



UNIVERSIDADE FEDERAL DO CEARÁ  
CAMPUS QUIXADÁ BACHARELADO EM  
SISTEMAS DE INFORMAÇÃO

**ANTONIO GUILHERME ESTEVÃO SOARES FEITOSA**

**COLETOR DE DADOS PARA MONITORAMENTO AÉREO USANDO ADS-B E  
MINI PC ANDROID**

**QUIXADÁ  
2016**

**ANTONIO GUILHERME ESTEVÃO SOARES FEITOSA**

**COLETOR DE DADOS PARA MONITORAMENTO AÉREO USANDO ADS-B E  
MINI PC ANDROID**

Monografia apresentada ao Curso de Sistemas de Informação da Universidade Federal do Ceará como requisito parcial para obtenção do Título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Arthur de Castro Callado.

**QUIXADÁ  
2016**

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca do Campus de Quixadá

---

F336c      Feitosa, Antônio Guilherme Estevão Soares  
              Coletor de dados para monitoramento aéreo usando ADS-B e mini PC Android / Antônio  
              Guilherme Estevão Soares Feitosa. – 2016.  
              37 f. : il. color., enc.; 30 cm.

              Monografia (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de  
              Bacharelado em Sistemas de Informação, Quixadá, 2016.  
              Orientação: Prof. Dr. Arthur de Castro Callado  
              Área de concentração: Computação

1. Android (Programade Computador). 2. Tráfego aéreo. 3. Vigilância. I. Título.

---

CDD 005.432

**ANTONIO GUILHERME ESTEVÃO SOARES FEITOSA**

**COLETOR DE DADOS PARA MONITORAMENTO AÉREO USANDO ADS-B E  
MINI PC ANDROID**

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel. Área de concentração: computação

Aprovado em: \_\_\_\_\_ / fevereiro / 2016.

BANCA EXAMINADORA

---

Prof. Dr. Arthur de Castro Callado (Orientador)  
Universidade Federal do Ceará-UFC

---

Prof. Dr. Miguel Franklin de Castro  
Universidade Federal do Ceará-UFC

---

Prof. Dr. Márcio Espíndola Freire Maia  
Universidade Federal do Ceará-UFC

A minha família...

Que não mediu esforços para fazer chegar onde estou

## **AGRADECIMENTOS**

Primeiramente a Deus, pela dádiva da vida e pela coragem, força e sabedorias adquiridas até aqui.

Aos meus pais, avós e tios por me apoiarem em todos esses anos de estudos.

Aos meus orientadores, prof. Enyo José, prof. Marcos de Oliveira, prof. David Romero e especialmente ao prof. Arthur Callado que me acolheu nesse último ano de graduação, pelos ensinamentos, puxões de orelha e conselhos dados, contribuindo com meu amadurecimento acadêmico, profissional e pessoal.

A todos os meus professores que contribuíram com minha formação desde o ensino fundamental até hoje. E especialmente ao prof. Ricardo Reis que compartilhou comigo suas experiências como aluno deficiente visual, e assim como eu, passou por muitas dificuldades em sala de aula, mostrando que compartilhamos da mesma vontade de aprender e que com muito esforço e dedicação é possível ir longe, apesar dos obstáculos.

Aos meus colegas de repúblicas, pelas alegrias, tristezas e contas compartilhadas, noites em claro estudando, fazendo trabalhos, ou simplesmente farreando.

"Se você insiste em permitir que pessoas negativas enfraqueçam seus sonhos, você não merece realizá-los. É sua a responsabilidade de selecionar a quem dá ouvidos."

(Flávio Augusto)

## RESUMO

Eficiência é uma questão fundamental em sistemas que necessitam captar e transferir dados em alta velocidade. Neste trabalho foi desenvolvido um aplicativo para ser executado em uma plataforma embarcada com sistema operacional Android, para fazer a coleta de mensagens aéreas compatíveis com a tecnologia *Automatic Dependent Surveillance - Broadcast* (ADS-B). Com o objetivo de avaliar essa união de hardware embarcado e software, foi realizada uma avaliação de desempenho com outra plataforma, porém essa de propósito geral. Um conjunto de métricas foi utilizado nessa avaliação e intervalos de confiança foram utilizados para validar os resultados obtidos.

**Palavras-chave:** Tráfego aéreo. Android. Avaliação de desempenho.



## **ABSTRACT**

Efficiency is a key issue in systems that need to catch and transfer data at high speed. In this work, an application was developed to run on an Android-based embedded platform to collect messages compatible with the *Automatic Dependent Surveillance - Broadcast (ADS-B)* technology. With the purpose of evaluating this joint embedded hardware plus software solution, performance evaluation was done and compared with other general-purpose platform. A set of metrics was used in the evaluation and confidence intervals were used to validate the results.

Keywords: Air traffic. Android. Performance evaluation.

## LISTA DE ILUSTRAÇÕES

Figura 1 Micro ADS-B.....	12
Figura 2 Estrutura Genérica do Coletor.....	16
Figura 3 MK802 .....	16
Figura 4 Estrutura do Coletor Android executando no MK802 .....	17
Figura 5 Estrutura do Coletor Python Executando em Desktop.....	17
Figura 6 Tela principal da interface WEB.....	21
Figura 7 Esquema do cenário avaliado.....	22
Figura 8 Arquivo Amostra de mensagens capturadas .....	23
Figura 9 Amostra de um arquivo que demonstra o uso da CPU em uma captura.....	23
Figura 10 Amostra de um arquivo que representa o “tempo de vida” de uma mensagem.....	24
Figura 11 Modelo Lógico das Informações Coletadas .....	25
Figura 12 Média do uso de CPU .....	26
Figura 13 Média de mensagens .....	27
Figura 14 Médias do tempo de envio .....	29
Figura 15 Média da vazão .....	29
Figura 16 Tela principal software python.....	31
Figura 17 Tela principal aplicativo Android .....	32
Figura 18 Andoer CS918.....	32

## LISTA DE ILUSTRAÇÕES

Tabela 1 Corpo de uma mensagem ADSB.....	14
Tabela 2 Especificações de hardware do MK802 .....	17
Tabela 3 Especificações do desktop .....	18
Tabela 4 Resultado da avaliação .....	30
Tabela 5 Especificações do Andoer CS918.....	33

## SUMÁRIO

1 INTRODUÇÃO.....	11
2 FUNDAMENTAÇÃO TEÓRICA .....	13
2.1 ADS-B .....	13
2.2 Coletores ADS-B de Baixo Custo .....	15
2.2.1 Características dos coletores.....	15
2.2.2 Coletor Android executando no MK802 .....	16
2.2.3 Coletor Python Executando em Desktop.....	17
2.3 Avaliação de desempenho .....	18
2.3.1 Motivação .....	18
2.3.2 Métricas .....	18
2.3.3 Intervalo de confiança .....	19
3 TRABALHOS RELACIONADOS .....	20
3.1 Tecnologia ADS-B no Linux, uma implementação de Decodificação e Persistência de Mensagens. ....	20
3.2 Interface Web de Monitoramento de Tráfego Aéreo.....	20
3.3 Medições de ADS-B em Frankfurt .....	21
4 PROCEDIMENTOS METODOLÓGICOS .....	22
4.1 Coleta de Mensagens .....	22
4.2 Processamento das Mensagens .....	22
4.2.1 Mapeamento das Mensagens .....	24
4.2.2 Compilação dos Dados .....	24
5 RESULTADOS .....	25
5.1 Desempenho de Máquina local.....	26
5.1.1 Uso da CPU .....	26
5.1.2 Média de mensagens.....	27
5.2 Desempenho de Rede.....	28
5.2.1 Tempo de envio .....	28
5.2.2 Vazão .....	29
5.3 Visão geral .....	30
6 CONSIDERAÇÕES FINAIS .....	30
REFERÊNCIAS .....	34
APÊNDICES .....	35
APÊNDICE A – Aplicativo Android para a coleta de mensagens ADS-B.....	35

## 1 INTRODUÇÃO

Segundo a Airbus<sup>1</sup>, estimativas apontam que o tráfego aéreo no Brasil deve crescer cerca de 6,9% por ano até 2033, superando a média mundial de 5,4% (EXAME, 2013), com isso a segurança de passageiros e companhias se torna cada vez mais dependente de sistemas críticos, que garantam confiabilidade durante períodos de voo. Diante desse contexto, o uso de tecnologias que auxiliam o monitoramento aéreo de forma hábil, em tempo real, torna-se necessário.

