

Número 01
Volume 06
Dezembro / 2013
ISSN 1983-4217

REVISTA BRASILEIRA
DE REDES DE COMPUTADORES
E SISTEMAS DISTRIBUÍDOS

Thaís Vasconcelos Batista
Lisandro Zambenedetti Granville
Cristiano Bonato Both

Número 1
Volume 6
Dezembro de 2013
ISSN 1983-4217

Revista Brasileira de Redes de Computadores e Sistemas Distribuídos



**Permitida somente a reprodução parcial dos artigos
publicados desde que a fonte seja citada**

Revista Brasileira de Redes de Computadores e Sistemas Distribuídos /
Laboratório Nacional de Redes de Computadores (LARC), Comissão
Especial de Redes de Computadores da Sociedade Brasileira de Computação.
Rio de Janeiro: LARC; SBC, 2012.
v. 6, n. 1 (jul-dez, 2013)
v. : il. cm.
Semestral.
Texto em português ou inglês.
ISSN 1983-4217

1. Redes de computadores 2. Sistemas distribuídos. I. Laboratório Nacional de
Redes de Computadores (LARC) II. Sociedade Brasileira de Computação.
Comissão Especial de Redes de Computadores. III. Título.

CDD 004.65

LARC - Laboratório Nacional de Redes de Computadores

Diretor do Conselho Técnico-Científico

Elias Procópio Duarte Jr.
Universidade Federal do Paraná (UFPR)

Diretor Executivo

Luciano Paschoal Gaspar
Universidade Federal do Rio Grande do Sul (UFRGS)

Vice-Diretora do Conselho Técnico-Científico

Rossana Maria de C. Andrade
Universidade Federal do Ceará (UFC)

Vice-Diretor Executivo

Paulo André da Silva Gonçalves
Universidade Federal de Pernambuco (UFPE)

SBC - Sociedade Brasileira de Computação

Presidente

Paulo Roberto Freire Cunha
Universidade Federal de Pernambuco (UFPE)

Vice-Presidente

Lisandro Zambenedetti Granville
Universidade Federal do Rio Grande do Sul (UFRGS)

Comissão Especial de Redes de Computadores e Sistemas Distribuídos

Dorival Olavo Guedes Neto - Coordenador
Universidade Federal de Minas Gerais (UFMG)

José Augusto Suruagy Monteiro
Universidade Federal de Pernambuco (UFPE)

José Ferreira de Rezende
Universidade Federal do Rio de Janeiro (UFRJ)

Editores

Thaís Vasconcelos Batista
Universidade Federal do Rio Grande do Sul (UFRN)

Lisandro Zambenedetti Granville
Universidade Federal do Rio Grande do Sul (UFRGS)

Cristiano Bonato Both – Editor associado
Universidade Federal do Rio Grande do Sul (UFRGS)

Capa

Adriano Barros da Silva
Universidade Federal do Rio de Janeiro (UFRJ)

Corpo Editorial

Antônio Jorge Gomes Abelém
Universidade Federal do Pará (UFPA)

Artur Ziviani
Laboratório Nacional de Computação Científica(LNCC)

Célio Vinicius Neves de Albuquerque
Universidade Federal Fluminense (UFF)

Djamel H. Sadok
Universidade Federal de Pernambuco (UFPE)

Edmundo Roberto Mauro Madeira
Universidade de Campinas (UNICAMP)

Elias Procópio Duarte Jr.
Universidade Federal do Paraná (UFPR)

Fábio Kon
Universidade de São Paulo (USP)

Joni Fraga
Universidade Federal de Santa Catarina (UFSC)

José Marcos Nogueira
Universidade Federal de Minas Gerais (UFMG)

Flávia Coimbra Delicato
Universidade Federal do Rio Grande do Norte (UFRN)

Lisandro Zambenedetti Granville
Universidade Federal do Rio Grande do Sul (UFRGS)

Luci Pirmez
Universidade Federal do Rio de Janeiro (UFRJ)

Luciano Paschoal Gaspar
Universidade Federal do Rio Grande do Sul (UFRGS)

Luiz Fernando Gomes G. Soares
Pontifícia Universidade Católica (PUC)

Luiz Fernando Rust da Costa Carmo
Universidade Federal do Rio de Janeiro (UFRJ)

Neuman Souza
Universidade Federal do Ceará (UFC)

Rossana Andrade
Universidade Federal do Ceará (UFC)

Thaís Vasconcelos Batista
Universidade Federal do Rio Grande do Norte (UFRN)

Laboratório Nacional de Redes de Computadores (LARC)
Contato: Luciano Paschoal Gaspar
Instituto de Informática - UFRGS
Av. Bento Gonçalves, 9500 - Porto Alegre, RS
Caixa Postal 15064 - CEP 91501-970

Sociedade Brasileira de Computação
Contato: Carlos André Guimarães Ferraz
Av. Bento Gonçalves, 9500 - Porto Alegre, RS
Caixa Postal 15012 - CEP 91501-970

Conteúdo

Carta dos Editores	7
<i>Thaís Vasconcelos Batista, Lisandro Zambenedetti Granville, Cristiano Bonato Both</i>	

Artigos

RemapRoute: Remapeando Mudanças de Rota na Internet	11
<i>Ítalo Cunha, Renata Teixeira, Darryl Veitch, Christophe Diot</i>	

Uma Fotografia do Instagram: Caracterização e Aplicação	23
<i>Thiago H. Silva, Pedro O. S. Vaz de Melo, Jussara M. Almeida, Antonio A. F. Loureiro</i>	

Análise e Geração de Vizinhança em Redes de Contato Intermitente	33
<i>Tiphaine Phe-Neau, Miguel Elias M. Campista, Marcelo Dias de Amorim, Vania Conan</i>	

Carta dos Editores

A Revista Brasileira de Redes de Computadores e Sistemas Distribuídos é um periódico promovido conjuntamente pelo Laboratório Nacional de Redes de Computadores (LARC) e Sociedade Brasileira da Computação (SBC) através de sua Comissão Especial em Redes de Computadores e Sistemas Distribuídos (CE-RESD). Como resultado das ações da comunidade nacional de pesquisa em redes de computadores e sistemas distribuídos, a Revista tem por objetivo se estabelecer como um veículo de divulgação dos avanços científicos e tecnológicos da área, e assim estender o atual alcance do prestigioso e tradicional Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC).

Nesta nona edição, a Revista apresenta versões atualizadas e estendidas de três artigos dentre os ganhadores do prêmio de melhor trabalho da 31ª edição do SBRC, realizado em 2013 em Brasília. O primeiro artigo propõe uma ferramenta para remapeamento local de mudanças de roteamento que considera apenas os roteadores afetados ao invés de toda a rota. O segundo artigo versa sobre o Sistema de Sensoriamento Participativo (SSP) (*participatory sensing systems*) com foco na análise do Instagram, um sistema para compartilhamento de fotos. Finalmente, o terceiro artigo apresenta uma análise e geração de vizinhança em redes de contato intermitente com uso de uma visão estendida, onde os nós rastreiam sua vizinhança estendida e não somente seus vizinhos diretos.

No primeiro artigo, "RemapRoute: Remapeando Mudanças de Rota na Internet", de Ítalo Cunha, Renata Teixeira, Darryl Veitch e Christophe Diot, os autores mostram que mapas topológicos da Internet coletados através do *traceroute* podem estar desatualizados ou incompletos pela impossibilidade de medições em frequências suficientes para detectar todas as mudanças de rota. Com isso, os autores propõem uma

ferramenta denominada RemapRoute, capaz de explorar o fato de que as mudanças de rota geralmente afetam poucos roteadores, e com isso a ferramenta de remapeamento local de mudanças de roteamento sonda apenas poucos roteadores ao invés de toda a rota. Através deste trabalho, os autores mostram uma redução significativa no número de sondas de remapeamento sem comprometer a exatidão e a latência do remapeamento, e com isso construir mapas topológicos mais completos e atualizados.

No segundo artigo, "Uma Fotografia do Instagram: Caracterização e Aplicação" de autoria de Thiago H. Siva, Pedro O. S. Vaz de Melo, Jussara M. Almeida e Antonio A. F. Loureiro, é apresentado o Sistema de Sensoriamento Participativo (SSP) com potencial de se tornar uma ferramenta fundamental para o estudo em larga escala do comportamento social urbano e da dinâmica de cidades. Neste sentido, os autores analisam o Instagram, um sistema para compartilhamento de fotos, um dos mais populares SSPs disponíveis. Baseado em um conjunto de 2,3 milhões de fotos, os autores mostram que existem muitas vantagens e oportunidades para o sensoriamento em grande escala e ainda apresentam uma aplicação baseada em dados obtidos do Instagram para identificar regiões de interesse dentro de uma cidade.

Por fim, no terceiro artigo, "Análise e Geração de Vizinhança em Redes de Contato Intermitente" de autoria de Tiphaine Phe Neau, Miguel Elias M. Campista, Marcelo Dias de Amorim e Vania Conan, propõe o uso de uma visão estendida, na qual os nós rastreiam os vizinhos dos seus vizinhos e não somente seus vizinhos diretos. Neste trabalho é realizada uma análise de desenvolvimento e modelagem, comportamento da vizinhança e tipos de movimentos, que resulta em um gerador de vizinhança capaz de gerar traços de contato entre pares de nós preservando as características reais.

Thaís Vasconcelos Batista
Editor
Universidade Federal do Rio Grande do Norte
Departamento de Informática e
Matemática Aplicada
Av. Sen. Salgado Filho, 3000
59072-970 – Natal, RN

Lisandro Zambenedetti Granville
Editor
Universidade Federal do Rio Grande do Sul
Instituto de Informática
Av. Bento Gonçalves, 9500
Caixa Postal 15064
91501-970 – Porto Alegre, RS

Cristiano Bonato Both
Editor Associado
Universidade Federal do Rio Grande do Sul
Instituto de Informática
Av. Bento Gonçalves, 9500
Caixa Postal 15064
91501-970 – Porto Alegre, RS

Uma Fotografia do Instagram: Caracterização e Aplicação

Thiago H. Silva, Pedro O. S. Vaz de Melo, Jussara M. Almeida & Antonio A. F. Loureiro

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG)
31270-010 Belo Horizonte, MG – Brasil
{thiagohs, olmo, jussara, loureiro}@dcc.ufmg.br

Abstract

Participatory sensing systems (PSS) have the potential to become fundamental tools for supporting the study, in large scale, of urban social behavior and city dynamics. To that end, this work characterizes the photo sharing system Instagram, considered one of the currently most popular PSSs on the Internet. Based on a dataset of approximately 2.3 million shared photos, we characterize user behavior in the system showing that there are several advantages and opportunities for large scale sensing, such as a global coverage at low cost, but also challenges, such as a very unequal photo sharing frequency, both spatially and temporally. Moreover, we present an application based on data obtained from Instagram to identify regions of interest in a city, which illustrates the promising potential of PSSs for the study of city dynamics.

Resumo

Sistemas de sensoriamento participativo (SSP) (participatory sensing systems) têm o potencial de se tornarem ferramentas fundamentais para o estudo em larga escala do comportamento social urbano e da dinâmica de cidades. Nessa direção, este trabalho analisa o Instagram, um sistema para compartilhamento de fotos que é considerado um dos mais populares SSPs disponíveis na Internet atualmente. Baseado em um conjunto de dados de aproximadamente 2,3 milhões de fotos compartilhadas, caracterizamos o comportamento de usuários desse sistema, mostrando que existem muitas vantagens e oportunidades para sensoriamento em grande escala, tais como uma abrangência global a baixo custo, mas também desafios, como uma frequência de compartilhamento de fotos espaço-temporal altamente desigual. Além disso, apresentamos uma aplicação baseada em dados obtidos do Instagram para identificar regiões de interesse dentro de uma cidade. Essa aplicação ilustra o potencial promissor de SSPs para o estudo da dinâmica das cidades.

Keywords: Characterization, Instagram, Participatory Sensing, Application, City Dynamics

1. INTRODUÇÃO

Mark Weiser, no seu trabalho clássico intitulado “*The computer for the 21st century*” publicado na *Scientific American* [30], popularizou o conceito da *computação ubíqua*, que prevê o acesso a ambientes de computação por qualquer pessoa, em qualquer lugar, a qualquer instante, com dispositivos computacionais acoplados aos mais triviais objetos, como etiquetas de roupas, copos de café, canetas ou qualquer objeto pessoal. Embora essa ainda não seja a realidade e esse conceito tenha sido entendido para incluir, por exemplo, Internet das Coisas (*Internet of Things*), muito foi feito nessa direção nestes últimos 20 anos após a publicação do trabalho de Weiser. Por exemplo, as Redes de Sensores Sem Fio (RSSFs) [1], que são um tipo especial de rede *ad hoc*, são projetadas para coletar dados referentes a grandezas físicas dos mais variados ambientes em que estão inseridas e a fornecer tais informações para o usuário final. Além disso, há o uso crescente de sistemas de sensoriamento participativo (SSPs) [23], que permitem a pessoas conectadas à Internet fornecerem dados de contexto sobre o ambiente em que estão em um determinado momento.

De fato, os SSPs têm o potencial para complementar as RSSFs em diversos aspectos. Enquanto as RSSFs foram projetadas para sensoriar áreas de tamanho limitado, como florestas e vulcões, os SSPs podem alcançar áreas de tamanhos variados e de larga escala, como grandes metrópoles, países ou até mesmo todo o planeta [22]. Além disso, uma RSSF está sujeita a falhas, uma vez que o seu funcionamento depende da correta coordenação das ações dos seus nós sensores, que possuem severas restrições de energia, processamento e memória. Por outro lado, SSPs são formados por entidades autônomas e

independentes, os seres humanos, o que torna a tarefa de sensoriamento altamente resiliente a falhas individuais.

O sucesso dos SSPs está diretamente ligado à popularização do smartphone, que se tornou o dispositivo computacional pessoal mais amplamente adotado e onipresente [12]. Os smartphones possuem um rico conjunto de sensores embutidos, tais como GPS, acelerômetro, microfone, câmera, giroscópio e bússola digital. Entretanto, o sensoriamento não depende apenas dos dados gerados por esses sensores, podendo vir também das observações subjetivas do seu usuário. É possível encontrar vários exemplos de SSPs já implantados e usados através de smartphones, como o Waze¹, para relatar condições de tráfego em tempo real, e o Weddar², para relatar condições meteorológicas. Além disso, há serviços de compartilhamento de localização, como o Foursquare³, ou de fotos, como o Instagram⁴, nos quais os usuários podem enviar imagens em tempo real para o sistema. Em particular, o Instagram é um dos mais populares SSPs atuais, com quase 100 milhões de usuários e mais de 1 bilhão de fotos recebidas, sendo que, a cada segundo, um novo usuário se registra no sistema e 58 novas fotos são inseridas [8].

O objetivo principal deste trabalho é caracterizar a rede de participação do Instagram, visando mostrar os desafios e as oportunidades que emergem do sensoriamento participativo realizado pelos usuários desta aplicação. Baseado em um conjunto de dados de aproximadamente 2,3 milhões de fotos, mostramos a abrangência planetária da rede, assim como a frequência altamente desigual do compartilhamento de fotos, tanto espacial quanto temporalmente, que é bastante correlacionada com as rotinas de atividades humanas. Além disso, mostramos também como é possível projetar aplicações a partir de sistemas como o Instagram, apresentando uma aplicação para identificar regiões de interesse dentro de uma cidade. Essa aplicação ilustra o potencial de SSPs para o estudo da dinâmica de cidades. Até onde sabemos, este é o primeiro trabalho de caracterização do uso do Instagram, particularmente com o foco no seu potencial, como um sistema de sensoriamento participativo no projeto de novas aplicações e serviços.

O restante deste trabalho é organizado como segue. A seção 2 apresenta os trabalhos relacionados. A seção 3 discute a participação dos seres humanos no processo de sensoriamento, abordando os sistemas participativos de sensoriamento e as redes de sensores participativos (RSPs), advindas de SSPs. A seção 4 apresenta a caracterização da RSP derivada do Instagram. A seção 5 descreve uma aplicação de classificação de regiões usando

esta RSP. Finalmente, a seção 6 apresenta as conclusões e trabalhos futuros.

2. TRABALHOS RELACIONADOS

O processo de sensoriamento do meio ambiente pode envolver seres humanos como: (i) alvos do processo [13], ou (ii) responsáveis por eles a partir do compartilhamento de dados locais [27, 11]. Neste trabalho, focamos no segundo caso, considerando sistemas que utilizam dispositivos móveis do dia a dia, como smartphones, para construir uma rede de sensoriamento participativa, que é descrita na seção 3.2. Na literatura, existem diversas propostas de sistemas que consideram a participação de humanos no processo de sensoriamento, como descrito anteriormente. Tais sistemas são chamados de sistemas de sensoriamento participativo (SSPs) e incluem, por exemplo, sistemas de monitoramento de tráfego [9] e monitoramento de ruídos [18].

O sucesso de SSPs depende fundamentalmente da participação sustentável dos usuários ao longo do tempo. Em [19], os autores propõem mecanismos de incentivos baseados em micro-pagamentos (*micro-payments*), que são pequenas quantias de dinheiro dadas ao usuário quando ele realiza determinadas atividades no sistema. Além da participação sustentável, é necessário garantir a qualidade dos dados compartilhados pelos usuários [15].

Por exemplo, em diversos SSPs os usuários podem fabricar dados falsos, que supostamente foram sensorizados, a baixo custo. Logo, a integridade dos dados não é sempre garantida [20].

Existem vários trabalhos dedicados ao estudo das características de SSPs específicos. Por exemplo, em serviços de compartilhamento de localização, como o Foursquare, os autores de [5] observaram que os usuários seguem um padrão de mobilidade simples e factível de ser reproduzido. Nessa direção, os autores de [6]

observaram que viagens de curta distância são periódicas no espaço e no tempo e não são afetadas pela estrutura social da rede, que, por sua vez, influencia somente as viagens de longa distância.

De fato, Scellato et al. [21] mostraram que 40% das relações sociais no sistema analisado acontecem a menos de 100 km.

Em [16], os autores analisaram a dinâmica de compartilhamento dos usuários de serviços de compartilhamento de localização mostrando, por exemplo, que a distribuição do número de check-ins é altamente desigual, sendo bem modelada por um comportamento de lei de potência (*power-law*).

Em [28], os autores analisaram o uso de *tips* no Foursquare, que são comentários curtos sobre determinado local, caracterizando como os usuários interagem

¹<http://www.waze.com>

²<http://www.weddar.com>

³<http://www.foursquare.com>

⁴<http://www.instagram.com>

entre si utilizando esta funcionalidade e propuseram um algoritmo para estimar a influência de usuários. Diferentemente dos demais trabalhos, os autores não exploraram características geográficas.

Outros trabalhos propõem utilizar dados derivados de SSPs em novas aplicações, já que esse tipo de dado auxilia no melhor entendimento das fronteiras físicas e noções de espaço [3]. Nessa direção, os autores de [7] apresentaram um modelo que classifica regiões de uma cidade a partir de padrões de atividades coletivas, enquanto Noulas et al. [17] propuseram classificar áreas e usuários de uma cidade usando as categorias dos locais registrados no Foursquare.

Em trabalho anterior [22], analisamos as propriedades de RSPs derivadas de duas aplicações de compartilhamento de localização: Gowalla e Brightkite. Nós analisamos as distribuições espacial e temporal dos *check-ins* realizados pelos usuários desses sistemas, visando levantar evidências relevantes para o projeto de novos serviços e aplicações. Já em [24], propusemos uma nova forma de visualizar a dinâmica de cidades a partir de hábitos e rotinas de usuários, derivados dos *check-ins* no Foursquare.

O trabalho aqui apresentado diferencia-se dos anteriores (incluindo os nossos) por focar em um novo sistema de grande popularidade atualmente – o Instagram. Até onde sabemos, esta é a primeira caracterização de uma aplicação de compartilhamento de fotos acessada, majoritariamente, a partir de *smartphones*. Mais ainda, dando continuidade aos trabalhos recentes [22, 24], este trabalho também aborda o estudo da dinâmica das cidades através de SSPs, mostrando que sistemas de compartilhamento de fotos, particularmente o Instagram, também podem ser utilizados para este propósito.

3. HUMANOS NO PROCESSO DE SENSORIAMENTO

O foco deste trabalho é em sistemas que dependem de humanos no processo de sensoriamento, sendo eles responsáveis pelo compartilhamento de dados locais. Tais dados podem ser obtidos com o auxílio de dispositivos de sensoriamento, como sensores embutidos em celulares (por exemplo, GPS), ou através de sensores humanos (por exemplo, visão), compartilhando informações produzidas por eles próprios.

3.1. SENSORIAMENTO PARTICIPATIVO

Sensoriamento participativo é o processo em que seres humanos usam dispositivos móveis e serviços de computação em nuvem para compartilhar dados sensoriais [23, 4]. Usualmente, sistemas de sensoriamento participativo consideram que o compartilhamento dos dados é gerado automaticamente, ou passivamente, por sen-

sos embutidos no dispositivo móvel. Porém, neste trabalho consideramos também observações geradas pelos usuários de forma manual, ou proativa. Sensoriamento participativo com essas características também pode ser chamado de *crowdsourcing* ubíquo (*ubiquitous crowdsourcing*) [15]. A popularidade de sistemas para sensoriamento participativo cresceu rapidamente com o aumento do uso de celulares com sensores embutidos e capacidade de acesso à Internet, ou seja, os chamados *smartphones*. Esses dispositivos se tornaram uma poderosa plataforma que inclui capacidades de sensoriamento, computação e comunicação.

Um dado sensoriado em uma aplicação de sensoriamento participativo é: (i) obtido através de sensores físicos (por exemplo, GPS) ou observações humanas (por exemplo, congestionamento na rodovia); (ii) definido no tempo e no espaço; (iii) obtido automaticamente ou manualmente; (iv) estruturado ou não; (v) compartilhado voluntariamente ou não. Para ilustrar esse tipo de sistema, considere uma aplicação para monitoramento de trânsito, como o Waze. Usuários podem compartilhar observações sobre acidentes ou congestionamentos manualmente. Uma aplicação poderia ainda calcular e compartilhar automaticamente a velocidade de um carro com o auxílio de dados obtidos com o GPS. Com as medidas da velocidade de diferentes veículos amostrados em uma área particular, é possível inferir, por exemplo, congestionamentos. Como nesse caso específico usuários operam uma aplicação que foi criada para esse propósito, o dado sensoriado é estruturado. Mas, caso os usuários usem um serviço de *microblogging*, como o Twitter, o dado sensoriado seria não estruturado. Por exemplo, o usuário “João” envia uma mensagem “trânsito agora está muito lento próximo da portaria do campus”.

Serviços de compartilhamento de fotos, como o Instagram, são exemplos de aplicações para sensoriamento participativo. O dado sensoriado é uma foto de um lugar específico. Podemos extrair informação desse tipo de dado de diversas maneiras. Por exemplo, é possível visualizar em tempo real como está a situação de uma certa área da cidade.

