Sensoriamento Espectral via Modo Adaptativo de Saltos em Frequência do Padrão Bluetooth

Elivander J. T. Pereira, Dayan A. Guimarães e Caroline S. Fonseca

***Resumo*— Na técnica de saltos em frequência adaptativa (*adaptive frequency hopping*, AFH) do padrão Bluetooth, canais sob interferência são excluídos do padrão de saltos, buscando evitar a queda na qualidade de serviço do sistema. Neste artigo apresenta-se uma prova de conceito de um módulo de sensoriamento espectral de baixo custo e baixa complexidade implementado a partir do modo AFH do Bluetooth, com microcontroladores ESP32. Demonstra-se que a proposta tem potencial de ser explorada para a implementação de sensores de espectro que podem ser conectados a dispositivos da Internet das coisas *(Internet of things*, IoT), visando ao massivo monitoramento de lacunas espectrais em tempo real de forma a dar suporte à operação de rádios cognitivos.**

***Palavras-Chave*— AFH, Bluetooth, mapa de canais, rádio cognitivo, saltos em frequência, sensoreamento espectral.**

***Abstract*— In the adaptive frequency hopping (AFH) technique used in the Bluetooth standard, channels under interference are excluded from the hopping list, as an attempt to avoid a drop in the system's quality of service. This article presents the proof of concept of a low cost and low complexity spectrum sensor that makes use of the Bluetooth AFH mode, with ESP32 microcontrollers. It demonstrates that the technique has the potential to be explored for the implementation of spectrum sensors that can be connected to IoT devices, aiming at the massive monitoring of spectral holes (whitespaces) in order to support the operation of cognitive radios.**

***Keywords*— Adaptive frequency hopping, Bluetooth, channel map, cognitive radio, spectrum sensing.**

I. Introdução

A quinta geração (*fifth generation*, 5G) das redes de telecomunicações está começando a ser implantada em vários países, mas a escassez de bandas de radiofrequência (*radio-frequency*, RF) livres é um notável empecilho à sua expansão. Isto se deve ao fato de que a política de alocação fixa de espectro hoje adotada pelos órgãos reguladores tem levado à utilização ineficiente do espectro de RF pelos sistemas de comunicação sem fio. Nesta forma de alocação, apenas o usuário detentor da licença de uso do espectro, chamado de usuário primário (*primary user*, PU), tem direito de ocupação de determinada banda. Entretanto, sabe-se que os terminais de comunicação dificilmente encontram-se a transmitir ininterruptamente ou em toda a extensão territorial, levando à existência de bandas desocupadas, denominadas na literatura de buracos espectrais (*spectral holes*) ou espaços em branco (whitespaces). Com isso cria-se a oportunidade de implantação de uma política de alocação dinâmica de espectro, na qual se permite que usuários secundários (*secondary users*, SUs) utilizem bandas de RF de forma oportunista, quando as mesmas se encontrarem ociosas, melhorando assim a eficiência do uso do espectro [1], [2].

O conceito de rádio cognitivo (*cognitive radio*, CR) surgiu como uma possibilidade de implementação da política de alocação dinâmica de espectro [3]. Um CR é um dispositivo transceptor inteligente que pode ter seus parâmetros de comunicação reconfigurados via software [4]. Dentre os atributos de um CR encontra-se o sensoriamento espectral, que é capaz de permitir que um CR monitore o espectro de RF, decida sobre o estado de ocupação dos canais sensoriados e transmita nas lacunas espectrais sem causar interferência na rede primária.

Para que um CR, na condição de SU, realize o sensoriamento espectral ele deve ter parte de seu hardware e de sua capacidade de processamento destinadas a este fim, o que eleva a sua complexidade, seu custo e seu consumo de energia, podendo até impactar na sua portabilidade devido ao aumento de dimensões físicas ou a necessidade de ser equipado com antenas próprias à tarefa em questão. Somam-se ao sensoriamento espectral todas a demais funcionalidades cognitivas previstas para um CR, o que permite conjecturar que tal tecnologia não terá adoção comercial em larga escala, em curto prazo, talvez podendo ser viabilizada por completo apenas em sistemas além da quinta geração.

