Filtros de precedência: agregação anónima de dados de trânsito Bluetooth*

Nelson Gonçalves[1], Carlos Baquero[1], Miguel Borges[1], and Rui José[2]

[1] HASLab, INESC TEC & Universidade do Minho
[2] Centro Algoritmi, Universidade do Minho

Resumo À medida que os dispositivos Bluetooth se tornam cada vez mais comuns, existe um potencial crescente para tirar partido das capacidades desta tecnologia assim como das infraestruturas já existentes para cenários de rastreio massivo de dispositivos. Todavia, a recolha de informação destes equipamentos pode criar problemas de privacidade, pois a criação de um registo permanente e/ou desprotegido dos dados recolhidos permitiria o escrutínio do dia-a-dia dos seus utilizadores. Esse registo será social e juridicamente inaceitável, potenciando a rejeição destas tecnologias.
Existem porém, técnicas não triviais que permitem guardar a informação sem comprometer a privacidade do utilizador. O objetivo deste artigo é demonstrar como essas técnicas podem ser usadas num cenário específico, o estudo da mobilidade humana, dos seus padrões e frequência dos mesmos, minimizando a quantidade de dados recolhidos assim como o risco para a privacidade dos utilizadores.

Palavras Chave: Privacidade, Algoritmos Probabilísticos, Bloom Filters, Relógios Vetoriais, Bluetooth

1 Introdução

À medida que a tecnologia Bluetooth se torna cada vez mais disseminada, esta promove um potencial cada vez maior para a sua utilização como uma tecnologia para rastreio e atuação em cenários urbanos. O rastreio Bluetooth consiste no processo de descoberta através do qual um dispositivo pode informar-se da presença de outros dispositivo nas redondezas. Se esses dispositivos estiverem em *modo visível* , irão responder com os seus endereços Bluetooth (MAC 48 bits) e possivelmente com informação adicional tal como o nome do dispositivo, o seu tipo (ex: telemóvel, computador ou acessório) e serviços disponíveis.

Um detetor Bluetooth é um dispositivo que rastreia periodicamente a vizinhança em busca de dispositivos que usem o mesmo protocolo. Sempre que deteta um dispositivo, regista localmente a ocorrência e atribui-lhe um marcador temporal. Esta informação pode posteriormente ser utilizada por serviços

* Este trabalho é financiado por Fundos FEDER através do Programa Operacional Fatores de Competitividade - COMPETE e por Fundos Nacionais através da FCT - Fundação para a Ciência e a Tecnologia no âmbito do projeto FCOMP-01-0124-FEDER-010114.

ou aplicações externas. A existência de vários detetores Bluetooth espalhados num determinado espaço físico, permite a criação de uma rede de rastreio Bluetooth que poderá ser usada como ferramenta para observar, modelar e analisar esse mesmo espaço do ponto de vista físico (as relações espaciais dos diferentes dispositivos), técnico (das propriedades tecnológicas dos dispositivos) e social/humano (dos padrões comportamentais que se podem inferir a partir de padrões de uso) [14].

As observações registadas nessa rede podem oferecer um novo ponto de vista acerca da maneira como as pessoas utilizam e se movem dentro do espaço. A existência de vários pontos de rastreio constituintes de um sistema, pode servir como suporte para a monitorização do trânsito de dispositivos entre os diferentes pontos de observação. Dada a natureza pessoal da maioria dos dispositivos Bluetooth, os resultados dos rastreios podem ser usados para inferir os padrões de mobilidade dos seus utilizadores. A administração de um centro comercial poderá estar interessada em identificar padrões de visita dos clientes (saber quais as lojas mais visitadas, ordem pela qual são visitadas, etc).

Num outro cenário, um parque temático pode desejar melhorar as atrações e reduzir as filas de espera através do estudo dos padrões de movimento dos visitantes. De modo semelhante, uma agência de turismo pode querer identificar quais os locais mais visitados assim como os padrões de visita mais usuais num determinado sítio, tudo isto de maneira a melhorar a informação disponibilizada e as sugestões dadas aos turistas.

