

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

GABRIEL BOLOGNANI CASELLA

**IMPLEMENTAÇÃO E AVALIAÇÃO DE DESEMPENHO DO
PROTOCOLO PBP-DCF PARA WLANS COM LARGURA
ADAPTATIVA DE CANAIS**

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA
2014

GABRIEL BOLOGNANI CASELLA

**IMPLEMENTAÇÃO E AVALIAÇÃO DE DESEMPENHO DO
PROTOCOLO PBP-DCF PARA WLANS COM LARGURA
ADAPTATIVA DE CANAIS**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação, do Departamento Acadêmico de Informática da Universidade Tecnológica Federal do Paraná.

Orientador: MSc. Saulo Jorge Beltrão de Queiroz

Coorientador: MSc. Marcos Talau

PONTA GROSSA
2014



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Ponta Grossa

Diretoria de Graduação e Educação Profissional



TERMO DE APROVAÇÃO

Implementação e avaliação de desempenho do protocolo PbP-DCF para WLANs
com largura adaptativa de canais

por

GABRIEL BOLOGNANI CASELLA

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 14 de Novembro de 2014 como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Msc Saulo Jorge Beltrão de Queiroz
Prof. Orientador

Prof. Msc Marcos Talau
Membro titular

Prof. Dr. Richard Duarte Ribeiro
Membro titular

Prof. Dr. Ionildo José Sanches
Responsável pelos Trabalhos
de Conclusão de Curso

Prof. Dr. Gleifer Vaz Alves
Coordenador do Curso
UTFPR - Campus Ponta Grossa

AGRADECIMENTOS

Agradeço ao meu orientador que soube aturar minha falta de paciência, dúvidas repetidas, falta de memória e outros variados acontecimentos, dedicando tempo e esforço imprescindíveis para a realização desse trabalho.

Agradeço também a minha família, a qual me suportou e ajudou durante a construção desse trabalho e especialmente por toda a jornada acadêmica. Sem isso, o caminho teria sido muito mais árduo.

Por fim, a todos os amigos e colegas ligados direta ou indiretamente a esse trabalho, que me ajudaram de alguma forma a chegar ao fim dessa etapa, meu sincero muito obrigado.

RESUMO

CASELLA, Gabriel Bolognani. **Implementação e avaliação de desempenho do protocolo PbP-DCF para WLANs com largura adaptativa de canais**. 2014. 39f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2014.

No paradigma de acesso ao meio adotado pelo padrão IEEE 802.11ac para redes locais sem fio as estações competem para acessar o mais largo dentre os canais disponíveis na rede. Pesquisas recentes indicam que, sob tal paradigma, o desempenho das redes pode ser prejudicado devido a fatores como maior requisito de razão sinal-ruído (para suportar uma comunicação com sucesso) e maior exposição à interferência como resultado do alargamento da banda. Visando contornar esses problemas, um protocolo alternativo foi proposto na literatura, a saber, o *Piece-by-Piece - Distributed Coordination Function (PbP-DCF)*, em que uma estação realiza múltiplas transmissões estreitas no lugar de uma transmissão larga por disputa. Resultados analíticos preliminares mostraram pelo menos um cenário em que tal protocolo dobra a capacidade do padrão IEEE 802.11. Neste trabalho é realizado a implementação de um *script* do protocolo PbP-DCF no simulador de rede ns-3, além de uma análise comparativa entre ambos os protocolos considerando canais com ruído. Os resultados confirmam a melhoria de desempenho do PbP-DCF em relação ao padrão sob tal cenário.

Palavras-chave: ns-3. PbP-DCF. CSMA/CA. IEEE 802.11. WLAN.

ABSTRACT

CASELLA, Gabriel Bolognani. **Implementation and performance analysis of PbP-DCF protocol to WLANs with Variable Channel Width**. 2014. 39f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Federal University of Technology - Paraná. Ponta Grossa, 2014.

In the IEEE 802.11ac standard for local wireless networks, the medium access paradigm employs the best effort to get the widest available channel. However, recent researches show that this paradigm requires higher signal to noise ratio to succeed. To surpass this problem an alternative protocol known as PBP-DCF was proposed in the literature. Instead of accessing a wider channel at once, the PBP-DCF proves analytically that multiple narrow width transmissions can outperform a single wide one in ideal channels. In this work a ns-3 script is implemented to simulate the PBP-DCF behavior and compare under a noisy environment how both protocols perform. The results show that the PBP-DCF can outperform the IEEE 802.11 CSMA/CA saturation throughput even in noisy scenarios.

Keywords: ns-3. PbP-DCF. CSMA/CA. IEEE 802.11. WLAN.

LISTA DE ILUSTRAÇÕES

Figura 1	– Exemplo de uma transmissão PbP-DCF com base em uma linha do tempo	16
Figura 2	– Fluxograma de transmissão de um pacote do protocolo PbP-DCF ..	17
Figura 3	– Algumas possibilidades para as larguras de canais do padrão Institute of Electrical and Electronics Engineers (IEEE) 802.11a	18
Figura 4	– Vazão da rede pela quantidade de nós conectados utilizando o padrão IEEE 802.11a	18
Figura 5	– Comunicação entre nós e dispositivos no cenário da simulação	22
Figura 6	– Comparação da média entre os protocolos por diferentes <i>path loss</i> .	28
Figura 7	– Resultado da simulação sem ruído no canal <i>wireless</i>	29
Figura 8	– Resultado da simulação com <i>path loss</i> definido em 8.80	29
Figura 9	– Resultado da simulação com <i>path loss</i> definido em 8.84	30
Figura 10	– Resultado da simulação com <i>path loss</i> definido em 8.85	30
Figura 11	– Resultado da simulação com <i>path loss</i> definido em 8.86	31
Figura 12	– Resultado da simulação com <i>path loss</i> definido em 8.87	31

LISTA DE TABELAS

Tabela 1	–	Comparação entre a variação dos níveis de sinais, protocolos pela largura de canal e velocidade de transferências em um mesmo cenário	14
Tabela 2	–	Descrição dos parâmetros usados na simulação analítica	16
Tabela 3	–	Resultados da simulação entre os vários ruídos aplicados.....	28

LISTA DE ABREVIATURAS E SIGLAS

WLAN	<i>Wireless LAN</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
CSMA/CA	<i>Carrier Sense Multiple Access/Collision Avoidance</i>
AaO	<i>All-at-Once</i>
STA	<i>Stations</i>
SNR	<i>Signal to Noise Ratio</i>
PbP-DCF	<i>Piece-by-Piece - Distributed Coordination Function</i>
MAC	<i>Medium Access Control</i>
CNB	<i>Channel Negotiation Bit</i>
RTS	<i>Request To Send</i>
CTS	<i>Clear To Send</i>
ECDF	<i>Empirical Cumulative Distribution Function</i>

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Objetivo	11
1.1.1	Objetivos Específicos	11
1.2	Justificativa	11
1.3	Organização do Trabalho	11
2	O PARADIGMA DE ACESSO AO MEIO <i>ALL-AT-ONCE</i>	12
2.1	Acesso a canais: largura fixa × adaptativa	12
2.2	Requisitos para transmissão em canais largos	13
3	O PARADIGMA DE ACESSO AO MEIO <i>PIECE-BY-PIECE</i>	15
3.1	O protocolo MAC PbP-DCF	15
3.1.1	Resultados preliminares	16
4	METODOLOGIA	19
5	IMPLEMENTAÇÃO DO PROTOCOLO PBP-DCF	20
5.1	A estrutura dos computadores	20
5.2	A configuração dos canais	21
5.3	A função de vazão	23
5.4	Modelo de propagação	24
6	RESULTADOS	26
7	CONCLUSÃO	32
	REFERÊNCIAS	33
	APÊNDICES	35
	APÊNDICE A – <i>SCRIPT</i> NS-3 UTILIZADO PARA SIMULAÇÃO	36

1 INTRODUÇÃO

Redes locais sem fio (*Wireless LAN*, WLANs) representam uma importante tecnologia de acesso a variados serviços oferecidos pela internet *e.g.*, *e-commerce*, redes sociais e comunicação multimídia em alta definição. Tendo em vista atender a crescente demanda por maiores velocidades dos usuários móveis, o padrão emergente *Institute of Electrical and Electronics Engineers* (acrônimo IEEE) 802.11ac (IEEE Standard for Information technology, 2013) tem por objetivo alcançar taxas de transmissão da ordem de Gigabits por segundo. A fim de alcançar tal objetivo, o padrão emprega o protocolo *Carrier Sense Multiple Access/Collision Avoidance* (CSMA/CA) sob o paradigma *All-at-Once* (AaO) para acesso ao meio.

No paradigma de acesso ao meio AaO, todas as estações (do inglês *Stations* e acrônimo STA) de uma WLAN empregam o maior esforço para acessar o mais largo dentre os canais disponíveis na rede. Na teoria, esse paradigma proporciona uma vazão nominal alta por transmissão, porém na prática, seu desempenho é limitado pela dificuldade inerente à alocação de um maior número de ondas portadoras e pela elevada exigência de razão sinal-ruído (do inglês *Signal to Noise Ratio (SNR)*) imposta por canais largos (QUEIROZ, 2013). Por sua vez, isso implica na elevação do custo do *hardware* em relação a versões anteriores do padrão IEEE 802.11 devido ao aumento do número de antenas.

