Universidade Federal de Pernambuco

Graduação em Ciência da Computação

Centro de Informática

2016.1

Aplicação de desvio de relógio como fingerprint para a identificação de ponto de acesso falso

---

Trabalho de Graduação

Aluno: Emanuel Felipe dos Santos (efs4@cin.ufpe.br)

Orientador: Paulo André da Silva Gonçalves (pasg@cin.ufpe.br)

Recife, Julho de 2016

Emanuel Felipe dos Santos

Aplicação de desvio de relógio como fingerprint para a identificação de ponto de acesso falso

Trabalho de Graduação

Trabalho de Graduação apresentado à graduação em Ciência da Computação do Centro de informática da Universidade Federal de Pernambuco para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador - Paulo André da Silva Gonçalves (pasg@cin.ufpe.br)

Recife

2016

Emanuel Felipe dos Santos

Aplicação de desvio de relógio como fingerprint para a identificação de ponto de acesso falso

Trabalho de Graduação

Trabalho de Graduação apresentado à graduação em Ciência da Computação do Centro de informática da Universidade Federal de Pernambuco para a obtenção do grau de Bacharel em Ciência da Computação.

Recife, ____ de Julho de 2016.

BANCA EXAMINADORA

_____

Prof. Paulo André da Silva Gonçalves

(Orientador)

_____

Prof. Carlos André Guimarães Ferraz

(Examinador)

Agradecimentos

Agradeço aos meus pais, por terem me dado todo o suporte, ferramentas e oportunidades necessárias para que eu pudesse chegar onde estou.

A toda minha família, em especial meus irmãos, avó e sobrinhos por terem me ajudado durante toda a minha graduação, pelo orgulho que têm de mim e por me ajudarem a ser essa pessoa que sou hoje.

Ao meu orientador Paulo Gonçalves pela paciência, confiança e suporte, que tornaram este trabalho possível.

Resumo

Com o impressionante crescimento da internet, o que começou a ser experimentado a partir do final dos anos 80 e com a expansão da sua comercialização, o número de dispositivos que passaram a se conectar a internet chegou a um nível em que a atual infraestrutura de rede e seus protocolos, à princípio, não foram desenvolvidos para funcionar.

A crescente necessidade de troca de informação de maneira rápida, móvel e eficiente, como também a grande demanda por serviços ubíquos[1], fez com que as redes Wi-Fi, que são as que seguem o protocolo IEE 802.11 [3], se tornassem uma necessidade na sociedade atualmente. Porém devido a sua importância em prover um serviço ubíquo de qualidade e também pela inerente vulnerabilidade de redes sem fio, devido à sua natureza de difusão, as redes sem fio locais (WLAN) tem se tornado alvos de uma variedade de ataques [1].

Uma das formas em que as redes locais sem fio podem ser atacadas, é através da instalação de um ou mais pontos de acesso falsos. Esses pontos de acesso podem adotar as mesmas configurações, Service Set Identifier (SSID), Medium Access Control (MAC) e Basic Service Set Identifier (BSSID), do ponto de acesso (AP) original e evitar identificação utilizando diferentes características de canal físico. Assim, um usuário qualquer pode se conectar ao ponto de acesso falso sem perceber que ele não é autorizado, justamente pelo fato do ponto de acesso falso responder todas as solicitações que foram feitas da mesma forma que o ponto de acesso original responderia.

Dessa maneira, existem diversos algoritmos na literatura que utilizam mecanismos distintos para a detecção de pontos de acesso falsos [1] [2], como a Verificação de Identidade, Monitoramento de Tráfego, Tempo de Viagem de

Pacotes, Intensidade de Sinal recebido e cálculo de desvio de relógios[1]. Além disso, existe a possibilidade da localização e desativação física do ponto de acesso falso.

Palavras-chaves: Desvio de relógio, ponto de acesso, redes sem fio, detecção.

Abstract

With the impressive internet growth, that started to be experienced in the final of the 80s and with the expansion of its comercialization, the number of devices that were able to connect to the internet has arrived in a patamar that the actual internet infrastructure and its protocols weren't developed to operate.

Also, the growing need for information exchange in a quick, mobile and efficient manner, as well as the high demand for ubiquitous services, made the wireless networks, which are those that follows the IEEE 802.11 protocol, necessary in the society nowadays. But owing to its importance in providing an ubiquitous quality service and also the inherent vulnerability of wireless networks, due to its diffusion nature, wireless local area networks (WLAN) have become the target of a variety of attacks.

One way in which wireless LANs can be attacked, is by installing one or more fake access points. These fake access points can adopt the same settings, Service Set Identifier (SSID), Medium Access Control (MAC) and Basic Service Set Identifier (BSSID), as the original access point and prevent identification using different physical channel characteristics. Thus, any user can connects to the fake access point without realizing that it is not allowed, just by the fact that the fake access point will respond all the request made by the user, in the same way that the original access points would.

Thus, there are several algorithms in the literature that uses different mechanisms for the detection of fake access points, such as Identity Verification, Traffic Monitoring, Packets Time Travel, Signal Intensity and Clock Skew. Beside, there is also the posibility of locating and phisically deactivating the fake access point.

Keywords: Clock skew, access point, wireless networks, detection.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Fórmulas

1. Introdução

A crescente necessidade de troca de informação de maneira rápida, móvel e eficiente, como também a grande demanda por serviços ubíquos, fez com que as redes Wi-Fi, que são as que seguem o protocolo IEE 802.11, se tornassem uma necessidade na sociedade atualmente. Porém devido a sua importância em prover um serviço ubíquo de qualidade e também pela inerente vulnerabilidade de redes sem fio, devido à sua natureza de difusão, as redes sem fio locais (WLAN) tem se tornado alvos de uma variedade de ataques.

Uma das formas em que as redes locais sem fio podem ser atacadas, é através da instalação de um ou mais pontos de acesso falsos. Esses pontos de acessos falsos são usados para enganar os nós da rede sem fio para que o acesso a rede seja através do ponto de acesso falso ao invés do ponto de acesso autorizado. Dessa forma, o ponto de acesso falso poderá efetuar uma variedade de ataques como por exemplo, MAC-spoofing, Man-in-the-Middle, Negação de Serviço (DoS) entre outros, que são detectados em primeira instância pelo Sistema de Detecção de Intrusão em redes sem fio (WIDS).

Esses pontos de acesso falsos podem adotar as mesmas configurações, Service Set Identifier (SSID), Medium Access Control (MAC) e Basic Service Set Identifier (BSSID), do ponto de acesso (AP) original e evitar identificação utilizando diferentes características de canal físico. Assim, um usuário qualquer pode se conectar ao ponto de acesso falso sem perceber que ele não é autorizado, justamente pelo fato do ponto de acesso falso responder todas as solicitações que foram feitas da mesma forma que o ponto de acesso original responderia.

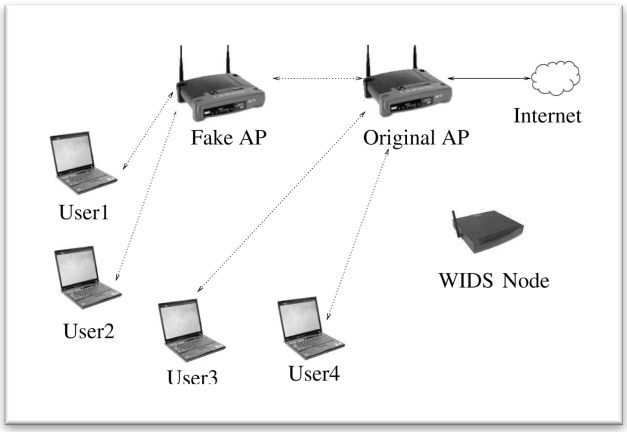
Existem, entretanto, uma diversidade de metodologias que utilizam os tradicionais métodos criptográficos para a identificação de pontos de acesso falso

através, por exemplo, de certificados e assinaturas digitais [1]. Da mesma forma, existem metodologias não-criptográficas que são utilizadas para a detecção de pontos de acesso não autorizados [22][5] estas, porém, não conseguem detectar pontos de acesso falso.

As metodologias existentes na literatura utilizam diversos mecanismos para a detecção de pontos de acesso falsos, como a Verificação de Identidade, Monitoramento de Tráfego, Tempo de Viagem de Pacotes, Intensidade de Sinal recebido e cálculo de desvio de relógios. Além disso, existe a possibilidade da localização e desativação física do ponto de acesso falso.

