de conteúdos, o paradigma CCN se caracteriza também pela replicação de pacotes de conteúdo em cache na rede. Assim, o nó que responde a uma solicitação pode ser tanto um provedor de conteúdo quanto um nó intermediário que possua uma cópia destes pacotes em cache. Esse armazenamento ao longo da rede facilita a entrega dos dados ao requisitante [4]. Além disso, a cada bloco de conteúdo é associado um nome, também chamado de prefixo, que independe da localização do seu provedor e são usados por aplicações para descoberta e recuperação dos dados.

Os nós que compõem um cenário CCN possuem três componentes básicos [4]: a *content store* (CS), que é a cache para armazenamento de dados; a *forwarding information*

*base* (FIB), uma tabela que mapeia os anúncios de provedores de conteúdo às interfaces pelas quais foram recebidos; e a *pending interest table* (PIT), uma tabela que registra as requisições de conteúdos realizadas e ainda não satisfeitas, mapeando-as para as interfaces por onde foram entregues.

## 1.1 Motivação

Em [8] são destacados alguns benefícios que o paradigma CCN é capaz de prover ao funcionamento de DTN para obtenção de dados ou serviços. Dentre eles, destaca-se o fato de CCN permitir comunicações assíncronas entre um par requisitante-provedor, isto é, sem a necessidade de que esses nós estabeleçam e mantenham uma comunicação fim-a-fim. Outro benefício é a busca de dados por nomes, que facilita a sua recuperação em ambientes de alta mobilidade, nos quais não se conhece a localização precisa de um nó, como ocorre em DTN. Por fim, a replicação de dados na rede reduz o número de requisições e o atraso na recuperação dos mesmos.

Além disso, CCN é um paradigma cujas características combinam com o funcionamento de ambientes de redes sem fio. Sendo assim, tal paradigma pode superar as ineficiências do TCP/IP em relação à mobilidade dos nós, a *links* de baixa confiabilidade e às limitações de recursos ao flexibilizar a necessidade de criar e manter conexões estáveis entre pontos finais (par origem-destino) [3]. Por fim, mesmo que o provedor seja conhecido pelo requisitante, alcançá-lo para a recuperação de conteúdo por meio do roteamento de pacotes usando protocolos orientados a hospedeiro em DTN ainda é um desafio. Portanto, a utilização de CCN no contexto de comunicação oportunista é capaz de auxiliar o processo de obtenção de dados, de modo que eles não precisem, necessariamente, ser recuperados diretamente do provedor (que, inclusive, pode nem sequer ser conhecido pelo nó requisitante), mas possam ser obtidos a partir da cache de nós intermediários, reduzindo o custo de sua entrega.

## 1.2 Justificativa

Apesar dos benefícios que a aplicação CCN é capaz de prover no contexto de DTN, essa combinação levanta questões de pesquisa que ainda precisam ser estudadas. Em DTN, por exemplo, um dos principais desafios de roteamento é determinar a estratégia apropriada de seleção de nós retransmissores a fim de aumentar a probabilidade de entrega de pacotes, minimizando o número de replicações na rede e reduzindo, se possível,

o atraso de tal entrega. O protocolo Epidêmico [9], que é baseado na transmissão de mensagens em toda oportunidade de contato, apresenta bons resultados em termos de atraso e taxa de entrega, mas acarreta um alto custo em replicações de pacotes e, portanto, sobrecarga na rede (*overhead*) [10]. Já as comunicações em CCN foram idealizadas para operarem predominantemente em *broadcast* e *multicast* [4]. Portanto, uma tendência dos trabalhos mais recentes, que utilizam a agregação entre CCN e DTN, é encontrar um meio termo entre a política de funcionamento daquele (conceitualmente projetado para operar em *broadcast* e *multicast*), às limitações deste (que precisa evitar sobrecarga e exaustão de recursos nos nós). Com isso, almeja-se tornar os processos de encaminhamento mais "conscientes" por meio de estratégias como a seleção de retransmissores, a fim de evitar o comportamento epidêmico na comunicação entre os nós [3].

Assim, existem trabalhos na literatura que utilizam diferentes estratégias para a definição de seus mecanismos de transmissão de pacotes nesse ambiente de redes, como [11], [12], [13], entre outros. Dentre as estratégias usadas, a seleção de nós retransmissores (ou de próximo salto) é uma das mais correlacionadas às propostas de protocolos voltados especificamente para DTN, cujo funcionamento busca evitar o comportamento epidêmico.

Essa estratégia se propõe a limitar o número de nós que recebem pacotes, por meio de uma escolha baseada em algum critério. A partir do escopo de estudo delimitado e na referida estratégia, neste trabalho propomos o DIRESC (*DIscovery and REtrieval based on Social Cooperation*), um protocolo para anúncio, descoberta e recuperação de dados em redes centradas em conteúdo e tolerantes a atraso (do inglês, *Content-Centric Delay Tolerant Network* - CCDTN) baseado na cooperação entre nós que possuem determinado grau de relacionamento social. Nosso trabalho define três possibilidades de envio de requisições e outras três para entrega de conteúdo, diferentemente de outras abordagens que, além de dar ênfase no processo de requisição dando menor destaque ao processo de entrega de dados, definem um único critério para determinar a competência de um nó em entregar um pacote. Assim, utilizamos a estratégia de seleção de retransmissores de maneira mais distribuída e, portanto, sem exigir intensamente de apenas um conjunto de nós específico na rede.

## 1.3 Objetivos

Nossa proposta tem como objetivo principal definir uma estratégia de seleção de nós retransmissores para o contexto de redes CCDTN, por meio da qual se eleve a taxa de satisfação de requisições, bem como se reduza o atraso de descoberta e recuperação de

conteúdo e a sobrecarga gerada, em comparação com as propostas da literatura, de modo que seja evitado o envio indiscriminado de mensagens na rede, o que representa um impacto significativo na quantidade de transmissões realizadas e pacotes replicados. Assim, recursos como energia e capacidade de *buffer* podem ser poupados, contribuíndo para o melhor funcionamento da rede.

O esquema de cooperação social proposto se baseia no relacionamento existente entre os nós, calculado por meio do seu histórico de encontros. Esse relacionamento é capaz de influenciar a probabilidade de entrega das mensagens, uma vez que permite conjecturar se um par de nós específico possui chances de se reencontrar na rede, qualificando-os ou não como aptos à retransmissão de pacotes.

O protocolo proposto se divide em três etapas: anúncio, descoberta e recuperação de dados. Cada uma delas leva em consideração o fator de relacionamento social entre nós e, de modo geral, visam reduzir a distância (temporal e em número de saltos) entre um requisitante e o dado solicitado.

## 1.4 Contribuições

As principais contribuições deste trabalho podem ser resumidas da seguinte forma:

1. A proposta do protocolo DIRESC para anúncio, descoberta e recuperação de dados em CCDTN. O protocolo proposto obteve um aumento no percentual de satisfação de requisições, ou seja, na relação entre interesses enviados e atendidos, mostrando resultados melhores quando comparado com outras propostas. Além disso, o DIRESC apresentou um expressivo decaimento no atraso médio da recuperação de dados quando comparamos nossa proposta com outras da literatura.

2. A implementação da arquitetura CCN e do protocolo DIRESC no ambiente de simulação ONE (*Opportunistic Network Environment*), projetado para reproduzir redes toletantes a atraso.

3. Um estudo sobre a maneira como os relacionamentos sociais entre nós podem auxiliar as tarefas de anúncio, descoberta e recuperação de dados em CCDTN de modo a evitar o comportamento epidêmico na disseminação de mensagens, bem como a seleção de nós preferidos para a execução destas tarefas.

4. Uma análise de complexidade sobre cada fase do protocolo, considerando os componentes propostos pela arquitetura CCN, bem como as estruturas de dados utilizadas.

## 1.5 Estrutura do trabalho

O Capítulo 1 exibe um breve apanhado acerca de redes DTN, pontuando suas principais características. Apresenta também a motivação e justificativa para o emprego de CCN neste contexto. Por fim, expõe os objetivos deste trabalho e suas contribuições.

O Capítulo 2 apresenta a fundamentação teórica que embasa nosso trabalho. Expõe os conceitos fundamentais de redes DTN e CCN. Aborda brevemente trabalhos que estudam a seleção de nós retransmissores em redes tolerantes a atraso. Apresenta outras propostas da literatura que também se propõem a empregar CCN no contexto de redes sem fio para a recuperação de conteúdo. Por fim, expõe as lacunas ao se eleger nós retransmissores para a obtenção de dados em CCDTN.

O Capítulo 3 apresenta o detalhamento do protocolo proposto para o ambiente de redes estudado. Explica como é realizado o cálculo de relacionamento social entre os nós e as fases de tomadas de decisão.

O Capítulo 4 aprofunda o estudo sobre o protocolo proposto. Apresenta o algoritmo correspondente a cada fase definida no Capítulo 3. Exibe uma análise de complexidade para tais algoritmos e as estruturas de dados utilizadas na sua implementação, as quais correspondem a um impacto significativo naquele estudo.

O Capítulo 5 detalha a ferramenta de simulação utilizada, bem como as modificações em seu código para que atendesse às necessidades de representação do ambiente de rede proposto. Descreve os cenários de avaliação do referido protocolo em comparação com outros três protocolos da literatura. Define as métricas de avaliação e suas formulações. Por fim, apresenta os resultados das simulações em forma de gráficos e a sua análise.

O Capítulo 6 apresenta as conclusões relativas à proposta e sugestões de trabalhos futuros.

# 2. Fundamentação teórica e trabalhos relacionados

Este capítulo apresenta os conceitos fundamentais de redes tolerantes a atraso e interrupções, bem como aqueles que concernem às redes centradas em conteúdo. Explana acerca da estratégia de seleção de nós retransmissores, justificando o seu emprego neste trabalho. Apresenta trabalhos relacionados ao nosso estudo, ou seja, que também propõem uma solução para o emprego de CCN no contexto de redes sem fio. Por fim, expõe, com base nos trabalhos mencionados da literatura, a dificuldade em se eleger nós retransmissores para a recuperação de conteúdo em DTN.

## 2.1 Redes tolerantes a atraso e interrupções

As redes tolerantes a atraso e interrupções compõem um paradigma de redes sem fio sem infraestrutura e de múltiplos saltos caracterizado pela ocorrência de longos atrasos e frequentes desconexões na comunicação entre os nós. Neste paradigma, um fator de grande influência é a mobilidade destes nós, responsável pela manifestação de tais características.

Segundo [1], as redes tolerantes a atraso se encontram frequentemente particionadas devido a diversas razões, dentre as quais: descarga de bateria dos dispositivos; curto alcance de rádio transmissão combinado com a mobilidade e a dispersão dos nós; e, por fim, a baixa confiabilidade de enlaces sem fio. Em razão disso, neste paradigma de rede, não há garantia da existência de caminhos fim-a-fim entre um par de nós remotos. Portanto, o conjunto de pacotes a ser entregue da origem ao destino precisa não apenas perpassar pelos nós intermediários, mas ser carregado por eles, nó a nó, até o destinatário da mensagem.

Em DTN, todos os nós que compõem a rede são dispositivos finais, sejam eles *smartphones, tablets* ou veículos. No entanto, se comportam como roteadores, armazenando os pacotes recebidos e aguardando a oportunidade de retransmití-los a outros nós. Por esta

razão, sua comunicação é chamada armazenar-carregar-e-encaminhar (do inglês, *store-carry-and-forward*), ou seja, os nós intermediários carregam pacotes até que hajam oportunidades de retransmissão, em oposição ao modelo armazenar-e-encaminhar (do inglês, *store-and-forward*), utilizado em comunicações fim-a-fim, nas quais estes nós, sempre conectados, não precisam carregar os pacotes de outros, mas os retransmitem diretamente após recebê-los [1].

### 2.1.1 Casos particulares de DTN

Como mencionado no Capítulo 1, as redes de comutação de bolso se destacam neste paradigma como sendo um de seus casos particulares. No entanto, além das PSNs, outros tipos de redes também são consideradas como tolerantes a atraso e interrupções.

As redes veiculares tolerantes a atraso (do inglês, *Vehicular Delay-Tolerant Networks* - VDTN), ou DTNs veiculares, são um caso particular das redes veiculares (do inglês, *Vehicular Ad-hoc Networks* - VANETs). Em ambos os casos, os dispositivos finais são veículos que trafegam ao longo de rodovias, e que podem se comunicar com outros veículos (*Vehicle-to-Vehicle* - V2V), ou com a rede infraestruturada disposta ao longo destas rodovias (*Vehicle-to-Infrastructure* - V2I) [1]. As VDTN, no entanto, possuem um cenário esparso, de modo que caminhos fim-a-fim entre o par origem-destino podem não existir. Assim, devido ao estado predominantemente particionado deste tipo de VANET, e ao alcance de transmisssão limitado dos pontos de acesso localizados à margem das rodovias (*Road Side Unit* - RSU), os nós precisam armazenar mensagens em *buffer* e encaminhá-las oportunisticamente a outros nós da rede [14]. A Figura 2.1 apresenta um cenário VDTN.

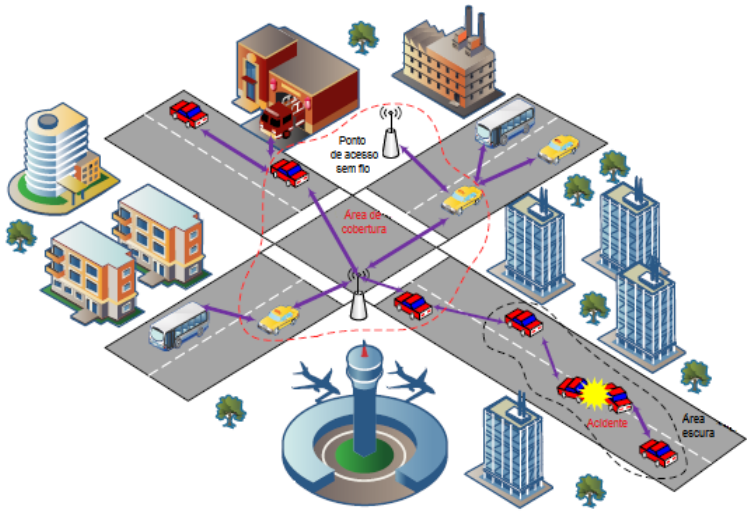


Figura 2.1: Rede veicular tolerante a atraso (VDTN)
[15]

Também se configuram como redes tolerantes a atraso algumas redes de sensores sem fio (do inglês *Wireless Sensor Networks* - WSN), emergentes da crescente integração de circuitos de comunicação e sensores em dispositivos miniaturizados [1]. Por meio delas, tornou-se possível controlar e monitorar eventos e fenômenos, por exemplo, no rastreamento de vida selvagem (*Wildlife Tracking*), que são de grande importância para o desenvolvimento e avanço das ciências relacionadas à biologia, por coletarem informações sobre mobilidade e migração de animais e sua interação com outras espécies. Um exemplo é o projeto ZebraNet [16], que equipa zebras com sensores *ad-hoc* sem fio com considerável largura de banda e poder computacional, e permite aos pesquisadores coletar informações sobre o comportamento destes animais.

Redes DTN também são aplicáveis em regiões de subdesenvolvimento, isto é, a partir do fornecimento de conectividade à ambientes remotos, nos quais há poucos sistemas computacionais com conexão habilitada. Por serem afastados dos grandes centros, estas regiões acabam tendo demandas de serviços, como acesso a e-mails por exemplo, que, infortunadamente, não podem ser atendidas de imediato pela falta de conexão com a Internet. Assim, as requisições de usuários destes ambientes são armazenadas em *buffer* em pontos de acesso dispostos nestes vilarejos afastados e transmitidas, oportunisticamente, via rede sem fio, a veículos com conexão habilitada (ônibus, carros, etc., equipados com tecnologia *Wi-Fi*). Estes, por sua vez, retrasmitem estas requisições dentros dos centros urbanos com acesso a Internet [15]. Um esquema desta aplicabilidade de DTN é apresentado na Figura 2.2.

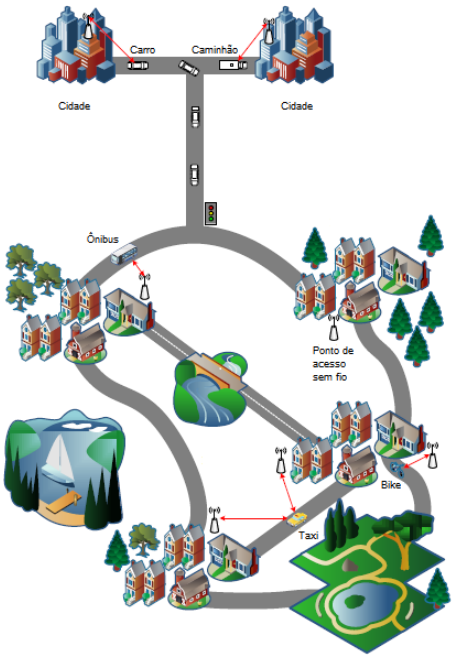


Figura 2.2: Conectividade em regiões subdesenvolvidas [15]

### 2.1.2 Protocolos de roteamento

Além dos cenários de aplicação de DTN, um aspecto de considerável importância acerca deste paradigma são os protocolos de roteamento da literatura, empenhados em superar as limitações destas redes. Assim, um dos desafios mais pertinentes em DTN é definir uma estratégia de envio de pacotes de modo a maximizar a probabilidade e a taxa de entrega, minimizando a sobrecarga na rede e, se possível, o atraso. Assim como outros trabalhos da literatura, [17] classifica os diferentes protocolos de roteamento para redes tolerantes a atraso em classes. São elas:

**Baseados em inundação**: esta classe propõe a replicação de mensagens e o envio destas réplicas a outros nós quando conexões são estabelecidas. O uso de muitas cópias de um pacote aumenta a probabilidade e a taxa de entrega de mensagens, em comparação com os protocolos baseados em encaminhamento, a serem tratados a seguir. Dentre os protocolos de roteamento em DTN baseados em inundação, destacam-se o Epidêmico [9], o *Spray-and-Wait* [18] o *PROPHET* [19], entre outros.

A estratégia do protocolo Epidêmico [9] é de que os nós encaminhem pacotes em toda oportunidade de contato, ou seja, a todo e qualquer nó dentro do alcance de transmissão. Esta política de encaminhamento garante uma alta taxa de entrega e com baixo atraso; contudo, o seu funcionamento inunda a rede com pacotes devido as muitas replicações feitas pelos nós intermediários, elevando significativamente o custo em termos de sobrecarga e consumo de energia [10].

O protocolo *Spray-and-Wait* [18], como uma alternativa a sobrecarga de mensagens gerada pelo Epidêmico, tenta solucionar este problema por meio do pré-estabelecimento da quantidade de cópias a serem feitas de uma mesma mensagem. O protocolo opera em duas etapas: na primeira (*spray*), o nó origem cria uma determinada quantidade de cópias de uma mensagem; na segunda (*wait*), um nó intermediário que recebe esta mensagem aguarda o contato com o destinatário para que possa entregá-la diretamente sem ter que retransmiti-la a outros nós.

Por sua vez, o protocolo *PROPHET* [19] se baseia no histórico de encontros entre os nós para determinar a probabilidade que eles possuem de se (re)encontrar na rede e, assim, determinar quais deles são candidatos propensos a entregar a mensagem ao destinatário. Assim, um nó de posse de uma mensagem faz o seu envio a todos os nós vizinhos que possuem probabilidade de encontro com outro nó específico maior que a sua.

**Baseados em encaminhamento**: Nesta classe, mensagens são encaminhadas a outros nós com base em informações contextuais (refere ao estado atual do nó, como nível da

bateria, velocidade, direção, etc.), históricas (usadas para inferir comportamentos futuros da rede e obtido ao longo do tempo, como tempo de contato, histórico de encontros) ou socias (usado para descrever relações entre os usuários, prevendo comportamentos e melhorando a eficácia das políticas de encaminhamento). Destacam-se nesta classe os protocolos *BubbleRap* [20], *SimBet* [21] e *SimBetTS* [22], entre outros.

Na estratégia definida pelo protocolo *BubbleRap* [20], os nós são organizados no cenário de rede em comunidades. Seu funcionamento determina que, se o destinatário de uma mensagem pertence à mesma comunidade que o nó encaminhador, o envio do pacote é realizado para o nó de maior centralidade dentro de tal comunidade (centralidade local). Caso contrário, isto é, se destinatário e encaminhador pertencem a comunidades distintas, o envio é realizado para o nó de maior centralidade global, ou seja, independente de comunidade, até que a mensagem atinja aquela a qual pertence o nó destino.

Já os protocolos *SimBet* [21] e *SimBetTS* [22], por sua vez, utilizam duas métricas sociais (centralidade e simularidade) para estimar e prever quais os nós em portencial (de acordo com os valores obtidos por tais métricas) a se tornarem retransmissores e encontrar o destinatário da mensagem. Ambos, no entanto, evitam a troca destas métricas entre todos os nós da topologia; em vez disso, a centralidade social de cada nó é medida apenas em sua vizinhança, enquanto que a similaridade social (número de vizinhos em comum) é avaliada entre dois nós em contato.

Além dos protocolos mencionados, a literatura apresenta ainda muitas outras propostas para o roteamento de pacotes em DTN. Desta forma, observa-se que, devido às características que compõem os diversos cenários de redes tolerantes a atraso, muitos são os esforços para superar as limitações decorrentes de tais características de modo que a comunicação entre os nós seja possível.

### 2.1.3 Seleção de nós retransmissores

De modo que se aumente a probabilidade de entrega de mensagens, reduzindo o atraso e o consumo de recursos na rede, a maioria dos protocolos de roteamento para DTN determina, por meio de algum criério, a aptidão de nós para a retransmissão de pacotes visando atender os referidos pressupostos.

A literatura apresenta trabalhos centrados no estudo de medidas a serem consideradas para a definição de estratégias de seleção capazes de mitigar as limitações de cenários DTN. É apresentado em [10], por exemplo, um apanhado de estratégias de roteamento para redes móveis, baseadas em seleção de nós retransmissores por meio de critérios como:

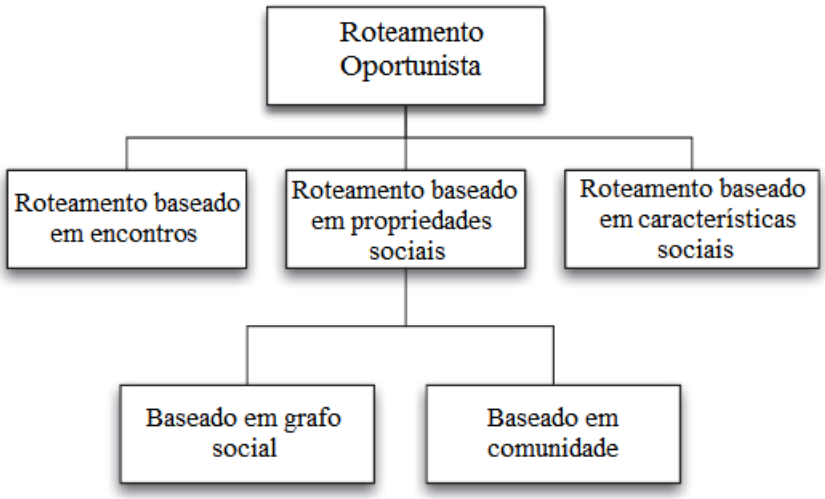


Figura 2.3: Taxonomia de esquemas de roteamento oportunista em MSN [10]

encontros, características ou propriedades sociais. A Figura 2.3 apresenta a taxonomia destas estratégias.

Segundo o texto, os esquemas de roteamento baseados em encontros levam em consideração o histórico de sua ocorrência e assim, encaminham pacotes de acordo com a frequência contabilizada de contatos e escolhem como retransmissor aquele com maior probabilidade de encontro com outros nós. Já o roteamento baseado em propriedades sociais dá importância ao *status* do nó e ao seu relacionamento dentro da rede. Na vertente deste tipo de roteamento baseada em grafo social, destacam-se métricas como: o grau do nó, definido pelo número de contatos estabelecidos com outros nós; a força de relação social, determinado pelo peso da aresta entre um par de nós no grafo social; a centralidade, que estabelece a importância do nó na rede, pela representação do quão conectado ele está a outros nós; e a similaridade social, definida pelos contatos em comum de dois nós no grafo. O conceito de comunidade, por outro lado, estabelece que indivíduos (neste caso, nós) com interesses ou localização geográfica comuns compõem comunidades, o que lhes permite encontrarem uns aos outros com maior frequência e regularidade em relação ao que aconteceria com indivíduos de outras comunidades. Por fim, os esquemas de roteamento baseados em características sociais consideram o conjunto de atributos inerentes em cada nó, isto é, aspectos intrínsecos, os quais não precisam ser calculados ou medidos para serem determinados, como por exemplo, sexo, naturalidade, profissão, etc.

Os experimentos de [23], por sua vez, apresentam uma avaliação minuciosa entre protocolos que levam em conta localização e propriedades sociais, comparando seus resultados em termos de taxa de entrega, atraso e custo. Por fim, o trabalho sugere que informações sociais são um bom subterfúgio para o desenvolvimento de mecanismos de

roteamento baseados em seleção de nós retransmissores em redes móveis.

A idealização de protocolos baseados em seleção de nós retransmissores tem se mostrado uma resposta viável não apenas para enviar mensagens em DTN, mas para se requisitar e recuperar dados neste tipo de rede, de modo a suprimir o impacto negativo do comportamento epidêmico. Embora a literatura tenha dedicado esforços em buscar novas soluções para DTN, dentre as quais, a utilização de CCN para facilitar o envio e obtenção de pacotes [24], faz-se ainda necessário evitar que o envio de mensagens seja feito de maneira epidêmica, por meio do encaminhamento cego (*blind forwarding*) típico de grande parte das comunicações CCN [4]. Deste modo, a seleção de nós é uma abordagem situada no meio termo entre o modo de funcionamento padrão de CCN e as limitações impostas por ambientes DTN, quando combinados estes dois paradigmas de rede. Portanto, na Seção 2.2, abordaremos os conceitos referentes a arquitetura e funcionamento de CCN, de modo a embasar a proposta de solução apresentada no próximo Capítulo.

## 2.2 Redes centradas em conteúdo

As redes centradas em conteúdo propõem uma mudança no modelo de entrega de pacotes, tradicionalmente baseado em hospedeiro e endereçamento, para outro, no qual o encaminhamento de pacotes é realizado com base nos nomes dos conteúdos buscados, sendo esta a principal característica deste tipo de rede. Além disso, outro princípio básico deste paradigma é o armazenamento de conteúdos em cache, a fim de facilitar a entrega dos mesmos aos solicitantes. Assim, a arquitetura CCN permite a recuperação de tais conteúdos por meio do envio de dois tipos de pacotes: interesse e dados [4], cujos modelos são apresentados na Figura 2.4.

