Nadir Bernardo Bianchini

Estabelecimento de um Enlace Ponto a Ponto IEEE 802.11 de Alta Vazão

São José-SC

Agosto/2016

Nadir Bernardo Bianchini

Estabelecimento de um Enlace Ponto a Ponto IEEE 802.11 de Alta Vazão

Monografia apresentada à Coordenação do Curso Superior de Tecnologia em Sistemas de Telecomunicações do Instituto Federal de Santa Catarina para a obtenção do diploma de Tecnólogo em Sistemas de Telecomunicações.

Orientador:

Prof. Marcelo Maia Sobral, Dr. Eng.

Curso Superior de Tecnologia em Sistemas de Telecomunicações

Instituto Federal de Santa Catarina

São José-SC

Agosto/2016

Monografia sob o título *"Estabelecimento de um Enlace Ponto a Ponto IEEE 802.11 de Alta Vazão"*, defendida por *Nadir Bernardo Bianchini* e aprovada em 22 de Agosto de 2016, em São José, Santa Catarina, pela banca examinadora assim constituída:

Prof. Marcelo Maia Sobral, Dr. Eng.
Orientador

Prof. Odilson Tadeu Valle, Dr. Eng.
IFSC

Prof. Tiago Semprebom, Dr. Eng.
IFSC

Resumo

O padrão IEEE 802.11 possui um protocolo MAC do tipo aleatório e com disputa. Este tipo de MAC define mecanismos de acesso ao meio que são utilizados para reduzir a probabilidade de perda de dados, em uma rede local sem fio, onde há múltiplas estações com chances de transmissões simultâneas. O protocolo foi projetado para tratar de redes multiacesso, porém quando a conexão é estabelecida por somente duas estações o meio poderia ser melhor aproveitado, pelo fato de as chances de colisões serem menores. É nesse cenário que buscou-se o desenvolvimento de uma maneira alternativa de acesso ao meio, definindo um protocolo específico para um enlace ponto a ponto em uma conexão IEEE 802.11. O mecanismo propõe uma comunicação, entre os dois nodos, controlada, temporizada e previsível, reduzindo ao máximo os parâmetros que atrasam uma transmissão empregados pelo protocolo e proporcionando aos dispositivos conectados uma melhor vazão e aproveitamento do meio.

Abstract

The IEEE 802.11 standard has a MAC protocol with competition and random type. This kind of MAC defines medium access mechanisms that are used to reduce the probability of data loss, in a wireless local area network, where it has multiple stations with chances of simultaneous transmission. This protocol was designed to treat a multipoint access. However, when the connection is established by only two stations, the medium could be better used, by the fact that the chances of collision are smaller. In this context, we seek to develop an alternative way to access the, defining a specific protocol for a point to point link in an IEEE 802.11.The mechanism offer a communication between two nodes, controlled, timed and predictable, while minimizing the parameters that cause a delay transmission employed by the protocol, and providing to the connected devices a better throughput and use of the medium.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

AIFS *Arbitration Inter-Frame Spacing*

AP *Access Point*

API *Application Program Interface*

CSMA/CA *Carrier Sense Multiple Access With Collision Avoidance*

DCF *Distributed Coordination function*

EDCA *Enhanced Distributed Channel Access*

FDM *Frequency Division Multiplexing*

GTS *Guaranteed Time Slot*

IEEE *Institute of Electrical and Electronics Engineers*

ioctrl *input/output control*

LLC *Logical Link Control*

MAC *Control Acess Medium*

MEF Máquina de Estado Finito

NOACK *No Acknowledgement*

PCF *Point Coordination Function*

PDU *Protocol Data Unit*

profs *proc filesystem*

PTP *Point to Point*

UDP *User Datagram Protocol*

USB *Universal Serial Bus*

TDM *Time Division Multiplexing*

WLAN *wireless Local Area Network*

1 Introdução

As redes IEEE 802.11 têm se destacado por prover soluções de conectividade em locais distantes e remotos. Muitos provedores fornecem esse serviço através de rádios *outdoor*, transmitindo o sinal via sem fio. Com antenas apropriadas e uma boa potência, segundo alguns fabricantes tais como Ubiquiti[1] e Deliberant[2], pode-se conseguir uma transmissão de quilômetros de distância onde não é possível a passagem de cabo e a implantação de fibra óptica teria um custo elevado. Dessa forma, necessitam de rádios que precisam implementar enlaces com a maior taxa de dados possível em uma grande distância. Por questões de custo e disponibilidade tecnológica, muitos desses rádios são na verdade pontos de acesso ou roteadores sem fio baseados no padrão IEEE 802.11.

A tecnologia utilizada para estabelecer este tipo de enlace ponto a ponto possui limitações que diminuem o desempenho desse tipo de conexão. Uma consequência é a baixa vazão causada pelos mecanismos de prevenção de colisões do protocolo MAC *(Control Acess Medium)* CSMA/CA *(Carrier Sense Multiple Access With Collision Avoidance)*. O CSMA/CA é o método definido pelo padrão IEEE 802.11 para acesso ao meio de transmissão em redes sem fio. O motivo do baixo desempenho em um enlace ponto a ponto de grande distância é que este adota mecanismos para controlar o acesso ao meio nos dispositivos que desejam transmitir, fazendo com que o meio torne-se ocioso uma parte significativa do tempo com o intuito de prevenir colisões.

Fabricantes como Ubiquiti e Deliberant, desenvolveram protocolos MAC proprietários que visam um aumento na vazão atingida pelo equipamento, além de controle de ruído e organização do tráfego de rede, porém tais soluções são fechadas e por isso não são interoperáveis.

Neste trabalho propõe-se o desenvolvimento de um protocolo MAC baseado no padrão IEEE 802.11. Dessa forma, junto com o ajuste dos mecanismos que tentam prevenir a colisão e temporização entre quadros, apesar de ser construído sobre o MAC CSMA/CA, esse novo MAC

[1] `https://www.ubnt.com/`
[2] `https://www.deliberant.com/`

deve alterar a forma de acesso ao meio para a transmissão de mensagens. Com isso, pretende-se obter um melhor aproveitamento do meio no enlace e, por consequência, uma melhor vazão.

1.1 Objetivo geral e específicos

O objetivo geral deste trabalho é estabelecer um protocolo para enlace ponto a ponto de alta vazão, o qual deve se basear em tecnologia IEEE 802.11.
Os objetivos específicos são:
- Desenvolver um protocolo MAC livre de disputa, ou seja, o controle do acesso ao meio é feito por mensagens de controle e não de forma aleatória, a ser construído sobre o MAC CSMA/CA na versão definida pelo método de acesso EDCA *(Enhanced Distributed Channel Access)* do padrão IEEE 802.11e.
- Desenvolver um ponto de acesso capaz de estabelecer o enlace ponto a ponto com o novo MAC.

1.2 Organização de texto

A organização textual está descrita da seguinte forma:
Capítulo 2: Este Capítulo inicia com uma breve passagem sobre o padrão IEEE 802.11. Também mostra como é o acesso ao meio utilizado por esse padrão, o funcionamento de cada um de seus mecanismos e o porque são implementados. Nas seções decorrentes, há a apresentação do EDCA e suas importantes características que foram exploradas na implementação do projeto, e a menção dos diferentes tipos de protocolo de acesso ao meio de transmissão e seus respectivos funcionamento. Por fim, segue a apresentação do chipset da Atheros, utilizado no desenvolvimento do trabalho, e sua compatibilidade com a ferramenta Netlink e o Sistema Operacional Linux, seguido da descrição do *Packet Socket*.
Capítulo 3: O Capítulo 3 apresenta o modelo do protocolo desenvolvido, junto com a abordagem para a criação de um protótipo que se integre ao subsistema de rede do Linux.
Capítulo 4: São apresentados os cenários de testes junto com os resultados obtidos, tanto em uma conexão IEEE 802.11 padrão quanto com a implementação do protótipo do MAC em duas interfaces sem fio, para assim comentar sobre o desempenho dos resultados.
Capítulo 5: Apresenta as conclusões obtidas do trabalho realizado e algumas propostas de ajustes que podem ser executados para adquirir resultados ainda melhores.

2 *Fundamentação teórica*

Este Capítulo aborda alguns conceitos sobre o padrão IEEE 802.11, uma apresentação do adendo IEEE 802.11e que é definido por regras do EDCA para controle de acesso ao canal, os protocolos de controle de acesso ao meio, algumas especificações sobre o chipset da Atheros, a descrição da API Netlink e as funcionalidades do *Packet Socket*. Tais informações se fazem necessárias para um bom entendimento do trabalho.

2.1 Padrão IEEE 802.11

Desenvolvido pelo IEEE *(Institute of Electrical and Eletronics Engineers)*, o padrão 802.11, conhecido também como Wi-Fi, caracteriza um conjunto de normas afim de prover o estabelecimento e uso de rede local sem fio (WLANs).

A rede IEEE 802.11 é uma tecnologia que está em constante crescimento e desenvolvimento, por motivos de facilidade de implantação, a comunicação com os dispositivos Wi-Fi é feita através de radiofrequência, dessa forma não é necessário o uso de cabos, é capaz de fornecer acesso de qualquer localização onde existir sinal e para usuários móveis, tudo isso pode ser considerado de baixo custo.

A arquitetura para as WLANs do IEEE 802.11 é definida com o propósito de abranger uma Camada de Controle de Acesso ao Meio (MAC) e uma Camada Física (PHY). A Figura 2.1, faz uma relação entre o padrão IEEE 802.11 com o modelo de redes de computadores, o Modelo OSI, onde cada Camada apresenta protocolos com funcionalidades diferentes. No caso do IEEE 802.11, os protocolos são implementados na Camada 2 e na Camada 1, Enlace de dados e Física respectivamente.

A Camada de Enlace é dividida em duas Subcamadas: LLC *(Logical Link Control)* responsável pela comunicação dos protocolos de rede com as Camadas superiores, o que equivale a um protocolo de enlace, porém utilizado com pouca frequência, e a Subcamada MAC, que

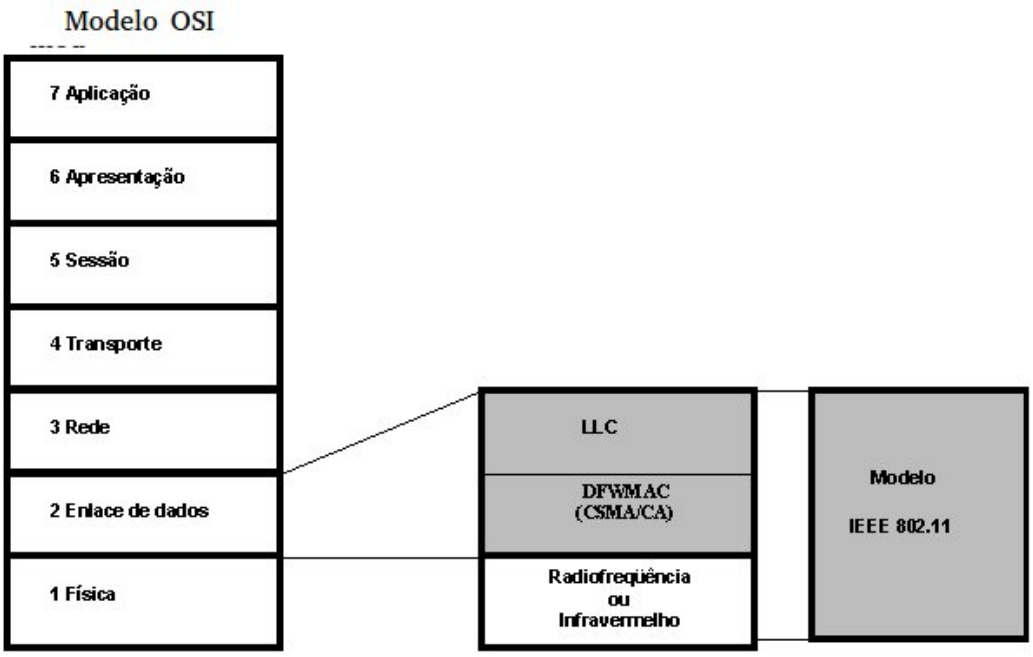

Figura 2.1: Relação do modelo OSI com o padrão IEEE 802.11. [8].

será relata na sub-seção seguinte.

A Camada Física, especifica o meio aonde os dados irão trafegar. No caso do protocolo IEEE 802.11, segundo [6], é definido uma série de adendos como por exemplo: 802.11a/b/g/n, que diferem-se na frequência utilizada, na largura de banda, na modulação e na filtragem em função da codificação do canal, o que resulta em diferentes taxas de transmissão e diferentes codificações. A Figura 2.2 representa a Camada de Enlace de dados e a Camada Física com alguns adendos IEEE 802.11.

2.1.1 Protocolo de Controle de Acesso ao Meio (MAC)

O MAC, é um protocolo responsável pelo acesso ao meio de transmissão. Esse protocolo é feito para trabalhar em um meio compartilhado, para assim evitar problemas de transmissão como os conflitos de acesso.