3.2. REDE DE SENSORIAMENTO PARTICIPATIVO

Em uma rede de sensoriamento participativo (RSP), o dispositivo móvel do usuário é uma peça fundamental. Indivíduos carregando esses dispositivos são capazes de sensoriar o ambiente e fazer observações relevantes. Assim, cada nó em uma RSP consiste de um usuário com o seu dispositivo móvel. De forma similar às RSSFs, o dado sensoriado é enviado para o servidor, ou “nó sorvedouro”. Mas, diferentemente das RSSFs, RSPs têm as seguintes características: (a) nós são entidades móveis autônomas, mas uma pessoa com um dispositivo móvel; (b) o custo da rede é distribuído entre os nós, proporcio-

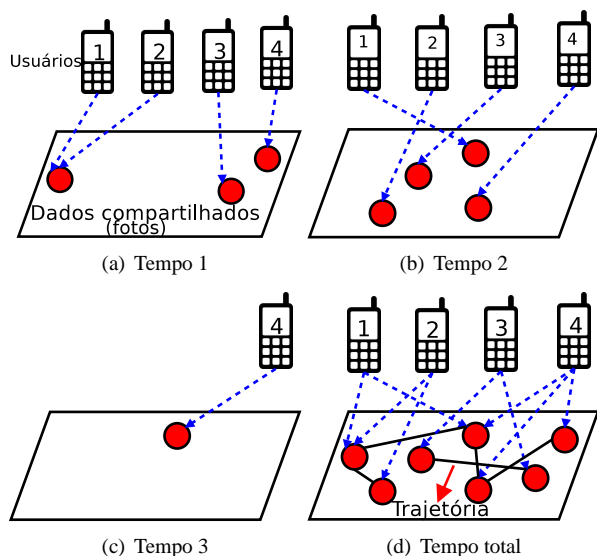


Figure 1. RSP analisada: serviço de compartilhamento de fotos

nando uma escalabilidade global; (c) o sensoriamento depende da vontade das pessoas participarem no processo de sensoriamento; (d) nós transmitimos o dado sensoriado diretamente para o sorvedouro; (e) nós não sofremos de severas limitações de energia; (f) o nó sorvedouro só recebe dados e não tem controle direto sobre os nós.

A figura 1 mostra um exemplo de RSP formada a partir de serviços de compartilhamento de fotos, que é a RSP analisada nas seções seguintes. As figuras 1-a, 1-b e 1-c representam quatro usuários em três diferentes momentos. Fotos compartilhadas pelos usuários a cada momento são marcadas por uma seta pontilhada. Observe que nem todos os usuários realizam atividades em todos os momentos. Depois de um certo intervalo, podemos analisar os dados de diversas maneiras. Por exemplo, a figura 1-d mostra um grafo onde os vértices representam os locais onde as fotos foram compartilhadas e as arestas conectam fotos compartilhadas pelo mesmo usuário. Com esse grafo é possível extrair várias informações interessantes de diferentes partes do mundo, fornecendo uma notável escala global a uma infraestrutura de baixo custo, como ilustrado na figura 2.

4. CARACTERIZAÇÃO DO INSTAGRAM

Nesta seção nós analisamos uma rede de sensores participativa (RSP) derivada do Instagram.

4.1. DESCRIÇÃO DOS DADOS

A RSP analisada é derivada de um conjunto de dados do Instagram, que é um serviço de compartilhamento de fotos online. Os dados do Instagram foram coletados

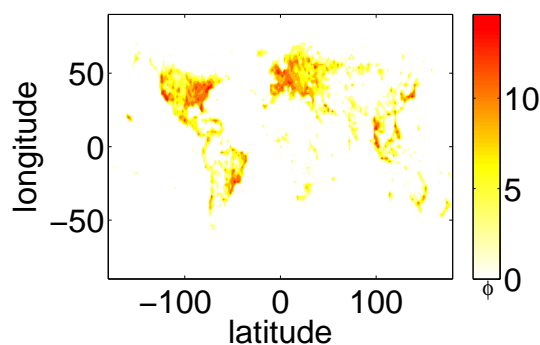


Figure 2. Número de fotos n por pixel dado pelo valor de ϕ mostrado na figura, em que $n = 2^\phi - 1$.

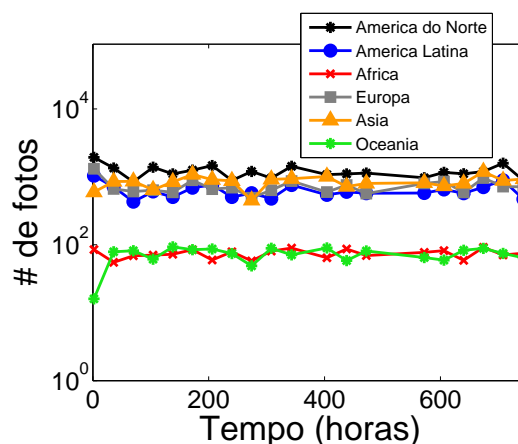


Figure 3. Variação temporal do número de fotos compartilhadas por continente.

através do Twitter⁵, que é um serviço de *microblogging*, ou seja, ele permite que os seus usuários enviem e recebam atualizações pessoais de outros contatos em textos de até 140 caracteres, conhecidos como “*tweets*”. Além de *tweets* de texto simples, os usuários também podem compartilhar fotos a partir de uma integração com o Instagram. Neste caso, fotos do Instagram anunciadas no Twitter passam a ficar disponíveis publicamente, o que por padrão não acontece quando a foto é publicada unicamente no sistema do Instagram.

Entre 30 de junho e 31 de julho de 2012, foram coletados 2 272 556 *tweets* contendo fotos georeferenciadas, postadas por 482 629 usuários. Cada *tweet* é composto de coordenadas GPS (latitude e longitude) e o horário do compartilhamento da foto.

4.2. COBERTURA DA REDE

Nesta seção, analisamos a cobertura da RSP do Instagram em diferentes granularidades espaciais, começando por todo o planeta, depois por continentes e cidades e,

⁵<http://www.twitter.com>

por fim, até áreas específicas de uma cidade. A figura 2 mostra a cobertura no planeta da RSP do Instagram na forma de um mapa de calor da participação dos usuários: cores mais escuras representam um maior número de fotos compartilhadas em determinada área. Apesar da cobertura ser bastante abrangente na escala planetária, ela não é homogênea. A figura 3 mostra o número de fotos compartilhadas por continente ao longo do tempo. Note que a atividade de sensoriamento nas Américas, Europa e Ásia é pelo menos uma ordem de magnitude maior que na África e Oceania. Além disso, pode-se observar ainda que a participação dos usuários da América do Norte é levemente superior a dos usuários da América Latina, Europa e Ásia.

Avaliamos agora a participação dos usuários do Instagram em oito grandes e populosas cidades localizadas em cinco continentes: Nova York, Rio de Janeiro, Belo Horizonte (BH), Roma, Paris, Sydney, Tokyo, e Cairo. A figura 4 mostra o mapa de calor da atividade de sensoriamento para cada uma dessas cidades. Mais uma vez, cores mais escuras representam um maior número de fotos em determinada área. Pode-se observar uma alta cobertura para algumas cidades, como mostrado nas figuras 4-a (Nova York), 4-e (Paris) e 4-g (Tokyo). No entanto, pode-se observar na figura 4-h que o sensoriamento no Cairo, que também possui um número elevado de habitantes, é significativamente mais baixo. T tamanha diferença na cobertura pode ser explicada por diversos fatores. Além dos aspectos econômicos, diferenças na cultura dos habitantes desta cidade quando comparadas com as culturas presentes nas outras cidades estudadas podem ter um impacto significativo na adoção e uso do Instagram [2].

Além disso, pode-se observar que a cobertura no Rio de Janeiro e em Sydney é bem mais heterogênea quando comparada com a cobertura em Paris, Tóquio e Nova York. Isto ocorre provavelmente por causa dos aspectos geográficos que estas cidades têm em comum, ou seja, grandes áreas verdes e grandes porções d'água. Rio de Janeiro, por exemplo, tem a maior floresta urbana do mundo, localizada no meio da cidade, além de muitas colinas de difícil acesso humano. Estes aspectos geográficos limitam a cobertura do sensoriamento. Além disso, em ambas as cidades os pontos de interesse público, tais como pontos turísticos e centros comerciais, são distribuídos de forma desigual pela cidade. Há grandes áreas residenciais com poucos pontos desse tipo, enquanto outras áreas têm grande concentração desses pontos. Estes resultados são qualitativamente semelhantes aos reportados em [22, 24] para RSPs derivadas de três aplicações de compartilhamento de localização e para diferentes cidades, o que demonstra o potencial do Instagram como ambiente para sensoriamento participativo em grandes regiões.

Uma vez que a atividade de participação pode ser bas-

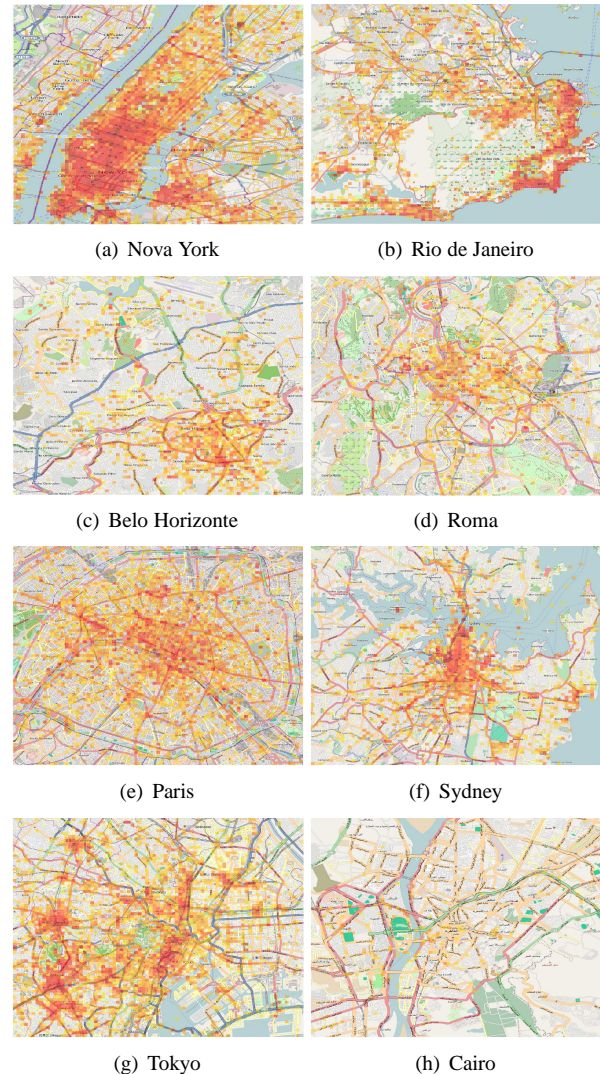


Figure 4. Cobertura espacial da RSP do Instagram em 8 cidades: todas as fotos compartilhadas. O número de fotos em cada área é representado por um mapa de cores, onde a escala vai de amarelo a vermelho (atividade mais intensa).

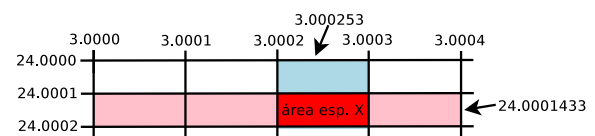


Figure 5. Exemplo de identificação de uma área específica

tante heterogênea dentro de uma cidade, propomos dividir a área de cidades em espaços retangulares menores, como em uma grade⁶. Chamaremos cada área retangular de uma *área específica* dentro de uma cidade e, a partir disso, analisaremos o número de fotos compartilhadas nessas áreas específicas. Neste trabalho, consider-

⁶Note que nas áreas selecionadas não é considerado fronteiras.

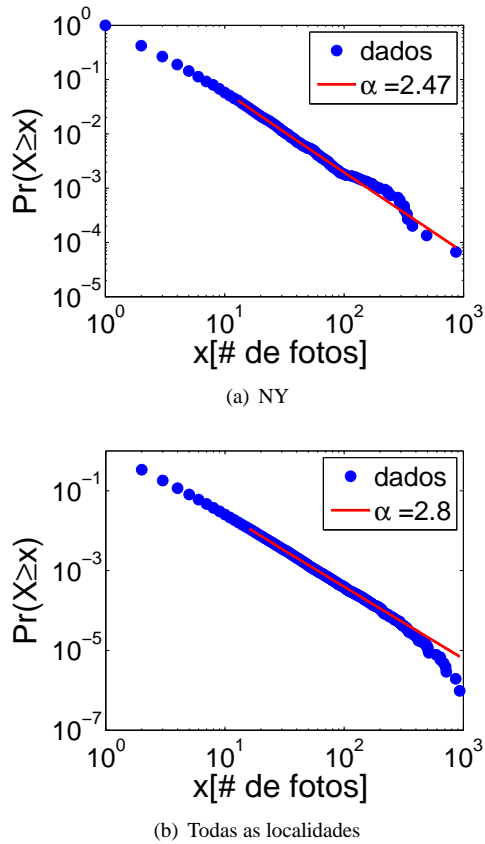


Figure 6. Distribuição do número de fotos em áreas específicas

amos que uma área específica possui a seguinte delimitação: $1 \cdot 10^{-4}^\circ$ (latitude) \times $1 \cdot 10^{-4}^\circ$ (longitude). Isso representa uma área de aproximadamente 8×11 metros em NY e 10×11 metros no Rio de Janeiro. Para outras cidades, as áreas também podem variar um pouco, mas não a ponto de afetar significativamente as análises realizadas. A figura 5 ilustra o processo de divisão da área de uma cidade em áreas específicas e de como é feita a associação da coordenada geográfica (24,0001433; 3,000253) a uma área específica X .

A figura 6 apresenta a função de distribuição acumulada complementar (CCDF) do número de fotos compartilhadas por área específica da cidade de Nova York (figura 6-a) e de todas as localidades em nossa base de dados (figura 6-b). Primeiramente, observe que, em ambos os casos, uma lei de potência⁷ descreve bem esta distribuição. Isso implica que, na maioria das áreas específicas, há poucas fotos compartilhadas, enquanto existem algumas poucas áreas com centenas de fotos compartilhadas. Estes resultados estão consistentes com os re-

⁷Matematicamente, uma quantidade x segue uma lei de potência se ela pode ser obtida de uma distribuição de probabilidade $p(x) \propto x^{-\alpha}$, onde α é um parâmetro constante conhecido como expoente ou parâmetro escalar, e é um valor tipicamente entre $2 < \alpha < 3$.

sultados apresentados em [16, 22], que estudaram a participação de usuários em sistemas de compartilhamento de localização. Em sistemas para compartilhamento de fotos, assim como em sistemas de compartilhamento de localizações, é natural que algumas áreas possuam mais atividade que outras. Por exemplo, em áreas turísticas o número de fotos compartilhadas tende a ser maior do que em um supermercado, apesar de um supermercado ser geralmente um local bastante popular. Se uma determinada aplicação requer uma cobertura mais abrangente, é necessário incentivar os usuários a participarem em locais que eles usualmente não o fariam. Micro-pagamentos ou sistemas de pontuação são exemplos de alternativas que poderiam funcionar nesse caso.

Como foi mostrado anteriormente, uma RSP pode ter uma cobertura em escala planetária. No entanto, foi mostrado também que essa cobertura pode ser bastante heterogênea, em que grandes áreas ficam praticamente descobertas.

A figura 7 mostra a cobertura da rede total considerando a dimensão temporal, ou seja, o número de localidades que estão ativas (i.e., sensoriadas) em um determinado intervalo de tempo, considerando todos os dados disponíveis. O número máximo de áreas específicas sensoriadas por hora corresponde a aproximadamente somente 0,2% do número total de áreas em nossa base de dados (1 030 558). Em outras palavras, a cobertura instantânea da RSP do Instagram é muito limitada quando consideramos todas as localidades que poderiam ser sensoriadas no planeta⁸. Isso significa que a probabilidade de uma área específica aleatória ser sensoriada em um horário aleatório é bem baixa.

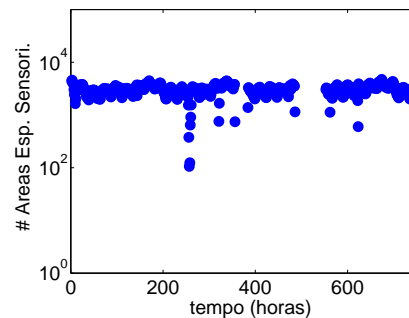


Figure 7. Variação temporal do número de áreas específicas sensoriadas.

4.3. INTERVALO DE SENSORIAMENTO

Redes de sensoriamento participativo são bastante escaláveis porque seus nós são autônomos, ou seja, os usuários são responsáveis pela sua própria operação e fun-

⁸Considerando nesse caso todas as localidades já sensoriadas pelo menos uma vez.

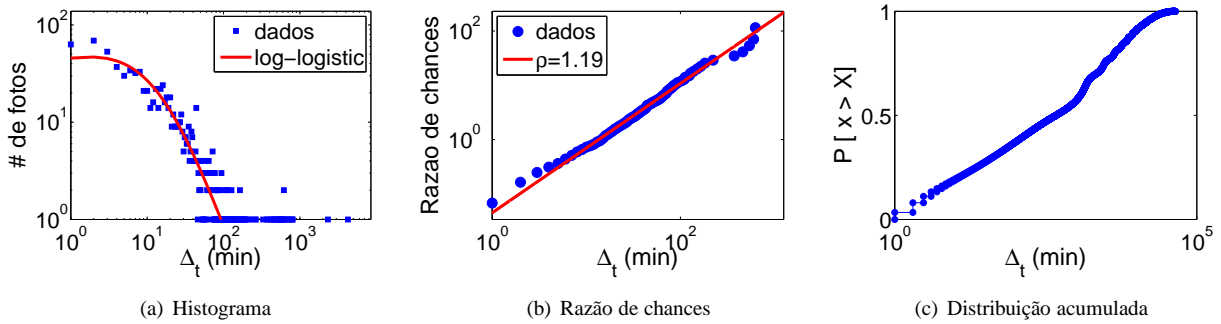


Figure 8. Distribuição do intervalo de tempo entre compartilhamentos de fotos em uma área específica popular.

cionamento. Como o custo da infraestrutura é distribuído entre os participantes, esta enorme escalabilidade e cobertura é alcançada mais facilmente. O sucesso desse tipo de rede consiste em ter participação sustentável e de alta qualidade. Em outras palavras, o sensoriamento é eficiente desde que os usuários sejam mantidos motivados a compartilhar seus recursos e dados sensorizados frequentemente.

Investigamos agora a frequência com que usuários do Instagram realizam o compartilhamento de fotos. A figura 8-a mostra o histograma do intervalo de tempo Δ_t entre compartilhamentos de fotos consecutivos em uma determinada área específica tipicamente popular. Note que o histograma é bem ajustado por uma distribuição *log-logistic* [10]. Observe as rajadas de atividade e os longos períodos de inatividade: há momentos em que muitas fotos são compartilhadas em intervalos de poucos minutos e momentos em que não há compartilhamento por horas. Isso pode indicar que a maioria do compartilhamento de fotos, nesta área popular (assim como em outras), acontece em intervalos específicos, provavelmente relacionados ao horário em que as pessoas usualmente as visitam. Por exemplo, o compartilhamento de fotos em restaurantes tende a acontecer mais nos horários de almoço e jantar. Aplicações baseadas nesse tipo de sensoriamento devem considerar que a participação do usuário pode variar significativamente ao longo do tempo.

Outra observação interessante relacionada ao intervalo de tempo Δ_t entre compartilhamentos pode ser extraída da figura 8-b, que mostra a função razão de chances⁹ (RC) desses intervalos. A RC é uma função acumulada que mostra claramente o comportamento cumulativo de uma dada distribuição na cabeça quanto na cauda. Sua fórmula é $RC(\Delta_t) = \frac{CDF(\Delta_t)}{1-CDF(\Delta_t)}$, onde $CDF(\Delta_t)$ é a função de densidade acumulada da distribuição sendo analisada, no caso a distribuição dos intervalos de tempo Δ_t entre compartilhamentos. Como em [29], a RC do intervalo de tempo entre fotos compartilhadas também

mostra um comportamento de lei de potência com inclinação $\rho \approx 1$. Isso sugere que os mecanismos por trás das atividades humanas podem ser mais simples e gerais do que aqueles propostos na literatura, pois dependem de uma grande quantidade de parâmetros [14]. A figura 8-c mostra a distribuição do intervalo entre eventos. Podemos observar que uma fatia significativa dos usuários realiza compartilhamento consecutivo de fotos em um curto intervalo de tempo. Cerca de 20% de todo compartilhamento observado acontece em até 10 minutos. Como será discutido na seção 4.5, isso sugere que os nós tendem a compartilhar mais de uma foto na mesma área.

4.4. SAZONALIDADE

Analisamos agora como a rotina dos humanos afeta o compartilhamento dos dados. A figura 9-a mostra o padrão semanal de compartilhamento de fotos do Instagram¹⁰. Como esperado, a atuação da rede de participação apresenta um padrão diurno, o que implica que durante a madrugada a atividade de sensoriamento é bastante baixa.

Considerando dias de semana, é possível observar um ligeiro aumento da atividade ao longo da semana, com exceção de terça-feira, quando há um pico de atividade. No trabalho [5], que analisou sistemas para compartilhamento de localização, foi observado esse mesmo comportamento, sem nenhum dia como exceção. Isso sugere que durante o período de coleta pode ter ocorrido um evento atípico durante a terça-feira que gerou muitos compartilhamento de fotos. Por fim, pode-se observar dois picos de atividades ao longo do dia, por volta dos horários de almoço e jantar. Diferentemente do comportamento observado para o compartilhamento de localizações [5], não foi observado picos de atividade no compartilhamento de fotos por volta do horário do café da manhã.

Analisamos ainda os diferentes padrões de comportamento para dias de semana e final de semana. A figura 9-

⁹Odds ratio function.

¹⁰O horário do compartilhamento foi normalizado de acordo com o local onde a foto foi tirada, utilizando para isso a informação geográfica do local.

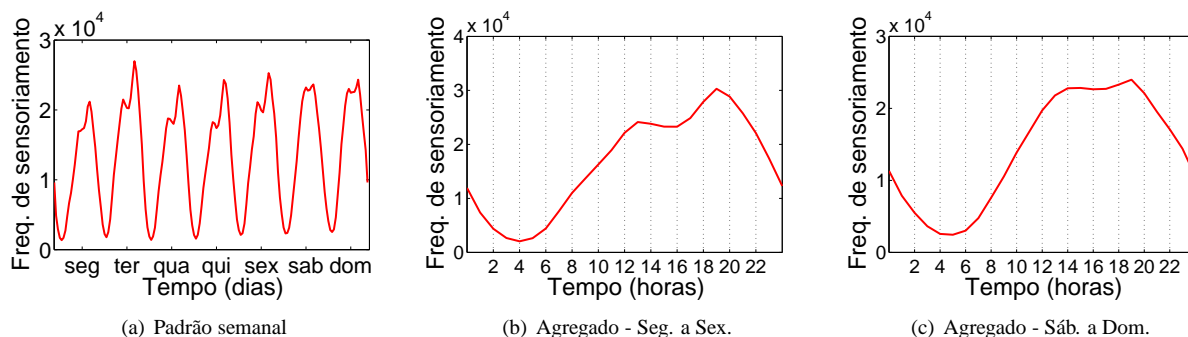


Figure 9. Padrão do compartilhamento de fotos durante os dias da semana

b mostra o número médio de fotos compartilhadas por hora, de segunda-feira a sexta-feira. A figura 9-c também mostra a mesma informação para sábado e domingo. Como podemos observar, os picos durante os dias de semana acontecem por volta de 13:00 (almoço) e 19:00 (jantar). Já no final de semana não é observado um pico de atividade claro durante o horário do almoço. Pelo contrário, a atividade permanece intensa durante toda a tarde até o início da noite, com um ligeiro aumento por volta das 19:00.