A infraestrutura necessária para este tipo de aplicações, pode ser construída de raiz dado o custo cada vez mais reduzido do equipamento tecnológico envolvido. Em alternativa, pode reutilizar recursos já existentes, como em certos espaços urbanos onde existe já um grande número de detetores Bluetooth. Estes são propriedade de várias entidades diferentes e servem diferentes propósitos tais como marketing de proximidade, localização de dispositivos, partilha de informação, entre outros. Estes mesmos detetores poderiam ser usados como nós de uma infraestrutura colaborativa de rastreio Bluetooth. Cada nó permaneceria autónomo e continuaria a sondar os dispositivos Bluetooth tendo em conta os seus objetivos, partilhando no entanto parte dessa informação e tornando-se assim num membro da rede colaborativa de rastreio. Ambas as estratégias fazem com que a criação deste tipo de infraestrutura seja relativamente simples e praticável a curto prazo.

O desafio maior será o de arranjar uma forma de permitir a colaboração entre os vários nós da rede sem no entanto descurar a garantia de privacidade dos utilizadores. Para identificar transições entre nós, é necessária a capacidade de associar duas observações do mesmo dispositivo em locais diferentes. Uma maneira de o fazer seria comparar os endereços Bluetooth dos dispositivos observados em cada nó. No entanto, esta abordagem não é apropriada em termos de privacidade. Apesar da capacidade que alguns dispositivos têm de alternar entre vários endereços, os endereços Bluetooth obtidos pelos detetores, são na sua maioria um identificador único, confiável e permanente de um dispositivo, e por extensão do seu respetivo utilizador. O fato de não existir nenhuma maneira

direta de estabelecer correspondência entre um endereço e uma pessoa em particular é apenas uma fronteira muito ténue no que diz respeito ao anonimato. Se alguém quisesse seguir um indivíduo em particular, seria suficiente sondar o espaço em que esse mesmo indivíduo se encontra em duas ou três ocasiões. Com essa informação e por disjunção seria muito fácil isolar o seu endereço Bluetooth, mesmo sem qualquer conhecimento do nome do dispositivo.

O uso de funções de dispersão (*hash*) é muitas vezes sugerido como uma técnica de anonimização dos dados. Recorrendo a esta estratégia, um endereço Bluetooth é substituído por uma chave a partir da qual não é possível recriar o endereço original. Contudo, esta técnica sem qualquer alteração, também não é adequada para rastreio Bluetooth em larga escala: é possível, mesmo com pouca informação acerca de um determinado dispositivo, identificar a respetiva chave, permitindo assim a sua utilização como identificador único.

Existem exemplos , tais como do caso Netflix [13], que mostram como é surpreendentemente fácil detetar a identidade do utilizador a partir de dados propostos como sendo anónimos, simplesmente com a adição de alguma informação básica à informação já recolhida [15]. Portanto, o simples registo sistemático de avistamentos Bluetooth em vários locais constitui uma ameaça à privacidade na medida em que potencia a criação de um sistema de vigilância/monitorização de pessoas.

1.1 Objetivos

Neste artigo, é discutido o uso de avistamentos de dispositivos Bluetooth, gerados por um grupo de nós cooperantes, com objetivo de adquirir conhecimento acerca dos movimentos dos seus utilizadores, sem o comprometimento da sua privacidade. Em particular, queremos explorar de que modo a utilização de técnicas de sumarização estocásticas podem servir como uma abordagem ao rastreio Bluetooth que preserva a privacidade.

A técnica proposta neste artigo permite que os nós gerem informação probabilística acerca da proveniência dos visitantes detetados nesse nó. Essa informação diz respeito aos nós que foram visitados imediatamente antes e a sua precisão de acerto pode ser ajustada para ter um nível de incerteza compatível com negação plausível, ou seja, um indivíduo deve poder negar com credibilidade que esteve num determinado local. No entanto, a informação acerca do agregado de todas as visitas deve ser exata o suficiente para ser relevante no estudo dos padrões de mobilidade do grupo.

2 Trabalho relacionado

No contexto da computação ubíqua, existem já várias técnicas cujo o objetivo é garantir a privacidade dos dados relativos à localização de dispositivos e em consequência, dos seus utilizadores.

Em cenários onde são os dispositivos que comunicam aos nós a sua localização, existem estratégias de dissimulação da localização como por exemplo a

introdução de ruído nos dados [7] ou a degradação explícita da sua resolução, i.e., os dispositivos reportam a região onde se encontram em vez de um ponto específico [16].