O que se aborda neste artigo é parte de uma pesquisa mais abrangente na qual se propõe o uso de uma rede de dispositivos de Internet das coisas *(Internet of things*, IoT) como rede de suporte para sensoriamento espectral, eliminando a necessidade de implementação de tal funcionalidade nos CRs, assim reduzindo sua complexidade e viabilizando antecipar a expansão da política de acesso dinâmico ao espectro (*dynamic spectrum access*, DSA). O artigo apresenta o desenvolvimento de uma prova de conceito de um sistema de sensoriamento espectral de baixo custo e baixa complexidade, implementado a partir do modo de saltos em frequência adaptativo (*adaptive frequency hopping*, AFH) do padrão Bluetooth, fazendo uso de microcontroladores ESP32. Demonstra-se que a proposta tem potencial para ser explorada na implementação de sensores de espectro que podem ser conectados a dispositivos IoT, visando ao massivo monitormento de lacunas espectrais de forma a dar suporte à operação de rádios cognitivos por meio do DSA.

O restante do artigo está assim organizado: a Seção II resume alguns conceitos relacionados ao sensoriamento espectral. Na Seção III é descrita a proposta de uso do modo AFH como forma de sensoriamento. A Seção IV apresenta a prova de conceito da proposta e a Seção V conclui do trabalho.

Elivander J. T. Pereira, Dayan A. Guimarães e Caroline S. Fonseca, Instituto Nacional de Telecomunicações - Inatel, Santa Rita do Sapucaí, MG, e-mails: elivander@mtel.inatel.br; dayan@inatel.br; carolinefonseca@gec.inatel.br. Este trabalho foi parcialmente financiado pela RNP, com recursos do MCTIC, processo No 01250.075413/2018-04, sob o projeto Centro de Referência em Radiocomunicações (CRR) do Inatel.

II. Sensoriamento espectral

Tipicamente, a operação do acesso dinâmico ao espectro prevê que a rede secundária possua um intervalo para a realização do sensoriamento espectral e outro para a comunicação, de forma não simultânea. O sensoriamento espectral pode ser realizado basicamente de duas formas: independente (ou individual), ou cooperativo. No sensoriamento independente, como o termo sugere, cada CR realiza o monitoramento do espectro de forma independente dos demais CRs. No sensoriamento cooperativo há a participação de vários CR no processo, valendo-se da diversidade espacial existente entre os rádios almejando reduzir a probabilidade de erros de detecção do sinal primário, assim aumentando a vazão da rede secundária e minimizando as chances de ocorrência de interferência na rede primária.

O sensoriamento cooperativo pode ser distribuído, caso em que os CRs trocam informações sobre o estado de ocupação da banda sensoriada e em seguida decidem por meio de um consenso, ou centralizado, caso em que as informações de sensoriamento são transmitidas para um centro de fusão (*fusion center*, FC) onde a decisão final é tomada. O FC pode receber diretamente as amostras de sinal colhidas pelos CRs, método chamado fusão de dados, ou, alternativamente, pode receber decisões locais tomadas em cada CR, método chamado fusão de decisões.

Quando se trata da fusão de decisões, parte da complexidade computacional é removida do FC, uma vez que as amostras do sinal primário são pré-processados localmente em cada CR, onde é formada a estatística de teste que será comparada com um limiar para que se decida sobre a presença ou a ausência de sinal na banda sensoriada. Dentre as estatísticas de teste mais conhecidas citam-se aquelas baseadas em detecção de energia, em filtragem casada, em detecção por propriedades cicloestacionárias do sinal primário, ou em autovalores [4], [5]. A decisão global é tomada a partir da aplicação da regra K-em-M, na qual decide-se pela presença do sinal primário na banda sensoriada se ao menos K dentre os M CRs em cooperação assim decidirem.

