UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS ARARANGUÁ

Luis Alberto Kurten Seemann

UMA PLATAFORMA EXPERIMENTAL PARA A ANÁLISE DE DESEMPENHO DE REDES SEM FIO PADRÃO IEEE 802.11

Araranguá, Dezembro de 2013.

Luis Alberto Kurten Seemann

UMA PLATAFORMA EXPERIMENTAL PARA A ANÁLISE DE DESEMPENHO DE REDES SEM FIO PADRÃO IEEE 802.11

Trabalho de Conclusão de Curso submetido à Universidade Federal de Santa Catarina, como parte dos requisitos necessários para a obtenção do Grau de Bacharel em Tecnologias da Informação e Comunicação.

Orientador: Prof. Dr. Ricardo Alexandre Reinaldo de Moraes

Araranguá, Dezembro de 2013.

Luis Alberto Kurten Seemann

UMA PLATAFORMA EXPERIMENTAL PARA A ANÁLISE DE DESEMPENHO DE REDES SEM FIO PADRÃO IEEE 802.11

Este Trabalho de Conclusão de Curso foi julgado aprovado para a obtenção do Título de "Bacharel em Tecnologias da Informação e Comunicação", e aprovado em sua forma final pela Curso de Graduação em Tecnologias da Informação e Comunicação.

Araranguá, Dezembro de 2013.

Vilson Gruber, Dr.
Coordenador do Curso

Prof. Dr. Ricardo Moraes
Orientador

Banca Examinadora:

Prof. Anderson Luiz Fernandes Perez, Dr.

Prof. Analúcia Schiaffino Morales, Dr.

Dedico meu trabalho a minha família, aos meus pais, Saulo Marcio Seemann, Albertina Kurten e a meu irmão Daniel Kurten Seemann.

AGRADECIMENTOS

Ao meu orientador, professor Ricardo Alexandre Reinaldo De Moraes por ter me acompanhado, não somente neste trabalho final mas desde o início do curso, a todos meus colegas que me auxiliaram de alguma maneira a realização deste trabalho. A minha família que sempre me ajudou, no que eu precisasse e a Universidade Federal de Santa Catarina.

Nós somos aquilo que fazemos repetidamente. Excelência, então, não é um modo de agir, mas um hábito

Aristóteles

RESUMO

Uma forte tendência atual na área de automação industrial, residencial e hospitalar, é o da transferência de tempo real, através de uma estrutura de redes sem fio, de dados relacionados a sensores, atuadores e diagnóstico dos sistemas. Porém, as tecnologias e técnicas atuais ainda são incapazes de controlar tarefas com requisitos temporais (*dead lines*) críticos utilizando estas redes. Este trabalho de conclusão de curso (TCC) está enquadrado nesta área de pesquisa e durante o seu desenvolvimento, um cenário experimental para avaliar a comunicação de tempo real em redes sem fio, compatíveis com o padrão IEEE 802.11, foi construído. Experimentos foram realizados para avaliar o desempenho de uma rede composta de estações tempo real (TR) que compartilha o canal de comunicação com estações não tempo real (NTR). O conjunto de estações NTR transmite tráfego de fundo para simular um ambiente real que é aberto. Neste TCC, apresentam-se resultados preliminares para alguns valores das janelas de contenção (CW) no qual é um dos mecanismos responsáveis para fornecer prioridades na função EDCA, onde foi possível verificar que os valores definidos no padrão não são adequados em ambientes de comunicação abertos. Os resultados experimentais obtidos demonstram que há uma alta taxa de perda de dados.

Palavras-chave: Redes de computadores. Padrão IEEE 802.11e. Padrão IEEE 802.15.4. Automação.

ABSTRACT

A strong current trend in the automation industrial, residential and hospital areas is the real-time transmission, through a wireless network of data related to sensors, actuators and systems diagnosis. However, current techniques and technologies are still unable to control tasks with critical deadlines requirements using these networks. This work has been developed in this research issue, and during its development, an experimental scenario to evaluate the real-time communication in wireless networks compatible with the IEEE 802.11 was built. Experiments have been conducted to evaluate the performance of a network composed of real-time (RT) stations that shares the communication channel with no real-time (NRT) stations. The set of NRT transmits background traffic to simulate a real open communication environment. In this work, some preliminary results for a set of values of contention window (CW), in which that is one of the mechanisms responsible for provide priorities in EDCA function, where it was possible to conclude that the defined values of the IEEE 802.11e are not adequate to work in open communication environments. The obtained experimental results shows that there are a high packet lost rate.

Keywords: Computer Networks, IEEE 802.11 standard, Automation Systems.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

SUMÁRIO

1 INTRODUÇÃO

A necessidade de alternativas mais eficientes e cômodas para prover a comunicação entre dispositivos computacionais está fazendo das redes sem fio a forma mais utilizada em ambientes domésticos e corporativos. As redes padrão IEEE 802.11 (IEEE COMPUTER SOCIETY, 2012), também conhecidas como redes WiFi, são hoje o padrão *de facto* em conectividade para redes locais sem fio (MORAES et al., 2007, 2010).

Segundo Moraes, Vasques e Portugal (2010), atualmente, um número significativo de trabalhos de pesquisa está sendo efetuado no desenvolvimento de redes sem fios de alto desempenho e esta tendência é uma consequência da crescente utilização de comunicações sem fios em ambientes de escritório e doméstico. Então, é provável que num futuro próximo, a ampla disponibilidade de soluções de redes sem fios irá também gerar um padrão de facto para comunicação sem fios na automação, onde o conjunto de protocolos normalizados IEEE 802.11 (IEEE COMPUTER SOCIETY, 2012) é um dos principais candidatos. Ainda segundo Moraes, Vasques e Portugal (2010), a utilização de uma infraestrutura de redes sem fio na automação apresenta grandes desafios. Os requisitos das aplicações nesta área são muito específicos, onde além do tráfego genérico, similar àquele encontrado em ambientes de escritório/doméstico, existe tráfego com requisitos de tempo real (TR). Este tráfego está tipicamente associado a aplicações de controle, para as quais os dados devem ser periodicamente transferidos entre sensores, controladores e atuadores de acordo com metas temporais. Além disso, o meio de comunicação em ambientes sem fio é essencialmente aberto. Isso quer dizer que nestes ambientes, um conjunto de estações externas pode formar, por exemplo, uma rede *ad hoc* e transmitir na mesma faixa de frequência que as estações de TR, perturbando o tráfego de TR, que poderá não ser capaz de cumprir as suas especificações temporais. Assim, as abordagens utilizadas em redes cabeadas, que se baseiam no conhecimento de todo o tráfego da rede, não são mais adequadas. Como consequência, atualmente, há muitos problemas em aberto no domínio das comunicações para a automação.

O presente Trabalho de Conclusão de Curso (TCC) colabora para demonstrar que as abordagens tradicionais para suportar comunicações de tempo real em redes CSMA (*Carrier Sense Multiple Access*) não são adequadas, uma vez que impõem modificações da interface de rede das estações externas que compartilham o mesmo meio de comunicação, ou seja, são baseadas no controle estrito das oportunidades

de comunicação de cada dispositivo, o que requer um ambiente de comunicação completamente sob a esfera-de-controle da arquitetura de comunicação de TR.[1]

1.1 PROBLEMÁTICA E JUSTIFICATIVA

Especialmente em áreas com grande densidade demográfica, há sempre um grande número de dispositivos sem fio em operação. Atualmente, as principais interfaces de redes sem fio são organizadas de acordo com os padrões IEEE 802.11, IEEE 802.15.1 e IEEE 802.15.4. Apesar destes padrões serem utilizados com diferentes propósitos, há uma característica em comum entre eles, que é a transmissão utilizando a mesma faixa de frequência de transmissão, ou seja, a frequência de 2.4GHz - 2,4835GHz, que equivalem a 83,5MHz. Esta faixa de frequência é reservada ao uso sem licença, da chamada Banda ISM (Industrial, Scientific, Medical) (WETHERALL; TANENBAUM, 2011). Considerando a coexistência de diversos dispositivos na mesma área de alcance, as transmissões realizadas por um dispositivo pode interferir nas transmissões de todos os outros, principalmente, se eles estiverem utilizando o mesmo canal de comunicação, Figura 1.

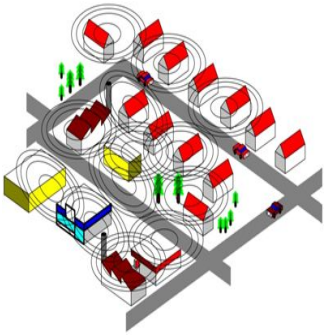

Figura 1 – Sobreposição de canais de comunicação.

Observando a Figura 1, pode-se constatar que haverá uma competição pelo uso do meio físico quando há a coexistência de sinais

[1] O conceito definido da esfera de controle por (KOPETZ, 1998). Foi usado, em conformidade com a definição de Kopetz. Sempre que uma entidade RT está no controle da esfera de um subsistema, que pertence a um subsistema possui a autoridade para mudar todo o valor desta entidade RT. Fora da sua esfera de controle, o valor de a entidade pode ser observado, mas não pode ser modificada.

emitidos por diferentes dispositivos de comunicação. Em redes cabeadas essa coexistência é possível, por exemplo, através de mecanismos de suavização de tráfego aplicados às estações que se deseja estabelecer uma maior prioridade de comunicação (Lobello; KACZYNSKI; MIRABELLA, 2005). Infelizmente, esta e outras abordagens que consideram um meio de comunicação fechado não são adequadas para ambientes de comunicação sem fio, uma vez que, não é possível impor qualquer restrição de tráfego às estações que não pertencem ao mesmo domínio de comunicação (MORAES et al., 2010). Então, pode-se prever a possibilidade de intersecção de faixas espectrais de frequência, o que pode causar problemas na comunicação. Para que haja soluções para esses tipos de deficiências, faz-se necessário o estudo aprofundado dos níveis interferências. Portanto, este trabalho justifica-se por abordar um problema de especial relevância na comunicação em redes sem fio, que pode ser resumido pela seguinte questão:

> Como garantir que os requisitos temporais dos dados de controle sejam respeitados, quando o meio de comunicação é partilhado com tráfego genérico e o perfil de carga não controlado? (MORAES, 2007).

Em resumo, pretende-se abordar estratégias de melhoria da comunicação para cada grupo de dispositivos sem fio que está operando no mesmo canal, o que não é uma tarefa trivial.

1.2 OBJETIVOS

Este TCC foi desenvolvido no âmbito do Projeto "Mecanismos de Comunicação de Tempo Real em Redes Sem Fio", coordenado pelo Prof. Ricardo Moraes. O referido projeto tem os seguintes objetivos gerais:

- Pesquisar e desenvolver soluções inovadoras para o estabelecimento de comunicações de tempo real em redes sem fio, compatíveis com o padrão IEEE 802.11.
- Avaliar experimentalmente as soluções propostas em aplicações reais.

Para atingir os objetivos supracitados, uma série de objetivos específicos devem ser alcançados no projeto. Assim sendo, especificamente neste TCC, definiu-se como principal objetivo desenvolver uma

plataforma experimental, baseada na utilização de plataformas de hardware/software abertas, para avaliar experimentalmente soluções inovadoras ao estabelecimento de comunicações de tempo real em redes sem fio, compatíveis com o padrão IEEE 802.11.

Para atingir o objetivo principal deste TCC, os seguintes objetivos específicos foram definidos:

- Estudar as limitações do padrão 802.11e para prover garantia de Qualidade de Serviço (QoS).
- Definir um gerador de cargas sintéticas responsável pela geração de tráfego com características reais.
- Analisar o nível de QoS provido pelas redes IEEE 802.11e em ambientes abertos, através de uma análise experimental. Nesta etapa, será estudado, especificamente, o comportamento do mecanismo EDCA (*Enhanced Distributed Channel Access*) no suporte de transmissão de tráfego com características de tempo real.

1.3 METODOLOGIA

Para atingir os objetivos anteriormente descritos, três atividades principais foram definidas, as quais são descritas abaixo:

1.3.1 Atividade 1: Estudo das redes sem fio e seleção de ferramentas

Nesta etapa, deve-se estudar as principais características relacionadas aos padrões Wireless, incluindo as normas, as aplicações e os trabalhos de pesquisas atuais. Com foco no protocolo IEEE 802.11e, que provê diferentes níveis de QoS às aplicações, incluindo transmissões de voz e vídeo. Esta emenda provê QoS através da inclusão de uma função de coordenação, chamada HCF (*Hybrid Coordination Function*) que fornece dois tipos de mecanismos de comunicação: o EDCA e o HCCA. O primeiro é o que fornece prioridades diferentes para as transmissões e representa uma evolução ao mecanismo DCF. Além do padrão IEEE 802.11 (IEEE COMPUTER SOCIETY, 2012), também foram selecionados os seguintes trabalhos Moraes (2007) e Moraes et al. (2010), que demonstram que o protocolo IEEE 802.11e pode ser melhor aproveitado alterando os parâmetros de configuração *default* por outros valores. A

determinação destes novos valores são dependentes do tráfego da rede e das características da aplicação.

Posteriormente, deve-se realizar uma pesquisa sobre os principais geradores de cargas sintéticas - que é um software utilizado para a geração de tráfegos similares a aplicações reais. Os geradores devem ter os seguintes requisitos: licença livre, envio de taxas constantes e variáveis, geração de gráficos e principalmente, a funcionalidade de envio de tráfego com características reais, como voz e vídeo.

Por fim, deve-se pesquisar, selecionar e comprar placas de redes adequadas para a realização dos experimentos. É importante mencionar que há recursos financeiros disponíveis para esta e outras aquisições do âmbito do projeto que este TCC está inserido.

1.3.2 Atividade 2: Construção do Cenário Experimental

Nesta etapa, deve-se construir um cenário experimental utilizando as placas wireless em conjunto com os softwares de geração de tráfego selecionados. Esta plataforma experimental deve conter estações geradoras de tráfegos (NTR), que causam interferências em nas transmissões das outras estações (TR).

1.3.3 Atividade 3: Avaliação Experimental

A última etapa deste TCC consiste em avaliar a plataforma desenvolvida através de um estudo de caso. O principal objetivo desta análise é verificar o funcionamento da plataforma implementada. Como estudo de caso, pretende-se verificar a influência das estações-NTR sobre as estações-TR. Para comparar os resultados do padrão IEEE 802.11 *versus* as novas soluções propostas, as estações-TR estarão utilizando o protocolo padrão IEEE 802.11 num primeiro experimento e uma das soluções propostas, num segundo experimento. Nestes experimentos, as novas soluções propostas estão simplesmente relacionadas com diferentes parametrizações do mecanismo EDCA. Porém, futuramente, este cenário experimental será utilizado para avaliar novas propostas no âmbito do projeto global. Em ambos os cenários haverá um conjunto de estações-NTR (estações externas) transmitindo dois tipos de tráfego: voz (VO) e background (BK). As estações-TR estarão transmitindo somente tráfego de voz. O tráfego de voz será obtido por um gerador de cargas sintéticas, onde os dados são transmitidos a

uma taxa de 64 kbps, com 160 bytes. O tráfego BK, também obtido através do gerador de cargas sintéticas, será transmitido a uma taxa de 1024 kbps e os pacotes terão um tamanho de 1500 bytes. Pretende-se analisar um cenário experimental com no mínimo 4 estações-TR , na presença de um número variável de estações-NTR (2, 4 ou 6). Neste cenário serão avaliadas métricas como: atraso médio e máximo dos pacotes, tamanho médio da fila, throughput, taxa de perdas de pacotes etc.