Em razão das limitações supracitadas, Queiroz (2013) propõe o protocolo Piece-by-Piece - Distributed Coordination Function (PbP-DCF) que emprega o uso de múltiplas transmissões de canais estreitos no lugar de uma única transmissão em um canal largo após uma dada estação vencer uma disputa. O trabalho também propõe um modelo analítico para comparar o desempenho do PbP-DCF com o protocolo CSMA/CA adotado no padrão IEEE 802.11. Apesar dos resultados reportados terem sido promissores em favor do protocolo PbP-DCF, somente um cenário típico de WLAN foi modelado, *i.e.*, nós estáticos com tráfego saturado e canal sem ruído.

A metodologia analítica cumpre papel fundamental para prova de conceito e motiva a necessidade de avaliar os protocolos sob condições diversas. Contudo, a obtenção de novos resultados comparativos sob variados parâmetros de tráfego e níveis de ruído implica em um grau de complexidade usualmente proibitivo na abordagem analítica. Devido a esse motivo, Jain (1991) argumenta que uma metodologia de avaliação baseada em modelo analítico incorpora menos detalhes e requer mais suposições do que uma simulação, e por isso torna-se mais distante da realidade.

1.1 OBJETIVO

O objetivo deste trabalho consiste em simular o protocolo PbP-DCF e comparar seu desempenho com o CSMA/CA sob diferentes níveis de ruído no canal.

1.1.1 Objetivos Específicos

Primeiramente é necessário identificar um simulador de rede que atende aos requisitos básicos do protocolo IEEE 802.11, nesse caso, largura variável de canal e protocolo de colisão CSMA/CA. Posteriormente, baseando-se no padrão IEEE 802.11, necessita-se simular o protocolo PbP-DCF para, por fim, executar uma avaliação comparativa do PbP-DCF com o padrão CSMA/CA sob diferentes níveis de ruído do canal.

1.2 JUSTIFICATIVA

Como citado anteriormente, a avaliação analítica além de não permitir facilmente a reprodução de vários cenários tende a tornar-se mais simplista devido ao alto custo de uma modelagem detalhista à realidade. Por isso, prover um cenário nos quais os parâmetros de simulação podem ser ajustados de maneira simples, *i.e.*, diferentes níveis de ruído do canal, torna-se necessário para analisar o comportamento do PbP-DCF sob diversas situações. Diante disso, consegue-se resultados mais próximos a realidade, garantindo uma comparação mais fiel ao atual padrão IEEE 802.11.

1.3 ORGANIZAÇÃO DO TRABALHO

O restante desse documento está organizado de forma que no capítulo 2 cita-se como o paradigma padrão de acesso ao meio *All-at-Once* funciona. No capítulo 3 mostra-se como o novo paradigma de acesso ao meio PbP-DCF trabalha, em contraste com o *All-at-Once*. No capítulo 4 é citado a forma de condução desse trabalho para no capítulo 5 ser possível mostrar como deu-se a criação do *script* do novo protocolo no ns-3. Os resultados comparativos entre o CSMA/CA e o PbP-DCF denotam-se no capítulo 6. Por fim, no capítulo 7 é relatada a conclusão da análise comparativa e possíveis trabalhos futuros.

2 O PARADIGMA DE ACESSO AO MEIO *ALL-AT-ONCE*

Como consequência da maior demanda por velocidade, devido a necessidade de suportar as aplicações existentes e a explosão de serviços com grande demanda de dados (*e.g.*, *cloud computing*, vídeo sob-demanda, vídeo conferência), a comunidade de pesquisa da área de WLANs sentiu necessidade de desenvolver novos protocolos de Controle de Acesso ao Meio (do inglês *Medium Access Control (MAC)*) com capacidade de alocar espectro sob demanda. Na seção seguinte aborda-se o paradigma de acesso utilizado pelo padrão IEEE 802.11. Em seguida, no capítulo 3 descreve-se um protocolo do estado da arte cujo paradigma de acesso constitui uma alternativa ao padrão IEEE 802.11.

2.1 ACESSO A CANAIS: LARGURA FIXA × ADAPTATIVA

Dentre os variados protocolos MAC disponíveis para redes locais sem fio, a utilização de canal fixo para acesso ao meio tem sido uma característica predominante (IEEE Standard for Information technology, 2013). O primeiro trabalho relacionado à variação de largura nos canais nas redes WLAN foi proposto por Chandra *et al.* (2008), o qual propôs um algoritmo para adaptar a largura dos canais dinamicamente de acordo com a carga demandada. Isso garantiu um desempenho no mínimo 65% superior em relação a abordagem baseadas em largura fixa nos cenários avaliados. Com base nesses resultados, os autores argumentam que WLANs deveriam adaptar a largura dos canais dinamicamente conforme uma dada política de desempenho (*e.g.*, maximização de vazão).

A transmissão por canais de larguras variadas é utilizado atualmente no padrão IEEE 802.11ac (IEEE Standard for Information technology, 2013) com canais de 20 MHz, 40 MHz e 80 MHz. No referido padrão, o método de acesso para seleção da largura do canal baseia-se em quatro canais ortogonais de 20 MHz. Para início da transmissão, um canal primário P_c necessita ser configurado, conseqüentemente atribuindo os outros canais como secundários, sendo este, denominado de canal primário por conter a disputa inicial ao CSMA/CA pelo meio.

O CSMA/CA é uma expansão do CSMA, e garante um acesso equitativo ao meio de transmissão, ao mesmo tempo que tenta evitar casos de colisão (motivo da sigla CA) em ambientes com meio compartilhado, como o *wireless*. De maneira simplificada, um nó, ao tentar iniciar uma transmissão, sensoreia o meio de modo a averiguar se esse está disponível ou não. Caso um meio ocioso seja detectado, a transmissão ocorre, porém caso o meio esteja ocupado, um número aleatório n é sorteado e seu decremento só é realizado quando o meio se encontra livre para transmissão nova-

mente (processo conhecido como *Backoff*). Após n tentativas de transmissão o nó finalmente transmite, mas caso ocorra uma colisão todo o processo equivalente ao meio estar ocupado é repetido.

Diante disso, no IEEE 802.11ac, caso o acesso ao meio ocorra antes do término do número aleatório n e o canal ainda esteja livre por um tempo adicional de *PCF Interframe Space* (do sigla inglesa, PIFS), uma transmissão de 40 MHz e posteriormente para uma próxima transmissão com o mesmo cenário anterior uma largura de 80 MHz é utilizada. No entanto, caso o tempo de disputa esgote-se, o canal utiliza-se de uma largura de banda mais estreita e adia a transmissão com 80 MHz (IEEE Standard for Information technology, 2013).

2.2 REQUISITOS PARA TRANSMISSÃO EM CANAIS LARGOS

O paradigma *All-at-Once* visa maximizar a vazão da rede através do uso de altas taxas de transmissão. Essa abordagem apresenta alguns problemas que podem comprometer os ganhos de desempenho. De fato, quanto maior a largura de um canal, maior o número de dados que podem ser transmitidos numa mesma unidade de transmissão (PARK, 2011; CHANDRA *et al.*, 2008), resultando em símbolos de transmissão com maior quantidade de bits. Conseqüentemente, é necessário um aumento na razão sinal-ruído a fim de decodificar com sucesso o símbolo transmitido, o que acarreta algumas situações serem favoráveis a utilização de canais estreitos.

A análise da tabela 1 revela que para cada nível de entrada de sinal mínimo pode-se utilizar uma velocidade de transferência máxima correspondente. Adicionalmente, essa velocidade também é alterada de acordo com a largura do canal, devido ao fato de um canal mais estreito possuir menos interferência e conseguir aumentar a sua potência de transmissão em 3dBm para cada 10 MHz reduzidos em largura (QUEIROZ, 2013).

Nessas condições, com uma força de sinal mínima de -81dBm, pode-se utilizar de uma velocidade de até 12 Mbps, em um canal de 20 MHz. No entanto, com o mesmo nível de entrada de sinal para um canal com largura reduzida de 10 MHz (ocasionando um nível maior de SNR) pode-se obter uma velocidade *por canal* de 9 Mbps.

Tendo em vista as limitações supracitadas e as vantagens relacionadas a transmissões em canais estreitos, Queiroz (2013) propõe um novo protocolo, onde a transmissão é baseada em canais estreitos, conseqüentemente aumentando a facilidade de alocação do meio para realizar múltiplas transmissões estreitas ao invés de uma única transmissão larga. O protocolo é referenciado como PbP-DCF sendo tratado em maiores detalhes na seção 3.

Tabela 1 – Comparação entre a variação dos níveis de sinais, protocolos pela largura de canal e velocidade de transferências em um mesmo cenário

Nível de Entrada de Sinal Mínimo (dBm)	Tipo de Acesso ao Meio e Largura do Canal(MHz)	Velocidade de Transferência Máxima do Canal (Mbps)
-82	Padrão de 1×20	1×6
	Não padrão de 2×10	2×6
-82	Padrão de 1×20	1×9
	Não padrão de 2×10	2×6
-81	Padrão de 1×20	1×12
	Não padrão de 2×10	2×9
-77	Padrão de 1×20	1×18
	Não padrão de 2×10	2×12
-74	Padrão de 1×20	1×24
	Não padrão de 2×10	2×12
-70	Padrão de 1×20	1×36
	Não padrão de 2×10	2×18
-66	Padrão de 1×20	1×48
	Não padrão de 2×10	2×27
-65	Padrão de 1×20	1×54
	Não padrão de 2×10	2×27

Fonte: (QUEIROZ, 2013).

3 O PARADIGMA DE ACESSO AO MEIO *PIECE-BY-PIECE*

A alocação de dinâmica de canais resolve o problema da subutilização de largura de banda à clientes com pouca necessidade de transmissão, ao mesmo tempo que aumenta a capacidade de transmissão aos clientes com maior necessidade de vazão (MOSCIBRODA *et al.*, 2008). Porém, em razão do grande número de equipamentos WLANs atualmente disponíveis e a própria natureza de redes *wireless* a ocorrência de interferências, interrupções de sinal e perda de pacotes, as alocações de canais largos, como proposto no paradigma AaO, são mais difíceis de serem obtidas (GUMMADI *et al.*, 2007).