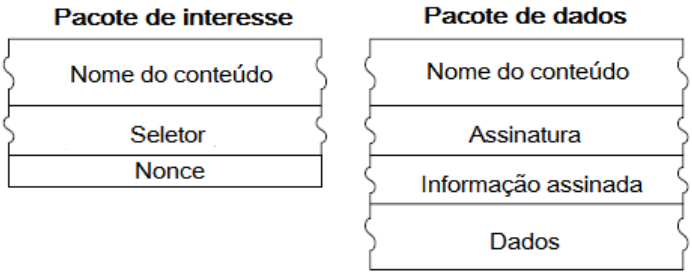


Figura 2.4: Modelos dos pacotes de interesse e de dados
[4]

Como apresentado na Figura 2.4, tanto pacotes de interesses quanto de dados possuem o campo correspondente ao nome (prefixo) do conteúdo a ser buscado ou entregue. No caso do pacote de interesse, há outros dois campos: o seletor, que corresponde à especifi-

cação de um bloco do conteúdo buscado, ou seja, uma parcela de tal conteúdo em vez do todo; e *nonce*, que compreende um mecanismo de controle de duplicidade de requisições, de modo que um provedor não responda mais de uma vez a uma mesma solicitação de conteúdo. Trata-se, portanto, de um identificador único para aquele pacote.

Por sua vez, o pacote de dados, além do campo correspondente ao nome do conteúdo, possui outros três campos: a assinatura, que diz respeito à validação do pacote, isto é, à autenticação entre o nome, o conteúdo correspondente e uma pequena quantidade de dados de suporte fornecidos (informação assinada); a informação assinada que, como dito anteriormente, se trata de uma porção de dados de suporte como identificador do produtor e informações sobre chaves de criptografia, entre outros; por fim, os próprios dados correspondentes ao prefixo do campo nome de conteúdo.

Além disso, em um cenário CCN podemos identificar, de acordo com a literatura, três tipos de atores (ou nós), cujos papeis são descritos de acordo com a sua atuação na rede, isto é, requisição ou fornecimento de conteúdo, ou encaminhamento de pacotes. A seguir, são descritos os comportamentos característicos de cada um deles:

- **Consumidor**: também chamado de requisitante ou solicitante, é o nó cuja aplicação demanda algum conteúdo e que, por esta razão, envia solicitações (interesses) na rede a fim de obtê-lo.
- **Produtor de conteúdo**: também chamado de provedor, é o nó que executa uma aplicação que gera os pacotes de conteúdo (dados) e, portanto, é o nó primário a atender a solicitação de um requisitante.
- **Intermediários**: são nós dispostos entre o requisitante e o provedor de conteúdo. Estes nós podem ser representados, dentro da rede, tanto por roteadores quanto por dispositivos finais de usuários. Eles possuem capacidade de armazenamento em cache, portanto, podem atender a solicitação de um requisitante, como provedor de conteúdo secundário.

Em CCN, a comunicação é conduzida pelos nós consumidores que, quando interessados em determinado conteúdo, disseminam pacotes de interesse na rede. Quando estas requisições chegam até um nó que possua o conteúdo buscado, tal nó envia os pacotes de dados ao requisitante, concluindo o processo de busca e entrega dos dados. Como o paradigma CCN também se caracteriza pela replicação de pacotes de dados em cache na rede, o nó que responde a uma solicitação pode ser tanto um produtor de conteúdo quanto um nó intermediário que possui uma cópia de tais pacotes em cache. Este armazenamento

ao longo da rede facilita a entrega dos dados ao requisitante [4]. Além disso, a cada bloco de conteúdo é associado um nome que independe da localização do seu provedor; estes nomes, também chamados de prefixos, são usados por aplicações para descoberta e recuperação dos dados.

Os nós que compõem um cenário CCN possuem três componentes básicos [4]: a *content store* (CS), que é a cache para armazenamento de dados; a *forwarding information base* (FIB), uma tabela que mapeia os anúncios de provedores de conteúdo à interface pela qual tal anúncio foi recebido; e a *pending interest table* (PIT), uma tabela que registra as requisições de conteúdos realizadas e ainda não satisfeitas, mapeando-as às interfaces por onde tais requisições chegaram. Registros são adicionados a cada um destes componentes de acordo com as comunicações que ocorrem dentro do cenário CCN. São destacadas em [4] e [8], a existência de três diferentes tipos de comunicação neste ambiente. São elas:

- **Anúncio**: corresponde ao envio, a partir dos provedores e para outros nós, de um registro contendo os nomes dos conteúdos que aqueles são capazes de atender. Quando tal envio é feito, via *broadcast*, os nós que o recebem inserem-no em suas FIBs fazendo o mapeamento do registro com a interface pela qual ele foi recebido. Este mapeamento permite aos nós que recebem pacotes de interesse saber para qual interface direcionar a retransmissão de tais pacotes, de modo que eles possam ser entregues ao provedor de conteúdo.

- **Descoberta**: refere-se ao envio de pacotes de interesse por parte do requisitante até que estes alcancem um nó provedor. Caso o nó que receba o pacote de interesse possua o conteúdo correspondente armazenado em cache, ele o envia para o requisitante. Caso contrário, este mesmo nó verifica em sua PIT se há um registro de solicitação (feita por outro nó) para aquele mesmo conteúdo; caso haja, a interface pela qual o interesse foi recebido é inserida neste mesmo registro, e é esperado o recebimento do pacote de dados, sem que seja necessário uma nova retransmissão do interesse. Caso não haja um registro correspondente na PIT, o nó verifica se há um registro na FIB que indique uma interface pela qual retransmitr o interesse; caso haja, o nó cria uma nova entrada na PIT, registrando o nome do conteúdo buscado e vinculado-o à interface pela qual o interesse foi recebido. Caso não haja um registro correspondente na FIB, o pacote de interesse é descartado. A Figura 2.5 apresenta o fluxo deste processo.

- **Recuperação**: trata-se do envio de pacotes de dados, em resposta a uma solicitação, até que atinjam o nó requisitante. Quando um nó recebe um pacote de dados, ele verifica se existe uma solicitação pendente para aquele conteúdo na PIT; caso haja,

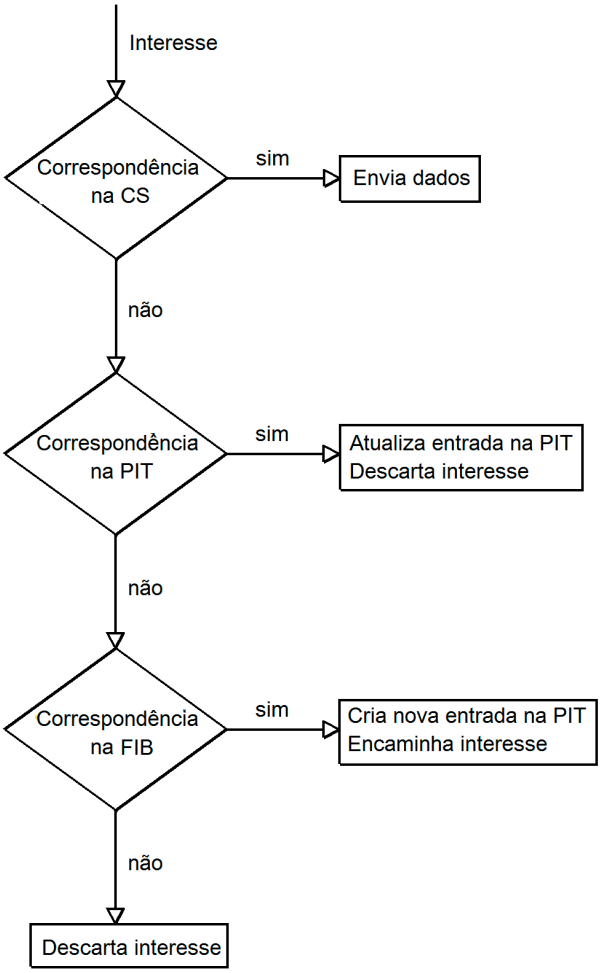


Figura 2.5: Processamento de pacotes de interesse em CCN

ele envia o pacote de dados pelas interfaces solicitantes, em seguida, o armazena na CS e, por fim, exclui a entrada de PIT; caso não haja, o pacote de dados é descartado. A Figura 2.6 apresenta o fluxo deste processo.

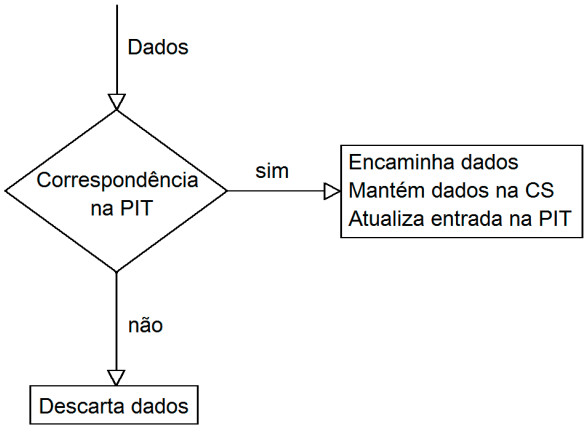


Figura 2.6: Processamento de pacotes de dados em CCN

Além disso, CCN faz uso do conceito de interfaces (ou faces) por meio das quais os anúncios e pacotes são recebidos e transmitidos. O seu identificador é utilizado para vinculá-la a um dos componetes de mapeamento da arquitetura CCN (PIT ou FIB), de modo que seja possível responder a uma solicitação pela mesma interface pela qual aquela adentrou, ou buscar um conteúdo pela face de entrada de um anúncio. Conceitualmente, CCN pode empregar diferentes tipos de interfaces, como mostra a Figura 2.7. No entanto, ao utilizarmos CCN no âmbito de redes DTN, assumiremos que interfaces equivalem a nós. Assim, faremos o mapeamento entre anúncio ou solicitação aos nós que efetuaram a sua transmissão.

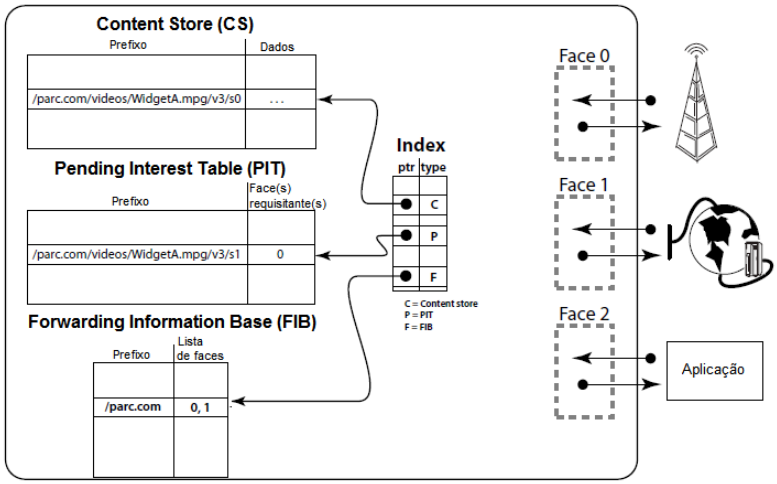


Figura 2.7: Modelo de encaminhamento CCN, via faces
[4]

São destacados em [3] alguns fatores intrínsecos em CCN e que são capazes de influenciar positivamente a aplicação deste paradigma em um ambiente de redes sem fio. Primeiramente, o suporte a mobilidade, tanto do consumidor, que pode submeter um mesmo interesse a partir de diferentes localizações quando uma solicitação não é atendida; quanto do produtor, uma vez que não há qualquer relação entre a sua identificação e a localização do conteúdo. É destacada também a possibilidade de recuperação de conteúdo sem necessidade de prévio conhecimento da identidade do nó originador dos dados, por exemplo, a busca realizada por veículos sobre informações de trânsito em uma VANET (ou VDTN). Salienta-se também o suporte à entrega de dados via multicast, uma vez que a maioria das aplicações CCN consiste em dirigir informação a mais de um receptor, como por exemplo, a disseminação de notícias, streaming de vídeo, ou publicações em redes sociais. Por fim, a capacidade de lidar com topologias dinâmicas, intermitentes e de baixa conectividade, uma vez que o armazenamento e a replicação descentralizada de dados na rede melhora a qualidade da comunicação, reduzindo o tempo de acesso a um conteúdo.

Assim, é possível observar que a implementação do conceito de CCN no ambiente sem fio, mais especificamente o das redes tolerantes a atrasos e interrupções, é capaz de trazer inúmeros benefícios para este cenário, de modo que se torne possível superar muitas de suas limitações, melhorando consideravelmente a sua qualidade e o seu funcionamento. Na Seção a seguir, apresentaremos brevemente alguns trabalhos da literatura que se propõem a empregar CCN conjuntamente com redes sem fio.

## 2.3 CCN empregada à rede sem fio

A literatura apresenta variadas abordagens para tratar o emprego de CCN em ambientes sem fio. Cada uma delas lida com as limitações do cenário e com o funcionamento de redes centradas em conteúdo de uma maneira específica, seja pelo uso de agentes, pela posição geográfica dos nós, relacionamentos sociais, etc. Vale destacar, porém, o fato de que os trabalhos mencionados nesta Seção tiveram suas análises de desempenho realizadas sob diferentes cenários e foram comparados com diferentes protocolos.

No que diz respeito ao emprego de CCN em ambiente de redes sem fio sem infraestrutura e de múltiplos saltos, alguns trabalhos recentes da literatura merecem destaque, tais como [12], [25] e [26]. O primeiro propõe o *Neighborhood-Aware Interest Forwarding* (NAIF), um protocolo para encaminhamento de pacotes em MANETs de dados nomeados (do inglês, *Named Data MANETs* - NDM), no qual a elegibilidade de um nó como retransmissor de interesses é decidida a partir da análise de sua taxa de recuperação de conteúdo para o prefixo buscado e da sua distância para o consumidor. Avaliando-se que um nó possua uma baixa taxa de recuperação de conteúdo para aquele prefixo ou que ele se encontra distante do requisitante, o interesse recebido é descartado localmente e não é retransmitido. Os resultados obtidos mostram que o NAIF, assim como o encaminhador de redes de dados nomeados (do inglês, *Named Data Networks Forwarding* - NDNF), obtiveram entre 30-60% de ganho em termos de recuperação de dados em cenários multi-consumidor, quando comparados com o modelo de encaminhamento *Listen-First, Broadcast-Later* (LFBL). Além disso, o NAIF apresentou uma redução de aproximadamente 54% no uso da largura de banda disponível em relação ao NDNF.

Já em [25], é elaborado um mecanismo proativo de atualização periódica de informações sobre nós da rede, e outro que utiliza a estratégia de seleção de nós retransmissores para a recuperação confiável de conteúdo baseada na classificação destes nós, onde se diferem os considerados elegíveis e os reservas. A referida classificação e escolha de retransmissores se baseia na distância em número de saltos de determinado nó até o destino

e qualidade do enlace. A proposta atinge 50% de ganho em taxa de trasferência quando comparada com o método básico de encaminhamento oportunista por *broadcast*. Por sua vez, em [26], são comparadas as estratégias de encaminhamento *broadcast* e *unicast* em redes CCN *ad-hoc* móveis. Além disso, é proposta uma abordagem, baseada em *unicast*, de restauração de caminhos de entrega que sofreram quedas de conectividade. O trabalho analisa o desempenho das estratégias em termos de taxa de entrega e eficiência energética: mostra que a entrega de dados por meio de *unicast* resulta em melhores taxas de entrega, a menos que haja demasiadas mudanças de topologia; e maior eficiência energética em comparação com o encaminhamento em *broadcast*.

Na abordagem de [13], voltada para um ambiente de computação em nuvem móvel (do inglês, *Mobile Cloud Computing* - MCC), o remetente de um pacote de interesse divide a área do seu raio de transmssão em quadrantes e dissemina, via *broadcast*, estes pacotes a todos os nós vizinhos cuja distância seja de um único salto. Estes, por sua vez, respondem com pacotes do tipo *ACK*, informando seu próprio identificador e localização geográfica. Por fim, o remetente do interesse seleciona como retransmissores os nós mais distantes de quem obteve resposta em cada quadrante. O trabalho apresentou uma redução no custo total em termos de sobrecarga de tráfego para recuperação de conteúdo no ambiente proposto, quando comparado com o esquema de encaminhamento probabilístico e com o encaminhamento "ingênuo"(ou "cego"), isto é, o mecanismo de encaminhamento padrão de CCN, via *broadcast*, além de reduzir o número de cópias duplicadas de interesses obtendo uma maior taxa de satisfação de requisições.

No que se refere a utilização do conceito de CCN no contexto de DTN, outros trabalhos merecem destaque, como [27], [28], [29] e [30]. No primeiro, é proposto o CEDO, um algoritmo para disseminação e entrega de conteúdo em redes tolerantes a atraso e interrupções, no qual são definidos três componentes para o seu funcionamento: um esquema de otimização de armazenamento de conteúdo que visa maximizar a taxa de entrega de dados a longo prazo na rede; um esquema de descarte e replicação de conteúdos, baseado em uma função de utilidade calculada localmente para cada um deles; e uma formulação de estimativas locais sobre conteúdos, de modo que seja possível obter valores confiáveis nas utilidades calculadas. Como principal resultado, quando comparado com a política de descarte de pacotes menos recentemente usados (do inglês, *Least Recently Used* - LRU), o CEDO atingiu um desempenho até 72% melhor em termos de taxa de entrega de conteúdo.

Em [28] é apresentada uma proposta baseada em agente, na qual um nó requisitante, por meio de um processo de sondagem e comunicação *handshake* de três vias, encontra nós agentes em potencial (de acordo com informações como histórico de locais visitados) e delega a um deles a função de descoberta e recuperação de conteúdo em uma rede

CCN oportunista. Os resultados do trabalho apontam que um intervalo de sondagem de 30s apresenta o melhor desempenho no cenário proposto em vista que o número de notificações (mensagens que agentes enviam aos requisitantes para indicar que determinado conteúdo foi encontrado e está pronto para ser recuperado) pode ser diminuído em até 80%.

Em [29], é proposto o STCR (*Social-Tie based Content Retrieval*), que determina o grau de centralidade dos nós da rede, por meio do cálculo de relacionamento social, e os agrupa em diferentes *clusters* (hierarquia social), utilizando o algoritmo de *clustering K-means*, de modo que nós de centralidade social semelhante pertençam ao mesmo agrupamento. Neste contexto, anúncios de nós provedores de conteúdo e pacotes de interesse convergem para os nós de maior centralidade social. Já o envio destes pacotes (de um nó de alta hierarquia social até um provedor de conteúdo), bem como os de dados (do provedor ao requisitante) é realizado através de nós que possuam alto relacionamento social com o destinatário do pacote em questão. A proposta apresenta ganhos em termos de atraso, custo e taxa de entrega em relação aos protocolos *Epidemic, BubbleRap e PeopleRank*. Por outro lado, o STCR, por seccionar o processo de busca por dados em duas etapas (enviar intererres primeiramente para os nós de mais alta centralidade e só então enviá-los para os provedores de conteúdo), sugere a existência de um tempo elevado para a recuperação de dados em comparação com outros mecanismos para CCN aplicada no cenário DTN, como é verificado em [30].

Por fim, baseado no protocolo *PROPHET*, de roteamento em DTN, em [30] é proposto o PIFP (*Probabilistic Interest Forwarding Protocol*), um protocolo probabilístico de encaminhamento de pacotes de interesse em DTN, a partir do qual os nós calculam e utilizam a probabilidade de encontrarem determinado conteúdo para selecionar retransmissores de interesses. Isto é, nós requisitantes de conteúdo apenas enviam interesses para outros nós que possuam uma probabilidade superior a sua. Já o envio de pacotes de dados possui duas variantes: no PIFP, este envio considera apenas o caminho reverso ao envio dos interesses, utilizando as informações da PIT dos nós (*bread crumbs* [4]) para traçar o caminho de entrega dos dados; já no PIFP *Proative*, os dados são enviados a qualquer nó da rede (*broadcasting*), o que aumenta as chances de entrega dos dados, mas onera a rede em termos de sobrecarga. O trabalho apresenta ganhos em relação à taxa de satisfação (entrega de dados aos requisitantes) e atraso de descoberta e recuperação em relação ao STCR. No entanto, por convocar sempre os nós com maiores probabilidades, o PIFP negligencia a competência de outros nós em encontrar o conteúdo buscado, oferecendo uma seleção não balanceada de retransmissores.

Uma análise mais acurada dos trabalhos supracitados nos permite levantar alguns ques-

tionamentos acerca das políticas e estratégias adotadas em suas abordagens. Na seção a seguir, apontaremos desvantagens resultantes de sua adoção em CCDTN.

## 2.4 Elegibilidade de retransmissores em CCDTN

De acordo com as abordagens existentes na literatura, observamos que alguns trabalhos analisados tendem a fazer uso de políticas e estratégias que, possivelmente, prejudicam a rede em algum aspecto. Dentre elas, destacamos a definição de um único critério de seleção de nós para torná-los retransmissores de pacotes. Este funcionamento, quando não permite que todos os nós enquadrados na sua avaliação sejam escolhidos, seleciona aqueles mais bem posicionados no ranqueamento dos nós que se encaixam em tal critério. Neste caso, observa-se que o mesmo conjunto de nós é quase sempre eleito para a tarefa de encaminhamento de pacotes, visto que, a cada intervalo de tempo, suas propriedades os colocam em melhores posições no referido ranqueamento, justamente por eles executarem frequentemente a tarefa que lhes é designada, a exemplo da seleção realizada em [12], pela análise da taxa de recuperação de determinado conteúdo por parte de um nó; ou a realizada em [30], pela preferência por nós com a maior probabilidade de entrar em contato com os dados. Ou seja, em ambos os casos, quanto mais um nó entregar determinado conteúdo, maior será a sua taxa de recuperação e probabilidade de encontro com o mesmo, dando a tal nó maiores chances de se tornar elegível no respectivo processo de seleção. Outro exemplo desta escolha de nós pouco distribuída é a realizada em [29], no qual claramente são selecionados os nós de maior centralidade social para a execução de dois dos três tipos de comunicação possíveis.

Esta escolha frequente do mesmo conjunto de nós acarreta, no entanto, uma ”distribuição injusta de carga”, ou seja, faz com que a maior parte do tráfego da rede seja realizada apenas por uma pequena parcela de nós, sobrecarregando-os e exaurindo rapidamente seus recursos, entre os quais energia e capacidade de armazenamento. Por sua vez, este esgotamento leva à saída destes nós da rede, fazendo com que a mesma tenha de se reorganizar no sentido de recalcular quais são agora os que melhor se encaixam no critério de seleção estabelecido. Em [31], este problema é definido como a “escolha de nós preferidos”.

Outra observação realizada sobre as propostas da literatura é a quantidade de processamento exigida para que, efetivamente, a seleção de restransmissores seja realizada e as trocas de mensagens aconteçam. Isto é, a exemplo do processo de seleção de retransmissores apresentado em [29], no qual os nós, sempre que desejam requisitar determinado conteúdo, previamente (i) avaliam o vetor de relacionamentos sociais de outros nós, dis-

tribuído na rede, e a centralidade social calculada para cada um deles e (ii) executam o algoritmo de *clustering K-means* para verificar se o contato em questão pertence ao *cluster* de maior nível social. Uma elevada demanda de processamento de informações durante a tomada de decisão pode prejudicar a troca de mensagens na rede, uma vez que, em DTN, o contato entre os dispositivos, além de intermitente, pode não ser duradouro, de modo que, se a seleção do retransmissor resulta em demasiado processamento, a oportunidade de contato e envio de pacotes pode ser perdida, justamente pela curta duração destas conexões [2]. Como exemplo desta mesma observação, destacamos ainda o processo de comunicação de três vias realizado em [28] e em [13], para que a seleção de um nó como retrasmissor seja concluída.

Desta forma, fica claro que, embora o emprego da arquitetura CCN no contexto das redes DTN possa agregar diversos benefícios para este paradigma, conforme exposto ao longo deste Capítulo, ainda se faz necessário elaborar estratégias que sejam capazes de fazer o melhor uso desta combinação, otimizando o seu funcionamento, isto é, de modo que a sua própria definição não prejudique a rede em algum aspecto.

# 3. O protocolo proposto

Visando superar os desafios da combinação de DTN e CCN e as questões levantadas no Capítulo anterior, propomos o DIRESC, um protocolo para descoberta e recuperação de conteúdo baseado em cooperação social (DIscovery and REtrieval based on Social Cooperation), que tem por objetivo principal reduzir o atraso de descoberta e recuperação de dados em redes centradas em conteúdo e tolerantes a atraso (CCDTN). Diferentemente de outras abordagens propostas na literatura, nas quais os conteúdos são alcançados através da convergência de mensagens de interesse por meio do que [31] e [32] chamam de "escolha de nós preferidos", uma vez que definem um único critério para determinar a competência de um nó em entregar um pacote, o DIRESC propõe uma abordagem distribuída, evitando assim tal comportamento indesejável, resultante do funcionamento de protocolos oportunistas com base social. A definição de três possibilidades de envio de requisições e outras três para entrega de dados evita o comportamento em *broadcast*, típico de CCN, mas torna a proposta mais tolerante em termos de critérios de seleção de nós retransmissores.

## 3.1 Relacionamento social

Para o funcionamento do protocolo DIRESC, os nós levam em consideração o fator de relacionamento social existente entre eles. Essa medida expressa o quão próximo um par de nós pode estar em termos de contato social. Isto é, a partir da avaliação do histórico de encontro entre eles, pode-se calcular a frequência com a qual tais encontros ocorrem, bem como a distância temporal entre eles, de modo que se torne possível predizer a ocorrência de contatos futuros, bem como inferir que um par de nós específico possui um alto valor de relacionamento social entre si devido à alta frequência de seus encontros em um pequeno intervalo de tempo.

Assim, inspirado pelo protocolo *SimBetTS* [22], o fator de relacionamento social $R_i(j)$,

entre o par de nós $i$ e $j$, leva em consideração duas medidas fundamentais: (i) frequência, medida pela quantidade de encontros no histórico, e (ii) recência, calculada pelos intervalos de tempo decorrido entre dois encontros. O valor de $R_i(j)$, definido por [29], é apresentado em 3.1.

$$R_i(j) = \sum_{k=1}^{N} E(t_{atual} - t_{jk}) \tag{3.1}$$

Observa-se em 3.1 que o fator de relacionamento social $R_i(j)$ é igual ao somatório da função $E(x)$, que é parametrizada com o intervalo de tempo decorrido desde um encontro no passado ($k$) até o tempo atual ($t_{atual}$). Assim, $t_{j1}, t_{j2}, t_{j3}, ...t_{jN}$ são os tempos de encontro entre $i$ e $j$ armazenados no vetor de encontros $E_N(i)$, sendo $t_{j1} < t_{j2} < t_{j3} < ... < t_{jN}$ e $N$ o total de encontros ocorridos entre o nó $i$ e o $j$ até o instante $t_{atual}$.

Assim, a influência de cada encontro entre os nós $i$ e $j$ é determinado pela função de encontros $E(x)$, que calcula o impacto gerado em $R_i(j)$ pelo quão recente esses encontros ocorreram, onde $x$ é o intervalo de tempo entre um encontro no passado e o tempo atual. A função $E(x)$, também definida por [29], é apresentada em 3.2.

$$E(x) = (\frac{1}{2})^{\lambda x}, \text{onde } \lambda = e^{-4} \tag{3.2}$$

É explicado em [29] que o $\lambda$ na função $E(x)$ funciona como um parâmetro de controle que varia entre $0 \leq \lambda \leq 1$ e que permite um balanço adequado entre frequência e recência, portanto, a variação de $\lambda$ neste intervalo, aproximando-se de 0 ou de 1, pode favorecer a apenas uma das métricas em questão, respectivamente. Sendo assim, explica [29], o valor de tal parâmetro é definido como $\lambda = e^{-4}$, de modo que não se dê maior importância a apenas uma daquelas medidas.