1.4 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado, além desta introdução, em mais 3 capítulos que abordam os seguintes conteúdos da seguinte maneira:

O Capítulo 2 apresenta o protocolo CSMA, seu funcionamento além das suas variações CSMA/CD utilizadas em redes ethernet e CSMA/CA nas redes 802.11. O protocolo que controla o acesso ao meio é o ponto chave em qualquer rede que utiliza um meio físico de comunicação compartilhado, como é o caso das tradicionais redes Ethernet (em barramento) e das redes WiFi. Neste capítulo, descreve-se o funcionamento do protocolo CSMA (*Carrier Sense Multiple Access*) para os casos de detecção de colisão (CSMA/CD - *CSMA with collision detection*) e de evitar colisão (CSMA/CA - *CSMA with collision avoidance*).

O Capítulo 3 descreve sobre a construção do cenário experimental onde é apresentado as ferramentas utilizadas, os equipamentos para a realização dos experimentos e os scripts de apoio para geração de tráfego.

No Capítulo 4 são apresentadas os experimentos, bem como a descrição do cenário construído e o resultado dos experimentos realizados.

O Capítulo 5 apresenta a conclusão e as dificuldades encontradas

2 O PROTOCOLO CSMA

O protocolo que controla o acesso ao meio é o ponto chave em qualquer rede que utiliza um meio físico de comunicação compartilhado, como é o caso das tradicionais redes Ethernet (em barramento) e das redes WiFi. Neste capítulo, descreve-se o funcionamento do protocolo CSMA (*Carrier Sense Multiple Access*) para os casos de detecção de colisão (CSMA/CD - *CSMA with collision detection*) e de evitar colisão (CSMA/CA - *CSMA with collision avoidance*).

2.1 DESCRIÇÃO DO PROTOCOLO

O protocolo CSMA define uma família de protocolos de controle de acesso ao meio, onde as estações que estão disputando um meio físico compartilhado de comunicação devem ouvir o meio antes de iniciar a transmissão. Basicamente, esta família de protocolos tem o seguinte comportamento:

- Quando uma estação quer transmitir, ela ouve o meio de comunicação;
- Se o meio de comunicação está ocioso, a estação iniciará a transmissão (imediatamente, ou depois de um intervalo de tempo pré-definido, dependendo do protocolo específico);
- Se o meio de comunicação está ocupado (*i.e.* uma outra estação está transmitindo), a estação adiará sua transmissão para um instante de tempo futuro que depende do protocolo utilizado;
- Uma colisão ocorrerá sempre que duas (ou mais) estações verificarem que o meio está livre e transmitirem simultaneamente.

Os métodos de controle de acesso ao meio que são implementados por diferentes protocolos de comunicação diferem em como os intervalos de tempo de espera antes de transmitir são avaliados, após a detecção do meio ocioso, ou antes de retransmitir após uma colisão.

2.1.1 O protocolo CSMA/CD

O CSMA com *Detecção de Colisão* (CSMA/CD) é o protocolo implementado na camada de enlace das redes locais IEEE 802.3 (IEEE

COMPUTER SOCIETY, 2000). Nesta camada, os quadros são transmitidos seguindo os padrões IEEE 802.3 ou Ethernet com, respectivamente, os seguintes formatos (Figures 2 and 3):

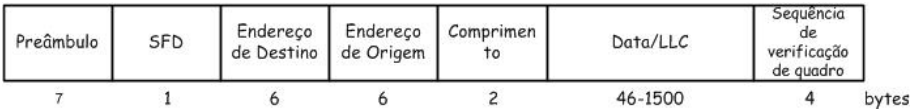

Figura 2 – Formato do quadro 802.3.

Figura 3 – Formato do quadro Ethernet.

Os padrões IEEE 802.3 e Ethernet são muito similares. A principal diferença está na subcamada de controle lógico, que é ausente no padrão IEEE 802.3 (sendo coberto pelo padrão IEEE 802.2 (IEEE COMPUTER SOCIETY, 1998)). No entanto, ambos os padrões são compatíveis e o termo Ethernet é utilizado como um nome popular para o padrão IEEE 802.3. Neste TCC, os termos IEEE 802.3/Ethernet são usados intercambiáveis para designar o padrão IEEE 802.3. Para uma rede de 10/100 Mbps, o conjunto de parâmetros usados é apresentado na Tabela 1.

Tabela 1 – Parâmetros IEEE 802.3.

Parameters	**Values**	**Bit Rate**	
		10 Mbps	100 Mbps
SlotTime	512 bits	51.2 μs	5.12 μs
InterFrameGap	96 bits	9.6 μs	0.96 μs
JamSize	32 bits	3.2 μs	0.32 μs
AttemptLimit	16	-	-
BackoffLimit	10	-	-
MaxFrameSize	12208 bits	-	-
MinFrameSize	512 bits	-	-
AddressSize	48 bits	-	-

Basicamente, o protocolo CSMA/CD implementado, tanto pelo IEEE 802.3 quanto pelo padrão Ethernet, funcionam da seguinte maneira: sempre que uma estação tem uma mensagem para transmitir, se

o meio físico está ocioso, ele transmitirá imediatamente. Se for detectada uma colisão, todas as estações transmissoras encerram suas transmissões e enviam uma sequência de bloqueio (para assegurar que todas as estações reconheçam a colisão[1]). Quando a transmissão é abortada devido a uma colisão, a sua retransmissão será repetida depois de um atraso avaliado aleatoriamente (tempo de backoff) até que a mensagem seja transmitida com sucesso, ou, eventualmente, descartada (depois de um número máximo de 16 tentativas).

Uma das questões-chave do protocolo CSMA/CD é a determinação dos atrasos de backoff, o que é feito por meio da execução local do algoritmo *Binary Exponential Backoff* (BEB). Este algoritmo funciona da seguinte maneira: após o fim da sequência de bloqueio, o tempo é dividido em *slots*, cujo tamanho é igual ao tempo de *slot* ($SlotTime$), que é dado por $t_{backoff} = r \times T$, onde r é um inteiro aleatório na faixa de $0 \leq r \leq 2^k - 1$, k é menor entre n ou 10 e T é o tempo do slot em segundos. Isso significa que a estação aguardará entre 0 e $2^n - 1$ tempos de slot antes de retransmitir, sendo n o número de colisões. Finalmente, após 10 tentativas, o intervalo máximo de espera é fixo em, no máximo, 1023 tempos de slot, e depois de 16 tentativas uma falha é reportada e a transmissão abortada (Figura 4).

O protocolo CSMA/CD escalona as mensagens a serem transmitidas através de uma disciplina de fila aleatória, ou seja, a mensagem a ser transferida, após uma transmissão bem sucedida, é escolhida aleatoriamente entre o total de estações com mensagens prontas a transmitir. No entanto, Christensen (1996) demonstrou que o algoritmo BEB impõe uma disciplina LIFO (último a chegar, primeiro a sair), pois, uma estação que começa a tentar o envio de suas mensagens depois que um processo de colisão entre outras estações está em andamento, terá uma maior probabilidade para a aquisição do meio.

2.1.2 O protocolo CSMA/CA

O procedimento de detecção de colisão não pode ser usado em redes sem fio, uma vez que exigiria a implementação de um canal de comunicação *full-duplex*, capaz de transmitir e receber simultaneamente.

Nas redes 802.11, quando uma estação se liga a um AP (*access point*), quadros de dados são enviados e recebidos entre eles, porém

[1]Mais precisamente, ao detectar uma colisão, a estação sempre termina a transmissão do preâmbulo e o início do quadro (64 bits), se estes ainda não foram completamente transmitidos. Posteriormente, ele transmite uma sequência de bloqueio (32 bits), e depois para.

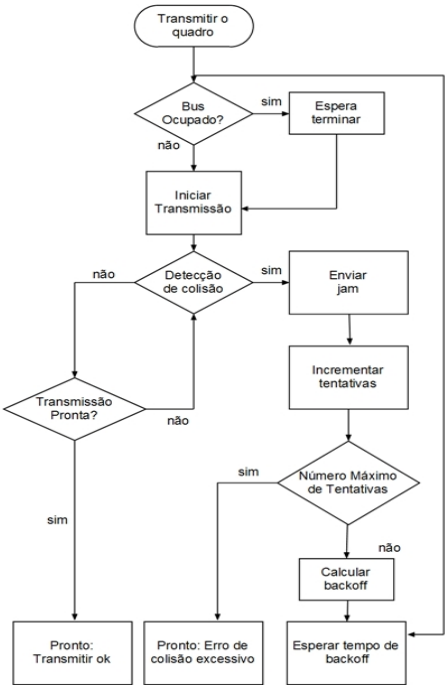

Figura 4 – Fluxograma do CSMA/CD.

pode ocorrer de várias estações transmitirem quadros ao mesmo tempo, acarretando duas estações enviando quadros ao mesmo tempo em um mesmo canal, no qual é necessário um protocolo de acesso múltiplo para que possa coordenar as transmissões. O funcionamento é similar aos, utilizados pelo CSMA/CD da Ethernet, porém apresentam diferenças, tais como, a de utilizar prevenção de colisão, ao invés de detecção de colisão. Alguns dos motivos que fazem com que não seja implementada a detecção são que para detectar colisões exige-se que uma maior potência do sinal recebido, já que ele é mais fraco do que o transmitido no adaptador 802.11 o que encarecia muito o hardware. Outros fatores determinantes para a utilização da prevenção é o problema do terminal escondido ou oculto, que se dá pelas obstruções físicas presentes no ambiente que podem impedir que estações escutem as transmissões umas da outras e o desvanecimento que consiste na força do sinal entre estações em uma determinada localização não são suficientes para que possam ser detectada transmissão de um e de outro, mas são fortes o

suficiente para interferir com uma terceira estação(KUROSE, 2010).

Consequentemente, o mecanismo de acesso ao meio do padrão IEEE 802.11 é o CSMA com *Collision Avoidance* (CSMA/CA), também chamado de *Função de Coordenação Distribuída* (DCF).

O MAC do padrão IEEE 802.11 introduz dois meios de acessar as funções de coordenação, o DCF que é obrigatório e função PCF (*Point Coordination Function*) que é opcional. O DCF[2] é o mecanismo básico do padrão IEEE 802.11. Quando uma estação quer transmitir, a estação detecta a portadora (*carrier sense*), se o meio está ocioso durante um intervalo de tempo específico (chamado DIFS - *Ditribbuted Interframe Space*), ele imediatamente transmite e as outras estações devem esperar até que o meio físico torne-se ocioso novamente, pelo menos, por um período de tempo igual a DIFS. Se o meio estiver ocupado, as estações selecionam um número aleatório, no intervalo de [*0, CW*], onde CW é inicialmente designado como CW_{min}. O parâmetro CW será aumentado, sempre que ocorrer uma falha de transmissão, ou seja, o estação de destino não responder com uma confirmação (quadro ACK).

Depois de uma tentativa de transmissão mal sucedida, outro intervalo de backoff será selecionado, onde o valor de CW é aumentado pela seguinte função $[(oldCW + 1) \times 2 - 1]$, com um limite superior dado por CW_{max}. Por outro lado, o temporizador de backoff diminui a cada intervalo de tempo que o meio é detectado ocioso. Assim que o temporizador de backoff torna-se igual zero, a estação pode repetir a sua tentativa de transmissão (Figura 5).

Além do mecanismo DCF, a subcamada MAC IEEE 802.11 também define a função PCF, que utiliza um esquema de *polling* centralizado para suportar a transmissão de dados síncronos sobre o mecanismo DCF.

O padrão IEEE 802.11e foi publicado (IEEE COMPUTER SOCIETY, 2005) como uma alteração do padrão original, destinado a proporcionar qualidade de Serviço (QoS). Esta alteração incorpora uma função adicional de coordenação chamada função de coordenação híbrido (HCF - *Hybrid Coordination Function*), que só é usada em configurações de rede com QoS de acordo com a Figura 6. O mecanismo HCF escalona o acesso ao canal, alocando oportunidades de transmissão (TXOP) para cada uma das estações. Cada TXOP é definida por um instante de tempo de início e um comprimento máximo e pode ser obtido através

[2]Um mecanismo adicional, RTS/CTS, é definido no padrão IEEE 802.11 para resolver o problema do terminal oculto e para obter um melhor comportamento de envio de mensagens longas. Para mais detalhes, consulte a (DENG; CHANG, 1999)

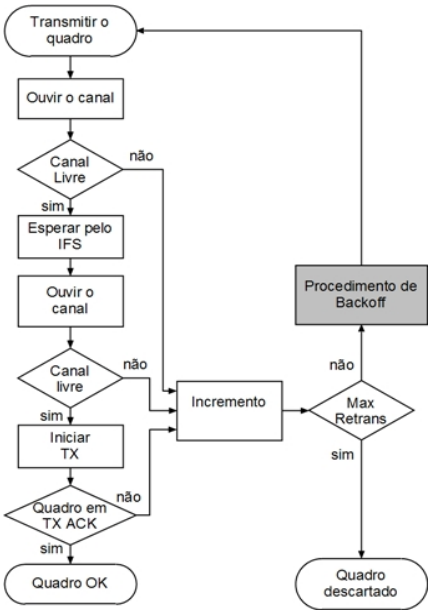

Figura 5 – Fluxograma do CSMA/CA.

Fonte: (MORAES, 2007).

dos dois mecanismos de acesso especificados pelo HCF: o *Enhanced Distributed Channel Access* (EDCA) e o HCF *Controlled Channel Access* (HCCA)(IEEE COMPUTER SOCIETY, 2005).

A ideia base do padrão EDCA foi primeiramente proposta por Deng e Chang (DENG; CHANG, 1999), onde as classes de mais alta prioridade utilizam a janela $[0, 2^{j+1} - 1]$ e as classes de mais baixa prioridade a janela $[2^{j+1}, 2^{j+2} - 1]$, onde j é o estágio atual do número de colisões.

A função EDCA implementa um mecanismo CSMA/CA para acesso ao meio físico sob o controle da função de coordenação HCF. Este mecanismo é projetado para fornecer serviços de transmissão diferenciados, com 4 níveis de prioridade. Esta função melhora o esquema DCF, já que cada quadro ao chegar na camada MAC com uma prioridade definida será mapeado em uma das quatro categorias de acesso (AC). Estas categorias são baseadas nos 8 níveis de prioridade definidos pelo padrão IEEE 802.1D (IEEE, 2004). O mapeamento das prioridades

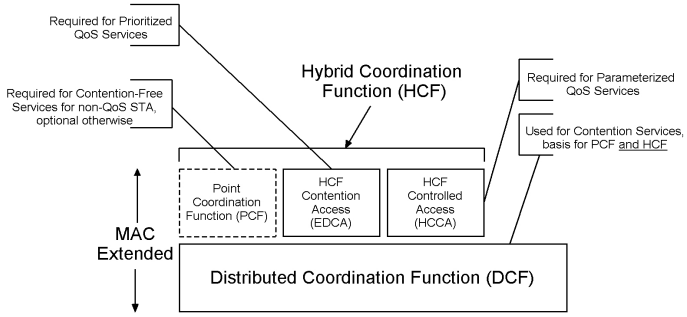

Figura 6 – Arquitetura IEEE 802.11e.

pode ser observado na Figura 7.

Diferentes níveis de serviço são fornecidos para cada uma das categorias, baseando-se em três mecanismos independentes:(i) o espaçamento entre os quadros (*Arbitration Interframe Space* (AIFS)); (ii) as oportunidades de transmissão (TXOP) e; (iii) o tamanho das janelas de contenção (CW). Os valores padrões utilizados pelo mecanismo EDCA para $AIFNS$, CW_{min} e CW_{max} são apresentados na Tabela 2, onde os valores dos parâmetros para o padrão 802.11g de aCW_{min} e aCW_{max} são definidos, respectivamente,por 31 e 1023 tempos de *slot*.