Devido a isso, o paradigma de acesso ao meio PbP proposto por Queiroz (2013) nunca acessa toda a largura de banda disponível unicamente, mesmo que ela esteja ociosa. O modelo de acesso empregado diferencia do usual AaO por sempre competir pela transmissão em um canal primário, e posterior acesso aos próximos canais secundários adjacentes disponíveis.

3.1 O PROTOCOLO MAC PBP-DCF

Baseado na filosofia do PbP, o protocolo PbP-DCF funciona de maneira na qual o canal mais largo disponível - B_w - é reorganizado em N_c canais estreitos ortogonais. Essa reorganização é realizada de maneira que os canais mais estreitos sejam sempre menores que a largura total do canal, neste caso $B_n < B_w$, e que a quantidade N_c de canais estreitos seja o maior número inteiro da divisão entre a largura do canal mais estreito pela banda total disponível, *i.e.*, $N_c = \lfloor B_w/B_n \rfloor$. Essa combinação de possibilidades de largura de canais pode ser vista na figura 3 para o padrão IEEE 802.11a.

Além disso, tendo por definição de canal primário P_c o canal onde *sempre* ocorrerá a disputa por transmissão ao meio seguindo o padrão CSMA/CA, P_c é sempre requisitado para início das transmissões e assim que a disputa é ganha, os canais secundários sequenciais $c \in [1, N_c - 1]$, se tornam disponíveis para transmissão sem concorrência de *backoff* (QUEIROZ, 2013).

O protocolo baseia-se na premissa de que toda transmissão que necessite do canal secundário altere para um o valor do bit de requisição de canal (do inglês *Channel Negotiation Bit (CNB)*). Ou seja, toda transmissão que ainda exista pacotes enfileirados para serem transmitidos irá requisitar o acesso ao canal secundário através do CNB.

Após a transmissão no canal primária ser concluída, se o próximo canal secundário sequencial c estiver livre por um tempo T_β , a transmissão ocorre; caso con-

trário uma nova disputa pelo canal primário é necessária, reiniciando novamente a requisição de acesso ao meio. Essa descrição do protocolo é interpretada graficamente através da figura 1 e detalhada no fluxograma mostrado na figura 2.

3.1.1 Resultados preliminares

O modelo proposto foi avaliado analiticamente assumindo canal sem ruído, tráfego saturado e um variado número de nós estáticos. Nessas circunstâncias, o PbP-DCF foi comparado com o padrão IEEE 802.11a de acordo com os parâmetros da tabela 2.

Os resultados obtidos em Queiroz (2013) são sintetizados através do gráfico 3. Como se pode observar, o PbP-DCF aproximadamente dobra a vazão do padrão IEEE 802.11. Diante disso, visando contabilizar o impacto de canal com ruído no desempenho do novo protocolo, esse trabalho reproduz a ação do protocolo PbP-DCF no simulador ns-3.

Tabela 2 – Descrição dos parâmetros usados na simulação analítica

Velocidade da rede	10 Mbps
Carga útil do pacote	1436 bytes
Número de estágios de backoff m	3
Tamanho mínimo de <i>Contention Window</i> W	16 slots
Tipo de Modulação	<i>BPSK</i> 1/2
Atraso de propagação	1 μ s
Largura do Canal	1 \times 20 MHz (IEEE 802.11a) 2 \times 10 MHz (PbP-DCF)

Fonte: (QUEIROZ, 2013).

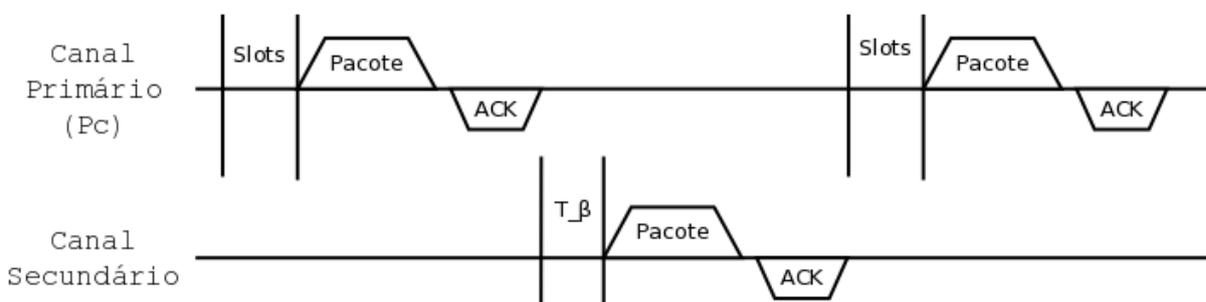


Figura 1 – Exemplo de uma transmissão PbP-DCF com base em uma linha do tempo

Fonte: Autoria Própria.

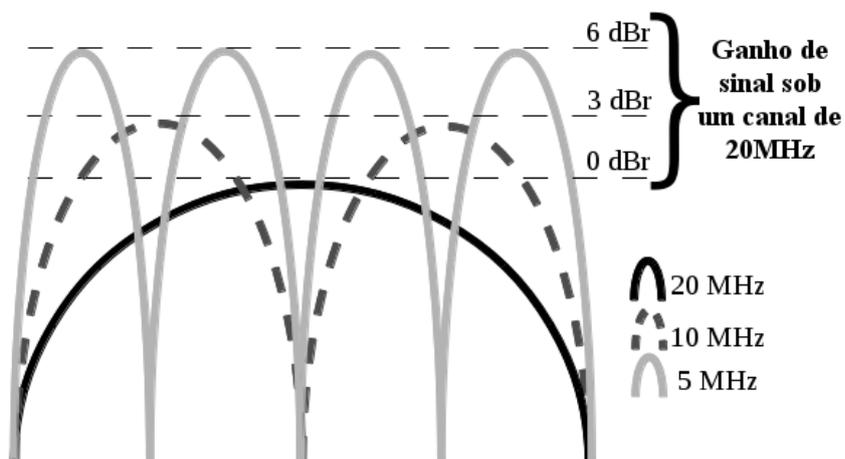


Figura 3 – Algumas possibilidades para as larguras de canais do padrão IEEE 802.11a
Fonte: Queiroz (2014).

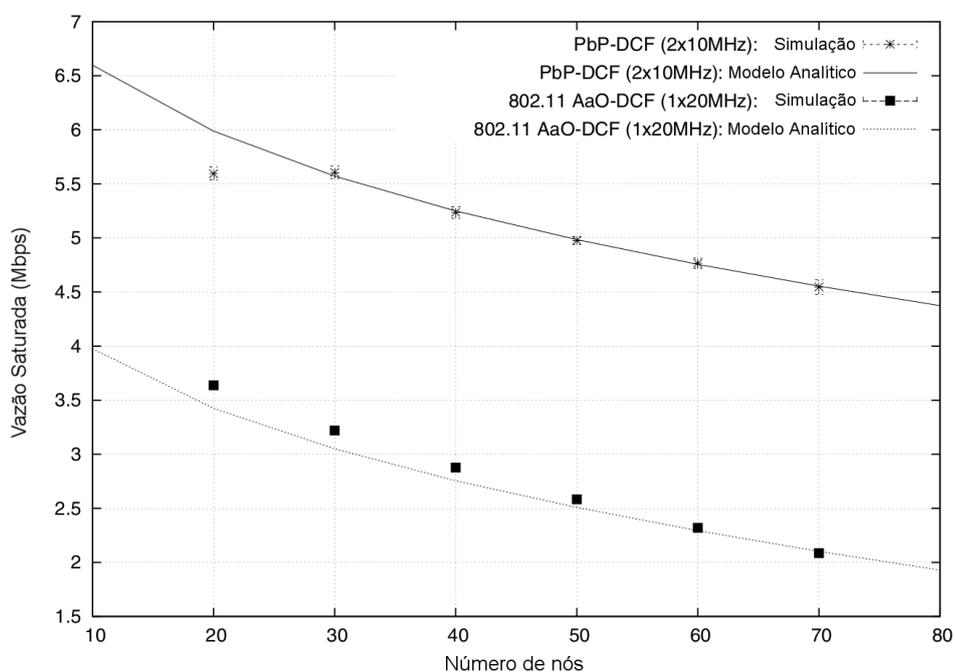


Figura 4 – Vazão da rede pela quantidade de nós conectados utilizando o padrão IEEE 802.11a
Fonte: Queiroz (2013).

4 METODOLOGIA

A condução desse projeto ocorre em três fases, as quais são, seleção do simulador, implementação do protocolo no simulador selecionado e análise comparativa de desempenho entre os protocolos PbP-DCF e CSMA/CA.

No campo da análise de desempenho de redes, avaliações analíticas são empregadas como um ponto de partida para mensurar a capacidade de propostas de protocolos MAC. Contudo, as suposições necessárias ao tratamento analítico do desempenho de um protocolo consistem em fatores limitantes em alguns cenários do mundo real. Adicionalmente, um mesmo modelo analítico está sujeito a profundas alterações estruturais mediante a modificação de uma suposição da rede, resultando num baixo custo-benefício quando deseja-se medir o desempenho sob cenários variados (KHAN; BILAL; OTHMAN, 2013). Neste contexto, o presente trabalho adota simulação como ferramenta de avaliação de desempenho dos protocolos em questão.