As aeronaves transmitem dados/mensagens aos radares e sistemas de monitoramento que as identificam no espaço aéreo. São atividades dos sistemas de radares requisitar, captar e interpretar as mensagens, fornecendo aos operadores humanos informações como: identificação; posição (latitude, longitude e altitude); velocidade; e trajetória. Os sistemas em operação no Brasil permitem uma taxa de atualização, que é o tempo de espera entre uma mensagem e outra, de aproximadamente 30 segundos (SILVA, 2014). O sistema *Automatic Dependent Surveillance* (ADS) já está em processo de implantação no Brasil, mas ainda não é o principal sistema de monitoramento e controle de tráfego aéreo, sendo considerado apenas uma extensão ao modelo de radares aeronáuticos atuais para a aviação civil. A taxa de atualização do ADS se estabelece entre 0,5 e 2,0 segundos, valor configurável, demonstrando assim mais eficiência do que o atual sistema de monitoramento de tráfego aéreo utilizado no país.

O modo mais comum de utilização do sistema ADS é o broadcast (ADS-B), em que a aeronave envia periodicamente mensagens de identificação, posição, velocidade e rota, ou seja, sem a necessidade de algum estímulo de requisição. Tais mensagens, são sequências finitas de letras e números que a princípio, são humanamente ilegíveis, não fazem nenhum sentido aparente, sendo necessário criar um mapeamento que converta os dados brutos em uma informação compreensível.

Os atuais radares para o monitoramento aéreo são sistemas embarcados ou computadores robustos executando sistemas operacionais e softwares proprietários. Ambos os tipos requerem hardware complexo, não facilmente portátil e custos altos de aquisição (incluindo o custo do software). Para um aeroporto de médio ou grande porte, não é problema adquirir e manter um sistema de radares e softwares desse tipo. Porém, ainda com a crescente expansão aeroviária, o Brasil é composto por muitos aeroportos de pequeno porte, a maioria

---

<sup>1</sup> Site da Airbus: <http://www.airbus.com/>

destes presentes em cidades pequenas e propriedades privadas (POLIMENO, 2015), tornando-se inviável a implantação dos sistemas de monitoramento mais modernos e eficientes nesses locais. Mesmo possuindo pequenos fluxos aéreos, a segurança dos voos deveria ser sempre colocada em evidência, não importando os custos de prevenção de possíveis acidentes.

Um decodificador ADS-B de baixo custo interligado a uma antena caseira, como demonstrados na Figura 1, tem alcance de aproximadamente 370 km, o que pode ser amplificado com o uso de uma antena de maior ganho em maior altitude (por exemplo, em uma torre). Porém esse decodificador necessita de uma plataforma composta de hardware e software, para que possa ser feito o armazenamento e leitura das mensagens capturadas pelo alcance da antena. Com essa junção de decodificador, antena e uma plataforma, temos um coletor de mensagens ADS-B completo.

Figura 1 Micro ADS-B



Fonte: [www.microadsb.com](http://www.microadsb.com)

O objetivo principal deste trabalho é avaliar dois coletores de mensagens ADS-B, um que funcione sobre uma plataforma embarcada e o outro em uma plataforma de uso geral, procurando criar recomendações baseadas em vantagens e desvantagens, causas e consequências da utilização dessas tecnologias em situações específicas. Uma plataforma de hardware embarcada é um computador de pequeno porte, com desempenho razoavelmente inferior a de um computador pessoal comum, onde o sistema operacional pode estar fortemente integrado ao hardware. Por outro lado uma plataforma de uso geral é um computador pessoal comum, como desktop, notebook ou ultrabook.

Esses dois tipos de plataformas de hardware apresentam vantagens e desvantagens entre si. Primeiro é importante destacar que plataformas embarcadas por possuírem um pequeno porte, conseqüentemente possuem capacidade de hardware menor que uma plataforma de uso geral, tendo como benefício disso, na maioria das vezes o custo final de uma plataforma embarcada é

inferior a um computador pessoal. Outro ponto importante é a portabilidade, plataformas embarcadas são mais fáceis de transportar devido seu peso e suas dimensões.

Um dos objetivos específicos desse trabalho foi desenvolver um software capaz de ser executado em uma plataforma embarcada para manipular mensagens ADS-B, fornecendo um coletor enxuto, confiável, de fácil instalação e manuseio. A plataforma de execução desse software foi um mini PC conhecido por MK802, que requer o sistema operacional Android para funcionar. Como a proposta utiliza tecnologias não proprietárias, o custo final do produto é mais barato, tornando-o acessível a pequenos aeroportos que queiram fornecer mais uma camada de segurança em sua região de tráfego, ou universidades que queiram realizar algum estudo sobre o tráfego do espaço aéreo na região em que está localizada ou algum tipo de estudo e experimento envolvendo a tecnologia ADS- B, por exemplo.

Na seção 2 é apresentada a fundamentação teórica, com os conceitos relevantes para o entendimento desse trabalho. Na seção 3 são apresentados os trabalhos relacionados. Os trabalhos relacionados aqui mencionados, com exceção do último, fazem parte de um projeto maior chamado “Atualização da Monitoração Aeronáutica e Auto-Sustentabilidade”, realizados por professores e alunos do Campus UFC em Quixadá e todos relacionados à tecnologia ADS-B. Esse projeto, coordenado pelo prof. Arthur Callado, tem como proposta atualizar o parque tecnológico de pequenos aeroportos no país e assim prover mais um nível de segurança aos seus sistemas, através da implantação do sistema ADS-B integrado a uma plataforma para auxiliar na investigação de possíveis acidentes ou colisões de forma automática. Na seção 4 apresenta-se a motivação e os passos do experimento realizado. Na seção 5 são apresentados os resultados obtidos e algumas conclusões. Na seção 6 são apresentadas as considerações finais e propostos trabalhos futuros.

## **2 FUNDAMENTAÇÃO TEÓRICA**

### **2.1 ADS-B**

Tradicionalmente, radares terrestres usados para determinar a posição ou velocidade de aeronaves enviam ondas de rádio que são captadas e respondidas pelas aeronaves. E com base no tempo de ida e volta dessas ondas, radares como o *Surface Movement Radar* (SMR) determinam as informações correntes de um voo (CARVALHO, 2012).

O Automatic Dependent Surveillance - Broadcast (ADS-B) é uma nova tecnologia que está redefinindo a maneira como a gerência do tráfego aéreo é feita alterando os paradigmas de comunicação, navegação e vigilância. A tecnologia vem ganhando aceitação entre os

ANSPs, sigla em inglês para Prestadores de Serviços de Navegação Aérea, substituindo os radares tradicionais a um custo menor (BARSHEHAT, 2011). Existem projetos de implantação da tecnologia ADS-B em toda a Europa até o fim do ano de 2015, de forma ordenada, mas sem qualquer tipo de padronização (REKKAS, 2008).

Com o ADS-B, as aeronaves enviam dados/mensagens periodicamente, sem que sejam solicitadas, sobre identificação, posição através de GPS, velocidade, rota e estado (por exemplo, de emergência médica ou falha de equipamento). Essas mensagens são captadas por qualquer aeronave equipada com receptor ADS-B<sup>2</sup> ou por um receptor terrestre no raio de alcance da transmissão, desde que haja visada entre transmissor e receptor. O tempo típico entre envios das mensagens é de 1 segundo para cada mensagem de posição e velocidade, e 0,2 segundo para a de identidade (CARVALHO, 2012).

A captura das mensagens nesse trabalho é feita por um receptor microADS-B e uma antena de pequeno porte ligada a um decodificador com uma saída USB. Abaixo, na Tabela 1, é demonstrado um modelo semelhante ao utilizado neste projeto

A antena é responsável por captar as mensagens enviadas pelas aeronaves através de radiofrequência, ou seja, por ondas de rádio. O decodificador que é acoplado a essa antena e faz o processo de transformar as mensagens recebidas em algo facilmente compreensível por qualquer computador, em outras palavras, uma *string* de dados.

Essas mensagens em formato de *string* podem possuir dois tipos diferentes, a DF 11 e DF 17, cada uma com 56 e 112 bits, respectivamente. A diferença entre elas está na quantidade de informações que cada uma pode trazer, obviamente quanto maior a *string* maior a quantidade de informações. A *string* a seguir exemplifica uma mensagem DF 17, e posteriormente é mostrada na Figura 3 uma tabela que explica o significado de cada grupo de bits dessa mensagem.