O MAC possui algumas funções fundamentais na comunicação entre as estações:
- Definição de um formato de quadro: As PDUs *(Protocol Data Unit)* vindas da Camada superior são encapsuladas no formato de quadro especificado na Figura 2.3.

- Endereçamento das estações: É necessário, pois o meio de comunicação é comparti-

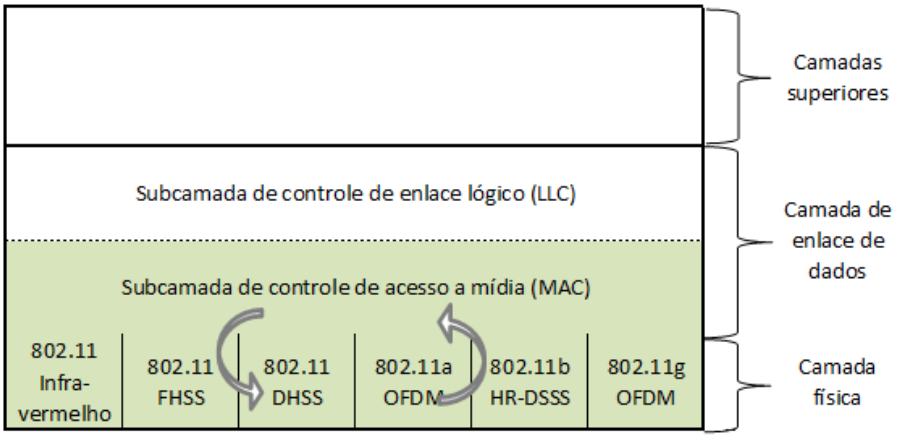

Figura 2.2: Camada de Enlace e Camada Física. [10].

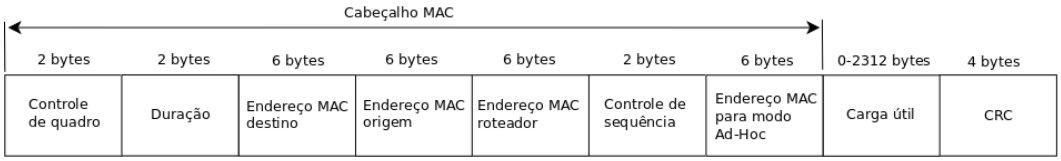

Figura 2.3: Formato de quadro MAC IEEE 802.11. (Fonte: Elaborada pelo autor).

lhado. A Figura 2.3, mostra os campos de endereçamento.

- Coordena controle de acesso ao meio para efetuar a transmissão de quadros: Impõe estratégias para que uma estação obtenha acesso ao meio afim de transmitir um quadro, e para tratar colisões, caso ocorram.

Há dois mecanismos de controle de acesso ao meio especificados pelo padrão IEEE 802.11, que são as chamadas funções de coordenação. O DCF *(Distributed Coordination Function)* é obrigatório, onde todas as estações transmitem de forma independente, ou seja, competem de forma autônoma pelo meio. O PCF *(Point Coordination Function)* é opcional, onde uma estação irá controlar o acesso das demais através de quadros de controle.

2.1.2 Protocolo CSMA/CA

De acordo com [6], o MAC CSMA/CA, mostrado na Figura 2.4, é o método utilizado pelo DCF para controle de acesso ao meio de transmissão, em uma rede sem fio, definido pelo padrão IEEE 802.11. Ele implementa acesso com contenção, visando reduzir as chances de colisões entre os quadros que são enviados por nodos (estações) da mesma rede ou redes

vizinhas, que estão dentro da área de cobertura do transmissor. Assim, as chances de pacotes serem perdidos e não chegarem ao destino final é reduzida. Este protocolo foi projetado para um acesso multiponto, onde várias estações acessam o meio de forma aleatória.

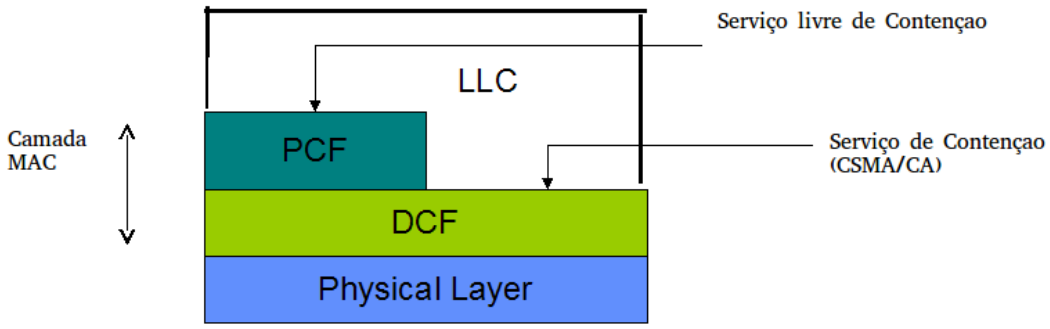

Figura 2.4: DCF na Subcamada MAC. [11].

Em uma rede sem fio, como não é viável escutar o meio e transmitir ao mesmo tempo, então não é possível verificar que uma colisão ocorreu, ou seja, as colisões não são detectadas, segundo [1]. E quando uma transmissão é iniciada ela não é bloqueada, por estes motivos, a estação transmissora ”escuta”o meio antes de enviar seus quadros, porém quando há mais de um nodo na rede querendo tomar posse do meio, o CSMA/CA introduz alguns mecanismos de controle que irão tentar prevenir o conflito entre os quadros, o que ocasionaria em perda de dados e tempo de transmissão efetiva. Os mecanismos impostos pelo protocolo são:

- Quadro de reconhecimento (ACK): Com base em [6], uma transmissão é considerada bem sucedida quando, a estação de origem recebe um quadro de controle ACK enviado pela estação receptora após esperar um curto intervalo de tempo denominado SIFS *(Short Inter Frame Space)*. Os diferentes tipos de intervalo de tempo serão explicados adiante. Todo quadro transmitido deve ser reconhecido, caso não haja confirmação da estação receptora, o pacote é dado como perdido e é feita uma retransmissão. A Figura 2.5, ilustra o funcionamento de reconhecimento de dados.

A Figura 2.6, ilustra o formato do quadro MAC de controle. O quadro é, basicamente, formado pelos campos: *Controle de Quadro* para indicar qual é o tipo de quadro MAC 802.11, *Duração* para especificar o tempo restante para receber a próxima transmissão, *RA* que informa o endereço do receptor do quadro ACK e *FCS* para controle de erro.

- Espera antes da transmissão (*backoff*): O *backoff* é um tempo aleatório obrigatório utilizado antes de efetuar uma transmissão ou uma retransmissão, quando um quadro ACK

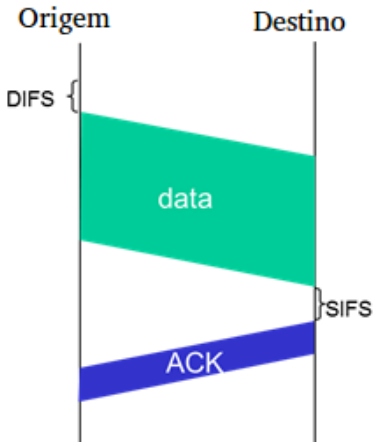

Figura 2.5: Quadro de reconhecimento ACK. [12].

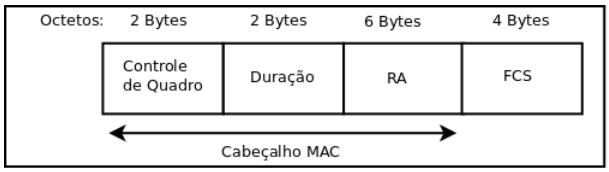

Figura 2.6: Formato do quadro ACK. (Fonte: Elaborada pelo autor).

não é recebido. Cada estação que deseja transmitir deve calcular seu próprio intervalo. O intervalo de *backoff* depende de duas variáveis, a Janela de Contenção *(Contention Window - CW)* e o Contador de Backoff. O valor do Contador de Backoff é um inteiro pseudoaleatório entre 0 e CW. O valor de CW varia de *CWmin* a *CWmax*, na norma IEEE 802.11a/g o *CWmin* equivale a 15 *slots* e na IEEE 802.11b o valor é de 31 *slots*, já para *CWmax* é 1023 *slots* para todos os padrões. O valor de CWmin é utilizado na primeira tentativa de acesso, porém esse tempo é recalculado, através da equação $CWmin = (2x(CW + 1) - 1)$, à medida que falhas sucessivas de transmissão ocorram, até chegar em *CWmax*. Entretanto, quando há sucesso em uma transmissão, o valor de CW é atualizado com o valor de *CWmin*. A Figura 2.7, resume o funcionamento do *backoff* com os valores de *Slot Time* citados na Tabela 2.1.

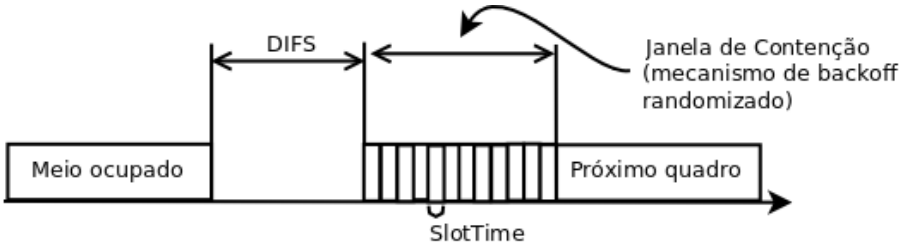

Figura 2.7: Janela de Conteção e Contador backoff. (Fonte: Elaborada pelo autor).

- Intervalos entre quadro (IFS): É o tempo mínimo que uma estação deve esperar antes de transmitir, após o meio ficar ocioso. Existem alguns tipos de valores que fornecem uma prioridade diferente para acesso ao meio. A Figura 2.8 ilustra os tipos de IFS.

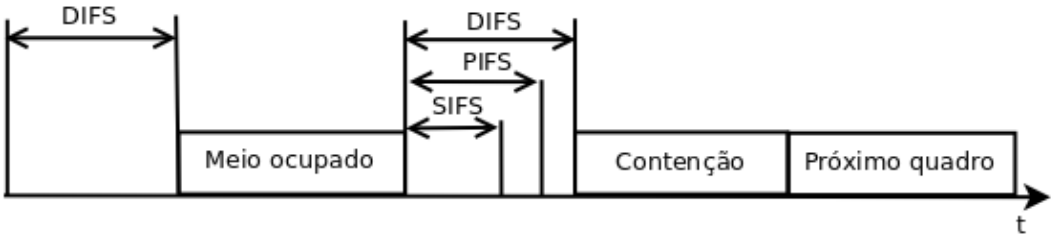

Figura 2.8: Tipos de IFS. (Fonte: Elaborada pelo autor).

- *Short Interframe Space* (SIFS): É um intervalo usado, especificamente, antes do envio de quadros de controle como o ACK. Os quadros de controle tem maior prioridade no acesso ao meio, pois o SIFS é mais curto que os outros intervalos. O seu valor depende da Camada Física e é definido na Tabela 2.1.

IFS	802.11b	802.11g	802.11a	802.11n 2.4GHz	802.11n 5GHz
SIFS	10µs	10µs	16µs	10µs	16µs
Slot Time	20µs	Long = 20µs Short = 9µs	9µs	Long = 20µs Short = 9µs	9µs
DIFS	50µs	Long = 50µs Short = 28µs	34µs	Long = 50µs Short = 28µs	34µs

Tabela 2.1: Valores de IFS. [13]

A Figura 2.9, exibe um fluxograma do MAC CSMA/CA em modo de contenção, especificando parâmetros como *backoff*, Janela de Contenção e a ação a ser tomada quando não é recebido um quadro ACK.

- *PCF Interframe Space* (PIFS): Intervalo utilizado antes de transmitir um quadro quando uma estação está em modo PCF *(Point Coordination Function)*, livre de conteção. O PCF implementa um tipo de acesso ao meio raramente utilizado nos equipamentos, e não é o foco neste trabalho. Pelo fato de ter um acesso livre de contenção, o PIFS possui um intervalo menor que o DIFS, por isso possui maior prioridade de acesso.

- *Distributed Interframe Space* (DIFS): É o intervalo mais frequente, aplicado no início da transmissão de quadros de dados em geral quando o acesso é por contenção via DCF. Esse intervalo pode ser calculado através da equação: $DIFS = SIFS + 2xSlotTime$. Alguns valores são mostrados na Tabela 2.2.