Em [2,6], a privacidade é garantida através da utilização de pseudónimos por parte dos dispositivos e do conceito de zonas mistas (*mix zones*). Nestas zonas, os nós de rastreio não recebem qualquer tipo de informação por parte dos dispositivos. Apenas é permitido aos dispositivos a mistura das suas identidades, partindo do principio que mudam de identificador (pseudónimo) assim que entram na zona mista. De notar que nos casos em que a cardinalidade do conjunto de anonimato (número de dispositivos na zona mista) é 1, é trivial para os nós de rastreio relacionar antigo e novo pseudónimos.

Uma outra estratégia para assegurar a privacidade, consiste na degradação dos dados [1,17]. Neste tipo de abordagens a privacidade é garantida através de uma generalização dos dados. Este procedimento é feito com recurso a árvores de generalização [1] ou grafos de generalização [17]. Este tipo de estruturas agrupa um conjunto de regras a aplicar na generalização dos dados. A aplicação das mesmas depende da política de ciclo de vida(*LFS - life cycle policy*). Esta política consiste no conjunto de transições entre os ramos/vértices das estruturas de generalização assim como dos eventos que as desencadeiam. O maior desafio deste tipo de abordagens está relacionado com a dificuldade na construção dos conjuntos de regras e de estruturas de generalização.

3 Modelo do Sistema

No modelo de sistema por nós definido, assumimos a existência de uma rede de nós heterogéneos e autónomos que colaboram entre si no processo de rastreio de dispositivos Bluetooth. Estes nós podem ter acesso a vários tipos de informação sobre os dispositivos Bluetooth: o seu endereço MAC, a hora a que foram avistados, o tipo de protocolos Bluetooth que suportam, entre outros.

Tal como foi dito na secção 1, existe a possibilidade de utilizar recursos (nós) já existentes e portanto, o nosso modelo não impõe restrições quanto ao tipo de informação registada pelos mesmos, uma vez que este tipo de nós continuará a desempenhar as mesmas funções que tinha antes de fazer parte do sistema. A nossa preocupação, é para com a informação que os nós podem trocar no contexto do rastreio coletivo e dos riscos ao nível de privacidade que daí podem advir, nomeadamente, a capacidade de seguir com exatidão a movimentação de um dispositivo e por conseguinte o seu respetivo dono.

No seu dia-a-dia e dependendo das suas necessidades, as pessoas visitam vários locais diferentes. Por exemplo, uma pessoa $P1$ quer comprar um computador portátil novo. Essa mesma pessoa visita a loja $L1$ que não tem o modelo que pretende, depois visita a loja $L2$ que também está sem stock, de seguida visita $L3$ onde o preço é um pouco elevado, até que o compra na loja $L4$. Para representar este comportamento vamos introduzir a noção de *traços de mobilidade*. Um traço de mobilidade é simplesmente uma representação dos locais visitados na ordem pela qual foram visitados, neste caso específico, o traço de $P1$

é $T_{P1} = \{L1, L2, L3, L4\}$. O nosso mecanismo, *Filtros de Precedência*, permite registar informação acerca dos traços individuais das pessoas, de uma maneira compatível com negação plausível. Essa informação pode depois ser processada para obter dados mais precisos acerca dos hábitos do agregado, por exemplo, neste caso a ordem pela qual estas lojas são visitadas provavelmente está correlacionada com o prestígio/popularidade das mesmas.

Uma estratégia comum que visa o reforço da privacidade é a redução ao indispensável dos dados recolhidos. No nosso caso em específico, o objetivo é a deteção de padrões de mobilidade entre os nós. Como tal, a capacidade de detetar o mesmo dispositivo em nós distintos, necessária para determinar a sequência de presenças nesses mesmos nós, é um requisito chave. Para minimizar a quantidade de informação recolhida, podemos ignorar informações tais como a duração do avistamento, hora a que foi avistado, nome do dispositivo e protocolos Bluetooth suportados, entre outros. Apenas necessitamos do endereço MAC do dispositivo.