Tabela 2 – Parâmetros padrões no mecanismo EDCA.

AC	$\boldsymbol{CW_{min}}$	$\boldsymbol{CW_{max}}$	**AIFSN**
AC_VO	$(aCW_{min}+1)/4-1$	$(aCW_{min}+1)/2-1$	2
AC_VI	$(aCW_{min}+1)/2-1$	aCW_{min}	2
AC_BE	aCW_{min}	aCW_{max}	3
AC_BK	aCW_{min}	aCW_{max}	7

Em primeiro lugar, para uma estação operando sob o EDCA, cada quadro vai esperar durante um intervalo definido por *AIFS [AC]*, em vez de esperar durante DIFS (como para o caso do DCF). Somente após o canal permanecer ocioso durante um intervalo de tempo igual a *AIFS [AC]*, a estação começará a transmitir o quadro. A duração do intervalo *AIFS [AC]* é dado por:

Prioridade	(mesmo como prioridade de usuário 802.1D)	Designação 802.1D	AC	Designação (Informativa)
Baixo ↓ Alto	1	BK	AC_BK	Background
	2	__	AC_BK	Background
	0	BE	AC_BE	Best Effort
	3	EE	AC_BE	Best Effort
	4	CL	AC_VI	Video
	5	VI	AC_VI	Video
	6	VO	AC_VO	Voice
	7	NC	AC_VO	Voice

Figura 7 – Mapeamento das prioridades do 802.1D para EDCA.

Fonte: (IEEE, 2012).

$$AIFS[AC] = AIFSN[AC] \times aSlotTime + aSIFSTime \qquad (2.1)$$

onde o valor de *AIFSN[AC]* deve ser maior ou igual a 2 para todas as estações, exceto para os Pontos de Acesso com Qualidade de Serviço (QAPs) onde o valor deve ser maior ou igual a 1. Os valores de $aSlotTime$ e $aSIFSTime$ dependem das características físicas do canal. Por exemplo, para o padrão IEEE 802.11a PHY o menor intervalo entre os quadros (SIFS), que é definido pela variável $aSIFSTime$, é igual a $16\mu s$ e, $aSlotTime$ é igual a $9\mu s$. A Figura 8 ilustra a relação entre os múltiplos *AIFSs* no mecanismo EDCA.

Além disso, o mecanismo de EDCA introduz o conceito de oportunidades de transmissão (TXOP), ou seja, um intervalo de tempo durante o qual a estação mantém o controle de acesso ao meio, conforme listado na Tabela 3. Consequentemente, vários quadros podem ser transmitidos dentro de uma TXOP, sempre que houver mais do que um quadro pendente a ser transferido na categoria de acesso que o canal foi adquirido.

Finalmente, se uma estação quer transmitir um quadro, enquanto o canal estiver ocupado, ou o canal tornar-se ocupado antes da expiração do *AIFS [AC]*, o procedimento de backoff é invocado (terceiro mecanismo de diferenciação de tráfego). A janela de contenção é definida pelos parâmetros $aCW_{min}[AC]$ and $aCW_{max}[AC]$, em contraste com o mecanismo DCF onde os valores iniciais são selecionados

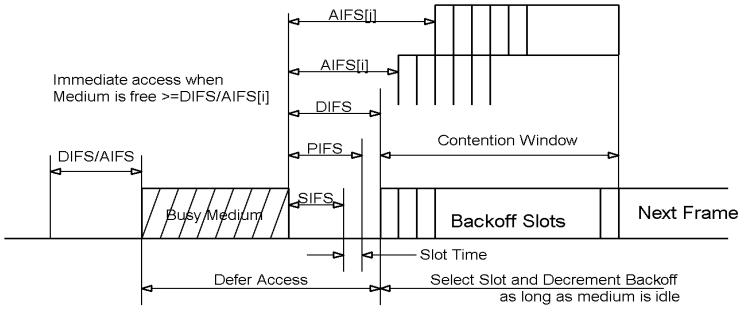

Figura 8 – Espaçamento entre quadros do mecanismo EDCA.

Fonte: (MORAES, 2007).

Tabela 3 – Parâmetros padrões para TXOP.

AC	**TXOP 802.11b**	**TXOP 802.11 a/g**
AC_VO	$3.264ms$	$1.504ms$
AC_VI	$6.016ms$	$3.008ms$
AC_BE	$0ms$	$0ms$
AC_BK	$0ms$	$0ms$

aleatoriamente entre o intervalo $[0, CW]$ definido pela camada física. No mecanismo EDCA, o procedimento de backoff seleciona um número aleatório, na intervalo *[0, CW]*, onde o tamanho de CW é inicializado por $aCW_{min}[AC]$. Quando a transmissão falha, o valor de CW é aumentado por *[(oldCW [AC] + 1) × PF] - 1* delimitado superiormente por $aCW_{max}AC]$, onde PF é o fator de persistência (o padrão é $PF = 2$). Por outro lado, o contador de backoff diminui sempre que o meio é detectado ocioso por *AIFS[AC]*.

Segundo Moraes et al. (2007)Em contraste com o mecanismo DCF, onde a estação tenta transmitir logo que o temporizador de backoff chega a zero, a estação EDCA inicia a transmissão no slot subsequente, após o contador chegar a zero. Consequentemente, não há nenhuma diferença no tempo de transmissão inicial entre DCF e EDCA, considerando o mesmo número de slots selecionados para backoff. Figura 10 ilustra a diferença entre os dois procedimentos (DCF e EDCA).

É importante destacar que no padrão IEEE 802.11 publicado em

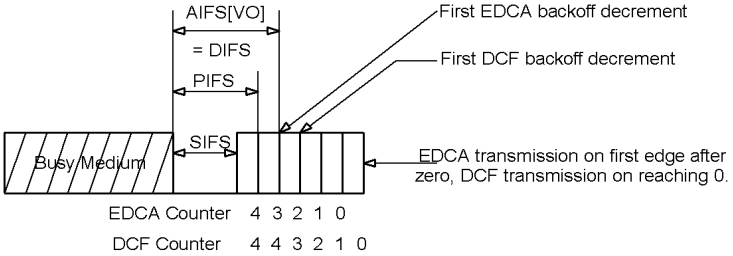

Figura 9 – Procedimentos de decremento nos mecanismos DCF e EDCA.

2012, já se encontram definidos os mecanismos de controle de acesso ao meio anteriormente definidos. A Figura 10 ilustra a arquitetura MAC atual.

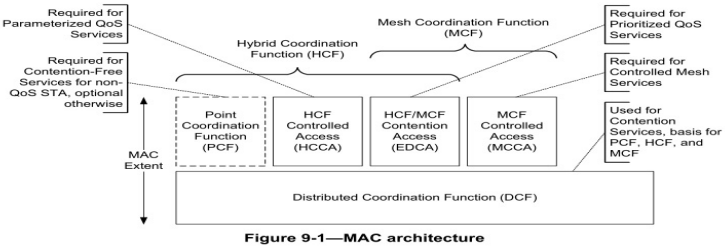

Figura 10 – Arquitetura MAC.

2.2 TRABALHOS RELACIONADOS

Os protocolos da família IEEE 802.11 estão entre os mais utilizados em redes locais sem fios (*Wireless Local Area Networks* - WLANs). Atualmente, estes protocolos têm sido utilizados em diversas aplicações essencialmente devido ao seu baixo custo de implementação e o seu ele-

vado desempenho.

Num passado recente, a grande maioria das análises de desempenho dos mecanismos de comunicação do padrão IEEE 802.11/802.11e eram direcionados para aplicações multimídia. Recentemente, diversos trabalhos de pesquisa analisaram estas soluções no domínio das aplicações industriais, onde os requisitos temporais de envio das mensagens são mais restritos. Estes trabalhos concluíram que este protocolo não é capaz de prover um nível de QoS aceitável. Por exemplo, uma possível solução para garantir comunicações de tempo real (TR) sob o mecanismo EDCA passaria por utilizar a categoria de mais alta prioridade (VO) para transmitir mensagens de TR. No entanto, quando se consideram ambientes de comunicações genéricos, onde o tráfego de TR tem de partilhar o meio físico com o tráfego gerado pelas restantes estações, verifica-se que utilizando a configuração típica das estações EDCA não é possível garantir os requisitos do tráfego de TR (MORAES et al., 2010). Outros trabalhos também demonstraram que o mecanismo de escalonamento do HCCA não é eficiente. Além disso, o HC constitui um ponto centralizado de falha, que pode conduzir toda a rede à falha.

Resumindo, suportar comunicações de TR em redes sem fios é reconhecidamente uma tarefa difícil. No entanto, recentemente, foram propostas novas soluções, sendo a sua maioria baseada na existência de fortes semelhanças entre as redes IEEE 802.3 (Ethernet) e IEEE 802.11 (Wi-Fi). Isto se deve ao fato de ambas utilizarem o mesmo mecanismo de acesso ao meio físico (CSMA). Desta forma, muitas das soluções propostas para garantir comunicações de TR utilizando o padrão IEEE 802.11 são adaptações das anteriormente formuladas para o IEEE 802.3.

Tradicionalmente, em ambientes com fios (wired) os requisitos de TR são garantidos através de um controle muito preciso do tráfego gerado por cada estação. Neste caso, segundo Moraes (2007) as propostas vão desde mecanismos que evitam as colisões (exemplos: TDMA, Master-Salve, token passing, micro-segmentação da rede), até mecanismos que resolvem deterministicamente as colisões (exemplo: CSMA/DCR), passando por técnicas que minimizam o número de colisões (exemplos: VTCSMA, Window Protocol, BLAM, traffic smoothing). Para o caso de ambientes sem fio, as abordagens seguem filosofias semelhantes, que vão desde mecanismos que evitam as colisões (exemplos: CP-Multipoll, Black Burst, HCCA) até mecanismos que minimizam o número de colisões (exemplo: EDCA). Contudo, muitas destas propostas implicam modificações no Software/Hardware de todas as estações da rede, o que na prática, se torna impossível de realizar devido as características dos ambientes de comunicação abertos, ou seja, a maioria das propostas

são baseadas no pressuposto de um meio físico de transmissão imune à interferências de estações externas.

Nesta seção, são descritas as principais soluções que foram propostas para melhorar o nível de qualidade de serviço (QoS) das redes IEEE 802.11. Por se tratar de um trabalho de conclusão de curso, deixa-se claro que não se tem a pretensão de cobrir todas as soluções que podem ser encontradas na literatura.

A primeira solução padronizada para prover QoS é a função PCF (*Point Coordination Function*), que foi proposta no padrão IEEE 802.11 original como um mecanismo de acesso opcional, onde o coordenador de acesso, normalmente localizado no AP (*Access Point*), implementa um esquema de *polling* centralizado para a transmissão de dados síncronos. Apesar do mecanismo PCF estar supostamente bem adaptado para suportar aplicações sensíveis ao atraso, a maioria das placas WLAN, nunca o implementaram devido à sua complexidade.

Conforme mencionado anteriormente, a última versão do padrão IEEE 802.11 (IEEE COMPUTER SOCIETY, 2012) incorporou a função de coordenação chamada HCF (*Hybrid Coordination Function*), que é usada somente em configurações com qualidade de serviço (QoS). No HCF são definidas duas funções: o EDCA (*Enhanced Distributed Channel Access*) - descrito anteriormente, que é um mecanismo de acesso baseado no DCF, e o HCCA (*HCF Controlled Channel Access*) que é um mecanismo baseado no controle do acesso ao meio das estações (tal como o PCF).

O HCCA foi proposto como uma melhoria do PCF. No entanto, alguns estudos demonstraram que este mecanismo pode não ser adequado para garantir os requisitos específicos das aplicações de TR (CASETTI et al., 2005). Para resolver este problema estão sendo propostas uma série de melhorias para reduzir o *polling overhead* do HCCA, por exemplo (SON et al., 2005).

Na literatura encontram-se diversas abordagens desenvolvidas com o objetivo de se oferecer serviços de comunicação de TR em redes sem fio. As principais soluções baseiam-se em esquemas de *Polling* (LEE et al., 2007; LO; LEE; CHEN, 2003), *Master-Slave* (MIORANDI; VITTURI, 2004) e *Token Passing* (ERGEN et al., 2004; CHENG et al., 2006). Algumas das propostas mais recentes são descritas abaixo.

Baseado no paradigma FTT (*Flexible Time Triggered*) Bartolomeu, Ferreira e Fonseca (2009) propõe o modelo WFTT (*Wireless FTT*). Este é um modelo *Master-Slave* que visa o aproveitamento tanto do *bandjacking* em ganhar prioridade no acesso ao meio quanto da flexibilidade, oportunidade e eficiência do FTT em suportar comunicações

de TR em aplicações com requisitos estáticos e/ou dinâmicos.

Em Friedrich, Alimenti e Reggiani (2010), é proposto um mecanismo de controle de acesso ao meio chamado WRTMAC (*Wireless Real Time Medium Access Control*), desenvolvido a partir de um esquema EDCA. Neste modelo, o AIFS é substituído por um RIFS (*Real-Time Inter-Frame Space*) com o objetivo de prover determinismo ao acesso ao meio. O principal problema desta abordagem é o fato de que embora o nome do novo IFS leve a pensar em uma verdadeira redução, isto na verdade não ocorre, pois o modelo substitui a função de backoff do padrão, definindo que o backoff será de $RIFS = DIFS + i \times aSlotTime$), onde i é um inteiro que define o nível de prioridade de tráfego transmitido. Sendo assim, se houver outra rede IEEE 802.11e sobreposta, pode ocorrer um atraso indeterminado quando tráfego de voz ou de vídeo são transmitidos.

Em Wu, Chiu e Sheu (2008), é proposto uma modificação do mecanismo EDCA, garantias de tempo real *soft* são providas através de um ajuste dinâmico do nível de prioridade do fluxo de dados. Este ajuste é baseado no atraso por salto (*hop*) efetuado, gerando um tempo de *backoff* não uniforme para retransmissões de mensagens de acordo com os requisitos individuais de atraso fim-a-fim. Contudo, este mecanismo não é capaz de evitar que o meio seja capturado por uma estação utilizando *aPIFSTime* (AP) ou então por um fluxo de dados com fragmentos MPDU, separados por *aSIFSTime*.

Em Villalón et al. (2008) é proposto o mecanismo B-EDCA, o qual é capaz de coexistir com estações DCF. Basicamente, o B-EDCA altera o valor do AIFS da categoria de acesso de mais alta prioridade (VO) para $SIFS + aSlotTime$, quando as estações estão no estado de *backoff*. Além disso, para manter a compatibilidade com o mecanismo HCCA, uma estação implementando o mecanismo B-EDCA deve esperar por um intervalo SIFS adicional quando o valor de *backoff* chegar a zero, ou seja $2 \times SIFS + aSlotTime$. O problema encontrado é que o valor de $SIFS + aSlotTime$ definido para a categoria de mais alta prioridade nas estações é o mesmo valor de PIFS e das categorias de voz e vídeo quando transmitidas do AP (usando AIFSN = 1). Neste caso, vão ocorrer colisões entre o AP e as estações implementando o mecanismo B-EDCA.

Em Moraes et al. (2007) é proposto o modelo VTP-CSMA, o qual é baseado em num mecanismo de passagem de token virtual (*Virtual Token Passing*). Este token circula entre as estações de TR autorizando o acesso ao meio. Como mecanismo complementar é utilizado um mecanismo que prioriza o acesso ao meio de estações TR frente a

estações NTR.