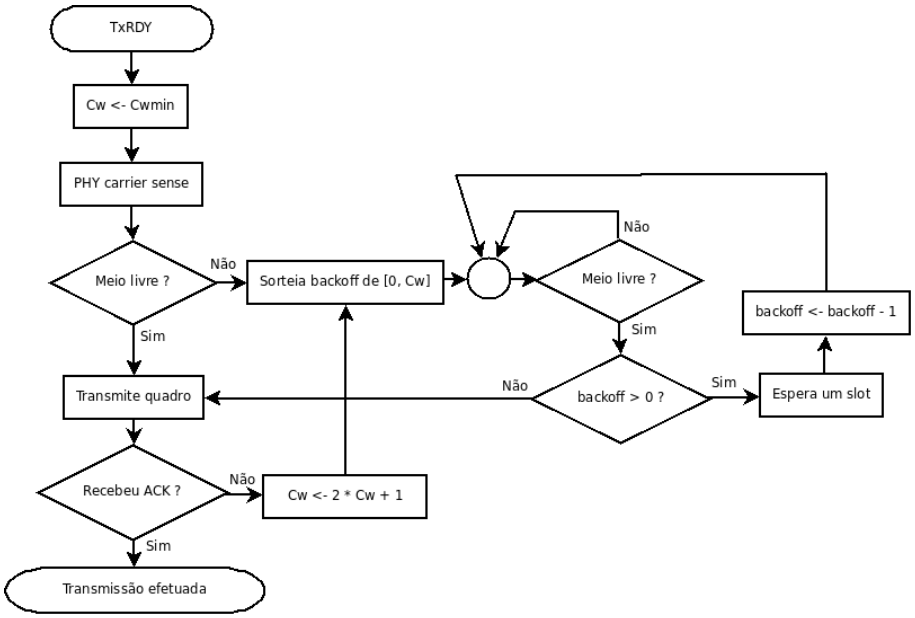

Figura 2.9: Fluxograma CSMA/CA.

2.1.3 EDCA

A função EDCA *(Enhanced Distributed Channel Access)* é definido pelo adendo IEEE 802.11e, especificado na Figura 2.10. De acordo com [14], o IEEE 802.11e introduz suporte QoS *(Quality of Service)* básico por definir quatro diferentes Categorias de Acesso (ACs), ou seja, VO (Voz) com alta prioridade, VI (Vídeo), BE (Melhor Esforço) e BK *(Background)* com a menor prioridade, conforme segue a Tabela 2.2.

O método EDCA é onde cada CA contém DCF com os seus próprios parâmetros de contenção (CWMin, CWmax, AIFS). Basicamente, quanto menor dos valores de CWMin, CWmax e AIFS [AC], mais curto é o atraso de acesso ao canal para o AC correspondente. Em EDCA um novo tipo de IFS é introduzido, o arbitrário IFS (AIFS), em vez de DIFS em DCF. Cada AIFS é um intervalo IFS com comprimento arbitrário da seguinte forma: $AIFS[AC] = SIFS + AIFSN[AC]xSlotTime$, onde AIFSN é chamado de número IFS arbitrário. Após a detecção do meio ter estado inativo durante um intervalo de tempo de AIFS [AC], cada AC calcula seu próprio tempo de *backoff* aleatório.

Os parâmetros de contenção podem ser ajustados, e dependendo da modificação pode-se ter um acesso ao meio livre de contenção, fazendo com que haja maior agilidade no processo de transmissão de cada estação da rede.

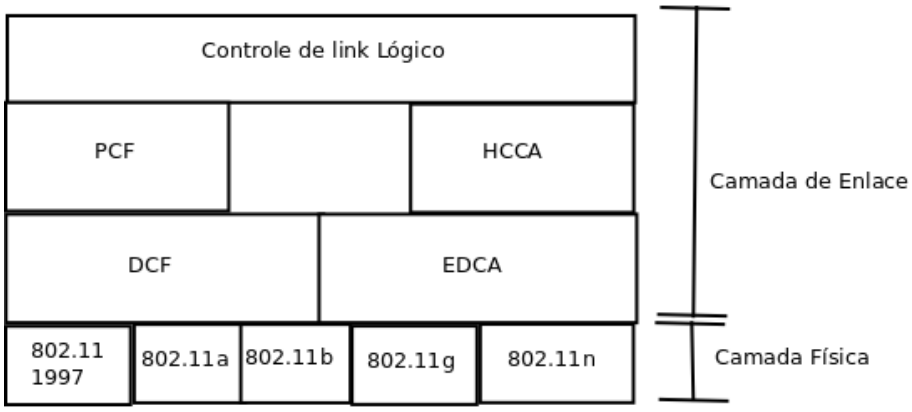

Figura 2.10: EDCA no Modelo OSI. (Fonte: Elaborada pelo autor).

AC	CWmin	CWmax	AIFSN
BK (Background)	aCWmin	aCWmax	7
BE (Best Effort)	aCWmin	aCWmax	3
VI (Video)	(aCWmin+1)/2-1	aCWmin	2
VO (Voice)	(aCWmin+1)/4-1	(aCWmin+1)/2-1	2

Tabela 2.2: Valores dos parâmetros das categorias de acesso. (Fonte: Elaborada pelo autor).

2.1.4 Limitações do método de acesso ao meio IEEE 802.11

Com todos esses mecanismos que o MAC CSMA/CA utiliza para tentar prevenir colisões entre quadros, é de fato que o modo de acesso incorporado, controlado pelo DCF, é mais apropriado em redes onde há múltiplas estações desejando transmitir simultaneamente. Esse tipo de prevenção faz com que os intervalos especificados anteriormente, limitem a vazão máxima do fluxo de dados em uma rede sem fio, até mesmo em casos quando apenas um nodo deseja transmitir. Sendo assim, neste trabalho, buscaram-se alternativas que possam ser implementadas em uma rede IEEE 802.11, para diminuir o tempo gasto pelo CSMA/CA, em um cenário onde há apenas duas estações pretendendo enviar seus dados, o típico enlace ponto a ponto. Os devidos ajustes nas características de acesso ao meio empregadas pelo EDCA, fazem com que, no cenário ponto a ponto, a transmissão não seja tão limitada quando o acesso é controlado pelo DCF, pelo fato de que há um tempo menor de espera na fila de transmissão de quadros das estações configuradas.

2.2 Tipos de Protocolos de Controle de Acesso ao Meio

Segundo [7], há três tipos de MAC existentes, o Acesso Aleatório que como alternativa utiliza o CSMA/CA para fazer uso do meio de transmissão, o Particionamento de Canal e o de Revezamento. O protocolo MAC, como descrito na seção anterior, tem a função de coordenar as transmissões de estações diferentes para minimizar ou até mesmo evitar colisões. Busca ser eficiente, justo, simples e descentralizado, porém essas metas serão bem sucedidas dependendo do tipo de MAC e do cenário que esse é empregado.

2.2.1 Particionamento de Canal

O Particionamento de Canal incorpora as técnicas de TDM *(Multiplexação por Divisão de Tempo)* ou FDM *(Multiplexação por Divisão de Frequência)* para particionar a largura de banda de um canal entre todas as estações que compartilham esse mesmo meio.

Baseado em [7], quando utilizando o TDM, um canal, com N nodos, irá se dividir em pelo menos N *time slots* (intervalos de tempo). Cada *time slot* é atribuído a um dos N nodos. O TDM elimina as colisões e é perfeitamente justo, pois toda estação é limitada a uma taxa mesmo quando todos querem enviar ao mesmo tempo. Torna-se ineficiente quando os usuários exigem pouca demanda ou quando a carga for baixa, ou seja, se somente um nodo transmitisse e os outros escutassem. Essa técnica seria interessante em um cenário com vários dispositivos conectados na mesma rede, onde todos teriam a mesma oportunidade de usufruir do meio. A Figura 2.11 ilustra a técnica TDM como um método para divisão do canal, onde todo *slot* de tempo 1 é dedicado a um nodo específico. Uma tecnologia que utiliza essa forma de acesso é o ZigBee do padrão IEEE 802.15.4[1]. O MAC do IEEE 802.15.4 possui suporte a alocação de intervalo de tempo garantidos (GTS) para um tempo de transmissão de dados.

Há a técnica com utilização do FDM, onde a taxa de transmissão do canal é divida em frequência, e atribui a cada frequência um dos N nodos, assim temos a subdivisão de um canal maior em canais menores, também com igual oportunidade a todos. A imagem 2.12, mostra a utilização da técnica FDM, onde há 6 estações, sendo 1, 3 e 6 têm pacotes, e as bandas de frequência 2, 4 e 5 estão ociosas.

[1] `http://ieeexplore.ieee.org/document/5685906/`

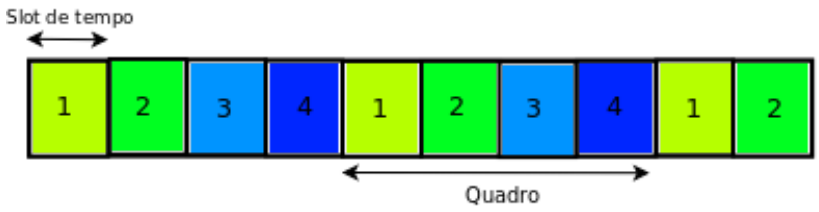

Figura 2.11: Particionamento de Canal - técnica TDM. (Fonte: Elaborada pelo autor).

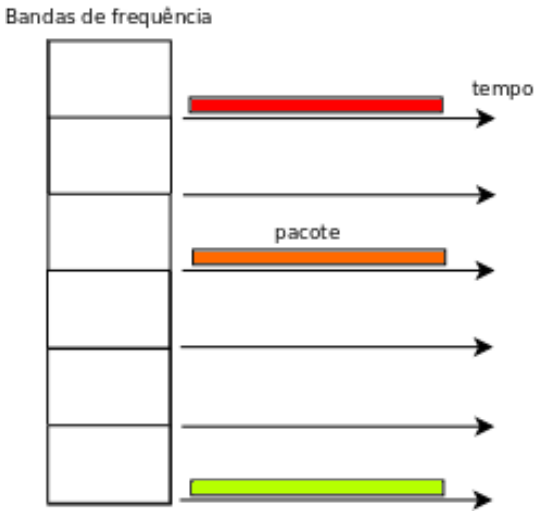

Figura 2.12: Particionamento de Canal - técnica FDM. (Fonte: Elaborada pelo autor).

2.2.2 Protocolo de Revezamento

O Protocolo de Revezamento é um tipo de protocolo de acesso ao meio que possui duas variações designadas *"Polling"* ou Mestre-Escravo *(Master-Slave)* e Passagem de Ficha *(Token-Passing)*. O princípio do MAC por revezamento é o acesso ao meio ser concedido a partir de uma mensagem de controle.

- Mestre-Escravo: Segundo [7], o *Polling* uma das estações é designada mestre *(master)* e as outras como escravo *(slave)*. A estação mestre pesquisa *(polls)* cada um dos escravos de uma forma cíclica, através de um algoritmo chamado *Round-Robin*. Esse algoritmo atribui frações de tempo para cada estação de forma circular, assim manipula a rede sem que um nodo tenha mais prioridade que o outro.

A estação mestre informa ao escravo, através de uma mensagem (*polling*), que ela tem um tempo máximo para transmitir, assim que a transmissão é concluída o mestre informa ao próximo nodo que também poderá transmitir dentro de um certo limite de tempo. O procedimento continua, com o mestre ”convidando”cada nodo de uma forma cíclica. Ele verifica que o escravo acabou de transmitir observando a falta de sinal no canal. A Figura 2.13 ilustra a comunicação do Mestre com as múltiplas estações Escravos, enviando a mensagem de permissão de transmissão. Um dos problemas de se trabalhar com esse tipo de protocolo é a perda de comunicação caso o mestre falhe, resultando em queda na rede, pois é ele quem permite que outros dispositivos sem fio transmitam. Outro caso seria se uma estação Escravo quisesse transmitir, o que só poderá fazer se receber a permissão do Mestre.

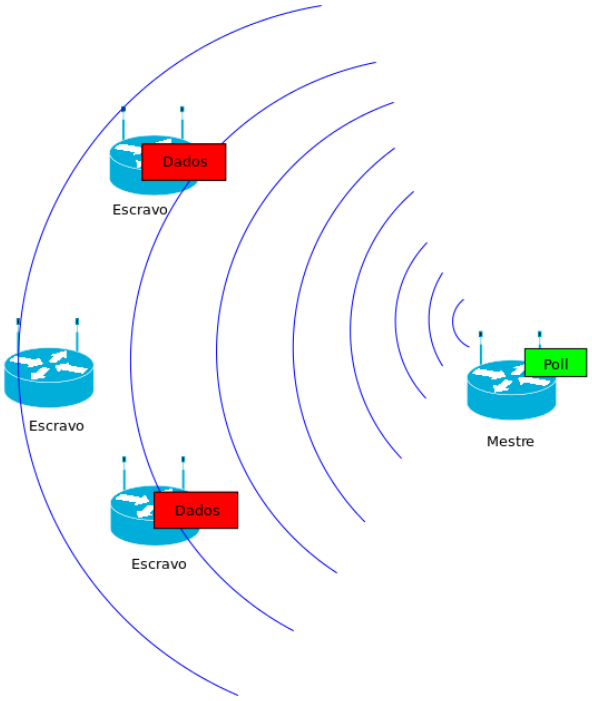

Figura 2.13: Revezamento - técnica Mestre-Escravo. (Fonte: Elaborada pelo autor).