Sempre que um dispositivo é detetado, o nó que fez o avistamento regista localmente esse acontecimento. Essa informação é depois utilizada na contabilização das transições dos dispositivos entre os vários nós do sistema. No caso de um sistema centralizado, essa contabilização apenas pode ser feita no servidor, uma vez que apenas este terá informação suficiente para o fazer. No caso de uma arquitetura descentralizada, essa contagem pode ser feita localmente pelos nós. Na abordagem centralizada, apenas o servidor tem informação acerca dos avistamentos nos vários nós do sistema, uma vez que cada nó apenas partilha a informação local com o servidor. Por sua vez, na abordagem descentralizada, a partilha de informação é feita entre todos os nós, portanto todos podem ter acesso à mesma informação. Ambos os modelos têm vantagens e desvantagens. Por exemplo, a abordagem centralizada não é tolerante a faltas, basta que o servidor falhe para que sistema de registo de transições deixe de funcionar. Tal não sucede na abordagem descentralizada, uma vez que a mesma informação está repetida em vários nós (redundância), o sistema pode continuar a funcionar corretamente mesmo em caso de falhas nalguns nós. Esta redundância do modelo descentralizado, confere também uma maior disponibilidade da informação, face ao modelo centralizado. No entanto, a troca de informação entre todos os nós, faz com que a carga exercida na rede seja maior no cenário descentralizado comparativamente ao centralizado.

De modo a alcançar o objetivo a que nos propomos e tendo em conta as restrições do modelo apresentadas, a nossa solução baseia-se no seguinte conjunto de premissas:

- Apesar de não ser possível fazer assunções acerca da maneira como cada nó lida com a informação Bluetooth resultante dos rastreios, a nossa solução nunca deverá exigir que o endereço Bluetooth deixe o sensor, ou qualquer outra informação que sirva para identificar univocamente o dispositivo.
- Nenhum elemento do sistema deve em algum ponto ter toda a informação necessária que lhe permita indicar com precisão o conjunto de nós previamente visitados por um qualquer individuo.

- O resultado agregado que um nó pode criar acerca do conjunto de dispositivos avistados deve ser suficientemente preciso para permitir o seu uso em cenários de observação de mobilidade tais como caminhos mais comuns e nível de semelhança entre locais.
- Não existem falhas de comunicação e a troca de informação entre quaisquer dois pontos no sistema é mais rápida do que o tempo que demora uma pessoa a deslocar-se entre os mesmos. Isto garante que a ordem pela qual as os avistamentos são registados é a real.

4 Mecanismo

Neste capítulo é descrito o modelo de funcionamento dos Filtros de Precedência. No entanto, primeiramente é dada uma pequena explicação dos algoritmos nos quais os filtros são baseados, nomeadamente, *Counting Bloom Filters* [5,12] e Relógios Vetoriais [11,8].

4.1 Bloom Filters

Os *Bloom Filters* (BFs) foram criados em 1970 por Burton Howard Bloom [3]. São uma estrutura probabilística simples e eficiente em termos de memória. Permitem representar conjuntos de elementos de maneira compacta e na sua versão base suportam duas operações: verificação de pertença e adição de elementos ao conjunto. Os *Bloom Filters* permitem a ocorrência de falsos positivos mas não de falsos negativos.

Um conjunto S, com n elementos, tal que $S = \{x_1, x_2, x_3, ..., x_n\}$, é representado por *Bloom Filters* tradicionalmente implementados com recurso a um vetor de M bits (inicializados a 0) e K funções de dispersão independentes $\{h_1, h_2, ..., h_k\}$. Cada uma das funções de dispersão mapeia um elemento do conjunto num número inteiro uniformemente distribuído no intervalo $\{1, ..., M\}$. Para cada elemento $x \in S$, os bits das posições $h_i(x)$ com $1 \leq i \leq k$, passam a 1. Tendo em conta a independência das funções de dispersão, uma qualquer posição de M pode ser marcada a 1 várias vezes. Em casos extremos é possível que $h_1(x) = h_2(x) = ... = h_k(x)$. Para evitar estes casos, usamos uma variante de BFs apresentada em [4] que divide os M bits entre as K funções de dispersão, através da criação de K partições com $m = M/K$ bits de tamanho. Isto assegura que cada elemento adicionado ao BF, é sempre representado por K bits. Dado um Bloom Filter BF_S, verificar se um elemento $z \in BF_S$, consiste em verificar se todos os índices $h_i(z)$ têm o bit a 1. Caso não estejam, temos a certeza que $z \notin BF_S$ (não há falsos negativos). Caso contrário, se todos os bits estiverem marcados a 1, assume-se que $z \in BF_S$, sabendo que, no entanto, esse pressuposto pode estar errado devido à ocorrência de falsos positivos.