É digno de se destacar a proposta apresentada por Sobrinho e Krishnakumar (1999), a qual adapta o mecanismo EQuB Sobrinho e Krishnakumar (1998) para redes sem fio. O esquema denominado *Black-Burst* (BB), implementa um sistema MAC distribuído em redes *ad hoc*. Este mecanismo exige a substituição do esquema de transmissão aleatória. Todas as estações de TR que implementam a abordagem BB tentam acessar o meio de transmissão, após detectá-lo livre por um período igual a t_{med}, o qual é menor que o tempo de espera das estações padrão IEEE 802.11. Desta forma, as estações de TR têm prioridade sobre as demais estações no acesso ao meio.

De acordo com os trabalhos apresentados anteriormente, conclui-se que a arquitetura VTP-CSMA Moraes et al. (2007) e a abordagem BB Sobrinho e Krishnakumar (1999), Hwang e Cho (2005) são as únicas soluções que permitem a comunicação de TR em ambientes abertos. A principal desvantagem do BB é que o mecanismo obriga a modificação da camada MAC (e possivelmente também de partes da camada PHY), impedindo o uso de COTS hardware. Embora os mecanismos VTP-CSMA e RT-WiFi também modifiquem partes da camada MAC, estes mecanismos podem ser implementados em COTS (*Commercial Off-The-Shelf*) hardware (por exemplo, FPGA) sobre dispositivos padrão IEEE 802.11.

2.3 CONCLUSÃO

A descrição deste capítulo levanta os aspectos do funcionamento do mecanismo de acesso ao meio EDCA tanto para redes ethernet como para redes padrão IEEE 802.11. O protocolo CSMA/CD utilizado nas redes ethernet no qual possui detecção de colisão já que o meio utilizado é controlado. Já o CSMA/CA no qual evita colisões é o utilizados para redes WiFi, pelo fato de se utilizar um mecanismo de detecção de colisão possui um custo elevado já que o sinal transmitido teria que ser mais forte utilizando um canal *full-duplex* onde é possível enviar e transmitir dados simultaneamente. Outro aspecto abordado estão nos trabalhos relacionados onde é possível verificar que há um grande número de pesquisas sendo feitas para melhorar o desempenho das redes sem fio já que estas redes ainda apresentam grandes desafios, os quais estão relacionados com o provimento da Qualidade de Serviço (QoS) ás aplicações.

3 CONSTRUÇÃO DO CENÁRIO EXPERIMENTAL

Neste capítulo, são apresentadas as ferramentas utilizadas e avaliadas para a construção do cenário experimental.

3.1 REQUISITOS DO CENÁRIO A IMPLEMENTAR

Para a realização de experimentos relativos a avaliação de desempenho das redes sem fio, foi necessário construir uma plataforma capaz de avaliar os diversos mecanismos em situações realísticas. Esta plataforma é composta por um conjunto de hardware/software que funcionam de maneira integrada. A Figura 11 ilustra de forma genérica o projeto da plataforma de testes construída, onde as estações TR e NTR compartilham o mesmo meio de comunicação, com duas estações especiais (*sniffers*) que monitoram as transmissões. As estações TR são responsáveis por transmitir tráfego periódico, enquanto as estações NTR geram tráfego a fim de causar interferência, simulando um ambiente de comunicação aberto.

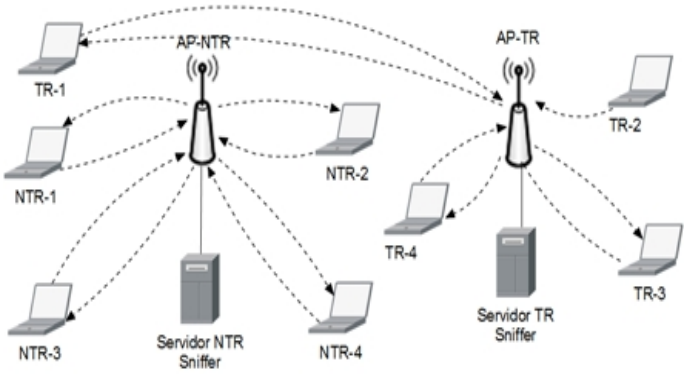

Figura 11 – Plataforma Genérica.

3.2 COMPONENTES DE HARDWARE

Durante o processo de definição da plataforma experimental, diversos adaptadores de rede foram avaliados, com o objetivo de selecionar o mais adequado para as estações TR. O principal requisito é que estes dispositivos implementem o mecanismo EDCA, conhecido comercialmente como WMM (*W*ireless Multimídia). Atualmente, poucos adaptadores e pontos de acesso implementam este mecanismo, além disso, muitos deles são configurados utilizando os parâmetros *default* e a alteração de valores consiste em alterar os parâmetros no driver e, consequentemente, na sua recompilação. Neste projeto foram avaliados os seguinte dispositivos: Adaptador 3COM modelo 3CRUSB275, Adaptador D-link modelo DWA-125, Adaptador Draytek modelo Vigor N65, ponto de acesso da CISCO modelo WRVS4400N e um ponto de acesso da Draytek modelo Vigor 2130VN.

3.2.1 Adaptador de Rede 3COM

O dispositivo da 3COM modelo 3CRUSB275 possui como características suportar redes sem fio, compatíveis com as seguintes camadas físicas 802.11a/b/g/n (2.4GHz e 5GHz). Este dispositivo suporta as atuais técnicas de autenticação tais como Wi-Fi Protect Access (WPA2) e encriptação de 128-bit Advance Encryption Standard (AES) e a adoção da tecnologia Wi-Fi Multimídia (WMM) que acrescenta Qualidade de Serviço (QoS) às estações (5TI, 2013). O chip utilizado pelo dispositivo é um Atheros ar9170, que possui drivers para as plataformas Windows. Porém, como optou-se pela plataforma GNU/Linux, o primeiro passo foi encontrar um kernel do Linux que suportasse o driver utilizado pela placa, sendo que os drivers suportariam até o kernel versão 2.6.32-51. A partir da versão do kernel é que foi determinado qual distribuição Linux a utilizar, a versão escolhida foi a versão 10.04 do Ubuntu que além de preencher os requisitos do kernel, é uma distribuição muito difundida, facilitando o acesso de informações sobre o sistema. Já com a versão do sistema determinada o próximo passo consistiu na instalação deste driver.

3.2.2 Adaptador de Rede D-LINK

O adaptador D-Link DWA-125 possui características semelhantes as apresentadas na placa da 3COM, ou seja, como suporte a redes sem fio de 2.4GHz e 5GHz e compatibilidade com as camadas físicas 802.11a/b/g/n, porém ela não adota o padrão IEEE 802.11e responsável pela tecnologia Wireless Multimídia(WMM)(D-LINK, 2013). O chip embarcado consiste em um Ralink RT3070 que dispõe de drivers para todas as plataformas, inclusive a GNU/Linux, porém em determinado dispositivo não seria necessário alteração de parâmetros no driver já que sua função é de tráfego de não tempo real, por esse motivo, seu processo de instalação não gerou nenhuma configuração específica pois os drivers na plataforma escolhida já vinham instalados.

3.2.3 Adaptador de Rede Draytec

A placa de rede da empresa Draytek modelo Vigor N65 suporta as camadas físicas 802.11a/b/g/n e destaca-se pelo suporte ao padrão IEEE 802.11e. Possui como diferencial, um utilitário presente no driver do dispositivo, que funciona na plataforma windows, a possibilidade de marcação da prioridade a ser utilizada, ilustrada na Figura 12. Esta funcionalidade auxilia no processo de marcação dos pacotes que serão enviados em cada fila (DRAYTEK, 2013). Esta placa funcionou corretamente em um sistema GNU/Linux, utilizando os drivers do chip Ralink modelo RT3070 no qual a Vigor N65 é baseada. Porém, não foi encontrada documentação necessária a fim de confirmar a localização dos parâmetros a serem alterados. Portanto, a manipulação dos parâmetros do mecanismo EDCA ficou restrita a placa da 3COM. As placas Draytec são utilizadas na estações NTR.

3.2.4 Access Point Draytec

O *Access Point* utilizado para a ligação das estações de TR foi o roteador da Draytek modelo Vigor 2130vn. Este dispositivo é compatível com a camada física definida no padrão IEEE 802.11g/n, possuindo portas gigabit, suporte a IPv6 e IPv4, duas portas USB para ligação de dispositivos de armazenamento, duas portas RJ-11 destinadas a VOIP e a implementação dos mecanismos de priorização de tráfego definidos no mecanismo EDCA do padrão IEEE 802.11.

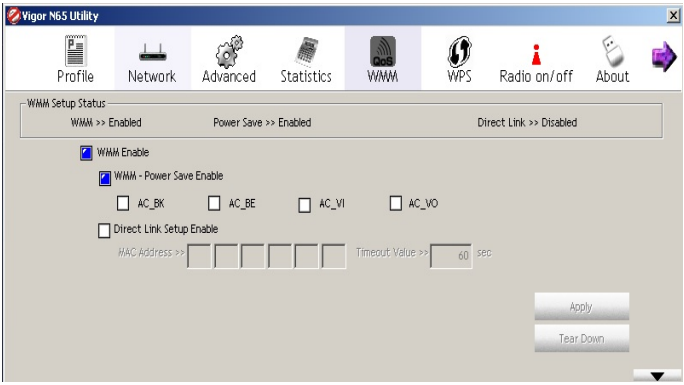

Figura 12 – Vigor Configuração WMM.

O diferencial desse modelo consiste na opção que permite alterar os parâmetros de cada fila de forma independente, ou seja, pode-se alterar os valores de AIFS, CW_{min}, CW_{max} e TXOP (CW_{min}, conforme pode ser visualizado na Figura 13 (DRAYTEK, 2013).

3.2.5 Access point CISCO

Avaliou-se também o *Access Point* da CISCO, modelo WRVS4400N, que possui as mesmas funcionalidades do anteriormente apresentado. Este AP foi utilizado para a ligação das estações NTR.

3.3 COMPONENTES DE SOFTWARE

Após a realização do estudo sobre Redes sem Fio, sobre o padrão IEEE 802.11 e as principais soluções para prover comunicação de tempo real, foi realizada uma pesquisa e testes de diversos softwares para a construção do cenário experimental. Nesta seção, apresenta-se as principais funcionalidades de cada um dos geradores de cargas sintéticas [1]. Foram encontrados diversos softwares que podem ser classificados

[1] É um software que permite a geração e reprodução de cargas de tráfego com características reais.

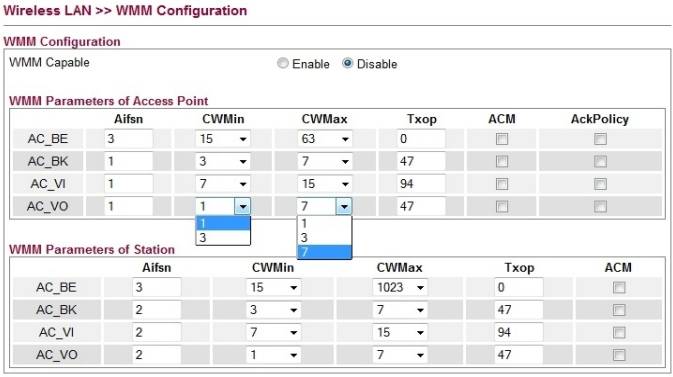

Figura 13 – Vigor alteração WMM.

nesta categoria, porém os mais adequados aos requisitos do projeto foram o MAUSEZAHN[2], o HTTPERF[3] e o o IPERF[4]. Estes geradores de tráfego possuem objetivos e características semelhantes, porém, a princípio cogitou-se de utilizar o MAUSEZAHN para a geração de tráfego periódico, enquanto o HTTPERF seria mais viável na geração de tráfego HTTP.

3.3.1 HTTPERF

É uma ferramenta que mensura o desempenho de um *web server*, fazendo isso através do protocolo HTTP. Durante a execução, ele mantém o controle de um número de medidas de desempenho que ficam resumidas na forma de estatísticas e que são impressas no final da execução do teste. A sua operação mais básica é gerar um número fixo de requisições HTTP (GET) e medir quantas respostas voltam do servidor e a que taxa as respostas chegam. Para obter resultados corretos, é necessário executar mais de uma vez o HTTPERF por máquina de cliente. O HTTPERF em uma forma mais precisa pode ser definida como um gerador de cargas sintéticas que simulam a geração de mensagens com características reais de uma rede. No caso do HTTPERF,

[2]disponível em www.perihel.at/sec/mz/mzguide.htm
[3]disponível em www.hpl.hp.com/research/linux/httperf/docs.php
[4]disponível em www.http://sourceforge.net/projects/iperf/

são geradas requisições reais a um servidor Web.

Esta ferramenta foi inicialmente escolhida, principalmente, por sua flexibilidade. Sendo possível definir diversos parâmetros para a geração de diferentes tipos e variações de cargas sintéticas. Através desses parâmetros, pode-se definir livremente a taxa de transmissão de pacotes e o número de conexões a serem feitas, o que equivale ao número de requisições HTTP que o cliente fará ao servidor.

A configuração é realizada, por exemplo, através dos parâmetros: *rate*, *num-conn* e *period*, onde o parâmetro *rate* especifica uma taxa fixa medida, em microsegundos, de transmissão em que as conexões são criadas. O parâmetro *num-conn* define a quantidade de conexões HTTP que a ferramenta fará ao servidor e, *period* diz respeito à distribuição estatística que será usada na transmissão de pacotes, ou seja de acordo com o intervalo de tempo em que cada conexão é criada. Para realização do trabalho a utilização do software não fará solicitações individuais, mas sim dividida em sessões, já que o HTTPERF será designado na utilização de estações que causam interferência as "NRT". Por esse motivo, devem ser criadas inúmeras sessões para que atinja um determinado valor de ocupação na banda. Devido a esse fator se utilizou um recurso do software capaz de gerar várias sessões simultaneamente ao invés de uma, desta forma é possível se aproximar de um tráfego real de utilização dos equipamentos. Abaixo pode-se observar um exemplo de utilização da ferramenta gerando as sessões:

```
httperf --server=192.168.1.102 --wsess=100,10,0,0.10 --rate=0 --timeout=5
```

Figura 14 – Geração de sessão no HTTPERF.

onde:
–server: Indica o servidor alvo, ou seja, qual o IP do servidor que as requisições serão enviadas, caso não for especificado, é gerado por default o nome "hostname"fazendo com que o servidor e o cliente executem na mesma máquina, por esse motivo esta opção sempre deverá ser especificada.
–wsess = n1,n2,X: Solicita a geração e medição de sessões em vez de solicitações individuais. As chamadas são emitidas da seguinte maneira: após a primeira chamada única a ser emitida e a resposta a esta primeira chamada for totalmente recebida, todas as restantes cha-

madas são emitidas simultaneamente. Utilizadas com o conjunto de parâmetros n1 e n2. Os parâmetros:

n1(–num-calls) é o número total de sessões a serem geradas.

n2(–num-conns), é o número de ligações por sessão(valor acima de 1 considera concão persistente).

X é o utilizado como temporizador (em segundos) que separa rajadas consecutivas de chamadas.

Ex: –wsess = 100,50,10 Resultaria em 100 sessões com um total de 100 chamadas cada. Sendo que cada rajada tem uma duração de 5 chamadas, um total de 20 rajadas de chamada seria gerada por sessão. O temporizador entre as rajadas de chamada seria de 10 segundos.

–rate: Especifica a taxa fixa na qual as conexões ou sessões são criadas. A taxa definida em 0 resulta em ligações ou sessões sendo gerados sequencialmente, sendo que o padrão para essa opção é 0.