Dentre os vários simuladores disponíveis (*e.g.*, NS-2 (1997), NS-3 (2006), Varga (2007), GloMoSiM (1995)) o simulador de redes ns-3 possui destaque dentro da comunidade científica por possibilitar emulação ou simulação em tempo real, análise de tráfego por ferramentas usuais como Wireshark (COMBS, 2010), OTcl API, simulação paralela e bom desempenho e gerenciamento de recursos computacionais. Mais que isso, no foco desse trabalho, o código fonte do ns-3 é bem estruturado e pode ser expandido através de classes C++, enquanto os *scripts* podem também utilizar o C++ ou Python (KHAN; BILAL; OTHMAN, 2013; WEINGARTNER; LEHN; WEHRLE, 2009).

Nessas características a implementação do protocolo PbP-DCF torna-se viável no simulador ns-3, versão 3.19 (a mais recente até o início desse trabalho), de modo a garantir futuras expansões (*i.e.*, para comunicação *full-duplex* (QUEIROZ, 2013)), desempenho eficiente e análise em tempo real de cenários diversos, *e.g.*, nós móveis e diferentes níveis de carga da rede.

Por fim, a análise comparativa de desempenho entre os protocolos PbP-DCF e CSMA/CA é realizada a partir de modelos de simulação que representam diferentes níveis de interferência.

5 IMPLEMENTAÇÃO DO PROTOCOLO PBP-DCF

Tendo o protocolo PBP-DCF como principal característica a modularidade de canais, a primeira etapa da implementação do protocolo em um *script* do ns-3 começa por determinar a largura do canal B_w para obter a quantidade de canais N_c disponíveis. Como na versão do ns-3 utilizada (3.19) o padrão IEEE 802.11ac não está disponível, o IEEE 802.11a foi utilizado, pois também permite larguras variáveis de canal. A largura máxima utilizada foi definida em 20 MHz, $B_w = 20$, e a largura mínima em 10 MHz, $B_n = 10$. Sendo assim, o maior número inteiro da divisão entre largura do canal mais largo pelo mais estreito resulta em 2, $N_c = \lfloor B_w/B_n \rfloor \implies 2 = \lfloor 20/10 \rfloor$.

Define-se como canal primário P_c o canal de número 1, ocupando os primeiros 10 MHz, e nos 10 MHz restantes o canal secundário de número 11. A disposição de canais é feita dessa maneira, pois esses canais são ortogonais devido a estarem distantes o suficientes para não causarem interferência entre si.

Dentro disso, o padrão ainda denota que toda transmissão no canal primário ocorre com disputa usual ao CSMA/CA, e após uma transmissão destinada ao canal primário necessitar do canal secundário, ela não sofre disputa nenhuma e tem apenas uma chance de transmissão. Essa característica para os canais secundários é obtida através do trecho de código mostrado no quadro 1, onde o tamanho de janela de disputa - *Congestion Window* - mínimo e máximo, além do número de tentativas de retransmissões, são estabelecidas em zero.

5.1 A ESTRUTURA DOS COMPUTADORES

O simulador ns-3 utiliza uma nomenclatura particular para o computador físico e a(s) placa(s) de rede(s) disponível(is) no computador. Todo computador é analogicamente nomeado de nó (do inglês, *node*) e toda placa de rede de dispositivo (do inglês, *device*). Diante disso, o ambiente da simulação pode ser configurado com n nó(s) e d dispositivo(s). Cada d dispositivo tem que estar alocado a somente um nó, porém em cada nó podem haver quantos d dispositivos forem necessários.

Nesse contexto, para simulação do ambiente descrito acima, são utilizados três nós, configurados com dois dispositivos cada, em que cada dispositivo de cada nó realiza a comunicação em um dos canais. Sabendo que o ns-3 permite a representação de dispositivos e nós a partir do número zero, todos os dispositivos zeros estabelecem comunicação no canal primário P_c de número um, enquanto todos os dispositivos *uns*, se comunicam através do canal secundário de número onze.

Para a comunicação desses dispositivos e nós dentro do meio sem fio, se utiliza o tipo de transmissão *ad-hoc* devido ao mínimo de sobrecarga computacional

```

void
Experiment::DisableContentionWindow (NodeContainer node_container)
{ // disable dcf congestion window on all secondary channels
  Ptr<Node> node = node_container.Get(0);
  Ptr<NetDevice> dev = node->GetDevice(1);
  Ptr<WifiNetDevice> wifi_dev = DynamicCast<WifiNetDevice>(dev);
  Ptr<WifiMac> mac = wifi_dev->GetMac();

  PointerValue ptr;
  mac->GetAttribute("DcaTxop", ptr);

  Ptr<DcaTxop> dca = ptr.Get<DcaTxop>();
  dca->SetMinCw(0);
  dca->SetMaxCw(0);
}
...
// secondary channels can not retransmit
Config::Set("/NodeList/*/DeviceList/1/$ns3::WifiNetDevice/\
RemoteStationManager/MaxSsrc", StringValue("0"));
Config::Set("/NodeList/*/DeviceList/1/$ns3::WifiNetDevice/\
RemoteStationManager/MaxSlrc", StringValue("0"));

```

Quadro 1 – Procedimento para eliminar disputa DCF

Fonte: Autoria Própria.

proporcionado, pois não são gerados pacotes para descoberta de equipamentos na rede, cadastramento de estações sem fio, autenticação, entre outros.

Para efeitos dessa simulação, e não do protocolo em si, a transmissão realizada foi *uplink*, onde todos os nós transmitem a um nó fixo e esse nó apenas responde se recebeu ou não o pacote com sucesso (*i.e.*, *ACK*), sem enviar dados para os outros nós na rede. Nesse contexto, o nó *um* será o receptor global das transmissões, para ambos os canais, enquanto que o nó *zero* e o nó *dois* serão os transmissores, em ambos os canais. Portanto, qualquer dispositivo *zero* transmitirá sempre com destino fixo ao nó *um*, dispositivo *zero*; enquanto que qualquer dispositivo *um* terá como destino sempre o nó *um*, dispositivo *um*. A representação gráfica supra citada pode ser vista na figura 5

5.2 A CONFIGURAÇÃO DOS CANAIS

A transmissão no canal primário dar-se-á de modo a sempre haver conteúdo a ser transmitido em todo o tempo de simulação, com intuito a sobrecarregar o canal primário sem que exista fragmentação de pacote e requisição de transmissão, do inglês *Request To Send (RTS)* e *Clear To Send (CTS)*. Além disso, as transmissões no canal primário são configuradas para nunca desistirem de transmitir, simulando o

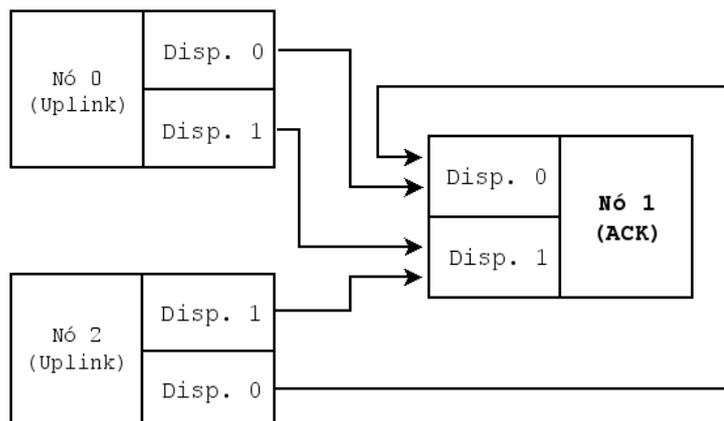


Figura 5 – Comunicação entre nós e dispositivos no cenário da simulação
Fonte: Autoria Própria.

```
// disable fragmentation
Config::SetDefault
  ("ns3::WifiRemoteStationManager::FragmentationThreshold", StringValue
  ("2200"));
Config::SetDefault ("ns3::WifiRemoteStationManager::RtsCtsThreshold",
  StringValue ("2200"));

// Bianchi's last state model
Config::Set("/NodeList/*/DeviceList/0/$ns3::WifiNetDevice/\
RemoteStationManager/MaxSsrc", StringValue ("10000000"));
Config::Set("/NodeList/*/DeviceList/0/$ns3::WifiNetDevice/\
RemoteStationManager/MaxSlrc", StringValue ("10000000"));
```

Quadro 2 – Configurações para o canal primário
Fonte: Autoria Própria.

último estado do modelo de Bianchi (2000). Essa configuração é obtida através do quadro visto em 2.

Para transmissão no canal secundário, ao invés de utilizar o *CNB* definido pelo padrão PbP-DCF, produz-se efeito similar ao realizar o agendamento da transmissão de um único pacote no canal secundário. Então, a cada recebimento de um ACK no canal primário, agenda-se uma transmissão de um *único* pacote no canal secundário com origem de mesmo nó transmitido anteriormente e destino fixo ao nó *um* como já estipulado. Por exemplo, caso o nó zero receba a resposta de um ACK no dispositivo zero, imediatamente realiza-se um agendamento para transmissão de *um* pacote no dispositivo *um* a partir dos mesmos nó zero e destino nó um. Esse exemplo de transmissão no canal secundário é executado pelo código mostrado em 3

Para configuração da velocidade de transmissão do meio de rede sem fio foi utilizado a velocidade fixa de *6 Mbps* em ambos os canais. Essa velocidade foi selecionada pois permite uma boa variação do nível de sinal *rss* antes que não seja mais possível transmitir dados sem diminuir a velocidade do canal. Essa variação do

```

PacketSocketAddress socketPbP;
socketPbP.SetSingleDevice (node0_dev1.Get (0)->GetIfIndex ());
socketPbP.SetPhysicalAddress (node1_dev1.Get (0)->GetAddress ());
socketPbP.SetProtocol (1);

OnOffHelper onoffPbP ("ns3::PacketSocketFactory", Address (socketPbP));
onoffPbP.SetConstantRate (DataRate (dataRateChannel2));
onoffPbP.SetAttribute ("PacketSize", (UIntegerValue) packetSize);
// make sure no more than one pkt is send
onoffPbP.SetAttribute ("MaxBytes", (UIntegerValue) packetSize);

ApplicationContainer appsPbP = onoffPbP.Install (node0.Get (0));
appsPbP.Start (Seconds (0.0));
appsPbP.Stop (Seconds (0.1));

```

Quadro 3 – Transmissão quando troca de canal é realizada

Fonte: Autoria Própria.

nível de sinal será abordada mais adiante na seção 5.4.