O fator de relacionamento social $R_i(j)$ é utilizado pelo DIRESC como princípio fundamental, uma vez que, a partir do seu valor calculado, decisões de transmissão são tomadas pelo protocolo. No entanto, diferentemente de [29], que utiliza essa medida para calcular a centralidade social dos nós e em seguida agrupá-los segundo a sua similaridade em relação ao valor obtido, o DIRESC desconsidera a informação sobre relacionamento global na rede (centralidade dos nós), e utiliza $R_i(j)$ apenas para determinar o valor do relacionamento social entre eles de forma local. Isto é, o vetor de relacionamentos sociais de $i$ com outros nós é mantido localmente em $i$, e não distribuído na rede para que tais informações sejam globais. Utilizamos essa abordagem diferenciada uma vez que o uso da métrica de centralidade em [29], além de ocasionar o problema da escolha de "nós prefe-

ridos", secciona o processo de descoberta de conteúdo, elevando o atraso de recuperação dos dados.

Assim, além do valor de $R_i(j)$, calcula-se também a média aritmética das relações sociais de $i$ armazenadas no vetor de relacionamentos $S_M(i)$. O DIRESC faz uso de uma medida denominada $R_{medio}(i)$ (ver Equação 3.4), que é uma variante da média supracitada, para verificar a amizade entre os nós. Isto é, para determinar que um par de nós, no caso $i$ e $j$, possuem um alto relacionamento social, o valor de $R_i(j)$ precisa ser maior ou igual a $R_{medio}(i)$. Caso essa premissa (Equação 3.3) seja verdadeira, os dois nós são considerados "amigos"[1] e podem agir entre si de modo socialmento cooperativo.

$$R_i(j) \geq R_{medio}(i) \tag{3.3}$$

O cálculo de $R_{medio}(i)$ envolve a própria média aritmética das relações sociais de $i$, como mencionado anteriormente, e, portanto, o vetor de relacionamentos deste nó, $S_M(i)$, que registra os valores de $R_i(h)$, para $h = 1, 2, ..., M$, onde $M$ é o total de nós com os quais $i$ teve contato até o instante da conexão com $j$. Além disso, multiplica-se a essa média o fator $\Omega$, denominado "fator de densidade", que permitirá a variação de $R_{medio}(i)$ em relação à referida média. Esse multiplicador varia de acordo com a densidade do cenário e é utilizado para restringir a quantidade de amigos de determinado nó em ambientes com elevada concentração de nós, bem como consentir a existência de mais vínculos de amizade para o mesmo nó em cenários mais esparsos. Para tanto, diferentes valores podem ser atribuídos a $\Omega$, de modo que as chances da atuação em modo socialmente cooperativo são reduzidas quando $\Omega$ assume um valor elevado, enquanto valores menores aumentam essas chances. Assim, o valor de $R_{medio}(i)$ é calculado da seguinte forma, como mostrado em 3.4.

$$R_{medio}(i) = \Omega \frac{1}{M} \sum_{h=1}^{M} R_i(h) \tag{3.4}$$

O fator $R_i(j)$ e $R_{medio}(i)$ são periodicamente atualizados pelo protocolo de modo que possam ser utilizados para tomadas de decisão a qualquer momento em que seja necessário. Assim, sempre que dois nós entram em contato, esses valores são calculados e armazenados para uso futuro.

A constatação de que dois nós são amigos é de fundamental importância para que

[1] A amizade entre dois indivíduos é uma característica social a partir da qual se compreende que ambos se encontram com determinada frequência e recência em um dado intervalo de tempo.

sejam realizadas as tarefas de anúncio, descoberta e recuperação de conteúdo de modo socialmente cooperativo na rede. Para melhor compreensão dessas tarefas, elas foram enquadradas em três fases, ou seja, momentos nos quais os nós tomam decisões em relação ao que fazer com informações acerca de conteúdos, pacotes de interesse, ou pacotes de dados. São elas: (i) anúncio, (ii) descoberta e (iii) recuperação.

## 3.2 Fase de anúncio

Como explicado em [4] e apresentado no Capítulo anterior, um conceito comum em CCN é o anúncio de prefixos que podem ser atendidos por parte de nós provedores. Isto é, aqueles que executam aplicações que geram pacotes de dados disseminam anúncios para outros nós, informando que são capazes de atender requisições para os prefixos correspondentes a tais dados.

Os anúncios de prefixos, quando disseminados em *broadcasting*, ou seja, indiscriminadamente na rede, elevam a quantidade de nós com informações na FIB sobre tais prefixos e capazes de retransmitir requisições de terceiros. Isto é, elevam-se as chances de que o determinado conteúdo seja descoberto na rede devido às múltiplas possibilidades de que uma requisição alcance um provedor do conteúdo buscado. Em um ambiente conectado, cenário para o qual CCN foi idealizada, esse modo de anúncio é favorável, uma vez que a tarefa de encaminhamento de interesses, ainda que delegada a muitos nós, não representa um risco de exaustão de recursos. De modo contrário, em um ambiente desconectado, como em DTN, onde os nós apresentam grandes limitações, elevar o número de retransmissores de mensagens pode ocasionar prejuízos a rede no que se refere ao consumo desnecessário de recursos como energia e capacidade de armazenamento.

Sendo assim, para um cenário CCDTN, faz-se necessário definir uma regra para o anúncio de prefixos, de modo que apenas nós específicos da rede possuam informações na FIB e possam participar dos processos de requisição, evitando que todos os nós da rede sejam elegíveis para a retransmissão de interesses até o provedor, o que geraria as desvantagens de um cenário epidêmico.

Desse modo, a fase de anúncio compreende a disseminação de informações sobre conteúdos, ou seja, anúncios de prefixos por parte de provedores que podem satisfazer requisições. Nela ocorrem dois tipos de anúncios: no primeiro, nós produtores (aqueles que geram pacotes de dados) informam para nós com os quais têm alto relacionamento social quais requisições de dados podem atender, por meio do anúncio de prefixos; no segundo, os nós que receberam anúncios diretamente dos provedores, informam aos seus próprios

amigos tal recebimento, de modo que se distribua para estes o comunicado de que aqueles conhecem o produtor e, portanto, também podem atender requisições, ainda que de modo indireto. Assim, delimitam-se agrupamentos, que por vezes chamaremos de *clusters*, embora não seja empregado nenhum algoritmo de *clustering* por parte do protocolo. Esses *clusters* são compostos por nós que receberam quaisquer comunicados acerca de conteúdos e que se tornaram capazes de auxiliar na entrega das solicitações. Portanto, nesta etapa, assumindo que dois nós sejam amigos, segue-se as seguintes trocas de anúncios:

- **Anúncio direto**: caso o nó i seja um produtor capaz de prover conteúdos cujos prefixos não estejam registrados na FIB de j, ou a FIB de j possui tais prefixos, mas sem o devido mapeamento para i, registra-se neste componente de j que o nó i é capaz de atender requisições para tais conteúdos.
- **Anúncio transitivo**: caso a FIB do nó i possua prefixos que não existem na FIB de j, esta é atualizada com os registros faltantes de modo que i seja mapeado como um nó de próximo salto para o envio de interesses.

Deste modo, uma vez que existe alto relacionamento social entre nós que receberam tais anúncios, criam-se agrupamentos capazes de responder às solicitações de forma ágil, tendo em vista o elevado nível de contatos entre os nós que os compõem. O intuito de um agrupamento é, portanto, reduzir o tempo de descoberta de conteúdos a partir do momento em que os pacotes de interesse o atingem. Além disso, garantimos que sejam formados agrupamentos distindos para diferentes conteúdo de acordo com as relações sociais dos

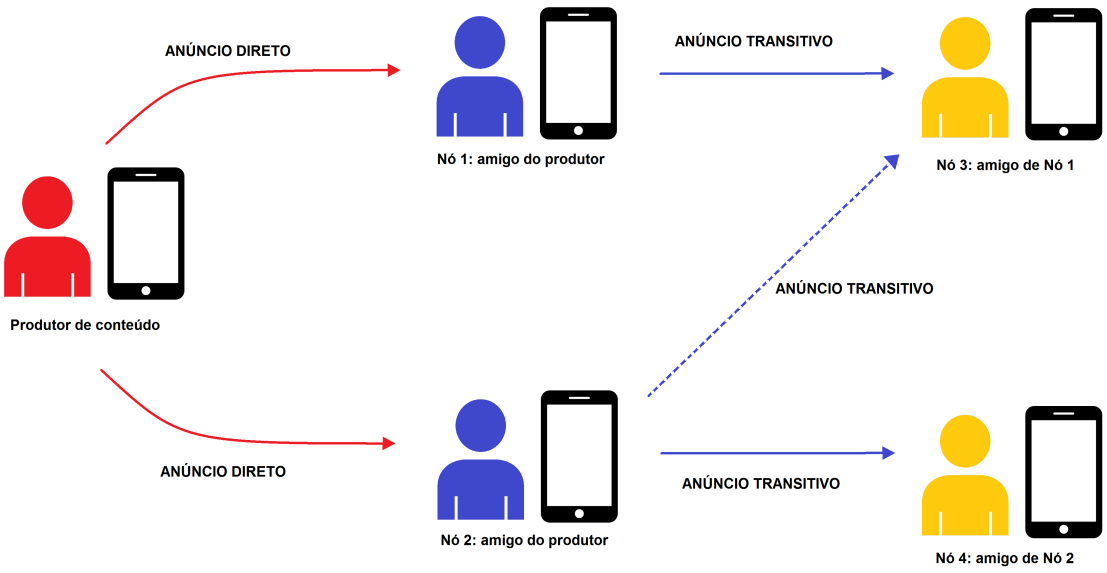


Figura 3.1: Anúncios direto e transitivo

provedores. Ou seja, cada produtor dispõe de um *cluster* que se forma ao seu redor a partir dos seus relacionamentos. A Figura 3.1 sumariza os tipos de anúncios no DIRESC.

Observa-se na Figura 3.1 que o produtor de conteúdo (em vermelho) efetua o anúncio direto a todo nó com quem possui alto relacionamento social (ver Equação 3.3), ou seja, seus amigos (em azul). Estes, por sua vez, realizam o anúncio transitivo com os seus próprios amigos (em amarelo), de modo que todos eles compõem o *cluster* de nós que possuem informações na FIB sobre prefixos de conteúdos anunciados e são capazes de responder às requisições por terem facilidade em encontrar o referido produtor, devido ao seu alto relacionamento social.

A linha pontilhada na Figura 3.1 representa uma possibilidade: supondo que o Nó 3 (em amarelo) receba um anúncio transitivo do Nó 2 (em azul) em razão de uma amizade existente entre eles, o Nó 2 passa a ser mapeado na FIB do Nó 3 como um nó de próximo salto para o envio de requisições dentro do mesmo registro do prefixo anunciado pelo Nó 1.

## 3.3 Fase de descoberta

Por sua vez, a fase de descoberta diz respeito à requisição de conteúdo. Nela, definimos como acontece a tomada de decisão para o envio de pacotes de interesse. Com o intuito de evitarmos que o protocolo envie mensagens a todos os contatos possíveis na rede e também não escolha apenas um grupo específico de nós por se enquadrarem em um único critério de seleção estabelecido, mas seja, de certa forma, tolerante quanto ao envio de tais mensagens, são definidos três critérios para o envio de interesses no DIRESC. Deste modo, nosso protocolo evita tanto o comportamemto epidêmico, cuja inundação de pacotes na rede é uma de suas principais características, quanto a escolha de "nós preferidos", o que ocasionaria um envio de mensagens de forma desigual na rede, tendo em vista que certos nós se encaixariam mais facilmente do que outros no critério de elegibilidade do protocolo e, muito provavelmente, seriam sempre escolhidos para a tarefa de encaminhamento.

Portanto, o DIRESC utiliza esses três critérios definidos como premissas de avaliação para determinar quais os nós competentes para encaminhar interesses de modo que a rede não seja sobrecarregada e que não se gere uma distribuição injusta de carga. Considerando que as requisições podem ser feitas com foco em conteúdos distintos e nem sempre pelos mesmos nós, torna-se vantajoso que o método de escolha leve em consideração tais diferenças e selecione nós mais propensos em responder a busca conforme essas suposições. Assim, para o envio de interesses, o DIRESC avalia se o nó encontrado tem chances de

localizar o conteúdo ou se pode agir de modo socialmente cooperativo em favor do requisitante, de modo a suprir a hipótese de vantagem levantada, operando de modo oposto aos protocolos que julgam a capacidade de descoberta de conteúdo baseando-se exclusivamente na centralidade dos nós, independentemente da requisição.

Neste contexto, e ainda projetando uma maneira para que não sejam encaminhadas mensagens de interesse de forma indiscriminada na rede. O protocolo DIRESC define também três componentes que garantem o controle na disseminação de mensagens de interesse. São eles:

- **Limitador de submissões (μ):** primeiramente, define-se a quantidade máxima de submissões de um mesmo interesse por parte de um nó. Ou seja, supondo que o nó $i$ esteja interessado no conteúdo $c$, esse mesmo nó poderá enviar, para determinados grupos de nós, apenas uma quantidade $\boldsymbol{\mu}$ de tal requisição. Deste modo, limitamos a quantidade de submissões feitas por um nó para certos grupos da rede. Este limitador é comparado aos contadores de submissões de determinado interesse, definidos a seguir.

- **Contador de submissões para o *cluster* (α):** o contador $\boldsymbol{\alpha}$ se designa a registrar localmente a quantidade de submissões de um mesmo interesse, feitas por um nó, para o grupo de nós que possui informações na FIB sobre tal conteúdo (esse grupo é criado na fase de anúncio, ver Seção 3.2). Assim, a cada submissão de um interesse no conteúdo $c$, realizado pelo nó $i$ para qualquer outro pertencente àquele *cluster*, o contador $\boldsymbol{\alpha}$ é incrementado em uma unidade.

- **Contador de submissões para os amigos (β):** por sua vez, o contador $\boldsymbol{\beta}$ registra localmente a quantidade de submissões de um mesmo interesse, feitas por um nó, para os seus amigos, ou seja, nós com os quais tal solicitante tem um alto relacionamento social (ver Equação 3.3). Assim, a cada submissão de um interesse no conteúdo $c$, realizado pelo nó $i$ para qualquer amigo, o contador $\boldsymbol{\beta}$ é incrementado em uma unidade.

Assim, na fase de descoberta, nosso protocolo garante que um nó requisitante (primário ou intermediário) apenas possa enviar interesses correspondentes a determinado conteúdo, se número de submissões feitas para cada grupo for menor que o limite permitido ($\boldsymbol{\mu}$), evitanto que se eleve o número de replicações desses pacotes exacerbadamente. Desta forma, os nós transmitem pacotes de interesse caso quaisquer das premissas a seguir seja verdadeira, como observado na Figura 3.2.

1. **Contato com um provedor:** o nó encontrado é um provedor do conteúdo buscado, ou seja, executa uma aplicação que gera os dados requisitados, ou possui uma cópia deles armazenada em cache (CS). Ao efetuar a requisição diretamente para o provedor, espera-se um atraso reduzido na entrega do conteúdo para o requisitante. Além disso, considerando que uma solicitação desse tipo não envolve nós retrasmissores de pacotes, não é contabilizado o envio desse interesse em quaisquer dos contadores definidos anteriormente.

2. **Contato com o *cluster*:** o nó encontrado pertence ao agrupamento de nós formado na fase de anúncio (ver Seção 3.2), e que possui informações na FIB sobre o conteúdo requisitado; e o número de submissões dessa requisição feitas para esse grupo ($\boldsymbol{\alpha}$) é menor que o valor máximo permitido ($\boldsymbol{\mu}$). Nesse caso, assumindo que esse processo envolve retransmissores, registra-se na PIT uma nova submissão do interesse pelo critério de agrupamento, ou seja, $\boldsymbol{\alpha}$ é incrementado em uma unidade.

3. **Contato com um amigo:** há um alto relacionamento social com o nó encontrado e o número de submissões de determinada requisição feitas para o grupo de amigos ($\boldsymbol{\beta}$) é menor que o valor máximo permitido ($\boldsymbol{\mu}$). Nesse caso, assim como no anterior, assumindo que o processo envolve retransmissores, registra-se na PIT uma nova submissão do interesse, agora pelo critério de amizade, isto é, $\boldsymbol{\beta}$ é incrementado em uma unidade. Esse tipo de submissão aumenta a persistência das requisições para descobrir conteúdo na rede, de modo que os nós considerados amigos do requisitante trabalham de modo socialmente cooperativo, buscando dados em favor deste.

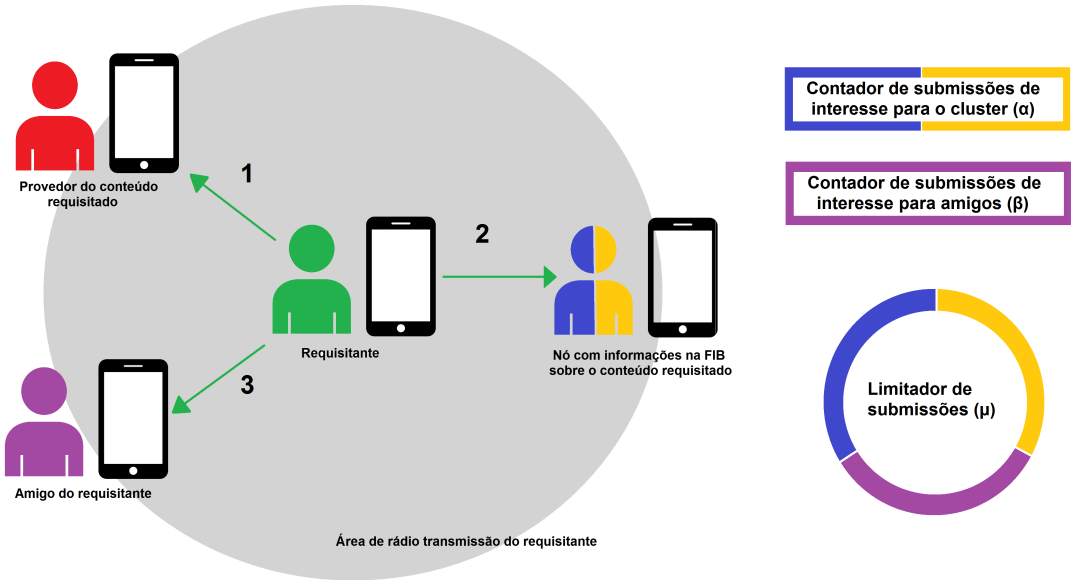


Figura 3.2: Possibilidades de envio de pacotes de interesse

Além disso, dados os critérios supracitados para envio de pacotes de interesse, definimos um esquema de priorização de envio, a partir do qual um nó interessado em determinado conteúdo dará prioridade de envio de solicitações aos nós que se enquadram nos critérios acima descritos, na ordem em que eles se apresentam.

Ao receber um pacote de interesse, um nó avaliará se possui competência para atender tal requisição, ou seja, verificará seu cache (CS) para comparar o prefixo expresso no interesse com aqueles que rotulam os conteúdos que esse nó armazena. Caso haja uma correspondência de valores, o pacote de dados é enviado de acordo com as premissas a serem definidas na Seção 3.4, que descreve a fase de recuperação. No entanto, se não for capaz de atender a tal requisição, o nó fará duas verificações para decidir se participará ou não do processo de busca: primeiramente, checará se possui um alto relacionamento social com o nó que lhe transmitiu o interesse (ver Equação 3.3) ou se há em sua FIB um registro correspondente ao prefixo buscado. Se pelo menos uma dessas condições for verdadeira, o nó atualizará a sua PIT com informações sobre o requisitante, criando um novo registro e mantendo o pacote de interesse para retransmissão, ou inserindo tais informações no registro existente e descartando o tal interesse. Adotamos um funcionamento diferente do tradicionalmente usado CCN, pois, no contexto CCDTN, a condição de existência de requisições correspondentes na PIT, logo após a avaliação da CS, não garantiria (i) que o encaminhador pudesse encontrar rapidamente um provedor do conteúdo; ou (ii) que o requisitante e tal encaminhador pudessem se encontrar novamente na rede. O processo descrito acima é apresentado no fluxograma da Figura 3.3.

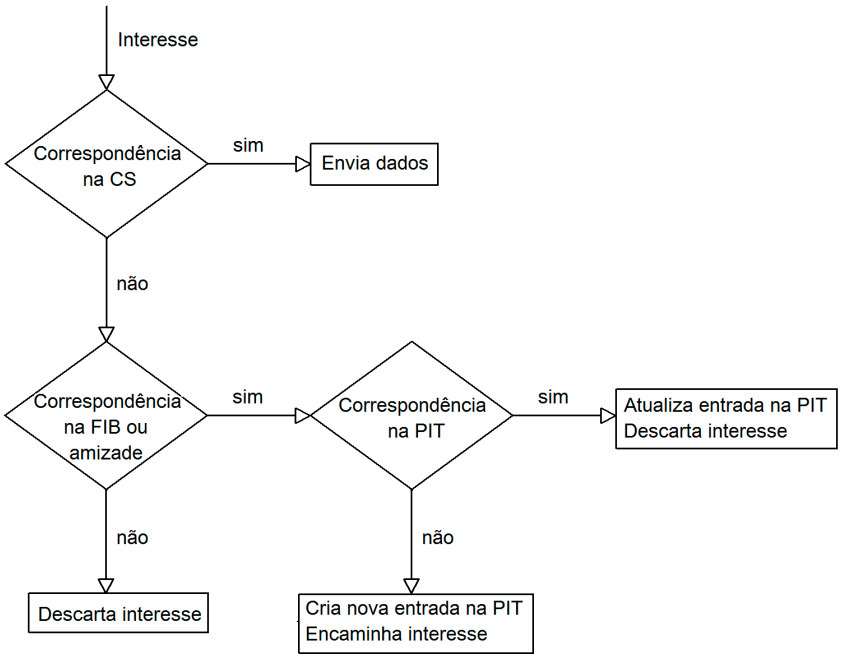


Figura 3.3: Processamento de pacotes de interesse no DIRESC

## 3.4 Fase de recuperação

Assim como na fase de descoberta, na fase de recuperação selecionamos nós que se enquadrem em três critérios para envio de pacotes, evitando que estes sejam transmitidos de forma epidêmica, e dando tolerância para o envio dos mesmos, de modo que se aumente suas chances de entrega. Essa fase se refere à entrega de dados, ou seja, quando um nó provedor do conteúdo requisitado responde a uma solicitação recebida. Assim, são levados em consideração quaisquer nós que estejam interessados em receber determinado pacote de dados, bem como nós que possam agir de forma socialmente cooperativa em favor do real requisitante de tal conteúdo. Portanto, ao receber um pacote de interesse em determinado prefixo, um nó provedor capaz de atender tal solicitação enviará os pacotes de dados correspondentes para outros nós de acordo com os seguintes critérios, como observado na Figura 3.4.

1. **Contato com o requisitante:** para todo nó que lhe fez tal solicitação, ou seja, todo aquele que lhe enviou um pacote de interesse requisitando os dados que ele possui. Esse critério representa o caso mais trivial de entrega de dados e está esquematizado na Figura 3.3. Ou seja, o nó provedor verifica se há uma correspondência do prefixo buscado em sua cache (CS) e obtém um resultado positivo. Após a constatação, prepara e envia o pacote de dados ao nó que lhe enviou o interesse.

2. **Contato com *qualquer* requisitante:** para todo nó que possua um registro na PIT correspondente a uma busca para qualquer prefixo de conteúdo que o provedor é capaz de atender. Nesse caso, o nó em questão é um solicitante, ainda que não tenha enviado um pacote de interesse para esse provedor, mas esteja aguardando receber os dados por meio de outro nó.

3. **Contato com um amigo do requisitante:** para todo nó amigo (ver Equação 3.3) do requisitante primário dos dados (o nó que inseriu na rede o pacote de interesse recebido pelo provedor). Deste modo, mais uma vez, utiliza-se o fator de relacionamento social para que os nós amigos possam agir de modo socialmente cooperativo; nesse caso, para auxiliar a recuperação dos dados.

Da mesma forma que na fase de descoberta, utiliza-se um esquema de priorização de envio no qual um nó provedor de determinado conteúdo dará prioridade de envio de dados aos nós que se enquadram nos critérios acima descritos, na ordem em que eles se apresentam.

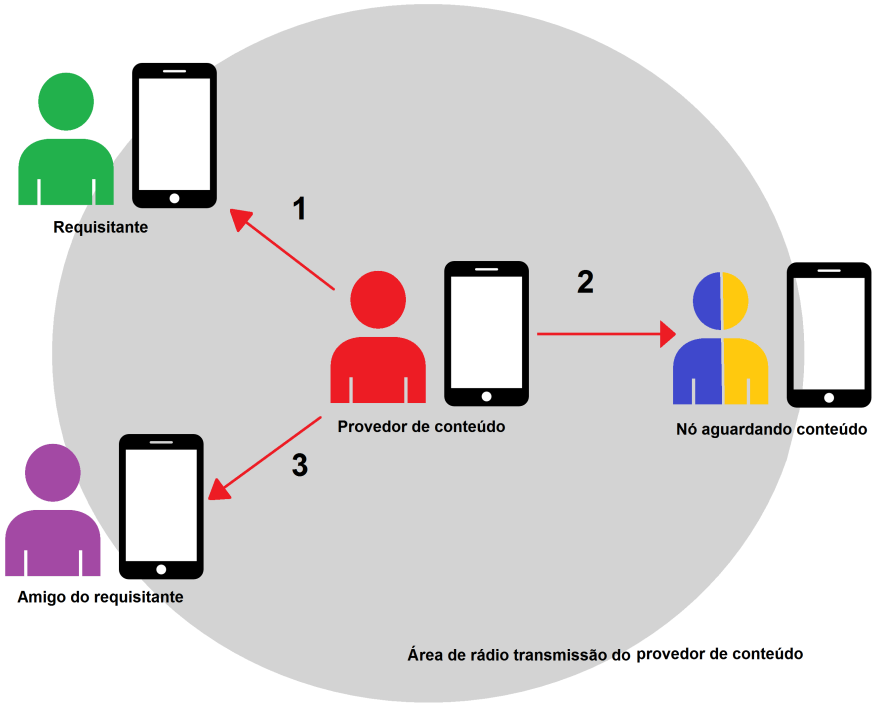


Figura 3.4: Possibilidades de envio de pacotes de dados

Por fim, ao receber um pacote de dados, um nó primeiramente avalia se existe um registro de busca em sua PIT correspondente ao pacote recebido. Caso positivo, o pacote é direcionado à interface solicitante, ou seja, se o nó for o requisitante primário daquele conteúdo, o pacote é encaminhado para a aplicação requisitante e armazenado em cache; caso o nó seja apenas um intermediário da requisição e entrega, o pacote é apenas mantido em cache, visando a retransmissão em um encontro com o requisitante primário dos dados ou com outro intermediário que se enquadre em um dos critérios definidos anteriormente nessa Fase. Caso não haja um registro correspondente na PIT, o mesmo pacote de dados é descartado. O processo descrito acima é apresentado no fluxograma da Figura 3.5.