–timeout: Especifica a quantidade de tempo X em segundos que o HTTPERF espera por uma reação do servidor. Este valor de tempo limite é utilizado ao estabelecer uma conexão TCP, ao enviar um pedido, quando espera por uma resposta e quando receber uma resposta. Caso alguma dessas atividades de solicitação falhar, é julgado como falha ocasionando o fechamento da conexão ou sessão. Por padrão, o valor de tempo limite é infinito.

Há diversos parâmetros no HTTPERF definidos por padrão (default). Por exemplo, o *timeout* que especifica quanto tempo em milisegundo, o cliente deve aguardar a resposta do servidor. O parâmetro –port é onde especifica a porta a ser utilizada, que por padrão se adota a porta 80. O parâmetro –send-buffer especifica o tamanho máximo do socket dos buffers utilizados para enviar solicitações HTTP, sendo que o limite é 4Kb ou seja 4096 bytes. O parâmetro –recv–buffer especifica o tamanho máximo do soquete dos buffers de recebimento usados para receber as respostas HTTP, com um limite de 16Kb ou seja 16383 bytes. Existem outras opções que podem ser utilizadas para personalizar ainda mais as requisições feitas ao servidor, sendo que nem todas as funcionalidades do programa foram utilizadas, tais como –hog que delimita as portas utilizadas, sendo que o valor default abrange portas de 1024 a 5000. Pode-se também determinar a quantidade de conexões criadas

–num-conns, a quantidade de chamadas em cada conexão –num-calls e o intervalo de tempo das conexões através do –rate. Vale salientar que existe um limitador na transmissão nas placas de rede, referente a quantidade de requisições/segundo que ela é capaz de realizar. Por esse motivo, se for utilizada uma taxa superior a suportada pelo dispositivo ocorrerá perda de pacotes.

O HTTPERF gera um relatório contendo informações após cada experimento realizado. O relatório é composto por seis grupos de estatísticas sendo elas: resultados globais (total), os resultados relacionados a conexão (Connection), os resultados relativos à emissão de requisições HTTP (Request), os resultados relativos às respostas recebidas do servidor (Reply), resultados diversos relacionados com a utilização de CPU (CPU) e rede (Net I/O)e por último um resumo dos erros encontrados (Erros).

```
root@luis-laptop:/home/luis# httperf --server=192.168.1.101 --wsess=10,10,1
httperf --client=0/1 --server=192.168.1.101 --port=80 --uri=/ --send-buffer=4096 --recv-buffer=16384 --wsess=10,10,1.000
Maximum connect burst length: 1

Total: connections 10 requests 0 replies 0 test-duration 0.001 s

Connection rate: 16206.7 conn/s (0.1 ms/conn, <=2 concurrent connections)
Connection time [ms]: min 0.0 avg 0.0 max 0.0 median 0.0 stddev 0.0
Connection time [ms]: connect 0.1
Connection length [replies/conn]: 0.000

Request rate: 0.0 req/s (0.0 ms/req)
Request size [B]: 0.0

Reply rate [replies/s]: min 0.0 avg 0.0 max 0.0 stddev 0.0 (0 samples)
Reply time [ms]: response 0.0 transfer 0.0
Reply size [B]: header 0.0 content 0.0 footer 0.0 (total 0.0)
Reply status: 1xx=0 2xx=0 3xx=0 4xx=0 5xx=0

CPU time [s]: user 0.00 system 0.00 (user 0.0% system 0.0% total 0.0%)
Net I/O: 0.0 KB/s (0.0*10^6 bps)

Errors: total 10 client-timo 0 socket-timo 0 connrefused 10 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0
```

Figura 15 – Saída HTTPERF.

onde:
Total Section (Seção total): resume quantas conexões TCP foram iniciadas pelo HTTPERF, quantos pedidos foram enviados, quantas respostas foram recebidas, e o tempo total de duração. Na Figura 15, 10

ligações foram criadas, 0 pedidos foram enviados e 0 foram recebidas respostas. A duração do teste foi de 0.001s.
Connection Section (Seção de conexão): fornece a informação relacionada com as ligações TCP geradas pela ferramenta. Especificamente, a taxa de linha de conexão que mostra que as novas ligações foram iniciadas a uma taxa de 16206.7 conexões por segundo. Esta taxa corresponde a um período de 0.1 milissegundos por conexão. O último número mostra que, no máximo, 2 ligações foram abertas em qualquer dado momento.

A primeira linha denominada "Connection time"(tempo de conexão) fornece estatísticas do tempo de vida das conexões bem sucedidas. A vida útil de uma conexão é definda com intervalo de tempo entre a abertura e o fechamento da conexão TCP, mostrando indicadores dos valores mínimos, médios e máximos do tempo de vida das conexões. O tempo de vida médio é calculado com base em um histograma com uma resolução de milissegundos e uma duração máxima de 100 segundos. A próxima estatística nesta sessão é a média de tempo para estabelecer uma conexão TCP, onde somente as conexões realizadas com sucesso são contabilizadas. No exemplo, a segunda linha "connection time"(tempo de conexão) mostra o tempo médio para estabelecer uma ligação, neste caso uma média 0,6 milissegundos. A linha final "Connection length"(Tamanho da conexão), fornece o número médio de respostas recebidas em cada ligação, podendo ter o valor maior que 1 devido a conexões persistentes.
Request Section (Pedido Seção): A linha rotulada como "Request rate"(taxa Pedido) dá a taxa em que foram emitidos os pedidos HTTP e o período que corresponde a esta taxa. Na Figura 15, a taxa de solicitações por segundo corresponde ao valor dado em milissegundos a cada pedido realizado. Já o "request size"(tamanho do pedido) fornece o tamanho médio dos pedidos HTTP em bytes.
Reply Section (Responder Seção): Realiza medições simples, como a linha rotulada "reply rate"(taxa de resposta) de várias estatísticas sobre os valores mínimos, máximos e a médios para a taxa de resposta. O número entre parênteses mostra a quantidade de amostras adquiridas. O HTTPERF recolhe uma taxa de amostragem, uma vez a cada cinco segundos para obter um desvio de padrão significativo. Sendo recomendado a execução de testes longos o suficiente para que, pelo menos trinta amostras sejam obtidas. A linha denominada "reply time"(tempo de resposta) fornece informações sobre o tempo que o servidor demorou para responder e quanto tempo para receber a resposta. A próxima

linha, chamada "reply size"(tamanho da resposta) possui estatísticas sobre o tamanho médio das respostas. A linha apresenta a duração média de cabeçalhos de resposta, o conteúdo e rodapés.
Miscellaneous Section (Seção Diversa): Esta seção resume a utilização da CPU na máquina do cliente. A linha "CPU time"(tempo de CPU) mostra o tempo em segundos gastos pelo modo "user"(usuário)e pelo modo "system"(sistema) sendo que a porcentagem de cada um dos modos corresponde ao valor total de utilização do CPU. Altas taxas de utilização são esperados pelo HTTPERF já que é um software que utiliza muito processamento, sendo que valor estiver muito abaixo do máximo de uso provavelmente é um sinal de existência de processos concorrentes que interferiram com o ensaio. A linha denominada "Net I/O"dá a vazão média da rede em kilobytes por segundo. No exemplo, um uso médio de rede é calculada com base no número de bytes enviados e recebidos nas conexões TCP, não levando em consideração os cabeçalhos de rede ou retransmissões TCP que possam ter ocorrido.
Errors Section (Seção de Erros): Contém estatísticas sobre os erros que foram encontrados durante um teste. No exemplo, as duas linhas mostram que havia um total de 10 erros e que todos os dez eram erros devido ao servidor recusar aceitar uma conexão "connrefused".

3.3.2 MAUSEZAHN

Mausezahn é um gerador de tráfego que permite o envio de tráfego sintético na rede, podendo ser utilizado para sobrecarregar redes, realizar testes de firewall e IDS (*Intrusion detection system*)[5], para verificação de bugs no software de rede ou em dispositivos. Este software pode ser utilizado tanto em linhas de comando através de uma sintaxe ou também através de scripts *bash* realizando assim combinações de testes. Disponível somente para plataforma Linux. (HAAS, 2010)

O Mausezahn possui dois modos de utilização. O primeiro deles, chamado de modo direto, permite a criação de quadros e pacotes diretamente na linha de comando do Linux. Este modo se refere ao modo legado, sendo que não possui suporte a alguns recursos como o suporte a múltiplas *threads*. O segundo modo de utilização é chamado de modo interativo, baseado em um sistema de pacotes MOPS onde maioria dos novos recursos do software estão disponíveis dentro desse

[5]IDS - Consiste em uma ferramenta para detecção de intrusão que possui a capacidade de detectar e até mesmo bloquear um ataque no momento em que ele está ocorrendo, podendo também informar dados sobre o ataque e a sua origem, além do tempo em que foi realizado, auxiliando na busca dos responsáveis.

subsistema. Fornece uma interface de comando interativa, multitarefa, possibilitado a criação de quaisquer valores de transmissão.

O Masezahn e o HTTPERF, porém, não são os únicos softwares geradores de cargas sintéticas, temos por exemplo o Kute que foi desenvolvido para ser um gerador de tráfego de máximo desempenho para redes gigabyte, porém não possui bom desempenho em termos de pacotes por segundo. Entre os diferencias que o MGEN ou multi-gerador possui dos demais geradores de carga, é de que, além de ser um software open source, ou seja, ele roda em várias plataformas, muito além do Linux, tais como MacOS e Windows. Há também o TGS que permite gerar um tráfego real na rede, simulando usuários e serviços, O IPerf pode ser utilizado no modo cliente/servidor analisando o tráfego da rede tanto unidirecionalmente como bidirecionalmente, existe também o NetPerf, programa criado pela IND *Networking Performance Team* da Hewlett-Packard (HP) que gera cargas sintéticas para realizar a verificação de vazão da rede e latência ponto-a-ponto.

O Mausezahn foi avaliado neste projeto por possuir recursos interessantes para a realização do trabalho, tais como recursos específicos para geração e análise de tráfegos de tempo real do protocolo RTP (*Real Time Protocol*) que permite a definição de tempos entre as transmissões, período e especificação do tamanho do pacote de acordo com o teste utilizado. Por padrão, quando gerado pacotes no protocolo RTP, é definido para os pacotes enviados um delay de 20 ms e tamanho de 160 bytes, porém, é possível mudar tais valores. Além disso, pode-se medir o *jitter* e a taxa de perda de pacotes, utilizando a ferramenta em duas estações diferentes. Uma estação em modo de transmissão e outra estação em modo de escuta, ou seja, recebendo os pacotes.

3.3.3 IPERF

IPerf é uma ferramenta do tipo cliente-servidor desenvolvida pelo *National Laboratory for Applied Network Research* (NLANR). Este software permite testar e medir a taxa de transferência da rede e pode ser utilizado também em conjunto com outros softwares para, por exemplo, testar se um controle de banda está realmente sendo executado. O software possui versões disponíveis para diversas plataformas e é simples de ser utilizado, bastando escolher entre um dos modos disponíveis, ou seja, pode ser usado no modo cliente, modo servidor ou modo bidirecional.

Para a sua utilização básica, o modo cliente-servidor é utilizado

dois computadores sendo que o computador cliente utiliza o comando iperf -c, para poder enviar diferentes tipos de tráfego podendo ele ser TCP ou UDP através de uma porta pré-configurada. Neste modo, o usuário pode configurar diversos parâmetros, entre eles o tempo de transmissão em segundos, o formato de saída dos dados enviados além de alterar a opção da camada de transporte no que se refere ao tamanho do buffer e do pacote a ser enviado. Utilizando o IPerf no modo servidor, captura-se os pacotes enviados pelo cliente, realizando a medição de alguns parâmetros que possibilitam medir o desempenho da rede. Os parâmetros do modo servidor consistem na escolha do tipo de tráfego a ser analisado, podendo escolher TCP ou UDP e a porta a ser verificada, as demais opções são relacionadas ao formato de saída dos resultados. O comando para utilização do IPerf em modo servidor se resume ao iperf –s, fazendo com que aguarde as conexões do cliente. Alguns dos dados mensurados pela ferramenta são o tamanho da banda utilizada (*Bandwidth*), o *jitter* que é uma medida da variação do atraso entre os pacotes de dados e o número total de pacotes transmitidos e pacotes perdidos. Os resultados dos testes são informados pela Figura 16. No final de cada teste realizado pelo IPerf, é realizado uma média do total dos valores recebidos durante a execução do programa localizados no final de cada resultado.

Para a realização dos experimentos foram utilizadas configurações diferentes do IPerf no modo cliente. As estações de tempo real RT de acordo com os requisitos apresentados, foi definido que o tamanho do pacote UDP seria de 45 bytes enviando a uma taxa de 348Kbits por segundo na porta 30000 durante um tempo de 20 segundos utilizando o comando de acordo com a Figura 17.

Já as estações de não tempo real NRT foram configuradas de acordo com a porcentagem de carga na rede e a quantidade de estações utilizadas. Foi utilizado quatro estações possuindo o tamanho do pacote em 1500 bytes cada, e valores de carga variando de 10%, 40% e 70% em relação ao padrão 802.11g no qual é de 54mbps de acordo com a Tabela 4. Para especificar a quantidade de carga gerada foi utilizado os comandos ilustrados na Tabela 5

Para a captura dos dados das estações *sniffers* foram utilizadas duas estações conectadas nos roteadores via conexão Ethernet que tiveram o IPerf configurado em modo servidor com o tipo de tráfego UDP na porta 30000 conforme ilustrado na Figura 18, somente variando o endereço já que eram duas redes distintas.

Além das opções já citadas para inicialização do IPerf, outros comandos foram ilustrados em figuras porém veremos alguns detalhes

```
iperf -s -u -P 0 -i 1 -p 30000 -f k
------------------------------------------------------------
Server listening on UDP port 30000
Receiving 1470 byte datagrams
UDP buffer size:   122 KByte (default)
------------------------------------------------------------
[  3] local 192.168.1.13 port 30000 connected with 192.168.1.14 port 50086
[  4] local 192.168.1.13 port 30000 connected with 192.168.1.12 port 36786
[  5] local 192.168.1.13 port 30000 connected with 192.168.1.10 port 59265
[  6] local 192.168.1.13 port 30000 connected with 192.168.1.11 port 41775
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[  3]  0.0- 1.0 sec  29.9 KBytes    245 Kbits/sec  3.188 ms    0/  680 (0%)
[  4]  0.0- 1.0 sec  36.5 KBytes    299 Kbits/sec  1.534 ms    0/  830 (0%)
[  5]  0.0- 1.0 sec  30.5 KBytes    250 Kbits/sec  1.030 ms    0/  694 (0%)
[  3]  1.0- 2.0 sec  13.4 KBytes    109 Kbits/sec  1.549 ms  710/ 1014 (70%)
                                          .
                                          .
                                          .
                                          .
[  4] 19.0-20.0 sec  26.4 KBytes    216 Kbits/sec  1.360 ms    0/  600 (0%)
[  4]  0.0-20.2 sec   585 KBytes    237 Kbits/sec  18.600 ms    0/13305 (0%)
[  4]  0.0-20.2 sec  1 datagrams received out-of-order
[  5] 19.0-20.0 sec  35.8 KBytes    293 Kbits/sec  0.604 ms    0/  815 (0%)
[  5]  0.0-20.2 sec   474 KBytes    192 Kbits/sec  0.705 ms    0/10784 (0%)
[  5]  0.0-20.2 sec  1 datagrams received out-of-order
[  6] 19.0-20.0 sec  14.0 KBytes    114 Kbits/sec  1.014 ms    0/  318 (0%)
[  6]  0.0-20.3 sec   199 KBytes  80.3 Kbits/sec  15.664 ms    0/ 4519 (0%)
[  6]  0.0-20.3 sec  1 datagrams received out-of-order

Iperf thread stopped [CAUSE=Stream Closed]
```

Figura 16 – LOG do IPERF.

```
iperf -c 192.168.1.13 -u -P 1 -i 1 -p 30000 -l 45.0B -f k -b 348.0k -t 20
```

Figura 17 – Configuração IPerf RT.

do significado de algumas siglas que podem ser utilizados de acordo com a necessidade.