Vale enfatizar a proposta apresentada neste trabalho não veio substituir os métodos criptográficos atuais, mas sim propor uma forma não criptográfica de detecção de pontos de acesso falso, de tal forma que possibilite uma integração das duas metodologias para que um nível mais elevado se segurança seja alcançado nas redes sem fio. Numa situação real, Figura1, espera-se que a solução proposta seja implementada nos nós de Sistema de Detecção de Intrusões de Redes Sem fio (WIDS).

Figura 1 – Cenário de um ataque de ponto de acesso falso.



Fonte: [1]

1.1. Objetivos

O principal objetivo deste trabalho é avaliar as soluções propostas na literatura para a detecção de pontos de acesso falso e replicar com possíveis melhorias a solução proposta por [1] que utiliza a técnica de desvio de relógios. Para atingir tal objetivo foi necessário a modificação do driver da placa wireless do dispositivo utilizado nos experimentos, com a finalidade de aumentar a granularidade do *timestamp* dos quadros de *probes e beacon* e com isso a precisão do mesmo.

Essa modificação foi feita utilizando os driver do *backports* [10] para o sistema operacional Linux Mint 17,03. A solução é dividida em quatro fase, a primeira de coleta, a segunda de separação, a terceira de análise e a quarta de comparação, onde cada fase será melhor explicada no decorrer deste trabalho. Os algoritmos, modelos matemáticos e o conjunto de dados utilizados neste trabalho foram desenvolvidos em R, C++, ShellScript e podem ser encontrados em [21] e utilizaram ferramentas como o aircrack[11] e o tshark[12].

1.2. Estrutura do Trabalho

Visando uma melhor estruturação do conteúdo deste trabalho a fim de facilitar a compreensão de replicação da solução proposta por [1], foram definidos 7 capítulos.

O primeiro capítulo apresenta a motivação inicial e o contexto em que se insere o tema do trabalho, bem como explicita os objetivos propostos.

No segundo capítulo, encontra-se um referencial teórico com os conceitos básicos e termos técnicos que foram utilizados durante o trabalho, a fim de proporcionar uma melhor experiência de leitura.

No terceiro capítulo, é descrito o modelo de ameaça que foi endereçado nesse trabalho.

O quarto capítulo compreende a metodologia aplicada no cálculo dos desvios de relógio. Softwares necessários, explicação da modificação do driver, algoritmos e métodos matemáticos utilizados.

No quinto capítulo, é descrita a implementação deste trabalho, com suas distintas fases e peculiaridades.

No sexto capítulo, são mostrados os resultados.

Este trabalho é concluído no sétimo capítulo, com um resumo deste trabalho e com direções para trabalhos futuros.

2. Conceitos Gerais

Nessa sessão, serão descritos os conceitos básicos dos termos técnicos utilizados neste trabalho, cujo objetivo é proporcionar uma melhor experiência durante a leitura.

2.1. IEE 802.1

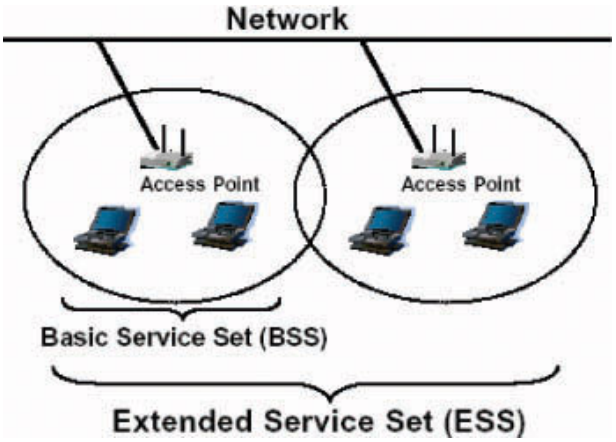
A rede sem fio IEEE 802.11, ou também como é conhecida e comercializada, Wi-Fi ou Wireless, consiste em um conjunto de especificações físicas e de Controle de Acesso ao Meio (MAC) que permite a comunicação de computadores em redes locais sem fio (WLAN) em diversas frequências de banda.

2.1.1. Arquitetura

O IEEE 802.11 defines dois tipos de serviços o *Basic Service Set* (BSS) e o *Extended Service Set* (ESS). O BSS consiste em um conjunto básico dotado de infraestrutura suficiente para fornecer serviços de comunicação entre dispositivos que podem fazer parte dele. Dentre esses dispositivos que podem fazer parte desse conjunto, estão os Pontos de Acesso (Access Point, AP) e clientes opcionais. Entretanto, quando BSSs não possuem Pontos de Acesso associado essa rede passa a ser chamada de Ad-hoc [3]

Podem ainda existir conjuntos ainda maiores, que são formados pela junção de dois ou mais BSSs, conectados por uma infraestrutura de comunicação como hubs, comutadores e roteadores. Esses conjuntos então passam a serem chamados de *Extended Service Set* (ESS). [3]

Figura 2 – Serviços BSS e ESS.



Fonte: [13]

2.1.2. Métodos de Associação

Para distinguir os BSSs é necessário um identificador, o qual é conhecido como *Service Set Identifier* (SSID). É por meio destes SSIDs que as estações poderão escolher a qual BSS elas irão se associar. A varredura que as estações fazem para tomar conhecimento de quais BSSs estão ao seu alcance em determinado momento pode ser feita de forma ativa ou passiva.

2.1.2.1. Varredura Passiva

Nesse tipo de varredura os pontos de acesso (APs) enviam periodicamente quadros de *beacon*, tipicamente de 10 a 100 quadros por segundo, contendo seus respectivos SSIDs. Como esses quadros são enviados em broadcast, qualquer estação ao alcance dos APs pode capturá-los e assim, ficar sabendo quais BSSs estão ao seu alcance. A partir dessa lista de BSSs, a estação (ou usuário) pode selecionar um deles e enviar um pedido de associação.

2.1.2.2. Varredura Ativa

Nesse tipo de varredura, a estação envia, em broadcast, quadros de solicitação de investigação. Os BSSs que recebem esses quadros respondem a ele com seus respectivos SSIDs. Assim a estação consegue a lista de redes Wi-Fi ao seu alcance.

2.1.3. Modos de operação de uma interface de rede

Uma placa de rede convencional normalmente consegue assumir diferentes papeis dependendo da maneira como está configurada [19]. Os modos de operação relevantes para este trabalho são: ativo, passivo e estação.

2.1.3.1. Modo Ativo

Em modo ativo (também conhecido como mestre, ou AP), a interface de rede se comporta como um AP convencional, fornecendo seu próprio SSID e permitindo que estações se associem a ela.

2.1.3.2. Modo Passivo.

Em modo passivo (também conhecido como modo monitor), a interface de rede pode capturar pacotes que estejam ao seu alcance, não importa quem os emitiu. É em modo passivo que ocorre a captura dos *beacons* de outros APs no protótipo implementado.

2.1.3.3. Modo Estação.

Em modo estação, a interface de rede está associada a um AP e troca pacotes com ele.

2.1.4. Tipos de Frames

O IEEE 802.11 especifica três tipos de frames para comunicação, um para dados, um para controle e o outro para gerenciamento [15][16].