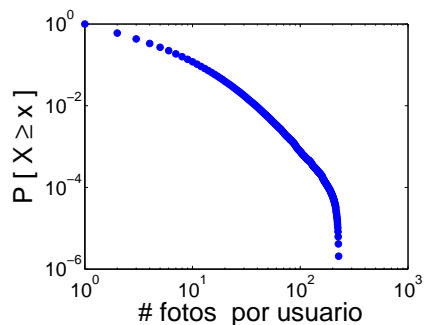
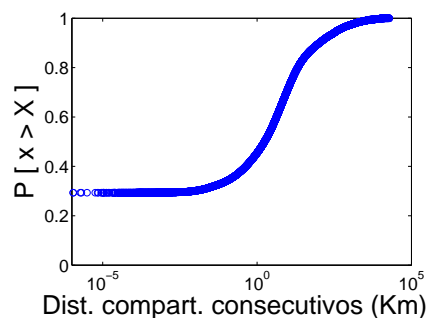


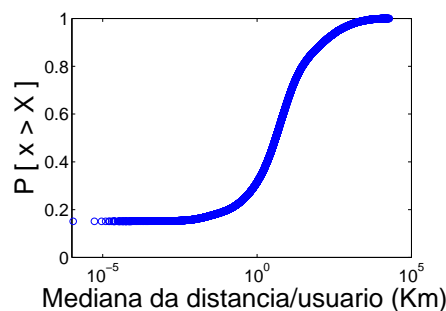
Figure 10. Distribuição do número de fotos compartilhadas pelos usuários

4.5. COMPORTAMENTO DOS NÓS

Nesta seção é analisado o desempenho dos nós da RSP (i.e., dos usuários) quanto ao compartilhamento de fotos. A figura 10 mostra que a distribuição do número de fotos compartilhadas por cada usuário da nossa base de dados possui cauda pesada, significando que a participação dos usuários pode variar muito. Por exemplo, aproximadamente 40% dos usuários contribuíram com apenas uma foto no período considerado, enquanto que somente 17% e 0.1% dos usuários contribuíram com mais de 10 e 100 fotos, respectivamente. É natural que essa variabilidade aconteça por diversos motivos. Por exemplo, alguns usuários podem dar mais importância para quesitos de privacidade do que outros.



(a) Todas as distâncias



(b) Mediana da distância por usuário

Figure 11. Distribuição da distância geográfica entre fotos consecutivas de um mesmo usuário.

Analisamos também a distância geográfica entre dois compartilhamentos de fotos consecutivos pelo mesmo usuário, usando, para tal, as coordenadas geográficas associadas a cada foto. A figura 11-a mostra a função de densidade acumulada da distância geográfica entre cada par de fotos consecutivas compartilhadas por cada usuário do nosso conjunto de dados. Pode-se observar que uma fatia significativa (aproximadamente 30%) das distâncias entre fotos consecutivas são muito curtas (menos de 1 metro). Isso indica que os usuários tendem a compartilhar várias fotos no mesmo local. Essa hipótese é reforçada pela significativa fatia de intervalos de tempo entre fotos consecutivas de curta duração mostrada na

figura 8-c: 20% destes intervalos (Δ_t) não ultrapassam 10 minutos. Isso não foi observado na mesma proporção para o compartilhamento de localização. Em [16], por exemplo, foi observado que 20% dos compartilhamentos de localizações acontece em até 1 km de distância. Para o compartilhamento de fotos, esse valor chega a aproximadamente 45%. Esse resultado pode ser explicado pelo simples fato de que uma foto pode conter muito mais informações que uma localização. Por exemplo, em um restaurante os usuários poderiam compartilhar fotos dos amigos presentes, da comida, ou de uma situação particular, mas tenderiam a compartilhar sua localização apenas uma única vez.

Por fim, analisamos cada usuário separadamente. A figura 11-b mostra a distribuição das medianas das distâncias entre compartilhamentos consecutivos computadas para cada usuário. Note que pelo menos 50% das fotos consecutivas de uma parcela significativa de usuários (aproximadamente 20%) são tiradas a uma distância muito pequena (≈ 1 metro).

5. APLICAÇÃO

É bastante comum existirem áreas em uma cidade que despertam um maior interesse dos residentes ou visitantes, os aqui denominados *pontos de interesse* (PDI). Dentre os PDIs mais visitados, podemos mencionar os pontos turísticos da cidade. No entanto, nem todos os PDIs de uma cidade são pontos turísticos. Por exemplo, uma área de bares pode ser bastante popular entre os residentes da cidade, mas sem atrativos turísticos. Além disso, PDIs são dinâmicos, ou seja, áreas que são populares hoje podem não o ser amanhã.

Assim, uma aplicação que emerge naturalmente a partir da análise de dados do Instagram é de identificação de PDIs em uma cidade. Isso é possível porque cada foto representa, implicitamente, um interesse de um indivíduo em um determinado instante. Com isso, quando muitas fotos de um determinado local são compartilhadas em um determinado instante, pode-se inferir que esse local é um PDI (observe a figura 6). Mais especificamente, o processo de identificação de PDIs envolve os seguintes passos:

1. Considera-se que cada par i de coordenadas (longitude, latitude) $(x, y)_i$ está associada a um ponto p_i ;
2. calcula-se a distância [25] entre cada par de pontos (p_i, p_j) ;
3. agrupa-se todos os pontos p_i que possuem uma distância inferior a 250 metros em um *cluster* C_k . Essa distância limite foi obtida através do método Acoplamento Completo (Complete-Linkage) [26]. O re-

sultado desse procedimento é exibido na figura 12-a. Nessa figura cores diferentes¹¹ representam diferentes clusters C_k para a cidade de Belo Horizonte;

4. para cada cluster C_k , consideramos apenas um ponto (foto) por usuário. Com isso, a popularidade de um cluster se baseia no número de diferentes usuários que compartilharam uma foto na área do cluster. Este procedimento evita considerar as áreas visitadas por poucos usuários, por exemplo casas, como populares;
5. para cada cluster C_k , cria-se um cluster alternativo C_r , assim $r = k$. A princípio cada cluster alternativo não possui nenhuma foto associada a ele. Em seguida, para cada foto f_i , escolhemos um cluster alternativo C_r de forma aleatória e atribuímos f_i a C_r . Após distribuir todas as fotos f_i seguindo esse procedimento, comprovamos que a distribuição do número de fotos atribuídas para cada cluster C_r é explicada por uma distribuição normal com média μ e desvio padrão σ ;
6. por fim, dos clusters originais C_k encontrados a partir do passo 3, excluimos todos aqueles em que o número de fotos do mesmo está a uma distância 2σ da média μ , ou seja, está no intervalo $[\mu - 2\sigma; \mu + 2\sigma]$. De acordo com a regra dos três sigmas (*three-sigma rule*) esse intervalo representa $\approx 95\%$ da distribuição de fotos aleatórias nos clusters C_r . A ideia deste passo é excluir aqueles clusters que podem ter sido gerados por situações aleatórias, ou seja, clusters que provavelmente não refletem um PDI da cidade.

Os PDIs obtidos através desse processo são mostrados na figura 12-b. Além de identificar PDIs em uma cidade, podemos separar dos PDIs os pontos turísticos. Para isso, primeiramente geramos um grafo $G(V, E)$, onde os vértices $v_i \in V$ são todos os PDIs e uma aresta (i, j) existe do vértice v_i para o vértice v_j se em um determinado momento um usuário compartilhou uma foto em um PDI v_j , logo após ter compartilhado uma foto no PDI v_i .

O peso $w(i, j)$ de uma aresta representa o número total de transições realizadas do PDI v_i para o PDI v_j , considerando as transições de todos os usuários. Para identificar pontos turísticos, consideramos que grande parte dos turistas seguem uma trajetória bem conhecida dentro da cidade, sendo guiada pelos principais pontos turísticos da mesma. Além disso, em cada ponto turístico ele tira uma ou mais fotos e parte para o ponto turístico seguinte. Dessa maneira, consideramos que arestas (i, j) com pesos $w(i, j)$ altos denotam essas transições frequentes de um ponto turístico para outro em uma cidade.

¹¹Devido ao grande número de clusters algumas cores se repetem, mas isso não possui nenhum significado especial.

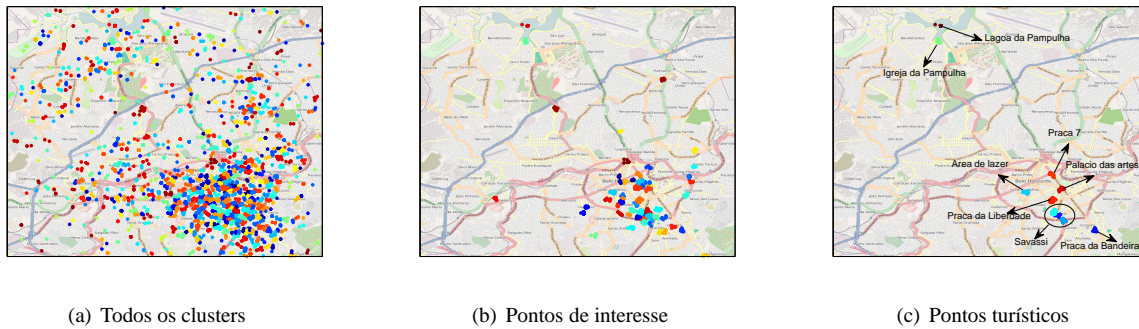


Figure 12. Pontos de Interesse de Belo Horizonte

Feito isso, excluimos de G todas as arestas (i, j) com peso $w(i, j)$ menor que um limiar t , que é dado pela probabilidade de gerar $w(i, j)$ aleatoriamente em um grafo aleatório $G_R(V, E_R)$. A identificação do valor que separa arestas de peso alto das de peso baixo é feita da seguinte maneira. Primeiro, criamos um grafo aleatório $G_R(V, E_R)$ contendo os mesmos nós de G . Depois, para cada sequência de n_u fotos $f_u^1, f_u^2, \dots, f_u^{n_u}$ de cada usuário u , atribuímos PDIs aleatoriamente para cada foto e geramos as arestas aleatórias E_R de G_R a partir dessa nova atribuição. Assim, a sequência das fotos é aleatória, o que gera arestas e pesos de arestas aleatórios em G_R , mas preserva o número total de fotos que foi tirada em um determinado local. A ideia é simular trajetórias aleatórias na cidade. Desta maneira aleatória, a distribuição dos pesos das arestas segue uma distribuição normal $N_w(\mu_w, \sigma_w)$ com média μ_w e desvio padrão σ_w .

De acordo com $N_w(\mu_w, \sigma_w)$, quando a probabilidade de gerar $w(i, j)$ em $G_R(V, E_R)$ é próxima de zero, ou seja, quando for pouco provável que o grafo aleatório $G_R(V, E_R)$ tenha um peso de aresta $w(i, j)$, então a transição de $v_i \rightarrow v_j$ é uma transição popular na cidade sendo, de acordo com a nossa conjectura, uma transição entre pontos turísticos. Para o nosso conjunto de dados, o valor de t que fornece uma probabilidade próxima de 0 é $t = 10$. Como podemos observar na figura 12-c, os vértices (PDIs) do grafo resultante representam a maioria dos pontos turísticos de Belo Horizonte. As áreas dos PDIs resultantes cobrem sete de todos os oito pontos recomendados pelo TripAdvisor¹², sendo as áreas culturais e de lazer mais importantes de BH.

É interessante notar a diferença entre as figuras 12-b e 12-c, a primeira contendo todos os PDIs e a segunda somente os pontos turísticos da cidade de BH. Novamente, esta aplicação é interessante porque ela é capaz de identificar os PDIs em um contexto espaço-temporal, o que é fundamental, uma vez que os PDIs são dinâmicos e mudam ao longo do tempo.

6. CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho apresentamos o que é, no melhor de nosso conhecimento, a primeira caracterização do Instagram. A análise do sistema foi feita tratando-o como uma rede de sensoriamento participativa. Assim, discutimos a cobertura espacial e temporal desta rede mostrando a sua abrangência planetária bem como uma frequência de compartilhamento de fotos espaço-temporal muito desigual e correlacionada com as rotinas de atividades humanas. Também discutimos uma aplicação que demonstra o potencial de uma RSP derivada do Instagram para o estudo da dinâmica de cidades.

Como trabalhos futuros, pretendemos analisar outras RSPs e desenvolver novas aplicações que explorem estas redes. Por exemplo, podemos imaginar aplicações que consideram conjuntamente dados provenientes de outros sistemas de sensoriamento participativo como o Waze (condições de tráfego) e o Weddar (condições meteorológicas), considerando inclusive, diferentes categorias/interesses das pessoas.

REFERENCES

- [1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393 – 422, 2002.
- [2] F. Barth. *Ethnic groups and boundaries: the social organization of culture difference*. Scandinavian university books. Little, Brown, 1969.
- [3] Mark Bilandzic and Marcus Foth. A review of locative media, mobile and embodied spatial interaction. *International Journal of Human-Computer Studies*, 70(1):66–71, January 2012.
- [4] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Partici-

¹²www.tripadvisor.com

- patory sensing. In *Proc. Workshop on World-Sensor-Web (WSW)*, Boulder, Colorado, USA, 2006.
- [5] Zhiyuan Cheng, James Caverlee, Kyumin Lee, and Daniel Z. Sui. Exploring Millions of Footprints in Location Sharing Services. In *Proc. 5th Int'l Conference on Weblogs and Social Media*, Barcelona, Spain, 2011.
- [6] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proc. 17th ACM Int'l Conf. on Knowledge Discovery and Data Mining (KDD'11)*, San Diego, CA, USA, 2011.
- [7] Justin Cranshaw, Raz Schwartz, Jason I. Hong, and Norman Sadeh. The Livehoods Project: Utilizing Social Media to Understand the Dynamics of a City. In *Proc. 6th Int. Conference on Weblogs and Social Media (ICWSM'12)*, Dublin, Ireland, 2012.
- [8] Katy Daniells. Infographic: Instagram statistics 2012. *Digital Buzz Blog*, 2012.
- [9] Shane B. Eisenman, Emiliano Miluzzo, Nicholas D. Lane, Ronald A. Peterson, Gahng-Seop Ahn, and Andrew T. Campbell. Bikenet: A mobile sensing system for cyclist experience mapping. *ACM Transactions on Sensor Networks*, 6(1), 2010.
- [10] Peter R. Fisk. The graduation of income distributions. *Econometrica*, 29(2):171–185, 1961.
- [11] M.F. Goodchild. Citizens as sensors: The world of volunteered geography. *GeoJournal*, 69(4):211–221, 2007.
- [12] J. Krumm. *Ubiquitous Computing Fundamentals*. Chapman & Hall/CRC, 1st ed., 2009.
- [13] Eric C. Larson, TienJui Lee, Sean Liu, Margaret Rosenfeld, and Shwetak N. Patel. Accurate and privacy preserving cough sensing using a low-cost microphone. In *Proc. 13th Int. Conference on Ubiquitous Computing (Ubicomp'11)*, Beijing, China, 2011.
- [14] R. Dean Malmgren, Daniel B. Stouffer, Adilson E. Motter, and Luís A. N. Amaral. A poissonian explanation for heavy tails in e-mail communication. *Proc. National Academy of Sciences*, 105(47):18153–18158, November 2008.
- [15] Afra J. Mashhadi and Licia Capra. Quality Control for Real-time Ubiquitous Crowdsourcing. In *Proc. 2nd Int. Workshop on Ubiquitous Crowdsourcing*, Beijing, China, 2011.
- [16] Anastasios Noulas, Salvatore Scellato, Cecilia Mascolo, and Massimiliano Pontil. An Empirical Study of Geographic User Activity Patterns in Foursquare. In *Proc. of the Fifth Int'l Conf. on Weblogs and Social Media (ICWSM'11)*, Barcelona, Spain, 2011.
- [17] Anastasios Noulas, Salvatore Scellato, Cecilia Mascolo, and Massimiliano Pontil. Exploiting Semantic Annotations for Clustering Geographic Areas and Users in Location-based Social Networks. In *Proc. 5th Int. Conf. on Weblogs and Social Media (ICWSM'11)*, Barcelona, Spain, 2011.
- [18] Rajib Kumar Rana, Chun Tung Chou, Salil S. Kanhere, Nirupama Bulusu, and Wen Hu. Ear-phone: an end-to-end participatory urban noise mapping system. In *Proc. 9th ACM/IEEE Int. Conference on Information Processing in Sensor Networks*, Stockholm, Sweden, 2010.
- [19] Sasank Reddy, Deborah Estrin, Mark Hansen, and Mani Srivastava. Examining micro-payments for participatory sensing data collections. In *Proc. 12th ACM Int. Conference on Ubiquitous Computing (Ubicomp'10)*, Copenhagen, Denmark, 2010.
- [20] Stefan Saroiu and Alec Wolman. I am a sensor, and i approve this message. In *Proc. 11th Workshop on Mobile Computing Systems and Applications*, Annapolis, Maryland, 2010.
- [21] Salvatore Scellato, Anastasios Noulas, Renaud Lambiotte, and Cecilia Mascolo. Socio-spatial Properties of Online Location-based Social Networks. In *Proc. 5th Int. Conf. on Weblogs and Social Media*, Barcelona, Spain, 2011.
- [22] Thiago H. Silva, Pedro O. S. Vaz de Melo, Jussara M. Almeida, and Antonio A. F. Loureiro. Uncovering Properties in Participatory Sensor Networks. In *Proc. 4th ACM Int. Workshop on Hot Topics in Planet-scale Measurement (HotPlanet'12)*, Low Wood Bay, Lake District, UK, 2012.
- [23] Thiago H. Silva, Pedro O. S. Vaz de Melo, Jussara M. Almeida, and Antonio A. F. Loureiro. Challenges and opportunities on the large scale study of city dynamics using participatory sensing. In *IEEE International Symposium on Computers and Communications (ISCC'13)*, Split, Croatia, July 2013.
- [24] Thiago H. Silva, Pedro O. S. Vaz de Melo, Jussara M. Almeida, Juliana Salles, and Antonio A. F. Loureiro. Visualizing the invisible image of cities. In *Proc. IEEE International Conference on Cyber, Physical and Social Computing*, 2012.

- [25] R. W. Sinnott. Virtues of the Haversine. *Sky and Telescope*, 68(2):159+, 1984.
- [26] T. Sørensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biologiske Skrifter*, 5(4), 1948.
- [27] Mani Srivastava, Tarek Abdelzaher, and Boleslaw Szymanski. Human-centric sensing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1958):176–197, January 2012.
- [28] Marisa Vasconcelos, S. Ricci, J. Almeida, F. Benvenuto, and V. Almeida. Caracterização e Influência do Uso de Tips e Dones no Foursquare. In *Proc. XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC'12)*, Ouro Preto, MG, Brasil, 2012.
- [29] Pedro Olmo Stancioli Vaz de Melo, C. Faloutsos, and A. A. Loureiro. Human dynamics in large communication networks. In *Proc. of the Int. Conference on Data Mining (SDM'11)*, Mesa, Arizona, USA, 2011.
- [30] Mark Weiser. The Computer in the 21st Century. *Scientific American*, 265(3):94–104, 1991.

RemapRoute: Remapeando Mudanças de Rota na Internet

Ítalo Cunha¹, Renata Teixeira², Darryl Veitch³, Christophe Diot⁴

¹ Departamento de Ciência da Computação
Universidade Federal de Minas Gerais, Belo Horizonte, MG

² INRIA

³ Department of Electrical and Electronic Engineering
University of Melbourne, Australia

⁴ Technicolor

Abstract

Mapas topológicos da Internet coletados com trace-route podem estar incompletos ou desatualizados porque não podemos realizar medições em frequência suficiente para detectar todas as mudanças de rota. Neste artigo mostramos que mudanças de rota geralmente afetam poucos roteadores. Exploramos essa propriedade no RemapRoute, nossa ferramenta para remapeamento local de mudanças de roteamento que sonda apenas os poucos roteadores afetados ao invés de toda a rota. Nossa avaliação via simulação e um protótipo real mostra que RemapRoute reduz significativamente o número de sondas de remapeamento, sem comprometer a exatidão e a latência de remapeamento. Esta redução do custo de remapeamento nos propicia construir mapas topológicos mais completos e atualizados.

Keywords: Internet, mapeamento, monitoramento, mudanças de roteamento

1. INTRODUÇÃO

Sistemas para identificação de falhas na Internet coletam medições frequentes de rotas na rede [9, 12, 15, 16]. De forma similar, redes de distribuição de conteúdo medem rotas na Internet para escolher o melhor servidor para atender uma requisição [10]. Esses e outros sistemas medem rotas frequentemente na esperança de rastrear mudanças de roteamento à medida que elas acontecem.

Medições de rota na Internet são frequentemente coletadas com traceroute [1, 14, 25], que envia sondas para identificar uma sequência de roteadores entre uma origem e um destino. A banda disponível para enviar sondas é finita. Medir rotas para mapear a topologia requer um grande número de sondas e pode levar de vários minutos a alguns dias [4, 6, 22]. É impossível medir com frequência suficiente para detectar todas as mudanças de roteamento

sem sobrecarregar a rede. Consequentemente, mapas da topologia da Internet podem estar desatualizados ou inconsistentes pois mudanças de roteamento podem acontecer durante o processo de medição.

Nosso sistema DTRACK rastreia mudanças de roteamento na Internet para manter mapas da topologia da Internet mais atualizados [7]. O DTRACK separa as tarefas de detectar e remapear mudanças de roteamento. Para detectar mudanças, o DTRACK usa um processo de sondagem leve que combina duas ideias: (1) redirecionar sondas de caminhos estáveis onde mudanças de roteamento são improváveis para caminhos instáveis onde mudanças são mais prováveis; e (2) espalhar sondas de forma uniforme na rede para reduzir medições redundantes. Para remapear mudanças, o DTRACK usa o Paris traceroute [1, 25] para medir a nova rota por inteiro. O Paris traceroute é uma versão moderna do traceroute capaz de identificar roteadores que fazem balanceamento de carga. Usamos o Paris traceroute por que é impossível inferir mudanças de roteamento de forma precisa sem informação sobre roteadores que fazem balanceamento de carga [6].

O DTRACK mantém um banco de dados com a última rota observada em cada um dos caminhos monitorados. Para detectar mudanças, o DTRACK envia uma sonda num ponto s' do caminho e compara a resposta da sonda com a última rota observada. Se a resposta da sonda for incompatível com a última rota observada, e.g., o endereço IP do roteador que enviou a resposta não pertence à última rota observada, uma mudança é detectada e o processo de remapeamento é disparado. Atualmente, o DTRACK remapeia o caminho por inteiro usando o Paris traceroute. Esta abordagem garante medição correta da nova rota, mas desperdiça várias sondas pois mudanças de caminho envolvem poucos roteadores (seção 3). Em particular, esta abordagem ignora duas informações disponíveis quando o processo de remapeamento é disparado: a última rota observada e o ponto s' onde a mudança foi detectada.

Neste artigo propomos RemapRoute, uma ferramenta para reduzir o custo do remapeamento de mudanças de roteamento na Internet (seção 4). Dadas a última rota observada e um ponto onde uma mudança foi detectada, o RemapRoute envia sondas em pontos estratégicos para localizar a mudança e remapeá-la localmente, sem desperdiçar sondas nos roteadores que não estão envolvidos na mudança. Nossa avaliação via simulação dirigida por dados reais mostra que RemapRoute reduz pela metade o custo do remapeamento de 88% das mudanças de roteamento em nossos dados (seção 5). A redução de custo é ainda maior em rotas longas ou com roteadores que fazem balanceamento de carga. Nossa avaliação de um protótipo do RemapRoute usando o PlanetLab confirma nossos resultados via simulação e demonstra a eficácia da ferramenta (seção 6). Sumarizando, neste artigo fazemos as seguintes contribuições:

- Caracterizamos mudanças de roteamento na Internet e mostramos que elas envolvem uma fração pequena dos roteadores numa rota (seção 3);
- Propomos métodos para localizar e remapear mudanças de roteamento que reduzem o desperdício de sondas (seção 4);
- Mostramos que nossa ferramenta reduz o custo de remapeamento de mudanças de rota via simulação e em cenários reais (seções 5 e 6).