A ideia desse tipo de protocolo de acesso ao meio, no cenário que iremos trabalhar, será a que mais agregará posições positivas, pelos seguintes motivos: possui uma maior eficiência, quando comparado com outros tipos de MAC, maior probabilidade de eliminar espaços vazios e colisões na rede, o que poderia acontecer quando o acesso ao meio é feito de forma aleatória através dos mecanismos de prevenção de colisão.

- Passagem de Ficha: Com base em [7], nessa variação não utiliza-se um nodo *master*, mas uma passagem de ficha de permissão o chamado *token*, o qual é passado sequencialmente de estação a estação, como segue a Figura 2.14. Quem estiver de posse do *token*, caso queira

transmitir, poderá enviar um número máximo de quadros e ao final encaminha para o próximo, caso não queira transmitir, ou seja, quando não houver quadros na fila de espera, passa o *token* adiante.

A Passagem de Ficha é descentralizada e de alta eficiência assim como o Mestre-Escravo. Entretanto, se um nodo só falhar, pode interromper o acesso ao canal, e se um nodo deixar de liberar o *token*, será gerado um processo de descoberta para que a circulação da ficha volte.

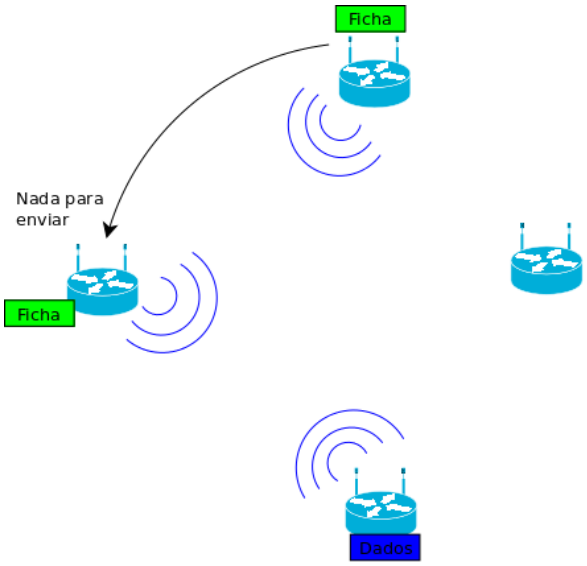

Figura 2.14: Revezamento - técnica Passagem de Ficha em uma rede sem fio. (Fonte: Elaborada pelo autor).

Fazendo uma comparação entre as técnicas Mestre-Escravo e a Passagem de Ficha, em um cenário ponto a ponto, a técnica Passagem de Ficha nada acrescentaria. Talvez em um ambiente com mais estações fosse mais atrativo. Segundo [7], o problema maior é a complexidade para a manutenção do *token* em casos de perda, pois o tempo gasto para a reestruturação da rede seria um tempo perdido.

Foi definido que o MAC por revezamento possui propriedades importantes a serem tomadas como base como por exemplo: a inexistência de colisões, além de ser determinístico, ou seja, possibilita conhecer quanto tempo um quadro levará para ser transmitido, e a capacidade do canal que pode ser usada por cada estação. Por não haver colisões por definição, não é necessário esperas de *backoff*, o que melhora o aproveitamento do meio.

2.3 Chipset Atheros e Linux

A interface USB Atheros, possui um modelo de chipset AR9271, que segundo [15], é uma solução de *chip* desenvolvida com alta integração a redes locais sem fio, capaz de configurar uma estação para obter melhor qualidade de conexão e máximo *throughput*. Este tipo de controlador Wi-Fi, apresenta integração com multi-protocolos MAC, característica que permite a calibração dos parâmetros de acesso ao meio como: a mensagem de controle ACK, janela de contenção, alteração nos valores dos intervalos. Possui suporte ao protocolo IEEE 802.11e, assim tornou-se o modelo mais atrativo para alcançar um dos objetivos específicos proposto. O MAC do AR9271 provê um controle de dados com processamento em fila, é uma fila do tipo FIFO *(First-in, First-out)*, a qual controlará a saída de transmissão de dados.

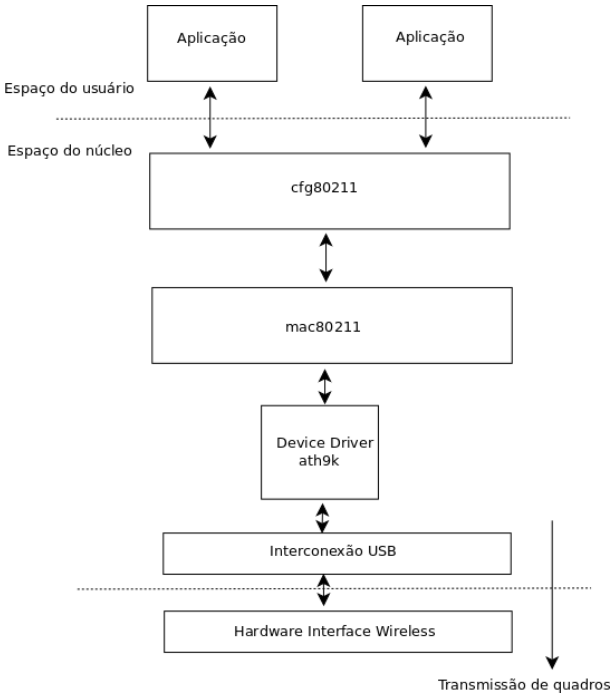

Figura 2.15: Caminho de transmissão no Linux. (Fonte: Elaborada pelo autor).

A Figura 2.15, mostra uma estrutura de compatibilidade do *wireless driver* ath9k, correspondente ao Chipset Atheros, com o *device driver* de modelo mac80211 do kernel do Linux, permite, além da instalação da interface sem fio USB, a configuração e os ajustes de parâmetros do MAC, que podem ser feitos através de utilitários do Linux. Outra característica interessante dessa compatibilidade, é o fato de poder estabelecer uma comunicação entre um programa, desenvolvido sobre o espaço do usuário, e o espaço do Kernel.

2.4 Netlink

A API Netlink, mais especificamente nl80211, é uma interface de programação para a comunicação entre processos de usuário e o módulo cfg80211, que controla os mecanismos do núcleo do Sistema Operacional Linux. Dessa forma, permite a comunicação com o *wireless driver* de modelo ath9k do Chipset Atheros, possibilitando ajustes no modo de operação da interface 802.11.

O protocolo Netlink utiliza a biblioteca *libnl*. Essa biblioteca contém as funções que possibilitam a comunicação para a construção, análise e o envio e recebimento das mensagens de controle e dados. Essa comunicação com o kernel se faz através da manipulação de *sockets*. A biblioteca possui algumas ramificações e uma delas é a API para protocolo genérico Netlink *(libnl-genl)*, esta é uma extensão da *libnl*, também em espaço do usuário, utilizada para o registro de comandos, criação de cabeçalhos para o envio das mensagens e a conexão e desconexão à *sockets* genéricos. Com o uso do *socket* e dessas bibliotecas, é possível que uma aplicação em espaço de usuário seja configurada para a comunicação com o interior do Linux. A Figura 2.16, segue a estrutura de comunicação entre um processo do usuário e o núcleo do Lunix.

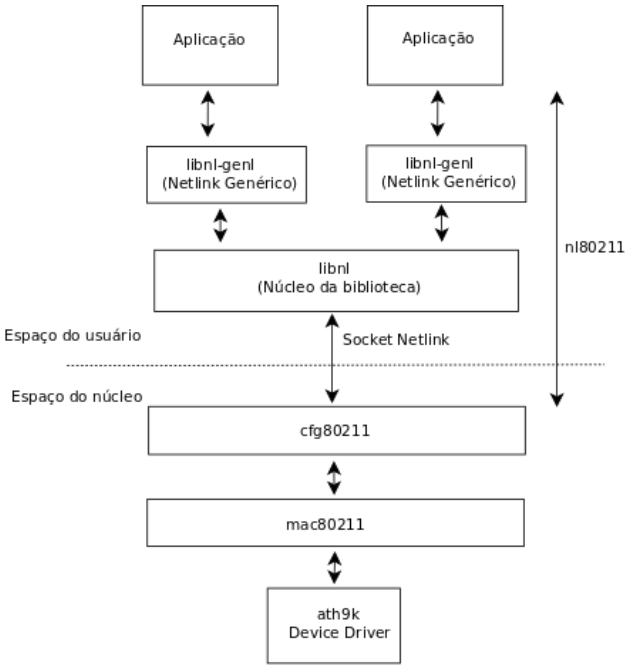

Figura 2.16: comunicação através do nl80211. (Fonte: Elaborada pelo autor).

Há outras interfaces do Linux que possibilitam a comunicação com o Kernel como: *ioctl* ou *procfs*, porém o Netlink é o sucessor desses métodos, além de ser mais flexível provendo

principalmente as configurações dos serviços de rede e monitoramento de interfaces.

Com base em todas as características do Chipset Atheros e a viabilidade de configuração através da ferramenta de comunicação Netlink, tornou-se possível a modificação do *CWmin, CWmax, AIFS*, influenciando diretamente no intervalo que cada quadro possui antes de fazer uma transmissão. E a ativação do comando NOACK, o qual irá desativar o quadro de controle ACK da interface sem fio. A ordenação desses parâmetros irão diferenciar o acesso ao meio do dispositivo sem fio, do protocolo atual, no caso o CSMA/CA, conforme relatado na Sub-seção 2.1.2.

2.5 Packet Socket

A comunicação entre processos do programa e a interface sem fio, é concebida pela API de *Socket*, mais precisamente *Packet Socket*. De acordo com [17], são utilizados para receber e transmitir pacotes do *driver* do dispositivo a nível de Camada 2, permitindo a implementação de um módulo em espaço do usuário no topo da Camada Física.

Dois tipo de *sockets* são disponibilizados. *Sock RAW* os pacotes são encaminhados e recebidos para o dispositivos sem nenhuma alteração no dado do pacote, incluindo cabeçalho a nível de enlace e *Sock DGRAM* é um datagrama *socket* que não inclui o cabeçalho, pois é removido antes de ser analisado pelo usuário, manobra que facilita a análise dos dados da comunicação, por este motivo o protocolo foi implementado com este tipo de *socket*.

3 *Protocolo MAC Ponto a Ponto sem fio*

Este Capítulo apresenta o desenvolvimento do protocolo MAC adaptado para um cenário ponto a ponto em uma conexão sem fio. É descrito o seu comportamento, a implementação, os testes e as discussões sobre os experimentos realizados.

3.1 Descrição do modelo

Para aumentar a eficiência do meio de transmissão de uma forma que também previna colisões, foi desenvolvido um protocolo para pontos de acesso, com características que se adéquam a um enlace ponto a ponto, baseado em redes IEEE 802.11. Para aumentar a vazão, a ideia foi implementar um novo MAC que consiga determinar quem possui o direito de transmissão, livre de disputa, diminuindo os atrasos entre cada quadro transmitido, sem a necessidade de confirmação de recebimento de quadro, construído a partir do CSMA/CA, porém com alguns ajustes que são especificados na versão do EDCA.

Os protocolos de acesso ao meio, como citado na seção 2.2, possuem particularidades que os tornam apropriados para diferentes situações, no que diz respeito a esse trabalho, o ideal foi trabalhar com um protocolo que fornecesse uma forma de acesso que evitasse perdas e com o melhor aproveitamento possível do canal, e consequentemente um aumento na vazão. Por este motivo, o protocolo de revezamento, mais especificamente o *Polling* ou Mestre-Escravo, foi tomado como base. A forma de controle de acesso que ele descreve foi a que mais atraiu, pela maneira como a estação Mestre manipula a estação Escravo, diminuindo a possibilidade do meio ficar ocioso, tornando-se a mais eficiente para um cenário PTP *(Point to Point)*.

3.1.1 Comportamento do protocolo MAC PTP

A implementação do código foi baseada no modelo de Máquina de Estado Finito (MEF). Uma MEF é composta por estados e transições. Os estados representam a maneira como o

sistema irá se comportar e uma transição seria uma mudança de estado. Para que a transição ocorra é necessário que um evento tenha acontecido.