Counting Bloom Filters Os *Counting Bloom Filters* (CBFs) foram originalmente apresentados por Fan *et al.* [5] apesar de só mais tarde, através Mitzenmacher [12] terem ganho a sua designação atual. Foram introduzidos de maneira

a permitir a eliminação de elementos de *Bloom Filters*. Vamos supor que temos um conjunto que varia ao longo do tempo, onde elementos podem ser inseridos e removidos. A inserção de elementos pode facilmente ser feita com recurso a um BF normal, utilizando o output das K funções de dispersão e marcando os bits correspondentes a 1. No entanto, a remoção de elementos não pode ser feita revertendo o processo. Os índices resultantes das funções de dispersão, não podem simplesmente ser marcados a 0 uma vez que essas posições podem também fazer parte do resultado de outros elementos do conjunto.

Num CBF, cada posição é um pequeno contador ao invés de um simples bit. Quando um elemento é inserido, os respetivos contadores são incrementados e quando é eliminado os mesmos contadores são decrementados. É no entanto necessário a escolha de contadores suficientemente grandes de maneira a prevenir a quebra (*overflow*) dos mesmos.

4.2 Relógios Vetoriais

Para perceber melhor o funcionamento dos relógios vetoriais, é necessário entender o conceito de causalidade. Causalidade é uma relação através da qual é possível relacionar 2 eventos, a *causa* e o *efeito*. No contexto dos sistemas distribuídos, a causalidade é expressa através da relação de precedência (*happens-before*) introduzida por Lamport [9] e denotada através do símbolo $\rightarrow$. Os relógios vetoriais, foram introduzidos por Colin Fidge [8] e Friedemann Mattern [11] em 1998 e são um algoritmo que implementa a relação de precedência. Neste algoritmo, cada processo P_i tem um vetor de inteiros com n posições $RV_i[1..n]$, n é o número processos do sistema. O seu funcionamento segue o seguinte conjunto de regras:

1. Todas as posições do vetor são inicializadas a 0.
2. De cada vez que o estado de um processo P_i muda, este deve incrementar o seu valor no vetor $RV_i[i]$, isto é, $RV_i[i] = RV_i[i] + 1$.
3. Sempre que um processo P_i envia uma mensagem, em anexo deve também enviar uma cópia do seu relógio vetorial RV_i.
4. Quando um processo P_i recebe uma mensagem m, deve atualizar o seu relógio, utilizando para isso a fórmula $\forall x : RV_i[x] = max(RV_i[x], m.RV[x])$, onde $m.RV$ simboliza o relógio vetorial anexado a m.

Os relógios vetoriais são então um algoritmo capaz representar eficazmente a relação de causalidade e a ordem parcial que ela define. Dados quaisquer 2 eventos distintos x e y:

$$\forall x, y : x \rightarrow y \iff RV_x < RV_y$$

Onde $RV_x < RV_y$ se traduz por:

$$\forall k : RV_x[k] \leq RV_y[k] \wedge (\exists k : RV_x[k] < RV_y[k])$$

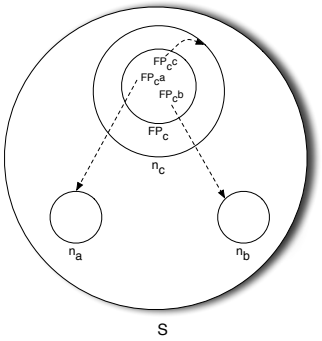

Figura 1. Diagrama das entidades e estruturas de dados relevantes

4.3 Filtros de Precedência

Equiparando os detetores Bluetooth a processos e o avistamento de dispositivos Bluetooth a eventos de transição de estado, é possível aplicar ao cenário de trânsito Bluetooth, os conceitos de sistemas distribuídos discutidos anteriormente. Os filtros de precedência baseiam-se nesta ideia, e permitem fornecer dados precisos sobre os padrões de mobilidade ao nível macroscópico, sem descurar a privacidade dos utilizadores.

Os filtros de precedência podem ser vistos como uma implementação de relógios vetoriais cuja diferença é a utilização de CFBs (um por cada nó do sistema) por parte dos filtros de precedência em lugar dos inteiros (um por cada processo) utilizados pelos relógios vetoriais.