A economia de sondas no processo de remapeamento de mudanças de roteamento aumenta o número de sondas disponíveis para mapeamento topológico. Podemos utilizar estas sondas para monitorar mais caminhos na Internet e melhorar a cobertura dos mapas da topologia, ou aumentar a frequência de sondagem e melhorar o rastreamento de mudanças de roteamento. RemapRoute é mais um passo na construção de mapas da topologia da Internet mais completos e consistentes.

2. DEFINIÇÕES E FUNDAMENTOS

Seguindo a nomenclatura proposta por Paxson [20], chamamos de *caminho virtual* a conectividade entre uma origem e um destino na Internet. Em um dado momento, um caminho virtual é instanciado por uma *rota*. Devido a mudanças de roteamento, um caminho pode ser visto como um processo contínuo $C(t)$ que muda de uma rota a outra ao longo do tempo. Uma rota é composta de *saltos* (*hops*) que são instanciados por roteadores. Saltos são enumerados a partir da origem e nos referimos a um salto s numa rota $C(t)$ por $C(t)[s]$. Uma rota pode ser *simples* se ela tem apenas uma sequência de roteadores da origem ao destino; ou *ramificada* se ela tem roteadores que realizam balanceamento de carga e múltiplas sequências

sobrepostas de roteadores da origem ao destino. Todos os saltos numa rota simples têm apenas um roteador e pelo menos um salto numa rota ramificada tem mais de um roteador.

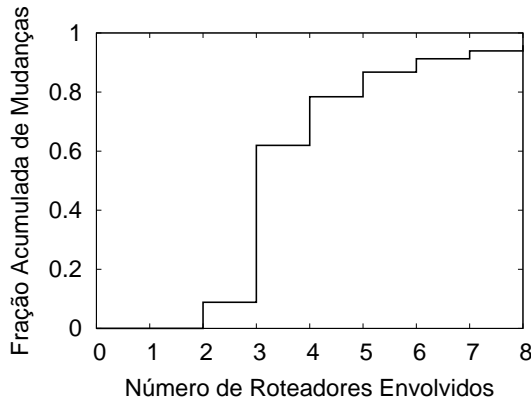
Dadas duas medições consecutivas de um caminho nos instantes $C(t_i)$ e $C(t_{i+1})$, definimos uma *mudança de caminho* como uma sequência de saltos contíguos no novo caminho que altera o caminho antigo. Computamos mudanças de caminho minimizando o número de edições (adição, remoção e substituição de saltos) necessárias para transformar a nova rota na rota antiga. Definimos o *salto de divergência* s_d e o *salto de convergência* s_c de uma mudança como os saltos imediatamente anterior e posterior aos saltos editados pela mudança, respectivamente. Dizemos que o salto de divergência, o salto de convergência e todos os saltos entre eles estão *envolvidos* na mudança de roteamento. Exemplificando, se $C(t_i) = \{a, b, c, d, e, f, g\}$ e $C(t_{i+1}) = \{a, b, e, x, y, g\}$, temos uma mudança com $s_d = 1$ e $s_c = 2$ (remoção de c e d), e outra mudança com $s_d = 2$ e $s_c = 5$ (troca de f por x e y).

3. CARACTERIZAÇÃO DE MUDANÇAS DE CAMINHOS

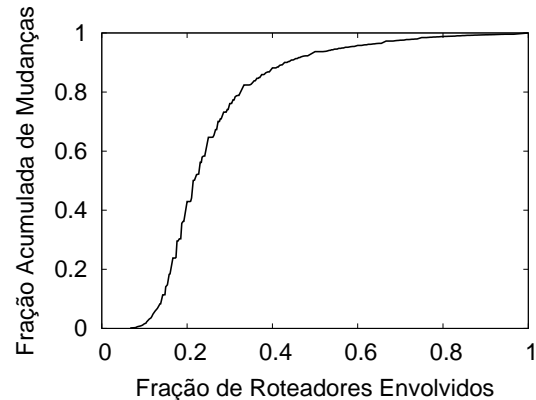
Nesta seção caracterizamos um conjunto de mudanças de caminho reais e verificamos que mudanças de caminho na Internet afetam poucos roteadores.

Implantamos o DTRACK [7] para medir caminhos em 72 monitores no PlanetLab por uma semana a partir de 4 de março de 2011. Cada monitor escolhe 1.000 destinos aleatoriamente de uma lista com 34.820 destinos alcançáveis escolhidos aleatoriamente na Internet. Em cada monitor, configuramos o DTRACK para rastrear mudanças nos caminhos escolhidos enviando 8 sondas por segundo, similar à taxa de sondagem do DIMES [21]. Em uma semana observamos 1.202.960 mudanças de caminho. Os caminhos medidos atravessam 7.315 sistemas autônomos e 97% dos sistemas autônomos de grande porte [19].

A figura 1a mostra a distribuição do número de saltos envolvidos em mudanças de caminho na Internet. O número de saltos envolvidos numa mudança de caminho é o mínimo de saltos que precisamos remapear para saber qual é a nova rota. Vemos que 78% das mudanças de caminho afetam quatro ou menos saltos, um número pequeno visto que a mediana do tamanho das rotas em nossos dados é 17 saltos. O tipo de mudança de caminho mais comum é quando o roteador de apenas um salto muda, resultando em mudanças que envolvem três saltos: o salto onde o roteador mudou mais os saltos de convergência e divergência. Notamos que 9% das mudanças de caminho envolvem dois saltos. Estas mudanças apenas removem saltos da rota antiga (a nova rota está contida na rota an-



(a) Distribuição do número roteadores envolvidos em mudanças de caminho



(b) Distribuição da fração de roteadores de uma rota envolvidos em mudanças de caminho

Figura 1: Caracterização de mudanças de caminho na Internet

tiga) e os únicos saltos envolvidos são os saltos de convergência e divergência. Causas típicas de mudanças de caminho que envolvem dois saltos são falhas de conectividade na Internet e erros de medição onde é impossível medir os saltos atrás da falha ou erro.

A figura 1b mostra a distribuição da fração de saltos de um caminho envolvidos numa mudança, i.e., o número de roteadores envolvidos na mudança dividido pelo tamanho da nova rota, para todas as mudanças nos nossos dados. Vemos que 76% das mudanças de caminho envolvem menos de 30% dos saltos no caminho. Isso mostra o potencial do remapeamento local para redução de sondas de medição, se comparado com a abordagem atual de remapear o caminho por inteiro. A curva começa em $x = 0.066 = 2/30$ acontece porque uma mudança envolve pelo menos dois saltos e porque o DTRACK só mede os 30 primeiros saltos num caminho (o padrão do traceroute e do Paris traceroute [1, 14]).

A figura 2 mostra a distribuição do número de sistemas autônomos envolvidos em mudanças de caminho. Convertemos endereços IPs medidos pelo DTRACK em sistemas autônomos combinando as tabelas de mapeamento do Team Cymru¹ e iPlane [17]. Cada endereço IP que não aparece na tabela de mapeamento combinada é associado a um sistema autônomo que contém apenas um endereço IP. Essa é uma medida conservadora que pode sobrestimar o número de sistemas autônomos envolvidos em uma mudança. A figura 2 explica porque mudanças de roteamento envolvem poucos saltos. Mesmo fazendo a conversão de endereços IP para sistemas autônomos de forma conservadora, vemos que 60% das mudanças de caminho é interna a um sistema autônomo e apenas 7% envolve mais de dois sistemas autônomos.

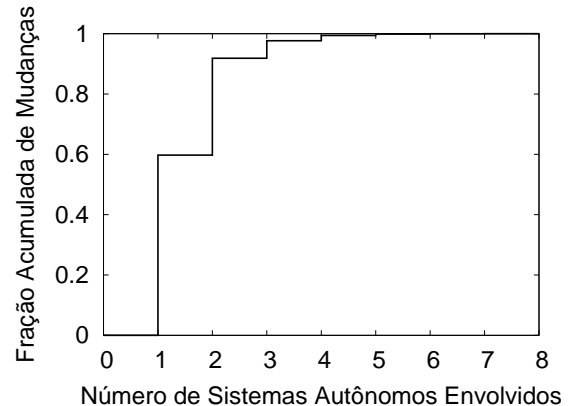


Figura 2: Distribuição do número de sistemas autônomos envolvidos em mudanças de caminho

4. REMAPROUTE

Agora apresentamos nosso algoritmo para remapeamento local de mudanças de caminho implementado no RemapRoute. O algoritmo recebe como entrada a rota antiga observada antes da mudança de roteamento $C(t_{i-1})$ e o salto s' do roteador onde a mudança foi detectada, $C(t_i)[s'] \neq C(t_{i-1})[s']$. Para remapear a mudança de rota, RemapRoute opera em duas etapas: primeiro ele localiza onde a mudança aconteceu (seção 4.1) e depois faz o remapeamento local da mudança (seção 4.2).

O processo de localização e remapeamento da mudança de rota envolve remapear saltos da rota atual e compará-los com saltos na rota antiga. Para remapear cada salto, o RemapRoute envia múltiplas sondas, modificando sistematicamente os campos do cabeçalho IP como o Paris traceroute [1, 25], para identificar roteadores que fazem balanceamento de carga.

¹Team Cymru, IP to ASN Mapping, <http://www.team-cymru.org/Services/ip-to-asn.html>

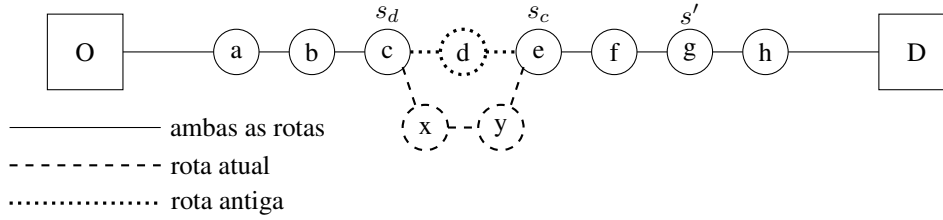


Figura 3: Exemplo de mudança de roteamento com adição de roteadores à rota.

4.1. LOCALIZAÇÃO DA MUDANÇA DE ROTEAMENTO

O RemapRoute parte de um salto s' onde o roteador na rota atual é diferente do roteador na rota antiga, $C(t_i)[s'] \neq C(t_{i-1})[s']$. Se o roteador da rota atual no salto s' não pertencer à rota antiga, $C(t_i)[s'] \notin C(t_{i-1})$, então s' é um dos saltos envolvidos na mudança de caminho e podemos passar para a próxima etapa para remapear a mudança (seção 4.2).

Se o roteador da rota atual no salto s' pertencer à rota antiga em outro salto s'' , $C(t_i)[s'] = C(t_{i-1})[s'']$, então temos uma mudança de roteamento em saltos anteriores a s' que adicionou ou removeu roteadores na rota. A figura 3 mostra um exemplo de uma mudança de roteamento onde o roteador d foi substituído por dois roteadores x e y . Uma sonda para o sétimo salto detecta uma mudança de roteamento pois a resposta da sonda vem do roteador $g = C(t_i)[7]$, que não é a resposta esperada na rota antiga, $h = C(t_{i-1})[7]$.

Para encontrar um roteador envolvido na mudança de caminho, i.e., um roteador na rota atual que não está presente na rota antiga, o RemapRoute faz uma busca binária no caminho. O RemapRoute inicializa $s_{\text{esq}} = 0$ e $s_{\text{dir}} = s'$. A cada iteração da busca, o RemapRoute remapeia o salto $s = (s_{\text{esq}} + s_{\text{dir}})/2$ e procura o roteador no salto s da rota atual na rota antiga. Se o roteador no salto s da rota atual não pertencer à rota antiga, $C(t_i)[s] \notin C(t_{i-1})$, a busca termina e passamos para a etapa de remapeamento. Se o roteador no salto s da rota atual estiver no salto s da rota antiga, $C(t_i)[s] = C(t_{i-1})[s]$, então a mudança está à direita de s e fazemos $s_{\text{esq}} = s$. Se o roteador no salto s da rota atual estiver em outro salto s'' na rota antiga, $C(t_i)[s] = C(t_{i-1})[s'']$, a mudança está à esquerda de s e fazemos $s_{\text{dir}} = s$.

Não temos como comparar a rota atual com a rota antiga se o roteador no salto s não responde a sondas. Neste caso tomamos a decisão conservadora de decrementar s e continuar a busca no salto anterior, sem alterar s_{esq} e s_{dir} .

Se uma mudança de caminho apenas remove saltos da rota antiga, então não existe salto na rota atual que não pertença à rota antiga. Neste caso, o processo de busca termina com $s_{\text{esq}} = s_{\text{dir}} + 1$ e nenhum remapeamento é necessário (seção 4.2).

4.2. REMAPEAMENTO LOCAL

O remapeamento local parte de um salto s onde o roteador na rota atual não pertence à rota antiga, $C(t_i)[s] \notin C(t_{i-1})$. O RemapRoute remapeia sequencialmente os saltos da rota atual posteriores a s até encontrar o salto de convergência $s_c > s$ com um roteador que pertence à rota antiga, $C(t_i)[s_c] \in C(t_{i-1})$. Se uma das rotas não chega até o destino, o salto de convergência pode não existir. Neste caso, o RemapRoute remapeia a rota atual até o destino ou até o último salto alcançável. Se o caminho não chega até o destino, o RemapRoute identifica o último salto alcançável após encontrar três saltos consecutivos que não respondem a sondas (igual o traceroute).

De forma similar, o RemapRoute remapeia sequencialmente os saltos da rota atual anteriores a s até encontrar o salto de divergência $s_d < s$ com um roteador que pertence à rota antiga, $C(t_i)[s_d] \in C(t_{i-1})$. No pior caso, a busca pelo salto de divergência termina na origem, que pertence a qualquer rota no caminho. Se o salto s_d não for idêntico nas duas rotas, $C(t_i)[s_d] \neq C(t_{i-1})[s_d]$, então existe outra mudança de roteamento anterior a s_d e voltamos à primeira etapa (seção 4.1), fazendo $s' = s_d$, para localizá-la e depois remapeá-la. Esse processo é realizado recursivamente até que não reste nenhuma mudança para remapear. O remapeamento dos roteadores nos saltos entre s_d e s_c precisa ser sequencial pois todos estão envolvidos na mudança de roteamento. Para mudanças de roteamento que apenas removem saltos, temos $s_d = s_{\text{esq}}$ e $s_c = s_{\text{dir}}$ e nenhum remapeamento é necessário.

4.3. EXEMPLO

Considere que a mudança de roteamento mostrada na figura 3 tenha sido detectada com uma sonda para o quinto salto no caminho. Temos $C(t_{i-1})[5] = f$ e $C(t_i)[5] = e$. Como $e \in C(t_{i-1})$, um nó foi inserido antes do quinto salto e o RemapRoute inicia uma busca binária para identificar o local da mudança. O RemapRoute começa remapeando o segundo salto, onde $C(t_{i-1})[2] = C(t_i)[2] = c$, indicando que a inserção foi realizada entre o segundo e o quinto salto. O RemapRoute remapeia então o terceiro salto, onde $C(t_{i-1})[3] = d$ e $C(t_i)[3] = x$. Como $x \notin C(t_{i-1})$ o RemapRoute passa para a fase de remapeamento local da mudança, onde remapeia o quarto salto e termina.

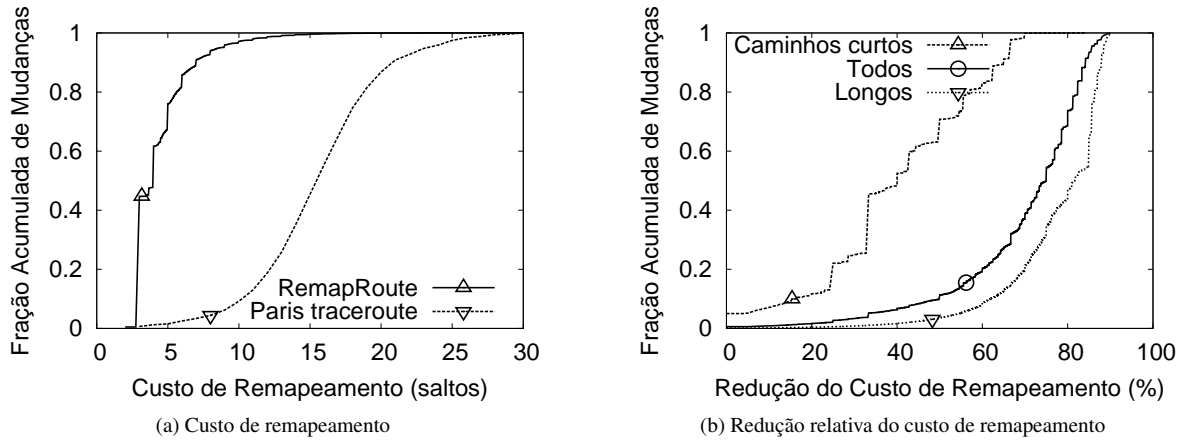


Figura 5: Comparação do custo de remapeamento entre RemapRoute e Paris traceroute.

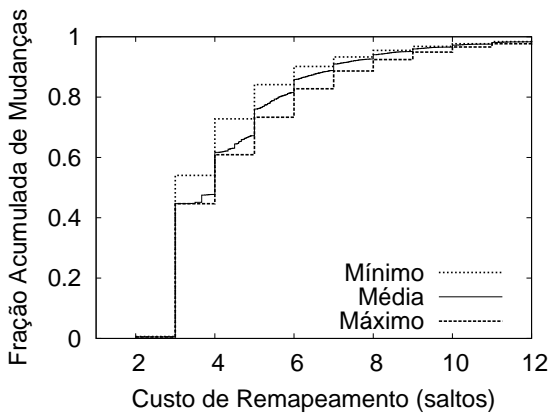


Figura 4: Custo de remapeamento com RemapRoute variando o salto s' de detecção da mudança.

5. AVALIAÇÃO VIA SIMULAÇÃO COM DADOS REAIS

Nesta seção avaliamos o RemapRoute usando simulações dirigidas com dados reais. Utilizamos o mesmo conjunto de dados descrito na seção 3. Nosso foco é comparar o custo de remapear mudanças de caminho usando o Paris traceroute com o custo de remapear usando o RemapRoute.

O custo de remapeamento usando o RemapRoute varia de acordo com o salto s' onde a mudança é detectada. Calculamos o custo de remapeamento do RemapRoute para todos os saltos do caminho onde a mudança pode ser detectada. A figura 4 mostra a distribuição do custo de remapeamento de mudanças para o RemapRoute. Mostramos a distribuição dos custos mínimo, médio e máximo, calculados sobre todos os saltos envolvidos em uma mudança. Vemos que o custo mínimo e máximo em

geral são muito próximos. Uma razão para este comportamento é que várias mudanças de caminho envolvem poucos roteadores, logo não existe muita variação em função do salto s' onde a mudança é detectada. No resto deste artigo usaremos o custo médio como representativo do custo de remapeamento com RemapRoute.

A figura 5a compara o custo de remapeamento do RemapRoute com o custo de remapeamento do Paris traceroute. Comparando com a figura 1a vemos que o RemapRoute frequentemente precisa sondar um número de saltos maior do que o número de saltos envolvidos numa mudança. Este aumento deve-se a mudanças de roteamento que precisam ser localizadas usando a técnica de pesquisa binária antes de serem remapeadas. Independente do processo de localização, o RemapRoute tem custo de remapeamento significativamente menor que o do Paris traceroute. Note que o custo de remapeamento do RemapRoute raramente é menor do que três saltos, mesmo com 9% das mudanças envolvendo apenas dois saltos (figura 1a). As mudanças que envolvem apenas dois roteadores apenas removem saltos da rota antiga, e precisamos localizar o salto onde a remoção aconteceu usando busca binária. A busca binária requer pelo menos três sondas a não ser que a mudança seja detectada nos três primeiros saltos do caminho, o que acontece em 0,4% das mudanças em nossos dados.

A figura 5b mostra a distribuição da redução no custo de remapeamento quando usamos RemapRoute em vez do Paris traceroute. Calculamos a redução do custo de remapeamento como $(C_{\text{Paris}} - C_{\text{RemapRoute}}) / C_{\text{Paris}}$, onde C_{Paris} é o número de saltos remapeados pelo Paris traceroute e $C_{\text{RemapRoute}}$ é o número de saltos remapeados pelo RemapRoute. A linha sólida, calculada para todas as mudanças de caminho em nossos dados, mostra que a redução no custo é significativa. O RemapRoute reduz pra menos da metade o custo de remapeamento de 88%

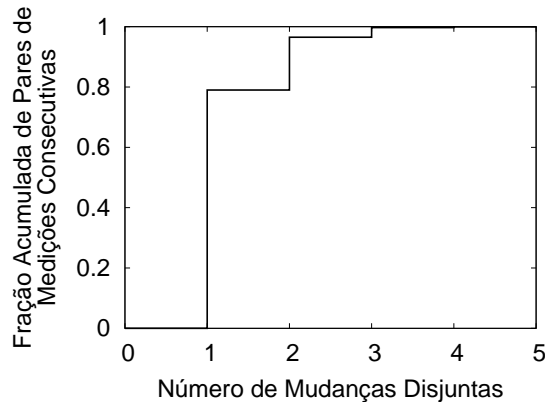


Figura 6: Distribuição do número de mudanças disjuntas entre duas medições com Paris traceroute.

das mudanças de caminho. As linhas tracejadas na figura 5b mostram a redução no custo para mudanças em rotas menores que 10 saltos (curtas) e maiores que 20 saltos (longas). Vemos que a redução no custo é mais acentuada para rotas longas, onde o Paris traceroute desperdiça sondas em vários roteadores que não estão envolvidos na mudança, e que o RemapRoute traz redução de custos para remapeamento mesmo em rotas curtas.

5.1. ERROS DE REMAPEAMENTO

O RemapRoute remapeia mudanças num caminho dado um salto s' onde uma mudança foi detectada. Isso pode levar a inconsistências caso um caminho sofra duas ou mais mudanças disjuntas antes de detectarmos a primeira mudança. Por exemplo, um caminho $\{a, b, c, d, e, f, g\}$ pode mudar para $\{a, x, c, d, y, f, g\}$ entre duas medições consecutivas com Paris traceroute. Neste caso, podemos detectar duas mudanças nos saltos $s' = 1$ e $s'' = 4$. Infelizmente, dado s' ou s'' , o RemapRoute remapeará apenas uma mudança.

Para avaliar a gravidade desse problema, a figura 6 mostra a distribuição do número de mudanças disjuntas para os pares de medições consecutivas dos caminhos em nosso conjunto de dados.² Vemos que 79% das medições consecutivas remapeiam apenas uma mudança disjunta, que o RemapRoute remapeará corretamente para qualquer salto s' onde a mudança for detectada. Apenas 3% das medições consecutivas remapeiam três ou mais mudanças disjuntas, indicando que o RemapRoute remapeará a nova rota corretamente na maioria dos casos.

Três outros fatores contribuem para minimizar o impacto de mudanças disjuntas. Primeiro, o RemapRoute pode remapear mudanças disjuntas anteriores ao salto

de detecção s' caso ele as detecte durante o processo de remapeamento. Em particular, a probabilidade do RemapRoute remapear uma mudança disjunta anterior ao salto de detecção s' no nosso conjunto de dados é 43%. Segundo, uma mudança disjunta que não for remapeada quando executarmos RemapRoute será detectada e remapeada no futuro (assumindo que o processo de sondagem para detecção de mudanças de caminho seja capaz de detectar todas as mudanças possíveis). Qualquer mudança disjunta que não for remapeada causa apenas uma inconsistência temporária nos dados. Terceiro, a redução do custo de remapeamento obtida com RemapRoute pode ser utilizada para aumentar a frequência de sondagem para detecção de mudanças e reduzir a chance de ocorrerem duas mudanças em um caminho antes de detectarmos a primeira.