No cenário de fusão de dados, as amostras do sinal primário recebidas pelos M dispositivos secundários são encaminhadas ao FC, onde é formada uma única estatística de teste, por exemplo dentre aquelas supramencionadas, para posterior decisão global.

Uma vez tomada a decisão global, ela é transmitida a todos os CRs, subsequentemente realizando-se o acesso à banda declarada desocupada por meio de alguma técnica de acesso múltiplo [5].

III. Aplicação do AFH do padrão Bluetooth como forma de sensoriamento espectral

O Bluetooth é um padrão de comunicação sem fio desenvolvido pelo Bluetooth SIG (*Special Interest Group*) para redes WPAN (*wireless personal area network*) operando na banda não licenciada de 2.4 GHz. Sua versão mais recente é a 5.2, apresentada em 6 de janeiro de 2020 [6]. No Bluetooth 5.0 houve a combinação de tecnologias de versões anteriores que caracterizavam dois modos distintos de operação do padrão, o Bluetooth 1.0/2.0 (*basic rate / enhanced data rate*, BR/EDR) e o Bluetooth 4.0 (*Bluetooth low energy*, BLE). As próximas subseções descrevem de forma resumida os aspectos do padrão Bluetooth de maior interesse à proposta do artigo, sendo esta apresentada logo em seguida.

A. Modos BR/EDR e BLE do padrão Bluetooth

Dentre os dois principais modos de operação do padrão Bluetooth, pode-se estabelecer uma divisão entre as versões legadas 1.0 e 2.0, e a versão legada mais recente, a 4.0. A versão 1.0 é conhecida como Bluetooth clássico ou Bluetooth BR. A versão 2.0 é chamada de Bluetooth EDR e a versão 4.0 é nomeada de BLE.

O Bluetooth BR operava utilizando 79 canais de 1 MHz de largura de banda, transmitindo com modulação do tipo GFSK (*Gaussian frequency-shift keying*) e atingia taxas de transmissão da ordem de 1 Mbit/s. Na sua segunda versão, o Bluetooth EDR, o protocolo continuou a utilizar da mesma divisão de canais, porém passou a adotar os esquemas de modulação QDPSK (*quaternary differential phase-shift keying*) e 8DPSK (*differential 8-level phase-shift keying*), o que elevou a taxa de transmissão para a casa dos 3 Mbit/s [7].

O BLE inicialmente não foi um padrão desenvolvido pelo Bluetooth SIG, mas sim em paralelo nos laboratórios da Nokia sob o nome de Wibree. O padrão desenvolvido pela Nokia tinha como objetivo o baixo consumo de energia, o que o tornaria adequado para sensores e dispositivos móveis limitados em energia. Posteriormente, o Bluetooth SIG incorporou o Wibree à especificação 4.0 do Bluetooth, dando origem ao BLE [8]. Nesta versão, a comunicação passou a ser feita por meio de 40 canais de 2 MHz de largura de banda, e não mais nos 79 canais de 1 MHz.

B. A tecnologia AFH do padrão Bluetooth

Para combater a interferência oriunda de outros sistemas de comunicação, muito comum na banda de 2,4 GHz, o Bluetooth SIG incorporou à especificação 1.2 do padrão a tecnologia AFH, a qual utiliza um mapa reconfigurável dos canais alocados ao processo de espalhamento espectral por saltos em frequência (*frequency hopping spread spectrum*, FHSS) adotado na interface aérea do padrão. No FHSS altera-se de forma pseudo-aleatória a frequência central do canal de comunicação entre o dispositivo mestre e o dispositivo escravo. A sequência de saltos em frequência contém 2^{27} saltos que ocorrem 1600 vezes por segundo, resultando em um período de 23,3 horas até que a sequência se repita. Durante a conexão Bluetooth, tanto o transmissor quanto o receptor conhecem o padrão pseudo-aleatório, que é calculado baseado na temporização (*clock*) do dispositivo mestre na piconet [9].