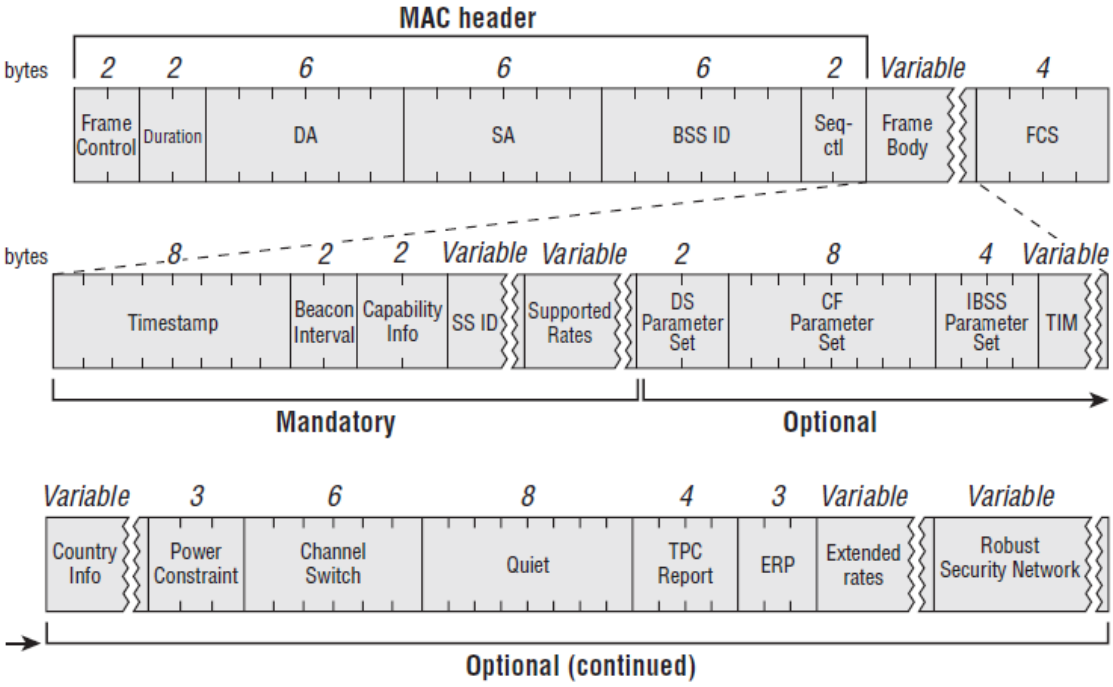
Quadro de Dados, é responsável por carregar protocolos e dados das camadas superiores dentro do corpo do quadro. Esse quadro é usado por enviar e receber dados.

Quadro de Controle, é responsável por ajudar na entrega de quadro de dados entre estações. Este tipo de quadro inclui vários subtipos, como: Quadro de Request-To-Send (RTS), Quadro de Clear-To-Sent (CTS), Quadro de Acknowledgement (ACK), etc.

Quadro de Gerenciamento, é responsável por permitir que estações estabeleçam e mantenham a comunicação. Esse tipo de quadro possui vários outros subtipos que ajudam na manutenção das Redes Locais Sem Fio (WLAN), como os Quadros de *Beacon*, Quadros de Associação, Quadros de Autenticação, Quadros de Associação, Quadros de Dissociação, etc. E são enviados durante todo o tempo e a uma taxa bem rápida independentemente da aplicação.

Nesse trabalho, nós utilizamos de forma extensiva o Quadro de Beacon para extração de informações de *Timestamp,* como será mostrado na metodologia.

Figura 3 – Quadro de Beacon



Fonte: [14]

2.1.5. Radiotap

O cabeçalho *radiotap* é o mecanismo utilizado para a obtenção de informações adicionais sobre os quadros. O cabeçalho *radiotap* provê uma flexibilidade maior do que os cabeçalhos Prism e AVS, na hora de reportar características de quadros [17] [18].

2.1.6. Tipos de Ponto de Acesso

Kim [4] classifica os APs que não são autorizados em duas categorias: os *Rogue APs* e os *Fake APs*. *Rogue APs,* são aqueles que normalmente são instalados

por usuários que buscam apenas comodidade, sem intenções maliciosas, e normalmente utilizam ethernet para se conectarem a rede. Já *Fake APs*, ou APs Falsos, são aqueles que são instalados por atacantes, com o objetivo de obter informações, falsificar mensagens ou outros tipos de ataques, e normalmente utilizam uma conexão sem fio para entrarem na rede alvo.

3. Modelo de Ameaça

A popularidade das redes locais sem fio (WLAN) baseadas no padrão IEEE 802.11, tem crescido muito nos últimos anos. Da mesma forma, tem-se observado um aumento na demanda de serviços ubíquos de qualidade e isso tem trazido uma preocupação ainda maior na segurança de tais tipos redes [6]. Um dos riscos é a instalação de pontos de acessos falso.

Existem dois cenários em que um AP falso pode operar para se passar como um AP autorizado.

No primeiro cenário, tanto o AP autorizado quanto o AP falso estão ativos ao mesmo tempo. Entretanto, como os atuais mecanismos de seleção utilizam a força do sinal como principal critério de seleção, o usuário irá selecionar o AP falso se o sinal deste for mais forte do que o do AP autorizado.

No segundo cenário, apenas o AP falso está ativo. Existem alguns fatores principais para que tal cenário ocorra, que são quando o AP falso falhou por conta própria, devido, por exemplo, a algum problema de hardware, alimentação elétrica e etc.; o usuário moveu-se e da sua atual posição apenas o sinal do AP falso é alcançável ou o atacante está efetuando um ataque de negação de serviço (DoS) no AP autorizado.

O AP falso pode adotar as mesmas configurações de um AP autorizado, incluído o BSSID, SSID e o endereço MAC, pois estamos considerando que o atacante seja poderoso o suficiente para isso. A partir de então, um usuário pode conectar-se ao AP falso sem perceber que ele não é o AP autorizado e dai todo o seu tráfego passará pelo AP falso.

4. Metodologia

Nessa sessão serão mostradas as dependências de software e hardwares existentes, como também os algoritmos e métodos matemáticos utilizados nesse trabalho.

4.1. Dependências

Os experimentos foram replicados com as seguintes dependências de software e hardware.

- Sistema operacional Linux. A distribuição do Linux utilizada neste trabalho foi o Mint, versão 17.03 Rosa.
- Versão do Kernel: 3.19.0-32-generic
- Adaptador USB Wireless TP-Link, Modelo TL- WN722N. Utilizado na fase de coleta.
- Driver backports modificado. A versão utilizada nesse projeto foi a 3.18 [10].

  Obs.: O driver do *backports* tem que ser inferior ou igual a versão do Kernel [10].

- O pacote aircrack, que contém o programa airmon-ng e airodump-ng, que serão utilizados na fase de coleta. A versão utilizada nesse projeto foi a 1.2 [11].
- O pacote tshark, que será utilizado também na fase de coleta na captura dos pacotes. A versão utilizada nesse projeto foi a 1.10.6 [12] compilado na versão 4.82 do gcc.
- IDE RStudio, como interface de desenvolvimento em R. A versão utilizada nesse projeto foi a 0.99.902 [20].
- Pacote lpSolveAPI, na versão 5.5.2.0-17. Utilizado como suporte para modelagem e resolução dos modelos matemáticos.

4.2. Modificação do Driver

A construção da réplica depende de uma modificação no protocolo IEEE 802.11 executada na placa de rede da estação que roda o programa. No entanto, cada fabricante possui um drive diferente para suas placas. A proposta que foi replicada utilizou placa Atheros.

A Atheros fornece drivers de código abertos, conhecidos como Madwifi [19], ath5k [23] e ath9k [24]. A placa do adaptador Wireless utilizado era compatível com os drivers ath5k e ath9k. Porém devido a facilidade na utilização por possuir uma biblioteca compartilhada que executa o protocolo 802.11 e que possui cópias dos drivers acima, o driver escolhido para ser modificado foi o ath9k, que foi compilado e modificado da versão 3.18 do *backports* [10].

O principal objetivo na modificação do driver é a aumentar a granularidade, ou seja, aumentar a precisão, utilizando o *timestamp* da Função de Sincronização de Relógio (TSF) para obter um resultado mais confiável de desvio de relógio.

O arquivo modificado foi o *net/mac80211/rx.c*. Ele é responsável pelo processamento de pacotes recebidos pelo protocolo 802.11. As modificações foram as seguintes:

Adicionar as seguintes declarações de variáveis dentro do método ***ieee80211_add_rx_radiotap_header()*** que começa na **linha 132**.

struct timeval timevalue;
unsigned long long int timeDay;

E substituir o código atual da **linha 186**:

put_unaligned_le64(ieee80211_calculate_rx_timestamp(local,status,mpdulen,0), pos);

Pelos seguintes comandos:

Comando 1 - *do_gettimedayof(&timevalue);*
Comando 2 - *timeDay = ((unsigned long long) timevalue.tv_sec)*1000000 + ((unsigned long long) timevalue.tv_sec);*
Comando 3 - *put_unaligned_led64(timeDay,pos);*

O primeiro comando é a chamada do sistema operacional requerida na proposta de [1]. O segundo comando faz apenas uma conversão da chamada do sistema para um inteiro de 64 bits. E finalmente, o terceiro comando coloca esse inteiro no campo do *Time Synchronization Function Timer* (TSFT) do cabeçalho *radiotap*. Vale ressaltar que as modificações de driver efetuadas não afetam de forma significativa a performance do dispositivo sem fio, uma vez que os

timestamps são armazenados apenas quando a placa está em modo monitor e o cabeçalho *radiotap* está habilitado.