Outra limitação do RemapRoute é que o mecanismo de busca binária pode falhar quando a ordem relativa de dois saltos se inverte da rota antiga para a nova rota. Um exemplo extremo, mas ilustrativo, é uma mudança de $C(t_{i-1}) = \{a, b, c, d, e, f\}$ para $C(t) = \{a, e, d, c, b, f\}$. Apenas 0,9% das mudanças de caminho em nossos dados invertem a ordem relativa de saltos. Como inversão da ordem relativa de saltos é um evento raro, tomamos a decisão conservadora de remapear o caminho por inteiro (como o Paris traceroute) quando uma inversão é detectada durante o processo de remapeamento. Como mostram os resultados anteriores, essa limitação não compromete a utilidade do RemapRoute.

6. AVALIAÇÃO COM PROTÓTIPO REAL NO PLANETLAB

Nesta seção avaliamos um protótipo do RemapRoute no PlanetLab. Implantamos o Paris traceroute e o RemapRoute em 140 nós PlanetLab e coletamos medições por 18 horas de 30 de Novembro de 2012. Cada nó executa o Paris traceroute para medir caminhos periodicamente. Como no conjunto de dados utilizado nas seções anteriores, cada nó monitora caminhos para 1.000 destinos escolhidos aleatoriamente de uma lista com 34.820 destinos alcançáveis na Internet. Quando duas medições consecutivas com Paris traceroute detectam uma mudança, executamos o RemapRoute para remapeá-la. Cada nó leva em média 7 horas e 42 minutos para medir os 1.000 caminhos. Devido à baixa frequência de sondagem, este conjunto de dados contém apenas 87.848 mudanças. Os caminhos medidos atravessam 7.143 sistemas autônomos e 95% dos sistemas autônomos de grande porte na Internet [19].

A figura 7 mostra a redução do custo de remapeamento quando usamos o RemapRoute em vez do Paris traceroute. Comparando com a figura 5b, a redução média

²No nosso conjunto de dados só medimos caminhos com Paris traceroute quando uma mudança é detectada. Todos os pares de medições consecutivas remapeiam pelo menos uma mudança disjunta.

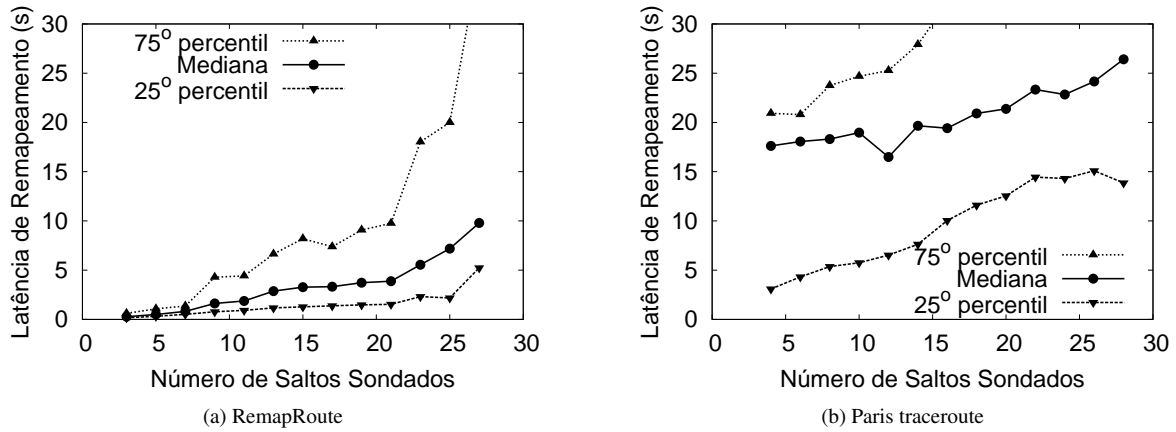


Figura 8: Latência de remapeamento em cenários reais.

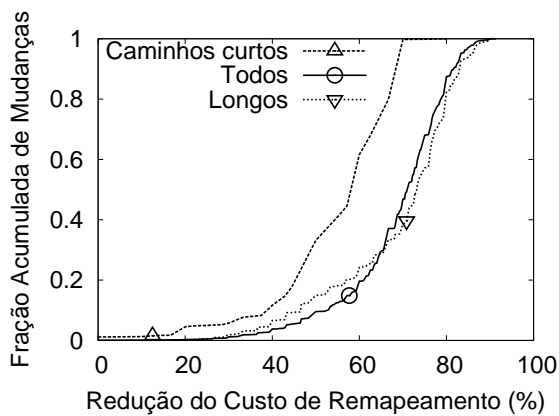


Figura 7: Redução do custo de remapeamento com RemapRoute em cenários reais.

do custo de remapeamento no cenário real é quantitativamente similar aos resultados obtidos via simulação (linha sólida). Por exemplo, reduzimos pra menos da metade o custo de remapeamento de 90% das mudanças de caminho no cenário real.

A redução de custo para rotas curtas e longas é mais similar à redução de custo geral no cenário real que nas simulações. Em outras palavras, as linhas tracejadas na figura 7 estão mais próximas da linha sólida que na figura 5b. Atribuímos essa mudança a três fatores: (i) o menor número de mudanças observadas no cenário real pode limitar a variedade de mudanças observadas; (ii) diferenças no conjunto de caminhos monitorados; e (iii) a diferente forma de detecção de mudanças (os dados utilizados nas simulações foram coletados com o DTRACK).

A figura 8a mostra os 25º, 50º e o 75º percentis da latência de remapeamento em função do número de saltos sondados durante o processo de remapeamento. Avali-

amos a latência de remapeamento por que o RemapRoute sonda saltos sequencialmente: a decisão do próximo salto a sondar depende do resultado do último salto sondado. O Paris traceroute, em contrapartida, poderia paralelizar a sondagem de saltos (apesar da implementação padrão não fazê-lo). Como a maior parte dos remapeamentos com RemapRoute requer sondagem de poucos saltos (figura 5a), a latência de remapeamento geralmente é menor que 5 segundos. A figura 8b mostra os 25º, 50º e 75º percentis da latência de remapeamento com Paris traceroute no cenário real. Nosso objetivo não é comparar a latência de remapeamento do RemapRoute com Paris traceroute, pois a latência é diretamente afetada por decisões de implementação da ferramenta. Nosso objetivo é mostrar que a latência de remapeamento com RemapRoute é aceitável para uso em sistemas reais. Notamos ainda que um sistema de mapeamento topológico como o DTRACK pode executar o RemapRoute simultaneamente em caminhos diferentes caso mais de uma mudança seja detectada num curto intervalo de tempo.

Por último, avaliamos se o remapeamento com o RemapRoute é equivalente a utilizar o Paris traceroute para medir o novo caminho por inteiro. Para cada mudança observada no cenário real, comparamos os saltos remapeados pelo RemapRoute com a rota medida pelo Paris traceroute. Apenas 0,6% das medições com RemapRoute são diferentes das medições com Paris traceroute. A identificação de roteadores que fazem balanceamento de carga usando Paris traceroute ou RemapRoute é probabilística [25]. Por exemplo, a configuração padrão do Paris traceroute e do RemapRoute identifica todos os roteadores que fazem balanceamento de carga numa rota com 95% de confiabilidade. Erros de medição são inevitáveis e causam diferenças entre remapeamentos independente da ferramenta utilizada. Outra causa para diferença nos remapeamentos são mudanças de roteamento que aconte-

cem no intervalo entre a medição com Paris traceroute e a medição com RemapRoute.

Sumarizando, o remapeamento de mudanças de caminho na Internet com RemapRoute é tão preciso quanto remapeamento com Paris traceroute, reduz significativamente o número de sondas enviadas e tem pouco impacto na latência de remapeamento.

7. TRABALHOS RELACIONADOS

Operadores podem utilizar mensagens dos protocolos de roteamento (e.g., OSPF e IS-IS) e arquivos de configuração dos roteadores para mapear a topologia de sua rede [13, 18, 24]. Esta abordagem resulta em mapas completos e precisos da topologia, mas só está disponível para operadores de redes e é restrita a uma única rede. Para mapear múltiplas redes, podemos utilizar coletores públicos de mensagens BGP³ para construir um mapa dos sistemas autônomos da Internet [8, 19]. Infelizmente, o BGP não expõe todos os enlaces da rede e coletores públicos de mensagens BGP não cobrem todos os sistemas autônomos da Internet [5, 19]. Neste trabalho tomamos a abordagem ortogonal de medir a topologia da rede no nível de roteadores usando medições ativas.

Pesquisa sobre mapeamento topológico no nível de roteadores usando medições ativas têm três objetivos principais: (i) aumentar a cobertura da Internet, (ii) aumentar a precisão da topologia construída e (iii) aumentar a frequência das medições.

A abordagem clássica para aumentar a cobertura da Internet é monitorar um grande número de caminhos. A plataforma Skitter/Ark da CAIDA [4] tenta cobrir toda a Internet usando alguns monitores para medir caminhos para todos os prefixos /24 anunciados na Internet. O problema é que o Skitter/Ark demora de dois a três dias para coletar a topologia devido ao grande número de caminhos monitorados e limitações de banda nos monitores. Uma alternativa é dividir a carga de sondagem das medições entre vários monitores, como nos sistemas DIMES [21] e Ono [3]. Neste trabalho assumimos que o conjunto de monitores e destinos é fixo. Porém, utilização do RemapRoute é ortogonal ao conjunto de monitores e destinos. O RemapRoute pode ser usado por qualquer um dos sistemas acima para reduzir a sobrecarga de rede.

Técnicas para aumentar a precisão da topologia coletada tentam inferir mais informações sobre a rede do que medições tradicionais com traceroute. O Paris traceroute envia sondas adicionais variando sistematicamente os valores nos campos do cabeçalho IP para detectar todos os roteadores que fazem balanceamento de carga em

uma caminho [1, 25]. O RemapRoute detecta roteadores que fazem balanceamento de carga utilizando o mesmo algoritmo que o Paris traceroute. O DisCarte usa traceroute e sondas com a opção *Record Route* do protocolo IP ativada para coletar duas sequências relacionadas de roteadores em caminhos da Internet [22]. O DisCarte pós-processa essas sequências com ferramentas de aprendizado de máquina para combiná-las em uma topologia mais precisa. Estas e outras técnicas enviam sondas adicionais e aumentam o custo de mapeamento da topologia. O objetivo do RemapRoute é complementar: reduzir o custo do remapeamento de mudanças de roteamento, aumentando a disponibilidade de sondas para coleta de topologias mais precisas.

Nosso trabalho é mais relacionado com técnicas para aumentar a frequência de medições da topologia da Internet. Em geral, monitores têm banda de rede limitada para mapear a topologia. Reduzir o custo de cada medição da topologia aumenta diretamente a frequência com a qual medições podem ser coletadas. O RocketFuel, por exemplo, reduz o custo para mapear a topologia de um sistema autônomo alvo descartando caminhos que têm roteadores de ingresso e egresso no sistema autônomo alvo idênticos a outro caminho já medido [23]. Outra abordagem é reduzir o custo de medições escolhendo apenas um destino em cada sub-rede num sistema autônomo [2]. O Doubletree reduz sondas redundantes nos roteadores próximos a monitores (compartilhados pelos caminhos partindo do monitor) e nos roteadores próximos a destinos (compartilhados pelos caminhos que terminam no destino) [11]. O RemapRoute foi desenvolvido para reduzir o custo do remapeamento de mudanças no DTRACK, nosso sistema de mapeamento topológico [7]. O RemapRoute e o DTRACK complementam as técnicas existentes para redução do custo de mapeamento topológico. O DTRACK reduz sondas redundantes para roteadores próximos aos monitores como o Doubletree e é compatível com técnicas descritas acima para selecionar quais caminhos mapear.

8. CONCLUSÕES E TRABALHOS FUTUROS

A manutenção de mapas completos e atualizados da Internet é difícil devido à grande quantidade de sondas necessárias e restrições de banda nos monitores. Neste trabalho propomos o RemapRoute, uma ferramenta para reduzir o custo de remapeamento de mudanças de roteamento na Internet. Dadas a rota anterior a uma mudança de roteamento e um salto (*hop*) onde a mudança foi detectada, o RemapRoute (1) realiza uma busca binária pelo salto onde a mudança aconteceu e (2) faz o remapeamento local da mudança. Comparado com a abordagem atual de remapear a nova rota por inteiro usando traceroute, o

³The University of Oregon Routeviews Project, <http://www.routeviews.org>
RIPE Routing Information Service, <http://www.ripe.net/data-tools/stats/ris>

RemapRoute reduz significativamente o número de saltos sondados para remapear mudanças de roteamento na Internet. Essa redução de saltos sondados aumenta a disponibilidade de sondas, potencializando a medição de mais caminhos ou aumento da frequência de medições. O remapeamento com RemapRoute é tão preciso quanto remapear o caminho inteiro com traceroute, e a latência de remapeamento é satisfatória para utilização do RemapRoute em sistemas reais. RemapRoute é mais um passo na construção de mapas da topologia da Internet mais completos e atualizados.

Como trabalho futuro, pretendemos integrar o RemapRoute no DTRACK e prover um serviço de mapeamento da Internet aberto para disponibilizar informações sobre a topologia para pesquisadores e aplicações. Queremos também reduzir ainda mais o custo de remapeamento de mudanças no DTRACK. Atualmente, o DTRACK remapeia cada caminho separadamente. Esta abordagem desperdiça sondas caso vários caminhos sejam afetados pela mesma mudança. Pretendemos desenvolver mecanismos para prever quais caminhos são afetados por uma mesma mudança e remapear apenas um deles.

Referências

- [1] B. Augustin, T. Friedman, and R. Teixeira. Measuring Load-balanced Paths in the Internet. In *Proc. IMC*, 2007.
- [2] R. Beverly, A. Berger, and G. Xie. Primitives for Active Internet Topology Mapping: Toward High-Frequency Characterization. In *Proc. IMC*, 2010.
- [3] D. R. Choffnes, F. E. Bustamante, and Z. Ge. Crowdsourcing Service-level Network Event Monitoring. In *Proc. ACM SIGCOMM*, 2010.
- [4] K. Claffy, Y. Hyun, K. Keys, M. Fomenkov, and D. Krioukov. Internet Mapping: from Art to Science. In *Proc. IEEE CATCH*, 2009.
- [5] R. Cohen and D. Raz. The Internet Dark Matter - on the Missing Links in the AS Connectivity Map. In *Proc. IEEE INFOCOM*, 2006.
- [6] Í. Cunha, R. Teixeira, and C. Diot. Measuring and Characterizing End-to-End Route Dynamics in the Presence of Load Balancing. In *Proc. PAM*, 2011.
- [7] Í. Cunha, R. Teixeira, D. Veitch, and C. Diot. Predicting and Tracking Internet Path Changes. In *Proc. ACM SIGCOMM*, 2011.
- [8] A. Dhamdhere and C. Dovrolis. Ten Years in the Evolution of the Internet Ecosystem. In *Proc. IMC*, 2008.
- [9] A. Dhamdhere, R. Teixeira, C. Dovrolis, and C. Diot. NetDiagnoser: Troubleshooting Network Unreachabilities Usind End-to-end Probes and Routing Data. In *Proc. ACM CoNEXT*, 2007.
- [10] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitarman, and B. Wehl. Globally Distributed Content Delivery. *IEEE Internet Computing*, 6(5):50–58, 2002.
- [11] B. Donnet, P. Raoult, T. Friedman, and M. Crovella. Efficient Algorithms for Large-scale Topology Discovery. In *Proc. ACM SIGMETRICS*, 2005.
- [12] N. Duffield. Network Tomography of Binary Network Performance Characteristics. *IEEE Trans. on Inf. Theory*, 52(12):5373–5388, 2006.
- [13] N. Feamster and H. Balakrishnan. Detecting BGP Configuration Faults with Static Analysis. In *Proc. USENIX NSDI*, 2005.
- [14] V. Jacobson. traceroute, Feb 1989. Available at <ftp://ftp.ee.lbl.gov/traceroute.tar.gz>.
- [15] E. Katz-Bassett, C. Scott, D. R. Choffnes, I. Cunha, V. Valancius, N. Feamster, H. V. Madhyastha, T. Anderson, and A. Krishnamurthy. LIFEGUARD: Practical Repair of Persistent Route Failures. In *Proc. ACM SIGCOMM*, 2012.
- [16] R. Kompella, J. Yates, A. Greenberg, and A. Snoeren. Detection and Localization of Network Blackholes. In *Proc. IEEE INFOCOM*, 2007.
- [17] H. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iPlane: an Information Plane for Distributed Services. In *Proc. USENIX OSDI*, 2006.
- [18] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. N. Chuah, Y. Ganjali, and C. Diot. Characterization of Failures in an Operational IP Backbone Network. *IEEE/ACM Trans. Netw.*, 16(4):749–762, 2008.
- [19] R. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang. Quantifying the Completeness of the Observed Internet AS-level Structure. *IEEE/ACM Trans. Netw.*, 18(1):109–122, 2010.
- [20] V. Paxson. End-to-end Routing Behavior in the Internet. *IEEE/ACM Trans. Netw.*, 5(5):601–615, 1997.
- [21] Y. Shavitt and U. Weinsberg. Quantifying the Importance of Vantage Points Distribution in Internet Topology Measurements. In *Proc. IEEE INFOCOM*, 2009.

- [22] R. Sherwood, A. Bender, and N. Spring. DisCarte: a Disjunctive Internet Cartographer. In *Proc. ACM SIGCOMM*, 2008.
- [23] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP Topologies with Rocketfuel. In *Proc. ACM SIGCOMM*, 2002.
- [24] D. Turner, K. Levchenko, A. Snoeren, and S. Savage. California Fault Lines: Understanding the Causes and Impact of Network Failures. In *Proc. ACM SIGCOMM*, 2010.
- [25] D. Veitch, B. Augustin, T. Friedman, and R. Teixeira. Failure Control in Multipath Route Tracing. In *Proc. IEEE INFOCOM*, 2009.

Análise e Geração de Vizinhança em Redes de Contato Intermitente *

Tiphaine Phe-Neau¹, Miguel Elias M. Campista², Marcelo Dias de Amorim¹ & Vania Conan³

¹CNRS e UPMC Sorbonne Universités
4 place Jussieu, 75005 – Paris, França
{tiphaine.phe-neau | marcelo.amorim }@lip6.fr

²Universidade Federal do Rio de Janeiro
GTA/PEE-COPPE/DEL-Poli
Av. Brig. Trompowski s/n, P.O.Box 68504 – Rio de Janeiro, Brasil
miguel@gta.ufrj.br

³Thales Communications & Security
4 Avenue des Louvresses, 92230 – Gennevilliers, França
vania.conan@thalesgroup.com

Resumo

A modelagem dinâmica de redes oportunísticas se baseia no conhecimento dual dos contatos e dos intercontatos. Este trabalho propõe o uso de uma visão estendida, na qual os nós rastreiam sua vizinhança estendida (a alguns saltos) e não somente seus vizinhos diretos. Para tal, é introduzido um método que permite aos nós preverem se outros estarão em alcance dada a posição atual e os movimentos anteriores. Essa abordagem é contrária às existentes nas quais os padrões de contato são extraídos da mobilidade espacial dos nós. O método proposto é aplicado a vários traços reais e sintéticos. Inicialmente, um novo algoritmo é provido bem como uma modelagem intuitiva para compreender o entorno de um nó. Em seguida, são destacadas duas cadeias principais de comportamento da vizinhança. Finalmente, três principais tipos de movimento (nascimento, morte e sequencial) são identificados assim como os seus padrões predominantes. Toda essa análise culmina no desenvolvimento de um gerador de vizinhança capaz de gerar traços de contato entre pares de nós. Tal gerador pode criar traços preservando características reais para simulações em diferentes escalas de tempo.

Palavras-chave: Redes tolerantes a atrasos e desconexões, DTN, Redes oportunísticas, vizinhança

Abstract

Modeling the dynamics of opportunistic networks relies on the dual notion of contacts and intercontacts. We propose the use of an extended view in which nodes track their extended vicinity (up to a few hops) and not only their direct neighbors. We introduce a method that allows nodes predicting whether other nodes will be within reach given their current position and previous movements. This approach is contrary to existing ones where contact patterns are derived from the spatial mobility of nodes. We apply our method to several real-world and synthetic datasets. Firstly, we provide a novel algorithm and an intuitive modeling to understand a node's surroundings. Then, we highlight two main behaviors of vicinity chains. Finally, three main types of movements (birth, death, and sequential) are identified as well as their predominant patterns. These whole analysis culminates in the development of a neighborhood generator capable of generate intercontact traces between pairs of nodes. Such generator can create traces preserving real characteristics for simulations in different time scales.

Keywords: Delay Tolerant Networks, DTN, Opportunistic networks, neighborhood

*Tiphaine Phe-Neau e Marcelo Dias de Amorim realizaram parte deste trabalho no LINC3 (<http://www.lincs.fr>). Este trabalho foi parcialmente financiado pelo Programa FP7 da Comunidade Europeia sob os recursos do acordo no. FP7-317959 MOTO. Miguel Elias M. Campista também gostaria de agradecer o suporte financeiro do CNPq, da Faperj, da CAPES e da FINEP.

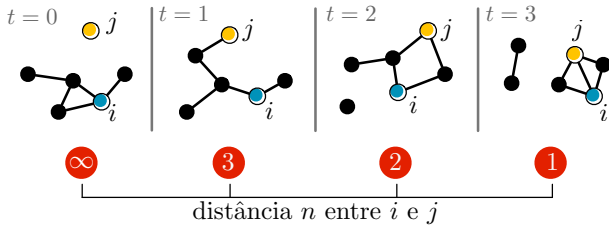


Figura 1. Em $t = 0$, o nó j está fora da vizinhança de i , embora esteja se aproximando. Em $t = 1$, j surge na vizinhança de i com uma distância de três saltos. Em $t = 2$, j se aproxima ainda mais, chegando a uma distância de dois saltos. O nó j entra em contato com i em $t = 3$.

1. INTRODUÇÃO

A compreensão dos padrões de mobilidade no contexto das redes móveis com conexão intermitente é fundamental para o desenvolvimento de protocolos e algoritmos de rede eficientes [8]. A literatura nessa área tem gerado um número significativo de contribuições que oferecem respostas a questões relacionadas à frequência com que os nós se encontram e como isso ocorre [3, 5, 11]. Uma característica comum nesses trabalhos é que todos se baseiam no conhecimento dos *contatos* e dos *intercontatos*. Um contato ocorre quando dois nós estão em alcance mútuo enquanto um intercontato ocorre quando os dois nós *não estão* em contato.

Este trabalho defende que os nós devem considerar uma visão estendida de suas vizinhanças ao incluir nós que não estão em contato, mas que mesmo assim podem ser considerados próximos. Tais nós podem ser alcançados em até κ saltos, sendo o conjunto desses nós referido como κ -vizinhança [15]. Acredita-se que nenhum trabalho anterior tenha investigado a evolução no tempo da vizinhança estendida de um nó. A Figura 1 ilustra tal evolução em uma pequena rede. Note que a definição tradicional de contato e intercontato consideraria os primeiros três primeiros intervalos de tempo como iguais para i e j , já que os nós estão em intercontato. Ao contrário, neste trabalho, distingue-se as quatro situações e o impacto dessa definição é investigado através das seguintes questões: Dado que os dois nós i e j estejam separados por n saltos, qual a probabilidade deles estarem separados por m saltos ($m \neq n$) assim que a distância mudar? Ou ainda, é possível identificar padrões nessa dinâmica tal que a movimentação possa ser antecipada?