Com base no conceito do protocolo Mestre-Escravo, o *driver* foi construído com dois modelos de MEF diferentes, um que representa o Mestre e o outro o Escravo. O programa do Mestre segue o modelo da Figura 3.1. Composto por um conjunto de 3 estados, o início da comunicação se dá pela transmissão da mensagem de controle *Poll* (estado Transmissão *Poll*), a qual permite que o Escravo envie os dados que estão armazenados na fila de transmissão. Logo que o evento envia *Poll* acontece, há uma transição do estado Transmissão *Poll* para o estado Recepção. O modo Recepção é responsável por receber os dados da estação origem e encaminhá-los a Camada de Rede. É também função do modo Recepção armazenar os dados, recebidos da Camada de Rede, em uma fila para serem transmitidos imediatamente após o recebimento da mensagem de *timeout* do Escravo, dessa forma faz a transição para o modo de transmissão de dados (estado Transmissão). No Transmissão, foi determinado um tempo de 200 milissegundos, esse é o período onde a estação poderá transmitir a quantidade máxima de dados, sem a necessidade de qualquer confirmação de recepção. Ao atingir o *timeout* o programa retorna para o estado Transmissão *Poll*.

O tempo de 200 milissegundos para transmissão, foi definido aleatoriamente, mas dentro de um intervalo aceitável, pelo fato de que não houve tempo suficiente para testes com outros intervalos.

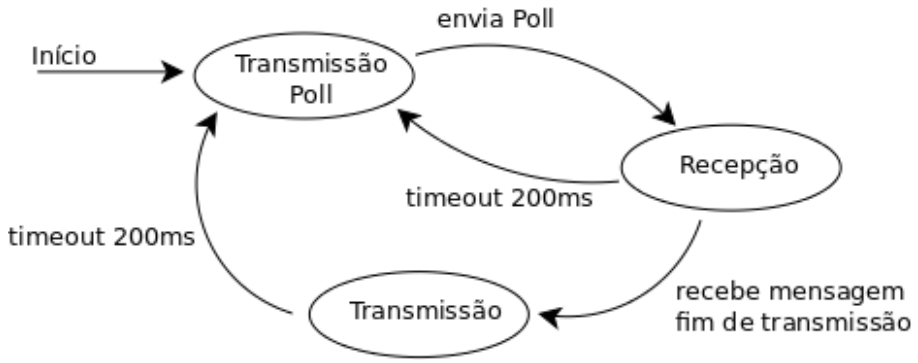

Figura 3.1: MEF Mestre. (Fonte: Elaborada pelo autor).

O comportamento dos estados para o tratamento das mensagens é praticamente igual tanto para o Mestre quanto para o Escravo. A diferença é que o estado de Recepção da estação Mestre também foi projetado para tratar o reenvio do *Poll*, se porventura esta estação não receba qualquer dado durante 200 milissegundos, como mostra a MEF do Mestre na Figura 3.1. Outro fato importante entre os *drivers* das estações é na maneira como o programa é iniciado. Como

o Mestre inicia no estado de Transmissão do *Poll*, obrigatoriamente, o Escravo deve iniciar no estado de Recepção para monitorar a chegada da mensagem de permissão de acesso ao meio e enfileirar os pacotes recebidos da Camada de Rede. A Figura 3.2, destaca a MEF do Escravo, começando pelo modo Recepção, transitando para o estado Transmissão, após receber o *Poll* e retornando para o estado Recepção ao completar o ciclo de 200 ms, enviando assim a mensagem fim de transmissão para a estação Mestre.

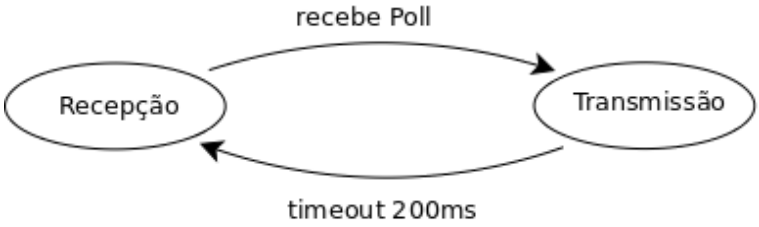

Figura 3.2: MEF Escravo. (Fonte: Elaborada pelo autor).

A Figura 3.3 apresenta um modelo da troca de mensagens entre as estações. A estação Mestre inicia a comunicação com o envio da mensagem *Poll* e o Escravo responde, imediatamente, com a transmissão dos dados durante o tempo pré-definido. Ao encerrar o temporizador é enviado a mensagem de *timeout*. Caso alguma das estações não disponha de dados na fila de transmissão, o *timeout* só é atingido após os mesmos 200 milissegundos. Da mesma forma é o tratamento, se por acaso, a mensagem *Poll* não chegue ao destino, entretanto como já descrito, quem faz a retransmissão é a estação Mestre. Neste trabalho, o modelo de máquina de estado foi desenvolvido para tratar o comportamento do acesso ao meio somente para uma estação Mestre e uma estação Escravo. Sendo assim para a comunicação do Mestre com múltiplos Escravos é necessário o desenho de um novo modelo de MEF.

Além do tratamento das mensagens, o *driver* da Camada de Enlace também é responsável pelo envio dos comandos necessários para a configuração do *device driver* da interface sem fio. Isso se faz com a ferramenta Netlink. O protocolo ponto a ponto não especifica a confirmação de quadros recebidos, sendo assim o programa ativa a função No ACK nas estações. Com isso não há o atraso de propagação para o quadro ACK, no tipo de enlace sem fio que foi considerado. A fim de reduzir o tempo de espera na fila de transmissão, os comandos para ajustes do CWmin e CWmax são considerados. Como o tempo de *backoff* é definido a partir dos valores de CWmin e CWmax, o *driver* especifica valores mínimos para estes parâmetros. Por fim para que não exista um tempo de espera antes de transmitir, após o meio ficar ocioso, é especificado o valor de *slot time* do parâmetro AIFS. A configuração dos parâmetros do *driver* da interface será

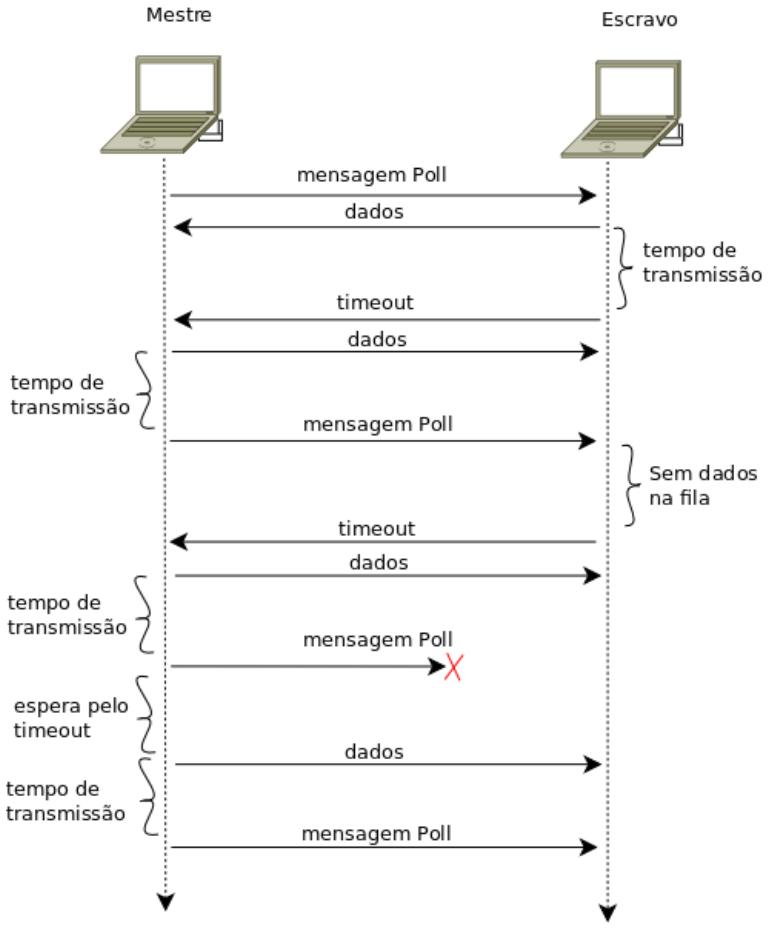

Figura 3.3: Comunicação Mestre-Escravo. (Fonte: Elaborada pelo autor).

detalhada na Sub seção 3.2.1.

Nesse contexto, tem-se uma noção de como é o funcionamento do controle de acesso ao meio dos dispositivos sem fio, implementados com o protocolo ponto a ponto para redes sem fio IEEE 802.11.

3.2 Protótipo

Com todas as possíveis funcionalidades, do CSMA/CA e do EDCA, definidas para configurar a ação da interface e com um modelo de protocolo construído, foi montado um protótipo de *driver* para ser instalado nas interfaces sem fio afim de atestar e comprovar o funcionamento, alcançando os objetivos propostos.

3.2.1 Construção do MAC ponto a ponto

Para a configuração dos mecanismos responsáveis pela prevenção de colisão, o código em espaço do usuário, utilizando as bibliotecas disponibilizadas pela API Netlink e, através da manipulação de chamada de sistema de *socket*, é feita a comunicação com o *device driver* da interface WLAN. Com o caminho estabelecido é feito o envio de comandos suportados pelo nl80211, para fazer os ajustes necessários:

- *No ACK policy:* Ativando a política de não reconhecimento no dispositivo de origem, significa que a interface destino não enviará quadros de controle ACK como confirmação de recebimento.

Parâmetros de fila de transmissão também foram ajustados, dos disponibilizados para a configuração pela interface são:

- *CWmin:* Foi possível configurar o valor mínimo deste atributo, no caso 1 *slot* para a Janela de Contenção Mínima.

- *CWmax:* O nl80211 dispõe de uma variação do valor da Janela de Contenção que vai de 1 a 32767 *slots*, para o caso da Janela de Contenção Máxima foi atribuído o valor de 1 *slot time*.

- *AIFS:* Com o nl80211, é possível setar um valor diretamente para o Espaço Arbitrário entre quadros, diante disso foi definido o valor de 0 *slot*.

A mensagem *Poll*, foi criada a partir de um quadro Ethernet, no código do programa. Conforme descrito na Sub-seção 2.1.1, o quadro possui um campo chamado *ethernet type*, o qual especifica o tipo do protocolo de quadro que está sendo transmitido. Para a criação do *Poll*, esse campo foi alterado, substituindo o tipo de quadro com uma sequência em hexadecimal, no caso 0x080a. Assim, recebendo este quadro com o campo alterado, a interface destino saberá que a mensagem não é referenciada por nenhum protocolo, classificando-a como um quadro de controle. O modelo do quadro segue a Figura 3.4.

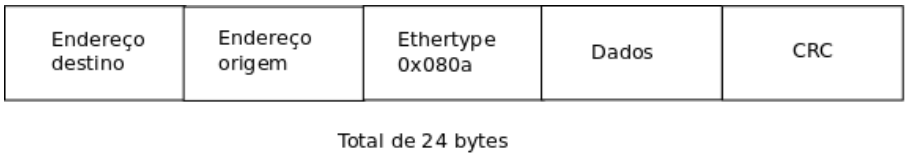

Figura 3.4: Modelo da mensagem Poll. (Fonte: Elaborada pelo autor).

3.2.2 Interface TUN/TAP

O protocolo desenvolvido, como mencionado anteriormente, é a implementação de um *driver* na Camada de Enlace, com ferramentas para viabilizar a comunicação com a interface sem fio. Nesse contexto, a interface TUN entra representando, usualmente, a Camada de Rede no modelo OSI de redes de computadores. É configurada junto ao *driver* para o envio e recebimentos de pacotes, utilizando o programa em espaço do usuário ao invés do meio físico. Com a interface TUN, pode-se estabelecer uma conexão ponto a ponto, já a interface TAP representa um enlace multiponto. Para o projeto, a TUN é a ideal, pois é, justamente, nesse tipo de cenário que o *device driver* trabalha.

A comunicação através da TUN se faz por meio de um descritor de arquivo retornado pela função de criação da interface, a *tun alloc()*. O descritor é um número inteiro que identifica a interface TUN no *Kernel* do Linux. A recepção de algo vindo dessa interface se faz com a chamada de sistema *read()* e o envio se faz com *write()*, de forma similar a leitura e escrita em arquivos. Isso significa que, algo vindo da TUN é receber pacotes da pilha de protocolos, no caso a Camada de Rede, lembrando que o programa reproduz a Camada de Enlace. E enviar para a interface TUN é entregar algo a Camada superior, novamente a Camada de Rede.

A interface TUN abre a facilidade de configurar os parâmetros de uma rede IP (endereço, máscara) para a comunicabilidade com o subsistema de rede, e rotas podem ser definidas para destinos alcançáveis através dela. Dessa forma qualquer aplicação TCP/IP pode se comunicar com o protocolo desenvolvido.

3.2.3 Diagrama do Protótipo

O modelo da integração dos componentes do Kernel do Linux com o protótipo desenvolvido segue o esboço da Figura 3.5. O MAC PTP, realiza a ponte entre o sistema interno de rede e a interface WLAN. A configuração da interface virtual TUN, para se comunicar com o subsistema, é feita com a definição de endereços IP do enlace ponto a ponto, máscara de rede e ativação da interface. Os endereços IPs envolvidos dizem respeito a como o enlace se apresenta ao Sistema Operacional. Esses endereços não aparecem para o MAC, que se preocupa somente com a Camada de Enlace. Por sua vez, o MAC se comunica com a TUN através do descritor de arquivo *fd net*. Com o estabelecimento desse acesso é possível que uma aplicação, também em espaço do usuário, receba e transmita dados para o MAC PTP.