Ex: 8D75804B580FF6B283EB7A157117

Tabela 1 Corpo de uma mensagem ADSB

Bits	1-5	6-8	9-32	33-88	89-112
Descrição	Tipo da mensagem (e.g., DF11 ou DF 17)	Código do tipo	Identificador da aeronave	Mensagem	Paridade

Fonte: Carvalho 2012

É difícil para um ser humano, mesmo um não leigo, extrair informações em tempo hábil de uma *string* desse tipo já que a decodificação envolve uma série de cálculos matemáticos.

<sup>2</sup> ADS-B: <http://www.ADS-B.com>



Para que isso seja feito de forma automática, foi desenvolvido um software que mapeia essas mensagens. Esse software foi executado em uma plataforma embarcada e para avaliar seu desempenho foi necessário compará-lo com outro software de mesmo propósito, porém que fosse executado em uma plataforma distinta de uma embarcada.

## 2.2 Coletores ADS-B de Baixo Custo

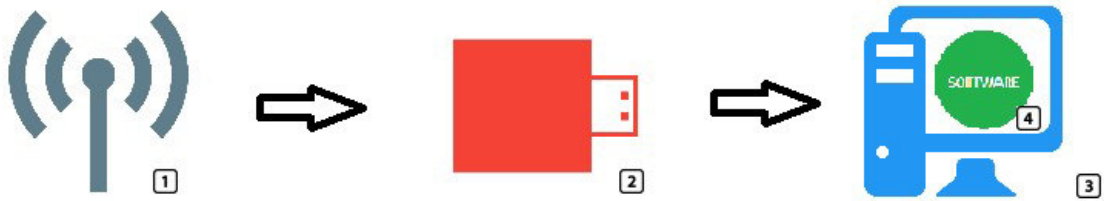
Para a realização dos testes de desempenho foram utilizados dois coletores de mensagens ADS-B. O primeiro foi desenvolvido neste trabalho e o segundo foi desenvolvido por alunos bolsistas do projeto “Atualização da Monitoração Aeronáutica e Auto-Sustentabilidade”. Mais características desses coletores encontram-se nas seções 3.2.1 e 3.2.2.

### 2.2.1 Características dos coletores

Cada coletor é formado por quatro partes distintas: uma antena, um decodificador ADS-B, um micro computador e um software de controle, como demonstrado na figura. A Figura 3 demonstra a estrutura que forma um coletor.

1. **Antena:** Responsável por fazer a detecção e a captura das mensagens no ambiente.
2. **Decodificador ADS-B:** Comunica-se com a antena através de um cabo coaxial, faz a conversão das ondas de radio recebidas por ela em *strings* de dados, ou seja, as mensagens propriamente ditas, e as armazenas por um curtíssimo período de tempo, pois possui um buffer de apenas 64 bytes.
3. **Computador:** Comunica-se com o decodificador ADS-B via interface USB e armazena as mensagens de forma permanente com o auxílio de um software. Nesse trabalho foram utilizados dois tipos de computadores, descritos nas seções 2.2.2 e 2.2.3.
4. **Software:** Responsável por controlar a comunicação entre as partes do coletor. Possui os requisitos fundamentais de solicitar constantemente novas mensagens que foram armazenadas no decodificador ADS-B, e posteriormente transferi-las para um servidor remoto.

Figura 2 Estrutura Genérica do Coletor



Fonte: elaborado pelo autor

### 2.2.2 Coletor Android executando no MK802

O primeiro coletor foi desenvolvido durante a execução desse projeto para ser executado em um PlugComputer, que nada mais é do que um computador de pequeno porte, com desempenho razoavelmente inferior a de um computador de uso geral, onde o sistema operacional pode estar fortemente integrado ao hardware. Neste projeto, usamos essa proposta de hardware para “abrigar” o software controlador do coletor devido ao seu baixo custo (o equipamento usado no projeto custou 55 dólares).

O modelo de PlugComputer utilizado foi o MK802, executando o sistema operacional Android na versão 4.0 (*Ice Cream Sandwich*), como pode ser visto na Figuras 3 e Tabela 2. O Android é um sistema operacional tipicamente encontrado em dispositivos móveis como smartphones e tablets, porém vem ganhando espaço em outras plataformas como em relógios inteligentes, televisões e até mesmo automóveis. O software foi desenvolvido na linguagem Java baseando-se no SDK (*Software Development Kit*) do Android. A Figura 4 demonstra a estrutura do coletor.

Figura 3 MK802



Fonte: CNXSoft<sup>3</sup>

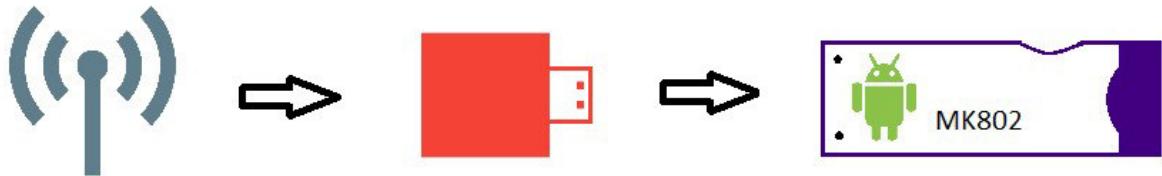
<sup>3</sup> CNXSoft <http://www.cnx-software.com>

Tabela 2 Especificações de hardware do MK802

CPU	AllWinner A10 @ 1.5GHz + Mali 400 GPU
Memória	512MB RAM
Armazenamento	4GB Flash microSD slot (Up to 32GB)
USB	micro USB 2.0/OTG port USB 2.0 Host port
Teclado	Teclado virtual do Android
Saída de Vídeo	HDMI (1080p)
Dimensões	8.8*3.5*1.2cm
Peso	200g

Fonte: CNXSoft, 2015

Figura 4 Estrutura do Coletor Android executando no MK802



Fonte: elaborado pelo autor

### 2.2.3 Coletor Python Executando em Desktop

O segundo coletor utiliza um computador de uso geral como hardware base, e o software pode ser utilizado em qualquer micro computador com sistema operacional Linux. Neste trabalho foi utilizado um desktop com sistema operacional Linux para executar o software desse coletor que foi implementado em linguagem Python. Essa proposta de coletor é atualmente utilizada pelo projeto Atualização da Monitoração Aeronáutica e Auto-Sustentabilidade. A Figura 5 abaixo demonstra a estrutura do coletor e na Tabela 3 vemos as configurações do computador utilizado.

Figura 5 Estrutura do Coletor Python Executando em Desktop



Fonte: elaborado pelo autor

Tabela 3 Especificações do desktop

CPU	AMD Athlon X2 Dual Core 3.0 GHz
Memória	2GB RAM DDR3
Armazenamento	60GB HD
Sistema Operacional	Xubuntu 14.04

Fonte: elaborado pelo autor

## 2.3 Avaliação de desempenho

### 2.3.1 Motivação

Para testar a viabilidade do uso de um sistema torna-se necessária sua avaliação, especialmente em se tratando de sistemas que demandam de uma alta taxa de manipulação e controle de dados. Nesse trabalho queremos avaliar dois coletores de mensagens ADS-B, um baseado em uma plataforma embarcada e ou outro em uma plataforma de uso geral, e para isso foram coletadas informações com base no uso de recursos de hardware e no consumo/produção de dados realizados por eles.

Observa-se que ambos os coletores possuem características bastante distintas, destacando-se a diferença na capacidade de hardware entre o MK802 e o Desktop. Ambos os softwares utilizados nos coletores têm como requisito principal, fazer requisições constantes ao decodificador por mensagens que foram captadas pela antena e nele foram armazenadas, e imediatamente, enviá-las a um servidor central, que armazena, decodifica e apresenta as informações presentes nessas mensagens (interface descrita na seção 2.2). Essa atividade é parte fundamental do sistema ADS-B, pois ele determina o tempo de atualização das informações de um dado voo, por isso queremos avaliar quanto de recurso é demandado dos coletores, e até que ponto ele suporta realizar essa atividade sem que haja perda de informação. Os coletores executaram essa única atividade para a realização dos testes de desempenho.