Durante a comunicação entre os dispositivos mestre e escravo, o mapa de canais é criado para ambos os dispositivos. Caso um canal esteja livre de interferência, ele é classificado como disponível, ou seja, é um canal que fará parte da sequência de saltos em frequência, neste caso atribuindo-se o bit 1 ao canal no mapa. Caso seja detectada uma interferência suficientemente alta no canal, ele será classificado como

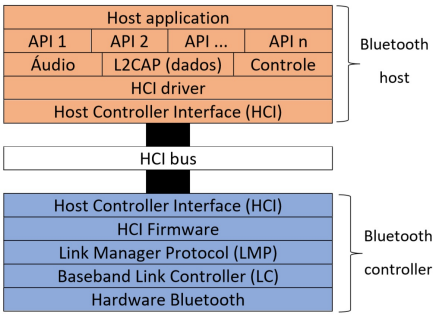

Fig. 1. Camadas da arquitetura do protocolo Bluetooth.

desabilitado, ou seja, o mesmo não fará parte da sequência de saltos, neste caso atribuindo-se a ele o bit 0.

A classificação da qualidade de um canal é normalmente realizada por meio do indicador da potência do sinal recebido (*received signal strength indicator*, RSSI), da taxa de erro de pacote ou de bit (*packet / bit error rate*, PER / BER), da razão sinal-ruído (*signal-to-noise ratio*, SNR), ou de alguma combinação destas métricas [9].

Considerando como exemplo o uso do BLE, os 40 canais de comunicação são enumerados de 0 a 39, sendo que os canais de 0 a 36 são utilizados para a comunicação de dados enquanto os canais 37, 38 e 39 são canais reservados para controle da comunicação, recebendo o nome de *advertising channels*. Os canais de dados são enumerados de forma sequencial no espectro, enquanto os canais de controle são distribuídos de forma que o canal 37 seja o primeiro da banda, o canal 38 seja localizado entre o canal 10 e o canal 11, e por último o canal 39 seja localizado no final do espectro.

A arquitetura do padrão Bluetooth pode ser descrita na forma de camadas [7], como ilustrado na Figura 1. A camada de controle (*Bluetooth controller*) esta relacionada à camada física do dispositivo e ao *firmware* implementado pelo fabricante. A camada de hospedeiro (*Bluetooth host*) contém as ferramentas disponíveis para o sistema operacional fazer uso da tecnologia, como APIs (*application programming interfaces*) para dados, áudio e controle, por exemplo. A camada de *host* é operada dando funcionalidades ao sistema operacional e entregando recursos à camada de aplicação. Uma interface intermediária que faz a ligação entre a camada de controle e a camada de *host* é a chamada *host controller interface* (HCI), que é responsável por prover funções que permitem acesso ao mapa de canais do AFH.

Na prática, o AFH é uma função do protocolo de gerenciamento de enlace (*link manager protocol*, LMP) que é responsável por gerenciar a conexão entre os dispositivos e é parte da camada de controle. Dados e funções pertencentes às funções desta camada inicialmente não foram planejados para terem uso por camadas superiores ou em quaisquer aplicações, uma vez que se referem à conexão entre os dispositivos Bluetooth. Entretanto, a HCI provê ferramentas para, se necessário, acessar dados da camada física.

C. Aplicação do AFH para sensoriamento espectral

Basicamente, o sensoriamento espectral através do modo AFH do padrão Bluetooth consiste em realizar, em um ambiente controlado, a leitura do mapa de canais dos saltos em frequência. Como ilustra a Figura 2, dois dispositivos Bluetooth são mantidos conectados por cabo de forma a manter a conexão livre de interferência externa. Ao enlace é adicionado um combinador ligado à uma antena responsável por receber o sinal que se deseja detectar, utilizando-o como fonte de interferência na comunicação Bluetooth.