Por outro lado, utilizou-se a API *packet socket* para criar duas vias de comunicação entre o

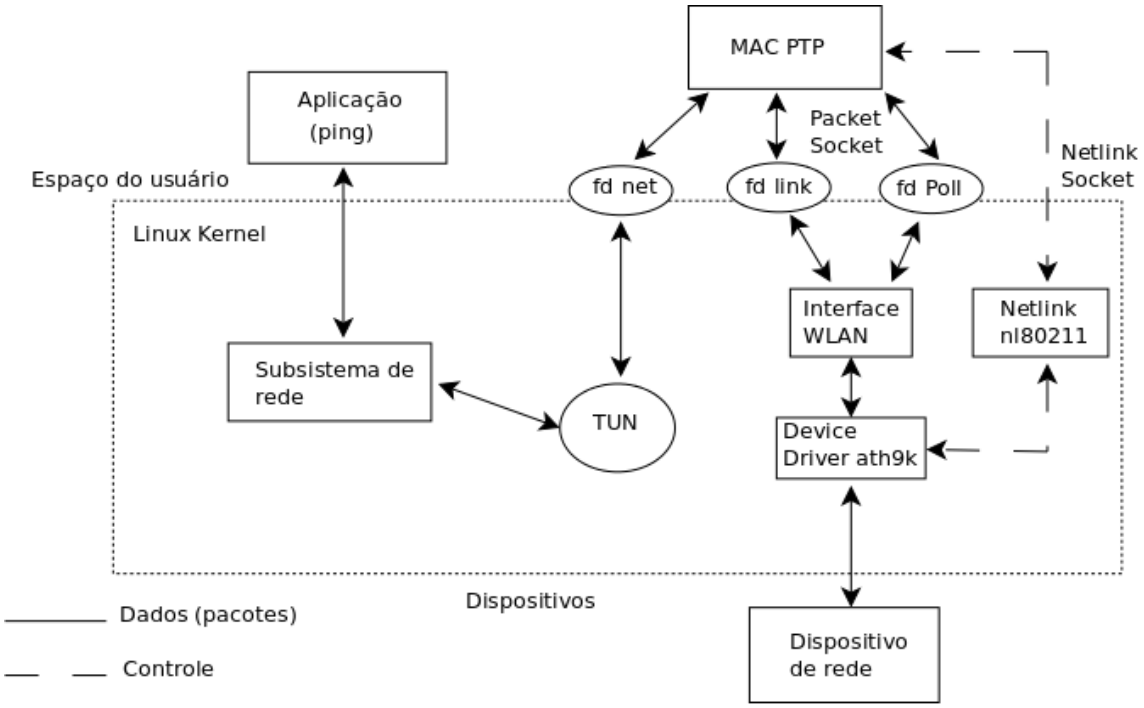

Figura 3.5: Integração do protótipo com o Linux. (Fonte: Elaborada pelo autor).

MAC ponto a ponto e a interface WLAN. Sendo assim, os pacotes recebidos da aplicação são enviados para a interface sem fio sobre o descritor *fd link* associado a um *socket*. Associado a outro *socket*, o descritor *fd Poll* tem a função de transmitir e receber o quadro de controle Poll, criado pelo próprio *driver* ponto a ponto. Os pacotes que são enviados a partir do *Sock DGRAM*, são especificados com informações de endereçamento, através da estrutura *sockaddrll*, tornando possível determinar o endereço MAC da interface destino para a conexão do *link*.

Foi definido um processo para otimizar o tempo de processamento de leitura dos dados vindo tanto da TUN quanto da interface WLAN. O sistema de leitura *read* utilizado é bloqueante por padrão, ao tentar ler um dado em um dos descritores o processo é bloqueado, assim o outro descritor é impossibilitado de ler. Para evitar o bloqueio de leitura de um conjunto de processos, foi utilizado um método de multiplexação de acesso a descritores. O protocolo implementa a técnica da chamada de sistema *select*, ela possui parâmetros que possibilitam o monitoramento dos descritores. Através da chamada é possível determinar uma medida para quando há dados vindo da interface sem fio, da Camada superior ou quando o *timeout* é atingido. Os dados recebidos da interface sem fio serão enviados diretamente a Camada de Rede, os dados vindo da Camada de Rede serão armazenados na fila para transmissão e o temporizador determinará o tempo de transmissão dos dados ou retransmissão do *Poll*.

Ao executar o MAC PTP, o *device driver* da interface WLAN é configurado com os parâmetros de rede especificados. A comunicação é feita por meio de um terceiro *socket*, o *netlink socket*, encarregado de encaminhar as mensagens para realizar os ajustes.

4 Testes e resultados

Os testes realizados nesse projeto, tiveram como objetivo coletar informações para uma análise de desempenho do protocolo MAC PTP. A fim de comparação foi feito medições do enlace com as interfaces na configuração padrão, de forma a verificar o desempenho do protocolo MAC atual e também os espaços entre quadros. O experimento visa atestar o funcionamento do protocolo desenvolvido e comprovar o aumento na vazão máxima de um enlace ponto a ponto com o padrão IEEE 802.11, tanto quanto a diminuição dos intervalos entre quadros.

4.1 Teste com o protocolo CSMA/CA padrão

O teste executado sem o uso do protocolo desenvolvido, visa medir a vazão, que um enlace ponto ponto pode obter com o mecanismo de acesso ao meio aleatório, além do calculo de uma média dos espaço entre quadros recebidos e o apontamento dos intervalos mais frequentes. Para esse experimento, foi utilizado um cenário onde as duas interfaces USB estavam configuradas em uma rede modo *Ad-Hoc*, cada uma conectada em uma máquina com Linux, assim não há necessidade de um *Access Point* centralizado para gerir a comunicação. Foi estabelecida três comunicações nos dois sentidos, com duração de 120 segundos e distância de aproximadamente 5 metros entre as máquinas.

A ferramenta de teste Iperf, foi utilizada para medir a performance do enlace. Os testes abaixo foram realizados sobre uma conexão UDP *(User Datagram Protocol)* entre dois nodos conectados na porta 8092 e usando o tamanho *default* de um datagrama que é 1470 bytes.

O Iperf possui uma funcionalidade de Cliente e Servidor, e possibilita mensurar a vazão em ambos os sentidos. Dessa forma, um dos dispositivos sem fio foi configurado como Servidor e o outro como Cliente. O Cliente envia os dados sem controle de fluxo, uma vez que o tráfego Iperf é sobre UDP. A taxa de transmissão foi medida através da quantidade de dados enviados durante o tempo pré determinado. Conclui-se então, que para um tempo de aproximadamente 2 minutos de transmissão efetiva, foram transferidos 15 MBytes o que resultou em uma taxa

```
|------------------------------------------------------------
Server listening on UDP port 8092
Receiving 1470 byte datagrams
UDP buffer size:  160 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.2 port 8092 connected with 10.0.0.1 port 34140
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0-119.0 sec  14.9 MBytes  1.05 Mbits/sec   0.363 ms   70/10701 (0.65%)
[  4] local 10.0.0.2 port 8092 connected with 10.0.0.1 port 55333
[  4]  0.0-120.0 sec  15.0 MBytes  1.05 Mbits/sec   0.866 ms    7/10701 (0.065%)
[  3] local 10.0.0.2 port 8092 connected with 10.0.0.1 port 42552
[  3]  0.0-120.0 sec  15.0 MBytes  1.05 Mbits/sec   0.376 ms    5/10701 (0.047%)
```

Figura 4.1: Iperf do Servidor com protocolo padrão. (Fonte: Elaborada pelo autor).

média de 1.05 Mbits/s.

Ao mesmo tempo que a conexão foi estabelecida para o envio dos dados com o Iperf, utilizou-se a ferramenta do Linux *tcpdump*, para capturar todos os quadros recebidos, e então viabilizar o cálculo da média dos intervalos entre quadros do enlaces. Com um parâmetro específico do *tcpdump*, foi possível mostrar o tempo decorrido desde o último pacote capturado, em microssegundos. Os pacotes foram capturados diretamente da interface WLAN destino. Aproveitando os dados do Iperf, foram recebidos em média 10647 pacotes, no tempo de 120 segundos, com um intervalo médio de 11.2 milissegundos.

```
00:00:00.011250 IP 10.0.0.1.34140 > 10.0.0.2.8092: UDP, length 1470
00:00:00.011204 IP 10.0.0.1.34140 > 10.0.0.2.8092: UDP, length 1470
00:00:00.011215 IP 10.0.0.1.34140 > 10.0.0.2.8092: UDP, length 1470
00:00:00.011225 IP 10.0.0.1.34140 > 10.0.0.2.8092: UDP, length 1470
00:00:00.011216 IP 10.0.0.1.34140 > 10.0.0.2.8092: UDP, length 1470
00:00:00.011236 IP 10.0.0.1.34140 > 10.0.0.2.8092: UDP, length 1470
00:00:00.011216 IP 10.0.0.1.34140 > 10.0.0.2.8092: UDP, length 1470
00:00:00.011204 IP 10.0.0.1.34140 > 10.0.0.2.8092: UDP, length 1470
00:00:00.011204 IP 10.0.0.1.34140 > 10.0.0.2.8092: UDP, length 1470
00:00:00.011211 IP 10.0.0.1.34140 > 10.0.0.2.8092: UDP, length 1470
00:00:00.011228 IP 10.0.0.1.34140 > 10.0.0.2.8092: UDP, length 1470
00:00:00.011205 IP 10.0.0.1.34140 > 10.0.0.2.8092: UDP, length 1470
00:00:00.011238 IP 10.0.0.1.34140 > 10.0.0.2.8092: UDP, length 1470
00:00:00.011215 IP 10.0.0.1.34140 > 10.0.0.2.8092: UDP, length 1470
00:00:00.012342 IP 10.0.0.1.34140 > 10.0.0.2.8092: UDP, length 1470
00:00:00.015322 IP 10.0.0.1.34140 > 10.0.0.2.8092: UDP, length 1470
```

Figura 4.2: Captura com Tcpdump. (Fonte: Elaborada pelo autor).

Na imagem 4.2, é mostrado um trecho da recepção dos pacotes enviados em direção ao Servidor. O campo destacado, refere-se ao tempo em micro segundos de um pacote recebido. Somando todos esses intervalos e dividindo pela quantidade de pacotes das três comunicações efetuadas, foi possível determinar um intervalo médio, do qual o valor é relatado no parágrafo anterior.

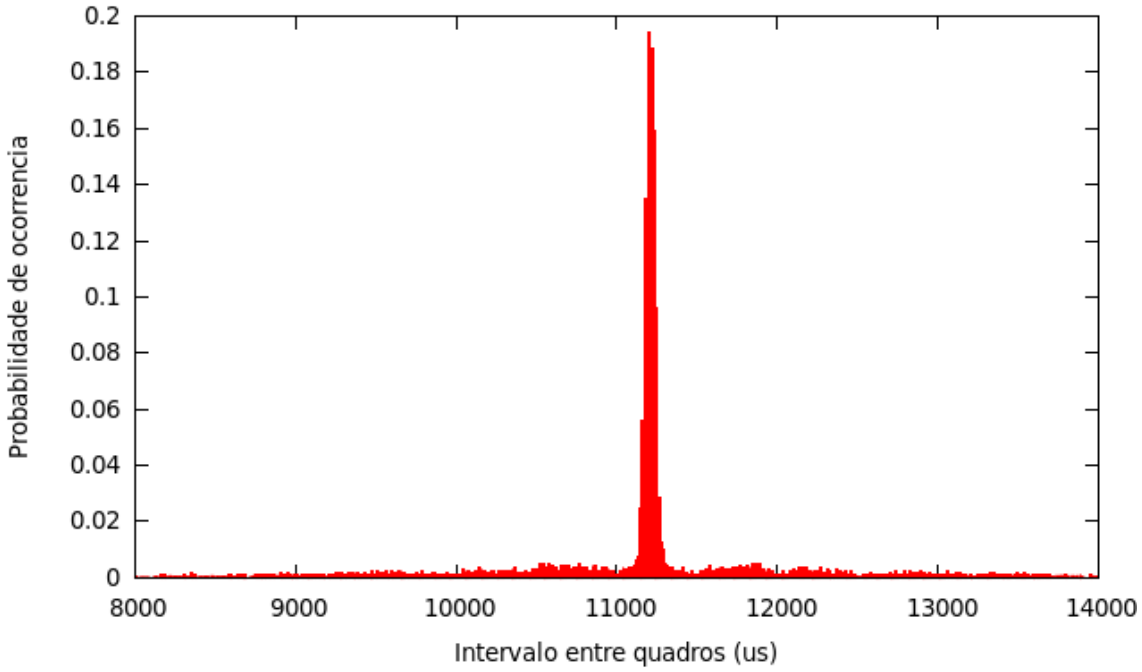

Figura 4.3: Espalhamento dos intervalos do protocolo CSMA/CA. (Fonte: Elaborada pelo autor).