5.3 A FUNÇÃO DE VAZÃO

O cenário de simulação proposto tem em vista dar suporte à medição da vazão atingida pelos respectivos protocolos MAC. Sendo assim, uma função denominada *Throughput* realiza a contagem do total de dados transmitidos em todos os canais, durante uma unidade de tempo, *i.e.*, um segundo. Essa contagem é feita pelo tamanho do pacote de dados a ser transmitido (parâmetro configurável), acrescido da carga útil (do inglês, *payload*) utilizado pelo protocolo, *i.e.*, camada *LLC*, gerando 8 bytes adicionais.

Após transcorrido o tempo estipulado, essa variável é zerada para uma nova contagem e seu resultado é impresso na saída padrão, juntamente com o tempo transcorrido de simulação. Dados os quais serão utilizados posteriormente para criação do gráfico de desempenho e comparação com o padrão atual.

Um fato a ser ressaltado é que estando todos os canais em modo *broadcast*, onde as transmissões são ouvidas por todos os nós, a soma do tamanho do pacote é realizada apenas para os nós que estão em modo *uplink*, *i.e.*, nós zero e dois. No código 4, a soma dos dados é disparada através da função *Experiment::AddPayload ()* e a impressão dos bytes acumulados é realizada por *Experiment::Throughput ()*.

5.4 MODELO DE PROPAGAÇÃO

Para modelar a dissipação sofrida pelo sinal *wireless* ao longo do caminho de propagação recorre-se à modelos de propagação. Isso constitui um dos avanços em relação aos resultados obtidos em Queiroz (2013), cujo objetivo foi auferir o “máximo que os protocolos conseguem obter” quando ruídos no meio não constituem obstáculos ao seus trabalhos, ou seja, em canal teórico ideal.

No presente trabalho foi utilizado o modelo de propagação log normal em virtude de sua acuidade para WLANs de 2.4 GHz já ter sido verificada em ambientes reais, e.g. Faria (2005). O referido modelo está implementado internamente ao ns-3 e permite a configuração do expoente de perda do ambiente $-n$, a distância referencial de perda $-d_0$ e a referência de perda dada a certa distância $-L_0$. Dentre esses parâmetros, os valores d_0 e L_0 são deixados pelo padrão estipulado no ns-3 (NS-3, 2014).

A variável de interesse em questão é o expoente de perda do ambiente, n , a qual define o expoente da função logarítmica que impacta a intensidade de dissipação do canal. Quanto maior o expoente de propagação, maior o ruído no canal. Para efeito deste trabalho, os valores de n a que os canais gerenciados pelos protocolos foram submetidos foram variações de 8.83 a 8.87, acrescendo em um ponto centesimal a cada iteração. Tais valores estão de acordo com estimativas obtidas em ambientes reais considerando valores de desvio padrão aceitáveis para o modelo (FARIA, 2005).

Adicionalmente, outros parâmetros de definição do cenário merecem destaque. Analisando os códigos 2 e 3 e 4 percebem-se algumas variáveis não definidas nos trechos de códigos apresentados. Essas variáveis são parâmetros configuráveis em tempo de execução da simulação para facilitar o ajuste dos mesmos, sem a necessidade de uma nova compilação. Os parâmetros ajustáveis pela linha de comando são:

- *packetSize*: Ajusta o tamanho do pacote a ser transmitido. Padronizado em 1000 bytes;
- *dataRate*: Define a velocidade de transmissão dos pacotes dentro da aplicação, independentemente da velocidade do canal *wireless*. Por Padrão definido em 6 Mbps, a mesma velocidade do canal.
- *stopTime*: Estipula o tempo máximo de execução da simulação;
- *sampleInterval*: Tempo de intervalo para coleta dos dados da função *Throughput*, definido em um segundo se não alterado;
- *pathLoss*: Valor do expoente do modelo de perda de propagação log normal.

```
void
Experiment::AddPayload ()
{
    m_bytesTotal += packetSize;
    // add llc headers size as we are working on app layer
    m_bytesTotal += 8;
}
void
Experiment::Throughput ()
{
    // convert time to seconds
    Time timeNow (Simulator::Now ()); int64x64_t nowSeconds =
        timeNow.To(Time::S);

    std::cout << nowSeconds << "\t" << ((m_bytesTotal * 8.0) /
        1000000)/sampleInterval << std::endl;
    m_bytesTotal = 0;

    // check for end of simulation time ('base case')
    if (nowSeconds <= ((int64x64_t) stopTime))
    {
        Simulator::Schedule (Seconds (sampleInterval), &Experiment::Throughput,
            this);
    }
}
```

Quadro 4 – Função para cálculo de vazão
Fonte: Autoria Própria.

6 RESULTADOS

De acordo com a descrição do ambiente, realizaram-se alguns testes para comparação de desempenho entre os protocolos de acesso ao meio, CSMA/CA e PbP-DCF. Os testes seguem o modelo de ambiente citado no capítulo 5, utilizando dois canais de 10 MHz cada, com dois nós em transmissão *uplink* e outro apenas para respostas *ACK*, velocidade de meio sem fio em 6 Mbps, pacotes de 1000 bytes e tempo de simulação de 1516 segundos. Esse tempo de simulação foi baseado na média do número de amostras em Queiroz (2013) e após a coleta dos resultados, os primeiros 253 resultados foram removidos, devido a média do número de amostras consideradas não estacionárias ser aproximadamente 252 (QUEIROZ, 2013).

Em cada simulação foi medida a vazão saturada do canal, conforme definido em Bianchi (2000), onde, a cada segundo, em todos os canais, conta-se todos os dados transmitidos *com sucesso*. Feito isso, o contador de vazão é zerado novamente para reinício da operação no próximo segundo.

Sendo assim, pode-se contabilizar os valores da média de cada simulação, os quais podem ser comparados pela tabela 3 e vistos no gráfico da figura 6. Nele, percebe-se que os valores da média de vazão do PbP-DCF para canal perfeito (sem ruídos) quase dobra a vazão do padrão CSMA/CA, ao mesmo tempo que equipara-se aos resultados obtidos por Queiroz (2013) (reproduzido neste trabalho através do gráfico da figura 4). No entanto, devido ao menor número de nós utilizados nesse trabalho (dois), em relação a Queiroz (2013) pode-se perceber uma vazão relativamente maior.

Para transmissões com ruídos o gráfico da figura 6 evidencia maior média de vazão em relação ao CSMA/CA. Tal vantagem diminui a medida em que os níveis de ruído aumentam, conforme pode-se observar especialmente para os valores 8.83, 8.84 e 8.85 de coeficiente de perda de propagação. Sob tais condições, a redução do esquema de modulação do canal seria necessária (não empregada nesse trabalho) para garantir um desempenho mais eficiente para ambos os protocolos. Nesse caso, o emprego de canais estreitos confere mais flexibilidade ao PbP-DCF para se recuperar de canais com ruído, posto que canais largos sempre exigem qualidade de sinal maior que canais estreitos para um mesmo esquema de modulação (IEEE Standard for Information technology, 2013).

Para averiguar a probabilidade com cada protocolo atinge uma determinada vazão no canal, utiliza-se nesse trabalho o gráfico de função empírica com distribuição cumulativa (do inglês, *Empirical Cumulative Distribution Function* e acrônimo ECDF). Nele, mostra-se a probabilidade cumulativa de uma variável aleatória (*i.e.* vazão, no caso do presente trabalho) assumir um valor menor ou igual a um ponto do eixo das abscissas. Em nosso contexto, um bom resultado traduz-se em uma curva mais à direita no plano cartesiano, significando uma baixa probabilidade da vazão assumir

valores pequenos.

O gráfico da figura 7 apresenta as curvas ECDF de vazão dos protocolos sob circunstância ideal de canal, *i.e.* sem ruído. Nessa circunstância, a probabilidade do protocolo CSMA/CA alcançar no máximo 5 Mbps é 60%. Por sua vez, o PbP-DCF alcança uma vazão de até 9 Mbps com a mesma probabilidade. A vazão nominal teórica máxima para ambos os protocolos é limitada pelo esquema de modulação configurado, sendo 6 Mbps para o primeiro protocolo e 12 Mbps (2×6 Mbps) para o segundo. Contudo, os *overheads* resultantes dos tempos de espera das janelas de disputa, bem como o tempo perdido com pacotes de controle, resultam em queda de vazão útil, mesmo sob canais ideais.

Por sua vez, em todos os cenários com canais sob diferentes níveis de ruído, o PbP-DCF manteve melhor desempenho, conforme pode-se observar nas curvas ECDF das figuras 8, 9, 10, 11 e 12. Observa-se ainda que, a medida em que o nível de ruído aumenta, a diferença no desempenho dos protocolos cai. Sob o cenário com o maior grau de obstrução avaliado (figura 12), a probabilidade de ambos os protocolos auferirem vazão inferior a 1.3 Mbps é 100%.