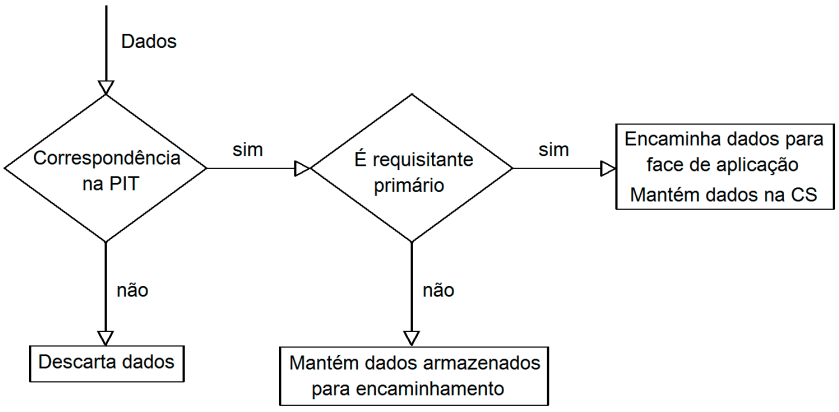


Figura 3.5: Processamento de pacotes de dados no DIRESC

# 4. Algoritmos, complexidade e estruturas

Neste capítulo, apresentaremos os algoritmos referentes ao protocolo DIRESC e suas análises de complexidade, a fim de apronfundarmos o estudo realizado sobre o protocolo proposto, de modo que seja possível detalhar mais o seu funcionamento, bem como o quanto ele demanda em termos de processamento nos nós da rede. Assim, subdividiremos esse capítulo em seções referentes a cada fase anteriormente definida, para que possamos analisá-las separadamente.

## 4.1 Análise do cálculo de $R_i(j)$ e $R_{medio}$

O Algoritmo 1 apresenta o cálculo do fator de relacionamento social entre os nós $i$ e $j$ e da média de relacionamentos sociais de $i$.

As entradas desse Algoritmo são: o próprio nó $i$; o nó $j$; o tempo do contato atual entre $i$ e $j$, $t_{atual}$; o vetor de encontros no passado entre esses mesmos nós, $E_N(i)$; e o vetor de relacionamentos sociais de $i$ com outros nós da rede, $S_M(i)$. A definição das duas últimas entradas se encontra na Seção 3.1.

**Algoritmo 1:** DIRESC: fator e média de relacionamento social

```
   Entrada: i, j, t_atual, vetor E_N(i), vetor S_M(i)
1  início
2      para todo contato de i com j faça
3          adiciona t_atual em E_N(i)
4          (re)calcula-se R_i(j) usando E_N(i) como parâmetro (ver Equação 3.1)
5          substitui R_i(j) em S_M(i)
6          (re)calcula-se R_medio(i) usando S_M(i) (ver Equação 3.4)
7      fim
8  fim
```

A complexidade do Algoritmo 1 é definida a partir das seguintes operações:

- **Linha 3**: a adição de $t_{atual}$ em $E_N(i)$ é uma operação que resulta em uma complexidade constante ($O(1)$), uma vez que compreende a inserção em uma lista existente dentro de um registro de uma estrutura de dados do tipo **tabela *hash***[1], sendo um tipo de operação que resulta em tal complexidade. Além disso, vale ressaltar que, em nosso trabalho, a maioria dessas estruturas são do tipo tabela *hash*, que possui complexidade constante, isto é, igual a $O(1)$, para inserção e busca na maioria dos casos, segundo a literatura.

- **Linha 4**: operação seguinte é calcular (ou recalcular) o valor de $R_i(j)$ a partir dos valores de tempo de encontros armazenados em $E_N(i)$. A complexidade referente a tal cálculo depende do tamanho dessa estrutura de dados, que será percorrido para o cálculo de somatório (ver Equação 3.1), portanto, pode ser definida como $O(E_N(i))$.

- **Linha 5**: a próxima operação do Algoritmo 1 é a substituição do valor de $R_i(j)$ no em $S_M(i)$. Sendo $S_M(i)$ uma estrutura de dados do tipo tabela *hash*, temos que a complexidade dessa substituição é igual a $O(1)$.

- **Linha 6**: por fim, a última operação diz respeito ao cálculo (ou atualização do valor) de $R_{medio}(i)$ utilizando os valores de relacionamentos sociais de $i$ com outros nós, armazenados em $S_M(i)$. A complexidade referente a essa operação depende do tamanho desta estrutura de dados, que será percorrido para o cálculo de somatório (ver Equação 3.4), portanto, pode ser definida como $O(S_M(i))$.

Assumindo que a complexidade total do Algoritmo 1 é definida por um conjunto de operações relizadas sobre as parciais resultantes de cada etapa do mesmo, de modo que aquela é igual à maior entre as verificadas, temos pela Equação 4.1 que a complexidade $C_{R_i(j)}$ do Algoritmo 1 é dada por:

$$\begin{aligned} C_{R_i(j)} &= O(1) + O(E_N(i)) + O(S_M(i)) + O(1) \\ C_{R_i(j)} &= O(E_N(i)) + O(S_M(i)) \\ C_{R_i(j)} &= O(E_N(i) + S_M(i)) \end{aligned} \tag{4.1}$$

Uma vez que se faz necessário armazenar os tempos de encontros de $i$ com todos os outros nós com os quais ele entre em contato, nosso protocolo faz uso de uma tabela *hash* local, denominada tabela de encontros, que mapeia os nós descobertos por $i$ (chave) a um

[1] Uma tabela de dispersão (também conhecida por tabela de espalhamento ou tabela *hash*) é uma estrutura de dados especial, que associa chaves de pesquisa a valores.

vetor $E_N$ que armazena os tempos de seus encontros (valor), como apresentado na Tabela 4.1.

Tabela 4.1: Tabela (*hash*) de encontros (mapeamento entre nós encontrados e $E_{Ns}(i)$)

| **Chave** | **Valor** |
|---|---|
| j = 0 | $E_{N0}(i) = \{t_{0,0}, t_{0,1}, ..., t_{0,N}\}$ |
| j = 1 | $E_{N1}(i) = \{t_{1,0}, t_{1,1}, ..., t_{1,N}\}$ |
| j = 2 | $E_{N2}(i) = \{t_{2,0}, t_{2,1}, ..., t_{2,N}\}$ |
| ... | ... |
| j = M | $E_{NM}(i) = \{t_{M,0}, t_{M,1}, ..., t_{M,N}\}$ |

De modo semelhante, o vetor de relacionamentos ($S_M(i)$) também é implementado por meio de uma tabela *hash*, que mapeia o nó encontrado (chave) ao valor de relacionamento social com $i$ (valor), como apresentado na Tabela 4.2

Tabela 4.2: Tabela (*hash*) de relacionamentos $S_M(i)$

| **Chave** | **Valor** |
|---|---|
| j = 0 | $R_i(0)$ |
| j = 1 | $R_i(1)$ |
| j = 2 | $R_i(2)$ |
| ... | ... |
| j = M | $R_i(M)$ |

## 4.2 Análise da fase de anúncio

O Algoritmo 2 descreve o funcionamento da fase de anúncio do protocolo DIRESC.

Como descrito na Seção 3.2, essa fase compreende a distribuição de informação sobre prefixos de conteúdo para o preenchimento das tabelas FIB dos nós que recebem os anúncios. As entradas do Algoritmo 2 são: o próprio nó $i$; o nó $j$; e os valores previamente atualizados de $R_i(j)$ e $R_{medio}(i)$. A definição das duas últimas entradas se encontra na Seção 3.1.

O Algoritmo 2 apresenta as operações referentes ao anúncio direto (linhas 3-9) e transitivo (linhas 10-17) e sua complexidade é definida a seguir:

- **Linha 2**: a operação de comparação entre os valores já calculados de $R_i(j)$ e $R_{medio}(i)$ tem complexidade $O(1)$.

**Algoritmo 2:** DIRESC: fase de anúncio

```
   Entrada: i, j, R_i(j), R_medio(i)
1  início
2      se R_i(j) ≥ R_medio então
3          se i é produtor então
4              para todo prefixo p na CS de i faça
5                  se p não existe na FIB de j então
6                      Atualiza-se a FIB de j: adiciona-se p mapeado para i
7                  fim
8              fim
9          fim
10         para todo prefixo p na FIB de i faça
11             se p não existe na FIB de j então
12                 Atualiza-se a FIB de j: adiciona-se p mapeado para i
13             fim
14             senão
15                 Atualiza-se o p da FIB de j: incluindo i no mapeamento existente
16             fim
17         fim
18     fim
19 fim
```

- **Linha 3**: a verificação do valor do atributo *booleano* que indica se um nó é produtor ou não também possui complexidade constante ($O(1)$).

- **Linhas 4-8**: a operação que percorre a CS tem complexidade $O(CS)$, ou seja, depende do tamanho dessa estrutura de dados, uma vez que será avaliado cada um de seus registros. A operação de busca na FIB (linha 5), bem como a adição de um novo mapeamento na mesma estrutura (linha 6), possuem complexidade $O(1)$, por se tratar de uma inserção em uma tabela *hash*. Assim, o anúncio direto possui complexidade na ordem de $O(CS)$.

- **Linhas 10-17**: a operação que percorre a FIB possui complexidade $O(FIB)$, isto é, depende do tamanho desse componente, uma vez que será avaliado cada um dos seus registros. A operação de busca na FIB (linha 11), bem como a adição de um novo mapeamento na mesma estrutura (linha 12) ou sua atualização (linha 15), possuem complexidade $O(1)$, por se tratar de uma inserção em uma tabela *hash*. Assim, o anúncio transitivo possui complexidade na ordem de $O(FIB)$.

Assumindo que a complexidade total do Algoritmo 2 é definida por um conjunto de operações relizadas sobre as parciais resultantes de cada etapa do mesmo, de modo que aquela é igual à maior entre as verificadas, temos pela Equação 4.2 que a complexidade

$C_{anuncio}$ do Algoritmo 2 é dada por:

$$\begin{aligned} C_{anuncio} &= O(1 * 1) + O(1 * CS) + O(1 * FIB) \\ C_{anuncio} &= O(CS) + O(FIB) \\ C_{anuncio} &= O(CS + FIB) \end{aligned} \tag{4.2}$$

Os componentes CS e FIB são implementados na forma de tabela *hash*, de modo que prefixos sejam mapeados com (i) os próprios dados; e (ii) nós que podem responder a uma solicitação ou auxiliar no processo de descoberta de conteúdo, como representado nas Tabelas 4.3 e 4.4, respectivamente.

Tabela 4.3: Componente CS da arquitetura CCDTN

| **Chave** | **Valor** |
|---|---|
| p = abc.html | dados de abc.html |
| p = def.mp3 | dados de def.mp3 |
| p = ghi.jpg | dados de ghi.jpg |

Tabela 4.4: Componente FIB da arquitetura CCDTN

| **Chave** | **Valor** |
|---|---|
| p = abc.html | {j = 0, j = 1} |
| p = def.mp3 | {j = 1, j = 2} |
| p = ghi.jpg | {j = 3} |

## 4.3 Análise da fase de descoberta

O Algoritmo 3 apresenta o funcionamento da fase de descoberta do protocolo DIRESC.

Como descrito na Seção 3.3, essa fase diz respeito à busca por conteúdo no ambiente CCDTN. As entradas do Algoritmo 3 são: o próprio nó $i$; o nó $j$; os valores previamente atualizados de $R_i(j)$ e $R_{medio}$; e o limitador de submissões $\mu$, cuja definição se encontra na seção supracitada.

Nesse algoritmo são apresentadas as operações referentes aos critérios de envio de pacotes de interesse (linhas 2-14), bem como o processamento realizado após o recebimento desses mesmos pacotes (linhas 15-35). A complexidade de tal algoritmo é detalhada a seguir:

**Algoritmo 3:** DIRESC: fase de descoberta

```
   Entrada: i, j, R_i(j), R_medio(i), μ
 1 início
 2     para todo prefixo p na PIT de i faça
 3         se p está na CS de j ou p está na FIB de j ou R_i(j) ≥ R_medio então
 4             i envia um pacote de interesse em p para j
 5             se (α < μ) e (p está na FIB de j ou j está no p da FIB de i) então
 6                 α = α + 1
 7             fim
 8             senão
 9                 se (β < μ) (e R_i(j) ≥ R_medio(i)) então
10                     β = β + 1
11                 fim
12             fim
13         fim
14     fim
15     para todo interesse em p recebido de j faça
16         se p está na CS de i então
17             i envia um pacote de dados de prefixo p para j
18         fim
19         senão
20             se p está na FIB de i ou R_i(j) ≥ R_medio(i) então
21                 se p está na PIT de i então
22                     Atualiza-se p na PIT de i: incluindo j no mapeamento existente
23                     Descarta interesse
24                 fim
25                 senão
26                     Atualiza-se a PIT de i: adiciona-se p mapeado para j
27                     Mantém interesse para encaminhamento futuro
28                 fim
29             fim
30             senão
31                 Descarta interesse
32             fim
33         fim
34     fim
35 fim
```

- **Linhas 2-14**: neste bloco, temos que a complexidade depende do tamanho da PIT percorrida, isto é, da quantidade de prefixos existentes e avaliados de forma unitária, sendo assim, na ordem de $O(PIT)$. As verificações de correspondência do prefixo p na CS e na FIB de j, bem como a comparação entre $R_i(j)$ e $R_{medio}$ (linha 3), todas têm complexidade $O(1)$.

  Do mesmo modo, a comparação entre o número de submissões (do interesse em p)

para o cluster ($\alpha$) e o limitador de submissões ($\mu$), a verificação de correspondência de $p$ na FIB de $j$, e o acesso ao vetor mapeado ao $p$ da FIB de $i$ (linha 5), todas também possuem complexidade constante ($O(1)$). Isso acontece devido ao tipo de estruturas de dados utilizadas.

Por fim, as comparações entre o número de submissões (do interesse em $p$) para os amigos ($\beta$) e o limitador de submissões ($\mu$), e entre $R_i(j)$ e $R_{medio}$ (linha 9), também apresentam complexidade $O(1)$, do mesmo modo que as operações de incremento das variáveis $\alpha$ e $\beta$ (linhas 6 e 10).

Portanto, para o envio de interesses na fase de busca, temos uma complexidade na ordem de $O(PIT)$.

- **Linhas 15-34**: como explicado anteriormente, a operação de busca em uma tabela *hash* possui complexidade constante. Portanto, a verificação de correspondência do prefixo $p$ na CS (linha 16), na FIB (linha 20) e na PIT (linha 21) de $i$ é da ordem $O(1)$. Do mesmo modo, a comparação entre $R_i(j)$ e $R_{medio}$ (linha 20), bem como as atualizações da PIT (linhas 22 e 26) são da mesma ordem de complexidade. Portanto, temos que o processamento de interesses recebidos é da ordem $O(1)$.

Assumindo que a complexidade total do Algoritmo 3 é definida por um conjunto de operações relizadas sobre as parciais resultantes de cada etapa do mesmo, de modo que aquela é igual à maior entre as verificadas, temos pela Equação 4.3 que a complexidade $C_{descoberta}$ do Algoritmo 3 é dada por:

$$\begin{aligned} C_{descoberta} &= O(PIT) + O(1) \\ C_{descoberta} &= O(PIT) \end{aligned} \tag{4.3}$$

O componente PIT é implementado na forma de tabela *hash*, de modo que cada prefixo seja mapeado para um objeto da classe *EntradaPIT* que armazena todas as informações de busca referentes ao prefixo questão. São eles: o prefixo de busca; o *booleano* que informa se o nó que mantém o objeto *entradaPit* é o gerador da requisição ou apenas um encaminhador; o tempo de vida do interesse; os limitadores $\alpha$ e $\beta$; e a lista que mantém os identifcadores de todos os nós que enviaram interesses nesse prefixo para o detentor desse objeto, como representado na Tabela 4.5 e na Figura 4.1.

Tabela 4.5: Componente PIT da arquitetura CCDTN

| Chave | Valor |
|---|---|
| p = abc.html | entradaPIT(abc.html) |
| p = def.mp3 | entradaPIT(def.mp3) |
| p = ghi.jpg | entradaPIT(ghi.jpg) |

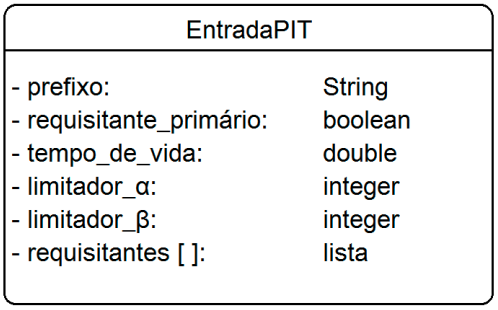


Figura 4.1: Classe *EntradaPIT*

## 4.4 Análise da fase de recuperação

O Algoritmo 4 apresenta o funcionamento da fase de recuperação do nosso protocolo.

Essa fase, como explicado na Seção 3.4, é responsável pela tomada de decisão para o envio de pacotes de dados e pelo processamento dos mesmos quando recebidos. Assim, o Algoritmo 4 possui as seguintes entradas: o próprio nó $i$; o nó $j$; o valor de relacionamento social entre $j$ e o requisitante, $R_j$(requisitante); e a média de relacionamentos sociais do nó $j$, $R_{medio}(j)$.

Nesse algoritmo são apresentadas as operações referentes aos critérios de envio de pacotes de dados (linhas 2-8), bem como o processamento realizado após o recebimento desses mesmos pacotes (linhas 9-21). A complexidade de tal algoritmo é detalhada a seguir:

- **Linhas 2-8**: na linha 2, observa-se que a PIT é percorrida para a verificação unitária de cada prefixo nela existente; portanto, temos que essa operação depende do tamanho de tal componente, resultando em uma complexidade de ordem $O(PIT)$. Por outro lado, todas as demais operações inseridas nesse bloco são de complexidade constante, ou seja, a verificação de correspondência de $p$ na CS de $i$ (linha 3) e na PIT de $j$, o acesso aos requisitantes de $p$ na PIT de $i$ e a comparação entre os valores pré-calculados de $R_j$(requisitante) e $R_{medio}(j)$ (linha 4) são todas de ordem $O(1)$. Assim, a complexidade para o envio de pacotes de dados pode ser declarada

**Algoritmo 4:** DIRESC: fase de recuperação

```
   Entrada: i, j, R_j(requisitante), R_medio(j)
 1 início
 2     para todo prefixo p na PIT de i faça
 3         se p está na CS de i então
 4             se j é um requisitante de p ou p está na PIT de j ou R_j(requisitante) ≥
                 R_medio(j) então
 5                 i envia um pacote de dados de prefixo p para j
 6             fim
 7         fim
 8     fim
 9     para todo dado de prefixo p recebido de j faça
10         se p está na PIT de i então
11             se i é o requisitante primário de p então
12                 dado é enviado para a face de aplicação requisitante
13             fim
14             senão
15                 conteúdo é mantido em CS para encaminhamento futuro
16             fim
17         fim
18         senão
19             conteúdo é descartado
20         fim
21     fim
22 fim
```

como sendo $O(PIT)$.

- **Linha 9-21**: como explicado anteriormente, a verificação de correspondência do prefixo p na PIT de i é uma comparação de complexidade constante, ou seja, $O(1)$, bem como a checagem do valor do *booleano* que informa se o nó que recebeu o dado é o seu requisitante primário. Portanto, temos que a complexidade para o processamento de dados recebidos é da ordem $O(1)$.

Assumindo que a complexidade total do Algoritmo 4 é definida por um conjunto de operações relizadas sobre as parciais resultantes de cada etapa do mesmo, de modo que aquela é igual à maior entre as verificadas, temos pela Equação 4.4 que a complexidade $C_{recuperacao}$ do Algoritmo 4 é dada por:

$$\begin{aligned} C_{recuperacao} &= O(PIT) + O(1) \\ C_{recuperacao} &= O(PIT) \end{aligned} \tag{4.4}$$

## 4.5 Complexidade do protocolo DIRESC

Após a análise realizada separadamente sobre cada fase do protocolo DIRESC, torna-se possível calcular e definir a representação final da complexidade total de nosso protocolo. Estando todas as operações supracidadas incluídas dentro do bloco iniciado na linha 2 do Algoritmo 1, que determina o contato entre os nós $i$ e $j$, e que possui complexidade $O(1)$, pode-se dizer que a representação final será dada pela soma de todas as complexidades parciais, resultantes da análise sobre cada fase do protocolo. Deste modo, temos que a complexidade do DIRESC é dada pela Equação 4.5:

$$\begin{aligned} C_{diresc} &= O(E_N(i) + S_M(i)) + O(CS + FIB) + O(PIT) + O(PIT) \\ C_{diresc} &= O(E_N(i) + S_M(i)) + O(CS + FIB) + O(PIT) \\ C_{diresc} &= O(E_N(i) + S_M(i) + CS + FIB + PIT) \end{aligned} \tag{4.5}$$

A partir dessa representação, e pela notação O, a complexidade do protocolo DIRESC é dada pela maior entre $O(E_N(i))$, $O(S_M(i))$, $O(CS)$, $O(FIB)$, $O(PIT)$. Acerca disto, algumas inferências podem ser elaboradas:

1. Para nós de alta centralidade na rede, ou seja, nós bem relacionados com grande parcela da rede, podemos inferir que as tabelas de encontros (ver Tabela 4.1) e de relacionamentos (ver Tabela 4.2) possuirão muitos registros. Assumindo que a quantidade de registros nessas estruturas seja superior à existente nos componentes básicos da arquitetura CCDTN, podemos dizer que a complexidade de execução do protocolo para esses nós será dirigida por $O(E_N(i) + S_M(i))$. Assim, definimos que, para nós intermediários com essas características, a complexidade do protocolo é dada por:

$$C_{diresc(intermediario)} = O(E_N(i) + S_M(i)) \tag{4.6}$$

2. Partindo da mesma premissa anteriormente apresentada, mas supondo que os nós em questão tenham amplo conhecimento de provedores de conteúdo, podemos inferir que aqueles podem vir a receber muitos anúncios oriundos desses e, portanto, ter uma elevada quantidade de registros na FIB. Assumindo tal conjectura, podemos dizer que a complexidade resultante de execução é da ordem $O(FIB)$. Assim, para nós intermediários que se comportem segundo o esquema supracitado, a com-

plexidade do protocolo é dada por:

$$C_{diresc(intermediario)} = O(FIB) \tag{4.7}$$

3. Por outro lado, assumindo nós produtores de conteúdo, que podem vir a gerar muitos dados e, consequentemente, ter o recurso de cache possuindo muitos registros, de modo que supere a quantidade existente em outros componentes CCDTN, temos que a complexidade resultante de execução do protocolo é $O(CS)$. Deste modo, para nós produtores, definimos que a complexidade do protocolo é dada por:

$$C_{diresc(produtor)} = O(CS) \tag{4.8}$$

4. Por fim, assumindo nós que atuem predominantemente como requisitantes de conteúdos (primários e/ou intermediários), supõe-se que suas tabelas PIT apresentem elevada quantidade de registros, o que permite inferir que a complexidade de execução do protocolo para esses nós será da ordem $O(PIT)$. Desta maneira, para nós requisitantes, sejam eles primários ou intermediários, estabelecemos que a complexidade do protocolo é dada por:

$$C_{diresc(requisitante)} = O(PIT) \tag{4.9}$$

Assim, a partir da análise detalhada sobre o protocolo DIRESC, traçamos o perfil comportamental dos diferentes atores do cenário CCDTN, bem como a complexidade de execução do referido protocolo para cada um deles e observamos que, de modo geral, a complexidade do nosso protocolo pode ser determinada em função dos componentes básicos de uma CCN. Ou seja, ainda que o DIRESC defina uma estratégia de seleção de nós retransmissores para descoberta e recuperação de dados, dependente de fatores como amizade e distribuição de informação sobre prefixos de conteúdo, nossa proposta apresenta um nível de complexidade relativamente baixo e próximo ao de uma CCN tradicional.

# 5. Avaliação de desempenho

Neste capítulo, serão descritos a ferramenta de simulação, os cenários propostos para a avaliação do protocolo e seus parâmetros, as métricas para análise de desempenho e, por fim, os resultados obtidos a partir de tais configurações.

O objetivo de avaliarmos o nosso protocolo é constatar o seu alto desempenho quando comparado a outras propostas da literatura em relação às métricas a serem definidas. Para tanto, o comparamos com o protocolo baseado em relacionamento social, STCR, e probabilístico, PIFP, ambos apresentados na Seção 2.3, bem como com o comportamento padrão de CCN, cuja implementação será apresentada na Seção 5.1.2. Deste modo, validaremos o DIRESC e o apresentaremos como uma solução promissora para a descoberta e recuperação de dados em CCDTN.

## 5.1 Ferramenta de simulação

Como descrito brevemente no Capítulo 1, uma das contribuições do nosso trabalho foi a implementação da arquitetura CCN no simulador ONE (*Opportunistic Network Environment*) [33], que é voltado para redes de comunicação oportunista. Nesse contexto, algumas modificações em seu código tiveram de ser realizadas para que o mesmo se adequasse à proposta de simular uma CCDTN.

Para tanto, algumas classes foram implementadas, bem como outras foram alteradas. A seguir, detalharemos essas modificações e implementações.

### 5.1.1 Geração de mensagens

Nos capítulos anteriores, explicamos a existência de dois tipos de mensagens que regem o comportamento de uma CCN e, consequentemente, de uma CCDTN; são elas:

interesses e dados. Esses dois tipos de pacotes são criados no *buffer* dos nós quando: (i) a aplicação requisitante apresenta interesse em determinado conteúdo; e (ii) quando a aplicação produtora gera o conteúdo e o mantém em cache para satisfazer requisições, respectivamente.

Para que essas mensagens fossem criadas, foram utilizados dois arquivos, definidos por [30], cujos conteúdos definem (i) as mensagens de interesse a serem criadas no *buffer* de nós aleatórios ao longo do tempo de simulação e (ii) os dados previamente carregados por nós produtores. Assim, por meio da classe de leitura de eventos *StandardEventsReader*, esses arquivos são lidos e as respectivas mensagens criadas.

No arquivo intitulado *lista_de_interesses.txt* existem informações como: nó de origem, identificador, tamanho e nome da mensagem, sendo esse último correspondente ao prefixo do conteúdo buscado. Vale ressaltar que as mensagens de interesse não apresentam um campo para a identificação do nó de destino, uma vez que esses pacotes não são encaminhados com vista a serem entregues a um destinatário específico, mas a buscar qualquer nó que armazene o conteúdo buscado.

Por outro lado, o arquivo *lista_de_conteúdos.txt* identifica os prefixos de conteúdos produzidos na rede, o seu tamanho e, para cada um deles, o nó produtor correspondente. Nesse contexto, são definidos 500 conteúdos distintos, cujos tamanhos variam de 250 a 350 bytes, distribuídos entre 15 nós produtores.