No modelo descentralizado cada nó tem um filtro de precedência e os nós comunicam entre si aquando dos avistamentos de dispositivos de maneira a manterem os seus filtros atualizados. Por sua vez, na abordagem centralizada, cada nó possui apenas um CBF, cuja atualização é garantida através de comunicação com o servidor. O servidor por sua vez, mantém uma cópia da informação registada em cada um dos CBFs dos nós, ou seja, um filtro de precedência.

Na experiência realizada neste trabalho, foi utilizado o modelo descentralizado de funcionamento dos filtros de precedência. Tendo isto em conta, o modelo de funcionamento dos mesmos é o seguinte: supondo que temos um conjunto de detetores(nós) S, cada nó $n \in S$ possui um filtro de precedência FP_n que por sua vez é constituído por um conjunto de *Counting Bloom Filters* um por cada nó $z \in S$. A notação FP_n^z é utilizada para fazer referência ao CBF para o detetor z presente em FP_n, de acordo com o que está representado na Figura 1.

Todos os CBFs têm o mesmo tamanho M, são inicializados a 0 e utilizam o mesmo conjunto de funções de dispersão K. Isto assegura que o mesmo dispositivo é corretamente identificado em nós diferentes (aquando do avistamento, será mapeado nos mesmo índices). Uma outra maneira de pensar sobre os filtros de precedência é ver os mesmos como uma matriz, cujo numero de linhas é igual ao número de nós do sistema e o número de colunas é igual a M.

De cada vez que um nó n avista um dispositivo d, o seu filtro de precedência FP_n é atualizado de acordo com o seguinte conjunto de regras:

1. Utilizando as K funções de dispersão, o nó n calcula o conjunto de índices I_d que consiste no conjunto de saída das K funções de dispersão para o dispositivo d ($I_d = \bigcup_{k \in K} h_k(d)$).
2. O nó n envia o conjunto de índices I_d para os quais pretende saber os valores, a todos os outros nós de S.
3. Cada um dos restantes nós q pertencentes a Z ($Z = S \backslash \{n\}$) responde com um conjunto de tuplos R_q, contendo os índices pretendidos e os valores associados a esses índices dos vários CBFs pertencentes a FP_z ($R_q^{I_d} = (i, \{FP_z^x[i]\}), \forall i \in I_d, \forall x \in Z$).
4. Depois da receção das respostas, o nó n calcula o máximo dos valores recebidos para cada um dos índices I_d dos CBFs pertencentes aos restantes nós ($FP_n^z[i] = \max(R_q[i][z] \forall q \in Z), \forall i \in I_d, \forall z \in Z$).
5. Por fim, n atualiza os índices I_d do seu CBF (FP_n^n). Para cada índice $i \in I_d, FP_n^n[i] = \max(FP_n^d[i]) + 1 \ \forall d \in S$. A soma de uma unidade ao valor máximo guardado nos vários nós, permite ao nó que avista o dispositivo dominar todos os restantes na operação que retorna a causalidade entre os locais visitados. Por outras palavras, é a chave para a obtenção da ordem pela qual os locais foram visitados.

Este conjunto de regras, permite aos filtros de precedência registar informação acerca da precedência dos locais visitados por um dispositivo. Dados o conjunto de índices I_d relativos ao dispositivo d e qualquer par de nós de rastreio (localizações), x e y, o avistamento de d em x precede o avistamento em y, $x \rightsquigarrow y$ se:

$$x \rightsquigarrow y \iff FP_x[I_d] < FP_y[I_d]$$

Onde $FP_x[I_d] < FP_y[I_d]$ se traduz por:

$$\forall i \in I_d : FP_x^x[i] < FP_y^y[i]$$

Os traços de mobilidade, que capturam o comportamento dos indivíduos, captam uma relação de ordem total entre os locais visitados, isto é, dados 2 locais é sempre possível saber a ordem de visita de um local relativamente ao outro. No entanto, dado que os filtros de precedência se baseiam na relação de precedência, tal não é possível. Os filtros de precedência apenas se recordam do último avistamento do dispositivo em cada local. Por exemplo, partindo do seguinte traço de mobilidade $T_{P1} = \{L1, L2, L1, L3, L2, L4, L1\}$, onde os locais $L1$ e $L2$ são visitados várias vezes, os filtros de precedência conseguem no melhor dos casos obter $T_{P1} = \{L3, L2, L4, L1\}$, devido às propriedades de irreflexividade e anti-simetria da relação de precedência. No entanto, esta característica dos filtros de precedência (que pode ser vista como uma espécie de degradação automática de dados), oferece a garantia de que o histórico de avistamentos relativo a um qualquer indivíduo está limitado superiormente pelo número de detetores Bluetooth existentes no sistema.