Um dos principais fatores que explica a queda da vantagem do PbP-DCF frente ao CSMA/CA padrão é que, no cenário avaliado, ambos dependem da mesma qualidade de recepção para demodulação com sucesso. De fato, apesar do canal de 10 MHz ter menos portadoras de dados, ele se beneficia de uma maior quantidade média de energia por portadora, permitindo modulações mais densas e atingindo a mesma taxa nominal que o canal de 20 MHz sob um SNR de -82 dBm (IEEE Standard for Information technology, 2013). Assim, quando o canal degrada, o PbP-DCF sofre aumento de perdas de pacotes no canal primário na mesma proporção que o CSMA/CA padrão. Aliado a isso, o número de transmissões no canal secundário reduzem, pois dependem de uma transmissão com sucesso no canal primário para ocorrer, impactando negativamente na vazão. Apesar disso, o protocolo mantém vantagem no desempenho, posto que cada transmissão de 6 Mbps com sucesso no canal primário significa, outra transmissão de 6 Mbps é assegurada no canal secundário, o que não ocorre com o CSMA/CA padrão IEEE 802.11.

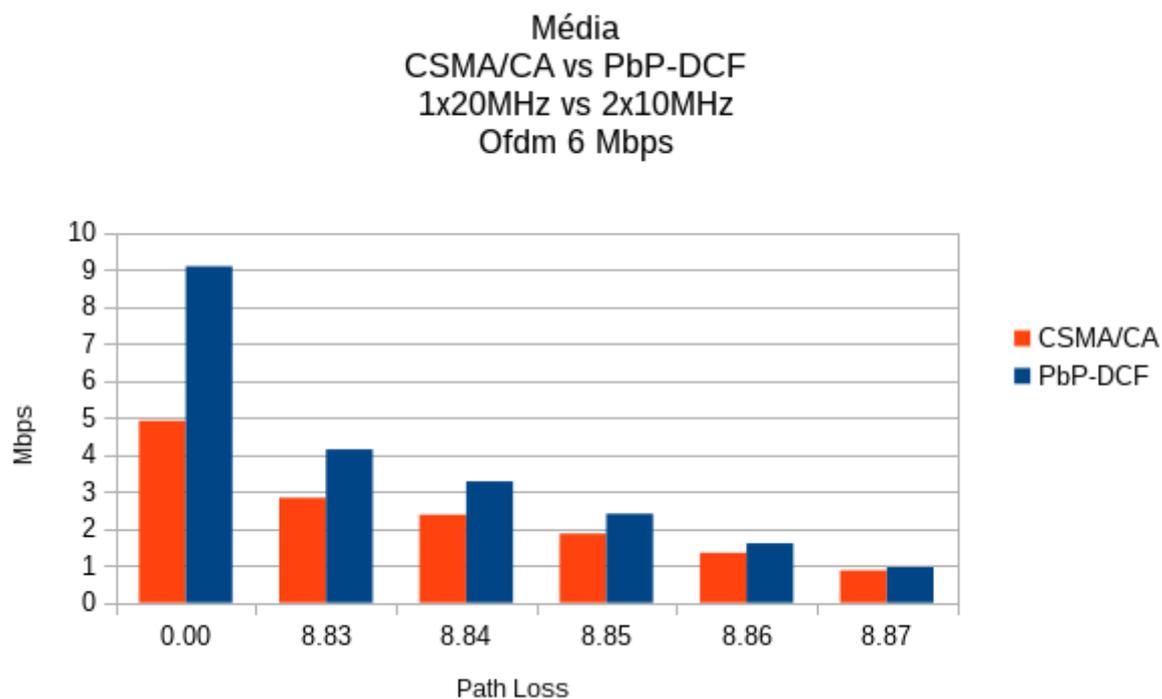


Figura 6 – Comparação da média entre os protocolos por diferentes *path loss*
Fonte: Autoria Própria.

Tabela 3 – Resultados da simulação entre os vários ruídos aplicados

Protocolo	<i>Path loss</i>	Média	Mediana	Desvio Padrão
CSMA/CA	0.0	4.9185	4.9490	0.0488
PbP-DCF		9.0930	9.0961	0.0936
CSMA/CA	8.80	3.8308	3.8304	0.0879
PbP-DCF		6.2765	6.2737	0.1646
CSMA/CA	8.84	2.3794	2.3788	0.1201
PbP-DCF		3.2796	3.2820	0.1948
CSMA/CA	8.85	1.8681	1.8708	0.1296
PbP-DCF		2.4048	2.4111	0.1949
CSMA/CA	8.86	1.3490	1.3466	0.1225
PbP-DCF		1.6037	1.6047	0.1714
CSMA/CA	8.87	0.8739	0.8709	0.1055
PbP-DCF		0.9561	0.9596	0.1319

Fonte: Autoria Própria.

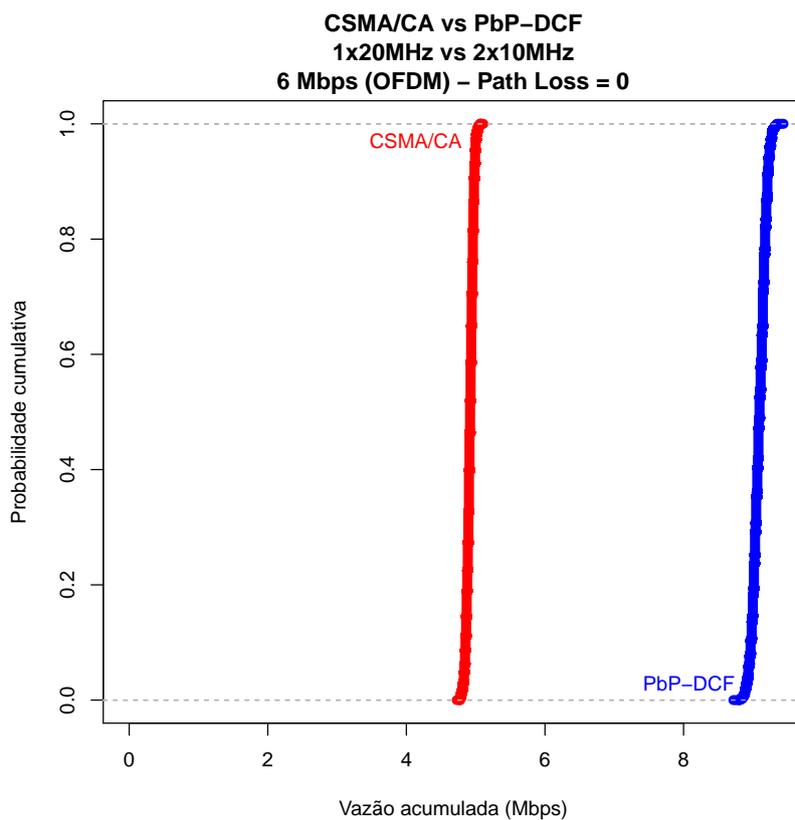


Figura 7 – Resultado da simulação sem ruído no canal *wireless*
Fonte: Autoria Própria.

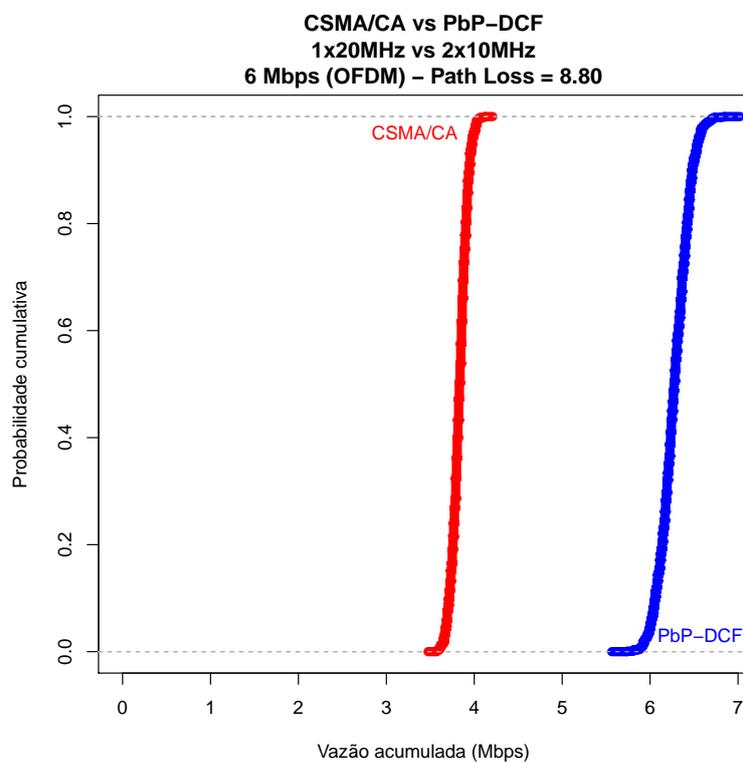


Figura 8 – Resultado da simulação com *path loss* definido em 8.80
Fonte: Autoria Própria.