Além disso, definimos que as mensagens, tanto de interesse quanto de dados, possuiriam um tempo de vida (*Time To Live* ou TTL) que varia de acordo com o cenário proposto. Ao término desse intervalo de tempo, as mesmas seriam descartadas do *buffer* dos nós. É importante destacar que nosso protocolo não propõe um mecanismo de gerenciamento de cache e, portanto, de substituição de dados neste componente. Assim, empregamos, para o descarte de pacotes (de interesse e dados), uma política semelhante ao FIFO[1], mas operante em função do TTL das mensagens. Ou seja, as primeiras mensagens transmitidas serão as primeiras a serem descartadas, em vista que seu tempo de vida se esgotará antes que o das mensagens mais recentemente enviadas. Vale destacar também que, no que diz respeito aos pacotes de dados, esta política de descarte é aplicável apenas em nós intermediários (que replicam conteúdos e operam como provedores secundários), isto é, os dados gerados por nós produtores (provedores primários) não são descartados de suas caches, de modo que os mesmos sejam capazes de sempre alimentar a rede com os conteúdos que produzem.

[1] Em Ciência da Computação, FIFO (acrônimo para *First In, First Out*, que em português significa primeiro a entrar, primeiro a sair) se refere a estruturas de dados do tipo fila, nas quais os primeiros elementos a deixarem tais estruturas são aqueles que primeiro adentraram nas mesmas.

Vale ressaltar também que nosso trabalho não abrange aspectos de segurança (sendo esses propostos como um trabalho futuro). Portanto, em termos de estrutura das mensagens, os pacotes de interesse e dados não se diferenciam daqueles definidos por [4], exceto no que diz respeito a esses campos específicos.

### 5.1.2 Arquitetura CCN

Anteriormente, mencionamos que, para a validação de nosso protocolo, implementamos a arquitetura CCN no simulador ONE. Assim, os componentes básicos dessa arquitetura (ver Seção 2.2) foram desenvolvidos em uma classe denominada *ArquiteturaCCN*. Nela, desenvolvemos também um componente denominado **lista de interesses recebidos**, por meio do qual, um nó pode registrar o identificador *nonce* (ver Figura 2.4) de cada mensagem de interesse recebida e evitar possíveis duplicações de armazenamento, bem como respostas repetidas a uma mesma requisição.

Além disso, na referida classe, são implementados os métodos (ou rotinas) tidos como padrão para o funcionamento de uma CCN, ainda que aplicada em um ambiente DTN, como:

1. A criação de entradas na CS, ou seja, o mapeamento entre o prefixo de conteúdo e os dados correspondentes;
2. A criação de entradas na PIT, na qual se mapeia o prefixo de conteúdo buscado por um novo interesse ao identificador do nó solicitante;
3. A atualização da PIT, isto é, a inserção do identificador de outro solicitante no mapeamento de um prefixo existente;
4. A atualização do tempo de vida (TTL) de uma entrada da PIT, ou seja, dado que exista um prefixo $p$ com $TTL_{pit} = t_a$ na PIT de $i$ e esse nó recebe um novo interesse em $p$ com $TTL_{interesse} = t_b$, de modo que $t_b > t_a$; assim, o $TTL_{pit}$ é incrementado com a diferença entre $t_b$ e $t_a$;
5. A criação de entradas na FIB, na qual se mapeia o prefixo de conteúdo anunciado por um provedor ao seu identificador;
6. A atualização da FIB, isto é, a inserção do identificador de outro anunciante no mapeamento de um prefixo existente;
7. A deleção de entradas da CS, da PIT e da FIB;

8. A deleção de mensagens expiradas de acordo com o TTL das mesmas.

Além da classe *ArquiteturaCCN*, que implementa os métodos e estruturas supracitadas, implementamos também a classe denominada *CCNbroadcasting*, que reproduz o comportamento padrão de CCN, isto é, as tomadas de decisão para o envio, processamento e descarte de mensagens segundo [4]. Assim, conforme apresentado na Seção 2.2, tornou-se possível avaliar seu funcionamento em um ambiente DTN.

### 5.1.3 Relação social

Finalmente, visando armazenar as informações relativas ao relacionamento entre nós, fortalecido a cada contato entre eles, e inspirados pela denominação dada em [29], desenvolvemos a classe *SocialTie*, cujos atributos são descritos a seguir:

- **Identificador do nó local**: refere-se a identificação do nó a quem pertence o objeto da referida classe. Esse atributo é do tipo *DTNHost*, classe que define o modelo de um nó pertencente a uma DTN e disponível no simulador ONE.
- **Identificador do nó encontrado**: atributo do tipo *DTNHost* que se refere ao identificador do nó descoberto na rede (encontrado pela primeira vez ou reencontrado).
- **Tempo de encontro**: corresponde à marcação do tempo no qual os dois nós se encontraram. Trata-se da marca de tempo do último encontro registrado entre eles.
- **Valor de relacionamento social ($R_i(j)$)**: este atributo armazena o valor calculado pela Equação 3.1 (ver Algoritmo 1) e que é atualizado a cada encontro entre os nós $i$ e $j$.

Além disso, é importante ressaltar que, na tabela de relacionamentos (ver Tabela 4.2), o mapeamento é realizado entre o nó descoberto (chave) e um objeto da classe *SocialTie* (valor). Assim, esses objetos seguem o modelo apresentado na Figura 5.1.

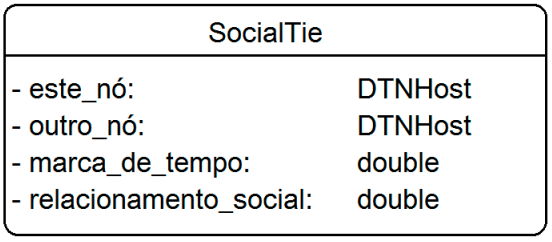


Figura 5.1: Classe *SocialTie*

## 5.2 Cenários de simulação

Para a avaliação do desempenho do DIRESC, configuramos dois cenários de simulação, de modo que um deles utiliza modelos de mobilidade, ou seja, um cenário sintético, enquanto que o outro faz uso de um *trace* (ou *dataset*) de mobilidade real, representando um cenário mais realístico. Deste modo, representamos o comportamento do nosso protocolo, bem como dos demais, em ambientes com tecnologias, taxas de transmissão e mobilidade distintas. Assim, nas subseções seguintes, tais cenários serão descritos e seus parâmetros detalhados.

### 5.2.1 Helsinki

O cenário de Helsinki, descrito e utilizado em [30], foi adotado por ser bastante diverso, com diferentes perfis de mobilidade. Com ele, foram executadas simulações com tempo de 24 horas. Para a comunicação dos nós, são utilizadas interfaces de rede simples com taxa de bits constante, nas quais, uma transmissão pode ser ativada por vez. A referida taxa foi configurada com o valor de 1 Mbps para um alcance de transmissão de até 30 metros e capacidade de armazenamento de 100 MB. Para todos os nós, foi atribuída uma carga inicial de energia suficiente para que não sejam observadas descargas totais e consequentes saídas dos participantes da rede.

Além disso, no cenário em questão foram inseridos um total de 100 nós, divididos da seguinte maneira:

- 62 **pedestres**, cuja velocidade de locomoção varia entre 0.5 e 1.5 m/s, e a mobilidade é definida a partir do modelo *Shortest Path Map-Based Movement*, que utiliza o algoritmo Dijkstra para encontrar caminhos mais curtos entre dois pontos no mapa. Para esses nós, a carga inicial é igual a 10 mil unidades de energia.
- 32 **carros**, cuja velocidade de locomoção varia entre 10 e 50 km/h, e a mobilidade é definida pelo mesmo modelo utilizado para pedestres, *Shortest Path Map-Based Movement*. Diferentemente da classe de nós pedestres, para os carros, a carga inicial foi configurada como 15 mil unidades de energia.
- 6 **bondes**, cuja velocidade de locomoção varia entre 25 e 36 km/h, e a mobilidade é definida a partir do modelo *Map-Based Movement*, o qual é parametrizado por rotas específicas para determinar sua movimentação. Para essa classe de nós, assim como de carros, a carga inicial foi configurada como 15 mil unidades de energia.

É importante destacar que, tendo sido definidos 15 produtores (ver Subseção 5.1.1), os mesmos foram distribuídos entre essas classes de nós da seguinte forma: 8 entre os pedestres; 5 entre os carros; e 2 entre os bondes.

Além disso, o mapa da cidade de Helsinki (Finlândia), ilustrado na Figura 5.2, possui uma área igual a $15,3$ km². Portanto, por ser um cenário de considerável densidade, atribuímos aos parâmetros do protocolo DIRESC os valores $\Omega = 2$, para a variação do $R_{medio}$, e $\mu = 1$, para o limitador de submissões por grupo, por corresponderem ao melhor *trade-off* entre as métricas definidas na Seção 5.3, após a realização de testes e comparação de diferentes valores atribuídos para tais parâmetros.

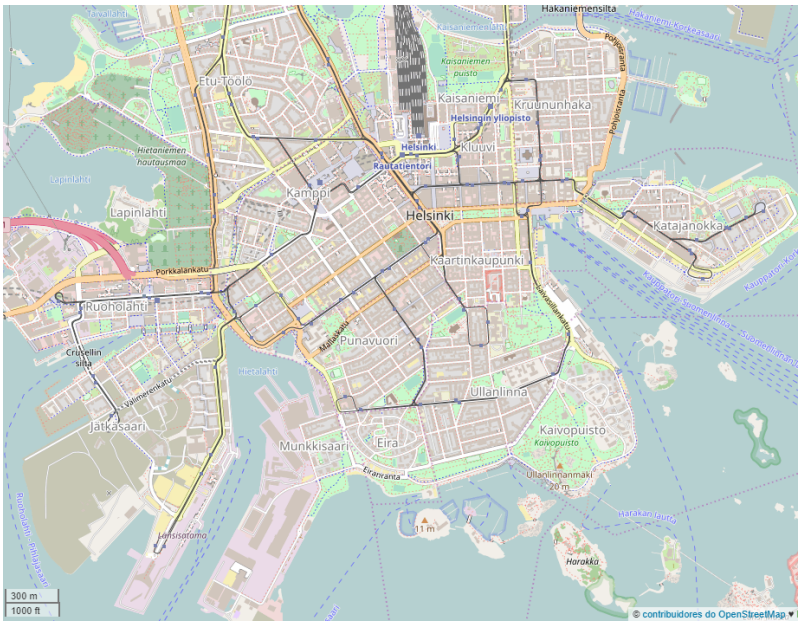


Figura 5.2: Mapa da cidade de Helsinki (Finlândia)

Deste modo, os parâmetros de simulação para o cenário de Helsinki podem ser resumidos conforme apresentado na Tabela 5.1

### 5.2.2 São Francisco

Por sua vez, o cenário de São Francisco compreende a utilização do *trace* de mobilidade real dos táxis da cidade, disponível em [34], para definir a locomoção dos nós. Nessa configuração, verificando-se que todos os nós são veículos, optamos por nos basear em alguns dos parâmetros de configuração utilizados em [35], que utiliza o mesmo *dataset* de mobilidade.

Uma vez que o conjunto de dados em questão possui registros de aproximadamente 500 táxis coletados ao longo de 30 dias, utilizamos, de modo semelhante à avaliação de

Tabela 5.1: Parâmetros do cenário de Helsinki

| **Parâmetro** | **Valor** |
|---|---|
| Duração | 24 horas |
| Tempo de vida de mensagens | 8 horas |
| Capacidade de armazenamento | 100 MB |
| Taxa de transferência | 1 Mbps |
| Alcance de transmissão | 30 metros |
| Conteúdos gerados | 500 conteúdos |
| Tamanho dos dados | De 250 a 350 bytes |
| Interesses gerados | aproximadamente 5700 interesses |
| Total de nós | 62 pedestres, 32 carros, 6 bondes |
| Carga inicial de energia | Pedestres: 10000, Carros e bondes: 15000 |
| Velocidade média | Pedestres: 0.5 a 1.5 m/s, Carros: 10 a 50 km/h, Bondes: 25 a 36 km/h |
| Área (km x km) | $4,5$ km x $3,5$ km ($15,3$ km²) |
| Variação do $R_{medio}$ | Ω: 2 |
| Limitador de submissões | μ: 1 |

[35], apenas um trecho do referido *dataset*, correspondente a um intervalo de 2 horas, no qual existem 100 nós, assim como no cenário anterior.

Assim, nesse cenário, também foram utilizadas interfaces de rede simples com taxa de bits constante, nas quais uma transmissão pode ser ativada por vez. A taxa de transmissão foi definida como 3 Mbps para um alcance de transmissão de 300 metros, o que, segundo [35], é típico em uma rede *ad hoc* veicular (VANET), e capacidade de armazenamento de 100 MB.

Além disso, apesar da ausência de grandes restrições de energia em redes veiculares [1], todos os nós foram configurados com carga inicial igual a 15 mil unidades de energia para a avaliação do comportamento desse recurso quando comparados os diferentes protocolos.

Finalmente, a área do cenário do mapa de São Francisco (EUA), ilustrado na Figura 5.3 é de aproximadamente $37,6$ km². Assim, por se tratar de um cenário de menor densidade, e após a realização de testes com diferentes valores, atribuímos aos parâmetros do protocolo os valores $\Omega = 1$, para a variação do $R_{medio}$, e $\mu = 1$, para o limitador de submissões por grupo, por corresponderem ao melhor *trade-off* entre as métricas definidas na Seção 5.3, após a realização de testes e comparação de diferentes valores atribuídos para tais parâmetros.

Portanto, do mesmo modo como para o cenário anterior, podemos resumir os parâmetros de simulação e valores atribuídos aos mesmos para o cenário de São Francisco

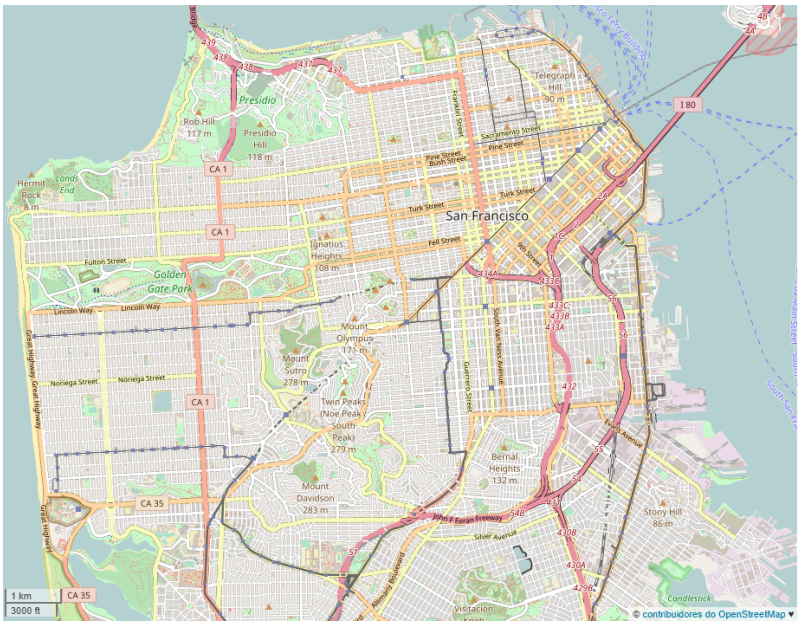


Figura 5.3: Mapa da cidade de São Francisco (EUA)

conforme apresentado na Tabela 5.2.

Tabela 5.2: Parâmetros do cenário de São Francisco

| **Parâmetro** | **Valor** |
|---|---|
| Duração | 2 horas |
| Tempo de vida de mensagens | 40 minutos |
| Capacidade de armazenamento | 100 MB |
| Taxa de transferência | 3 Mbps |
| Alcance de transmissão | 300 metros |
| Conteúdos gerados | 500 conteúdos |
| Tamanho dos dados | De 250 a 350 bytes |
| Interesses gerados | aproximadamente 500 interesses |
| Total de nós | 100 carros |
| Carga inicial de energia | 15000 A |
| Velocidade média | Varia de acordo com o *trace* de mobilidade de São Francisco |
| Área (km x km) | $5,7$ km x $6,6$ km ($37,6$ km²) |
| Variação do $R_{medio}$ | $\Omega$: 1 |
| Limitador de submissões | $\mu = 1$ |

Assim, foram definidos dois cenários distintos em termos de comportamento de movimentação dos nós e características das tecnologias de comunicação, permitindo a avaliação do nosso protocolo em diferentes ambientes de redes DTN.

## 5.3 Métricas de avaliação

Tendo sido finalizadas as configurações dos cenários Helsinki e São Francisco, para a validação do protocolo DIRESC, faz-se necessário definir as métricas a serem avaliadas e comparadas com os demais protocolos executados nos mesmos cenários. Portanto, conforme proposto em alguns trabalhos da literatura, como [29], [30], [35], [36], e [37], mediremos o desempenho dos referidos protocolos em função das seguintes métricas:

- **Taxa de dados entregues (satisfação):** nesta métrica, mediremos o que a literatura denomina como satisfação de requisições (entrega de dados) de forma absoluta ($Tx_{Abs}$) e percentual ($Tx_{\%}$). Isto é, para a medida absoluta, consideraremos a quantidade de dados entregues ($D_{Entregues}$) aos requisitantes primários a cada intervalo amostral. Já para a percentual, a qual é medida para o instante de tempo que varia de zero até o intervalo corrente, utilizaremos a relação entre o número de requisições atendidas (satisfeitas), ou seja, a quantidade absoluta de dados entregues aos requisitantes primários ($D_{Entregues}$), e o número de solicitações de conteúdo realizadas, isto é, a quantidade de interesses enviados por esses mesmos requisitantes ($I_{Enviados}$). Assim, temos que as duas medidas dessa métrica serão calculadas de acordo com as Equações 5.1 e 5.2.

$$Tx_{Abs} = D_{Entregues} \tag{5.1}$$

$$Tx_{\%} = \frac{D_{Entregues}}{I_{Enviados}} \tag{5.2}$$

- **Atraso médio de descoberta e recuperação:** a métrica de atraso médio ($At_{Medio}$) é calculada em função de todas as requisições atendidas no tempo de amostragem corrente, ainda que uma ou mais solicitações tenham sido realizadas em um intervalo de amostragem anterior. Além disso, o atraso total para cada descoberta e recuperação é medido a partir do instante de tempo no qual o interesse foi enviado pelo requisitante primário ($T_{Req}$) até o momento no qual o conteúdo foi entregue a ele ($T_{St}$). Assim, a métrica de atraso médio é definida como apresentado na Equação 5.3.

$$At_{Medio} = \frac{1}{R} \sum_{r=1}^{R} T_{St}(r) - T_{Req}(r) \tag{5.3}$$

No qual, R representa o total de requisições atendidas até o instante no qual a métrica

em questão foi calculada.

- **Sobrecarga na rede (*overhead*):** por sua vez, a métrica de sobrecarga ($\mathrm{Ovh}$) medirá a quantidade de pacotes replicados na rede, ou seja, uma vez enviados interesses e dados na rede pelos requisitantes e provedores primários, respectivamente, calcularemos quantos foram reproduzidos a partir destes por nós intermediários (retransmissores). Assim, o número de pacotes entregues ($\mathrm{P_{Entregues}}$) pela primeira vez ao destinatário (no caso de pacotes de interesse, o destinatário será qualquer nó que armazene o conteúdo requisitado e, no caso de pacotes de dados, será o próprio requisitante primário) será subtraído do total de pacotes (interesses ou dados) enviados na rede ($\mathrm{P_{Enviados}}$). Especificamente, tal métrica será normalizada pela quantidade total de pacotes enviados, portanto, sua definição pode ser representada pela Equação 5.4.

$$\mathrm{Ovh} = \frac{\mathrm{P_{Enviados}} - \mathrm{P_{Entregues}}}{\mathrm{P_{Enviados}}} \tag{5.4}$$

  É importante destacar que essa métrica será calculada separadamente para pacotes de interesse e para pacotes de dados, de modo que se obtenha as seguintes equações:

$$\mathrm{Ovh_{Interesses}} = \frac{\mathrm{I_{Enviados}} - \mathrm{I_{Entregues}}}{\mathrm{I_{Enviados}}} \tag{5.5}$$

$$\mathrm{Ovh_{Dados}} = \frac{\mathrm{D_{Enviados}} - \mathrm{D_{Entregues}}}{\mathrm{D_{Enviados}}} \tag{5.6}$$

Além destas métricas sugeridas por parte da literatura, optamos por fazer uso de outras, cuja medição poderá evidenciar o funcionamento eficaz do DIRESC no que diz respeito aos objetivos a que se propõe. São elas:

- **Média de pacotes retransmitidos por nó:** por meio dessa métrica, verificaremos a média aritmética de retransmissões de mensagens ($\mathrm{Rtr_{Media}}$) que todo nó da rede realiza a cada intervalo de amostragem. Assim, será possível observar qual dos protocolos avaliados gera mais saltos de pacotes entre os nós para que se realize a descoberta e recuperação do conteúdo buscado e que, consequentemente, resulta em maior consumo de recursos. A Equação 5.7 apresenta o cálculo dessa métrica.

$$\mathrm{Rtr_{Media}} = \frac{\mathrm{N_{Retransmissoes}}}{\mathrm{N_{Nos}}} \tag{5.7}$$

- **Consumo de energia:** A fim de complementar a análise proposta pela métrica anterior, verificaremos o consumo percentual de energia ($E_{Consumo\%}$) dos nós ao final do tempo de simulação. Assumindo que a carga de energia seja consumida pelos fatores: (i) manter o dispositivo ativo; (ii) efetuar a varredura (descoberta) de outros nós; e (iii) transmitir e receber pacotes. E compreendendo que os dois primeiros serão iguais para todas as simulações e com os diferentes protocolos, verificaremos qual o impacto do funcionamento de cada protocolo no consumo energético pelo envio e recebimento de pacotes. Assim, a referida métrica pode ser representada pela Equação 5.8.

$$E_{Consumo\%} = E_{Inicial\%} - E_{Final\%} \tag{5.8}$$

O cálculo das métricas supracitadas foi realizado sobre a média aritmética de 30 rodadas de simulação em cada um dos cenários definidos na Seção 5.2 para cada protocolo avaliado (DIRESC, PIFP, STCR e CCN*broadcasting*).

## 5.4 Resultados experimentais

Nesta seção, apresentaremos os gráficos referentes às métricas de avaliação supracitadas, bem como a discussão acerca do comportamento de cada protocolo em relação as mesmas.

Para a geração de tais gráficos, utilizamos a ferramenta matemática de análise de dados *R Software* [38], com a qual calculamos a média aritmética dos resultados quantitativos das 30 rodadas de simulação efetuadas para cada protocolo e o nível de confiança de 95% para cada intervalo de amostragem.

### 5.4.1 Cenário de Helsinki

Para este cenário, além das configurações apresentadas na Seção 5.2.1, utilizou-se intervalos de amostragem de 1 hora, nos quais são gerados aproximadamente, em cada um deles, 250 pacotes de interesse por meio do *trace* abordado na Seção 5.1.1.

#### Taxa de dados entregues

A Figura 5.4 apresenta a taxa absoluta de dados entregues (em verde) por meio de cada protocolo. Nos gráficos que a compõem, foram adicionadas também as curvas referentes ao envio de pacotes de interesse (em vermelho).

Vemos, na Figura 5.4 (a), a relação entre interesses enviados e satisfeitos pelo protocolo *CCNbroadcasting*. Observa-se que nas primeiras horas de simulação ele se comporta de maneira satisfatória no que concerne ao envio de mensagens de interesse, isto é, a quantidade enviada se aproxima do montante gerado a cada intervalo. No entanto, essa taxa de envio tende a decair ao longo do tempo de simulação, devido a natureza epidêmica do protocolo, que tende a consumir rapidamente a capacidade de *buffer* dos demais nós da rede, ocasionando, por conseguinte, rejeições no recebimento de novos pacotes, por parte de retransmissores, pela falta de recursos suficientes para armazenamento.

Pode-se dizer que a taxa absoluta de dados entregues do protocolo também é satisfatória nas primeiras horas de simulação em vista que aproxima bastante da quantidade de

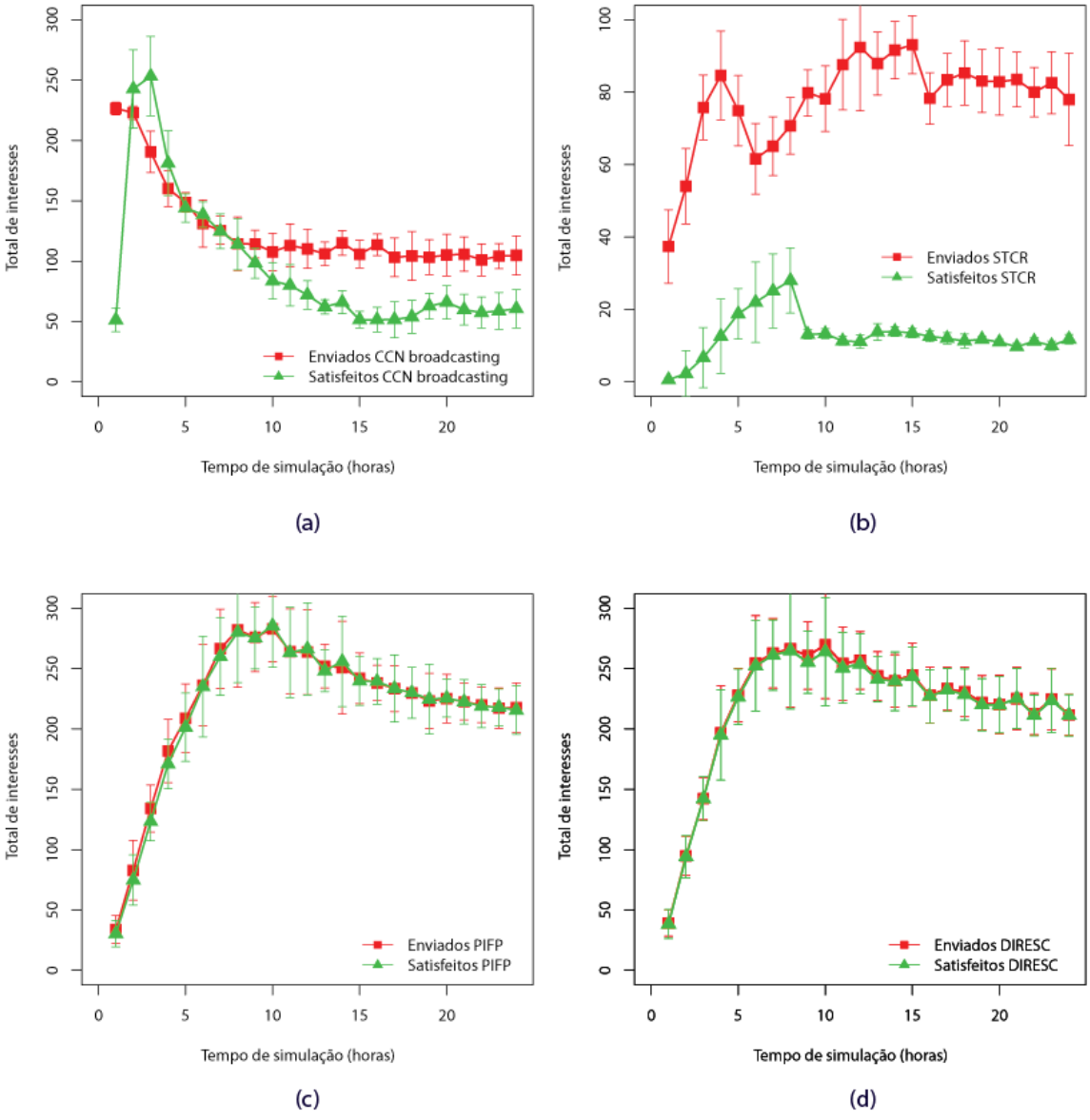


Figura 5.4: Taxa absoluta de dados entregues no cenário de Helsinki

requisições realizadas (exceto na primeira hora). Contudo, essa satisfação também tende a reduzir ao longo da simulação pela mesma razão que justifica o decaimento no envio de pacotes de interesse. Além disso, devido às recusas de recebimento de pacotes pela falta de capacidade de armazenamento suficiente, podemos inferir também que parte das requisições deixam de ser entregues aos provedores de conteúdo e, portanto, não são atendidas até que seu tempo de vida de esgote.