-u =Tipo de tráfego = UDP

-P =Somente em modo cliente, gera tráfego simulando vários clientes em paralelo

-i =Define o intervalo de tempo de registro do relatório de saída

-p =Especifica a porta a ser utilizada

-l =Tamanho do buffer(default 8 KB)

Tabela 4 – Valor de banda por máquina.

Quantidade de Carga	Valor em relação a 54Mbps	Valor de cada estação
10%	5,4	1,35Mbps
40%	21,6	5,4Mbps
70%	37,8	9,45Mbps

Tabela 5 – Comandos para alteração de carga no IPerf.

Carga	Comando
10%	iperf -c 192.168.1.100 -u -P -i -p 30000 -I 1500.0B -f k -b 1.35M -t 30 -T
40%	iperf -c 192.168.1.100 -u -P -i -p 30000 -I 1500.0B -f k -b 5.4M -t 30 -T
70%	iperf -c 192.168.1.100 -u -P -i -p 30000 -I 1500.0B -f k -b 9.45M -t 30 -T

-f =Define a unidade do relatório, que pode ser: Kbits, Mbits, KBytes, MBytes

-b =Define a banda a ser utilizada em bps (apenas para UDP)

-t =Define o tempo de duração dos testes, sendo o padrão 10 segundos

3.3.4 IPTABLES

IPtables é uma ferramenta que controla o módulo do Linux chamado *netfilter* no qual é responsável por prover um conjunto de funcionalidades. O *netfilter* é prove funções de firewall e NAT através da ferramenta do ambiente de usuário IPtables, no qual permite o estabelecimento de regras gerenciadas e controladas pelo módulo *netfilter*.

Possui funções de criação de regras de firewall e serviços de NAT. Possibilita a edição da tabela de filtragem de pacotes, capaz de criar e de analisar o cabeçalho (header) e tomar decisões sobre os destinos destes pacotes ditando qual o tipo de tráfego que é enviado e recebido pelo sistema.

O funcionamento do firewall funciona como sendo uma barreira entre duas redes, através do qual só passa tráfego autorizado sendo este realizado de acordo com a política de segurança estabelecida. Quando um pacote chega em um firewall, ele inspeciona, no caso o IPtables utiliza uma tabela de regras onde é verificado se o pacote se aplica a alguma regra caso não seja aplicado é utilizado uma política default que con-

```
iperf -s -u -P 0 -i 1 -p 30000 -f k
```

Figura 18 – Configuração IPerf servidor.

siste na regra caso algum pacote não se encaixe em nenhuma das regras estabelecidas. O IPtables é constituído basicamente de três tipos de regras chamadas também de "CHAINS"podendo ser INPUT, OUTPUT e FORWARD no qual direcionam pacotes. O primeiro é chamado de INPUT que se destina os pacotes a máquina firewall, sendo os pacotes de chegada. O segundo é o OUTPUT que são os pacotes originados da máquina firewall e por último temos o FORWARD que são os pacotes com destino e origem separados pela máquina firewall ou seja são os pacotes que serão redirecionados. O fluxo do funcionamento do IPtables é o seguinte: quando um pacote chega, o KERNEL analisa qual será o destino do pacote (routing), caso ele for destinado a própria máquina o pacote irá para o CHAIN INPUT, caso ele passar pelo CHAN INPUT então a máquina efetivamente recebe o pacote. Caso o destino não seja destinado a máquina com o firewall que roda o IPtables e a mesma possuir um serviço de roteamento habilitado o pacote vai para o CHAN FORWARD redirecionamento e o pacote é destinado a outra interface, onde será encaminhado para o CHAIN OUTPUT que caso aceite o pacote, ele será encaminhado, caso contrário, será descartado como ilustra a Figura 19.

O *netfilter* atua analisando o cabeçalho dos pacotes decidindo o destino do pacote, podendo aceitar ou descartar o pacote. É uma ferramenta já integrada na maioria das distribuições atuais. Ele interage diretamente com o kernel determinando quais são os pacotes a serem filtrados. Há diversas opções que podem ser utilizadas pelos IPtables, além das regras de principais INPUT, OUTPUT e FORWARD, há operações para poder gerenciar chains através da sintaxe: iptables -t [tabela] [opções] [alvo] possuindo três opções de acordo com a Tabela 6. Já as opções ilustradas na Figura 7, são inúmeras sendo exemplificado alguns exemplos das possibilidades e por último o alvo que serão os endereços ou interfaces a serem utilizadas.

Das três tabelas possíveis pode-se descrever cada uma de suas funcionalidades. a primeira tabela chamada "filter"é uma tabela default caso não seja colocado nenhuma opção de regra, ela permite a

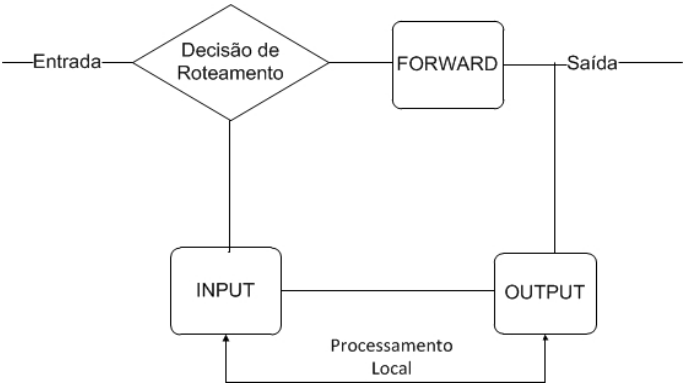

Figura 19 – Fluxo IPtables.

Tabela 6 – Tabela IPtables.

Tipos de Tabelas	
Filter	O que entra e sai pela interface de rede
Mangle	Características do pacote
NAT	Tabela de roteamento

filtragem de pacotes destinado a própria máquina "INPUT"e de pacotes gerados localmente OUTPUT e de FORWARD para qualquer pacote que atravessa o firewall vindo de uma máquina e direcionada a uma outra. Ao ser utilizado a opção "NAT"(network address translation) é quando existe a necessidade de criar uma nova conexão para passagem de dados de uma rede privada para a internet ou seja quando o destino é diferente dos endereços de IPs conhecidos, utilizando para tal fim as chains PREROUTING que altera os pacotes recebidos antes do roteamento, o OUTPUT que alterar localmente pacotes antes do roteamento e o POSTROUTING para mudar o endereço de origem das conexões. Por último, temos a tabela "mangle"no qual é utilizada para especificar ações diferenciadas no tráfego a ser atravessado nos chains. Na tabela há dois chains o PREROUTING e o OUTPUT que possibilitam a alteração do conteúdo dos pacotes. Esta opção foi o principal fator determinante da escolha da ferramenta para realizar a marcação dos pacotes que são enviados das estações de tempo real TR

Tabela 7 – Opções IPtables.

Parâmetro	Descrição
-N	Criar nova chain
-P	Mudar a política de uma chain build-in
-A	Adicionar uma nova regra dentro de uma chain
-I	Inserir uma nova regra em alguma posição chain
-D	Apagar uma regra em alguma posição da chain
-j	Qual ação será tomada
-p	Escolha do tipo de protocolo utilizado

para o *sniffer* pois era necessário nos requisitos do cenário realizar esta alteração. O comando utilizado para a marcação dos pacotes gerados pelas estações de tempo real 20. O comando faz com que o IPtables marque todos os pacotes UDP que são enviados pela máquina para a porta 30000 tenham o conteúdo de um campo do pacote IP chamado DSCP(ponto de código de serviços diferenciados) no qual permite que que diversos níveis de serviços sejam atribuídos ao tráfego de rede. Para isso, cada pacote de rede é marcado com um código DSCP de acordo com a Tabela 8 (MICROSOFT, 2013), o valor para marcação dos pacotes para prioridade de voz possui o código 0x38, para realizar a alteração da marcação dos pacotes para a classe de VOZ é usado o comando do IPtables "–set-dscp-class cs7". Para confirmar a alteração feita pelo IPtables, verificou-se juntamente com a ferramenta Wireshark se efetivamente, o pacote estava sendo marcado de forma correta através da checagem do cabeçalho IP.

Tabela 8 – Tabela DSCP.

Tipos de tráfego DSCP	Campos DSCP IP	Níveis prioridade
DSCPBestEffort	$0x00$	BE(prioridade esforço em massa)
DSCPBackground	$0x08$	BK(prioridade em massa)
DSCPExcellentEffort	$0x18$	BE(prioridade esforço em massa)
DSCPVideo	$0x28$	VI(prioridade de vídeo)
DSCPAudio	$0x38$	VO(prioridade de voz)
DSCPControl	$0x38$	VO(prioridade de voz)

A Ferramenta, além de poder especificar as operações a serem feitas dentro de cada regra CHAIN possui mais alguns tipos de filtros como a especificação do endereço de origem e destino através dos comandos (–source) e (–destination) além de também definir qual porta

```
iptables -t mangle -A OUTPUT -p udp --dport 30000 -j DSCP --set-dscp-class cs7&
```

Figura 20 – Comando de Marcação dos Pacotes.

será utilizada através do comando *-sport* porta de origem e *-dport* porta de destino. Pode-se também especificar o tipo de protocolo utilizado pelo comando -p podendo ser do tipo TCP, UDP ou ICMP e caso o comando for acrescentado de '!' a regra é invertida e caso seja utilizada como '-p! UDP', a regra especificará todos os pacotes que não são UDP. Existem muitos outros filtros que podem ser utilizados porém não será descrito todos, funções extras e mais detalhadas estão todas reunidas na página do projeto (NETFILTER, 2013).

3.3.5 WIRESHARK

Além das principais ferramentas utilizadas nos quais foram as que geram e capturam tráfego, outras ferramentas auxiliares foram utilizadas para verificar se os softwares de geração de tráfego estavam funcionando adequadamente, sendo que o escolhido para captura e análise do tráfego foi o Wireshark no qual também possui código livre, sendo então um analisador de pacotes. É utilizado principalmente para solução de problemas na rede, análises, desenvolvimento de protocolo de comunicações e para educação. Possui características que dificilmente são vistas em outros softwares pois como é baseado em código aberto open-source, permite que especialistas de rede ao redor do mundo propor e adicionar melhorias. É executado em todos as plataformas populares de computação, Linux e Windows. Dentre as suas funções que podem ser realizadas, a que será desenvolvida será a parte informativa ou seja, não a de resolução de problemas da rede em si, mas de captura de pacotes e análise da sua estrutura. Tais como: quadro MAC, datagrama, segmento de pacotes TCP e outro conteúdo e transmissão. Todo o estudo da ferramenta para sua utilização e interpretação dos dados capturados tinham como objetivo o de comprovar a quantidades de dados enviados na rede, estavam de acordo com o estipulado em softwares de geração de tráfico, para que a quantidade de dados transmitida seja o mais próximo de aplicações reais. Alguns

dados importantes para o trabalho estão dispostos na Figura 21, tais como o tamanho do cabeçalho, tamanho total do pacote e ip de origem dos dados capturados já que os dados vinham de múltiplos computadores com o mesmo destino em comum.

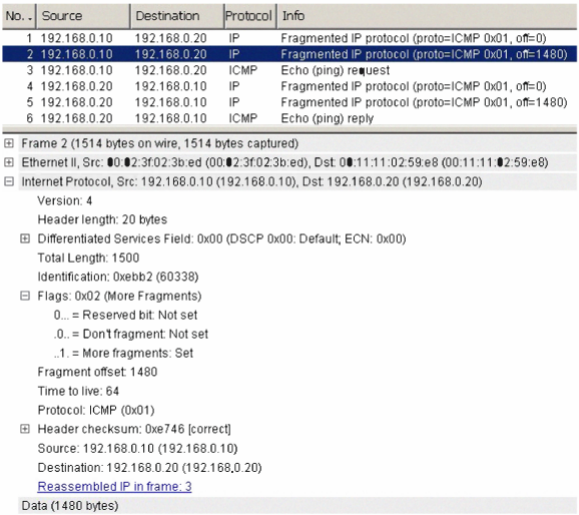

Figura 21 – Captura dos pacotes ARP

Em resumo, a plataforma utilizada para analisar o desempenho das redes sem fio baseadas no padrão IEEE 802.11e, consiste em estações base para analisar o tráfego na rede. Sendo a primeira responsável pelo tráfego de tempo real e a segunda pelo tráfego de não tempo real, ambas as estações dispõem de uma máquina ligada a elas com softwares que mensuram os dados que chegam no qual são responsáveis pelos resultados de todo o cenário. As estações base recebem os dados de estações geradores de tráfego. Estas estações também estão divididas em tipos de dados da mesma maneira em que as estações bases. Cada tipo de estação possui um padrão, formado por uma placa de rede conectada a estação de tráfego correspondente, um computador com um sistema operacional e um software para geração de tráfego. As estações de tempo real se diferenciam das de não tempo real por possuirem placas de rede que implementam o padrão IEEE 802.11e e softwares de geração de tráfego de tempo real.

3.4 PLACAS DE REDE

Um controlador de interface de rede, também conhecido como adaptador de rede, é um componente de hardware utilizado para conectar fisicamente um computador para obter acesso as comunicações da rede (POSEY, 2006).

Paralelamente à seleção das ferramentas, foi feita uma pesquisa relativa a seleção de placas de redes adequadas para a realização dos experimentos. Constatou-se a disponibilidade comercial de placas que implementam o padrão IEEE 802.11e, conhecido também como Wi-Fi Multimídia "WMM"Wireless Multimídia Extensions. Eles provem serviços de QoS nas redes IEEE 802.11, priorizando o tráfego de acordo com as categorias: voz, vídeo, best effort e background. Sendo assim, foram adquiridas três placas da 3COM modelo 3CRUSBN275, uma NETGEAR modelo WN111V2 e três D-Link modelo DWA-125. Sendo que somente as D-link, não possuem serviços de QoS, ou seja que não implementam o padrão IEE 802.11e. As placas com Wi-Fi Multimídia além de serem diferenciadas pelo padrão, foram escolhidas também por possuir seu controlador o chip Atheros Ar9170 drivers open-source que pudessem ser manipulados. Esses drivers nos permitiram alterar alguns valores. Os valores em questão foram os de Cwmax e Cwmin propostos pelo mecanismo EDCA, sendo proposto alguns valores para que pudessem verificar quais poderiam melhorar o desempenho da rede em relação ao valor default. Os testes foram realizados primeiramente com o valor default implementado pelo padrão que é de 3 de aCWmin e 15 de aCWmax, primeiro teste com essa especificação foi feita sem cargas de interferência ou seja somente com as três estações de tempo real mandando.

3.5 CONCLUSÃO

Este capítulo descreve aspectos da construção do cenário experimental, detalhando quais os requisitos do cenário e quais componentes são necessários para a avaliação de uma parametrização do mecanismo EDCA. A plataforma possui estações de TR e NTR juntamente com os *sniffers*. Também são detalhados os modelos de adaptadores de rede e *access-points* escolhidos de acordo com as suas especificações, que deveriam ter como requisito principal a implementação do IEEE 802.11e. Descreve-se também os softwares analisados e utilizados para a realização dos experimentos, tais como geradores *sniffers*, descrevendo as suas principais características e funcionalidades.