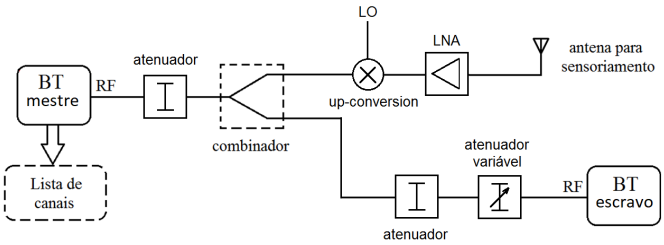

Fig. 2. Configuração de hardware para sensoriamento via Bluetooth.

O sinal sensoriado recebido pela antena é amplificado por um amplificador de baixo ruído (*low noise amplifier*, LNA) e tem sua frequência transladada para a faixa de frequências de operação dos transceptores Bluetooth (2.4 GHz), caso tal sinal esteja fora desta faixa. A translação é realizada pelo batimento (multiplicação) do sinal de saída do LNA pela portadora senoidal gerada pelo oscilador local (*local oscillator*, LO). O sinal resultante do batimento é acoplado ao enlace Bluetooth por meio do combinador (*splitter* ou *combiner*). Atenuadores estão presentes no sistema para que o nível de potência da comunicação dos dispositivos Bluetooth (BT mestre e BT escravo) seja colocado em um limiar próximo ao de classificação do canal como desabilitado, de forma que o menor sinal interferente recebido pela antena de sensoriamento seja capaz de alterar a classificação de um ou mais canais no mapa do AFH. A leitura do mapa é então feita através de uma função específica da HCI, permitindo interpretar os canais desabilitados como a porção do espectro em uso pela rede primária, e os canais habilitados como faixas de frequências ociosas.

O sensoriamento espectral através do AFH não permite utilizar vários dispositivos para uma fusão de dados no cenário cooperativo, uma vez que a consulta do mapa de canais no dispositivo mestre irá reportar um dado binário referente a canais livres e ocupados. Entretanto, pode-se valer de vários módulos de sensoriamento como aquele ilustrado na Figura 2, voltados para a implementação do sensoriamento espectral cooperativo com fusão de decisões a fim de aumentar a precisão de detecção em relação ao sensoriamento não cooperativo. Como trata-se de um sistema de baixo custo comparado aos hardwares atualmente utilizados para sensoriamento espectral, almeja-se atingir as métricas de desempenho por meio da alta densidade e abrangente distribuição dos dispositivos, em vez de alta qualidade individual no sensoriamento, o que se pode conseguir com o acoplamento de módulos de sensoriamento a dispositivos IoT. Esta medida visa prover um amplo sistema de sensoriamento espectral, com elevada resolução espacial e abrangência territorial. Os resultados do sensoriamento podem então alimentar bases de dados de ocupação espectral, as quais podem ser consultadas pela rede secundária, eliminando a

necessidade de se ter terminais secundários equipados com a funcionalidade de sensoriamento espectral.

IV. Prova de conceito

Para realização da prova de conceito foram utilizados, como dispositivos Bluetooth, dois módulos SoC (*system-on-a-chip*) ESP32 produzidos pela empresa Espressif Systems [10]. Além destes, foi utilizado um equipamento de rádio definido por software Ettus USRP B210 como gerador de sinal interferente na comunicação entre os ESP32, bem como um computador para a leitura do mapa de canais do AFH via comunicação serial. A Figura 3 apresenta os principais dispositivos utilizados, bem como suas interconexões.