Por outro lado, ainda assim o *CCNbroadcasting* apresenta uma elevada taxa percentual de dados entregues, como apresentado na Figura 5.5 (em roxo). Isso acontece pois, devido ao mesmo comportamento epidêmico, muitos nós tendem a replicar pacotes de dados transmitidos, diminuindo a distância entre requisitantes e provedores e, portanto, aumentando as chances de se satisfazer interesses.

O protocolo STCR, cuja taxa absoluta de dados entregues é apresentada na Figura 5.4 (b), possui um mecanismo de seleção de retransmissores de pacotes de interesses e dados bastante específico, o que restringe muito as chances de que as referidas mensagens sejam entregues aos respectivos destinatários. Nesse estudo, avaliamos este protocolo atribuindo ao seu parâmetro K o valor 10, uma vez que apresenta o melhor equilíbrio entre as métricas avaliadas em [29]. Assim, ao ser executado, o STCR dividiu os nós existentes no cenário em 10 grupos distintos, cujos participantes possuiam valores similares de centralidade social. Dadas as condições de envio de pacotes de interesse estabelecidas pelo protocolo [29], o mesmo foi capaz de enviar uma quantidade bem reduzida de tais pacotes em relação ao montante que é gerado a cada intervalo, como se observa na figura supracitada.

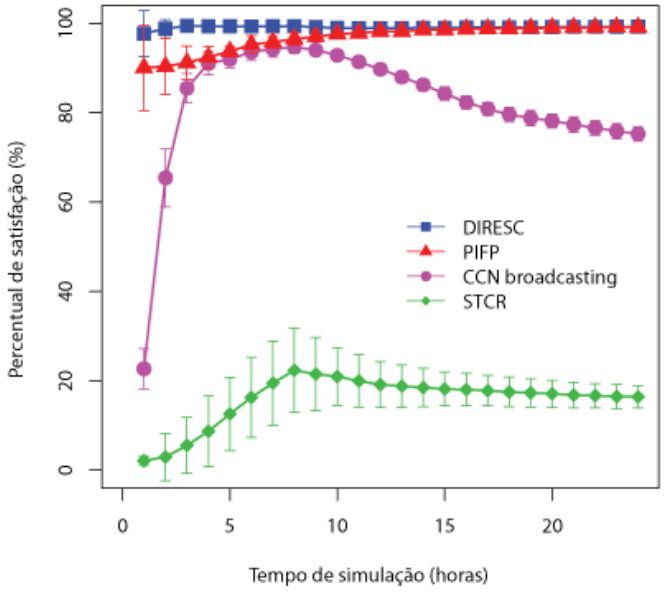


Figura 5.5: Taxa percentual de dados entregues no cenário de Helsinki

Além disso, o STCR negligencia a possibilidade do contato direto entre requisitantes e qualquer provedor, de modo que solicitações não podem ser diretamente atendidas, mas necessitam obrigatoriamente passar pelo processo de retransmissão seccionada do protocolo. Ou seja, interesses devem ser enviados apenas para nós de maior centralidade social, para que esses se encarregem de retransmitir aqueles pacotes, selecionando retransmissores com base no seu relacionamento social com tais provedores, para que então os pacotes de dados sejam enviados também com base na verificação do relacionamento social entre os nós selecionados por estes provedores e os requisitantes.

Outro problema evidente do STCR é a falta de participação mais clara e ativa do componente PIT no funcionamento do protocolo. Uma vez que não são feitos registros ao longo da rede acerca de interesses ainda não atendidos, se as condições para o encaminhamento de pacotes demoram mais para acontecer do que o tempo de vida das mensagens, informações como "para quem encaminhar o interesse" ou "para quem entregar o dado" se tornam vagas e são perdidas, de modo que o processo de descoberta e recuperação não é finalizado com sucesso. Por essa razão, observa-se o decaimento abrupto na curva de interesses satisfeitos após oito horas (tempo de vida das mensasgens no cenário proposto) de simulação. Assim, o protocolo STCR apresenta a menor taxa absoluta de dados entregues e percentual, como apresentado na Figura 5.5 (em verde).

Os protocolos PIFP e DIRESC apresentam os melhores comportamentos na relação entre interesses enviados e satisfeitos apresentada na Figura 5.4 (c) e (d), respectivamente. Isso acontece, pois o funcionamento de ambos sugere a menor utilização de retransmissores, ainda que o DIRESC seja menos restritivo que o PIFP acerca disso, de modo que as requisições sejam transmitidas para nós que de fato têm maiores chances de atender tais demandas. Assim, devido a esse comportamento, a capacidade de *buffer* não é consumida tão rapidamente, o que permite a eles enviar tantos interesses quanto são gerados, a partir de determinado tempo de simulação. Isto é, nas primeiras cinco horas para o PIFP e quatro para o DIRESC, a quantidade de interesses enviados é baixa, porém crescente, em relação ao montante gerado a cada intervado de uma hora (aproximadamente 250 pacotes interesses). Isso acontece, pois, nestas primeiras horas, ambos os protocolos ainda estão preparando as informações que alimentam o seu funcionamento e que definirão o comportamento da rede. No caso do PIFP, os cálculos de probabilidade de encontro entre os nós e os conteúdos; já no caso do DIRESC, os relacionamentos sociais e os anúncios de prefixos. Portanto, somente após essas primeiras horas a quantidade de interesses enviados se aproxima da quantidade gerada.

Mesmo possuindo comportamentos semelhantes nas curvas da Figura anterior, o DIRESC ainda apresenta melhor taxa percentual de dados entregues em relação ao PIFP,

pelo menos até as nove primeiras horas de simulação, instante de tempo a partir do qual ambos os protocolos possuem taxas muito próximas, como mostra a Figura 5.5 (DIRESC em azul e PIFP em vermelho).

A Tabela 5.3 exibe um resumo dos valores obtidos para a taxa absoluta de dados entregues sobre a média aritmética das 30 simulações para cada protocolo, e revela que o PIFP obtém um melhor desempenho para a referida métrica em relação aos demais.

Tabela 5.3: Taxa absoluta de dados entregues no cenário de Helsinki

| **Protocolo** | **Mínimo** | **Máximo** | **Média** | **Desvio padrão** |
|---|---|---|---|---|
| *CCNbroadcasing* | 51,4 | 253,2 | 95,25 | 58,57 |
| STCR | 1,6 | 28 | 12,81 | 6,02 |
| PIFP | 30,4 | 285,5 | 218,27 | 62,22 |
| DIRESC | 38,4 | 265 | 217,60 | 54,09 |

Na Tabela 5.3, observamos que o DIRESC apresenta, nas colunas referentes aos valores máximo e médio, números que indicam perdas do protocolo na taxa de dados entregues em relação ao PIFP. No entanto, tratam-se de quantias obtidas sobre valores absolutos. Isto é, quando se leva em consideração a quantidade de interesses enviados e se calcula esses mesmos valores de forma percentual, nota-se que o DIRESC resulta em um melhor desempenho quando comparado aos demais protocolos, como exposto na Tabela 5.4.

Tabela 5.4: Taxa percentual de dados entregues no cenário de Helsinki

| **Protocolo** | **Mínimo** | **Máximo** | **Média** | **Desvio padrão** |
|---|---|---|---|---|
| *CCNbroadcasing* | 22,7 | 94,69 | 82,10 | 14,83 |
| STCR | 2,13 | 22,37 | 15,89 | 5,52 |
| PIFP | 90,04 | 99,13 | 96,74 | 2,97 |
| DIRESC | 97,68 | 99,41 | 99,09 | 0,33 |

### Atraso médio de descoberta e recuperação

No que concerne ao tempo de descoberta e recuperação de dados apresentado na Figura 5.6 (a), observa-se que, dentre todos os protocolos, o STCR (em verde) apresenta o maior atraso médio. Isso acontece, como explicado anteriormente, em razão das muitas características do funcionamento do protocolo, entre as quais o processo de anúncio realizado apenas para nós de mais alta centralidade, de modo que nós de alto relacionamento social com os provedores e do mesmo nível social ou até um nível social mais baixo (níveis são determinados de acordo com a centralidade dos nós) podem vir a não ter qualquer conhecimento dos prefixos que tais provedores atendem, o que reduz significantemente

as chances de recuperação de dados, uma vez que aqueles nós, ao atuarem como requisitantes precisarão, necessariamente, efetuar o processo de descoberta seccionado, que por sua vez aumenta o atraso para a obtenção do conteúdo solicitado.

Na Figura 5.6 (b), que representa a mesma métrica em minutos, para a melhor visualização do comportamento dos demais protocolos, observa-se que, nas quatro primeiras horas de simulação, o protocolo *CCNbroadcasting* apresenta um elevado atraso médio em relação aos intervalos de tempo seguintes. Isso acontece, pois, nessas primeiras horas os pacotes de dados ainda não foram devidamente replicados ao longo da rede, e, portanto, demanda-se certo tempo até que os mesmos sejam recuperados a partir dos provedores primários. Após algum tempo, são geradas cópias dos referidos pacotes nas caches dos nós intermediários, o que reduz a distância entre o requisitante e os provedores secundários, bem como reduz o atraso de descoberta e recuperação.

O protocolo PIFP, por sua vez, apresenta o segundo menor atraso obtido. Podemos inferir que esse atraso é observado uma vez que o protocolo se baseia unicamente no cálculo da probabilidade dos nós encontrarem determinado conteúdo, de modo que os mais propensos a efetuarem a recuperação são frequentemente escolhidos. No entanto, para que tal cálculo seja efetuado por outros nós que não sejam o produtor, uma primeira requisição precisa ser feita e atendida diretamente para tal produtor e para cada conteúdo, para que então se dê início aos cálculos de probabilidade por parte de outros nós. Portanto, demanda-se um certo tempo até que cada participante da rede possua o seu próprio percentual de chance de encontrar qualquer dado.

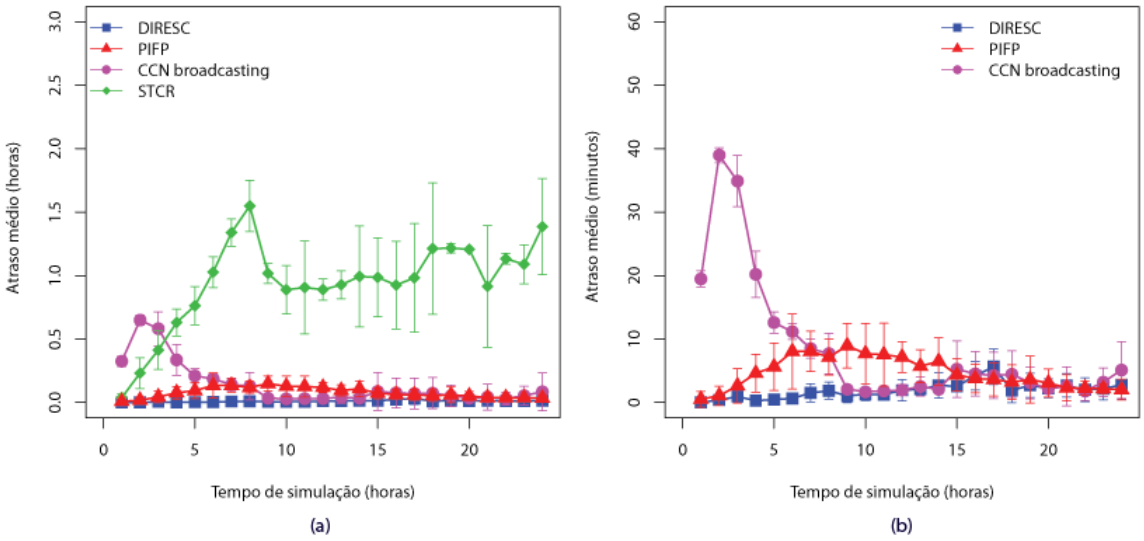


Figura 5.6: Atraso médio no cenário de Helsinki

O DIRESC, por outro lado, por definir três possibilidades de envio de interesses e dados, aumenta significantemente as chances de descoberta e recuperação de conteúdo. Além disso, o esquema de priorização de envio de pacotes garante que os mesmos sejam enviados primeiramente aos nós com maiores chances de atender às requisições (envio de interesses) ou finalizar o processo de entrega (envio de dados). Assim, nosso protocolo apresenta o menor atraso observado para descoberta e recuperação no cenário proposto.

A Tabela 5.5 exibe um resumo dos valores obtidos para o atraso médio de descoberta e recuperação (em minutos) sobre a média aritmética das 30 simulações para cada protocolo, no qual é evidenciado que o DIRESC resulta em um menor atraso e, portanto, maior rapidez para a obtenção dos dados solicitados em comparação com as outras abordagens.

Tabela 5.5: Atraso médio de descoberta e recuperação (em minutos) no cenário de Helsinki

| **Protocolo** | **Mínimo** | **Máximo** | **Média** | **Desvio padrão** |
|---|---|---|---|---|
| *CCNbroadcasing* | 1,7 | 38,96 | 8,42 | 10,23 |
| STCR | 2,28 | 92,96 | 56,66 | 20,78 |
| PIFP | 0,51 | 8,95 | 4,65 | 2,49 |
| DIRESC | 0,43 | 5,64 | 1,89 | 1,27 |

### Sobrecarga na rede (*Overhead*)

No que diz respeito à sobrecarga na rede, medimos a quantidade de pacotes de interesses e de dados replicados e normalizamos o valor obtido pela quantidade total de pacotes transmitidos. Assim, obtivemos os gráficos apresentados na Figura 5.7. Nela, observamos que, com relação ao *overhead* gerado por pacotes de interesse, apresentado no gráfico da

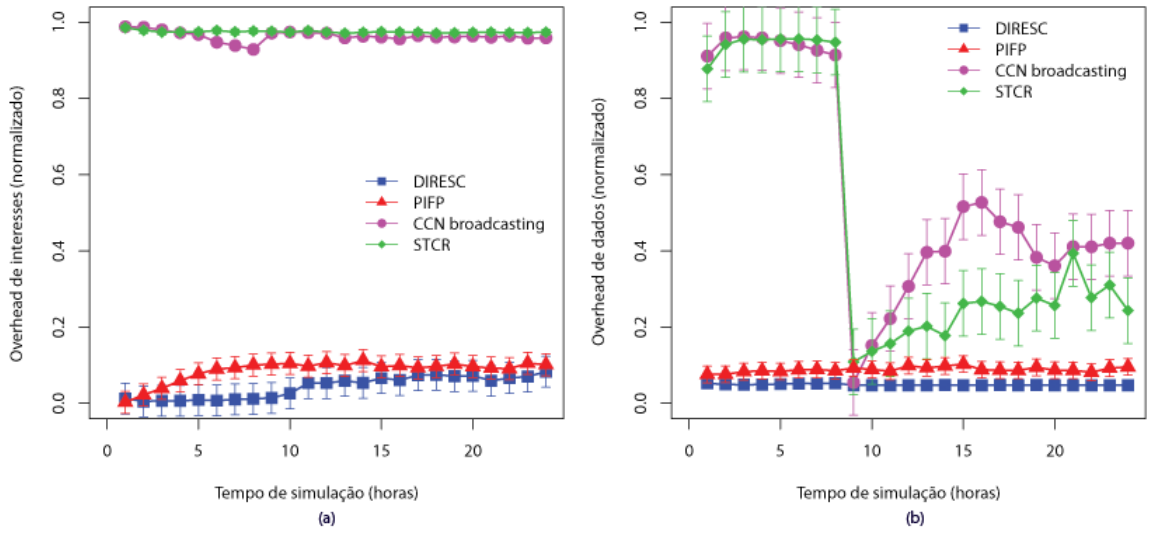


Figura 5.7: Overhead de interesses e dados no cenário de Helsinki

Figura 5.7 (a), dois protocolos apresentam os maiores índices de sobrecarga, os quais são muito semelhantes. No caso do *CCNbroadcasting*, isso acontece devido a sua natureza epidêmica, ou seja, seu comportamento em *broadcasting* faz com que uma carga excessiva de pacotes de interesse seja replicada na rede. Já o elevado overhead gerado pelo STCR pode ser explicado em razão da baixa taxa de entrega dos pacotes em questão, de modo que, quando normalizada pelo total de transmissões, a métrica tende a apresentar o comportamento observado.

Por sua vez, os protocolos PIFP e DIRESC apresentam índices baixos e semelhantes de sobrecarga. Isto acontece, como explicado anteriormente, pois o comportamento de ambos sugere a menor utilização de retransmissores, resultando no comportamento observado na métrica de *overhead*. No entanto, apesar da semelhança no comportamento, o DIRESC ainda apresenta um ganho em relação ao PIFP.

No que concerne à sobrecarga gerada por pacotes de dados, apresentada no gráfico da Figura 5.7 (b), observamos um alto índice nas primeiras oito horas e em seguida um decaimento abrupto nas curvas do *CCNbroadcsating* e STCR. Novamente, o comportamento obtido nas primeiras horas, para ambos os protocolos, acontece devido à natureza epidêmica do primeiro e a baixa taxa de entrega do segundo, de modo que, quando normalizadas, as replicações ocasionadas por eles apresentam o comportamento observado. O decaimento, por sua vez, é resultado do término do tempo de vida das primeiras mensagens enviadas e, com isso, da liberação de espaço em *buffer* pelo descarte de muitas réplicas das mesmas e, com isso, a diminuição em escala da transmissão dessas réplicas. O crescimento seguinte em ambas as curvas é consequência de novos envios de dados. Porém, não atinge o valor das primeiras horas uma vez que aqueles começam a ser entregues em razão da liberação da capacidade de armazenamento, antes esgotada pela replicação exacerbada de mensagens pelo *CCNbroadcasting*, e pela manutenção relativamente constante da taxa de satisfação dos interesses ainda com tempo de vida restante, pelo STCR, respectivamente.

Por fim, PIFP e DIRESC novamente apresentam comportamentos semelhantes, desta vez na métrica de sobrecarga de dados. No entanto, diferentemente do *overhead* gerado pelos pacotes de interesse, no qual ambos apresentam certa inclinação de crescimento em razão do consentimento de novas transmissões conforme o tempo de simulação avança, o *overhead* resultante dos pacotes de dados é moderadamente constante. Esse comportamento é justificado, primeiramente, pela elevada taxa de dados entregues de ambos os protocolos que, por sua vez, resulta na replicação desses dados nos requisitantes satisfeitos, o que torna esses pacotes distribuídos na rede e reduz a necessidade de muitas retransmissões quando solicitados por outros nós e, portanto, reduz também a sobrecarga

gerada na rede.

As Tabelas 5.6 e 5.7 exibem um resumo dos valores obtidos para a sobrecarga normalizada gerada por pacotes de interesses e dados, respectivamente, sobre a média aritmética das 30 simulações para cada protocolo. Elas revelam que o DIRESC obtém os menores índices de *overhead*, de interesses e de dados, quando comparados aos dos demais protocolos.

Tabela 5.6: Sobrecarga normalizada de interesses no cenário de Helsinki

| **Protocolo** | **Mínimo** | **Máximo** | **Média** | **Desvio padrão** |
|---|---|---|---|---|
| *CCNbroadcasing* | 0,92 | 0,98 | 0,96 | 0,01 |
| STCR | 0,97 | 0,98 | 0,97 | 0,003 |
| PIFP | 0,002 | 0,11 | 0,08 | 0,02 |
| DIRESC | 0,005 | 0,08 | 0,04 | 0,02 |

Tabela 5.7: Sobrecarga normalizada de dados no cenário de Helsinki

| **Protocolo** | **Mínimo** | **Máximo** | **Média** | **Desvio padrão** |
|---|---|---|---|---|
| *CCNbroadcasing* | 0,05 | 0,96 | 0,56 | 0,29 |
| STCR | 0,1 | 0,95 | 0,47 | 0,34 |
| PIFP | 0,07 | 0,10 | 0,08 | 0,006 |
| DIRESC | 0,05 | 0,055 | 0,051 | 0,001 |

### Média de pacotes retransmitidos por nó

A Figura 5.8 apresenta a média de pacotes retransmitidos por nó a cada intervalo amos-

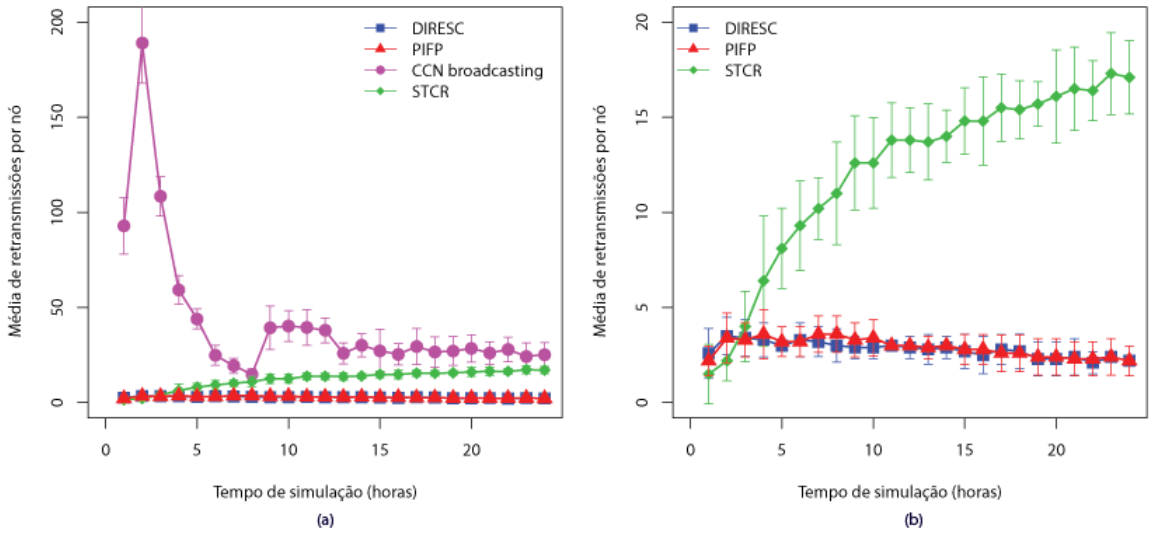


Figura 5.8: Média de retransmissões por nó no cenário de Helsinki

tral, de modo que o gráfico (b) representa uma aproximação do (a), para uma melhor visualização do comportamento de três dos protocolos. Vemos que o *CCNbroadcasting* apresenta o maior valor para essa média ao longo de toda a simulação. As primeiras variações observadas, isto é, o pico no segundo intervalo e o decaimento seguinte são resultado da transmissão indiscriminada de pacotes na rede, de modo que tal decaimento a partir da terceira hora é fruto do esgotamento da capacidade de armazenamento dos nós e, portanto, da rejeição no recebimento de pacotes e, por consequência, a redução no número de retransmissões.

Por sua vez, o STCR apresenta o segundo maior valor para a métrica em questão. Esse comportamento é justificado pelo mecanismo de seleção de retransmissores de pacotes de interesse empregado pelo protocolo, que, como explicado anteriormente, força a convergência desses pacotes primeiramente para um grupo de nós de mais alta centralidade social para que então possam ser retransmitidos para os provedores, ocasionando um desvio que poderia ser evitado caso os anúncios sobre prefixos de conteúdo (em [29], intitulado "*Content Name Digest*") fosse melhor distribuído na rede.

Por fim, novamente observamos grande proximidade no comportamento dos protocolos PIFP e DIRESC. Em alguns pontos, no entanto, nossa proposta apresenta um melhor desempenho, isto é, uma média de retransmissões por nó menor que a do protocolo probabilístico.

Este comportamento é alcançado, pois, ainda que estabeleça mais de um critério para envio de pacotes, tanto de interesse quanto de dados, o DIRESC sempre opta pelo melhor retransmissor dentre as conexões de um nó. Além disso, o fator de densidade ($\Omega$), bem como o limitador de submissões ($\mu$) garantem que não sejam feitas transmissões de forma exacerbada na rede. Com isso, limita-se fortemente o número de retransmissões. Assim, apuramos a importância de se fornecer flexibilidade no envio de pacotes desde que de modo controlado, para que não cause exaustão de recursos e prejuízos à rede.

A Tabela 5.8 exibe um resumo dos valores obtidos para a média de pacotes retransmitidos por nó, sobre a média aritmética das 30 simulações para cada protocolo, revelando que o nosso protocolo consegue balancear o número de transmissões de pacotes entre os nós, embora permita que as mesmas sejam realizadas com menor restrição que as efetuadas por dois dos outros três protocolos, resultando na menor média de retransmissão para cada participante da rede.