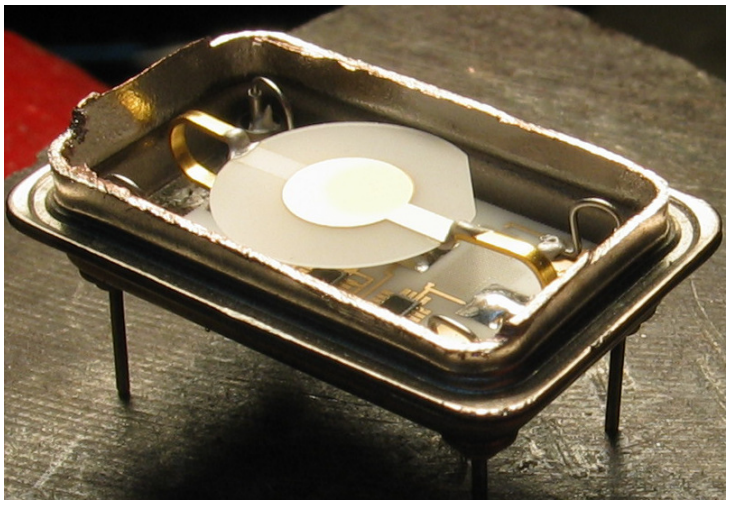
4.3. Conjunto de dados

Os quadros de frames de *probes* e de *beacon* ambos têm em sua estrutura um valor de 8 bytes que representa o *timestamp* conforme a Figura 3. O campo de *timestamp* contém o valor da TSF que é o valor do tempo que o AP envia o quadro. Os *beacons* são agendados para serem enviados periodicamente pelo AP. Vale ressaltar que os quadros de *beacon* não são afetados por nenhum atraso do meio, pois o valor do *timestamp* do *beacon* só é atribuído exatamente no momento da sua atual transmissão.

Na solução proposta por [1] e replicada neste trabalho, é utilizado o valor do *timestamp* da TSF dos quadros de *beacon/probes* para estimar o desvio de relógio de um AP e utilizar esse valor como o *fingerprint* do AP. O relógio de um AP consiste principalmente de duas partes:

- **Oscilador:** É controlado por um cristal e oscila a frequências fixas.
- **Contador:** O contador mantém o registro do número de oscilações do oscilador.

Figura 4 – Oscilador e cristal



Fonte: [25]

A frequência exata de um cristal depende primordialmente do tipo de cristal e do ângulo o qual o cristal foi cortado em relação aos seus eixos. Embora, existam dois cristais do mesmo tipo e mesmo corte, ainda assim irá existir uma pequena diferença entre eles, devido principalmente às limitações no mecanismo de corta dos cristais [26]. Esse é um dos principais motivos da existência de desvio de relógio até mesmo entre relógios similares.

Vamos agora assumir que um dispositivo *fingerprinter* (um nó WIDS) tenha recebido n quadros de *beacons* de um ponto de acesso em particular. Vamos assumir que o *timestamp* do i^{th} quadro de *beacon* seja T_i e que t_i ser o tempo em microssegundos que o dispositivo *fingerprinter* recebe o mesmo i^{th} quadro de beacon. Assumamos também que tamanho do i^{th} quadro de *beacon* enviado seja S_i e R_i seja a taxa com a qual esse quadro de *beacon* é enviado. Entretanto, o tempo de acordo com o relógio do AP, que o dispositivo *fingerprinter* recebe o i^{th} quadro de *beacon* é $T_i + S_i/R_i$. Deixe que o nosso *offset* aproximado do i^{th} quadro de *beacon* seja denotado por θ_i e a diferença de tempo entre o

primeiro e o i^{th} quadros recebidos pelo dispositivo *fingerprinter*, de acordo com seu relógio, seja denotado por x_i.

Então,

$$x_i = t_i - t_1 \quad \text{(Fórmula 1)}$$

$$\theta_i = \left(\left(T_i + \frac{S_i}{R_i}\right) - \left(T_1 + \frac{S_1}{R_1}\right)\right) - (t_i - t_1) \quad \text{(Fórmula 2)}$$

Na maioria dos casos, os quadros de *beacon* são enviados à uma taxa constante e o tamanho também permanece fixos [15]. Logo, podemos assumir que $\frac{S_i}{R_i} = \frac{S_1}{R_1}$, o que resulta em:

$$\theta_i = (T_i - T_1) - (t_i - t_1) \quad \text{(Fórmula 2.1)}$$

Nessas circunstancias, se o desvio de relógio de um dispositivo em particular permanecer constante e nós fizermos um gráfico dos pontos (x_i, θ_i), nós iremos obter aproximadamente um padrão linear. E o desvio do relógio para esse dispositivo em particular pode ser estimado como o coeficiente angular desse padrão linear.

A partir de então, passaremos a referenciar o conjunto de pontos compostos pelos pontos $\{(x_1, \theta_1), (x_2, \theta_2), \ldots, (x_n, \theta_n)\}$ como o conjunto de dados *offset* do AP.

4.4. Método da Programação Linear (MPL)

Utilizamos o método de programação linear (MPL), quando queremos otimizar algum problema de tal forma que a sua função objetivo e todas as restrições são lineares.

Iremos utilizá-lo afim de estimar o desvio de relógio de AP, a partir do conjunto de dados *offset.* Dado o conjunto de dados de *offset* do tipo $\{(x_1, \theta_1), (x_2, \theta_2), \dots, (x_n, \theta_n)\}$, o método da programação linear encontra uma linha $\delta x + \phi$, que é o limite superior dos pontos que compõem o conjunto de dados *offset,* tal que, δ é a inclinação desta linha e ϕ é o ponto em que esta linha intercepta o eixo Y, de tal forma que obedeça as seguintes restrições:

$$\delta . x_i + \phi \geq \theta_i , \forall i = 1,2, \dots, n. \qquad \text{(Fórmula 3)}$$

E a seguinte função objetivo é minimizada.

$$\frac{1}{n} \sum_{i=1}^{n} (\delta . x_i + \phi - \theta_i)$$

(Fórmula 4)

Este problema pode ser resolvido utilizando métodos de programação linear para duas variáveis.

O método de programação linear (MPL) minimiza quaisquer atrasos inesperados já que ele tem uma alta tolerância à *outliers.* O desvio de relógio, na maioria das vezes, permanece estável mesmo quando existe um número significante de *outliers.*

Entretanto, devido a natureza do MPL à baixa tolerância a *outliers* alguns problemas sérios de segurança podem acontecer, pelo menos no nosso contexto. Por exemplo, se um adversário tiver o poder necessário de misturar uma pequena quantidade de quadros de *beacon* de um AP falso com os quadros de *beacon* do AP original e os desvios de relógios dos dois APs forem próximos, então o MPL poderá considerar que os frames do AP falso que foram injetados são *outliers* e calcular de forma normal o desvio de relógio do AP original. Neste caso, será difícil detectar pontos de acesso falso apenas comparando os desvios de relógio.

4.5. Método dos Mínimos Quadrados (MMQ)

Utilizamos o método dos mínimos quadrados (MMQ) quando queremos ajustar uma curva a um conjunto de dados.

Iremos utilizá-lo afim de estimar o desvio de relógio de um AP, a partir do conjunto de dados *offset.* Dado o conjunto de dados de *offset* do tipo $\{(x_1, \theta_1), (x_2, \theta_2), \dots, (x_n, \theta_n)\}$, o método dos mínimos quadrados encontra uma linha $\delta x + \phi$, tal que, δ é a inclinação desta linha e ϕ é o ponto em que esta linha intercepta o eixo Y, de tal forma que:

$$\sum_{i=1}^{n} (\theta_i - (\delta . x_i + \phi))^2$$

(Fórmula 5)

permaneça mínimo. A inclinação desta reta, δ, é estimada como o desvio de relógio do conjunto de dados *offset.*

Com o conjunto de dados de *offset* definido, um método mais direto de encontrar a inclinação, δ, e também o ponto de intercessão desta reta com o eixo Y, ϕ, de tal forma que o somatório da Fórmula 5 permaneça mínimo, podem ser feitos da seguinte forma:

$$\delta = \frac{\sum x\theta \; - \; \frac{\sum x \; \sum \theta}{n}}{\sum x^2 \; - \; \frac{(\sum x)^2}{n}}$$

(Fórmula 6)

$$\phi = \frac{\sum \theta - (\delta . \sum x)}{n}$$

(Fórmula 7)

Uma das principais diferenças entre o Método dos Mínimos Quadrados (MMQ) em relação ao Método de Programação Linear (MPL), é a baixa tolerância a *outliers* como mostraremos na sessão de resultados.