A movimentação da vizinhança é modelada como um processo Markoviano de tempo contínuo calculado a partir de uma estrutura de dados contendo a evolução da vizinhança de um nó, chamada de “linha de tempo”. A ideia é rastrear a vizinhança de um nó até uma distância κ . O limiar κ é ajustado de acordo com a extensão necessária do monitoramento e pela quantidade de sobrecarga tolerada, uma vez que o monitoramento da vizinhança pode gerar

sobrecarga de controle. Ao final, a rede pode se basear na movimentação da vizinhança para realizar estimativas de parâmetros fundamentais como o atraso de entrega e as relações sociais entre os nós. O trabalho conhecido mais próximo a este é o CTG (*Connectivity Trace Generator*) que propõe um modelo de conectividade para redes oportunísticas [1]. Entretanto, o CTG também foca na mera noção de contatos e intercontatos, negligenciando os eventos no entorno dos nós.

Este trabalho apresenta as seguintes contribuições, das quais as três primeiras apareceram em [12], confirmadas a partir de diferentes análises usando traços reais e sintéticos:

- **Um modelo para compreensão do comportamento da vizinhança móvel.** É definida a vizinhança de um nó com a noção de κ -vizinhança assim como é proposto um arcabouço para analisá-la conforme a movimentação. É ainda proposto um fluxo de trabalho (*workflow*) correspondente para geração de cadeias de vizinhança que capturam a evolução estatística da distância entre os nós.
- **Dois tipos de cadeias de vizinhança.** São identificadas duas cadeias principais, curta e estendida, que diferem em função dos estados atravessados. As cadeias estendidas representam uma vizinhança com muitos saltos em potencial dentro do limiar κ , enquanto as curtas possuem vizinhanças de até dois saltos (conforme os resultados obtidos).
- **Três tendências principais de movimento.** Nas cadeias estendidas, são identificados três tipos predominantes de movimentos. Os movimentos de nascimento, de morte e de sequência podem representar 87% de todos os movimentos de um dado traço. Logo, considerar que um desses movimentos é a provável próxima decisão de um nó na vizinhança torna-se razoável.
- **Um gerador genérico de vizinhança.** Baseado nos tipos predominantes de movimentos (nascimento, morte e sequência) e nas probabilidades de transição entre esses tipos de movimentos, é proposto um gerador de vizinhanças entre pares de nós da rede que pode ser utilizado em diferentes escalas de tempo sem perder as características fundamentais de contato.

Acredita-se que a compreensão da movimentação da vizinhança possa ajudar no entendimento do entorno de um nó em uma rede oportunística. Além disso, acredita-se que a metodologia proposta neste trabalho seja um primeiro esforço direcionado ao desenvolvimento de geradores de conectividade. Tais geradores podem ser usados

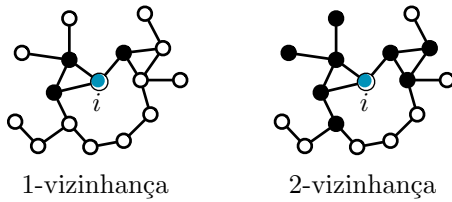


Figura 2. Na esquerda: 1-vizinhança do nó i representa o conjunto de vizinhos a um salto de distância de i . Na direita: 2-vizinhança do nó i representa o conjunto de nós a dois saltos de distância de i . As distâncias mais curtas não são discriminadas em uma dada κ -vizinhança.

na criação de traços realistas com um número qualquer de nós.

Este artigo está organizado da seguinte forma. A Seção 2 introduz as informações necessárias para a definição de vizinhança. A Seção 3 propõe o fluxo de trabalho para a geração de movimentação da vizinhança. Já as Seções 4, 5 e 6 apresentam os resultados obtidos da análise dos padrões de vizinhança. Finalmente, a Seção 8 conclui este trabalho e apresenta as direções futuras.

2. DEFINIÇÕES

Antes de continuar com a análise da movimentação da vizinhança, é importante introduzir formalmente as definições utilizadas.

2.1. κ -VIZINHANÇA

O conceito de κ -vizinhança, definido em um artigo precedente [15], é fundamental já que ele define a extensão na qual a análise se aplica. Logo, discrimina-se a vizinhança de i conforme o número de saltos entre i e os seus vizinhos.

Definição 1. κ -vizinhança. A κ -vizinhança \mathcal{V}_{κ}^i de um nó i é o conjunto de todos os nós cujo o caminho mais curto de i é de no máximo κ saltos.

A partir da Definição 1, conclui-se que $\mathcal{V}_{\kappa-1}^A \subseteq \mathcal{V}_{\kappa}^A$. A Figura 2 ilustra a 1-vizinhança e a 2-vizinhança do nó i . Neste trabalho, as análises propostas focam em movimentos na κ -vizinhança de um dado nó.

2.2. INTERCONTATO FAVORÁVEL E DESCONEXÃO

Como dito anteriormente, nas redes móveis com conexão intermitente, esforços para caracterizar a dinâmica da rede estão relacionados à noção dual de *contato* e *intercontato*. Um exemplo é ilustrado na Figura 3(a). Este trabalho distingue os nós que não estão em contato direto, mas que possuem um caminho que os conecta, dos

nós que não possuem nenhuma possibilidade de comunicação [13]. Um par de nós está em intercontato favorável com parâmetro n sempre que há um caminho entre eles com distância mais curta de n saltos. Vale mencionar que dois nós a um salto estão em contato. Formalmente, define-se intercontatos favoráveis e desconexão como se segue:

Definição 2. Intercontato favorável. Um intercontato é considerado “favorável” com parâmetro n quando há um caminho mais curto de comprimento n separando os dois nós em questão, tal que $\{n \in \mathbb{N}^* \mid 2 \leq n < \infty\}$.

Definição 3. Desconexão. Ao contrário das situações favoráveis, a “desconexão” indica a falta de caminhos entre um par de nós. Logo, $n \rightarrow \infty$.

Essa visão, que chamamos de “estendida”, é ilustrada na Figura 3(b). Os movimentos partindo de contato ou de qualquer um dos n intercontatos favoráveis para qualquer outro estado possível são investigados neste trabalho. Essa análise ajuda a compreender o comportamento da κ -vizinhança de um dado nó.

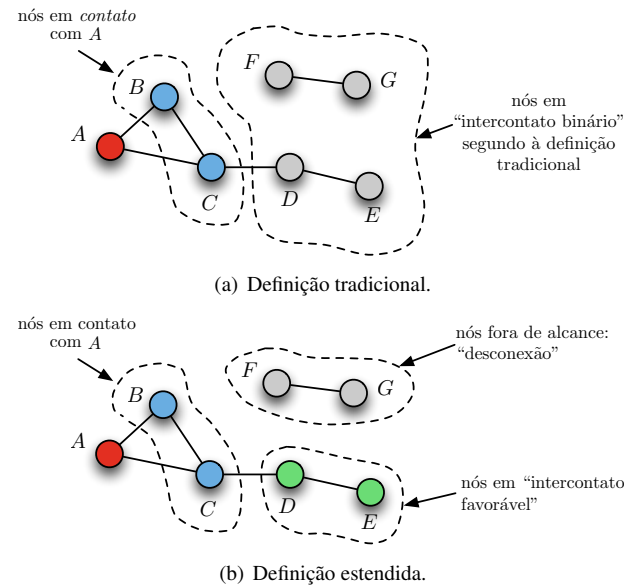


Figura 3. Definição de vizinhança segundo o método tradicional e a proposta.

3. MOVIMENTAÇÃO DA VIZINHANÇA: METODOLOGIA

A movimentação da vizinhança considera todos os movimentos dentro da κ -vizinhança de um nó. Para isso, são oferecidos elementos que respondam questões como “quando a distância n entre os nós i e j muda, qual a

probabilidade da distância se tornar m , onde $m \neq n$?. A resposta é obtida a partir de uma metodologia em duas etapas:

1. **Geração da linha de tempo.** Calcula-se a *linha de tempo* da vizinhança, que é a progressão da menor distância entre quaisquer dois nós ao longo do tempo. Através do uso dessas linhas é possível realizar diferentes análises probabilísticas.
2. **Análise da vizinhança.** A linha de tempo provê a informação necessária para caracterizar a probabilidade de transição entre distâncias quaisquer.

3.1. GERAÇÃO DA LINHA DE TEMPO

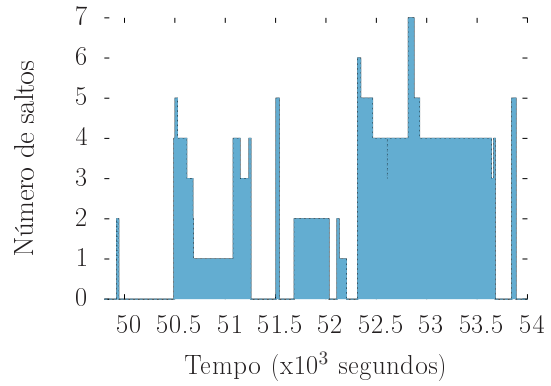
O método proposto usa os traços de contato como entrada, organizados de forma cronológica de eventos instantâneos. Os eventos podem ser tanto o surgimento quanto o desaparecimento de um enlace entre um par de nós (i, j) no instante t . Denota-se esse tipo de evento como $e = \langle t, i, j, \text{UP/DOWN} \rangle$, onde UP e DOWN indicam, respectivamente, o surgimento e o desaparecimento de um enlace entre i e j .

Para um dado par de nós (i, j) , uma linha de tempo consiste em uma sequência de distâncias mais curtas entre eles ao longo do tempo (Figura 4(a)). Formalmente, a linha de tempo é representada como uma sequência de tuplas $\langle n, i, j, t_{\text{início}}, t_{\text{fim}} \rangle$. Isso significa que entre $t_{\text{início}}$ e t_{fim} , os nós i e j estiveram a uma distância de n saltos.

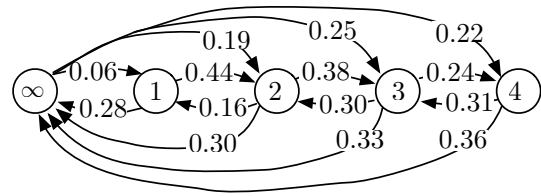
A geração de linhas de tempo requer como entrada o traço de contatos (\mathcal{C}) e o número de nós na rede (\mathcal{N}), conforme apresentado no Algoritmo 1. Todas as linhas de tempo são inicializadas com a tupla $\langle \infty, 0 \rangle$, indicando que os nós estão desconectados no momento 0. Todas as tuplas seguintes indicam uma mudança no estado e o momento na qual ela ocorreu. Todos os eventos nos traços são lidos e a matriz de adjacências atualizada antes de computar todas as distâncias mais curtas entre os pares de nós. Os dados obtidos são formatados e impressos nas linhas de tempo.

3.2. ANÁLISE DA VIZINHANÇA

A movimentação da vizinhança é modelada através de um processo Markoviano de tempo contínuo (*Continuous Time Markov Process – CTMP*) para cada par de nós. Para um dado nó i , considere que $X_{i,j}^s$ seja a variável aleatória representando a distância entre os nós i e j no passo s . O CTMP é usado ao invés de um processo Markoviano simples para permitir a observação da evolução dos movimentos independentemente da amostragem de tempo. A etapa da análise da vizinhança recebe como entrada as linhas de tempo e oferece como saída as probabilidades de transição correspondentes das cadeias de vizinhança.



(a)



(b)

Figura 4. Nesta figura, são apresentadas a linha de tempo de um dos pares do traço *Unimi* (Figura 4(a)) e a movimentação média da vizinhança de um par (i, j) com $\kappa = 4$ no traço *Infocom05* (Figura 4(b)). Por simplicidade, poucas transições são apresentadas (até 4). Nessa κ -vizinhança, a probabilidade de um nó entrar em contato $\{\infty \rightarrow 1\}$ é de 6%. Já considerando os nós a uma distância de três saltos, a probabilidade para que eles estejam a dois saltos é de 30%.

Estados. Os estados do CTMP dependem da escolha de κ , isto é, do tamanho da vizinhança que se queira monitorar. O número de estados é $\kappa + 1$; o primeiro estado, representado por ' ∞ ', corresponde ao caso onde os dois nós estão desconectados. Já o estado $\{1\}$ representa o contato e os estados remanescentes $\{2, \dots, M = \kappa\}$ correspondem a uma situação de intercontato favorável. Note que é considerado cada movimento de um par de nós como um único passo. Não são considerados intervalos de tempo específicos para evitar a dependência do traço com o tempo. Assume-se que X satisfaz as propriedades da cadeia de Markov e que $X_{i,j}^s$ é independente de $X_{i,j}^{s-1}$. Essa premissa pode parecer questionável, mas a natureza dos traços leva à independência dos movimentos dos pares, como será visto na Seção 6.3.

Probabilidades de transição. A compreensão da movimentação da vizinhança requer foco nas taxas de transição CTMP entre estados, isto é, na probabilidade de dois nós estarem a uma distância m no passo s sabendo que eles estavam a uma distância n no passo precedente: $\mathbb{P}(X_{i,j}^s = m \mid X_{i,j}^{s-1} = n)$, $m \neq n$. Para a caracteri-

Algoritmo 1: Geração da linha de tempo (LT).

Requer: \mathcal{C}, \mathcal{N} // traço de contatos, número de nós
Garante: $\mathcal{N} \times (\mathcal{N} - 1)$ linha de tempo (LT)
Local: $\{adj\}$ // matriz de adjacências de tamanho \mathcal{N}^2

```

1 inicialização; // todas as linhas de tempo são inicializadas com  $\langle \infty, 0 \rangle$ 
2 while size of  $\mathcal{C} \neq 0$  do
3    $t_{atual}, i, j, evento =$  remoção da primeira tupla de  $\mathcal{C}$ ;
4   if  $evento = UP$  then
5      $adj_{i,j} = 1$ ;
6      $adj_{j,i} = 1$ ;
7   else if  $evento = DOWN$  then
8      $adj_{i,j} = 0$ ;
9      $adj_{j,i} = 0$ ;
10  for  $i \leftarrow 1$  to  $\mathcal{N}$  do
11    for  $j \leftarrow 1$  to  $\mathcal{N}$  do
12      if  $i \neq j$  then
13         $d_{atual} =$  caminho mais curto( $i, j$ );
14        if comprimento de  $LT_{(i,j)} = 1$  then
15          anexa ( $d_{atual}, t_{atual}$ ) to  $LT_{(i,j)}$ ;
16        else
17          ( $d_{último}, t_{último}$ ) = pegar última tupla de  $LT_{(i,j)}$ ;
18          if  $d_{último} \neq d_{atual}$  then
19            anexa ( $d_{atual}, t_{último}$ ) to  $TL_{(i,j)}$ ;
20  for  $i \leftarrow 1$  to  $\mathcal{N}$  do
21    for  $j \leftarrow 1$  to  $\mathcal{N}$  do
22      if  $i \neq j$  then
23        formata e imprime  $LT_{(i,j)}$ ;
```

zação completa do CMTP, são usadas probabilidades de transição entre estados e o tempo médio de permanência em cada um deles. Os tempos de permanência média em cada estado são dados na Tabela 1.

A Figura 4(b) mostra um exemplo da probabilidade de transição média do traço *Infocom05*, que será descrito em maiores detalhes na Seção 3.3. Algumas transições estão omitidas para aumentar a clareza da figura. Como pode ser observado, quando os nós i e j estão desconectados (∞), a probabilidade de que eles se encontrem diretamente é de 6%, enquanto a probabilidade de um intercontato favorável de três saltos é de 25%. O fluxo de trabalho proposto da movimentação da vizinhança pode ser visto na Figura 5.

3.3. TRAÇOS

A análise da proposta é baseada na movimentação da vizinhança obtida em traços de experimentos reais e sintéticos como descritos a seguir. Os parâmetros usados foram escolhidos para representar situações particulares.

- **Infocom05.** Baseado em medidas conduzidas em uma conferência de cinco dias em 2005 [2], onde 41 participantes carregaram iMotes para coletar informações sobre outros em um raio de 10m. O estudo deste trabalho foca em um trecho de 12 horas com maior atividade da rede. Cada iMote envia sondas a cada 120 segundos. Esse traço representa um encontro profissional.
- **Rollernet.** Conta com 62 participantes medindo a conectividade mútua entre iMotes enquanto patinavam pelas ruas de Paris em um encontro de 3 horas [17]. Os pesquisadores ajustaram os dispositivos para enviar sondas a cada 30 segundos. Esse traço representa um cenário esportivo.
- **Unimi.** É um traço capturado por estudantes, professores e funcionários da Universidade de Milão em 2008 [4]. Foram envolvidos 48 pessoas com dispositivos especiais que enviavam uma sonda por segundo para a vizinhança. Esse traço representa um cenário

Tabela 1. Tempo médio de permanência em cada estado em segundos.

Traço	Estado								
	∞	1	2	3	4	5	6	7	≥ 8
Infocom05	2.029	399	296	224	175	131	154	212	229
Rollernet	167	51	74	86	102	117	127	142	166
Sassy	157.504	2.315	53.871	1	–	–	–	–	–
StanfordHigh	2.972	1	1	0	–	–	–	–	–
Unimi	18.041	1.300	447	305	214	155	208	74	35
Community	5.210	108	120	114	118	326	1.330	295	15
RT	203	221	117	82	61	47	40	34	35

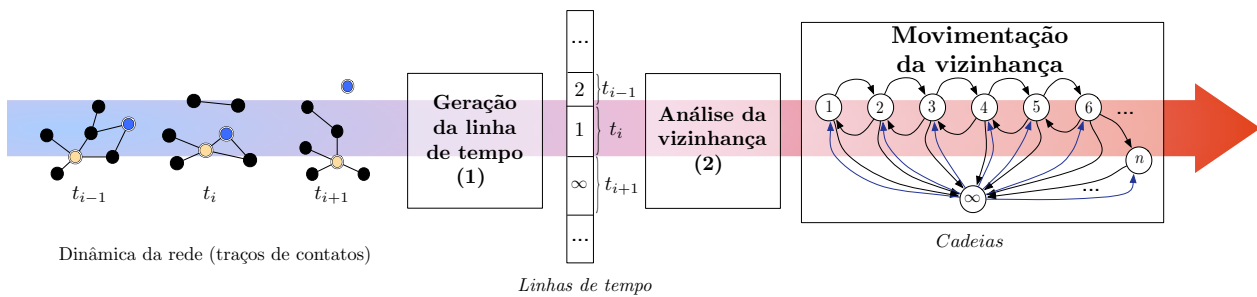


Figura 5. O fluxo de trabalho para a geração da movimentação da vizinhança. Inicia-se através da leitura dos traços de contato que descrevem a conectividade da rede ao longo do tempo. Essa entrada é processada usando o módulo de geração de linhas de tempo (1). As linhas de tempo são seqüências de distâncias mais curtas entre todos os pares de nós. A etapa (2), chamada de análise da vizinhança, examina essas seqüências para calcular as probabilidades de transição e as cadeias correspondentes da movimentação da vizinhança.

escolar e de trabalho.

- **StanfordHigh.** Conta com 789 pessoas em uma escola nos EUA carregando motes TelosB para detectar contatos até 3m de alcance [16]. Os motes que enviavam sondas a cada 20 segundos foram fornecidos aos estudantes, professores e funcionários durante um dia inteiro. Este trabalho usa um subconjunto de 200 participantes por questões de limitação na geração da linha de tempo. O traço *StanfordHigh* possui uma configuração com maioria de adolescentes com tendência de relacionamento em grupos de interesse.
- **Sassy.** Gerado na Universidade de Saint Andrews por pesquisadores que usaram 27 T-motes para capturar contatos entre os alunos e cientistas [10]. Os T-motes enviavam sondas a cada 6,67 segundos durante 79 dias. O traço *Sassy* possui uma configuração acadêmica esparsa com longa duração.
- **RT.** É um modelo de mobilidade que corrige falhas do modelo *Random Waypoint* [9]. Foram amostrados o comportamento de vinte nós seguindo esse modelo em uma superfície de $50 \times 60 \text{m}^2$ utilizando 10m de alcance com velocidades entre 0 e 7m/s.

- **Community.** É um modelo de mobilidade baseado em comportamento social [7]. Ele coloca nós com relações sociais em uma posição específica ao mesmo tempo, como grupos de amigos. São simulados 50 nós com um raio de 10m em um plano de $1.500 \times 2.500 \text{m}^2$ durante 9 horas.

4. CENÁRIO vs. κ

Um requisito importante antes de abordar a movimentação da vizinhança é entender as características gerais da rede para saber qual o valor mais apropriado de κ . Neste trabalho, não se pretende ajustar um valor específico para cada traço, mas apenas estimar um valor genérico que permita alcançar conclusões não equivocadas. Na prática, deseja-se encontrar um κ específico que sirva bem no cenário e que tenha uma boa relação entre sobrecarga e conhecimento da vizinhança.

A primeira análise tem como objetivo encontrar um valor de κ que seja suficientemente grande para cobrir todos os componentes conectados dos nós. Para cada par de nós, é computado a distância max-min do traço, que é diâmetro do componente conectado. No traço *Infocom05*, observa-se que a maior proporção de pares (aproximadamente 75%) tem entre 7 e 9 saltos de distância no má-

ximo, enquanto apenas 13% têm entre 4 e 6 saltos, e 9% têm entre 10 e 12 saltos. Poucos pares têm um caminho entre 1 e 3 saltos de distância. O traço *Unimi* apresenta a maior proporção de nós com caminhos com distância máxima entre 7 e 9 saltos.

Esses resultados confirmam que é importante discriminar o comportamento dos nós, mesmo considerando um alto nível de movimentação de sua vizinhança. O restante deste trabalho foca em cadeias de vizinhança com estados até $\{7, 8, 9\}$, já que elas representam a maior parte das situações observadas.

5. CADEIAS DE VIZINHANÇA

5.1. TEMPO MÉDIO DE PERMANÊNCIA EM CADA ESTADO

A Tabela 1 apresenta a duração média de permanência no estado κ em segundos. Nos traços *RT* e *Unimi*, é observada uma redução gradual nas durações. Por outro lado, o traço *Rollernet* tem uma tendência de crescimento, enquanto o *Infocom05* e o *Community* têm comportamentos combinados. O status específico do *Rollernet* como um esporte dinâmico pode explicar o aumento dos valores. Pequenas distâncias têm uma pequena duração por causa da conectividade variável e da dinâmica da configuração. Já as distâncias mais longas são absorvidas pela multidão (note que não são discriminadas as mudanças de caminhos se elas forem do mesmo comprimento).

5.2. DISTRIBUIÇÕES ESTACIONÁRIAS

Observa-se em todos os traços que a cadeia de Markov embutida (*embedded Markov chain* - EMC) é irredutível. Logo, uma distribuição estacionária existe em todos os casos. A Tabela 2 apresenta as distribuições estacionárias quando $\kappa \geq 8$. No traço *Infocom05*, há 25,3% de chance do nó procurado não pertencer à κ -vizinhança em questão; 5,5% de chance do nó estar em contato; 15,4% dele estar a dois saltos; 20% a três saltos e assim por diante. Note que com a observação da vizinhança para $\kappa = 4$, tem-se 77% de chances de encontrar o nó procurado. Tal conhecimento posterior é útil para avaliar a probabilidade de encontrar um nó rapidamente na chegada ou até mesmo quantificar os limiares de sondagem para manter os custos de manutenção reduzidos.

5.3. CADEIAS CURTAS E ESTENDIDAS

Existem dois tipos de cadeias observadas: as estendidas que podem variar até estados com dez ou doze saltos e as curtas com movimentos de apenas um ou dois saltos.

Cadeias curtas. As cadeias curtas reforçam a premissa anterior na qual os nós ou estão em contato ou desconectados; a diferença nesse caso é que elas podem ter até dois saltos de distância. Nota-se tal configuração para dois

dos traços usados: *Sassy* e *StanfordHigh*. A cadeia observada consiste em estados $\{\infty, 1, 2\}$. Como resultado, tal configuração não tira nenhum ou quase nenhum proveito dos intercontatos favoráveis. Na maior parte do tempo, quando se detecta um nó, seu próximo movimento é quase sempre de desaparecimento da vizinhança. Protocolos oportunistas devem também considerar esses comportamentos quando necessário.