**Consumo de energia**

No que concerne ao consumo médio de energia apresentado nos gráficos da Figura

Tabela 5.8: Pacotes retransmitidos por nó no cenário de Helsinki

| Protocolo | Mínimo | Máximo | Média | Desvio padrão |
|---|---|---|---|---|
| *CCNbroadcasing* | 14,9 | 189 | 43,07 | 37,99 |
| STCR | 1,5 | 17,3 | 12,2 | 4,68 |
| PIFP | 2,2 | 3,6 | 2,86 | 0,47 |
| DIRESC | 2,1 | 3,5 | 2,79 | 0,38 |

5.9, observa-se que, dentre os quatro protocolos avaliados, o DIRESC é o que apresenta o menor valor obtido para essa métrica para todos os nós da rede. Isso se deve ao estabelecimento de mais de um critério de seleção de retransmissores, o que torna a participação dos nós na transmissão de pacotes mais distribuída em comparação ao que acontece com

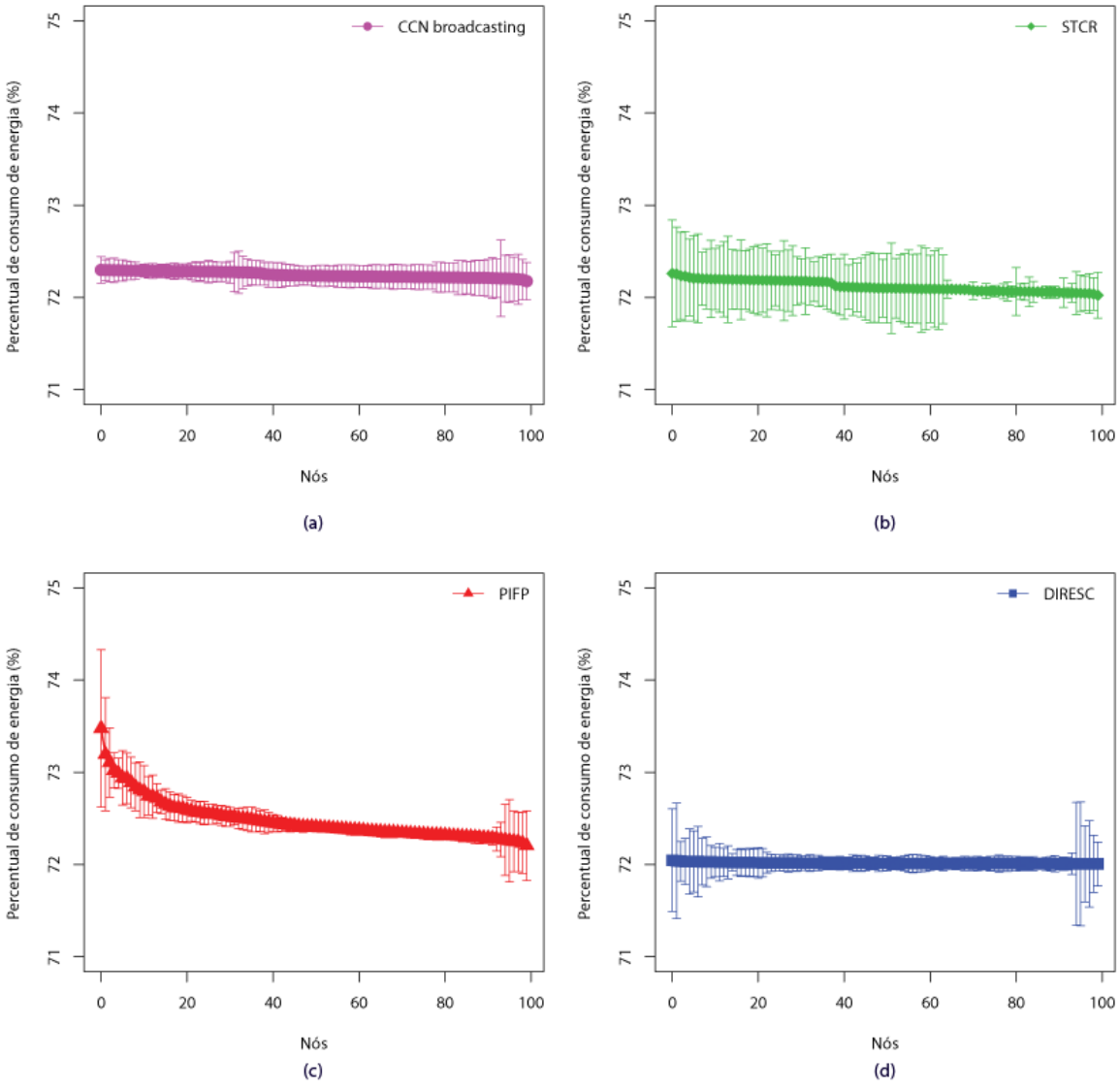


Figura 5.9: Consumo percentual de energia no cenário de Helsinki

o uso dos demais protocolos.

Assim, observamos que o desgaste energético gerado pelo protocolo DIRESC apresenta uma pequena variação em torno da média, permitindo-nos inferir que nosso protocolo gera uma participação quase que idêntica dos nós na rede.

Nota-se, por outro lado, que o protocolo STCR, que provoca o problema da "escolha de nós preferidos", apresenta uma percentagem considerável de nós, aproximadamente 40%, que têm um consumo de energia relativamente superior aos demais e com elevada variação. O que nos permite supor que se tratam dos nós de maior centralidade social e, portanto, mais participativos, o que evidencia um maior consumo de recursos desses mesmos nós quando a rede utiliza o referido protocolo.

De modo semelhante, o PIFP também apresenta aproximadamente 40% dos nós com um consumo de energia superior à percentagem restante, e com grande variação. Essa observação nos leva a entender que aquela quantidade de nós corresponde aos que apresentam maior probabilidade de encontrar conteúdos, e, portanto, nós também de maior centralidade e bem relacionados na rede (ainda que tais conceitos não sejam abordados em [30]). No que concerne ao consumo gerado pela outra parcela de nós, essa ainda gera um desgaste energético superior ao apresentado pelo nosso protocolo.

Com um comportamento quase tão equilibrado quanto o DIRESC, o *CCNbroadcasting* apresenta um gasto energético de baixa variação. Isso acontece em razão do comportamento quase que invariável do protocolo para várias execuções de simulação com o mesmo cenário proposto. Ou seja, o mesmo não avalia características inerentes aos nós, mas efetua o envio de mensagens em *broadcasting*, padrão de seu funcionamento.

Assim, na Tabela 5.9, apresentamos um resumo acerca dos valores remanescentes de energia, que ratifica a discussão anteriormente realizada, revelando que o DIRESC apresenta o menor consumo energético na rede dentre todos os protocolos, bem como a menor variação em relação à média calculada.

Tabela 5.9: Consumo percentual de energia no cenário de Helsinki

| **Protocolo** | **Máximo** | **Mínimo** | **Média** | **Desvio padrão** |
|---|---|---|---|---|
| *CCNbroadcasting* | 72,30% | 72,18% | 72,25% | 0,031 |
| STCR | 72,27% | 72,03% | 72,13% | 0,060 |
| PIFP | 73,48% | 72,21% | 72,49% | 0,226 |
| DIRESC | 72,05% | 72,01% | 72,02% | 0,007 |

### 5.4.2 Cenário de São Francisco

Para este cenário, além das configurações apresentadas na Seção 5.2.2, utilizou-se intervalos de amostragem de 15 minutos, nos quais são gerados aproximadamente, em cada um deles, 60 pacotes de interesse por meio do trace abordado na Seção 5.1.1

**Taxa de dados entregues**

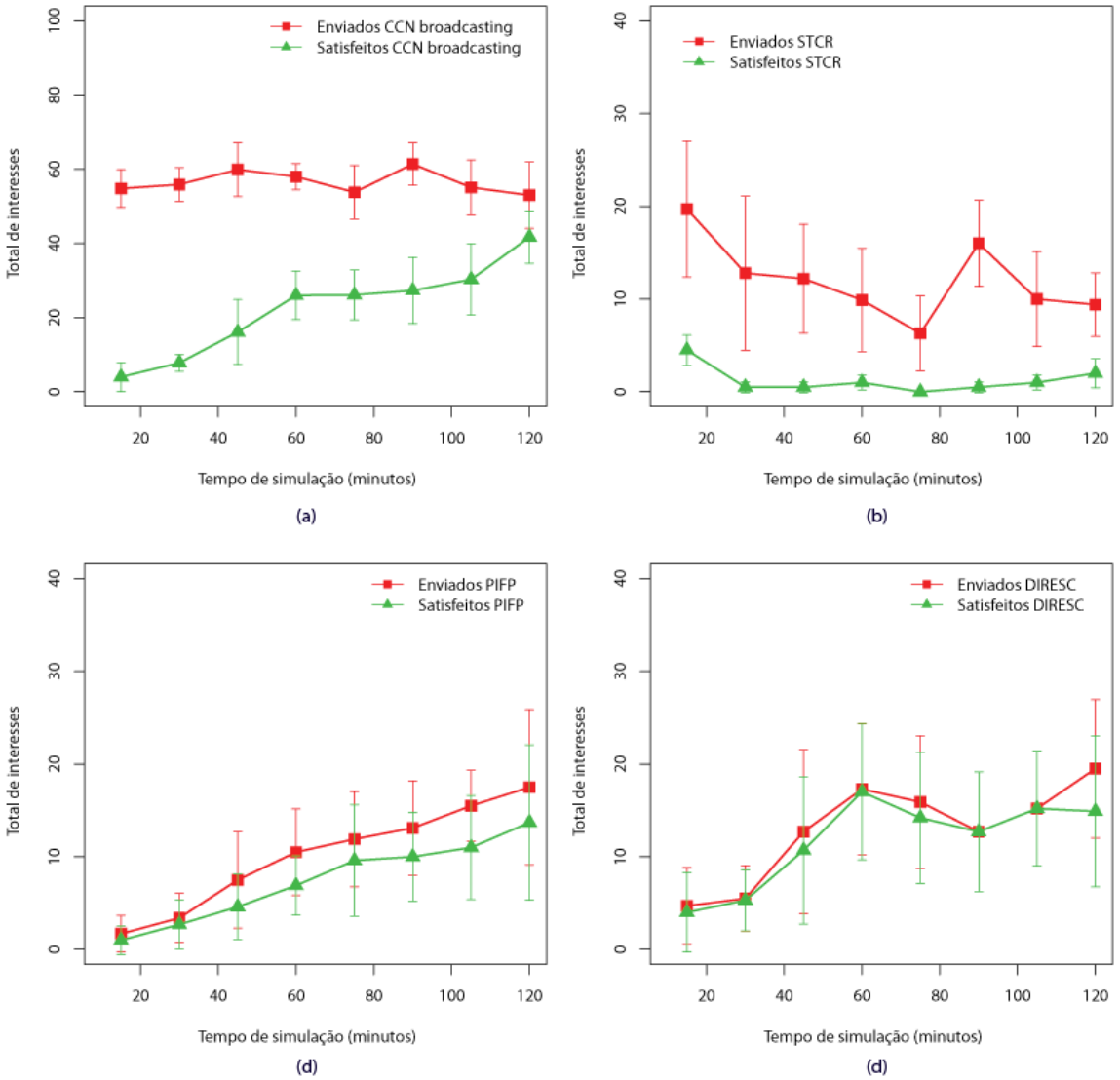


Figura 5.10: Taxa absoluta de dados entregues no cenário de São Francisco

Do mesmo modo como nos gráficos de Helsinki, a Figura 5.10 apresenta a taxa absoluta de dados entregues (em verde) para cada protocolo e as curvas referentes ao envio de pacotes de interesse (em vermelho).

No cenário de São Francisco, observamos um comportamento razoavelmente seme-

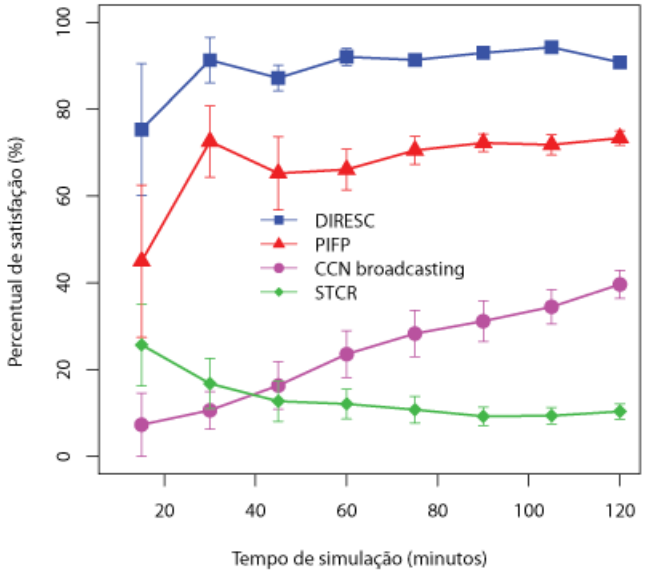


Figura 5.11: Taxa percentual de dados entregues no cenário de São Francisco

lhante de alguns protocolos em relação ao cenário anterior. O *CCNbroadcasting*, por exemplo, mantém, até o instante final de simulação, a taxa de envios de interesses bem próxima do montante gerado a cada intervalo. Isso acontece, primeiramente, porque a quantidade de pacotes de interesse criados a partir do *trace* (ver Seção 5.1.1) é menor nesse período, bem como o tempo total de simulação. Assim, temos um comportamento semelhante ao observado nas duas primeiras horas do cenário de Helsinki. Em relação a taxa absoluta de dados entregues resultante desse protocolo, obervamos o seu crescimento gradual até o fim do tempo de simulação, chegando próximo de atingir o total de requisições enviadas a cada intervalo. A observação isolada dessa janela de tempo de simulação nos permite inferir que, em tempos futuros, o funcionamento do protocolo resultaria em uma taxa percentual de dados entregues que possivelmente atingiria o valor máximo. Contudo, o estudo de caso anterior nos leva a supor também que essa curva poderia crescer até certo ponto e, em seguida, apresentar um decaimento. Assim, para o tempo de simulação proposto, temos o comportamento percentual desta métrica para o *CCNbroadcasting* como apresentado na Figura 5.11 (em roxo).

Por sua vez, o STCR apresenta um comportamento de envio de requisições diferente do obtido anteriormente. No cenário de São Francisco, além de uma densidade menor, tem-se nós que se movem com maior velocidade e, portanto, menores oportunidades de contato. Assim, as chances de atender as especificidades do critério de envio de interesses são reduzidas. Em razão disso, observa-se não apenas o decaimento no total dessas transmissões a cada intervalo, mas também o lento crescimento da taxa absoluta de dados entregues e, consequentemente, o comportamento percentual apresentado na Figura 5.11

(em verde).

Os protocolos PIFP e DIRESC, como se sabe, demandam certo tempo até obter as informações necessárias para alimentar os mecanismos de tomada de decisão para envio de pacotes. Portanto, observa-se que ambos apresentam inclinações de crescimento, mas que, na janela de tempo proposta, ficam bem abaixo do montante de interesses gerados a cada intervalo. Contudo, nota-se que o DIRESC alcança determinado valor, tanto no envio quanto na satisfação de interesses, com maior rapidez que o PIFP.

Nesse cenário, nota-se um expressivo ganho do DIRESC na taxa percentual de dados entregues em relação aos demais protocolos. Isso acontece porque, no ambiente proposto, são observadas menores oportunidades de contato e de troca de mensagens entre os nós devido a sua elevada mobilidade e a ampla área de deslocamento dos mesmos, de modo que o referido cenário é considerado de baixa densidade. Assim, quanto menor a tolerância para o envio de mensagens, menores são as chances de que as mesmas sejam entregues e, portanto, que o processo de requisição e entrega seja finalizado com sucesso. Uma vez que o DIRESC apresenta maior tolerância no mecanismo de seleção de nós retransmissores, o mesmo atinge maiores chances de descoberta e recuperação de dados que os demais protocolos. No que diz respeito ao *CCNbroadcasting* que, dentre todos os protocolos, apresenta a maior tolerância no envio de mensagens, o baixo percentual de satisfação acontece devido ao *overhead* causado na rede e, consequentemente, ao elevado índice de recusas no recebimento de pacotes.

A Tabela 5.10 exibe um resumo dos valores obtidos para a taxa absoluta de dados entregues sobre a média aritmética das 30 simulações para cada protocolo, e revela o melhor desempenho do *CCNbroadcasting* para a referida métrica em relação aos demais.

Tabela 5.10: Taxa absoluta de dados entregues no cenário de São Francisco

| **Protocolo** | **Mínimo** | **Máximo** | **Média** | **Desvio padrão** |
|---|---|---|---|---|
| *CCNbroadcasing* | 4 | 41,7 | 22,41 | 12,40 |
| STCR | 0 | 4,5 | 1,25 | 1,43 |
| PIFP | 1 | 13,7 | 7,43 | 4,39 |
| DIRESC | 4 | 17 | 11,75 | 4,76 |

Além disso, observamos um ganho do *CCNbroadcasting* na taxa absoluta de dados entregues em relação a todos os demais protocolos. Contudo, quando consideramos a quantidade de interesses enviados por requisitantes primários a cada intervalo e calculamos a taxa percentual, verificamos que o DIRESC assume essa posição de ganho em relação aos outros protocolos, como exibe a Tabela 5.11.

Tabela 5.11: Taxa percentual de dados entregues no cenário de São Francisco

| Protocolo | Mínimo | Máximo | Média | Desvio padrão |
|---|---|---|---|---|
| *CCNbroadcasing* | 7,33 | 39,66 | 23,95 | 11,58 |
| STCR | 9,27 | 25,72 | 13,40 | 5,53 |
| PIFP | 45 | 73,35 | 67,10 | 9,42 |
| DIRESC | 75,33 | 94,26 | 89,40 | 6,04 |

### Atraso médio de descoberta e recuperação

Em relação ao atraso médio de descoberta e recuperação de dados apresentada na Figura 5.12, observamos que o protocolo *CCNbroadcasting* (em roxo) apresenta o maior valor para essa métrica em todos os intervalos de amostragem. Isso é resultado da soma de fatores, internos e externos, que levam a tal comportamento, como o envio indiscriminado de pacotes na rede, o que resulta no consumo acelerado da capacidade de armazenamento, gerando as, anteriormente mencionadas, recusas no recebimento de pacotes. Assim, tanto pacotes de interesse quanto de dados podem ser amplamente rejeitados ao longo dos intermitentes caminhos entre origem e destino. Além disso, as características inerentes no ambiente veicular contribuem fortemente para o referido atraso.

O STCR (em verde), por sua vez, apresenta uma redução drástica do atraso médio em relação ao obtido no cenário anterior. Pelo menos é o que se pode constatar ao ser considerada apenas a janela de tempo proposta para as simulações com o mapa de São Francisco, pois, no estudo de caso anterior, considerando o tempo total de simulação, esse

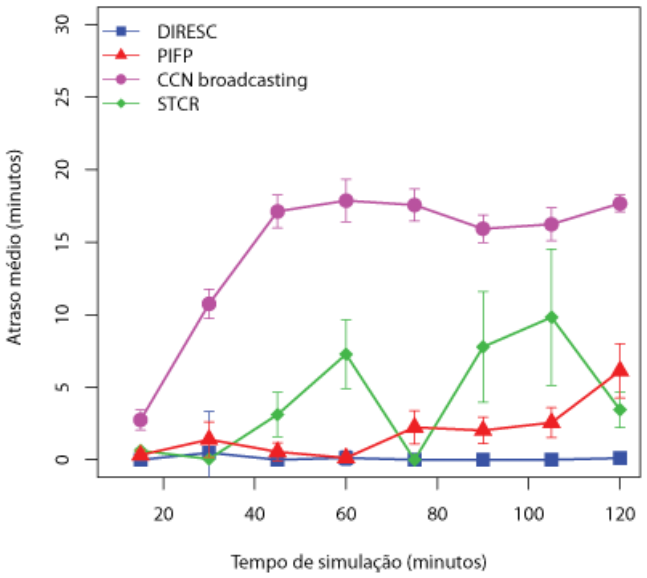


Figura 5.12: Atraso médio no tempo no cenário de São Francisco

protocolo gerou o maior atraso. No entanto, ao se analisar apenas as duas primeiras horas, nota-se que o STCR de fato exibiu um atraso menor que o do *CCNbroadcasting*.

No que diz respeito ao PIFP (em vermelho), observa-se que ele apresenta uma variação de crescimento a partir do início da segunda hora. Esse comportamento pode ser justificado pelo descarte de pacotes de dados das caches dos nós intermediários em razão do término de seu tempo de vida e, portanto, pela atualização do cálculo de probabilidade desses mesmos nós entrarem em contato com tais conteúdos, os quais terão de ser novamente recuperados a partir dos provedores primários até que sejam novamente replicados nas caches ao longo da rede. Observa-se que, assim como no cenário anterior, o protocolo apresenta um crescimento semelhante quando atinge a segunda hora de simulação.

Por fim, o DIRESC (em azul), dentre todos os protocolos, apresenta o menor atraso observado. Isso acontece, pois, como explicado anteriormente, uma vez que é definida a política de prioridades para o envio de pacotes pelo protocolo, as mensagens de interesse e dados tendem a ser transmitidas, nos primeiros instantes de simulação, para nós que têm maiores chances de finalizar o processo de descoberta e recuperação de dados, de modo que é obtido um baixo atraso para a conclusão do mesmo. Nos intervalos seguintes, o valor da métrica é mantido na mesma escala em razão das replicações de dados nas caches dos requisitantes e, ainda que descartados pelo término de seu tempo de vida, as informações sobre esses mesmos dados ainda se mantém ativas na rede, ou seja, os registros de FIB sobre prefixos de conteúdo anunciados por produtores primários, os agrupamentos de nós que mantém tais registros e os valores de relacionamentos socias não são atualizados ou recalculados em razão do descarte de dados das caches pelo término de seu tempo de vida.

A Tabela 5.12 exibe um resumo dos valores obtidos para o atraso médio de descoberta e recuperação (em minutos) sobre a média aritmética das 30 simulações para cada protocolo, no qual se confirma que o DIRESC resulta em um menor atraso e, portanto, maior rapidez para a obtenção dos dados solicitados em comparação com as demais abordagens.

Tabela 5.12: Atraso médio de descoberta e recuperação (em minutos) no cenário de São Francisco

| **Protocolo** | **Mínimo** | **Máximo** | **Média** | **Desvio padrão** |
|---|---|---|---|---|
| *CCNbroadcasing* | 2,74 | 17,86 | 14,48 | 5,27 |
| STCR | 0 | 9,82 | 4,01 | 3,83 |
| PIFP | 0,13 | 6,13 | 1,92 | 1,93 |
| DIRESC | 0,003 | 0,48 | 0,09 | 0,16 |

**Sobrecarga na rede (*Overhead*)**

No que concerne à métrica de sobrecarga de pacotes de interesse apresentada na Figura 5.13 (a), observamos que, para os protocolos *CCNbroadcasting* e STCR, o resultado obtido é bem semelhante ao apresentado no cenário anterior. As razões pelas quais esses comportamentos são obtidos, como explicados outrora, são: o elevado número de replicações de mensagens feitas pelo protocolo de característica epidêmica, o que enche o buffer dos nós intermediários e eleva o número de retransmissões, de modo que tais mensagens apenas deixam de ser recebidas pelo esgotamento da capacidade de armazenamento ou por duplicidade. No caso do STCR, a quantidade das referidas mensagens que de fato são entregues ao destinatário é ínfima comparada àquela que é (re)transmitida na rede, resultando no comportamento observado da métrica.

No *overhead* de interesses do cenário de São Francisco, nosso protocolo apresenta uma perda em alguns pontos para o PIFP. Explicamos esse resultado negativo pela garantia da maior liberdade no envio daqueles pacotes por parte do DIRESC. Tal característica, que para outras métricas apresentou resultados satisfatórios, para a sobrecarga de interesses resultou no comportamento observado. O PIFP mostra maior regularidade no comportamento dessa métrica uma vez que seu funcionamento se baseia unicamente no cálculo de probabilidade de um nó entrar em contato com determinado conteúdo, o que pode manter a baixa variação por um determinado período.

Quanto ao *overhead* de dados apresentado na Figura 5.13 (b), tanto *CCNbroadcasting* quanto STCR exibem altos valores a cada intervalo. No caso do primeiro, isso acontece em razão do comportamento epidêmico do protocolo, como explicado anteriormente, o que eleva consideravelmente o número de réplicas destas mensagens e, consequentemente, de suas retransmissões em comparação ao montante que é entregue aos requisitantes. No

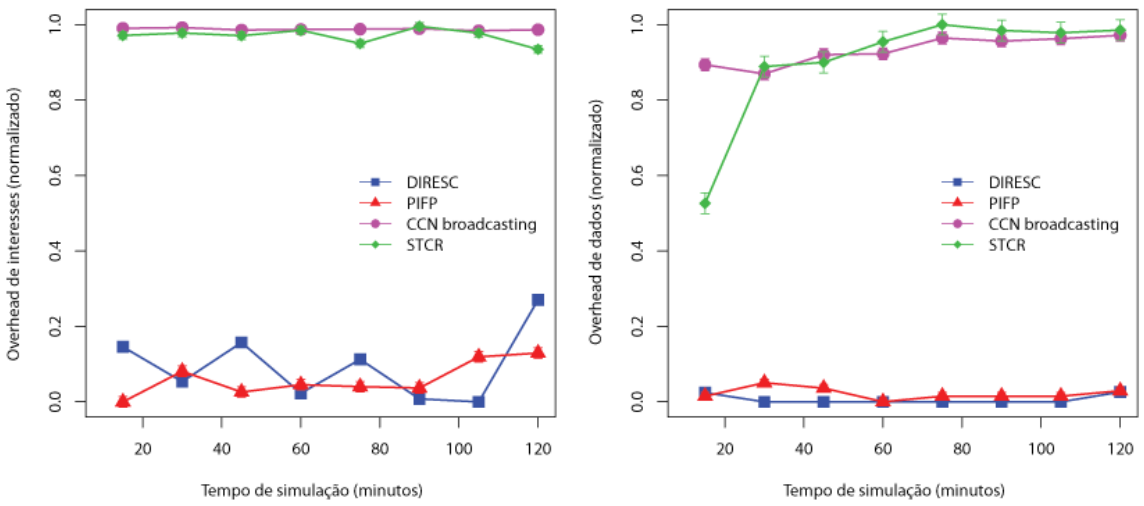


Figura 5.13: Overhead de interesses e dados no cenário de São Francisco

caso do segundo, observa-se um alto índice de sobrecarga que é resultante da baixa taxa de entrega de dados aos requisitantes, a qual resulta da expectativa do protocolo de que apenas contatos entre nós de alto relacionamento social seja capaz de concluir o processo de recuperação de dados, quando é sabido que, em redes tolerantes a atraso, grandes são as possibilidades de um par de nós não se reencontrar na rede. Assim, como mencionado anteriormente, o STCR falha em não fazer uma utilização funcional do componente PIT da arquitetura CCN, o qual possibilitaria a existência de diferentes caminhos para a entrega de dados.

A prova de tal afirmação é o comportamento do PIFP na referida métrica. Esse protocolo não implementa nenhum mecanismo de entrega de dados e, portanto, basei-se unicamente nas "*bread crumbs*" ("migalhas de pão") [4] registradas em PIT nos caminhos de requisição para executar a recuperação de tais dados. Ou seja, para entregá-los, o protocolo conta, exclusivamente, com as informações de "quem são os requisitantes de determinado conteúdo" armazenadas na PIT dos nós que intermediam a comunicação entre solicitante e provedor, conforme os pacotes de interesse são enviados. Por essa razão, o protocolo atinge um baixo índice de sobrecarga de dados, uma vez que interesses são enviados para nós com maiores chances de contato com o conteúdo requisitado e, por consequência, tem-se uma baixa quantidade de saltos para a entrega do conteúdo solicitado.

No entanto, depender exclusivamente de um mecanismo ou outro, em uma rede de alta variação e intermitência como DTN, é uma política arriscada, pois, se porventura o mesmo mecanismo apresenta falha por um evento externo, a entrega de pacotes é comprometida e o protocolo deixa de cumprir com o seu objetivo de roteamento. Por exemplo, se um ou mais nós no caminho de requisição, por alguma razão, deixam o cenário.

Por essa razão, nosso protocolo faz uso de ambos: o componente PIT, como aquele que mantém ativa a informação de quem são os requisitantes; e os relacionamentos sociais, que podem auxiliar a entrega de mensagens. Assim, observamos que o DIRESC apresenta um baixo *overhead* pois, assim como o PIFP, utiliza o caminho de requisição como meio para entregar os dados, e também faz com que provedores os entregem a todo nó que possui um registro na PIT interessado no mesmo conteúdo, o que ajuda a dimunir não apenas o atraso, como observado anteriormente (ver Figura 5.12), mas também o envio destes dados para nós que atuariam apenas como retransmissores, uma vez que a política de prioridade garante a transmissão primeiramente para nós de fato interessados em obter tais dados.