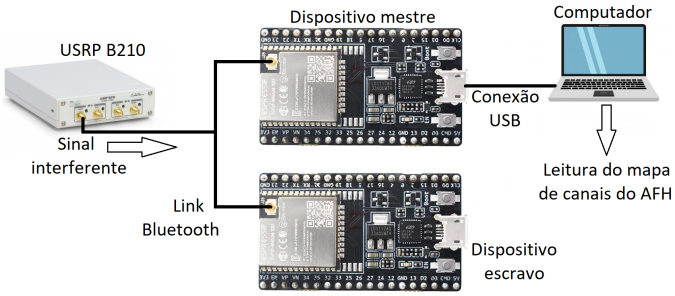

Fig. 3. Prova de conceito do sistema de sensoriamento espectral via modo AFH do padrão Bluetooth.

O ESP32 é um microcontrolador de baixo custo com diversas funcionalidades já integradas, diversos pinos de entrada e saída analógica e digital, além de interfaces de conexão Bluetooth e WiFi. Suas várias versões podem ser encontradas no mercado pelo preço unitário de menos de 10 dólares americanos. A versão escolhida para este projeto foi o ESP32-WROOM-32U, que oferece conectores Hirose U.FL em vez de antenas integradas, permitindo ligar os dois dispositivos via cabo, como prevê a Figura 2. A Tabela I lista algumas das especificações do ESP32-WROOM-32U.

TABELA I
Algumas especificações do ESP32-WROOM-32U.

Item	Especificação
Memória Flash SPI	4 MB
Cristal integrado	40 MHz
Antena	Conector U.FL
WiFi	802.11 b/g/n
Bluetooth	versão 4.2 compatível com BR/EDR e BLE
Sensibilidade de recepção (Bluetooth)	-97 dBm
Potência de transmissão (Bluetooth)	Controlável de -12 a $+9$ dBm
Tensão de alimentação	3.0 a 3.6 V dc
Corrente mínima	500 mA

Como a sensibilidade do ESP32 é elevada, a potência de transmissão foi configurada para seu valor mínimo de -12 dBm. Além disso, uma atenuação total de 30 dB foi inserida no enlace de comunicação para manter os estados dos canais próximos do limiar de classificação, como desabilitados, e para evitar danos aos ESP32 que poderiam ser causados por elevada potência do sinal interferente gerado pela USRP.

Para a programação foram utilizadas as linguagens C/C++ com as ferramentas providas pelo fabricante, chamadas de *ESP IDF tools* [11]. O sistema operacional de tempo real (*real time operational system*, RTOS) embarcado foi o FreeRTOS [12] e a pilha (*stack*) do protocolo Bluetooth utilizada para camada de *host* foi o Apache NimBLE [13].

Apesar de as ferramentas de programação do ESP32 serem em grande parte de código aberto, a camada de controle do protocolo Bluetooth é de código fechado e proprietário do fabricante do chip, não sendo, portanto, passível de substituição ou alterações nesta parte do *firmware*.

A programação do código compilado em ambos dispositivos foi feita partindo de duas aplicações de exemplo existentes na biblioteca do NimBLE. Para o dispositivo mestre o código base utilizado foi o do exemplo *blecent* e para o escravo o código base foi o do exemplo *blephpr*. Como o FreeRTOS permite que se executem tarefas (*tasks*) simultâneas no microcontrolador, foi adicionada uma aplicação no código para manipulação dos dados lidos do mapa de canais e exibição serial dos mesmos na tela do computador. A arquitetura de código utilizado na programação do dispositivo mestre é apresentada na Figura 4.

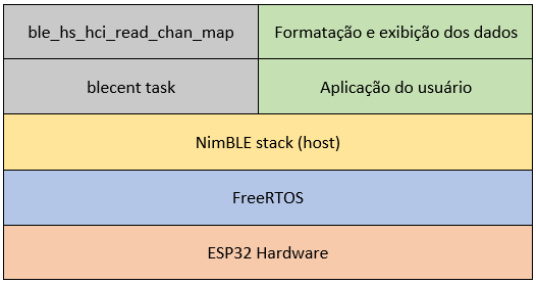

Fig. 4. Arquitetura de código no dispositivo mestre utilizado nos testes.