Mesmo quando existe uma baixa quantidade de *outliers* o desvio de relógio estimados com MMQ irá variar consideravelmente do desvio de relógio determinado pela maioria dos pontos. Isso pode causar problemas no momento de estimar o desvio de relógio para dados muito ruidosos [7]. Entretanto, ainda poderemos utilizar o MMQ para estimarmos de forma eficiente o desvio de relógio. Pois o MMQ tem uma vantagem em relação ao MPL que é justamente essa alta sensibilidade à até mesmo um número muito baixo de *outliers,* o que dificultará para um intruso a injeção imperceptível de frames do ponto de acesso falso quando o MMQ é utilizado para a estimativa de desvio de relógio.

4.6. Diferenciando Quadros de APs Falsos

Um dos passos primordiais para podermos diferenciar os quadros de AP autorizado dos quadros de AP falso é a separação dos dados. Essa separação nos ajuda também a compreender mais sobre os pontos de acesso falsos.

O problema de separação de dados em seus respectivos subconjuntos não é novo. Existem atualmente vários algoritmos como, por exemplo, o GHT [8] e o EM [9] que são capazes de fazer a separação de dados de forma confiável, porém são algoritmos complexos e necessitam de um poder computacional, tanto de processamento quanto de armazenamento muito elevado, o que não se encaixou na metodologia proposta por [1]. Dessa forma, [1] propôs uma heurística que é muito menos complexa e mais leve e que consiste em que quadros de *beacon* recebidos de AP diferentes irão conter certos *pulos* (i.e. diferenças abrutas nos valores) nas bordas onde um pacote é de um AP e o pacote sucessor é de outro AP. A heurística proposta identifica esses *pulos* e diferencia o conjunto de dados baseado neles.

Inicialmente, iremos introduzir o *desvio relativo* entre pacotes que será calculado de acordo com a Fórmula 8; em seguida introduziremos o parâmetro *threshold* que é a métrica que diferenciará entre pulos, cujo cálculo será mostrado no Algoritmo 1 e por fim explicaremos, no Algoritmo 2, como o uso do *threshold* é utilizado para a separação dos dados.

Seja Δ_{ij} o desvio relativo entre dois pacotes do conjunto de offset, (x_i, θ_i) e (x_j, θ_j). Logo, Δ_{ij}, é definido da seguinte maneira:

$$\Delta_{ij} = \frac{|\theta_i - \theta_j|}{|x_i - x_j|}$$

(Fórmula 8)

Onde, $\Delta_{ij} > 0$, pois $|\theta_i - \theta_j|$ e $|x_i - x_j|$ são valores absolutos. Deste modo, dois pontos consecutivos $(x_i, \theta_i)\ e\ (x_j, \theta_j)$ são considerados um *pulo* apenas se $\Delta_{ij}, >$ *threshold.*

O algoritmo que utilizamos para o cálculo do *threshold* é o Algoritmo 1, mostrado a seguir:

***Algoritmo 1**: Cálculo do **threshold** para conjunto de dados offset*

```
threshold_final = 0
for each data_set do
        threshold = Δ12
        for i = 3 to n do
                if Δi(i−1) > threshold then
                        threshold = Δi(i−1)
                end if
        end for
        if threshold ≥ threshold_final then
                threshold_final = threshold
        end if
end for
print(threshold_final)
```

Com o valor do *threshold* calculado do algoritmo anterior, iremos então utilizar esse valor para separarmos o conjunto de dados completos em conjunto de dados correspondentes a cada AP originário, conforme o algoritmo a seguir:

***Algoritmo 2**: Separação do conjunto de dados offset baseado no AP originário.*

---

accumulator[0].dataset = [(x_1, y_1)]
accumulator [0].current_point = (x_1, y_1)
accumulator [0].current_offset = 1
accumulator [0].count = 1
for** i = 2 **to** n **do
    for each** entry j in accumulator **do
        k = accumulator[j].current_offset
        if** $\Delta_{ik} \leq$ threshold **then
            add (x_i, y_i) to data set of accumulator entry j
            accumulator [j].count = accumulator [j].count + 1
            accumulator [j].current_point = (x_i, y_i)
            accumulator [j].current_offset = i
        end if
    end for
    if** none of the entry in accumulator satisfies ($\Delta_{ik} \leq$ threshold) **then
        add a new accumulator entry p
        p.dataset = [(x_i, y_i)]
        p.count = 1
        p.current_point = (x_i, y_i)
        p.current_offset = i
    end if
end for
print(**accumulator.size**)
print(**accumulator**)

---

5. Implementação

Nós implementamos nossa metodologia para a captura, armazenamento e cálculo dos desvios de relógio em um único computador – um Lenovo Thinkpad T440 rodando o Linux Mint 17.3 Rosa. E apenas um dispositivo de wireless, o adaptador wireless TP-Link de Modelo TL-WN722N. Não utilizamos a placa de rede wireless do Thinkpad, uma Intel Wireless 7260, pois não encontramos a parte específica do código *open source* que deveríamos modificar para replicarmos a proposta apresentada neste trabalho. Entretanto, utilizamos a TL-WN722N, pois ela suporta o modo monitor e também por seu driver ser *open source* e termos conseguido modificar o driver com sucesso.

O sucesso da nossa metodologia está intimamente ligado ao quão bem as fases a seguir são implementadas. Dividimos nosso trabalho em 4 fases: Coleta, Separação, Análise e Comparação, as quais serão explicadas com mais detalhes a seguir.

5.1. Coleta

Durante a fase de coleta, a interface de rede é colocada em modo monitor e começa a capturar *beacons* dos APs a serem analisados. *Beacons*, como dito anteriormente, são quadros emitidos por pontos de acessos utilizados para associação entre pontos de acesso e estações[3]. Esses *beacons* são capturados pela versão modificada do driver de rede descrito na sessão anterior. Essa modificação faz com que o campo do *timestamp* da TSFT do cabeçalho *Radiotap* presente nos *beacons* capturados tome o valor retornado pela chamada do sistema *do_gettimeofday* no instante que o pacote é recebido, ao invés de tomar o valor comum da TSFT da estação que recebeu o pacote.

5.2. Separação

Após a captura dos *beacons*, é realizada a separação dos dados. Essa separação é necessária, pois o atacante pode mascarar seu SSID, MAC, fazendo com que os *beacons* emitidos tanto pelo AP falso quanto pelo AP original sejam coletados como se pertencessem ao mesmo AP. Durante a fase de separação, é possível perceber a existência de mais de um AP a partir do conjunto de dados.

5.3. Análise

Com o conjunto de dados separados em vários subconjuntos, cada um pertencente a um AP diferente, inicia-se a fase de análise dos dados. Cada subconjunto é submetido a um procedimento que calcula o desvio do relógio daquele subconjunto específico. Jana [1] utiliza os modelos matemáticos descritos na sessão anterior, que é Método dos Mínimos Quadrados e o Método da Programação Linear para traçar uma reta representativa do conjunto de dados. A inclinação dessa reta determina o valor estimado do desvio do relógio.

5.4. Comparação

Com os desvios dos subconjuntos já calculados, e também de posse do desvio previamente conhecido do ponto de acesso original, é possível comparar ambos e determinar se o ponto de acesso que está sendo avaliado é de fato o original.

6. Resultados

Todos os dados experimentais utilizados nesse trabalho, para testar nossa metodologia na detecção de pontos de acesso não autorizados e falsos foram coletados em três ambientes residenciais distintos, nos quais múltiplos APs operavam simultaneamente.

No primeiro ambiente residencial (Ambiente Residencial A) tinha 12 APs, dos quais 5 tínhamos total controle sobre. No segundo ambiente residencial (Ambiente Residencial B) tinha 6 APs, e no terceiro ambiente residencial (Ambiente Residencial C) tinha 9 APs. Nos ambientes residenciais B e C não temos controle total sobre nenhum ponto de acesso. Os dados coletados no ambiente residencial A, foram coletados em dias variados cujo objetivo foi verificar a consistência dos desvios de relógio com o tempo. Enquanto que os dados coletados nos ambientes residenciais B e C foram coletados no mesmo dia. Como não tínhamos controle sobre todos pontos de acessos, o fabricante do mesmo foi previstos, quando possível, através dos 3 primeiros bytes do endereço MAC.