A Figura 4.3, caracteriza a probabilidade de ocorrência de dispersão dos intervalos dos quadros, coletados através do *tcpdump*, da comunicação *Iperf*. Há uma centralização dos intervalos em torno de 11 ms, com quantidades espalhadas regularmente de 8 a 14ms. Em um acesso com possibilidade de disputa e aleatório, os mecanismos de prevenção de colisão e de confirmação de quadro são impostos, evidenciando maiores intervalos entre os quadros. Com a utilização dos mecanismos de controle, o acesso ao meio torna-se imprevisível e aleatório, possibilitando a transmissão de um quadro somente após o término de cada tempo de controle, resultando em uma probabilidade de espalhamento dos intervalos consideravelmente grande.

4.2 Teste com o protocolo MAC desenvolvido

Para a realização dos testes, foi explorado um ambiente onde houvesse o mínimo de interferência possível, para assim obter resultados mais precisos. O cenário do experimento foi exatamente o mesmo que o praticado sem a aplicação do protocolo MAC ponto a ponto. As interfaces sem fio foram configuradas em modo *Ad Hoc*, cada uma em uma máquina com Linux, em uma distância cuja a potência recebida do sinal atingiu uma média de -45dBm.

Primeiro buscou-se a ativação do protocolo Escravo em uma das máquinas, seguidamente

do protocolo Mestre. Dentro desse contexto, uma interface não se comunicará diretamente com a outra através do Iperf, todos os dados no sentido do Cliente para o Servidor serão direcionados para o protocolo ponto a ponto. A arquitetura da Figura 4.5, esboça o caminho que os dados irão percorrer quando é iniciada a execução do programa.

```
tun0      Link encap:Não Especificado  Endereço de HW 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet end.: 10.1.1.2  P-a-P:10.1.1.1  Masc:255.255.255.0
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1518  Métrica:1
          pacotes RX:51 erros:0 descartados:0 excesso:0 quadro:0
          Pacotes TX:0 erros:0 descartados:0 excesso:0 portadora:0
          colisões:0 txqueuelen:500
          RX bytes:6694 (6.6 KB) TX bytes:0 (0.0 B)
```

Figura 4.4: Interface TUN ativada. (Fonte: Elaborada pelo autor).

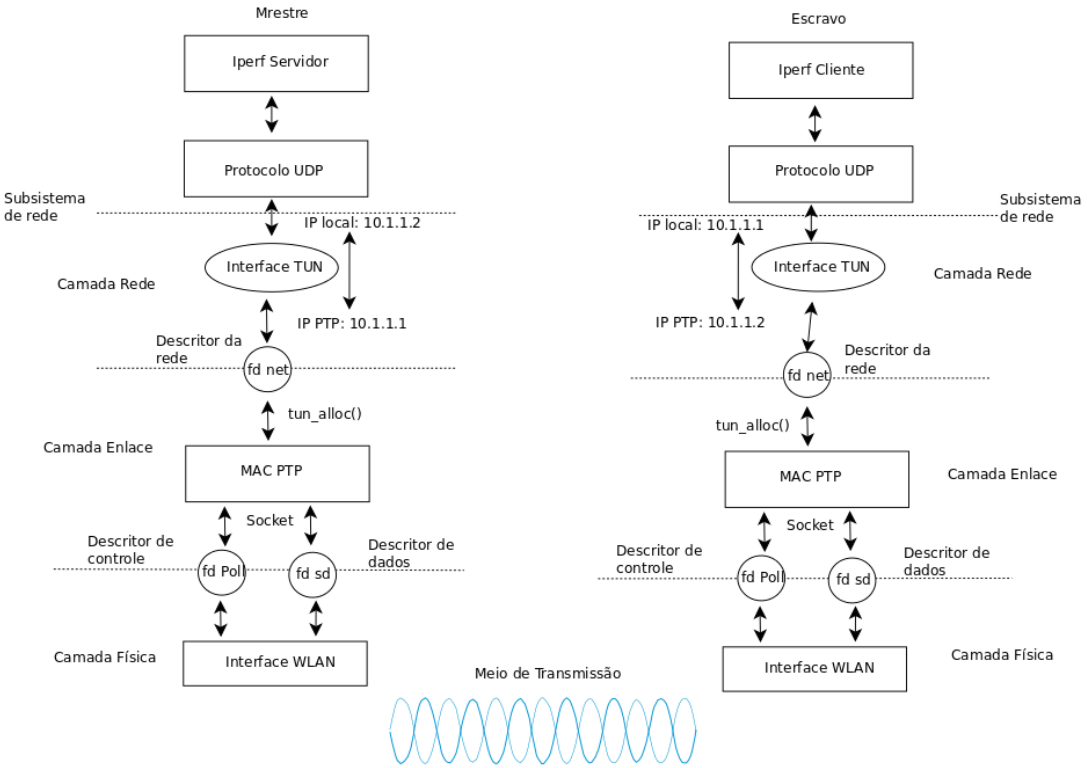

Figura 4.5: Arquitetura da comunicação Mestre-Escravo. (Fonte: Elaborada pelo autor).

Assim que o programa é ativado, a interface TUN é configurada e acionada, como mostra a Figura 4.4. Com ela inicia-se uma fila, com capacidade de armazenamento de 500 pacotes. Estes são armazenados na fila, caso o nodo esteja em modo de recepção. A TUN também é pré-configurada com endereçamento IP local e IP do ponto a ponto. O IP PTP, é usado como atributo do Cliente Iperf para a conexão com a interface destino.

Para mensurar a taxa de dados do enlace foi estabelecida uma conexão Cliente-Servidor com o Iperf. Foram feita 3 rodadas de transmissão nos dois sentidos. O experimento do Mestre transmitindo para o Escravo, conectados na porta 8091, obteve uma transferência de 30 MBytes de dados em 120 segundos de transmissão, resultando em uma taxa de 2.10 Mbits/sec.

```
|------------------------------------------------------------
Server listening on UDP port 8091
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 10.1.1.1 port 8091 connected with 10.1.1.2 port 40919
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0-120.0 sec  30.0 MBytes  2.10 Mbits/sec   1.481 ms    1/10701 (0.0093%)
[  3]  0.0-120.0 sec  10698 datagrams received out-of-order
```

Figura 4.6: Taxa de transmissão do Mestre. (Fonte: Elaborada pelo autor).

Do outro lado, a transmissão do Escravo em direção ao Mestre, alcançou um aproveitamento de 2.11 Mbits/sec, com aproximadamente 2 minutos de transmissão e 30 MBytes de transferência.

```
|------------------------------------------------------------
Server listening on UDP port 2424
Receiving 1470 byte datagrams
UDP buffer size:  160 KByte (default)
------------------------------------------------------------
[  3] local 10.1.1.2 port 2424 connected with 10.1.1.1 port 42398
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0-119.4 sec  30.0 MBytes  2.11 Mbits/sec  13.307 ms    1/10701 (0.0093%)
[  3]  0.0-119.4 sec  10698 datagrams received out-of-order
[  4] local 10.1.1.2 port 2424 connected with 10.1.1.1 port 34534
[  4]  0.0-119.3 sec  30.0 MBytes  2.11 Mbits/sec   1.501 ms    1/10701 (0.0093%)
[  4]  0.0-119.3 sec  10697 datagrams received out-of-order
```

Figura 4.7: Taxa de transmissão do Escravo. (Fonte: Elaborada pelo autor).

A perda de pacotes é informada nos resultados dos testes do Iperf. Como não há quadro de controle ACK, quadros perdidos não são retransmitidos. Conforme é representado nas Figuras 4.6 e 4.7, para um total de 10701 pacotes enviados foi perdido apenas 1, neste teste, praticamente, não houve perdas com o uso do protocolo PTP.

Novamente com o uso do *tcpdump*, foi efetuada uma captura dos pacotes recebidos, com intuito de medir o período de atraso de recepção de cada quadro de dado. Com o Mestre transmitindo, foi realizada a coleta dos pacotes na interface Escravo.

A Figura 4.8 indica os intervalos, inclusive da mensagem Poll. Como a transmissão é apenas em um sentido, mesmo que o Escravo não tenha nada para transmitir, o Mestre aguarda um

```
.011247 IP 10.1.1.2.53478 > 10.1.1.1.8091: UDP, length 1470
.011263 IP 10.1.1.2.53478 > 10.1.1.1.8091: UDP, length 1470
.011616 IP 10.1.1.2.53478 > 10.1.1.1.8091: UDP, length 1470
.004119 e8:94:f6:09:aa:12 (oui Unknown) > 10:fe:ed:27:05:82 (oui Unknown), ethertype Unknown (0x080a), length 38:
)x0000:  10fe ed27 0582 0000 c0aa 72b7 080a 0000  ...'......r.....
)x0010:  0000 0000 0000 0000                      ........
.200413 10:fe:ed:27:05:82 (oui Unknown) > e8:94:f6:09:aa:12 (oui Unknown), ethertype Unknown (0x080a), length 38:
)x0000:  e894 f609 aa12 0000 0000 0000 080a 0000  ................
)x0010:  0000 0000 0000 0000                      ........
.001212 IP 10.1.1.2.53478 > 10.1.1.1.8091: UDP, length 1470
.000500 IP 10.1.1.2.53478 > 10.1.1.1.8091: UDP, length 1470
.000500 IP 10.1.1.2.53478 > 10.1.1.1.8091: UDP, length 1470
.000501 IP 10.1.1.2.53478 > 10.1.1.1.8091: UDP, length 1470
.000500 IP 10.1.1.2.53478 > 10.1.1.1.8091: UDP, length 1470
```

Figura 4.8: Pacotes capturados no Escravo. (Fonte: Elaborada pelo autor).

tempo *default* de 200 milissegundos até desenfileirar e enviar, seus dados recebidos da Camada de Rede, assim é garantido que se chegar algum dado na fila da interface TUN do Escravo, durante esse tempo, ele será imediatamente transmitido. Por conta disso, são recebidos 11291 quadros, no tempo de 120 segundos, o que resulta em intervalo médio de 10,6 milissegundos.

No outro sentido da conexão, quadros enviados pelo Escravo são recebidos pelo Mestre e capturados pelo *tcpdump*. Conforme segue o resultado foi alcançado, em um tempo de transmissão de 2 minutos, a quantidade de 11289 quadros e um intervalo médio de 10.5 milissegundos.

```
00:00:00.011192 IP 10.1.1.1.42398 > 10.1.1.2.2424: UDP, length 1470
00:00:00.011245 IP 10.1.1.1.42398 > 10.1.1.2.2424: UDP, length 1470
00:00:00.011353 IP 10.1.1.1.42398 > 10.1.1.2.2424: UDP, length 1470
00:00:00.011371 IP 10.1.1.1.42398 > 10.1.1.2.2424: UDP, length 1470
00:00:00.000390 10:fe:ed:27:05:82 (oui Unknown) > e8:94:f6:09:aa:12 (oui Unknown), ethertype Unknown (0x080a), length 38:
        0x0000:  e894 f609 aa12 0000 0000 0000 080a 0000  ................
        0x0010:  0000 0000 0000 0000                      ........
00:00:00.200448 e8:94:f6:09:aa:12 (oui Unknown) > 10:fe:ed:27:05:82 (oui Unknown), ethertype Unknown (0x080a), length 38:
        0x0000:  10fe ed27 0582 0000 c0ba 6db7 080a 0000  ...'......m.....
        0x0010:  0000 0000 0000 0000                      ........
00:00:00.001194 IP 10.1.1.1.42398 > 10.1.1.2.2424: UDP, length 1470
00:00:00.000553 IP 10.1.1.1.42398 > 10.1.1.2.2424: UDP, length 1470
00:00:00.000460 IP 10.1.1.1.42398 > 10.1.1.2.2424: UDP, length 1470
00:00:00.000604 IP 10.1.1.1.42398 > 10.1.1.2.2424: UDP, length 1470
00:00:00.000528 IP 10.1.1.1.42398 > 10.1.1.2.2424: UDP, length 1470
00:00:00.000515 IP 10.1.1.1.42398 > 10.1.1.2.2424: UDP, length 1470
00:00:00.000540 IP 10.1.1.1.42398 > 10.1.1.2.2424: UDP, length 1470
```

Figura 4.9: Pacotes capturados no Mestre. (Fonte: Elaborada pelo autor).

Representada graficamente, a Figura 4.10 ilustra a distribuição dos intervalos de cada quadro recebido. A grande maioria dos intervalos ficou em torno de 500 us, com alguns entre 100 a 1500 us. A concentração dos intervalos em torno de 500 us, remete-se a inexistência de confirmação de cada quadro, a diminuição do atraso de *backoff* e do tempo antes de efetuar a transmissão (AIFS). Como o acesso é constituído pela transmissão de quadros sucessivos, em ”rajada”, dentro de um período de tempo, não há uma grande dispersão sobre os intervalos.