### 2.3.2 Métricas

Métricas podem ser definidas como indicadores que determinam quantitativamente o desempenho de um dado processo. Nesse trabalho definimos dois grupos de métrica para análise de desempenho, uma para análise de desempenho do hardware e o outro para medir o desempenho de rede.

Para o grupo de métricas de desempenho de hardware foram definidas duas métricas, média de mensagens e uso da CPU. Entende-se como média de mensagens, a quantidade

média de mensagens por minuto coletadas e transferidas para o servidor remoto, a quantidade de mensagens interceptadas pelo coletor determinar o grau de frequência com que um voo monitorado é atualizado. O uso de CPU entende-se pelo percentual de CPU utilizado em um dado instante de tempo, a quantidade de CPU utilizada determina o esforço feito pela plataforma de hardware do coletor para executar alguma atividade, quanto menor o esforço melhor para o coletor de dados, indicando uma menor possibilidade de perda de informações devido a uso da CPU.

Para o grupo de métricas de rede também foram definidas duas, vazão e tempo de envio. A vazão define-se como a quantidade de bytes trocada entre o coletor e o servidor, essa medida está diretamente relacionada com a média de mensagens recebidas, já que quanto maior a vazão maior a média de mensagens e conseqüentemente maior a frequência de atualização dos voos monitorados. O atraso é o tempo gasto para a sincronização entre as informações do coletor e o servidor, ou seja, é medido o tempo usado para uma mensagem ser levada do banco de dados local do coletor até o banco de dados do servidor remoto e a plataforma emissora receba a confirmação do recebimento dessa mensagem. Ambos os coletores apresentam interfaces diferentes de comunicação com a internet, tornando-se necessário a visualização do impacto causado por essa diferença no tempo de envio das mensagens coletadas,

### **2.3.3 Intervalo de confiança**

Na estatística inferencial o intervalo de confiança é utilizado para demonstrar a confiabilidade de uma estimativa. Segundo (Moreschi, 2010), é um valor estimado de um parâmetro estatístico. Em vez de se estimar algum parâmetro por um único valor, é fornecida uma faixa de estimativas prováveis. Quão prováveis são essas estimativas é determinado pelo nível de confiança. Quanto maior a probabilidade de o intervalo conter o parâmetro, maior será o intervalo.

A técnica foi aplicada para as quatro métricas citadas na seção anterior para demonstrar com maior precisão o seu resultado.

## **3 TRABALHOS RELACIONADOS**

### **3.1 Tecnologia ADS-B no Linux, uma implementação de Decodificação e Persistência de Mensagens.**

Um programa decodificador de mensagens ADS-B foi desenvolvido por Carvalho (2012), para ser executado em uma plataforma de hardware de uso geral - desktops ou notebooks - utilizando sistema operacional Linux. Entretanto, o mesmo deixa como trabalho futuro a adaptação do programa para um PlugComputer, que é um tipo de computador com hardware mais enxuto e, sendo menor e mais barato que computadores pessoais, torna-se uma boa alternativa de instalação em ambientes de difícil acesso e com pouca estrutura tecnológica (CARVALHO, 2012). Este projeto tem como um de seus objetivos dar continuidade a este trabalho, propondo um software coletor e decodificador de mensagens ADS-B que funcione em uma plataforma embarcada.

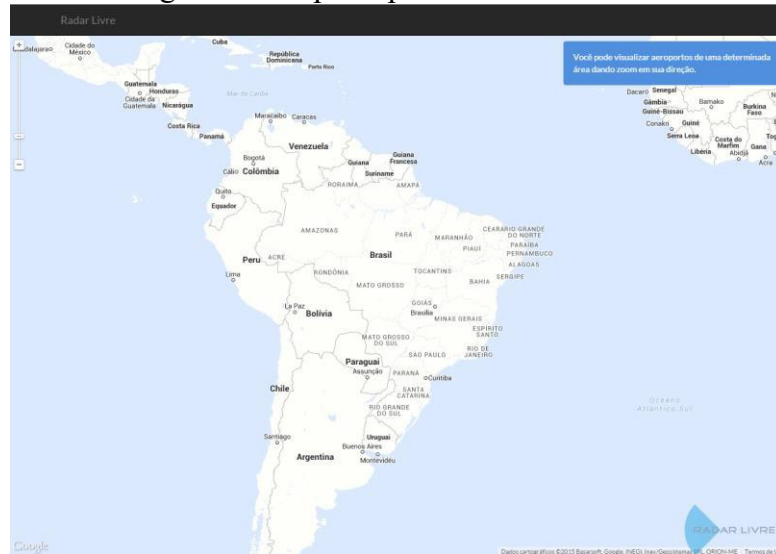
O MK802 foi escolhido como o PlugComputer que conterà o software decodificador das mensagens capturadas. É importante ressaltar que o MK802 não faz a captura dos dados por si só, esta função é feita pelo receptor ADS-B que estará diretamente conectado a este dispositivo.

### **3.2 Interface Web de Monitoramento de Tráfego Aéreo**

Silva (2014) desenvolveu uma interface web responsável por apresentar um mapa com dados de voos advindos de mensagens ADS-B já decodificadas e humanamente legíveis obtidas a partir de um banco de dados modelado por ele que recebia dados (mensagens) de outros coletores ADS-B. O mesmo também apresenta uma proposta de decodificador que já disponibiliza suas mensagens parcialmente mapeadas com a ajuda de outros softwares, a fim de validar as funcionalidades de sua aplicação web. A Figura 6 demonstra a tela inicial da desta aplicação.

O sistema decodificador proposto por este projeto tem dois requisitos funcionais que em conjunto contribuem com o fornecimento de dados para interface WEB. O primeiro é armazenar as mensagens em um banco de dados local e o segundo é, mediante a disponibilidade de internet, fazer uma sincronização entre a base local e o banco de dados remoto que alimenta essa interface web. Assim interface web sempre estará atualizada com dados de voos capturados pelos receptores.

Figura 6 Tela principal da interface WEB



Fonte: Silva, 2014

### 3.3 Medições de ADS-B em Frankfurt

Em HARMAN (2002) foi feita uma análise da implantação da tecnologia ADS-B na cidade de Frankfurt, uma das maiores cidades da Alemanha. Durante cinco dias, foram realizados teste a fim de medir o nível de interferência que existia na frequência de 1090 Mhz, que é a mesma utilizada pela tecnologia ADS-B, sendo que essa interferência é causada por radares secundários.

Nos testes foram utilizadas três aeronaves dedicadas aos experimentos e dois radares terrestres. Durante os testes procurou-se medir a recepção sob dois pontos de vista, primeiro, a recepção de aeronave para aeronave e a segunda de aeronave por radar terrestre. Todos esses testes de desempenho foram feitos em função da distancia entre emissor e receptor, simulando rotas reais de voo que circulam a região de Frankfurt.

Como comparação dos resultados utilizaram dados coletados de testes semelhantes realizados em Los Angeles nos Estados Unidos.

Concluindo como já era esperado, demonstrou-se a presença de altas taxas de interferência causada por radares secundários, porém com taxas de transmissão de dados tidas como excelentes em função de certas distancias entre aeronaves e radares terrestres.

## 4 PROCEDIMENTOS METODOLÓGICOS

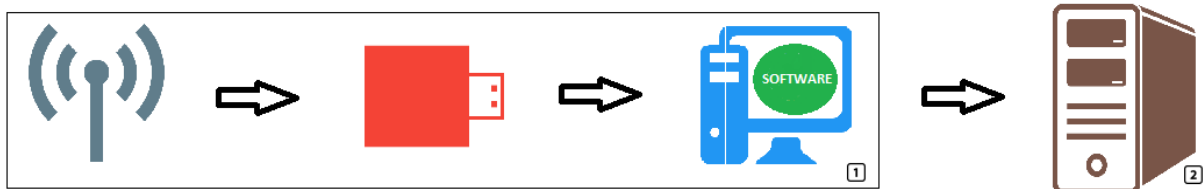
Nessa seção, é apresentada a descrição dos experimentos realizados, demonstrando como os dados foram coletados e estruturado para análise.