Cadeias estendidas. Os traços *Infocom05*, *Community*, *RT*, *Rollernet* e *Unimi* mostram cadeias estendidas de vizinhança. As cadeias estendidas possuem maior potencial em mudanças de estados. Alguns com até doze ou mais saltos. As cadeias estendidas permitem estados de interconexão favorável e, portanto, maiores possibilidade de transmissões fim-a-fim. As cadeias estendidas podem também exibir uma larga quantidade de movimentos internos. Será visto que há três tipos de movimentos que dominam as tendências. Com apenas poucos padrões de movimento, será demonstrado que é possível abstrair muitos dos movimentos dos nós.

6. PADRÕES EM CADEIAS ESTENDIDAS

Os traços que contêm mais cadeias estendidas oferecem mais possibilidades de transições para o próximo salto. Como consequência, por economia de espaço, não serão apresentados de agora em diante os resultados obtidos com os traços *Sassy* e *StanfordHigh*. Nos traços analisados, são observados três tipos principais de transições, chamados de *nascimento*, de *morte*, e de *movimentos sequenciais*.

6.1. NASCIMENTO NA κ -VIZINHANÇA

O fenômeno de nascimento é caracterizado pelo surgimento de um nó na κ -vizinhança após um período de desconexão. O conhecimento desse fenômeno permite que um nó ou um protocolo saiba em qual distância outro nó pode aparecer. Considere que o nó i queira enviar uma mensagem ao nó j no traço *Infocom05*. Se j estiver atualmente fora da κ -vizinhança do nó i , ele não precisará confiar em um encaminhamento totalmente oportunístico. Dados os valores estacionários calculados na Figura 4(b), sabe-se que j irá surgir com uma probabilidade de 25% a uma distância de três saltos.

Na Tabela 3, são apresentados os valores relacionados ao evento nascimento nos traços analisados. A linha “estado” descreve o estado com a maior probabilidade de nascimento, enquanto a linha “probabilidade” indica a probabilidade de nascimento desse estado em particular. A linha “acumulado” representa a probabilidade de nascimento acumulada em qualquer distância dentro da κ -vizinhança para $\kappa =$ “estado” (isto é, para qualquer valor entre 1 e “estado”). Nota-se que o estado com maior pro-

Tabela 2. Distribuição estacionária em porcentagem.

Traço	Estado								
	∞	1	2	3	4	5	6	7	≥ 8
Infocom05	25,3	5,5	15,4	20,0	16,0	9,7	5,1	2,2	0,8
Rollernet	28,2	2,3	7,7	11,5	12,5	11,5	9,5	7,3	9,5
Sassy	49,2	34,8	15,5	0,5	0,0	0,0	0,0	0,0	0,0
StanfordHigh	45,0	48,0	6,9	0,1	0,0	0,0	0,0	0,0	0,0
Unimi	35,0	9,0	14,0	15,0	12,0	8,0	4,0	2,0	1,0
Community	24,6	9,7	26,3	26,1	10,1	2,5	0,4	0,0	0,3
RT	29,1	5,0	10,6	14,1	14,3	11,5	7,7	4,5	3,2

Tabela 3. Valores de nascimento.

	Traço				
	Infocom05	Unimi	Community	RT	Rollernet
Estado	3	2	2	4	4
Probabilidade	0,25	0,22	0,37	0,19	0,15
Acumulado	0,50	0,40	0,52	0,59	0,44

babilidade de nascimento pode cobrir de 40 até 59% das chegadas (somente uma distância de dois ou três saltos nos traços *Unimi*, *Community* e *Infocom05*). Para todos os traços, a maior probabilidade de nascimento está no conjunto $\{1, 2, 3, 4\}$. Se for escolhido estender os limites para envio de sondas somente até o estado 4, a probabilidade acumulada se torna 50 até 70%, dependendo dos traços. Logo, o envio de sondas até 4-vizinhança é suficiente para alcançar a maioria dos padrões de nascimento no entorno de um nó [14].

6.2. MORTE NA κ -VIZINHANÇA

Em oposição ao nascimento para padrões de chegada, define-se a morte como o fenômeno de saída dos nós da κ -vizinhança. Os traços são analisados em dois diferentes aspectos: a proporção de mortes relacionadas à cadeia inteira (absoluta) e a comparação com somente movimentos naturais (o que exclui transições entre estados não consecutivos exceto para ∞).

Na Figura 6(a), a evolução das probabilidades é mostrada para os diferentes estados da cadeia. Todos os traços, exceto o *Community*, têm taxas de morte aproximadamente estáveis, cuja máxima variação absoluta é 12%. Na Figura 6(b), são mostrados os resultados no caso dos movimentos naturais. Observa-se um fenômeno interessante: todos os traços possuem a mesma evolução na taxa de morte relativa. Relacionada a movimentos naturais, a proporção de movimentos de morte tem um padrão similar (decréscimo leve seguido de acréscimo leve). A principal diferença vem a ser os valores iniciais no eixo y para cada traço.

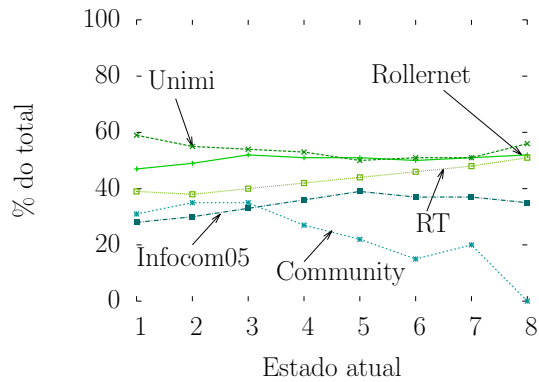
Os eventos de nascimento e morte e os sequenciais,

apresentados a seguir, representam uma grande parte dos movimentos identificados.

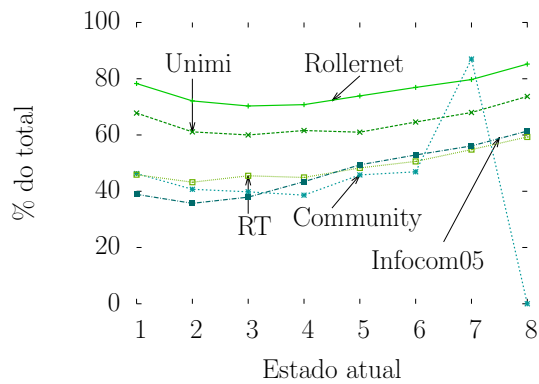
6.3. MOVIMENTOS SEQUENCIAIS

Define-se como um movimento sequencial para dois nós, o processo de se aproximar ou se distanciar um do outro em estados adjacentes da cadeia: quando os nós (i, j) estão a uma distância de n saltos (para $n > 1$), eles se movem sequencialmente para mais próximo se estiverem a exatamente $n - 1$ saltos no passo seguinte. Em oposição, eles sequencialmente se distanciarão se eles estivessem a $n + 1$ saltos.

Observa-se que uma parte não desprezível de movimentos da vizinhança surge de comportamentos sequenciais. Nos traços *Unimi* e *Infocom05*, considerando que os nós permaneçam na κ -vizinhança, os movimentos sequenciais representam entre 50 e 80% dos movimentos. Outra observação é que quanto maior a distância do par de nós, maior é a proporção de movimentos erráticos (movimentos que não são nem nascimento, nem morte e nem sequenciais). Entretanto, movimentos sequenciais continuam a predominar. Na Figura 7, é apresentada a proporção de movimentos sequenciais, nascimento e erráticos dentre todos os movimentos da vizinhança. Os movimentos erráticos aumentam com a distância entre os nós enquanto os de morte continuam estacionários em torno de 30%. A presença desse tipo de movimento está de acordo com a premissa de independência de movimentos dos pares. Já os movimentos sequenciais estão concentrados na 4-vizinhança. O efeito dos padrões sequenciais influencia menos em distâncias maiores devido às perturbações do ambiente.



(a) Cadeias completas.



(b) Movimentos naturais.

Figura 6. Proporção de mortes.

A análise em maiores detalhes dos movimentos sequenciais permite ainda dividi-los em mais duas subclasses: movimentos *incrementais* (*inc*) e *decrementais* (*dec*). Enquanto os movimentos *inc* consistem em movimentos onde as distâncias aumentam para o estado imediatamente mais alto, os *dec* consistem no oposto, ou seja, as distâncias diminuem para o estado imediatamente mais baixo. Similarmente ao realizado para as taxas de morte, investiga-se os movimentos *inc* e *dec* usando escalas diferentes: absoluta, relativa para movimentos naturais e relativamente à proporção de movimentos sequenciais. Sobre os movimentos *dec*, todos os traços que mostram cadeias estendidas (*Infocom05*, *Rollernet*, *Unimi* e *RT*) apresentam um aumento lento até a distância de dois saltos, seguida por um suave decréscimo (Figura 8(a)). A proporção de movimentos sequenciais é então simples de prever. A mesma dedução pode ser feita em geral sobre a proporção de movimentos sequenciais relacionados aos movimentos naturais (Figura 8(b)).

Aumentando a granularidade da observação sobre os movimentos sequenciais, é possível entender quais os pa-

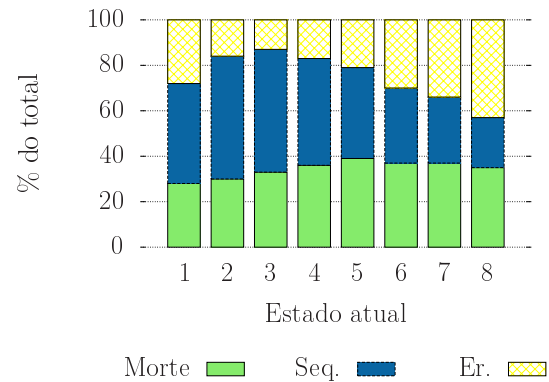


Figura 7. A movimentação da vizinhança (morte, sequencial e errático) para o traço *Infocom05*.

drões de movimento são predominantes: aproximação (*inc*) ou afastamento (*dec*)? Na Figura 9, as proporções *dec* e *inc* são apresentadas. Os valores absolutos de *dec* e *inc* (Figuras 9(a) e 9(d)) mostram que o padrão de aproximação tem uma distribuição estacionária em todos os traços, exceto o *Community*. O *dec* não varia muito em torno do seu valor inicial enquanto o padrão de distanciamento (*inc*) rapidamente diminui nos estados mais altos. Isso pode ser explicado pela conectividade variável em distâncias mais longas que resultam em morte ao invés de movimentos de distanciamento.

Considerando os movimentos naturais e sequenciais, a proporção de movimentos *dec* e *inc* entre os movimentos naturais é diferente em termos de valores, mas possui a mesma evolução que as absolutas. O *dec* mantém valores quase estacionários enquanto o *inc* mostra rápida redução nas Figuras 9(b) e 9(e), respectivamente. A comparação dos movimentos *dec* e *inc*, vistos nas Figuras 9(c) e 9(f), com os movimentos sequenciais mostra padrões claros. Para todos os traços, exceto o *Community*, a proporção de *dec* possui um crescimento aproximadamente linear enquanto o *inc* decresce linearmente. Como o *dec* e o *inc* representam uma partição completa dos movimentos sequenciais, as observações parecem lógicas. O crescimento dos movimentos *dec* tem origem no declínio do *inc* no caso natural, atribuindo uma maior proporção de *dec* no caso sequencial.

6.4. TRAÇO *Community*: CASO EXCEPCIONAL

O traço *Community* não se encaixa nas principais observações. Isso pode ser uma consequência direta da natureza do processo de geração, já que o objetivo principal do modelo de mobilidade é gerar padrões de movimento onde grupos de pessoas estejam no mesmo lugar ao mesmo tempo. Apesar de ser um esforço válido e resultar em padrões plausíveis, ele gera movimentos não naturais. O traço *Community* é mantido mesmo assim para mostrar

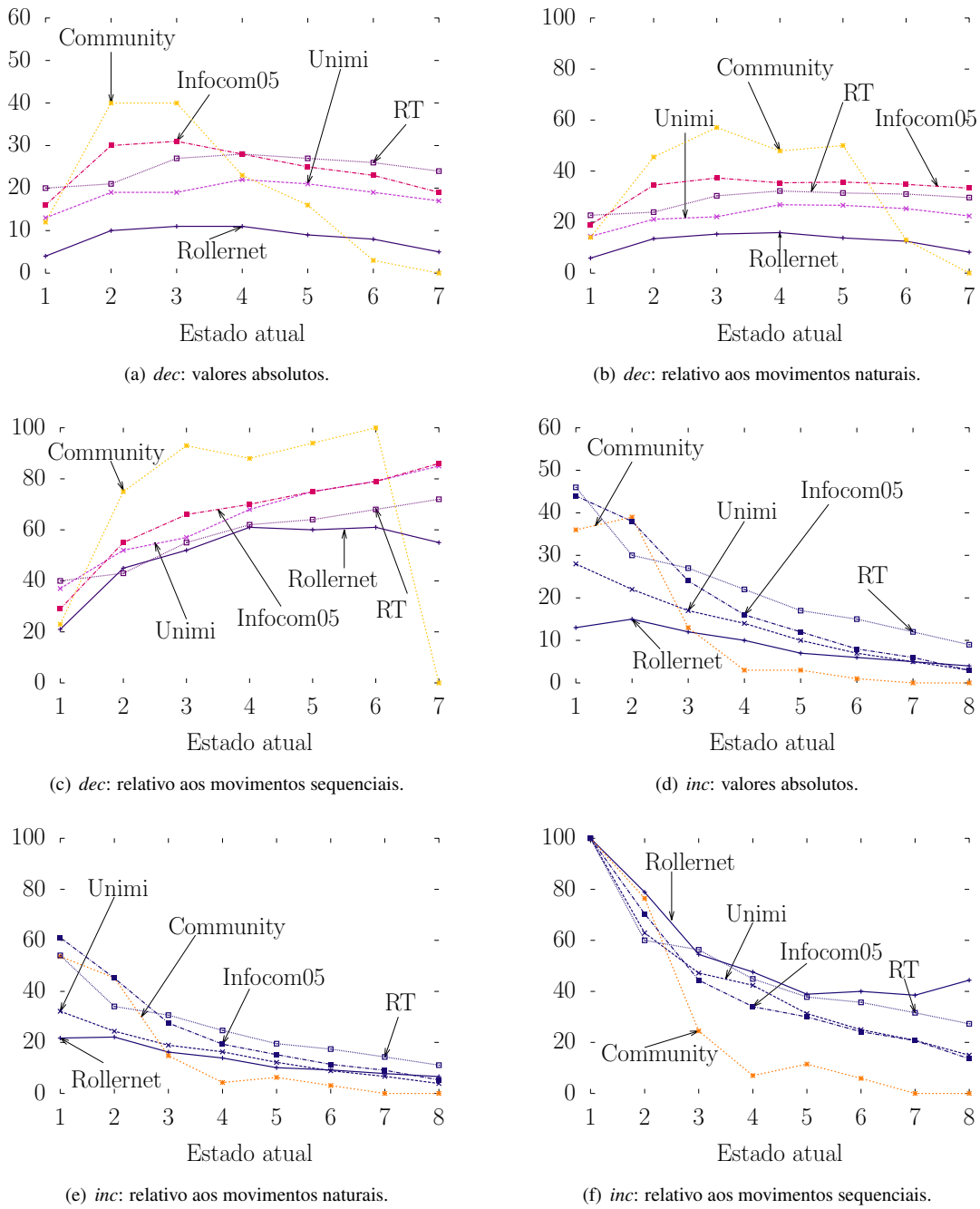


Figura 9. Proporções médias de movimentos *inc* e *dec*. Os movimentos *inc* indicam movimentos de distanciamento incremental e os *dec* indicam o oposto. O eixo-*x* apresenta os estados atuais dos nós envolvidos e o eixo-*y* a porcentagem de movimentos da vizinhança.

o seu comportamento original.

7. GERADOR DE VIZINHANÇA

A aplicação direta da movimentação da vizinhança é a possibilidade de gerar linhas de tempo sintéticas. Esta

seção apresenta o gerador de vizinhança proposto, chamado de Tiger (TImeline GENERATOR). O Tiger se baseia nas saídas do módulo de movimentação da vizinhança (linhas de tempo extraídas e probabilidade de transição). Entretanto, o gerador pode também ser usado de maneira isolada, caso as probabilidades de transição e a duração

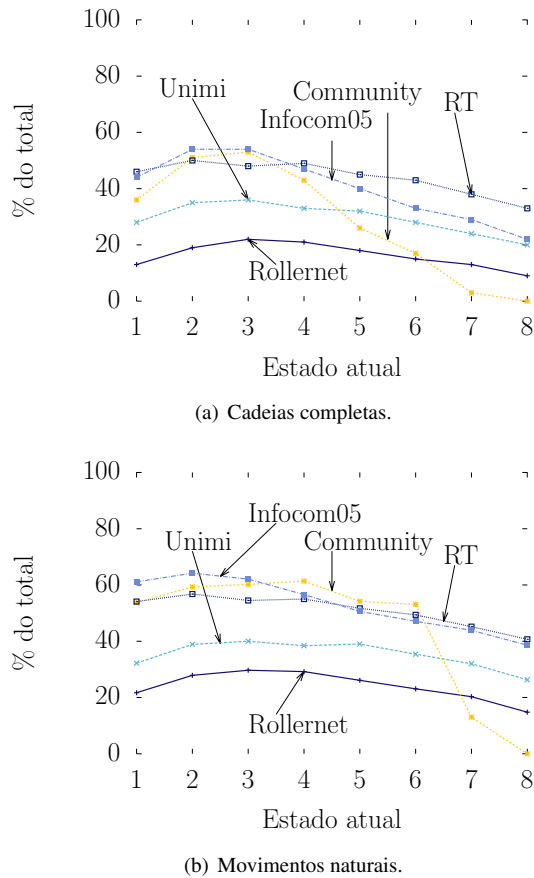


Figura 8. Proporção de movimentos sequenciais.

média dos κ -intervalos forem fornecidos. O Tiger foi implementado como uma aplicação em Python 2.7.

7.1. MOTIVAÇÃO

O uso de linhas de tempo para conhecer a vizinhaça dos nós em redes oportunísticas é original. Antes, protocolos como o BubbleRap usavam períodos de contato entre pares de nós para prever os encontros futuros [6]. Neste trabalho, ao invés de focar em contatos apenas, o conhecimento é estendido para a κ -vizinhaça de um nó. O uso da vizinhaça permite maior conhecimento da rede e, portanto, multiplica as possibilidades de encontros com outros nós. Como os padrões de contato entre nós podem ser extraídos das linhas de tempo, a geração dessas linhas pode ser útil inclusive para testes de protocolos oportunísticos. A partir das linhas de tempo, pode-se prever, por exemplo, a distância que um dado nó vai aparecer de outro.

7.2. PROCESSO DE GERAÇÃO

Para gerar um comportamento entre pares de nós, o Tiger se baseia nas probabilidades de transição para um

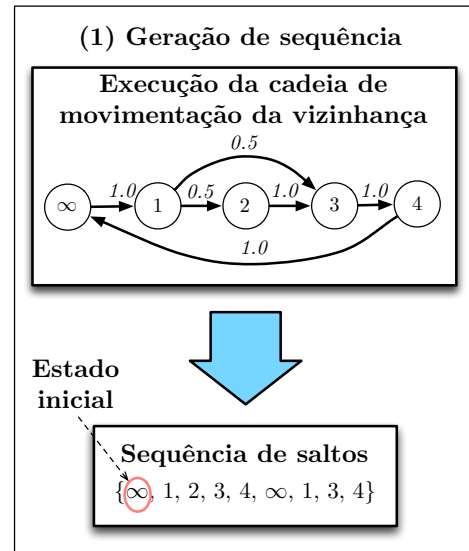


Figura 10. Exemplo de geração de sequência de saltos obtida com o Tiger. De uma dada probabilidade de transição de movimentação de vizinhaça, o Tiger produz uma possível sequência de saltos, ou sequência de estados, s . A sequência s tem probabilidade de transição similar à probabilidade de transição inicial de movimentação de vizinhaça.

dado valor de κ e para uma dada distribuição de durações de contatos. Baseado nas probabilidades de transição da movimentação da vizinhaça, uma sequência de distâncias mais curtas entre pares de nós é gerada. Um primeiro passo é sempre gerar uma sequência de distâncias que corresponda à probabilidade de transição da movimentação da vizinhaça dada. O desafio é então encontrar as correspondências dessas distâncias geradas com os intervalos de duração corretos e com um número plausível de intervalos de distâncias. Para cada uma das distâncias máximas obtidas, são observados os seguintes passos:

7.2.1. Geração de sequência de saltos: Esse passo gera uma sequência compatível de saltos (uma lista de distâncias ao qual a transição da movimentação da vizinhaça é similar à transição oferecida) de uma dada movimentação de vizinhaça. Uma distância máxima D é assumida e as transições de movimentação de vizinhaça são processadas como se segue:

- **Estado inicial.** É necessário iniciar o gerador de linhas de tempo com uma primeira distância (um estado inicial para a movimentação da vizinhaça oferecida). Foi escolhido começar com uma distância aleatória denotada por d_0 entre todos os estados existentes $\{\infty, 1, \dots, D\}$. Por exemplo, $d_0 = \infty$.
- **Execução da cadeia de movimentação da vizinhaça.** A cadeia correspondente de movimentação

ção da vizinhança é então executada do estado inicial $d_0 = \infty$. É escolhida a probabilidade de saída mais alta de ∞ e é decrementada as taxas de transição assumidas de um dado valor Δ . No Tiger, o Δ é ajustado para ser o maior fator comum entre todas as taxas de transição. Ao recair em um nó sorvedouro, no qual todas as transições de saída são nulas, outro estado de saída é aleatoriamente escolhido. A geração de distância é parada quando todas as taxas de transição são esgotadas.

Os mesmos passos são repetidos para todos os valores máximos de distância em $[1:D]$. Considerando a distribuição de distância máxima, pode-se gerar várias linhas de tempo sintéticas de acordo com a movimentação da vizinhança limitada a essa distância máxima. A única precaução a ser tomada é o de normalizar as probabilidades de transição de movimentação da vizinhança correspondentes, antes de executar a cadeia de movimentação da vizinhança normalizada.

O exemplo da Figura 10 é mostrado em detalhes. Para uma distância max-min igual a 4, assume-se as seguintes taxas de transição: $\{(\infty \rightarrow 1 = 1, 0), (1 \rightarrow 2 = 1, 0), (2 \rightarrow 3 = 0, 5), (2 \rightarrow 4 = 0, 5), (3 \rightarrow 4 = 1, 0) \text{ e } (4 \rightarrow \infty = 1, 0)\}$. Todas as outras probabilidades de transição são consideradas nulas. Inicialmente, $d_0 = \infty$. Assim, o $\Delta = 0,5$ é determinado (porque ele é o maior fator comum entre $\{1,0, 0,5\}$). Da movimentação da vizinhança da Figura 10, é obtida a transição $\infty \rightarrow 1$. A movimentação da vizinhança resultante é a mesma que antes, exceto pelo valor da transição $\infty \rightarrow 1$, que é agora $1,0-0,5 = 0,5$. Esse valor é normalizado pela probabilidade total de saída e $\infty \rightarrow 1$ se torna 1,0. Até agora, o estado é o 1, onde se pode decidir em ir ou para o estado 2 ou 3 porque eles têm a mesma probabilidade 0,5. O estado 2 é aleatoriamente escolhido e o $1 \rightarrow 2$ é decrementado para 0,0 e normalizado $1 \rightarrow 3$ para 1,0. Logo, após o estado 2, tem-se o estado 3 e assim por diante, até que todas as probabilidade de transição sejam $\leq 0,0$. Na proposta deste trabalho, a sequência de saltos s correspondente seria $s = \{\infty, 1, 2, 3, 4, \infty, 1, 3, 4\}$.