A privacidade oferecida pelos filtros de precedência, pode ainda ser ajustada com recurso à taxa de falsos positivos P dos CBFs. Quanto maior for esta taxa, maior será a imprecisão dos filtros de precedência. A ocorrência de falsos positivos nos CBFs, tem como consequência o aparecimento de *transições fictícias*, isto é, o traço causal obtido a partir da consulta dos filtros, contém locais que o traço original não possui. São estas propriedades dos filtros de precedência, que permitem que seja plausível a um utilizador negar a veracidade dos dados que lhe dizem respeito.

5 Métricas

Para testar o desempenho deste mecanismo, comparamos os traços de mobilidade que a relação de precedência permite obter com os traços que realmente são obtidos a partir dos filtros de precedência. Exemplificando: dado o traço real $T_P = \{L1, L2, L2, L1, L3\}$, o traço ideal que lhe corresponderia do ponto de vista da relação de precedência (última presença num dado local) e que é utilizado como referência, será $T_P = \{L2, L1, L3\}$. O seu conjunto de transições é $\{(L2, L1), (L2, L3), (L1, L3)\}$. Este conjunto de transições é então comparado com um conjunto de transições semelhante mas proveniente dos filtros de precedência.

O desempenho dos filtros de precedência é medido com recurso a dois tipos de métricas. A métrica individual, que mede a probabilidade de ser falsa a seguinte afirmação - "a pessoa P, visitou o local $L1$ antes de visitar o local $L2$". Para tal, calcula-se a cardinalidade do conjunto mutuamente exclusivo entre as transições pertencentes ao traço referência (T_{ref}) e as transições pertencentes ao traço do FP (T_{FP}), de acordo com a Equação (1).

$$\frac{|(T_{ref} \bigcup T_{FP}) \backslash (T_{ref} \bigcap T_{FP})|}{|T_{FP}|} \tag{1}$$

A métrica global mede o peso relativo de uma transição (ex. (L1,L3)) face a todas as transições ocorridas. Para tal, divide o número total de ocorrências da transição, pela soma do número total de todas as ocorrências. Esta métrica permite caracterizar a importância relativa de cada tipo de transição.

6 Avaliação

Para avaliar o desempenho dos filtros de precedência, utilizamos um conjunto de dados real, com informação sobre avistamentos de dispositivos Bluetooth em nós estáticos. Este conjunto de dados é resultado do trabalho de Leguay *et al.* [10].

A partir da análise da Figura 2, observa-se que o comportamento dos filtros de precedência vai de encontro ao pretendido e que tal como seria expectável o aumento da probabilidade de falsos positivos provoca o aumento da imprecisão do mecanismo. Este aumento tem como consequência o aparecimento de *transições fictícias*. Tem também como consequência o aumento das imprecisões

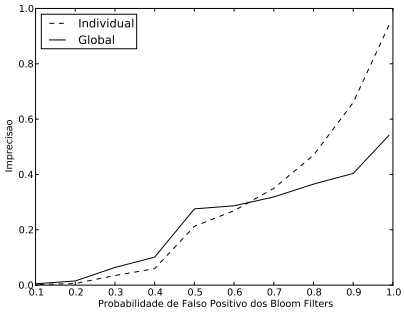

Figura 2. Resultados gráficos que ilustram a variação da imprecisão dos filtros de precedência de acordo com o aumento da taxa de falsos positivos dos Counting Bloom Filters

individual e global do mecanismo. No entanto, para probabilidades de falsos positivos elevadas ($\gtrsim 0.6$), o aumento da incerteza global do mecanismo (linha contínua da Figura 2) é inferior ao aumento da imprecisão dos traços individuais (linha tracejada da mesma Figura). Tal sucede devido ao facto de a incerteza global medir a relação entre cada tipo de transição e o total de transições (i.e, o seu quociente mantem-se aproximadamente constante). Acresce também que do ponto de vista global, o efeito das *transições fictícias* acaba por se diluir dado o aumento do número total de transições.