### 4.1 Coleta de Mensagens

Para medir o desempenho dos dois coletores uma série de testes foi realizada, através das coletas de mensagens de voos em tempo real, sendo todas executadas na cidade de Fortaleza-CE. Ambos os coletores eram postos em execução paralela durante processo de captura, atribuindo um intervalo de tempo determinado, podendo ser de 30 ou 60 minutos, e com isso tinham o mesmo tempo de inicio e fim. Tudo isso para garantir que pudessem ter acesso ao mesmo conjunto de dados, ou seja, o mesmo conjunto de mensagens e os voos aos quais elas pertencem.

Os processos de capturas foram divididos em dois grupos, O primeiro grupo para a avaliação de desempenho de rede e o segundo para a avaliação de hardware. Antes de demonstrar o que se buscou medir nesses dois grupos de captura, é necessário entender que os dados captados não ficam retidos internamente nos coletores. Existe um servidor responsável por receber todas as mensagens que foram capturadas pelos coletores, expandimos a Figura 7 para demonstrar o cenário completo que foi avaliado. O coletor (1) envia todos os dados capturados para o servidor (2) através de um socket TCP estabelecido entre ambos. Esse mesmo servidor é utilizado como base para a interface web descrita na seção 3.2.

Figura 7 Esquema do cenário avaliado



Fonte: elaborado pelo autor

### 4.2 Processamento das Mensagens

No fim de uma coleta todas as mensagens capturadas eram armazenadas em um arquivo CSV, dividido em duas colunas, a primeira continha o tempo em milissegundos em que a mensagem foi capturada, e a segunda coluna continha a *string* da mensagem. A Figura 8



demonstra uma pequena amostra de um arquivo com cinco linhas, onde cada linha representa uma mensagem e o momento da sua captura.

Figura 8 Arquivo Amostra de mensagens capturadas

1453038229073,8DE480A49910ED1048480E648D59
1453038240116,8DE480A4582311722D3D6E389F67
1453038241701,8DE480A45823057D3B74EE83A653
1453038242796,8DE480A49910EB10883C0E946D19
1453038246826,8DE480A49910EB1048580D7183EF

Fonte: elaborado pelo autor

Como já mencionado, o processo de coleta foi dividido em dois grupos, em um dos grupos foi avaliado o consumo de CPU por parte do coletor, e como resultado as coletas realizadas nesse grupo geraram um segundo arquivo similar ao da Figura 8, porém com a seguinte estrutura: há duas colunas separadas por ponto e vírgula, onde na primeira linha temos um instante de tempo em milissegundos e na segunda coluna temos o percentual da CPU utilizado pelo hardware neste momento, mostrado na Figura 9.

Figura 9 Amostra de um arquivo que demonstra o uso da CPU em uma captura

1453038240058;2,86
1453038245421;2,78
1453038250784;5,56
1453038256148;8,11
1453038261511;5,56
1453038266877;21,05

Fonte: elaborado pelo autor

O outro grupo de coletas busca avaliar o tempo que a mensagem leva do momento em que é captada pelo coletor até o momento que é transmitida ao servidor remoto, e para se determinar essa velocidade, foram gerados arquivos como demonstrado na amostra da Figura 10. A primeira coluna representa o momento em que a mensagem foi capturada, a segunda coluna representa o conteúdo da mensagem e a terceira coluna representa o tempo de confirmação do servidor por ter recebido essa mensagem. Com uma subtração simples, entre a terceira e a primeira coluna, podemos obter o tempo em que uma mensagem leva pra ser sincronizada com o servidor remoto.

Figura 10 Amostra de um arquivo que representa o “tempo de vida” de uma mensagem

```
1449533387221,9F0761F24082CC0FD384000088DB,1449533388163
1449533427383,9C043641080B88B1B0AB2ADDACE5,1449533428101
1449533447701,9F43051CFA2ABB870733900B0BF9,1449533447962
1449533526099,97CEF3288CB495C600ED05100B9D,1449533526198
1449533545579,8F239D109AC8202BD84E028265C6,1449533545960
1449533594588,95822902CB89AA276742DAE0CB5D,1449533595014
```

Fonte: elaborado pelo autor

#### 4.2.1 Mapeamento das Mensagens

Para realizar a extração das informações das mensagens foi utilizado a API Java-adsb, foi desenvolvida pelo grupo **OpenSky Network**<sup>4</sup> e é mantida sob *GNU General Public License*.

Essa API atualmente suporta e decodifica os seguintes tipos de mensagens:

- Mensagens de identificação;
- Mensagens de velocidade;
- Mensagens de posição em solo;
- Mensagens de posição em voo;
- Status (emergências/prioridades).

#### 4.2.2 Compilação dos Dados

Ao final de cada captura eram recolhidos todos os arquivos de texto gerados pelos coletores após o processo. Todas as informações eram extraídas e processadas por um script capaz de ler os arquivos e transferir suas informações de forma estruturadas a uma base de dados. A Figura 11 demonstra a distribuição e relacionamentos das tabelas e seus respectivos campos.

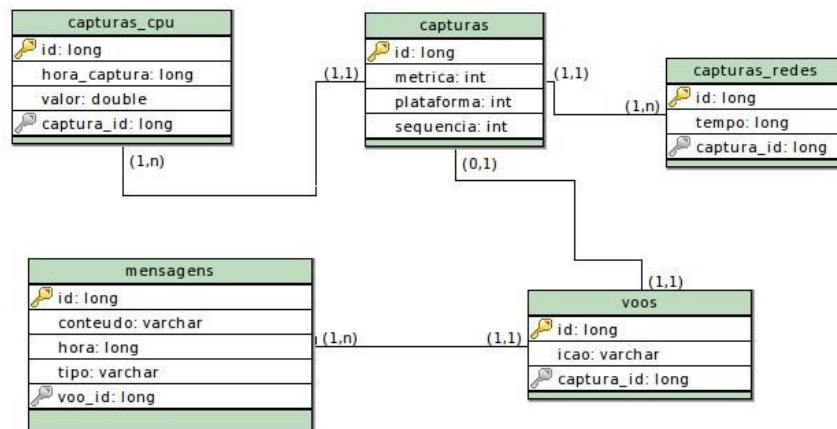
- **Tabela capturas:** possui o campo *métrica* que determina o que está sendo avaliado naquela captura, podendo ser encontrado os valores 0 para indicar que está sendo avaliado o desempenho de hardware ou 1 para demonstra que está sendo avaliado o desempenho de rede, o campo *plataforma* indica qual a plataforma está sob análise, podendo assumir 0 para indicar que os dados daquela captura pertencem a plataforma embarcada (MK802 com aplicação Android) ou 1 para indicar que está sendo avaliado a computador de uso geral (Desktop com aplicação python). Por último o campo

<sup>4</sup> OpenSky Network <https://opensky-network.org/>

*sequência* indica os pares de coletores utilizados em paralelo, ou seja, até duas tuplas podem possuir o mesmo número de sequência, indicando que aquelas duas capturas foram realizadas em paralelo e tiveram acesso ao mesmo conjunto de dados.

- **Tabela *capturas\_cpu*:** armazena as informações do uso de CPU durante a captura, caso a coluna *métrica* na tabela *capturas* tenha sido marcada com 0.
- **Tabela *capturas\_rede*:** armazena as informações do uso de rede durante a captura, caso a coluna *métrica* na tabela *capturas* tenha sido marcada com 1.
- **Tabela *voos*:** a coluna *icao* armazena o identificador do voo, mais especificamente o identificador do *transponder* da aeronave.
- **Tabela *mensagens*:** a coluna *conteúdo* apresenta a *string* da mensagem, a coluna *tipo* indica que tipo de informação apresenta a mensagem, podendo ser identificação, velocidade, posição ou status.

Figura 11 Modelo Lógico das Informações Coletadas



Fonte: elaborado pelo autor

## 5 RESULTADOS

Para análise dos resultados foi utilizado um total de 40 horas de capturas divididas entre os dois coletores, e as quatro métricas usadas na avaliação de desempenho. Apenas duas atividades eram executadas pelos coletores durante o experimento: a captura de mensagens aéreas pelo coletor; e o envio dessa mensagem para um servidor remoto. Cada experimento (captura) tem um tempo estipulado em 10 minutos.

Antes dessas quarenta horas de testes terem sido realizadas, outras muitas horas de testes haviam sido executados, porém percebeu-se uma falha em um dos decodificadores ADS-B. Até que isso tivesse sido descoberto havia sido notada certa deficiência na coleta de mensagens em um dos coletores e para determinar onde estava o problema foi preciso fazer uma série de teste permutando as antenas, decodificadores e plataformas de hardware disponíveis entre si pra concluir onde estava a falha. Assim que o problema com um dos decodificadores foi confirmado, houve uma troca imediata de equipamento para a realização dos testes para validação desse trabalho.