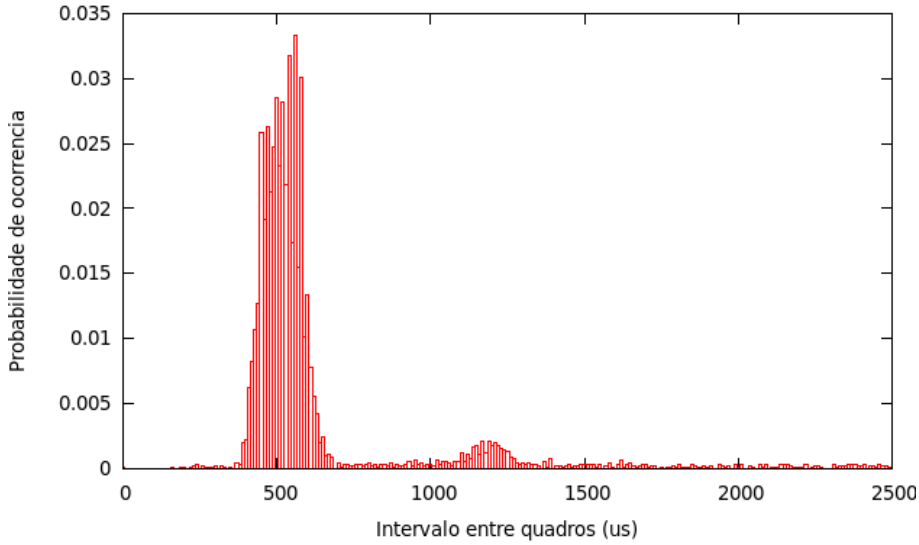

Figura 4.10: Gráfico dos intervalos entre quadros no protocolo PTP. (Fonte: Elaborada pelo autor).

4.3 Análise dos resultados dos experimentos

O protocolo MAC ponto a ponto segue a estrutura Mestre-Escravo para a comunicação de dados efetivo e de controle. Foi introduzido um conceito de troca de mensagens onde o meio de comunicação fosse melhor aproveitado em ambos os sentidos, com o mínimo de informações de controle e maior quantidade de quadro de dados. Com base nisso, os experimentos visaram investigar:

A taxa de dados obtida no enlace com e sem o protocolo MAC PTP: foram realizadas uma série de transmissões unidirecional no enlace, com a intenção de comparar a quantidade de bits que podem ser transmitidos em um determinado tempo, em uma rede com o protocolo MAC CSMA/CA e depois como protocolo desenvolvido. Em vista disso e com os resultados apresentados na seção anterior, tem-se que o desempenho do MAC ponto a ponto foi superior, pela quantidade de dados transmitidos no mesmo período de tempo, proporcionando praticamente o dobro da vazão sobre o MAC CSMA/CA. Sendo assim, pode-se aferir que as técnicas empregadas para adotar um método que melhore o aproveitamento do meio de transmissão, tiveram seu funcionamento validado.

A média do atraso de quadros no enlace ponto a ponto com e sem o protocolo MAC PTP: baseado nos dados enviados afim de calcular a largura de banda do enlace, foi analisado também o tempo que cada pacote demorou para chegar desde a origem até o destino.

Os números apontados na seção 4.2, expõe que os recursos do protocolo MAC ponto a ponto contrastando com o protocolo padrão conseguiram diminuir em pelo menos 1 milissegundo, a probabilidade de atraso no enlace em questão. Porém esse intervalo poderá ser ainda menor, dependendo da distância em que os dispositivos sem fio irão operar e com uma segunda versão do código do protocolo, ajustando a troca de mensagens para a recuperação do Poll.

A distribuição de probabilidade dos atrasos de quadros no enlace, com e sem o protocolo MAC ponto a ponto: Com uma análise sobre os gráficos de probabilidade de ocorrência dos intervalos entre quadros, ficou visível a grande diferença dos intervalos de quadros recebidos entre os protocolos. O protocolo MAC ponto a ponto, com a grande proporção de intervalos em torno de 500us, obteve intervalos menores, pois o MAC reprime os quadros na fila e envia-os seguidamente após a recepção do *Poll*. Sobre os intervalos do protocolo CSMA/CA, utilizando dos mecanismos de controle e prevenção, os intervalos se concentraram à volta de 11ms e com espalhamento temporal regular ao longo de outros intervalos. Não se pode concluir que essa grande diferença de intervalos foi gerada somente pelo protocolo em questão. A própria ferramenta *iperf*, ao escolher uma taxa de dados para a medição com UDP, pode ter gerado uma *stream* regular de datagramas de tamanho máximo (para MTU de 1500 bytes). Ele é capaz de ter calculado um período de 11 ms para as transmissões para se obter uma determinada taxa.

Análise dos pacotes perdidos: Com o uso do protocolo ponto a ponto e, de acordo com os dados resultantes da comunicação *Iperf*, deduz-se que, considerando a quantidade de quadros enviados sobre a quantidade de quadros perdidos, o quadro ACK não se faz necessário para o cenário em questão. Isso se deve a forma como o acesso ao meio é controlado. O comportamento da comunicação do protocolo, praticamente, anula as chances de transmissão simultânea entre as duas estações, evitando risco de colisões. Porém também deve-se levar em consideração o ambiente em que o experimento foi realizado.

5 *Conclusão*

Em redes que utilizam o padrão IEEE 802.11, quando há várias estações com filas de quadros para transmissão, é utilizado uma variedade de recursos para evitar que as estações comecem a enviar simultaneamente, prevenindo possibilidade de colisão entre quadros. Os métodos empregados diminuem as chances de perdas de dados, porém quando não há uma grande quantidade de nodos na rede, mais especificamente dois, localizados em um ambiente com sinal de interferência despresível, as técnicas aplicadas acabam sendo desnecessárias, pelo fato de as chances de extravio do pacote por colisão serem reduzidas. Isto mostra que a eficiência do meio poderia ser melhor aproveitada se a forma de acesso ao meio fosse aprimorada, levando em conta um mais satisfatório aproveitamento do meio de transmissão e, também evitando riscos de colisão.

Esse trabalho teve como propósito o desenvolvimento de um protótipo, configurado em uma rede IEEE 802.11, com um mecanismo capaz de reduzir o tempo de espera antes de cada transmissão e como decorrência disso, um aumento na vazão. O mecanismo estruturado busca a adaptação do método de controle de acesso ao meio para um melhor rendimento em um enlace ponto a ponto. Por meio de uma mensagem de autorização, o acesso ao meio é controlado e temporizado designando a duração de cada transmissão.

A implementação de um código habilitado a configurar o *driver* do dispositivo de uma interface sem fio, tornou viável ajustes dos parâmetros identificados como controladores do acesso ao meio do padrão CSMA/CA, usados para proteger os pacotes contra colisões. Somente a configuração da interface não houve uma redução drástica nos intervalos de quadros. Ainda que estes estivessem reduzidos, houve uma disputa pelo meio. Assim, foi necessário a alteração da forma de comunicação entre os dispositivos para acesso ao canal de transmissão. Integrando com o código de ajuste do *driver* da interface, foi construído um protótipo de MAC, denominado MAC ponto a ponto, com objetivo de alterar a forma como o MAC CSMA/CA acessa o meio em uma rede *wifi*, passando de um acesso ao meio aleatório para um acesso controlado, temporizado e previsível. Com o auxílio de APIs usadas para a comunicação com o Kernel do linux, aonde

encontra-se o hardware da interface WLAN, o protótipo foi desenvolvido conforme o conceito do protocolo Mestre-Escravo, utilizado em redes cabeadas. Foi realizado uma série de testes afim de comparar o desempenho do MAC ponto a ponto com o MAC CSMA/CA, em uma conexão sem fio. Os experimentos mostraram que o desempenho do MAC ponto a ponto em conjunto com a configuração do *driver* da interface, foi superior tanto no ganho de vazão quanto na percepção da redução do tempo de espera para uma transmissão.

Ao longo do desenvolvimento do trabalho foram encontradas certas dificuldades, que apesar da base adquirida nas matérias de Redes de Computadores e Programação no decorrer do curso, foi necessário realizar uma grande pesquisa em torno da interface de programação Netlink, para fazer a comunicação do driver com o *Kernel* do Linux. Estudo sobre a arquitetura do S.O. Linux e sua compatibilidade com os dispositivos utilizados. A linguagem de programação em C, que mesmo com alguma experiência foram utilizadas uma série de estruturas e bibliotecas como: Packet Socket, a criação da interface TUN e também as chamadas de sistemas. Todas essas só foram conhecidas e tratadas ao longo da implementação do trabalho.

- Trabalhos Futuros: Uma segunda versão do protótipo pode ser desenvolvida para tornar seu desempenho ainda mais eficaz ajustando a forma de comunicação entre os dispositivos sem fio ou até mesmo a implementação de um código em espaço do *Kernel* do Linux, embora a confecção seja mais complicada o tempo de processamento é menor se comparado a uma aplicação em espaço do usuário. Como o meio de transmissão é imprevisível, não é possível garantir que não haverá perda de pacotes, mesmo com um acesso controlado. Para tentar evitar que muitos quadros sejam perdidos e sem retransmissão, o protocolo pode ser desenvolvido com um método de confirmação multi quadros, semelhante ao utilizado pelo 802.11n, tornando o sistema mais precavido contra extravio de dados.

Referências Bibliográficas

[1] TANENBAUM, A. *Redes de Computadores, 4a edição.*

[2] DEMARCH, D.D. *Uma Proposta de Escalonamento Confiável para Redes Sem Fio Baseadas no Padrão IEEE 802.11/11e.* Dissertação (Mestrado) - Universidade Federal de Santa Catarina, 2007.

[3] IEEE. *Part 11: LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* New York, NY, March 2012. IEEE Standard.

[4] HEINZEN, S.L. *Redução do tempo de varredura na transição de BSS em redes IEEE 802.11.* Monografia (Trabalho de Conclusão de Curso) - Instituto Federal de Santa Catarina, 2011.

[5] FOROUZAN, B. *Comunicação de Dados e Redes de Computadores - Quarta edição.* Editora MCGRAW-HILL INTERAMERICANA, Janeiro de 2008.

[6] KUROSE, J. *Redes de Computadores e a Internet - Terceira edição.* 2005.

[7] KUROSE, J. ROSS, K. *Computer Networking: A top-down approach - Sexta edição.*

[8] GARCIA, G.L.U. *REDES 802.11 (Camada de Enlace).* Disponível em:<http://www.gta.ufrj.br/grad/01_2/802-mac/R802_11.htm.> Último acesso em Janeiro de 2016.

[9] TELECO. *Redes LAN/MAN Wireless II: Protocolo 802.11.* Disponível em: <http://www.teleco.com.br/tutoriais/tutorialrwlanman2/pagina_4.asp> Último acesso em Janeiro de 2016.

[10] JUNNIOR, O.B.*Planejamento da capacidade de tráfego de voz sobre rede Wi-Fi com foco na cobertura e interferência.* Monografia (Trabalho de Conclusão de Curso) - Universidade Regional de Blumenau Centro de Ciências Tecnológicas Departamento de Engenharia Elétrica e Telecomunicações, 2010.

[11] LEONARDO, E. *Redes de Computadores 4. Subcamada de Controle de Acesso ao Meio.* Disponível em: <http://docplayer.com.br/931275-Redes-de-computadores.html> Último acesso em Janeiro de 2016.

[12] MAH, C. *Courses/Computer Science/CPSC 441.W2014/Chapter 6: Wireless and Mobile Networks - Wikipedia, The Free Encyclopedia.* Disponível em: <http://wiki.ucalgary.ca/page/Courses/Computer_Science/CPSC_441.W2014/Chapter_6:_Wireless_and_Mobile_Networks> Último acesso em Janeiro de 2016.

[13] MIRANDA, F. *CSMA/CA X CSMA/CD*. Disponível em:<http://routios.blogspot.com.br/2012_07_01_archive.html>. Último acesso em Janeiro de 2016.

[14] CHAOUCHI, H. MAKNAVICIUS, M.L. *Wireless and Mobile Network Security, Security Basics, Security in On-the-shelf and Emerging Technologies* Wiley-ISTE, 2009.

[15] ATHEROS COMMUNICATIONS. *AR9271 Single-Chip 1x1 MAC/BB/Radio/PA/LNA with USB Interface for 802.11n 2.4 GHz WLANs*. Novembro de 2011.

[16] NETLINK PROTOCOL. *Netlink Protocol Library Suite (libnl)*. Disponível em:<https://www.infradead.org/~tgr/libnl/>. Último acesso em Julho de 2016.

[17] MANUAL LINUX. *Linux Programmer's Manual*. Disponível em:<http://man7.org/linux/man-pages/man7/packet.7.html>. Último acesso em Julho de 2016.