4 EXPERIMENTOS

Foram realizados experimentos para avaliar a utilização do mecanismo EDCA através dos equipamentos utilizados no cenário realizado. Composto por estações de TR e NTR em uma mesma rede gerando tráfego, tendo como objetivo verificar o provimento de qualidade de serviço (QoS). Os experimentos tiveram como base, uma análise realizada por simulação e publicada anteriormente em Moraes et al. (2010).

4.1 DESCRIÇÃO DOS CENÁRIOS

Conforme especificado no capítulo 3 e ilustrado na Figura 11, a plataforma experimental é composta de estações de tempo real (TR) e não tempo real (NTR), *Access Points* TR e NTR e por *Sniffers* TR e NTR, estes com a função de coletar os dados de cada experimento.

Neste capítulo realiza-se um teste da plataforma experimental com o objetivo de avaliar o comportamento da categoria de mais alta prioridade (voz) do mecanismo EDCA, quando esta é utilizada para transmitir dados de TR[1]. O ambiente de comunicação é compartilhado com fonte de tráfego externas. Os dados periódicos de TR são utilizados com a intenção de modelar as mensagens enviadas de sensores para os controladores e também dos controladores para os atuadores, em uma planta industrial. Poderiam também modular os dados transmitidos em uma aplicação do VoIP (voz sobre IP).

Os cenários foram construídos considerando uma topologia de rede infraestruturada, onde múltiplas estações de TR e NTR operam na mesma frequência de transmissão. As estações encontram-se na mesma área de cobertura e não há obstáculos entre elas, portanto, não ocorrem os problemas das estações escondidas e/ou oculta. As estações TR e NTR operam de acordo com as características físicas do padrão IEEE 802.11g, onde aCWmin=15 e o aCWmax=1023. É importante ressaltar que a taxa máxima de transmissão destas redes é de 54 Mbps.

Todos os experimentos são repetidos 5 vezes e os resultados apresentados consistem na obtenção da média e desvio padrão destes experimentos. As métricas analisadas em cada cenário incluem: a variação do jitter, percentual de pacotes perdidos e a taxa média de transmissão obtida. O valor do jitter é uma variação estatística do atraso na entrega

[1]Neste trabalho considera-se dados de TR, pacotes com pequenas quantidades de dados transmitidos periodicamente.

de dados em uma rede. A quantidade de pacotes perdidos mostra o percentual de pacotes não recebido durante o intervalo de tempo e a taxa de transmissão representa o valor médio alcançado em KBits/segundo.

Cada tipo de estação e cada *sniffer* possui alguns software que vai depender da sua utilização. Nos experimentos realizados as estações de TR e NTR utilizam o software IPerf para gerar o tráfego, assim como os servidores TR e NTR (*sniffers*). Nos capítulos anteriores foram descritos os softwares Mausezahn e HTTPERF, pois, inicialmente, pretendia-se utilizá-los. Mas, devido a algumas dificuldades nas análise dos resultados gerados pelos softwares, optou-se por utilizar somente o software IPerf para a geração de dados TR e NTR, principalmente, devido a facilidade de uso e de obtenção dos resultados.

4.1.1 Cenário de Testes

No cenário considera-se que 4 estações TR operam na mesma área de cobertura de 4 estações NTR, conforme Figura 11. As estações TR enviam mensagens com destino ao servidor TR (*Sniffer* TR), enquanto que as estações NTR enviam mensagens para o servidor NTR (*Sniffer* NTR). As estações TR enviam mensagens com 45 bytes de dados a cada 2ms, portanto, há a geração de 500 mensagens por segundo. As estações NTR enviam mensagens com tamanho de 1500 bytes, sendo que a quantidade de mensagens por segundo varia de acordo com a carga de interferência desejada. Optou-se por avaliar cenários onde as estações NTR ocupam 0%, 10%, 40% e 70% da capacidade do meio físico, ou seja, num primeiro experimento não há interferência, e nos experimentos seguintes as estações NTR enviam em torno de 5,4 Mbps, 21,6 Mbps e 47,8 Mbps. Estas cargas são divididas igualmente entre as 4 estações NTR.

Conforme descrito nos capítulos anteriores, o mecanismo EDCA provê qualidade de serviço (QoS) com base em três mecanismos independentes: o espaçamento entre frames (AIFS), as oportunidades de transmissões (TXOP) e o tamanho das janelas de contenção (CW). Uma das principais conclusões dos experimentos realizados em Moraes et al. (2010), através de simulação, foi que o principal mecanismo que provê melhoria do nível de QoS é a variação de CW. Portanto, neste trabalho optou-se por avaliar a transmissão de mensagens pelas estações de TR para os valores de CW apresentados na Tabela 9. Foram testados somente estes valores porque o *access point* utilizado define somente este conjunto para a fila de voz.

Tabela 9 – Parâmetros avaliados para CW.

aCWmin	aCWmax
1	3
1	7
3	7

Todos os outros valores utilizados durante os testes (e.g.AIFS e TXOP) foram os padrões.

4.1.2 Resultados

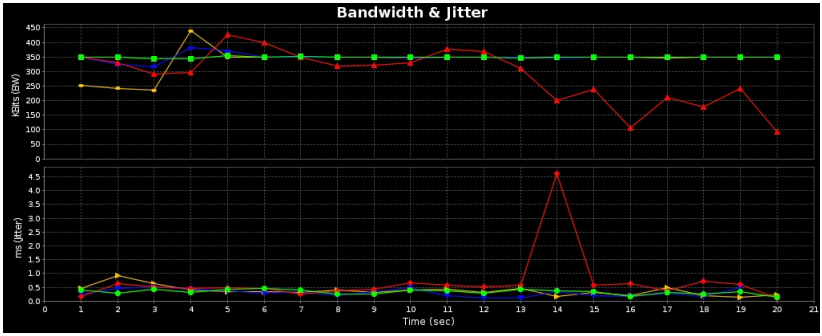

Figura 22 – aCWmin=1 aCWmax=3 - sem carga externa

O primeiro experimento consistiu na avaliação do mecanismo EDCA para a transmissão de mensagens de TR para aCWmin=1 e aCWmax=3. As Figuras 22 e 23 ilustram os resultados de um dos 5 experimentos quando a carga externa imposta pelas estações NTR é de 0% e 40%, respectivamente. Na Figura 22, que ilustra os resultados para as 4 estações de TR (cada linha representa um estação), observa-se uma variação da taxa de transmissão das estações. É importante mencionar que esta variação acontece até mesmo no cenário em que não há interferências externas, porém, as estações atingem a taxa de transmissão esperada 348 KBits/s. Por outro lado, quando a carga externa na rede é de 40%, as estações atingem uma taxa de transmissão

que não ultrapassa 100 KBits/s, Figura 23.

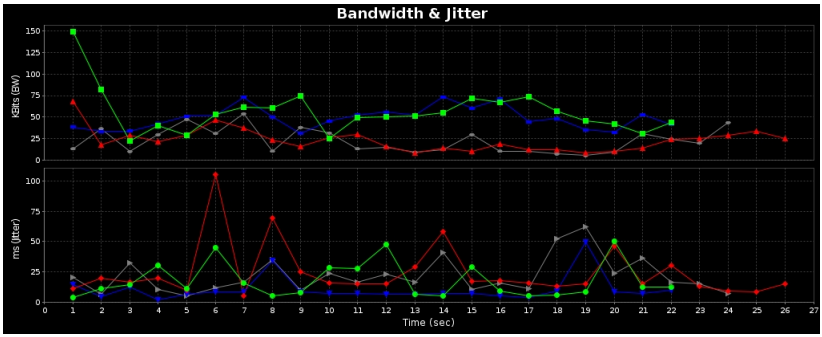

Figura 23 – aCWmin=1 aCWmax=3 - 40% de carga externa

A Tabela 10 ilustra os resultados das médias dos 5 experimentos, bem como o desvio padrão obtido. Observa-se nestes resultados que os parâmetros para aCWmin=1 e aCWmax=3 não são adequados, pois, há um aumento significativo na taxa de perda de pacotes, *jitter* e uma redução na taxa de transmissão.

Tabela 10 – Resultados para aCwmin=1 e aCWmax3.

Métrica/ Carga	Taxa de perdas	Desvio Taxa Perda	Bandwidth (kbits/s)	Desvio Padrão Bandwidth	Jitter (ms)	Desvio Padrão Jitter(ms)
Sem Carga	3,29	7,153	333,05	24,84	0,99	3,39
10%	7,39	14,43	261,85	62,87	4,37	7,20
40%	32,80	36,35	45,84	18,15	89,52	42,35
70%	57,20	33,14	26,96	8,754	137,46	49,86

O segundo experimento utilizou os parâmetros: aCWmin=1 e aCWmax=7. A Figura 24 demonstra que em um ambiente sem carga, a taxa de transmissão fica em torno dos 348 KBits/seg. Ao adicionar uma carga externa com 40% de ocupação da rede, verifica-se que nenhuma das estações consegue manter uma taxa de mais de 50 KBits/seg, variando muito o tempo do *jitter* médio em todas as estações (Figura 25). Como conclusão do segundo experimento, observa-se na Tabela 11 que os parâmetros aCWmin=1 e aCWmax=7 melhoram os resultados através de um aumento na taxa de transmissão, bem como uma redução das taxas de perdas do *jitter* em relação ao experimento utilizando aCWmin=1 e aCWmax=3.

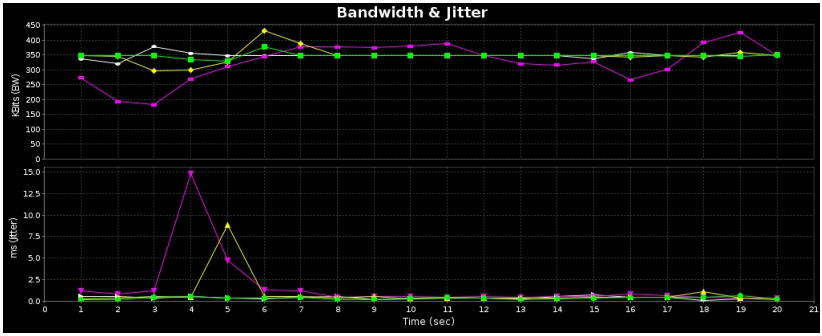

Figura 24 – aCWmin=1 aCWmax=7 - sem carga externa

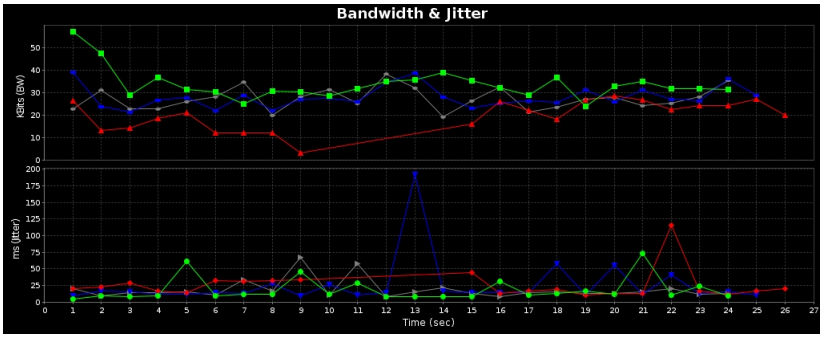

Figura 25 – aCWmin=1 aCWmax=7 - 40% de carga externa

Tabela 11 – Resultados para aCWmin=1 e aCWmax7.

Métrica/ Carga	Taxa de perdas	Desvio Taxa Perda	Bandwidth (kbits/s)	Desvio Padrão Bandwidth	Jitter (ms)	Desvio Padrão Jitter(ms)
Sem Carga	7,06	9,79	319,6	31,66	0,357	0,157
10%	7,133	11,80	256,55	54,57	6,10	7,496
40%	12,77	26,29	58,4	15,97	72,59	23,49
70%	55,38	29,45	27,93	8,965	137,32	49,54

No terceiro experimento foi utilizado os seguinte parâmetros: aCWmin=3 e aCWmax=7 e através destes valores foram realizados testes com e sem cargas externas. A primeira análise realizada quando não há estações NRT, ilustrado na Figura 26 onde observa-se que não ocorre uma diferença significativa em relação aos experimentos anteriores atingindo também o valor estipulado de 348 KBits/seg. Porém, quando a carga imposta pelas estações NTR é de 40%, há uma grande variação da taxa de transmissão, variando de 8 KBits/seg até 30 KBits/seg, além do *jitter* que obteve variações menores das relatadas nos outros experimentos. Com a carga externa de 40% em comparação com a mesma quantidade de carga realizada com o experimento de aCWmin=1 e aCWmax=7, nota-se uma redução da taxa de perdas e do *jitter* e a melhora da taxa de transmissão de acordo com a Tabela 12.

A Tabela 12 ilustra os resultados médios e o desvio padrão dos 5 experimentos.

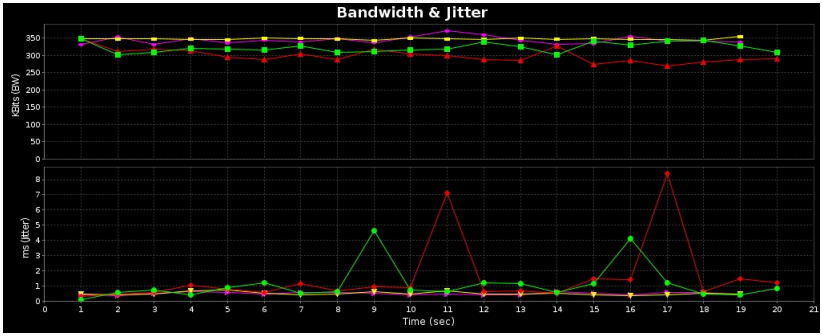

Figura 26 – aCWmin=3 aCWmax=7 - sem carga externa

Tabela 12 – Resultados para aCwmin=3 e aCWmax7.

Métrica/ Carga	Taxa de perdas	Desvio Taxa Perda	Bandwidth (kbits/s)	Desvio Padrão Bandwidth	Jitter (ms)	Desvio Padrão Jitter(ms)
Sem Carga	2,752	6,432	322,6	30,78	3,62	6,149
10%	9,395	15,92	246,2	42,74	7,012	7,368
40%	8,227	14,049	56,25	7,961	74,526	36,41
70%	41,375	34,95	33,43	9,084	107,08	38,26

Os resultados observados nos experimentos, demonstram que existe uma diferença significativa nas taxas de perdas, quando há carga

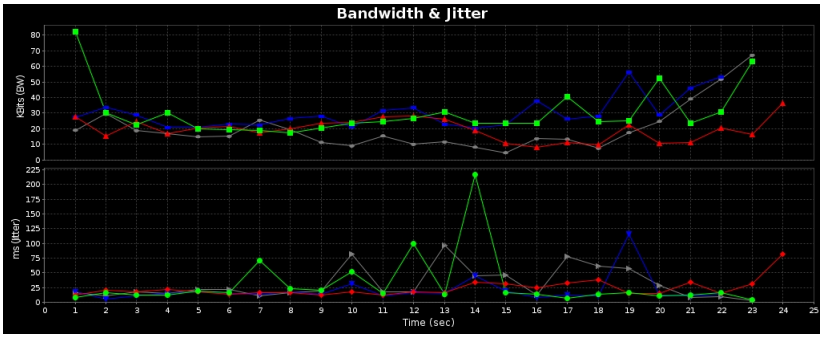

Figura 27 – aCWmin=3 aCWmax=7 - 40% de cargas externas

sendo imposta pelas estações NTR. As Figuras 28-30 apresentam os resultados obtidos para os valores de CWmin e CWmax avaliados.