A medida utilizada nos gráficos desta sessão é a partes por milhão, essencialmente μs/s, denotada por *ppm*, para quantificar os desvios de relógio. Descreveremos os resultados dos nossos experimentos nos ambientes residenciais nas subseções seguintes.

6.1. Resultados ambientes residenciais

Inicialmente, no ambiente residencial A, o qual temos maior controle sobre os APs, fizemos uma análise mais elaborada e detalhada. Foram feitas análises em dias e horários distintos afim de verificar se o desvio de relógio permanecia consistente com o tempo. Fizemos, também, uma análise para verificar a

velocidade com o que o nosso experimento iria convergir, isto é, qual a quantidade mínima de pacotes que precisaríamos analisar para termos um desvio de relógio aproximado. O resultado da estimativa do desvio de relógio baseada na quantidade de pacotes pode ser visualizado a seguir, na Tabela 1.

AP	Sagemcom1		Dlink		HonHaiPr		Bewan		Sagemcom2	
# Pacotes Examinados	MPL	MMQ	MPL	MMQ	LPM	MMQ	MPL	MMQ	MPL	MMQ
100	10.31	10.04	-17.23	-15.42	-6.97	-12.17	0.00	0.53	-14.64	-7.9
200	8.96	9.69	-23.14	-22.12	-16.85	-11.72	0.00	0.51	-6.11	7.12
300	9.59	9.78	-26.37	-23.49	-16.85	-11.85	0.78	1.07	-6.83	-6.99
400	9.82	9.82	-26.37	-23.95	-13.26	-11.91	0.78	0.87	-6.87	-7.00
500	9.80	9.82	-24.54	-24.63	-13.26	-11.87	0.82	0.85	-6.87	-7.04
600	9.79	9.82	-24.54	-24.11	-11.10	-11.87	0.82	0.82	-6.87	-7.07
1000	9.78	9.39	-24.13	-23.48	-10.91	-11.81	0.75	0.85	-6.88	-6.98

Tabela 1. Estimativa de desvio de relógio de vários pontos de acessos com diferentes tamanhos de amostra.

Como podemos ver na Tabela 1, o número médio mínimo de pacotes necessários para começarmos a convergir para uma estimativa do desvio de relógio, é de 300 pacotes, utilizando o MPL, enquanto que utilizando o MMQ a conversão inicia-se por volta dos 400 pacotes.

Para verificarmos se o desvio de relógio permanece constante com o tempo utilizando o MPL, nós pegamos 50 amostras aleatórias, cada uma com 300 pacotes de uma coleta de 10000 pacotes do AP *Sagemcom1* e calculamos o desvio de relógio para cada uma. O mesmo procedimento foi feito também para o MMQ, só que utilizando 400 pacotes e o resultado obtido nos dois experimentos pode ser observada na Figura 5 a seguir.

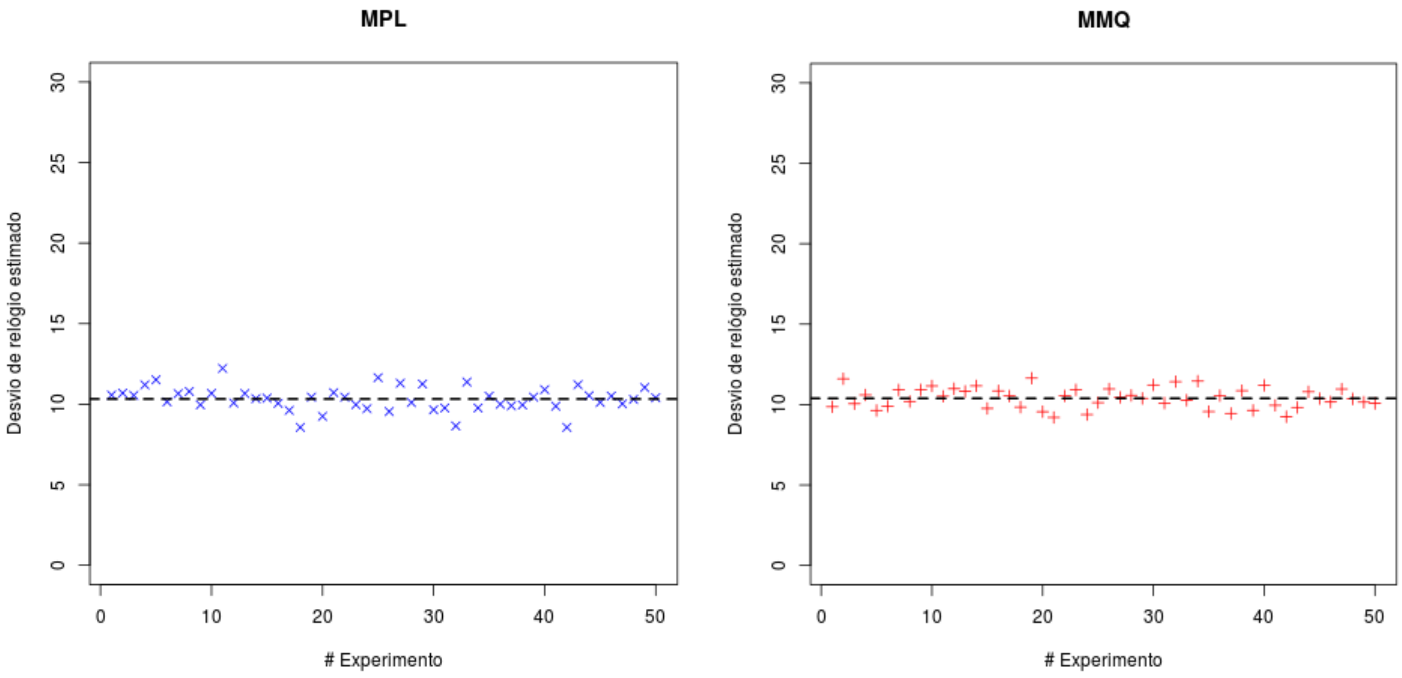


Figura 5 – Estimativa do desvio de relógio de 50 amostras para MPL e MMQ.

Nós encontramos que o desvio de relógio estimado para todas as amostras, permanecem muito próximos de 10.32 (MPL) e 10.38 (MMQ), que é a estimativa do desvio de relógio de toda a amostra que contém os 10000 pacotes, sinalizada pela linha pontilhada.

Para mostrarmos que os valores dos desvios de relógio permanecem constante com o tempo, fizemos uma análise estendida e em dias e horários distintos afim de demonstrarmos tal fato. A Tabela 2, mostra dados de uma análise de 5 medidas, nas quais cada medida representa um dia distinto.

	Medida 1		Medida 2		Medida 3		Medida 4		Medida 5	
AP	**MPL**	**MMQ**	**MPL**	**MMQ**	**MPL**	**MMQ**	**MPL**	**MMQ**	**MPL**	**MMQ**
Dlink	-24.13	-23.48	-22.54	-23.57	-24.59	-23.53	-23.78	-23.09	-23.81	-23.70
Sagemcom1	9.78	9.39	9.68	9.67	9.55	9.49	9.84	9.82	9.46	9.75
HonHaiPr	-10.91	-11.81	-12.28	-11.82	-11.92	-11.52	-11.77	-12.23	-12.10	-11.92
Bewan	0.75	0.85	0.91	0.81	1.24	0.87	1.59	1.68	1.50	1.39
Sagemcom2	-7.10	-6.98	-6.89	-6.96	-7.08	-7.17	-7.52	-7.24	-6.92	-7.21

Tabela 2. Estimativa de desvio de relógio em dias distintos de vários pontos de acesso.

Utilizamos os gráficos a seguir para suportar nosso resultado de que os desvios de relógio são diferentes para cada AP e que o padrão linear previsto se mantém com o tempo. A Figura 6 mostra o comportamento do desvio de relógio calculado com MPL para cada AP individualmente, enquanto que a Figura 7, mostra o comportamento do desvio de relógio calculado com MMQ para cada AP individualmente.