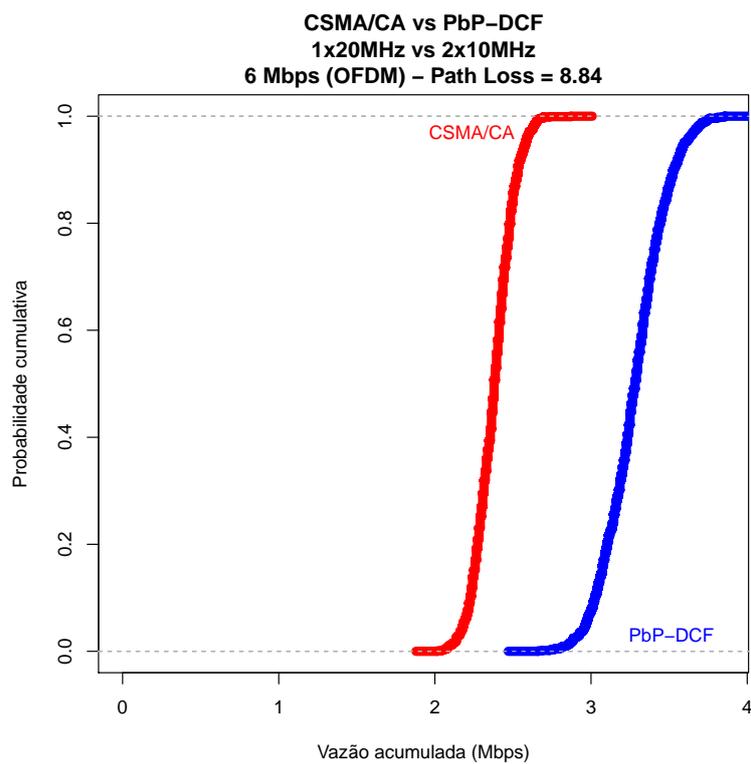


Figura 9 – Resultado da simulação com *path loss* definido em 8.84
Fonte: Autoria Própria.

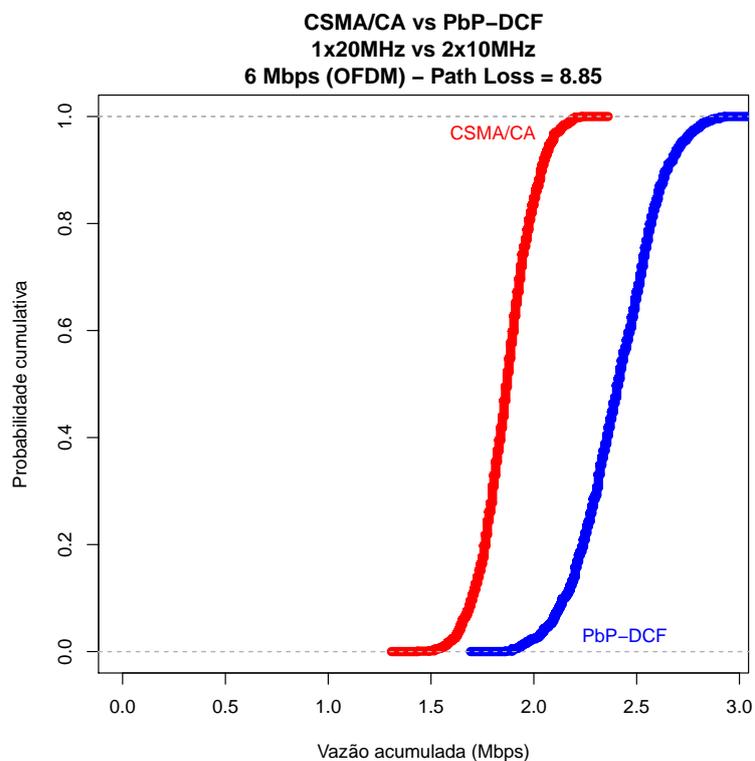


Figura 10 – Resultado da simulação com *path loss* definido em 8.85
Fonte: Autoria Própria.

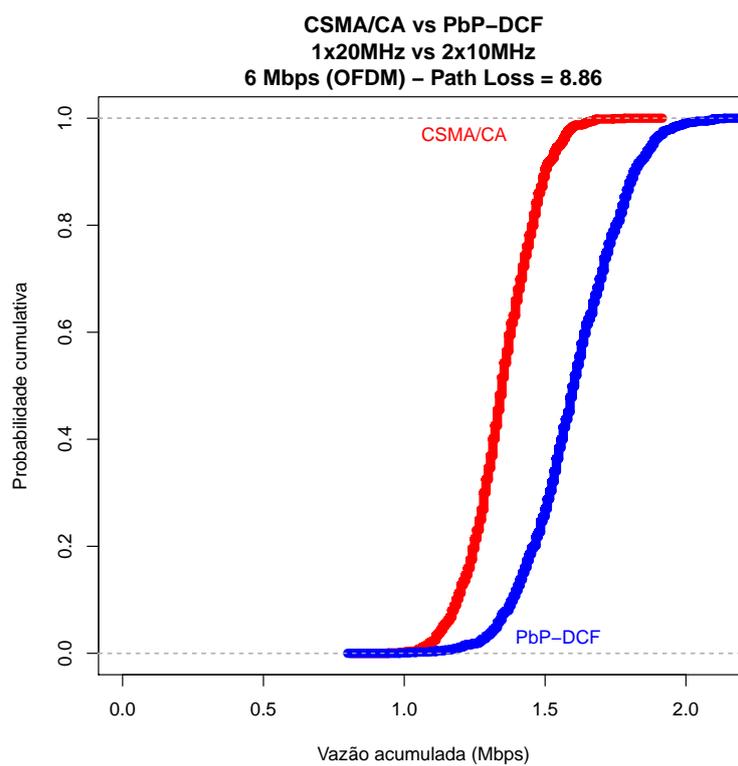


Figura 11 – Resultado da simulação com *path loss* definido em 8.86
Fonte: Autoria Própria.

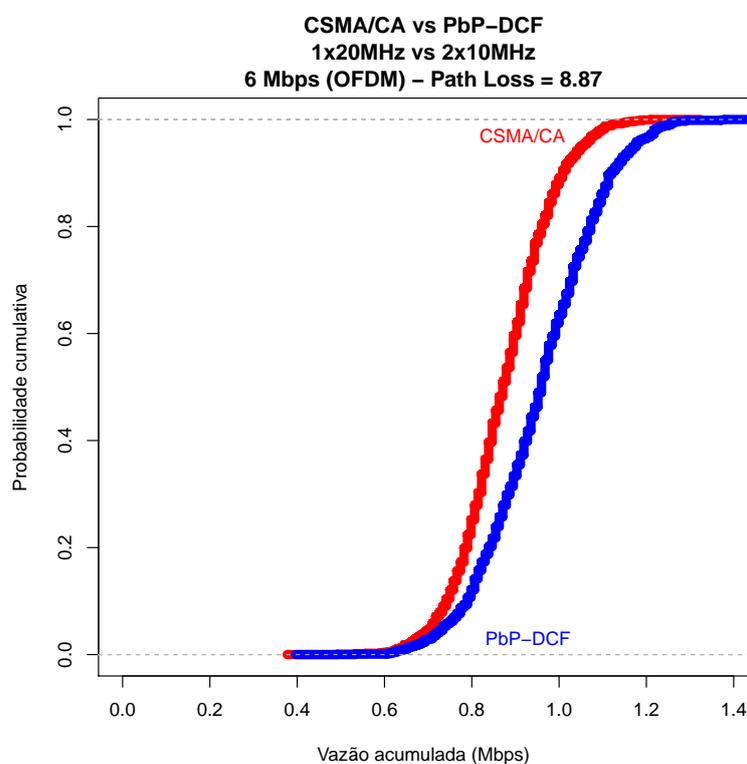


Figura 12 – Resultado da simulação com *path loss* definido em 8.87
Fonte: Autoria Própria.

7 CONCLUSÃO

Nesse trabalho apresentou-se a simulação e os resultados do protocolo de acesso ao meio PbP-DCF considerando canais com ruídos. Utilizando o simulador ns-3, versão 3.19, implementou-se um *script* simulando as características do protocolo para análise de comportamento em alguns cenários com diferentes níveis de ruídos simulados pelo módulo de propagação *Log Distance Propagation Loss Model* disponível no simulador ns-3.

No cenário onde não encontra-se ruído a vantagem do PbP-DCF em relação ao CSMA/CA chegou a quase o dobro. Em contrapartida, em cenários com pouco ruído, a vantagem chega a ser de até 1 Mbps maior em vazão média, enquanto que em cenários com altos níveis de ruído, a diferença cai para aproximadamente 0.3 Mbps.

Um potencial trabalho futuro, consiste na avaliação de desempenho do PbP-DCF em redes veiculares (e.g. IEEE 802.11p). Devido a essas redes empregarem canais estreitos únicos, o PbP-DCF poder-se-ia conseguir um aumento na vazão pelo uso de múltiplas transmissões estreitas sem prejuízo do grau de disputa devido ao emprego de canais ortogonais. Esse estudo requer o uso de modelos de mobilidade e seu aprofundamento é mais adequado para trabalhos futuros.