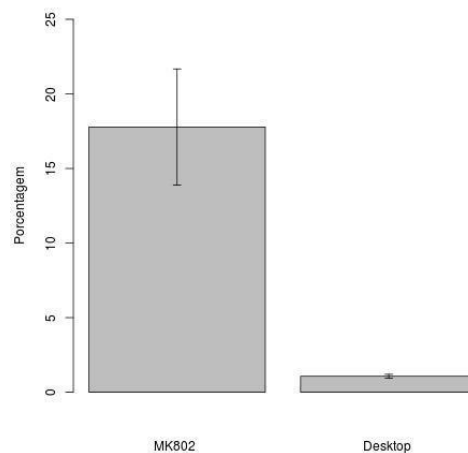
## 5.1 Desempenho de Máquina local

Ao se iniciar uma captura determinava-se a quantidade de amostras da CPU que iriam ser capturadas e qual o intervalo de tempo entre uma e outra amostra. Em todos os casos foram definidas 120 capturas de CPU com um intervalo temporal de 5 segundos, totalizando uma captura de 10 minutos.

### 5.1.1 Uso da CPU

Foram realizados 30 experimentos, e em cada um deles foi extraída a média do uso da CPU durante o processo de captura. Ao final, foi calculado o intervalo de confiança sobre as 30 médias obtidas, obtendo uma estimativa que indique confiança sobre o parâmetro calculada. A Figura 12 demonstram as médias e um nível de confiança 95% para as amostras.

Figura 12 Média do uso de CPU



Fonte: elaborado pelo autor

O intervalo de confiança sobre a média do uso da CPU para o MK802 foi de 13,88% e 21,67% enquanto a média de CPU do Desktop gira em torno de 0,93% e 1,19%. Os dados de

CPU utilizados levam em conta todos os processos em execução no dispositivo, ou seja, não apenas os processos criados pelo software do coletor.

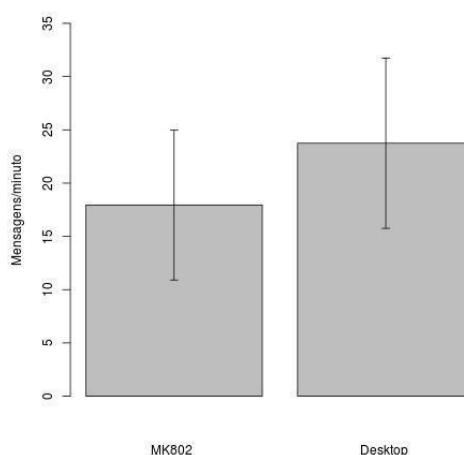
Nota-se que a diferença entre os limites do intervalo de confiança do dispositivo Android é bastante expressiva se comparada com os limites do intervalo do Desktop. Observou-se que as capturas de CPU do MK802 oscilavam bastante, variando entre diferentes níveis (porcentagem) e atingindo muitas vezes picos de 100%, enquanto o uso da CPU no Desktop se mantinha com um nível de uso mais constante.

O fato de o MK802 atingir muitas vezes picos de 100% no uso de sua CPU, mesmo havendo apenas os processos do software sendo executado, pode ter acarretado perda de mensagens durante a captura, diminuindo assim sua eficiência na coleta de dados aéreos. Supõe-se que esses picos elevados sejam causados por outros processos que executam em segundo plano, e o fato da tela do MK802 nunca entrar em modo de descanso, o que acaba consumindo muitos recursos.

### 5.1.2 Média de mensagens

Foram realizados 30 experimentos, e em cada um deles foi extraída a média de mensagens que foram recebidas e enviadas pelos coletores durante o processo de captura. A Figura 13 demonstra as médias e um nível de confiança de 95% para as amostras.

Figura 13 Média de mensagens



Fonte: elaborado pelo autor

O intervalo de confiança para se determinar a capacidade do MK802 ficou entre 10,9 mens./min e 24,9 mens./min enquanto o no Desktop o intervalo é de 15,7 mens./min e 31,7 mens./min. Nesse quesito observou-se que as diferenças entre os limites do intervalo de

confiança das plataformas são bem pequenas. Porém, os intervalos estão muito grandes isso demonstra que as mensagens não são recebidas com frequência constante, Observou-se que durante as capturas, havia muitos momentos de ociosidade, ou seja, nenhuma mensagem estava sendo processada. Muitas vezes em um experimento de 10 minutos, em apenas 10% do seu tempo estava sendo processada alguma mensagem.

Para o cálculo da média de mensagens foi utilizado o número de mensagens capturadas durante um experimento, em alguns casos o MK802 que possui um poder de processamento inferior, conseguiu capturar mais mensagens que o Desktop. Supõe-se que isso tenha acontecido devido o posicionamento da antena que favoreceu a coleta do dispositivo Android.

## **5.2 Desempenho de Rede**

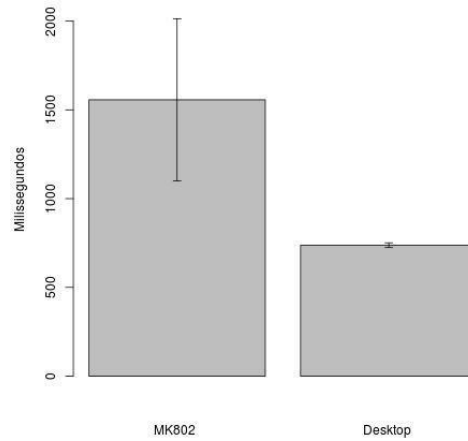
Ambos os coletores estavam conectados ao servidor remoto com IP e porta através de um socket TCP, a conexão por eles utilizadas foi de 35Mb. É importante ressaltar que o MK802 possui como única forma de comunicação com internet através de rede sem fio, enquanto o Desktop utiliza apenas conexão cabeada.

### **5.2.1 Tempo de envio**

O tempo de envio foi definido como o tempo de ida de uma mensagem para o servidor mais o tempo de volta da confirmação de seu recebimento. Para determinar o tempo médio de atraso de uma determinada captura (experimento), foi somando todos os tempos de atraso de todas as mensagens e posteriormente dividido pela quantidade mensagens transferidas. A Figura 14 demonstra o tempo médio de envio ocorrido em 30 capturas, com intervalo de 95% de confiança.

O intervalo de confiança para a média de atraso no MK802 ficou entre 1099 e 2013 milissegundos, enquanto a média do Desktop ficou entre 724 e 751 milissegundos. Nota-se certa diferença entre os tempos médios dos dois coletores. Isso ocorre devido às interfaces com a qual eles se comunicam com o meio externo, o MK802 utiliza apenas a Wi-Fi para a comunicação com a rede, o que pode gerar perdas de pacotes e atrasos devido à taxa de ruídos presentes em comunicação wireless, já o Desktop utiliza um meio físico para se comunicar com a rede, em uma comunicação full-duplex sem ruídos ou colisões.

Figura 14 Médias do tempo de envio



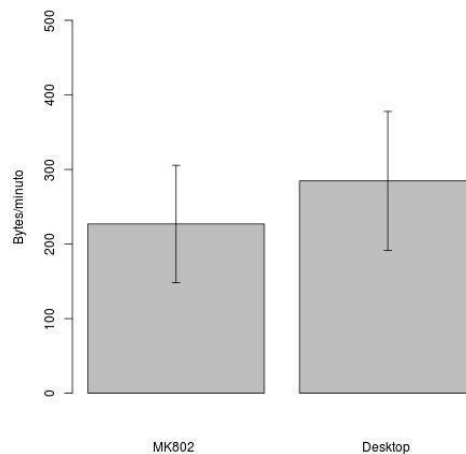
Fonte: elaborado pelo autor

### 5.2.2 Vazão

A vazão é a média da relação bytes/minuto transmitidos entre o coletor e o servidor em uma determinada captura. A Figura 15 mostra a média da vazão em 30 capturas, com intervalos de confiança calculados para um nível de confiança de 95%.

O intervalo de confiança para a vazão média do MK802 é de 148 bytes/min. a 305 bytes/min. e do Desktop 191 bytes/min e 377 bytes/min. Essa relação está diretamente relacionada com a capacidade (seção 5.1.2), ou seja, quanto maior a média de mensagens, maior a vazão.