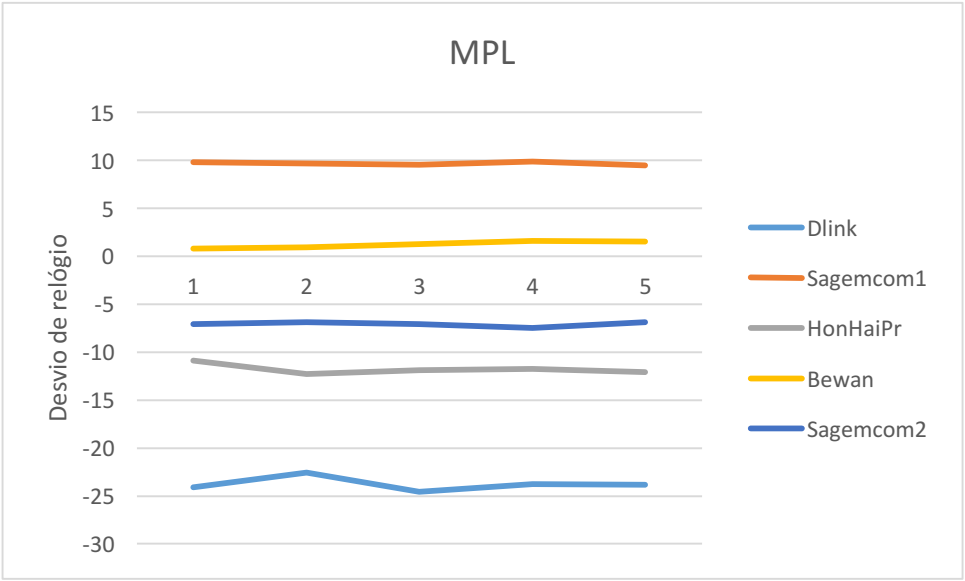


Figura 6 – Comportamento do desvio de relógio calculado com MPL.

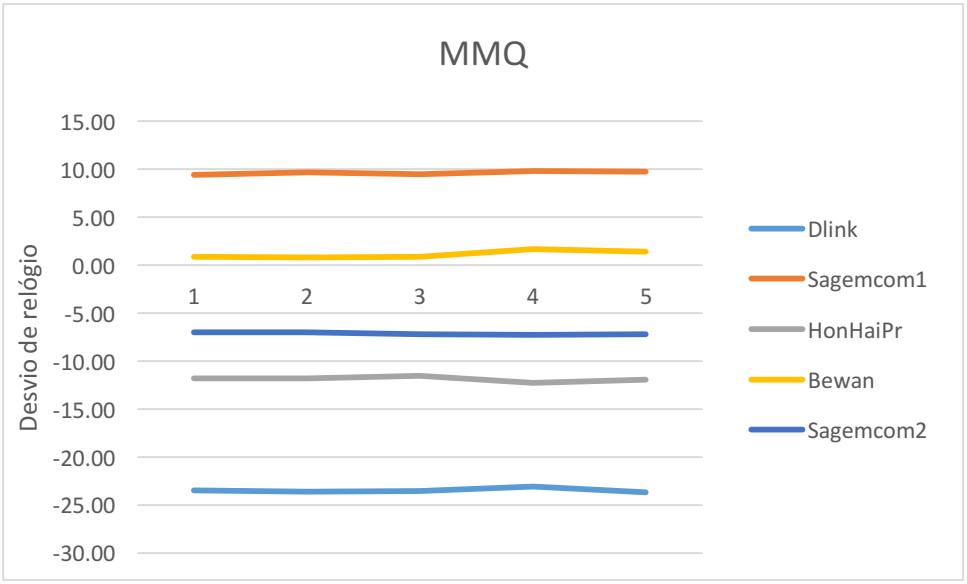


Figura 8 – Comportamento do desvio de relógio calculado com MMQ.

Por fim, fizemos uma análise de todos os APs alcançáveis no ambiente residencial A, no qual só foi possível a coleta de *beacons* de 12 APs. Fizemos essa análise para verificarmos se o desvio de relógio entre APs é diferente e se o desvio de relógio tende a convergir, como mostra a Tabela 3.

	300 Pacotes		1000 Pacotes	
AP	**MPL**	**MMQ**	**MPL**	**MMQ**
Sagemcom	9.07	10.08	9.83	9.90
Dlink	-23.3	-22.64	-20.01	-22.9
HonHaiPr	-14.16	-12.46	-13.29	-12.72
Bewan1	0.68	0.93	0.94	0.99
Sagemcom2	0	-6.63	-3.68	-6.43
Dlink2	-0.92	-0.90	-0.85	-0.88
Sagemcom3	0.84	1.33	1.06	1.00
Sagemcom3	8.40	8.16	8.10	8.13
Unknown3	1.48	1.26	1.39	1.4
Unknown4	12.30	11.78	11.65	11.64
Unknown1	0	-1.18	-3.44	1.37
Unknown2	6.01	6.25	6.42	6.38

Tabela 3 – Estimativa do desvio de relógio no ambiente residencial A.

A mesma análise utilizada para a Tabela 3, foi feita para os ambientes residenciais B e C, os quais não tínhamos controle sobre os pontos de acessos. A análise mais uma vez foi feita para verificação se Aps distintos têm desvios de relógio distintos e se tendem a convergir.

	300 Pacotes		1000 Pacotes	
AP	**MPL**	**MMQ**	**MPL**	**MMQ**
Unknown1	1.61	1.95	1.59	1.65
Unknown2	5.68	5.44	5.47	5.43
Unknown3	1.47	1.46	1.58	1.54
Unknown4	4.28	4.42	4.40	4.36
Palladiu	575.60	576.62	575.90	576.01
Asustek	0.00	0.43	0.00	0.47

Tabela 4 – Estimativa do desvio de relógio no ambiente residencial B.

	300 Pacotes		1000 Pacotes	
AP	**MPL**	**MMQ**	**MPL**	**MMQ**
Unknown1	-3.74	-3.77	-3.88	-3.87
Unknown2	2.62	2.08	2.08	2.24
Unknown3	5.44	4.97	4.77	4l79
Sagemcom1	5.11	4.88	5.12	5.06
TpLink1	2.97	2.99	2.67	2.81
Zte1	-9.09	-3.93	-3.82	-4.08
Cisco1	0.00	0.73	0.00	-0.39
TpLink2	1229.83	1230.07	1230.01	1230.03
Unknown4	-5.8	-2.51	-3.53	-2.43

Tabela 5 – Estimativa do desvio de relógio no ambiente residencial C.

Nessa sessão conseguimos observar que com o aumento da resolução de tempo para microssegundo do dispositivo *fingerprinter*, que utilizamos na nossa metodologia, foi possível notar uma grande melhoria em relação a um dispositivo

fingerprinter com resolução de tempo em milissegundos proposta por [7]. A metodologia abordada neste trabalho, tem uma melhoria significativa em relação a de [7], que utiliza resolução em milissegundo e leva em média de 30 a 60 minutos para o desvio de relógio convergir, enquanto que com uma resolução de tempo em microssegundos, e com uma coleta de cerca de 2 a 3 minutos é possível estimar com precisão o desvio do relógio. Isso faz que a estimativa do desvio de relógio, pelo menos em ambientes WLAN controlados e sem muito ruído seja de 15 a 20 vezes mais rápida.

6.2. Resultado quadros de APs falsos

Para simularmos um cenário de um real ataque no qual o AP autorizado e o AP falso estão ativos ao mesmo tempo, nós construímos um cenário em que foi possível simular uma coleta real de vários conjuntos de dados capturados de vários APs operando simultaneamente. O cenário foi simulado no ambiente residencial A onde temos total controle sobre grande parte dos APs. Ou seja, foi possível manipular o endereço MAC, SSID e também temos conhecimentos prévios dos desvios de relógio de cada AP individualmente, com também a quantidade de APs simultâneos que estão sendo utilizados.

Este teste, mostrado na Tabela 6, foi importante para analisarmos o entrelaçamento de dados no nosso método de estimativa como também a eficiência do algoritmo de separação.