Esta característica inviabiliza o caso onde se pretenda utilizar a informação dos filtros para identificar com precisão os padrões de visita de uma determinada pessoa. Em simultâneo, providenciam informação credível a um nível macroscópico, onde o relevante não é o número absoluto de pessoas que transitaram de um local $L1$ para um local $L2$, mas sim o peso relativo de ocorrências dessa mesma transição comparativamente ao conjunto de todas as transições.

Ainda no contexto de análise de desempenho, desenvolvemos um gerador de traços sintético modelado a partir do conjunto de dados real. A motivação para criação do mesmo foi a modelação de cenários em que número de locais e de pessoas são arbitrários.

7 Conclusão

Neste trabalho apresentamos uma estratégia de reconhecimento de movimento de dispositivos Bluetooth por uma rede de nós de rastreio cooperantes. Mostramos que o reconhecimento de movimentos preserva a privacidade dos utilizadores com recurso a técnicas de sumarização estocásticas (*Bloom Filters*) e simultaneamente mantém a relação de precedência (traços) entre locais visitados graças à utilização de relógios vetoriais.

A técnica proposta neste artigo permite que os nós gerem informação probabilística com taxa de acerto variável acerca da proveniência dos dispositivos visitantes, dando-lhes a possibilidade de negação plausível.

Neste trabalho mostramos também resultados de simulação que corroboram a nossa abordagem: é possível a caracterização fiável dos padrões de mobilidade do grupo (bem como do número de dispositivos em transito) equilibrando em simultâneo a preservação da privacidade dos utilizadores individuais.

Referências

1. N. Anciaux, L. Bouganim, H. Van Heerde, P. Pucheral, and P.M.G. Apers. Instantdb: enforcing timely degradation of sensitive data. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 1373–1375. IEEE, 2008.
2. A.R. Beresford and F. Stajano. Location privacy in pervasive computing. *Pervasive Computing, IEEE*, 2(1):46 – 55, jan-mar 2003.
3. Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, July 1970.
4. F. Chang, Francis Chang, and Wu chang Feng. Approximate caches for packet classification. In *In IEEE INFOCOM*, pages 2196–2207, 2004.
5. Li Fan, Pei Cao, Jussara Almeida, and Andrei Z. Broder. Summary cache: A scalable wide-area web cache sharing protocol. In *IEEE/ACM Transactions on Networking*, pages 254–265, 1998.
6. J. Freudiger, R. Shokri, and J.P. Hubaux. On the optimal placement of mix zones. In *Privacy Enhancing Technologies*, pages 216–234. Springer, 2009.
7. M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 31–42. ACM, 2003.
8. Colin J.Fidge. Timestamps in message-passing systems that preserve the partial ordering. *Australian Computer Science Communications Vol.10*, 1988.
9. Leslie Lamport. Ti clocks, and the ordering of events in a distributed system. *Commun. ACM*, 21:558–565, July 1978.
10. J. Leguay, A. Lindgren, J. Scott, T. Friedman, and J. Crowcroft. Opportunistic content distribution in an urban setting. In *Proceedings of the 2006 SIGCOMM workshop on Challenged networks*, pages 205–212. ACM, 2006.
11. Friedmann Mattern. Virtual time and global states of distributed systems. *Workshop on Parallel and Distributed Algorithms*, 1988.
12. Michael Mitzenmacher. Compressed bloom filters. *IEEE/ACM Trans. Netw.*, 10:604–612, October 2002.
13. Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset. *CoRR*, abs/cs/0610105, 2006.
14. E O neill, V Kostakos, T Kindberg, A Schiek, A Penn, D Fraser, and T Jones. Instrumenting the city: Developing methods for observing and understanding the digital cityscape. *UbiComp 2006: Ubiquitous Computing*, pages 315–332, 2006.
15. P Ohm. SSRN-Broken Promises of Privacy: Responding to the Surprising Failure of Anonymization by Paul Ohm. *UCLA L Rev*, 2010.
16. L. Sweeney et al. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty Fuzziness and Knowledge Based Systems*, 10(5):557–570, 2002.
17. H.J.W Heerde van and N. Anciaux. Data degradation to enhance privacy for the ambient intelligence, December 2006.