

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS**

**CENTRO DE CIÊNCIAS EXATAS, AMBIENTAIS E DE  
TECNOLOGIAS**

**LUIS FERNANDO FONTOLAN**

**POLÍTICA DE QOS PARA REDES IEEE802.11 COM  
SELEÇÃO DE TAXA DE SERVIÇO BASEADA EM ÍNDICE  
DE JUSTIÇA**

**PUC CAMPINAS**

**2010**

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS**

**GRÃO-CHANCELER**

Dom Bruno Gamberini

**MAGNÍFICA REITORA**

Profa. Dra. Angela de Mendonça Engelbrecht

**VICE-REITOR**

Prof. Dr. Eduard Prancic

**PRÓ-REITORA DE PESQUISA E PÓS-GRADUAÇÃO**

Profa. Dra. Vera Engler Cury

**DIRETOR DO CENTRO DE CIÊNCIAS EXATAS, AMBIENTAIS E DE  
TECNOLOGIAS**

Prof. Ricardo Luís de Freitas

**COORDENADOR DO PROGRAMA DE PÓS-GRADUAÇÃO *STRICTO SENSU*  
EM ENGENHARIA ELÉTRICA**

**CURSO DE MESTRADO PROFISSIONAL EM GESTÃO DE REDES DE  
TELECOMUNICAÇÕES**

**ÁREA DE CONCENTRAÇÃO: GESTÃO DE REDES E SERVIÇOS**

Prof. Dr. Marcelo Luis Francisco Abbade

**LUIS FERNANDO FONTOLAN**

**POLÍTICA DE QOS PARA REDES IEEE802.11 COM  
SELEÇÃO DE TAXA DE SERVIÇO BASEADA EM ÍNDICE  
DE JUSTIÇA**

Dissertação apresentada como exigência para obtenção do Título de Mestre em Gestão de Redes de Telecomunicações, ao Programa de Pós-Graduação em Engenharia Elétrica, Pontifícia Universidade Católica de Campinas.

Orientador: Prof. Dr. Alexandre de Assis Mota

**PUC CAMPINAS**

**2010**

Ficha Catalográfica  
Elaborada pelo Sistema de Bibliotecas e  
Informação - SBI - PUC-Campinas

t621.3845 Fontolan, Luis Fernando.  
F684p Política de QOS para redes IEEE802.11 com seleção de taxa de  
serviço baseada em índice de justiça / Luis Fernando Fontolan. –  
Campinas: PUC-Campinas, 2010.  
108p., ilu.

Orientador: Alexandre de Assis Mota.  
Dissertação (mestrado) - Pontifícia Universidade Católica de  
Campinas, Centro de Ciências Exatas, Ambientais e de Tecnologias,  
Pós-Graduação em Engenharia Elétrica.  
Inclui bibliografia e anexos.

1. Sistemas de comunicação sem fio. 2. Algoritmos de  
computador. 3. Provedores de serviços da Internet. 4. Simulação  
(Computadores). 5. Software de comunicação. 6. Sistemas de  
telecomunicação. I. Mota, Alexandre de Assis. II. Pontifícia  
Universidade Católica de Campinas. Centro de Ciências Exatas,  
Ambientais e de Tecnologias. Pós-Graduação em Engenharia  
Elétrica. III. Título.

22.ed. CDD – t621.3845

**LUIS FERNANDO FONTOLAN**

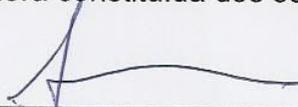
**POLÍTICA DE QOS PARA REDES IEEE802.11 COM  
SELEÇÃO DE TAXA DE SERVIÇO BASEADA EM ÍNDICE  
DE JUSTIÇA**

Dissertação apresentada ao Curso de Mestrado Profissional em Gestão de Redes de Telecomunicações do Centro de Ciências Exatas, Ambientais e de Tecnologias da Pontifícia Universidade Católica de Campinas como requisito parcial para obtenção do título de Mestre em Gestão de Redes de Telecomunicações.

Área de Concentração: Gestão de Redes e Serviços.