Figura 15 Média da vazão



Fonte: elaborado pelo autor

### 5.3 Visão geral

Os gráficos nas seções anteriores são uma representação ilustrativa da dispersão das médias das amostras através de um estimado intervalo de confiança. Sendo assim, ao invés de estimar a média (o parâmetro) por um único valor, determinou-se um intervalo de estimativas. O nível de confiança determina a confiabilidade do intervalo.

A Tabela 4 demonstra um compilado com todas as métricas usadas e três intervalos de confiança calculados para um nível de confiança de 95%. Através dessa tabela pode-se demonstrar com maior clareza o resultado de uma determinada métrica. É importante lembrar que o nível de confiança, nesse caso 95%, demonstra que essa mesma porcentagem de amostras está dentro do intervalo de confiança. O ideal para obter um nível de confiança alto e um intervalo de confiança pequeno, seria um número maior de amostras (médias das capturas).

Tabela 4 Resultado da avaliação

Métricas	Intervalo de confiança (media + ou - margem de erro)	
	MK802	Desktop
Uso de CPU (%)	17,7 (+ ou - 3,8)	1,06 (+ ou - 0,13)
Média de mensagens (Mensagens/minuto)	17,9 (+ ou - 7)	23,7 (+ ou - 7,9)
Tempo de envio (milissegundos)	1556 (+ ou - 456)	737 (+ ou - 13)
Vazão (bytes/minuto)	226,9 (+ ou - 78,6)	284,3 (+ ou - 93,1)

## 6 CONSIDERAÇÕES FINAIS

Durante a execução dos testes não foi possível extrair o máximo de desempenho dos coletores, devido o fato deles estarem operando em uma região cujo tráfego não é tão intenso, o que acaba gerando certos períodos de ociosidade, ou seja, nenhuma mensagem é coletada em algum dado período do captura. Inicialmente pretendia-se fazer a medição das métricas em função da quantidade de aeronaves que estavam sendo observadas durante o experimento, porém, o baixo fluxo de aeronaves que eram observados em simultâneo mantinha-se em torno de uma ou duas, e algumas vezes nenhuma aeronave era detectada durante a coleta, o que obviamente anulava o experimento.

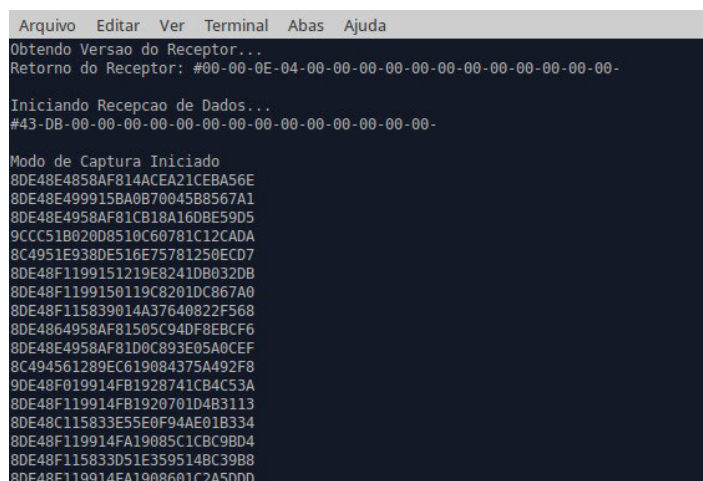


Observou-se em alguns experimentos que o MK802 conseguiu capturar mais mensagens de aeronaves do que o Desktop que possui uma qualidade de hardware mais elevada. Acredita-se que isso se deu devido à disposição da antena, embora ambas as antenas usadas no experimento fossem dispostas à mesma altura e uma ao lado da outra. Porém como observado o Desktop conseguiu lidar com um maior número de mensagens e com menor esforço, tornando-o mais confiável a possíveis instabilidades. É importante mencionar que MK802 passa por diversos momentos de instabilidade, trava e encerra aplicações com certa frequência, tornando seu uso cansativo, pois muitas vezes havia demora em responder às interações com a interface.

Em relação ao desempenho de redes já foi mencionando que o MK802 possui um desempenho inferior por possui apenas conexão Wi-Fi prejudicando o envio das mensagens para o servidor devido à alta taxa de ruídos presentes nesse tipo de conexão. Porém um fato que chamou a atenção durante os testes foi a dificuldade que o dispositivo tem em manter uma conexão com o ponto de acesso da rede local, demorando muito para conectar-se a rede mesmo estando bem próximo ao roteador wireless.

O aplicativo Android possui diversas funcionalidades que o software em python não possui que facilitam o uso do coletor, como por exemplo, a tradução prévia de mensagens, o aplicativo é capaz de fornecer informações das mensagens que ele coletou, enquanto o software em python apenas demonstra uma lista de mensagens que foram capturadas, podemos ver a diferença nas Figuras 16 e 17, onde são exibidas as interfaces principais dos dois coletores e mostram o que está sendo capturado. Mais informações sobre as funcionalidades do aplicativo desenvolvido nesse trabalho podem ser encontradas no Apêndice A.

Figura 16 Tela principal software python



```
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
Obtendo Versao do Receptor...
Retorno do Receptor: #00-00-0E-04-00-00-00-00-00-00-00-00-00-00-00-00-
Iniciando Recepcao de Dados...
#43-DB-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-
Modo de Captura Iniciado
8DE48E4858AF814ACEA21CEBA56E
8DE48E499915BA0B70045B8567A1
8DE48E4958AF81CB18A160BE5905
9CC51B02008510C60781C12CADA
8C4951E9380E516E75781250ECD7
8DE48F1199151219E8241DB032DB
8DE48F1199150119C8201DC867A0
8DE48F115839014A37640822F568
8DE4864958AF81505C94DF8EBCF6
8DE48E4958AF81D0C893E05A0CEF
8C494561289EC619084375A492F8
9DE48F019914FB1928741CB4C53A
8DE48F119914FB1920701D4B3113
8DE48C115833E55E0F94AE01B334
8DE48F119914FA19085C1CBC9B04
8DE48F115833D51E359514BC39B8
8DE48F119914FA1908601C2A5DD0
```

Figura 17 Tela principal aplicativo Android

Hex Code	Timestamp	Label
8DE490EE9940963D00045FB788B5 e490ee	29/01/2016 - 11:55	VELOCITY
8DE490EE58C7F406513EC2B86A48 e490ee	29/01/2016 - 11:55	AIRBORN_POSITION
8DE490EE9940963D00045FB788B5 e490ee	29/01/2016 - 11:55	VELOCITY
8DE490EE58C7F3F49B066565A708 e490ee	29/01/2016 - 11:55	AIRBORN_POSITION
8DE490EE9940963D00045FB788B5 e490ee	29/01/2016 - 11:54	VELOCITY
8DE490EE58C7F3F3D506477F2BC6 e490ee	29/01/2016 - 11:54	AIRBORN_POSITION
8DE490EE58C7F404B93E8558E174 e490ee	29/01/2016 - 11:54	AIRBORN_POSITION
8DE490EE9940963D00045FB788B5 e490ee	29/01/2016 - 11:54	VELOCITY
8DE490EE9940963D00045FB788B5 e490ee	29/01/2016 - 11:54	AIRBORN_POSITION

Em resumo o aplicativo desenvolvido não teve todo seu potencial explorado por dois fatores: o baixo desempenho e os problemas citados com o MK802; e o fato das coletas terem sido realizadas em uma região de pouca intensidade de tráfego aéreo. Como trabalhos futuros pretende-se realizar os mesmos testes, porém com o aplicativo executando sobre outras plataformas Android. Um exemplo de plataforma Android alternativa seria o Andoer CS918. A Figura 18 e Tabela 5 mostram suas características.