Agora que se tem s , é necessário encontrar a correspondência entre essa sequência de estados/distâncias e intervalos de duração acurados.

7.2.2. Correspondência do Tempo: Usando a sequência de saltos s , encontra-se a correspondência de todas as distâncias com um intervalo plausível de duração. Dependendo na necessidade do usuário, o Tiger oferece dois modos. O primeiro, Modo I, produz linhas de tempo com intervalos de duração como os da vida real; enquanto o segundo, Modo II, gera linhas de tempo com mais transições compatíveis com a movimentação da vizinhança. Esse passo requer que

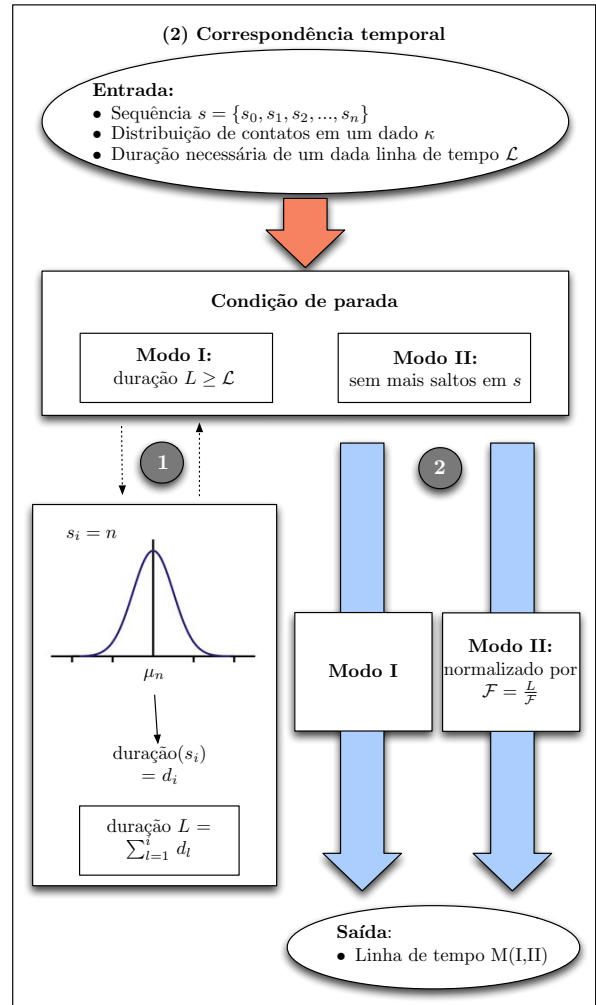


Figura 11. Processo de correspondência de tempo do Tiger.

o usuário entre com o tamanho da linha de tempo \mathcal{L} desejado. Chama-se a linha de tempo gerada pelo Modo I de MI-linha-de-tempo e as geradas pelo Modo II de MII-linhas-de-tempo. O funcionamento de ambos os modos é detalhado a seguir.

- **O Modo I reflete intervalos de duração plausíveis.** A primeira opção disponível significa refletir mais acuradamente as durações de um κ -intervalo. Para cada distância de s , é usada a distribuição de durações de contato para um dado κ obtido a partir do módulo de movimentação da vizinhança. Assumindo que $s = n$, então uma distribuição Gaussiana é usada baseada na distribuição de durações de contatos n (duração média, primeiro e terceiro quartil) para extrair um valor plausível de intervalo. Logo, é gravada a soma das durações obtidas até que a duração total dos intervalos exceda \mathcal{L} . As linhas de

tempo MI podem perder algum passo de s para respeitar a duração necessária \mathcal{L} e as durações de intervalos plausíveis.

• **O Modo II foca nas probabilidades de transição.**

Na segunda opção disponível, o foco é dado nas probabilidades de transição de movimentação da vizinhança. O processo do Modo I é mantido sem a limitação da correspondência do tempo em relação à duração necessária. Manteve-se a geração de durações plausíveis de κ -intervalos durante todo o passo s . Logo, até o final da sequência foi usado um fator de adaptação (*fitting*) \mathcal{F} , onde L é a duração total da sequência s e \mathcal{F} é o tamanho necessário. Assim,

$$\mathcal{F} = \frac{\mathcal{L}}{L}.$$

Se $\mathcal{F} < 1$, significa que a duração da sequência gerada é maior que a duração necessária e que toda a duração da sequência gerada é multiplicada por \mathcal{F} . Caso contrário, se $\mathcal{F} > 1$, significa que a duração da sequência gerada é menor que a duração necessária e, portanto, é necessário ou repetir a sequência do passo ou alongar as durações multiplicando-as por um fator \mathcal{F} .

Uma revisão do estágio “Correspondência do Tempo” é vista na Figura 11.

7.3. AVALIAÇÃO

Para avaliar a correção da abordagem proposta, são comparadas as características das linhas de tempo M(I,II) com as linhas de tempo originais. A acurácia das taxas de transição das linhas de tempo M(I,II) e a precisão das durações dos κ -intervalos são analisadas.

7.3.1. Metodologia: Uma vez que os pares de nós têm comportamentos específicos e as suas distâncias max-min variam de um par para outro, para cada uma dessas distâncias max-min e cada um dos traços são geradas linhas de tempo M(I,II) correspondentes. Assim, linhas de tempo para distâncias max-min de 1 à 7 são obtidas. Por exemplo, a linha de tempo MI-7 replica o comportamento da vizinhança entre pares de uma distância max-min de 7 no Modo I. São geradas linhas de tempo MI e MII de aproximadamente 50.000 segundos.

Este trabalho compara pares de nós com distância max-min D com linhas de tempo MI- D e MII- D . Para cada linha de tempo, são analisadas a média das durações dos κ -contatos em comparação às linhas de tempo originais escolhidas. São também comparadas as transições de movimentação da vizinhança em linhas de tempo sintéticas M(I,II). As análises são realizadas com valores médios dos traços e linhas de tempo específicas entre pares de nós.

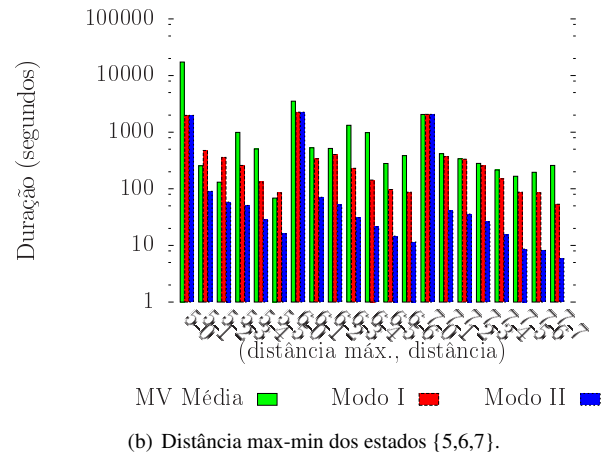
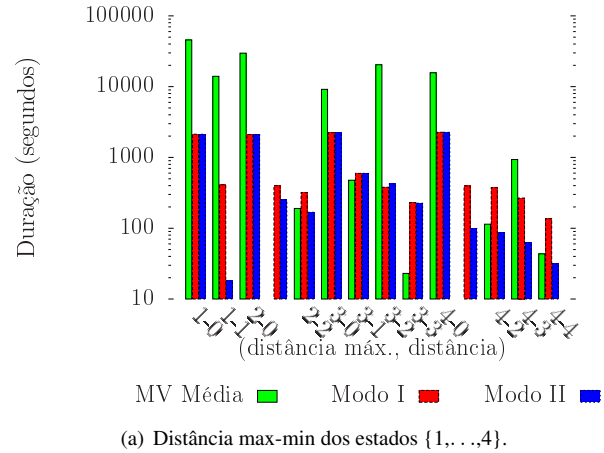


Figura 12. Duração média para κ -intervalo das linhas de tempo médias e das linhas de tempo M(I,II) geradas. Note que é usada uma escala logarítmica no eixo y .

7.3.2. Distribuição da duração do κ -intervalo: Os valores médios da duração do κ -intervalo para as linhas de tempo MI e MII foram calculados para todos os traços e para cada distância max-min em [1:7]. Os resultados para os traços *Infocom05* são apresentados, mas as observações podem ser aplicadas da mesma forma para os outros traços.

A Figura 12 apresenta a duração média do contato κ para o traço *Infocom05*. A primeira barra mostra o valor da movimentação média; a segunda barra indica o valor para as linhas de tempo MI; e a terceira mostra o valor para a linha de tempo MII. No eixo x , é observado o “valor D da distância max-min – a distância κ ”. Isso significa que D é a distância max-min necessária ($D \in [1:7]$) e κ é o valor considerado para a duração κ ($\kappa \leq D$). A Figura 12(a) apresenta as durações médias para $D \in [1:4]$, enquanto a Figura 12(b) mostra a duração média para $D \in [5:7]$. Note que o eixo y está em escala logarítmica.

Como esperado ao propor o Tiger, o Modo I foca em

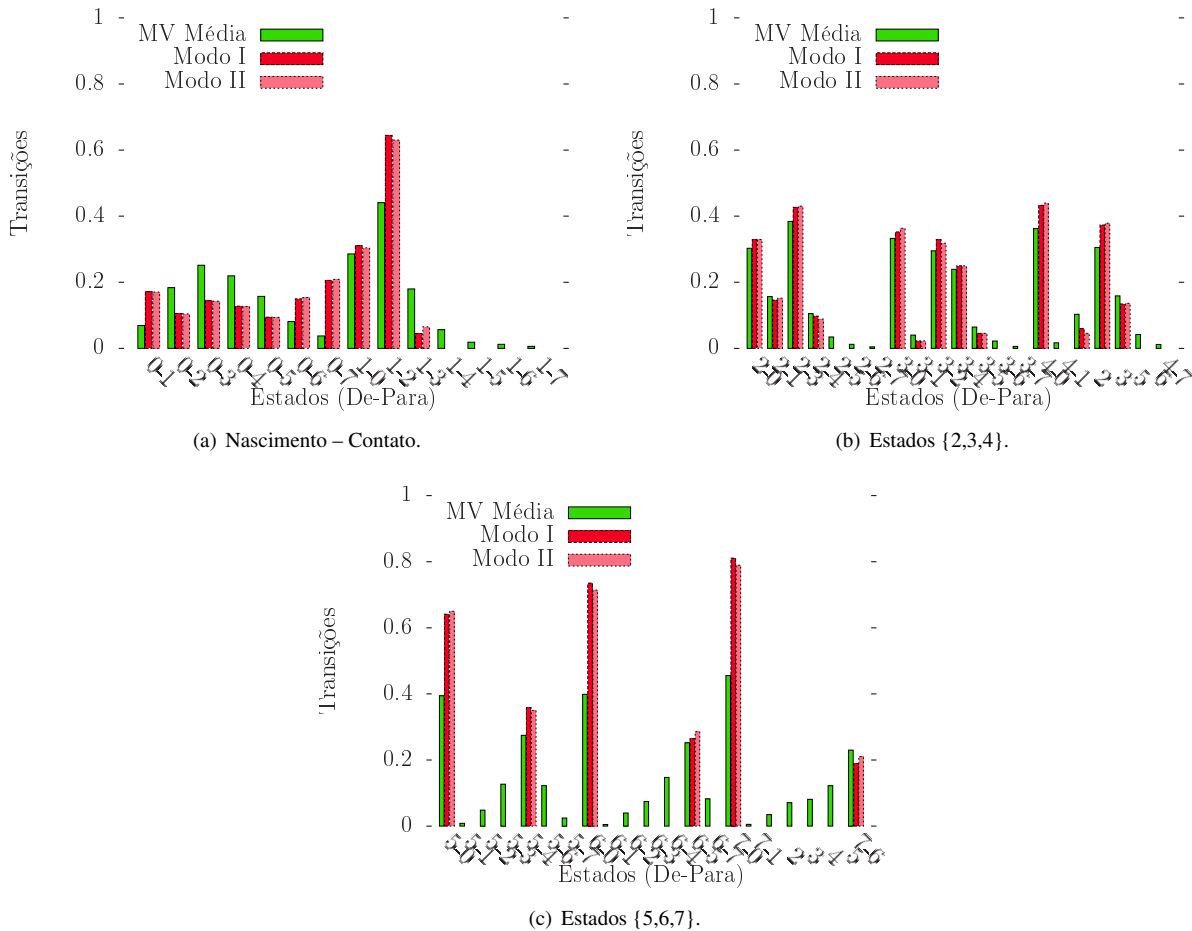


Figura 13. Transições da movimentação da vizinhança para as linhas de tempo médias do *Infocom05* e as linhas de tempo M(I,II).

refletir durações plausíveis de contato κ . Logo, a maioria das linhas de tempo MI observadas possui durações médias mais alinhadas com a duração da movimentação da vizinhança que as linhas de tempo MII. Para menores distâncias $D \in [1:3]$, há a presença das maiores diferenças entre a movimentação da vizinhança média e as linhas de tempo geradas. Essa diferença pode ir até um fator de 52 para 3-2 ($D = 3$ e $\kappa = 2$). Para valores maiores de $D \in [4:7]$, essa diferença diminui em média com um fator de 2,7. Percebe-se que as diferenças variam entre 0,30 e 8,7. As diferenças para valores D menores podem ser explicadas pelos parâmetros de geração do Tiger. Com linhas de tempo MI de duração máxima de 50.000 segundos, não se tem a mesma duração obtida com os traços originais onde alguns nós mal se vêem. As linhas de tempo MI tendem a coincidir com o comportamento médio da duração da movimentação da vizinhança. As linhas de tempo MII também coincidem em menor grau já que elas são basicamente linhas de tempo MI divididas pelo fator de adaptação \mathcal{F} .

7.3.3. Taxas de transição: Outro aspecto que deve ser cuidado ao replicar o comportamento entre pares vizinhos são as probabilidades de transição de um estado para outro. A Figura 13 mostra a análise das transições da movimentação da vizinhança para o comportamento médio do traço *Infocom05* para as linhas de tempo MI e MII. No eixo x , tem-se os estados “de-para”. Por exemplo, 1-2 indica a transição de uma distância de um salto (contato) para uma distância de dois saltos. Já no eixo y , o valor da probabilidade de transição é exibido. Na Figura 13(a), o valor da transição 1-2 é 0,45 para a movimentação da vizinhança média e por volta de 0,63 para as linhas de tempo MI e MII. As transições geradas pelos Modos I e II são ligeiramente diferentes entre si. Essas transições estão de acordo com a aparência geral da linha de tempo da movimentação da vizinhança. Para estados abaixo de 4, as transições tendem a estar de acordo com as linhas de tempo médias. Para estados acima de 5, o Tiger gera um número menor de transições não-nulas. Entretanto, deve-se compreender que a linhas de tempo média representa um comportamento agregado e que movimentos na

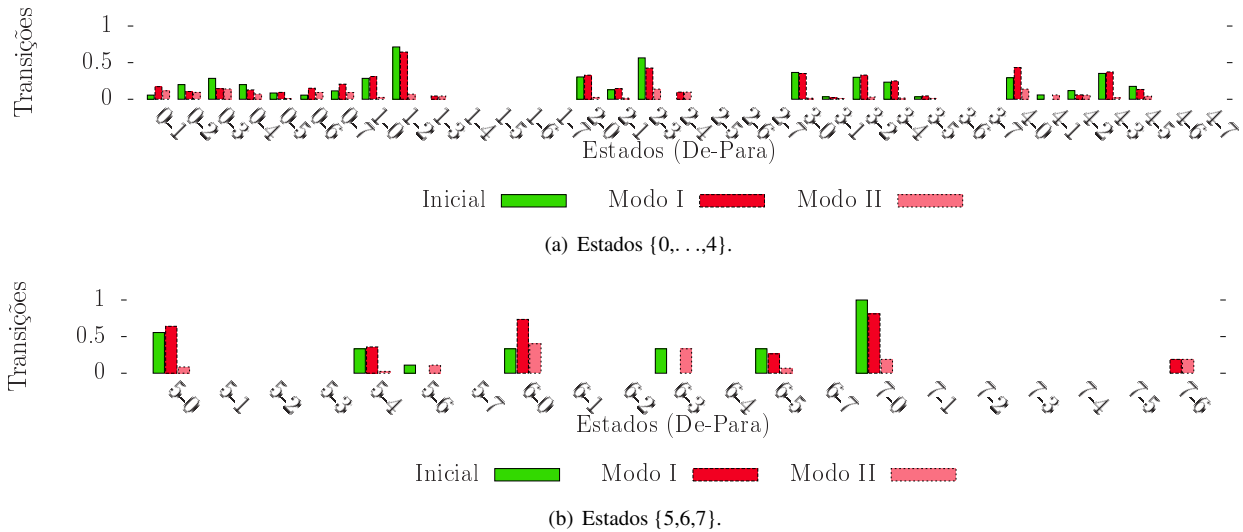


Figura 14. Valores de transição para a linha de tempo (30,36) e para a linha de tempo M(I,II) do traço *Infocom05*.

vizinhança entre 5 e 7 saltos são, de fato, bem reduzidos.

Para verificar esse fenômeno na vizinhança entre 5 e 7 saltos e observar como as linhas de tempo M(I,II) se comportam em comparação às linhas de tempo originais, foram observados o par (30, 36) do traço *Infocom05*. Na Figura 14, as transições da movimentação da vizinhança desse par, assim como as linhas de tempo MI e MII, são representadas. Para os estados abaixo de 4 na Figura 14(a), considerando tanto os valores das transições quanto as suas presenças, as linhas de tempo M(I,II) e a linha de tempo inicial (30,36) têm um comportamento muito similar. Na Figura 14(b), para estados maiores que 5, percebe-se que a linha de tempo (30,36) não tem muitas transições na zona de vizinhança entre 5 e 7 saltos. Há apenas movimentos do tipo $7 \rightarrow \infty$, $6 \rightarrow (\infty, 2, 5)$ e $5 \rightarrow (\infty, 3, 4)$. Para esses movimentos, nota-se que as linhas de tempo MI e MII têm valores próximos. Nesse caso, observa-se claramente como as linhas de tempo M(I,II) estão de acordo com a linha de tempo (30,36) original. Isso mostra o quanto o Tiger pode gerar linhas de tempo realistas.

8. CONCLUSÃO

Este trabalho modela a vizinhança de um nó usando a noção de κ -vizinhança assim como propõe um fluxo de trabalho para compreensão do seu comportamento. Esse fluxo de trabalho gera informações como a *linha de tempo* e as *probabilidades de transição*. As linhas de tempo permitem a análise da dinâmica das distâncias entre os pares de nós enquanto as probabilidades de transição detalham como os nós se movimentam uns em relação aos outros. Este estudo apresentou dois tipos principais de cadeias de

vizinhança: estendidas e curtas. Cada tipo foi discriminado de acordo com os estados alcançáveis. Além das cadeias identificadas, a predominância de apenas poucos tipos de movimentos na rede foi identificada. Esses movimentos, chamados de nascimento, morte e sequencial cobrem até 87% de todos os padrões analisados. Os padrões de movimentação de vizinhança ajudam a entender como é o comportamento da vizinhança, que é fundamental em redes oportunistas. Adicionalmente, os padrões de movimentação ainda apresentam um padrão de linha de tempo que foi usado neste trabalho para a criação de um gerador de vizinhança. O gerador proposto integra a análise de padrões de vizinhança propostas para fundamentar a geração de múltiplos tipos de traços de contato com características similares de vários tamanhos.

Referências

- [1] Roberta Calegari, Mirco Musolesi, Franco Raimondi, and Cecilia Mascolo. CTG: A Connectivity Trace Generator for Testing the Performance of Opportunistic Mobile Systems. In *ACM SIGSOFT Symposium on the Foundations of Software Engineering*, Dubrovnik, Croatia, September 2007.
- [2] Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing*, 6(6):606–620, 2007.
- [3] Vania Conan, Jérémie Leguay, and Timur Friedman. Characterizing Pairwise Inter-contact Patterns in Delay Tolerant Networks. In *International Con-*

- ference on Autonomic Computing and Communication Systems*, Rome, Italy, October 2007.
- [4] Sabrina Gaito, Elena Pagani, and Gian P. Rossi. Fine-Grained Tracking of Human Mobility in Dense Scenarios. In *IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, Rome, Italy, June 2009.
- [5] Marta C. Gonzalez, Cesar A. Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008.
- [6] Pan Hui, Jon Crowcroft, and Eiko Yoneki. BUBBLE Rap: Social-Based Forwarding in Delay-Tolerant Networks. In *ACM Mobihoc*, New York, NY, USA, November 2008.
- [7] Mirco Musolesi and Cecilia Mascolo. Designing mobility models based on social network theory. *SIGMOBILE Mob. Comput. Commun. Rev.*, 11:59–70, July 2007.
- [8] Carina T. Oliveira, Marcelo D. D. Moreira, Marcelo G. Rubinstein, Luís Henrique M. K. Costa, and Otto Carlos M. B. Duarte. *Redes Tolerantes a Atrasos e Desconexões*, chapter 5, pages 203–256. Minicursos do Simpósio Brasileiro de Redes de Computadores. Sociedade Brasileira de Computação, Belém, PA, Brazil, May 2007.
- [9] Santashil Pal Chaudhuri, Jean-Yves Le Boudec, and Milan Vojnovic. Perfect Simulations for Random Trip Mobility Models. In *IEEE Infocom*, Miami, Florida, USA, August 2005.
- [10] Iain Parris, Greg Bigwood, and Tristan Henderson. Privacy-enhanced social network routing in opportunistic networks. In *IEEE International Conference on Pervasive Computing and Communications*, Mannheim, Germany, March 2010.
- [11] Andrea Passarella and Marco Conti. Characterising aggregate inter-contact times in heterogeneous opportunistic networks. In *IFIP Networking*, Valencia, Spain, May 2011.
- [12] Tiphaine Phe-Neau, Miguel Elias M. Campista, Marcelo D. de Amorim, and Vania Conan. Padrões de Mobilidade de Vizinhança em Redes de Contato Intermitente. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, Brasília, Brazil, May 2013.
- [13] Tiphaine Phe-Neau, Marcelo Dias de Amorim, and Vania Conan. Fine-Grained Intercontact Characterization in Disruption-Tolerant Networks. In *IEEE Symposium on Computers and Communication*, Kerkyra, Greece, June 2011.
- [14] Tiphaine Phe-Neau, Marcelo Dias de Amorim, and Vania Conan. Using neighborhood beyond one hop in disruption-tolerant networks, 2011. <http://arxiv.org/abs/1111.0882v1>.
- [15] Tiphaine Phe-Neau, Marcelo Dias de Amorim, and Vania Conan. Vicinity-based DTN Characterization. In *ACM MobiOpp*, Zurich, Switzerland, March 2012.
- [16] Marcel Salathé, Maria Kazandjieva, Jung Woo Lee, Philip Levis, Marcus W. Feldman, and James H. Jones. A high-resolution human contact network for infectious disease transmission. *PNAS*, 107(50):pp. 22020–22025, 2010.
- [17] Pierre-Ugo Tournoux, Jeremie Leguay, Farid Benbadis, John Whitbeck, Vania Conan, and Marcelo Dias de Amorim. Density-aware routing in highly dynamic DTNs: The rollernet case. *IEEE Transactions on Mobile Computing*, 10:1755–1768, 2011.

**Revista Brasileira de Redes de Computadores
e Sistemas Distribuídos**

**LARC - Laboratório Nacional de Redes de Computadores
SBC - Sociedade Brasileira de Computação**