Caso	Conjuntos	Desvio original MPL	Desvio original MMQ	Desvio estimado MPL	Desvio estimado MMQ	Conjuntos Estimados
1	2	(9.66, -23.77)	(9.62, -23.77)	10.45	460862.1722	2
2	2	(1.19, 9.66)	(1.12, 9.62)	-3.30	5540888.668	2
3	2	(-7.10, 9.66)	(-7.11, 9.62)	0	-151611692.3	2
4	2	(-23.77, 1.19)	(-23.47, 1.12)	-22.00	42313887.58	2
5	3	(1.19, 9.66 , -7.10)	(1.12, 9.62, -7.11)	10.67	-23028310	3
6	2	(1.19, -7.10)	(1.12, -7.11)	1.40	231335.0128	2
7	2	(-22.54, -7.10)	(-23.57, -7.11)	-23.07	-5698984.819	2
8	3	(-23.78, -7.10, 1.19)	(-23.09, -7.11, 1.12)	-19.53	117232253.7	3
9	4	(-24.13, 1.19, -7.10, 9.66)	(-23.48, 1.12, -7.11, 9.62)	-19.32	-537434.5448	4

Tabela 6 – Medidas da simulação de um ataque real.

A tabela 6 mostra que em alguns casos (i.e. casos 1, 4, 5, 6, 7) o valor do desvio de relógio utilizando o MPL é muito próximo ao desvio de um dos pontos de acesso o quais os pacotes estão misturados. Vale ressaltar também que os valores obtidos utilizando o MMQ são extremamente altos devido a mistura de pacotes existente, o que nos ajuda a identifica a presença de um AP falso muito mais rápido do que no MPL.

O resultado dessa análise sugere que quando estamos utilizando apenas o MPL nós podemos nos confundir no momento de identificarmos um AP falso que está operando ao mesmo tempo que o AP original. Isso suporta a ideia de que não podemos utilizar apenas o MPL para identificarmos APs falsos. Por outro lado, o desvio de relógio utilizando o MMQ são extremamente grandes e distintos dos atuais desvios de relógios dos APs individuais, dessa forma apenas observando

esse valor podemos concluir que pelo menos um AP falso está ativo simultaneamente ao AP original.

Nós aplicamos o algoritmo de separação (Algoritmo 2) e como podemos verificar na Tabela 6, a estimativa da quantidade de APs operando simultaneamente foi 100% correta.

7. Conclusões e Trabalhos Futuros

Em suma, devido à inerente vulnerabilidade das redes sem fio, e devido à sua natureza de difusão, as redes sem fio locais têm se tornado cada vez mais alvos de uma variedade de ataque. Umas das várias formas que uma rede local sem fio pode ser atacada, é a través da instalação de APs falsos. Esses APs falsos comportam-se da mesma forma que um AP original e podem ter total controle sobre os dados trafegados por ele.

Entretanto, a metodologia aplicada neste trabalho se mostrou um método eficiente e robusto na detecção de pontos de acesso não autorizados em redes locais sem fio. A metodologia aqui aplicada, consistiu em replicar a solução proposta por Jana [1], cuja principal contribuição foi propor o aumento da precisão para microssegundos do *timestamp* da TSFT do quadro de *beacon* afim de calcular de forma mais precisa e eficiente o desvio de relógio de um AP.

A implementação da metodologia foi dividida em 4 fases, as quais contêm os modelos matemáticos, heurísticas e algoritmos que suportam os resultados apresentados. Nós validamos a metodologia apresentada neste trabalho utilizando APs de 3 ambientes residenciais distintos, um dos quais tínhamos total controle sobre o que tornou possível a simulação de cenário de ataque real com vários APs operando simultaneamente.

O tema em questão é relativamente recente [1][7] e devido grande demanda por serviços ubíquos e difusão das redes sem fio, a necessidade de um passo a mais na cadeia de segurança passa a ser essencial.

Como trabalho futuro, pretende-se realizar uma análise voltada a dispositivos móveis. Coleta de mais dados e novos cálculos em ambientes diferentes, como por exemplo, sistemas operacionais distintos; coleta de dados em laptops sem fornecimento direto de energia, funcionando apenas pela bateria; proposta de novos modelos matemáticos mais precisos e eficientes a ponto de ser possível o desenvolvimento de aplicativos mobile capaz de detectar AP falsos.

8. Referências

[1] Jana, S.; Kasera, S.K., **On Fast and Accurate Detection of Unauthorized Wireless Access Points Using Clock Skews,** Mobile Computing, IEEE Transactions on, vol.9, no.3, pp.449,462, March 2010.

[2] Arackaparambil, Chrisil; Bratus, Sergey; Shubina, Anna; and Kotz, David. 2010. **On the reliability of wireless fingerprinting using clock skews**. In Proceedings of the third ACM conference on Wireless network security (WiSec '10). ACM, New York, NY, USA, 169-174.

[3] Kurose, James F; Ross, Keith W., **Redes de computadores e a Internet: uma abordagem top-down, 5. ed.**, São Paulo : Addison Wesley, 2010.

[4] Taebeom Kim; Haemin Park; Hyunchul Jung; Heejo Lee,. **Online Detection of Fake Access Points Using Received Signal Strengths**, Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th , pp.1,5, 6-9 May 2012

[5] **Rogue Access Point Detection: Automatically Detect and Manage Wireless Threats to Your Network**, http://www.proxim.com, 2009.

[6] Loureiro, A. A. F.; Oliveira, R. A. R.; Moura, T. R. de; Júnior, W. R. P.; Oliveira, L. B. R. de; Moreira, R. A.; Siqueira, R. G.; Rocha, B. P. S.; Ruiz, L. B. **Computação Ubíqua Ciente de Contexto: Desafios e Tendências.** In: Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC 2009 - Minicurso, 2009. p.99–149.

[7] T. Kohno, A. Broido and K.C. Claffy, **Remote Physical Device Fingerprinting**, IEEE Trans. Dependable Secure Computing, vol. 2, no. 2, pp. 93-108, Apr.-June 2005.

[8] P. Hough, **Method and Means for Recognizing Complex Patterns,** US Patent 3069654, 1962.

[9] A.P. Dempster, N.M. Laird and D.B. Rubin,. **Maximum Likelihood from**

Incomplete Data via the EM Algorithm. J. Royal Statistical Soc., vol. 39, no. 1, pp. 1-38, 1977.

[10] **Backports**. Disponível em:
< https://backports.wiki.kernel.org/index.php/Main_Page >. Acesso em 18 Abril de 2016.

[11] **Aircrack**. Disponível em: < http://www.aircrack-ng.org/ >. Acesso em 19 Abril de 2016.

[12] Wireshark, **Tshark.** Disponível em: < https://wiki.wireshark.org/ >. Acesso em 19 Abril de 2016.

[13] **Flylib**. Disponível em: <http://flylib.com >. Acesso em 28 Março de 2016.

[14] Rasika Nayanajith, **Beacon Frame.** Disponível em: <http://mrncciew.com/2014/10/08/802-11-mgmt-beacon-frame/>. Acesso em 28 Março de 2016.

[15] IEEE Standard 802.11.,**Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications**, The Institute of Electrical and Electronics Engineers, Inc., 1999.

[16] **Understanding 802.11 Frame Types.** Disponível em: <http://www.wi-fiplanet.com/tutorials/article.php/1447501/Understanding-80211-Frame-Types.htm> Acesso em 7 de Abril de 2016.

[17] **Radiotap**. Disponível em: <http://www.radiotap.org>. Acesso em 16 de Abril de 2016.

[18] Linux Wireless, **Radiotap**. Disponível em: <http://linuxwireless.org/en/developers/Documentation/radiotap/>. Acesso em 16 de Abril de 2016.

[19] **MadWifi.** Disponível em: <http://madwifi-project.org/wiki/About/MadWifi>. Acesso em 10 de Março de 2016.

[20] **RStudio.** Disponível em < https://www.rstudio.com/>. Acesso em 2 de Maio

de 2016.

[21] Emanuel Santos, **TG**. Disponível em < https://github.com/emanuelfs4>. Acesso em 26 de Maio de 2016.

[22] **NetStumbler**, Disponível em < http://www.netstumbler.com>. Acesso 15 de Abril de 2016.

[23] Linux Wireless, **About Ath5k**. Disponível em < https://wireless.wiki.kernel.org/en/users/Drivers/ath5k >. Acesso em 9 de Março de 2016.

[24] Linux Wireless, **About Ath9k**. Disponível em < https://wireless.wiki.kernel.org/en/users/Drivers/ath9k >. Acesso em 9 de Março de 2016.

[25] HiFiDuino. **Oscillator**. Disponível em < https://hifiduino.files.wordpress.com>. Acesso em 15 de Maio de 2016.

[26] IEEE Guide for Measurement of Environmental Sensitivities of Standard Frequency Generators, **IEEE Standards Coordinating Committee 27-SCC27-on Time and Frequency**, 1995.