Na pilha de *host* NimBLE para implementação do protocolo Bluetooth existe uma função da camada HCI que permite a leitura do mapa de canais do AFH. Esta função é utilizada através da tarefa responsável pela comunicação do dispositivo, a qual é baseada nos dois exemplos anteriormente citados. A mesma pode ser utilizada sob a sintaxe:

```
return=ble_hs_hci_read_chan_map(conn_handle,
output),
```

cujos principais campos são assim descritos:

- `return`: é o comando de retorno da função. A função de leitura do mapa de canais é uma função do tipo `int`, cujo valor inteiro retornado indica o estado da requisição. Caso retorne zero, indica que a leitura do mapa de canais foi bem sucedida; caso contrário, a função foi negada ou não pôde ser executada por causa de algum erro como, por exemplo, nenhuma conexão ativa entre os dispositivos.
- `conn_handle`: informa de qual conexão se deseja fazer a leitura do mapa de canais, uma vez que pode haver mais

de uma conexão ativa. Esta variável é do tipo `uint16_t`.

- `output`: é o parâmetro de saída da função que receberá o mapa de canais. Trata-se de um vetor de 5 posições do tipo `uint8_t`, ou seja, um vetor onde cada posição representa 8 bits e cada bit representa o respectivo canal. A primeira posição representa os canais de 0 a 7, e a segunda posição representa os canais de 8 a 15, e assim até o último bit da última posição, o qual representa o canal 39.

De forma a facilitar a leitura do mapa de canais na variável de saída durante o monitoramento via interface serial do computador, uma função com execução como segunda tarefa no sistema operacional foi criada para organizar os dados existentes em um vetor de 40 posições no qual o índice da posição coincide com o índice do canal sensoriado. Para cada um dos 5 Bytes da variável de saída aplica-se uma máscara a fim de identificar se o bit de interesse é 1 ou 0, o que é feito para os 8 bits de cada um dos 5 bytes de forma que o retorno desta função indexe corretamente cada canal. Os canais 37, 38 e 39, como antes mencionado, são canais de controle da comunicação e, por isso, sempre constarão com o valor 0 no mapa de canais, uma vez que os mesmos não participam do processo de AFH.

Durante os testes constatou-se que a forma como o AFH foi implementado no controlador do Bluetooth do ESP32 não gerava o mapa de canais da maneira esperada, em concordância com os canais afetados de forma controlada pela USRP. Por ser uma parte de código fechado do *firmware*, não foi possível alterar o método utilizado para classificação dos canais como ocupados. Os mesmos alteravam de forma muito rápida, não sendo possível obter dados confiáveis nas requisições feitas. Para contornar o problema criou-se um esquema de janela temporal, onde o mapa de canal final após um período era calculado através de um E-lógico entre todas as requisições feitas durante aquele tempo. Dessa forma foi possível detectar sinais gerados pela USRP na banda de opeção do Bluetooth, os quais podiam ser visualizados através do *prompt* de comando do Windows. A Figura 5 exemplifica, na linha superior, o caso da leitura dos canais sem a detecção de nenhuma interferência em toda a banda. Na segunda linha, quatro canais no centro da banda foram desabilitados no mapa de canais do AFH por estarem sob interferência, neste caso um ruído de 8 MHz de largura de faixa foi adicionado ao enlace de comunicação.

```
Channel map: 1111111111111111111111111111111111111000
Channel map: 1111111111111111110000111111111111111000
```

Fig. 5. Exemplo de medições do mapa de canais do AFH.