Na Figura 28 verifica-se que ao utilizar cargas na rede de 40% e 70% a taxa de perdas tem uma redução considerável, principalmente, com a utilização dos parâmetros aCWmin=3 e aCWmax=7. A redução na taxa de perdas também pode ser relacionada pela diminuição da quantidade de dados enviados, variando de acordo com interferência na rede. De acordo com a Figura 29 quanto maior o tráfego das estação de NRT menor será a taxa de dados que as estações TR conseguem enviar.

Outra análise é a referente ao *jitter*, que normalmente aumenta com as interferências impostas pelas estações NTR, porém pode-se notar que quando existe um alto grau de utilização da rede, os parâmetros aCWmin=3 e aCWmax=7 tem os *jitters* médios menores. Esta tendência ocorre quando os valores para as janelas de contenção CW são maiores, possibilitando que as estações tenham uma probabilidade menor de escolher os mesmos valores de backoff, reduzindo o número de colisões.

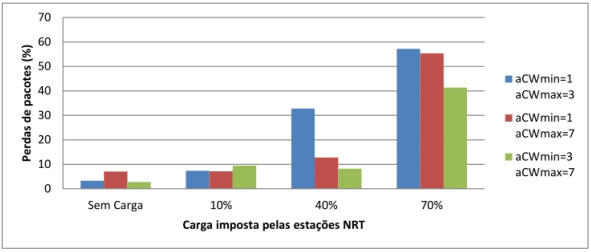

Figura 28 – Taxa de perdas

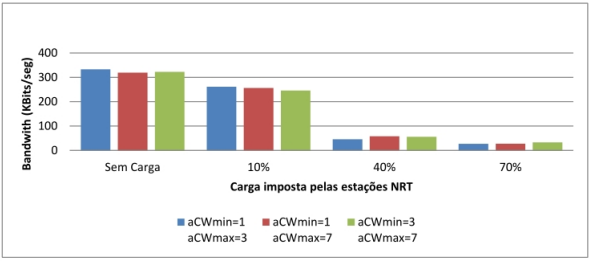

Figura 29 – Taxa de transmissão (Bandwidth)

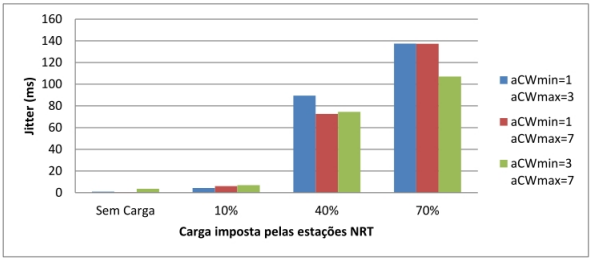

Figura 30 – Jitter da taxa de transmissão

4.2 ANÁLISE DOS EXPERIMENTOS

Este capítulo avaliou a plataforma experimental construída, descrevendo os cenários e a forma que estes foram implementados. Em seguida detalhou-se os resultados obtidos, que avaliaram diferentes valores para os parâmetros aCWmin e aCWmax, quando as estações TR operam em ambientes livres de interferências externas e, quando operam no mesmo ambiente com um conjunto de estações NTR enviando tráfego na rede com cargas de 10%, 40% e 70%. No primeiro experimento pode ser observado que os valores não foram os adequados, devido as altas taxas de perda de pacotes, de transmissão e grande variação do *jitter*. No segundo experimento com a alteração somente do CWmax, percebe-se uma melhora significativa dos resultados, principalmente na taxa de perdas. No terceiro experimento, onde utilizou-se os valores de aCWmin=3 e aCWmax=7, obteve-se os melhores resultados. Porém, constatou-se que os parâmetros testados não são adequados para prover QoS pelas redes IEEE 802.11e, quando estas operam em ambientes abertos.

4.3 DIFICULDADES ENCONTRADAS

Algumas dificuldades foram encontradas durante a realização do TCC, a primeira dificuldade foi a respeito da localização dos parâmetros, já que era necessário encontrar dentro do driver qual a localização do arquivo que efetivamente permitia a mudança nos parâmetros do mecanismo EDCA. A compilação dos drivers também foi complicada devido a forma de utilização dos comandos para que as alterações sejam efetivamente realizadas e a quantidade de mudanças realizadas a cada experimento realizado, sendo necessário realizar a compilação do driver diversas vezes. Neste processo foi o que permitiu um maior conhecimento do sistema GNU/Linux, facilitando o manuseio com esse sistema. Outro aspecto que acabou dificultando a realização dos experimentos, devido as opções pré estabelecidas no AP Draytek, que possui poucos valores a serem alterados na fila de voz, limitando a quantidade de experimentos realizados. Outro aspecto que dificultou a realização dos experimentos foi manter a estabilidade das transmissões, em cada experimento realizado as estações eram monitoradas constantemente para verificar se estavam efetivamente transmitindo dados. Constantemente estações paravam de transmitir por razões específicas do sistema operacional, sendo necessário a repetição do experimento.

5 CONCLUSÃO

A plataforma experimental construída e utilizada para a realização dos experimentos é composta por um conjunto de hardware e software, onde estações de TR e NTR compartilham o mesmo meio de comunicação. Os dispositivos utilizados na plataforma implementam o mecanismo EDCA conhecido comercialmente como WMM, as estações que foram utilizadas como TR tiveram os valores dos parâmetros da janela de contenção(CW) alterados no driver das placas de rede.

No decorrer deste trabalho efetuaram-se testes com os softwares HTTPERF, MAUSEZAHN e IPtables, sendo que somente este último foi utilizado na realização dos experimentos. Os demais softwares não foram utilizados, principalmente, devido a dificuldades de interpretação dos resultados gerados por estas ferramentas. Para substituir as duas ferramentas foi utilizado o software IPerf, que foi capaz de gerar os tráfegos de TR e NTR e atuar como o servidor dos dois tipos de tráfego. Utilizando somente um software de geração, o funcionamento e a obtenção dos dados gerados pelos experimentos foram facilitados. Outros softwares foram utilizados nos experimentos tais como o IPtables, que realizava a marcação dos pacotes enviados pelas estações de TR e a utilização do Wireshark como ferramenta auxiliar, fazendo a verificação dos dados enviados e recebidos bem como se a marcação dos pacotes estava funcionando corretamente.

Ao estudar o padrão IEEE 802.11e, primeiramente foi verificado quais as principais características relacionadas ao padrão, as diferenças entre os diferentes padrões, normas e as aplicações atuais que abrangem as redes sem fio. Com o foco no padrão IEEE 802.11e, verificou-se que este protocolo provê níveis diferentes de QoS às aplicações que utilizam voz e vídeo através da inclusão de uma função chamada HCF que escalona oportunidades de transmissão TXOP, que pode ter o seu valor definido pelo EDCA. É no mecanismo EDCA onde é alterado as janelas de contenção CW nos drivers para a realização dos experimentos.

Os experimentos foram realizados com o objetivo de avaliar o comportamento da categoria de voz, por ser a de maior prioridade do mecanismo EDCA. Utilizando para esse fim um ambiente de comunicação compartilhado, com estações transmitindo tráfego TR e estações NTR gerando quantidades de carga variáveis de 10%, 40% e 70%. As estações de TR por sua vez geravam a mesma quantidade de carga na rede, porém com valores de CW distintos. Os valores testados foram limitados devido as opções disponíveis no AP.

Os experimentos realizados demonstram uma melhoria do nível de QoS providos às estações de TR, à medida que os valores da janela de contenção (aCWmin e aCWmax) são alterados para valores mais altos. Porém, mesmo com a alteração dos parâmetros realizados, pode ser observado que as estações NTR obtinham resultados superiores às estações TR mesmo estas possuindo mecanismos para a melhoria da transmissão. Esta fato ocorre devido às múltiplas colisões que ocorrem nas transmissões de dados das estações de TR, que tem uma alta probabilidade deescolha do mesmo tempo de *backoff* para os valores avaliados. Por outro lado, as estações NTR operam com valores (*default*) para aCWmin=15 e aCWmax=1023. Portanto a principal conclusão deste trabalho é que os parâmetros definidos para o mecanismo EDCA não são adequados para estações que operam em um ambiente de comunicação aberto.

REFERÊNCIAS

5TI. *Adaptador USB 3Com 3CRUSBN275*. http://www.5ti.com.br/p64016-adaptador-usb-3com-3crusbn275.html, Agosto 2013.

BARTOLOMEU, P.; FERREIRA, J.; FONSECA, J. Enforcing flexibility in real-time wireless communications: a bandjacking enabled protocol. p. 1730–1733, 2009.

CASETTI, C. et al. Notes on the Inefficiency of 802.11e HCCA. In: *In Proceedings of the 62nd IEEE Vehicular Technology Conference*. EUA: [s.n.], 2005. v. 4, p. 2513–2517. <http://www.di.unito.it/ garetto/conferences/notes.pdf>.

CHENG, R. G. et al. Ripple: a wireless token-passing protocol for multi-hop wireless mesh networks. *IEEE Communications Letters*, v. 10, n. 2, p. 123–125, Fevereiro 2006. ISSN 1089-7798.

CHRISTENSEN, K. J. Performance evaluation of the binary logarithmic arbitration method (BLAM). *Conference on Local Computer Networks (LCN)*, Minneapolis, MN, USA, p. 396 – 403, 1996. ISSN 0742-1303.

D-LINK. *DWA-125 Wireless 150 USB Adapter*. http://www.dlink.com.br/produtos-detalhes/items/dwa-125.html, Outubro 2013.

DENG, J.; CHANG, R.-S. A priority scheme for IEEE 802.11 DCF access method. *IEICE Trans. Commun. (Japan)*, E82-B, n. 1, p. 96 – 102, 1999. ISSN 0916-8516.

DRAYTEK. *Vigor N65 and Vigor 2130n*. [S.l.], Outubro 2013. <http://www.draytek.com>.

ERGEN, M. et al. WTRP - Wireless Token Ring Protocol. *IEEE Transactions on Vehicular Technology*, v. 53, n. 6, p. 1863–1881, Novembro 2004. ISSN 0018-9545.

FRIEDRICH, G. R.; ALIMENTI, O. R.; REGGIANI, G. H. WRTMAC: A MAC Proposal for 802.11 Networks in Factory Automation. 2010.

HAAS, H. *Mausezahn User's Guide*. herbert AT perihel DOT at http://www.perihel.at/sec/mz: [s.n.], 02 2010. <www.perihel.at/sec/mz/mzguide.html>.

HWANG, G.-H.; CHO, D.-H. New access scheme for VoIP packets in IEEE 802.11e wireless LANs. *IEEE Communications Letters*, v. 9, n. 7, p. 667 – 669, 2005. ISSN 1089-7798.

IEEE. Ieee standard for local and metropolitan area networks: Media access control (mac) bridges. *IEEE Std 802.1D-2004 (Revision of IEEE Std 802.1D-1998)*, p. 1–277, 2004.

IEEE. Ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, p. 1–2793, 2012.

IEEE COMPUTER SOCIETY. IEEE Standard for Information Technology - "Logical Link Control". 1998.

IEEE COMPUTER SOCIETY. *IEEE Standard for Information Technology Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*. 2000.

IEEE COMPUTER SOCIETY. IEEE Standard for Information Technology - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements. 2005.

IEEE COMPUTER SOCIETY. *IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. 2012.

KOPETZ, K. The time-triggered model of computation. In: *Proceedings of the 19th IEEE Real-Time Systems Symposium (RTSS)*. [S.l.: s.n.], 1998. p. 168–177.

KUROSE, J. F. *Redes de Computadores e a Internet*. 5°. ed. [S.l.]: Pearson, 2010.

LEE, S. et al. NDIS-based virtual polling algorithm for IEEE 802.11b for guaranteeing the real-time requirements. *Computer Standards & Interfaces*, Elsevier, v. 29, n. 3, p. 316–324, 2007.

LO, S.; LEE, G.; CHEN, W. An efficient multipolling mechanism for IEEE 802.11 wireless LANs. *IEEE Transactions on Computers*, Published by the IEEE Computer Society, p. 764–778, 2003.

Lobello, L.; KACZYNSKI, G. A.; MIRABELLA, O. Improving the real-time behavior of ethernet networks using traffic smoothing. *IEEE Transactions on Industrial Informatics*, v. 1, n. 3, p. 151–161, 2005. ISSN 1551-3203.

MICROSOFT, W. fi S. *Visão geral sobre o ponto de código de serviços diferenciados (DSCP)*. 10 2013. <http://msdn.microsoft.com/pt-br/library/aa916767.aspx>.

MIORANDI, D.; VITTURI, S. Analysis of master-slave protocols for real-time-industrial communications over IEEE 802.11 WLANs. In: *Proceedings of the 2nd IEEE International Conference on Industrial Informatics (INDIN)*. Alemanha: [s.n.], 2004. p. 143–148.

MORAES, R. *Supporting Real-Time Communication in CSMA-Based Networks: The VTP-CSMA Virtual Token Passing Approach*. Tese (Doutorado) — Universidade do Porto, 2007.

MORAES, R. et al. Assessment of the IEEE 802.11e EDCA Protocol Limitations when Dealing with Real-Time Communication. *EURASIP Journal on Wireless Communications and Networking*, 2010.

MORAES, R.; VASQUES, F.; PORTUGAL, P. Survey of real-time communication in csma-based networks. *Network Protocols and Algorithms*, v. 2, p. 158–183, 2010.

MORAES, R. et al. VTP-CSMA: A Virtual Token Passing Approach for Real-Time Communication in IEEE 802.11 Wireless Networks. *IEEE Transactions on Industrial Informatics*, v. 3, n. 3, p. 215–224, Agosto 2007. ISSN 1551-3203.

MORAES, R. et al. A forcing collision resolution approach able to prioritize traffic in csma-based networks. *Computer Communications*, v. 33, n. 1, p. 54–64, 2010. ISSN 0140-3664. <http://www.sciencedirect.com/science/article/B6TYP-4WXC24P-1/2/e53c9aede41cf7a723caafaf0f01db77>.

NETFILTER. *The netfilter.org project*. out. 2013. <http://netfilter.org/>.

POSEY, B. M. *"Networking Basics: Part 1 - Networking Hardware"*. Outubro 2006.

SOBRINHO, J.; KRISHNAKUMAR, A. Quality-of-service in ad hoc carrier sense multiple access wireless networks. *IEEE J. Sel. Areas Commun*, v. 17, n. 8, p. 1353 – 68, 1999. ISSN 0733-8716.

SOBRINHO, J. L.; KRISHNAKUMAR, A. S. EQuB - Ethernet Quality-of-Service using Black Bursts. In: *Proceedings of the 23rd Annual Conference on Local Computer Networks (LCN)*. EUA: [s.n.], 1998. p. 286–296. ISSN 0742-1303.

SON, J. et al. An effective polling scheme for IEEE 802.11e. *IEICE Transactions on Communications*, IEICE, E88.B, n. 12, p. 4690–4693, 2005. ISSN 0916-8516. <http://dx.doi.org/10.1093/ietcom/e88-b.12.4690>.

VILLALóN, J. et al. B-EDCA: A QoS mechanism for multimedia communications over heterogeneous 802.11/802.11e WLANs. *Computer Communications*, Butterworth-Heinemann, Newton, MA, USA, v. 31, n. 17, p. 3905–3921, 2008. ISSN 0140-3664.

WETHERALL, D. J.; TANENBAUM, A. *Redes de Computadores*. [S.l.]: PEARSON EDUCATION - BR, 2011.

WU, Y.-J.; CHIU, J.-H.; SHEU, T.-L. A modified EDCA with dynamic contention control for real-time traffic in multi-hop ad hoc networks. *Journal of Information Science and Engineering*, Nankang, Taipei, 115, Taiwan, v. 24, n. 4, p. 1065 – 1079, 2008. ISSN 1016-2364.