Figura 18 Andoer CS918



Fonte: amazon.com, 2015

Tabela 5 Especificações do Andoer CS918

CPU	RK3188T Quad-core
Memória	1 G DDR3
Armazenamento	8 GB NAND Flash
Sistema operacional	Android 4,4
Conexão	WI-FI 802,11 b/g/n 2,4 GHz e cabada com RJ45
Saída de vídeo	1920x1080 p @ 60 Hz Full HD

Fonte: amazon.com, 2015

## REFERÊNCIAS

- BAUD, O.; HONORE, N.; TAUPIN, O. Radar/ADS-B Data Fusion Architecture for Experimentation Purpose. In: Proceedings of the 9th International Conference on Information Fusion, Florence, Italy, 10–13 jul 2006; pp. 1–6.
- BARSHEHAT, A. A. Implementation of ADS-B Systems - Benefits and Considerations In: Proceedings of ESAV, 2011 - September 12 - 14 Capri, Italy.
- HARMAN, W. H. ADS-B Airborne measurements in Frankfurt. MIT Lincoln Laboratory, Lexington, MA. USA. 2002.
- CARVALHO, S. A. Tecnologia ADS-B no Linux, uma Implementação de Decodificação e Persistência de Mensagens. Monografia - Universidade Federal do Ceará Campus Quixadá, Quixadá. 2012.
- EXAME. Setor aéreo brasileiro deve crescer acima da média mundial Disponível em: <<http://exame.abril.com.br/publicidade/ge/setor-aereo-brasileiro-deve-crescer-acima-da-media-mundial/>>. Acesso em 08 de fev. 2016.
- FLYNN, Geraldine; BENKOUAR, A.; CHRISTIEN, R. Pessimistic sector capacity estimation. EEC Note, v. 21, n. 03, 2003.
- MOON, Woo-Choon et al. Air traffic volume and air traffic control human errors. **Journal of Transportation Technologies**, v. 1, n. 03, p. 47, 2011.
- POLIMENO. O Potencial dos Pequenos Disponível em: <[http://aeromagazine.uol.com.br/artigo/o-potencial-dos-pequenos\\_260.html](http://aeromagazine.uol.com.br/artigo/o-potencial-dos-pequenos_260.html)>. Acesso em 10 de jun. 2015.
- REKKAS, C., REES, M. Towards ADS-B implementation in Europe, in: Proceedings of ESAV, 2008, September 3 – 5, Capri, Italy.
- SILVA, E. R. Interface Web de Monitoramento de Tráfego Aéreo. Monografia - Universidade Federal do Ceará Campus Quixadá, Quixadá. 2014.
- STANISCIA, Giacomo Feres. Gerenciamento de Tráfego Aéreo de Nova Geração. ATECH Tecnologias Críticas. LAAD 2007 – Latin America Aero & Defence 07. Slides: <http://dc354.4shared.com/doc/pkv8ekFG/preview.html>.

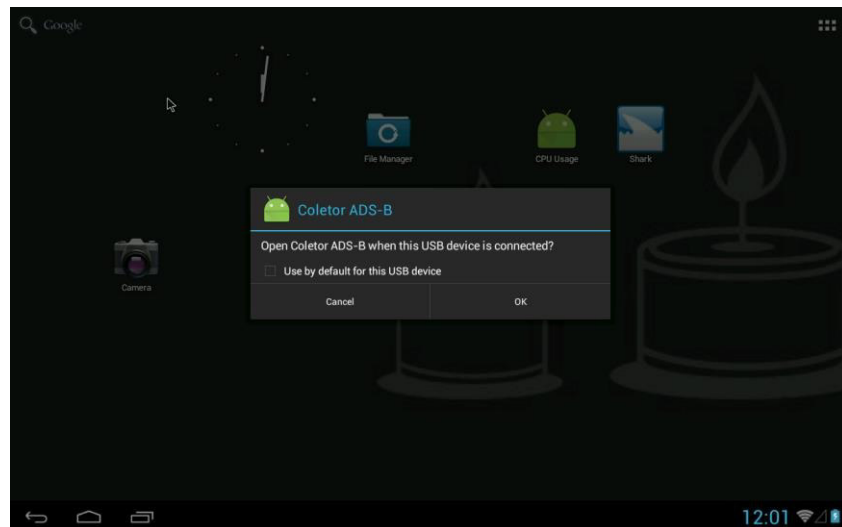
## APÊNDICES

### APÊNDICE A – Aplicativo Android para a coleta de mensagens ADS-B

Esse apêndice demonstra como tirar proveito das principais funcionalidades e presente no aplicativo desenvolvido nesse projeto.

#### 1) Iniciando a aplicação:

A aplicação inicia automaticamente ao inserir o Decodificador ADS-B na entrada USB do dispositivo Android. Imediatamente surge a mensagem na tela abaixo:



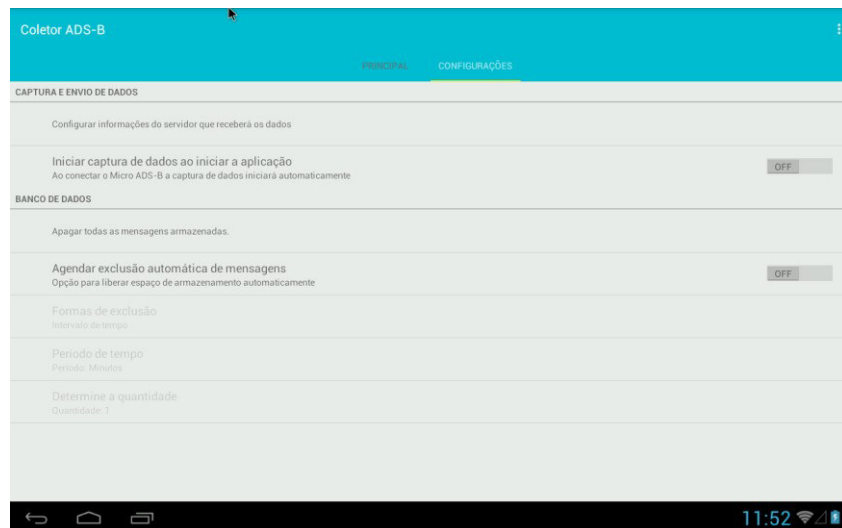
- Essa tela indica que o Sistema Android está pedindo permissão de leitura e escrita na interface USB por ele detectadas
- **DICA:** Marque o checkbox da opção “Use by default for this USB device”, para que ela não apareça todas as vezes que o Decodificado ADS-B seja conectado ao device.
- Ao clicar em “OK” a tela principal da aplicação é aberta.



- Para iniciar a captura das mensagens basta clicar no botão “+” no canto inferior direito da tela.

## 2) Configurando a aplicação

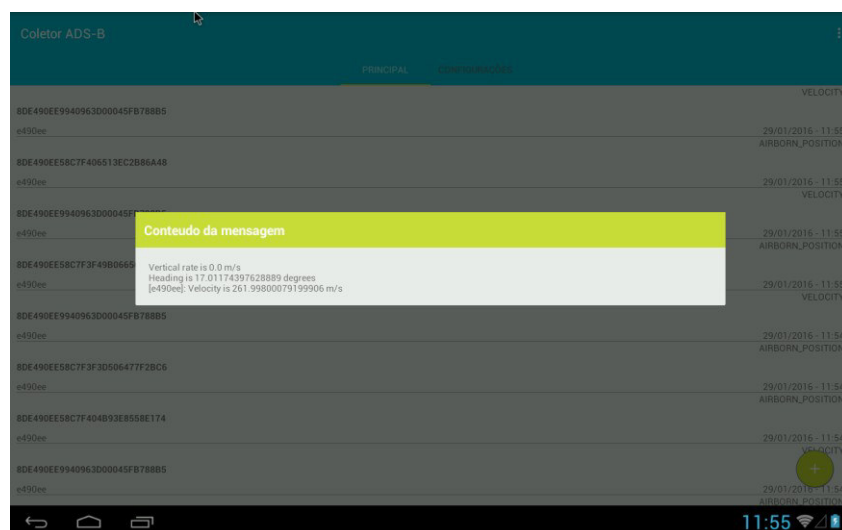
- A tela abaixo representa a tela de configurações da aplicação



- Inicialmente é necessário configurar o endereço IP e porta a qual as mensagens são destinadas, para isso basta ir na opção “Configurar informações do servidor que receberá os dados”, e então uma será oferecida uma tela para preenchimento das informações.

## 3) Vendo o conteúdo das mensagens capturadas

- Para visualizar o conteúdo de uma mensagem basta clicar sobre qualquer uma mensagem que estiver sendo exibida na tela principal. Nota-se que a listagem além de trazer a *string* das mensagens também é possível visualizar qual o tipo de mensagem, a data da mensagem e a qual voo ela pertence.



- Na imagem abaixo é possível visualizar a uma mensagem de localização