As Tabelas 5.13 e 5.14 exibem um resumo dos valores obtidos para a sobrecarga normalizada de interesses e dados, respectivamente, sobre a média aritmética das 30 simu-

Tabela 5.13: Sobrecarga normalizada de interesses no cenário de São Francisco

| Protocolo | Mínimo | Máximo | Média | Desvio padrão |
|---|---|---|---|---|
| *CCNbroadcasing* | 0,98 | 0,99 | 0,98 | 0,002 |
| STCR | 0,93 | 0,99 | 0,97 | 0,01 |
| PIFP | 0 | 0,12 | 0,05 | 0,04 |
| DIRESC | 0 | 0,27 | 0,09 | 0,09 |

lações para cada protocolo. A primeira revela que o DIRESC, apesar de obter um baixo *overhead* de interesses em comparação ao *CCNbroadcasting* e ao STCR, acaba perdendo para o protocolo PIFP no desempenho desta métrica. A segunda, contudo, evidencia que o nosso protocolo resulta no menor índice de sobrecarga de dados quando comparado a todos os protocolos estudados.

Tabela 5.14: Sobrecarga normalizada de dados no cenário de São Francisco

| Protocolo | Mínimo | Máximo | Média | Desvio padrão |
|---|---|---|---|---|
| *CCNbroadcasing* | 0,86 | 0,97 | 0,93 | 0,03 |
| STCR | 0,52 | 1 | 0,90 | 0,15 |
| PIFP | 0,0002 | 0,05 | 0,022 | 0,015 |
| DIRESC | 0,0001 | 0,02 | 0,006 | 0,011 |

### Média de pacotes retransmitidos por nó

A Figura 5.14 apresenta a média de pacotes retransmitidos por nó da rede a cada intervalo, de modo que o gráfico (b) representa uma aproximação do (a), para melhor visualiza-

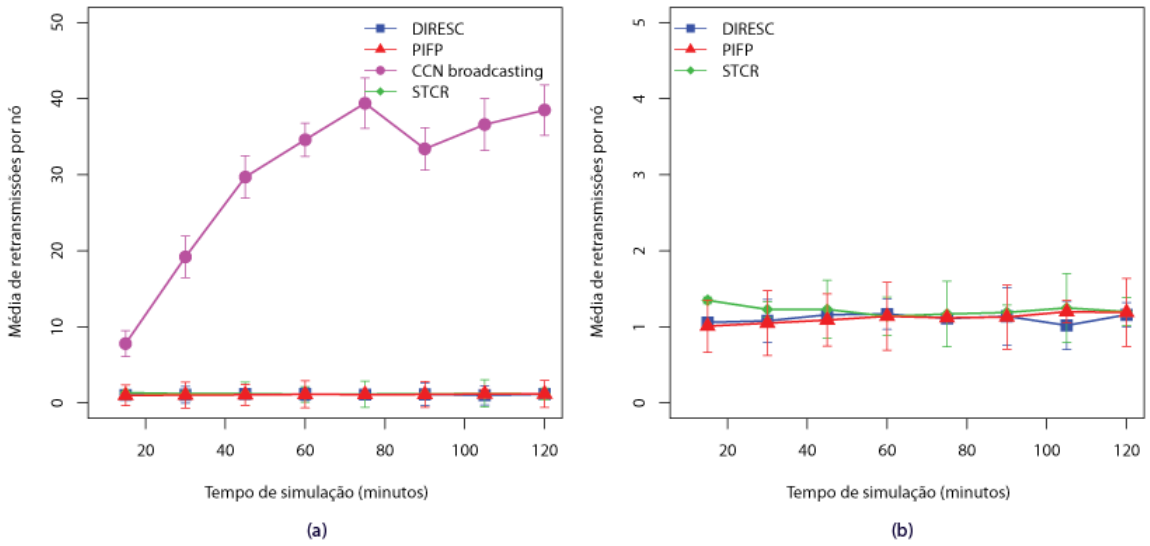


Figura 5.14: Média de retransmissões por nó a cada intervalo no cenário de São Francisco

Tabela 5.15: Pacotes retransmitidos por nó no cenário de São Francisco

| **Protocolo** | **Mínimo** | **Máximo** | **Média** | **Desvio padrão** |
|---|---|---|---|---|
| *CCNbroadcasing* | 7,8 | 39,4 | 29,9 | 10,99 |
| STCR | 1,14 | 1,35 | 1,22 | 0,06 |
| PIFP | 1,01 | 1,2 | 1,116 | 0,06 |
| DIRESC | 1,02 | 1,17 | 1,112 | 0,05 |

ção do comportamento de três dos protocolos. Por meio deles, observa-se claramente que, por não ter qualquer critério para transmitir mensagens, o *CCNbroadcasting* apresenta um elevado valor para essa métrica, de modo que na maioria dos intervalos nota-se apenas o seu crescimento. Esse comportamento é prejudicial para a rede em termos de exaustão de recursos, o que gera uma elevada sobrecarga de pacotes e um alto atraso devido ao descarte de pacotes resultantes de tal exaustão.

Os três protocolos restantes, apresentados com maior precisão na Figura 5.14 (b), exibem um valor relativamente próximo para a métrica em questão, variando em torno de uma unidade ao longo de toda a simulação. No caso do STCR, o resultado obtido é fruto do baixo índice de envio de pacotes na rede, combinado às poucas oportunidades de retransmissão dadas aos nós devido à especificidade dos critérios definidos pelo protocolo.

O PIFP apresenta o comportamento mostrado em razão das elevadas chances de que os conteúdos sejam recuperados com o menor número de saltos quando interesses são transmitidos, isso porque estes apenas são enviados para nós que apresentem a maior probabilidade de responder às requisições. Portanto, o protocolo resulta em um baixo número de retransmissões tanto de interesses quanto de dados.

Por sua vez, o DIRESC também apresenta uma baixa média de retransmissões de pacotes. Isso acontece em razão da distribuição de informação sobre prefixos na rede, de modo que sejam formados agrupamentos de nós que rapidamente (com poucos saltos para descoberta e recuperação) podem entregar determinado conteúdo devido ao seu alto relacionamento social com os produtores que formam tais agrupamentos. Além disso, a replicação dos dados correspondentes ao longo da rede facilita a sua obtenção uma vez que os aproxima dos requisitantes.

A Tabela 5.15 exibe um resumo dos valores obtidos para a média de pacotes retransmitidos por nó, sobre a média aritmética das 30 simulações para cada protocolo, revelando mais uma vez que o nosso protocolo consegue controlar o número de envios de pacotes entre os nós, resultando na menor média de retransmissão para cada participante da rede.

### Consumo de energia

Em relação ao consumo médio de energia apresentado nos gráficos da Figura 5.15, observamos que, assim como no cenário anterior, o protocolo PIFP apresenta a maior variação da referida métrica para uma pequena parcela de nós que também apresenta o maior desgaste energético, de modo que a quantidade restante (aproximadamente 80% dos nós) corresponde ao menor consumo desse recurso e variação em torno da média. Podemos inferir que esses se tratam de nós de menor participação na rede, ou seja, que apresentam baixa probabilidade de contato com conteúdos; enquanto aqueles, consomem mais energia pois participam mais dos processos de descoberta e recuperação de dados devido sua maior probabilidade de encontrar tais conteúdos.

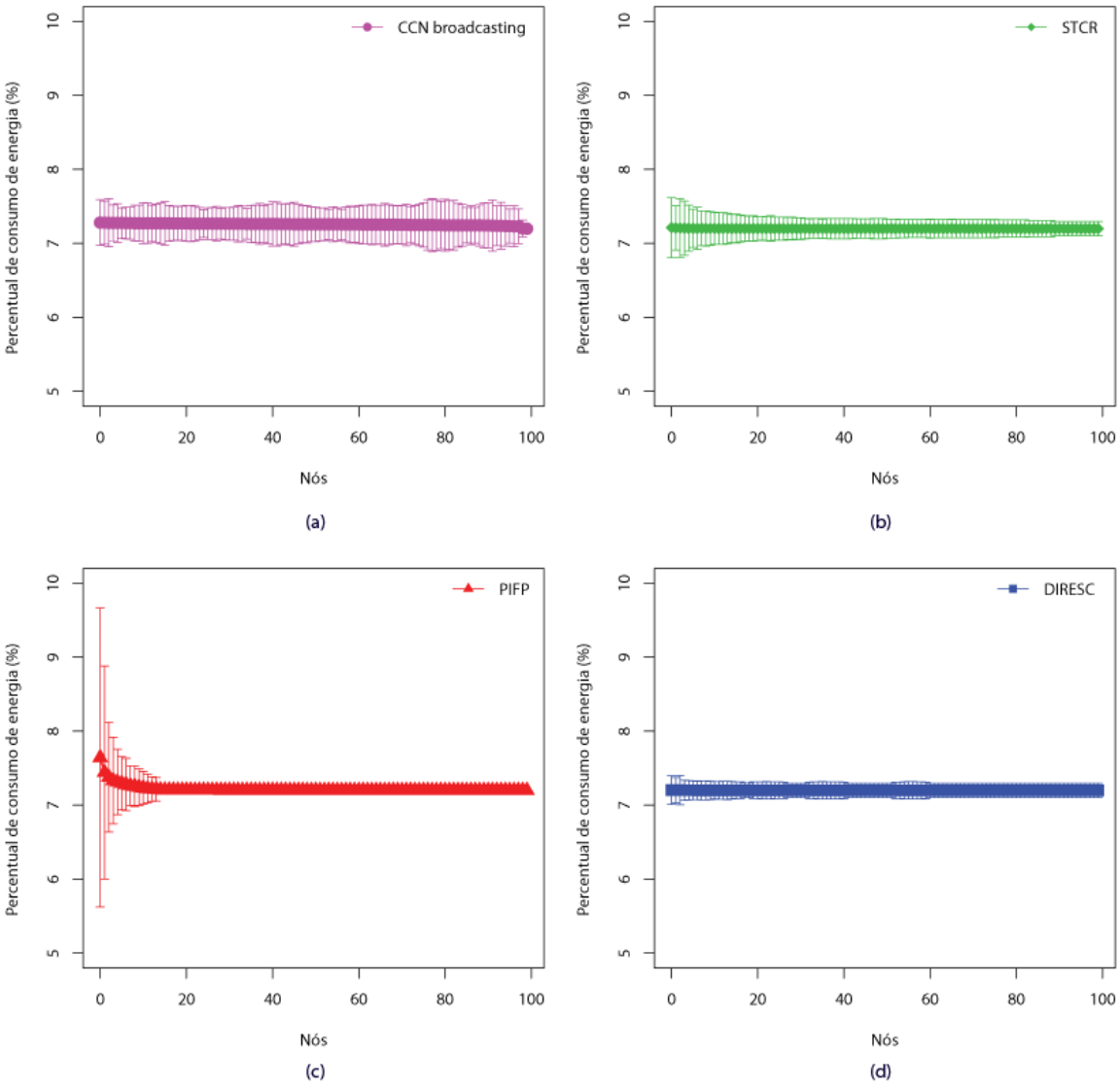


Figura 5.15: Consumo percentual de energia no cenário de São Francisco

O STCR exibe um comportamento no qual, em média, todos os nós apresentam aproximadamente o mesmo consumo de energia, de modo que cerca de 10% deles exibem uma maior variação em torno dessa média. Esse desgaste energético relativamente constante e de baixa variação para a maioria dos nós é resultado da pequena participação dos mesmos na recuperação de conteúdo. Isto é, como apresentado nas métricas anteriores, no cenário de São Francisco, observa-se que o protocolo permite o envio de uma baixa quantidade de interesses e, consequentemente, de dados, com uma média de retrasmissão de pacotes por nó também baixa, o que resulta na menor participação dos nós na rede e, portanto, do comportamento obtido na métrica de energia.

Por sua vez, o *CCNbroadcasting* apresenta grande variação em torno da média resultante para a referida métrica. Contudo, nota-se que essa variação se apresenta bem distribuída para todos os nós da rede. Isso é resultado da elevada participação que todos eles assumem nos processos de descoberta e recuperação de dados.

Por fim, cumprindo com o objetivo de tornar os nós mais participativos na rede, de forma que fosse oferecida maior tolerância de envio de mensagens para que não se repetisse o problema de "escolha de nós preferidos", o DIRESC apresenta a menor variação em torno do consumo médio de energia, de modo que é possível inferir que todos os nós apresentam aproximadamente o mesmo índice de desgaste energético quando atuam por meio desse protocolo, comprovando que os mecanismos de tomada de decisão propostos são eficazes para gerar alto índice de satifação de requisições, com equilibrada distribuição de mensagens e participação igualitária na rede.

Assim, na Tabela 5.16, apresentamos um resumo acerca dos valores remanescentes de energia, que ratifica a discussão anteriormente realizada, revelando que o DIRESC apresenta o menor consumo energético na rede dentre todos os protocolos, bem como a menor variação em relação à média calculada.

Tabela 5.16: Consumo percentual de energia no cenário de São Francisco

| **Protocolo** | **Máximo** | **Mínimo** | **Média** | **Desvio padrão** |
|---|---|---|---|---|
| *CCNbroadcasting* | 7,29% | 7,2% | 7,26% | 0,014 |
| STCR | 7,22% | 7,2% | 7,21% | 0,002 |
| PIFP | 7,65% | 7,2% | 7,22% | 0,056 |
| DIRESC | 7,21% | 7,2% | 7,21% | 0,0006 |

# 6. Conclusão e trabalhos futuros

Neste trabalho, tratamos da recuperação de dados em Redes Centradas em Conteúdo e Tolerantes a Atraso (CCDTN). Para tanto, apresentamos algumas abordagens da literatura que empregam a arquitetura de Redes Centradas em Conteúdo (CCN) em ambientes nos quais as comunicações entre dispositivos são Tolerantes a Atraso (DTN), para a obtenção de dados ou serviços. Explicamos os principais conceitos referentes aos paradigmas supracitados, e expusemos como problema a lacuna existente naquelas aborgadens quando, em seus mecanismos, utilizam a estratégia de seleção de nós retransmissores, de modo que os critérios definidos provocam o problema apresentado por [32], denominado "escolha de nós preferidos".

A partir disso, elaboramos nossa proposta e definimos o DIRESC, um protocolo para descoberta e recuperação de dados em redes centradas em conteúdo e tolerantes a atraso baseado em cooperação social, cujos principais objetivos são: elevar a taxa de entrega de dados e, portanto, de satisfação de requisições; reduzir o atraso e o *overhead*; e tornar mais distribuída a participação dos nós na rede, de modo a evitar o problema anteriormente apresentado, bem como reduzir o desgaste de recursos, como capacidade de armazenamento e energia, oriundo do mesmo.

Assim, diferentemente das abordagens propostas na literatura, o DIRESC define três possibilidades de transmissão de mensagens para a descoberta e recuperação de dados, aumentando as chances de se elevar a taxa de entrega de pacotes e reduzir o atraso, não pelo seu envio para os nós mais bem posicionados em um ranqueamento definido por um único critério de seleção, mas pela distribuição de diferentes tipos de *bread crumbs* ao longo da rede, gerados pelas variadas possibilidades de escolha de retransmissores, de modo que se possa fazer o melhor uso dos múltiplos caminhos sugeridos por DTN para a entrega de mensagens. Para controlar o *overhead* na rede, o DIRESC define também um limitador de submissões e um esquema de prioridade de envios de pacotes para casos específicos de transmissão.

Como princípio fundamental, nosso protocolo utiliza o fator de relacionamento social, proposto em [29], para efetuar as tomadas de decisão que definem o comportamento da rede em três fases distintas: anúncio, descoberta e recuperação.

Uma das vantagens do mecanismo de seleção de retransmissores proposto no DIRESC é a produção de uma quantidade de informações mais robusta e consistente acerca das requisições feitas, de modo que nós que possam entregar conteúdos não precisem aguardar o acontecimento de condições muito isoladas e específicas para realizar ou atender às socilitações, como por exemplo, a entrega de dados considerando apenas o caminho inverso pelo qual certa requisição foi enviada, ou por meio do eventual encontro do provedor de conteúdo com nós amigos do requisitante. Portanto, o referido protocolo emprega em seu funcionamento o conhecimento produzido acerca dos relacionamentos sociais entre nós calculados localmente nos mesmos e todas as informações presentes nos componentes básicos da arquitetura CCN (CS, PIT e FIB). Deste modo, o DIRESC garante que a entrega de dados não seja dependente de uma única informação que possa ser facilmente perdida e da qual o protocolo necessita para operar, mas que seja realizada mediante um conjunto delas, as quais se desdobram em diferentes meios de satisfazer a solicitação.

Para constatar o funcionamento satisfatório e validar o DIRESC perante a literatura existente, avaliamos o seu desempenho em dois cenários distintos: um ambiente de mobilidade sintética, com o mapa da cidade de Helsinki (Finlândia); e outro de mobilidade real, com o mapa da cidade de São Francisco (EUA) e cujo *dataset* está disponível em [34]. O primeiro agregando diferentes tipos de nós, como pedestres, carros e bondes, enquanto o segundo, apenas carros. Para reproduzir esses cenários, utilizamos o simulador ONE (*Opportunistic Network Environment*) [33], no qual implementamos a arquitetura CCN para que fosse possível a ele representar o contexto de rede no qual nosso trabalho está ambientado. Codificamos também o funcionamento padrão de uma CCN, o qual intitulamos de *CCNbroadcasting*, em razão do modo como as comunicações acontecem através dele. Por fim, codificamos no simulador o nosso protocolo proposto.

Os resultados de simulação do DIRESC foram comparados aos de outros três protocolos da literatura: o *CCNbroadcasting*; o também baseado em relacionamentos sociais STCR; e o probabilístico PIFP. Confrontamos os resultados obtidos com o DIRESC aos destes protocolos para que pudessemos comprovar que o nosso, estando em um meio termo entre o comportamento epidêmico e um extremamente criterioso, refletiria os benefícios de ambos. Portanto, sendo os três protocolos excelentes representações de tais comportamentos, optamos por escolhê-los para nossa avaliação.

Para a comparação dos protocolos, definimos cinco métricas de avaliação: taxa de da-

dos entregues (satisfação); atraso médio de descoberta e recuperação; sobrecarga na rede (*overhead*); média de pacotes retransmitidos por nó; e consumo de energia. Os gráficos gerados por meio da ferramenta estatística R *Software* [38] apresentam ganhos expressivos do DIRESC em relação aos protocolos *CCNbroadcasting* e STCR em todas as métricas avaliadas. Para esses protocolos, os ganhos chegam a, pelo menos, 17.1% e 73.2% na taxa de dados entregues nos cenários de mobilidade sintética e real, respectivamente. Por outro lado, quando comparado ao PIFP, alguns dos ganhos observados são mais discretos, como os obtidos também na taxa de dados entregues, que atingem 2.36% e 24.9% nos cenários supracitados, respectivamente. Como resultado negativo, nosso protocolo perde para o probabilístico no *overhead* de interesses no cenário de São Francisco. Apesar disso, o DIRESC se mostra vantajoso e eficaz para a descoberta e recuperação de dados de forma ágil e sem grandes prejuízos à rede, cumprindo os objetivos propostos em sua definição.

Futuramente, pretendemos aprimorar o funcionamento do DIRESC utilizando mecanismos semelhantes aos propostos nos chamados *sistemas de recomendação* de dados, para refinar o relacionamento social entre os nós. Isto é, usar a similaridade das requisições dos nós como um fator de influência para o relacionamento entre eles.

Outra possibilidade de trabalho futuro concerne aprofundar os estudos em relação aos aspectos de segurança que permeiam CCN, principalmente na entrega de dados, e fazer uso dos campos que existem nestes pacotes, segundo a definição da arquitetura, e que remetem tais aspectos.

Além disso, planejamos também agregar ao nosso protocolo uma política de substituição de dados em cache, tópico de considerada importância no estudo de CCN empregada em DTN. Primeiramente, visamos utilizar esquemas propostos na literatura, como LRU (*Least Recently Used* ou Menos Recentemente Usado) e LFU (*Least Frequently Used* ou Menos Frequentemente Usado), entre outros, a fim de verificar qual se adequa melhor ao DIRESC. Em seguida, faremos os ajustes necessários com base no funcionamento do protocolo de modo que possamos verificar como ele se comporta ao empregar sua própria política de gestão do recurso de cache.

Por fim, pretendemos avaliar o funcionamento do DIRESC quando os parâmetros que ele define, como a variação do $R_{medio}$ ($\Omega$) e o limitador de submissões ($\mu$), assumem diferentes valores, de modo que possamos comparar o seu funcionamento em diferentes cenários.

# Referências Bibliográficas


[1] M. Conti, C. Boldrini, S. S. Kanhere, E. Mingozzi, E. Pagani, P. M. Ruiz, and M. Younis, "From MANET to people-centric networking: Milestones and open research challenges," *Computer Communications*, vol. 71, pp. 1–21, 2015.

[2] P. Santi, *Mobility Models for Next Generation Wireless Networks: Ad Hoc, Vehicular and Mesh Networks*. Wiley Publishing, 1st ed., 2012.

[3] M. Amadeo, C. Campolo, A. Molinaro, and G. Ruggeri, "Content-centric wireless networking: A survey," *Computer Networks*, vol. 72, pp. 1–13, 2014.

[4] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *5th CoNEXT*, pp. 1–12, ACM, 2009.

[5] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos, *et al.*, "Named data networking (ndn) project," *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, 2010.

[6] CCNx, "The CCNx project." Available at: `https://wiki.fd.io/view/Cicn` (accessed: 2017-06-20), 2009.

[7] NFD, "Named Data Networking Forwarding Daemon.." Available at: `http://named-data.net/doc/NFD/current/overview.html` (accessed: 2017-06-20), 2013.

[8] X. Liu, Z. Li, P. Yang, and Y. Dong, "Information-centric mobile ad hoc networks and content routing: A survey," *Ad Hoc Networks*, vol. 0, pp. 1–14, 2016.

[9] W. Mitchener and A. Vadhat, "Epidemic routing for partially connected ad hoc networks," *Technical Report CS-2000-06*, 2000.

[10] K. Zhu, W. Li, X. Fu, and L. Zhang, "Data routing strategies in opportunistic mobile social networks: Taxonomy and open challenges," *Computer Networks*, vol. 93, pp. 183–198, 2015.

[11] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A case for stateful forwarding plane," *Computer Communications*, vol. 36, no. 7, pp. 779–791, 2013.

[12] Y.-T. Yu, R. B. Dilmaghani, S. Calo, M. Sanadidi, and M. Gerla, "Interest propagation in named data manets," in *Computing, Networking and Communications (ICNC), 2013 International Conference on*, pp. 1118–1122, IEEE, 2013.

[13] Y. Lu, B. Zhou, L.-C. Tung, M. Gerla, A. Ramesh, and L. Nagaraja, "Energy-efficient content retrieval in mobile cloud," in *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*, pp. 21–26, ACM, 2013.

[14] N. Benamar, K. D. Singh, M. Benamar, D. El Ouadghiri, and J.-M. Bonnin, "Routing protocols in vehicular delay tolerant networks: A comprehensive survey," *Computer Communications*, vol. 48, pp. 141–158, 2014.

[15] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, "Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 607–640, 2012.

[16] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet," *SIGARCH Comput. Archit. News*, vol. 30, 2002.

[17] R. Mangrulkar and M. Atique, "Routing protocol for delay tolerant network: A survey and comparison," in *Communication Control and Computing Technologies (ICCCCT), 2010 IEEE International Conference on*, pp. 210–215, IEEE, 2010.

[18] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *IEEE/ACM Transactions on Networking (ToN)*, vol. 16, 2008.

[19] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE mobile computing and communications review*, vol. 7, no. 3, pp. 19–20, 2003.

[20] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, 2011.

[21] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pp. 32–40, ACM, 2007.

[22] E. M. Daly and M. Haahr, "Social network analysis for information flow in disconnected delay-tolerant manets," *IEEE Transactions on Mobile Computing*, vol. 8, no. 5, pp. 606–621, 2009.

[23] K. Zhu, W. Li, and X. Fu, "Rethinking routing information in mobile social networks: Location-based or social-based?," *Computer Communications*, vol. 42, pp. 24–37, 2014.

[24] M. Amadeo, C. Campolo, and A. Molinaro, "Forwarding strategies in named data wireless ad hoc networks: Design and evaluation," *Journal of Network and Computer Applications*, vol. 50, pp. 148–158, 2015.

[25] D. Kim and Y.-B. Ko, "A novel message broadcasting strategy for reliable content retrieval in multi-hop wireless content centric networks," *9th IMCOM '15*, pp. 1–8, 2015.

[26] D. Kim, J.-h. Kim, C. Moon, J. Choi, and I. Yeom, "Efficient content delivery in mobile ad-hoc networks using CCN," *Ad Hoc Networks*, vol. 36, pp. 81–99, 2016.

[27] F. Neves dos Santos, B. Ertl, C. Barakat, T. Spyropoulos, and T. Turletti, "CEDO: Content-Centric Dissemination Algorithm for Delay-Tolerant Networks," *Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems - MSWiM '13*, pp. 377–386, 2013.

[28] C. Anastasiades, W. E. M. El Alami, and T. Braun, "Agent-based content retrieval for opportunistic content-centric networks," in *WWIC*, pp. 175–188, 2014.

[29] Y. Lu, X. Li, Y.-T. Yu, and M. Gerla, "Information-centric Delay-Tolerant Mobile Ad-Hoc Networks," *IEEE Infocom*, pp. 428–433, 2014.

[30] P. Duarte, J. Macedo, A. D. Costa, M. J. Nicolau, and A. Santos, "A Probabilistic Interest Forwarding Protocol for Named Data Delay Tolerant Networks," *7th AdhocNets 2015 San Remo, Italy, September 1-2, 2015 proceedings*, vol. 155, pp. 94–107, 2015.

[31] N. M. Junior, C. A. V. Campos, and S. C. Lucena, "Uma proposta de solução para o problema de nós preferidos em protocolos oportunistas com base social," in *Anais do 32º SBRC*, pp. 91–104, 2014.

[32] N. M. Junior and C. A. V. Campos, "Socleer: A social-based energy-efficient forwarding protocol for opportunistic networks," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2015 International*, pp. 757–762, IEEE, 2015.

[33] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," in *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, (New York, NY, USA), ICST, 2009.

[34] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "CRAWDAD dataset epfl/mobility (v. 2009-02-24)." Downloaded from `http://crawdad.org/epfl/mobility/20090224`, Feb. 2009.

[35] T. Le, Y. Lu, and M. Gerla, "Social caching and content retrieval in Disruption Tolerant Networks (DTNs)," *2015 International Conference on Computing, Networking and Communications, ICNC 2015*, pp. 905–910, 2015.

[36] L. Pu, X. Chen, J. Xu, and X. Fu, "sNDN: A Social-aware Named Data Framework for Cooperative Content Retrieval via D2D Communications," *Proceedings of the 10th International Workshop on Mobility in the Evolving Internet Architecture*, pp. 14–19, 2015.

[37] Y. Lu, M. Gerla, T. Le, V. Rabsatt, and H. Kalantarian, "Community aware content retrieval in disruption-tolerant networks," *13th MED-HOC-NET 2014*, pp. 172–179, 2014.

[38] R Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016.