No entanto, era esperado que outros hardwares pudessem entregar similar leitura sem a necessidade da criação da janela temporal de leituras do mapa de canais, fato constatado em testes preliminares utilizando um *smartphone* Android como mestre e o ESP32 como escravo. Uma vez que quem gera o mapa de canais e o retransmite é o dispositivo mestre, dispensa-se a necessidade de nova programação para leitura do mapa em cada tipo de hardware que se almeja avaliar. Utilizando-se de um ESP32 compilado com o código baseado no exemplo *blephpr*, pode-se alternar o dispositivo mestre com flexibilidade, sem necessidade de alterações no código caso se mantenha o uso do BLE. Uma segunda etapa de prototipagem e avaliação da prova de conceito começou a ser desenvolvida utilizando-se do Raspberry Pi 3 e a pilha BlueZ do sistema operacional Linux. O BlueZ é a pilha oficial do protocolo Bluetooth para Linux, a qual permite leitura do mapa AFH via comando pelo console do sistema operacional. A partir do Raspberry Pi 3, usando de uma versão do Linux, o Raspibian, é possível conectar diferentes dispositivos como terminal escravo sem a necessidade de programação individual, com código customizado de acordo com a pilha Bluetooth mais recomendada pelo fabricante. Outra opção existente consiste em migrar a implementação para o Bluetooth BR/EDR, contudo perdendo-se a flexibilidade acima citada.

V. Conclusão

O modo AFH do padrão Bluetooth foi proposto para sensoriamento espectral. Uma prova de conceito foi avaliada com o uso de um SoC de baixo custo, o ESP32. Resultados indicaram que é possível detectar sinais primários interferentes na banda de interesse, através do mapa de canais do AFH. Entretanto, devido ao código fechado do *firmware* da camada de controle do Bluetooth do ESP32, foram iniciados novos testes baseados no Raspberry Pi 3 e na pilha BlueZ do Linux.

Referências

[1] F. C. Commission., *Spectrum Policy Task Force Report*. Federal Communications Commission, Spectrum Policy Task Force, 2002. [Online]. Available: https://books.google.com.br/books?id=p9MQtwAACAAJ

[2] M. A. McHenry, P. A. Tenhula, D. McCloskey, D. A. Roberson, and C. S. Hood, "Chicago spectrum occupancy measurements & analysis and a long-term studies proposal," in *Proceedings of the first international workshop on Technology and policy for accessing spectrum*. ACM, 2006, p. 1.

[3] J. Mitola and J. Maguire, G.Q., "Cognitive radio: making software radios more personal," *Personal Communications, IEEE*, vol. 6, no. 4, pp. 13–18, Aug 1999.

[4] Y. Arjoune and N. Kaabouch, "A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions," *Sensors*, vol. 19, no. 1, p. 126, 2019.

[5] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Physical communication*, vol. 4, no. 1, pp. 40–62, 2011.

[6] "Bluetooth special interest group - bluetooth 5.2 feature overview," Disponível em: https://www.bluetooth.com/wp-content/uploads/2020/01/Bluetooth_5.2_Feature_Overview.pdf, acesso em: 08/06/2020.

[7] "National instruments - introduction to bluetooth device testing," Disponível em: https://download.ni.com/evaluation/rf/intro_to_bluetooth_test.pdf, acesso em: 08/06/2020.

[8] "Nokia - wibree forum merges with bluetooth sig," Disponível em: http://www.wibree.com/wp-content/uploads/Wibree_pressrelease_final_1206.pdf, acesso em: 08/06/2020.

[9] "Rohde & schwarz - bluetooth adaptive frequency hopping on a R&S CMW," Disponível em: https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_application/application_notes/1c108/1C108_0e_Bluetooth_BR_EDR_AFH.pdf, acesso em: 08/06/2020.

[10] "Espressif Systems company description," Disponível em: https://www.espressif.com/en, acesso em: 16/06/2020.

[11] "Standard setup of toolchain for windows," Disponível em: https://docs.espressif.com/projects/esp-idf/en/latest/get-started/windows-setup.html#get-started-windows-tools-installer, acesso em: 17/10/2019.

[12] "Free RTOS - market leading RTOS," Disponível em: https://www.freertos.org/, acesso em: 08/06/2020.

[13] "Apache - mynewt-nimble," Disponível em: https://github.com/apache/mynewt-nimble, acesso em: 08/06/2020.