REFERÊNCIAS

- BIANCHI, G. Performance analysis of the ieee 802.11 distributed coordination function. **Selected Areas in Communications, IEEE Journal on**, v. 18, n. 3, p. 535–547, March 2000. ISSN 0733-8716.
- CHANDRA, R. *et al.* A case for adapting channel width in wireless networks. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 38, n. 4, p. 135–146, ago. 2008. ISSN 0146-4833. Disponível em: <<http://doi.acm.org/10.1145/1402946.1402975>>.
- COMBS, G. **Wireshark**. 2010. Disponível em: <<http://www.wireshark.org>>. Acesso em: 12 fev. 2014.
- FARIA, D. B. **Modeling Signal Attenuation in IEEE 802.11 Wireless LANs - Vol. 1**. [S.l.], 2005. Disponível em: <<http://www-cs-students.stanford.edu/~dbfaria/files/faria-TR-KP06-0118.pdf>>.
- GLOMOSIM. **GloMoSiM - Global Mobile Information Systems Simulation Library**. 1995. Disponível em: <<http://pcl.cs.ucla.edu/projects/glomosim/>>. Acesso em: 12 fev. 2014.
- GUMMADI, R. *et al.* Understanding and mitigating the impact of rf interference on 802.11 networks. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 37, n. 4, p. 385–396, ago. 2007. ISSN 0146-4833. Disponível em: <<http://doi.acm.org/10.1145/1282427.1282424>>.
- IEEE Standard for Information technology. 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–amendment 4: Enhancements for very high throughput for operation in bands below 6 ghz. **IEEE Std 802.11ac-2013 (Amendment to IEEE Std 802.11-2012, as amended by IEEE Std 802.11ae-2012, IEEE Std 802.11aa-2012, and IEEE Std 802.11ad-2012)**, p. 1–425, Dec 2013.
- JAIN, R. **The art of computer systems performance analysis**. [S.l.]: John Wiley & Sons Chichester, 1991.
- KHAN, A. ur R.; BILAL, S. M.; OTHMAN, M. A performance comparison of network simulators for wireless networks. **CoRR**, abs/1307.4129, 2013.
- MOSCIBRODA, T. *et al.* Load-aware spectrum distribution in wireless lans. In: **Network Protocols, 2008. ICNP 2008. IEEE International Conference on**. [S.l.: s.n.], 2008. p. 137–146. ISSN 1092-1648.
- NS-2. **Network Simulator-2 is an open source, discrete event network simulator**. 1997. Disponível em: <http://nslam.isi.edu/nslam/index.php/User_Information>. Acesso em: 21 nov. 2013.
- NS-3. **NS-3 is a discrete-event network simulator for Internet systems, targeted primarily for research and educational use**. 2006. Disponível em: <<https://www.nslam.org/>>. Acesso em: 11 nov. 2013.

_____. **ns3::LogDistancePropagationLossModel Class Referenc**. 2014. Ns-3 website. Disponível em: <http://www.nsnam.org/doxygen/classns3_1_1_log_distance_propagation_loss_model.html>. Acesso em: 11 nov. 2013.

PARK, M. Ieee 802.11ac: Dynamic bandwidth channel access. In: **Communications (ICC), 2011 IEEE International Conference on**. [S.l.: s.n.], 2011. p. 1–5. ISSN 1550-3607.

QUEIROZ, S. All-at-once or piece-by-piece: How to access wide channels in wlans with channel width diversity? **Communications Letters, IEEE**, v. 17, n. 11, p. 2188–2191, November 2013. ISSN 1089-7798.

QUEIROZ, S. **WIRELESS FULL-DUPLEX ON-DEMAND SPECTRUM ALLOCATION MAC PROTOCOL**. Tese (Doutorado) — UFPR, 2014.

VARGA, A. **OMNeT++ is a discrete event simulation environment**. 2007. Disponível em: <<http://www.omnetpp.org/>>. Acesso em: 12 fev. 2014.

WEINGARTNER, E.; LEHN, H. vom; WEHRLE, K. A performance comparison of recent network simulators. In: **Communications, 2009. ICC '09. IEEE International Conference on**. [S.l.: s.n.], 2009. p. 1–5. ISSN 1938-1883.

Apêndices

APÊNDICE A – SCRIPT NS-3 UTILIZADO PARA SIMULAÇÃO

```

Gnuplot2dDataset
Experiment::Run (const WifiHelper &wifi, const YansWifiPhyHelper &wifiPhy,
                const NqosWifiMacHelper &wifiMac, const YansWifiChannelHelper
                &wifiChannel)
{
    m_bytesTotal = 0;

    node0.Create (1);
    node1.Create (1);
    node2.Create (1);
    utilizado
    PacketSocketHelper packetSocket;
    packetSocket.Install (node0);
    packetSocket.Install (node1);
    packetSocket.Install (node2);

    phy = wifiPhy;
    phy.SetChannel (wifiChannel.Create ());

    mac = wifiMac;
    phy.Set("ChannelNumber", UIntegerValue (1));
    node0_dev0 = wifi.Install (phy, mac, node0);
    node1_dev0 = wifi.Install (phy, mac, node1);
    node2_dev0 = wifi.Install (phy, mac, node2);

    WifiHelper wifiPbp = WifiHelper::Default ();
    NqosWifiMacHelper wifiMacPbp = NqosWifiMacHelper::Default ();
    YansWifiPhyHelper wifiPhyPbp = YansWifiPhyHelper::Default ();
    YansWifiChannelHelper wifiChannelPbp = YansWifiChannelHelper::Default ();
    // Radiotap extension
    wifiPhyPbp.SetPcapDataLinkType (YansWifiPhyHelper::DLT_IEEE802_11_RADIO);
    wifiPbp.SetStandard (WIFI_PHY_STANDARD_80211_10MHZ);

    Ssid ssidPbp = Ssid ("enable-pbp");
    wifiMacPbp.SetType ("ns3::AdhocWifiMac",
                       "Ssid", SsidValue (ssidPbp));
    std::string phyModePbp ("OfdmRate6MbpsBW10MHz");
    wifiPbp.SetRemoteStationManager ("ns3::ConstantRateWifiManager",
                                     "DataMode", StringValue (phyModePbp));

```

```

YansWifiPhyHelper phyPbp = wifiPhyPbp;
phyPbp.SetChannel (wifiChannelPbp.Create ());
phyPbp.Set("ChannelNumber", UIntegerValue (11));
node0_dev1 = wifiPbp.Install (phyPbp, wifiMacPbp, node0);
node1_dev1 = wifiPbp.Install (phyPbp, wifiMacPbp, node1);
node2_dev1 = wifiPbp.Install (phyPbp, wifiMacPbp, node2);
// permitir 'infinitas' retransmissoes (canal primario) para simular modelo
// de Bianchi em que no nao
// desiste de transmitir i.e. nao sai do ultimo estado enquanto nao
// transmitir com sucesso
Config::Set("/NodeList/*/DeviceList/0/$ns3::WifiNetDevice/\
RemoteStationManager/MaxSsrc", StringValue ("10000000"));
Config::Set("/NodeList/*/DeviceList/0/$ns3::WifiNetDevice/\
RemoteStationManager/MaxSlrc", StringValue ("10000000"));
//Config::Connect("/NodeList/*/DeviceList/0/$ns3::WifiNetDevice/\
RemoteStationManager/MaxSlrc", MakeCallback (&PhyTxBegin, this));

// just primary channels need to have backoff
// need to be called after installing nodes
DisableContentionWindow (node0);
DisableContentionWindow (node1);
DisableContentionWindow (node2);

// secondary channels can not retransmit
Config::Set("/NodeList/*/DeviceList/1/$ns3::WifiNetDevice/\
RemoteStationManager/MaxSsrc", StringValue("0"));
Config::Set("/NodeList/*/DeviceList/1/$ns3::WifiNetDevice/\
RemoteStationManager/MaxSlrc", StringValue("0"));
// Path loss exponent
// need to be called nearby. In main () does not works!
Config::Set("/ChannelList*/$ns3::YansWifiChannel/\
PropagationLossModel/$ns3::LogDistancePropagationLossModel/Exponent",
StringValue(pathLoss));

MobilityHelper mobility;
Ptr<ListPositionAllocator> positionAlloc =
    CreateObject<ListPositionAllocator> ();
positionAlloc->Add (Vector (0.0, 0.0, 0.0));
positionAlloc->Add (Vector (5.0, 0.0, 0.0));
mobility.SetPositionAllocator (positionAlloc);
mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");

```

```

mobility.Install (node0);
mobility.Install (node1);
mobility.Install (node2);

// node 0
PacketSocketAddress socket;
socket.SetSingleDevice (node0_dev0.Get (0)->GetIfIndex ());
socket.SetPhysicalAddress (node1_dev0.Get (0)->GetAddress ());
socket.SetProtocol (1);

OnOffHelper onoff ("ns3::PacketSocketFactory", Address (socket));
onoff.SetConstantRate (DataRate (dataRate));
onoff.SetAttribute ("PacketSize", (UIntegerValue) packetSize);

ApplicationContainer apps = onoff.Install (node0.Get (0));
apps.Start (Seconds (0.0));
apps.Stop (Seconds (stopTime));

// node 2
PacketSocketAddress socket_node2;
socket_node2.SetSingleDevice (node2_dev0.Get (0)->GetIfIndex ());
socket_node2.SetPhysicalAddress (node1_dev0.Get (0)->GetAddress ());
socket_node2.SetProtocol (1);

OnOffHelper onoff_node2 ("ns3::PacketSocketFactory", Address (socket_node2));
onoff_node2.SetConstantRate (DataRate (dataRate));
onoff_node2.SetAttribute ("PacketSize", (UIntegerValue) packetSize);

ApplicationContainer apps_node2 = onoff_node2.Install (node2.Get (0));
apps_node2.Start (Seconds (0.0));
apps_node2.Stop (Seconds (stopTime));

// prepare out file header
std::cout << "Segundos" << "\t" << "Total de Mbits/s" << std::endl;
Config::Connect
    ("/NodeList/*/DeviceList*/$ns3::WifiNetDevice/Phy/State/RxOk",
    MakeCallback (&Experiment::PhyRxOkTrace, this));

Simulator::Schedule (Seconds (sampleInterval), &Experiment::Throughput, this);

Simulator::Run ();

```

```
    Simulator::Destroy ();  
  
    return m_output;  
}  
// vim:set ts=2 sw=2 sta et ai si cin:
```

Quadro 3 – Script ns-3 utilizado para simulação do PbP-DCF

Fonte: Autoria Própria.