Orientador: Prof. Dr. Alexandre de Assis Mota

Dissertação defendida e aprovada em 26 de fevereiro de 2010 pela Comissão Examinadora constituída dos seguintes professores:



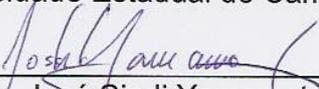
---

Prof. Dr. Alexandre de Assis Mota  
Orientador da Dissertação e Presidente da Comissão Examinadora  
Pontifícia Universidade Católica de Campinas



---

Prof. Dr. Marcos Júlio Rider Flores  
Universidade Estadual de Campinas



---

Prof. Dr. José Sindi Yamamoto  
Pontifícia Universidade Católica de Campinas

A meus pais, Regina e Antonio, que sempre me apoiam e que nunca medem esforços para que eu possa vencer todas as etapas de minha formação.

À minha esposa Marilza, por todo amor, carinho e compreensão.

## **AGRADECIMENTOS**

Agradeço a Deus, por ter me dado a oportunidade e a força necessária para conclusão deste trabalho.

A meus pais, Regina e Antonio, que nunca medem esforços para me dar o melhor, e que sempre se preocupam e investem em minha formação.

À minha esposa Marilza, por todo amor e pelos incontáveis momentos de carinho e compreensão.

À minha família, pela compreensão e apoio.

Ao meu orientador, Prof. Dr. Alexandre de Assis Mota, por ter dado vida a este trabalho, e por sua dedicação e inteligência, fundamentais na conclusão desta Dissertação.

Ao Prof. Dr. Omar Carvalho Branquinho, pelas informações, experiências e materiais compartilhados.

Aos Professores Dr. Marcelo Abbade, Dra. Norma Reggiani e Dr. Davi Bianchini pelas valiosas experiências compartilhadas durante as aulas.

À minha amiga Rina, companheira de mestrado, por ter sido um exemplo de esforço e dedicação.

Aos meus companheiros de mestrado, pelos momentos de alegria, sucesso, esforço e trabalho compartilhados durante este desafio.

Aos amigos e colegas de trabalho da PST Eletrônica SA, pelo excelente ambiente de trabalho, e valiosas contribuições.

“Feliz do homem que encontrou a sabedoria, daquele que adquiriu a inteligência, porque mais vale esse lucro que o da prata, e o fruto que se obtém é melhor que o fino ouro.”

Provérbios de Salomão, Capítulo 3, versículos 13 e 14.

## RESUMO

FONTOLAN, Luis Fernando. *Política de QoS para redes IEEE802.11 com seleção de taxa de serviço baseada em índice de justiça*. Folhas 108f. Dissertação (Mestrado em Gestão de Redes de Telecomunicações) – Pós-Graduação em Engenharia Elétrica, Centro de Ciências Exatas, Ambientais e de Tecnologias, Pontifícia Universidade Católica de Campinas. Campinas, 2010.

O padrão IEEE802.11 foi especificado presumindo condições de igualdade em relação ao acesso ao meio por parte das *STAs* de uma rede, pelo emprego do método *DCF*, onde todas as *STAs* possuem a mesma probabilidade de acesso ao meio, sem levar em consideração sua taxa de transmissão ou sua relação sinal-ruído. No entanto, é necessário ressaltar que tais redes apresentam uma anomalia, através da qual é possível que uma *STA* com baixa taxa de transmissão venha a ocupar o canal por um grande período de tempo. Nesse contexto, esse trabalho apresenta uma proposta de política de QoS baseada na implementação do padrão 802.11e, que objetiva manter um cenário de maior justiça em redes IEEE802.11. Para isso é proposto um algoritmo de cálculo de índice de justiça através da análise da taxa de conexão, do tipo de serviço e dos requisitos necessários para atendimento do serviço. Assim, as *STAs* são servidas a taxas que levam em conta as características físicas da rede, de forma a otimizar os seus recursos, do ponto de vista do provedor. Resultados obtidos com a política proposta em redes simuladas através do *software* NS2 demonstram que os cenários analisados resultaram em um ganho em relação ao *throughput* da rede ou, no mínimo, em uma condição semelhante à obtida sem a utilização da política.

**Termos de indexação:** IEEE 802.11, IEEE 802.11e, *Wifi*, QoS, Índice de Justiça

## **ABSTRACT**

FONTOLAN, Luis Fernando. *QoS policy for IEEE802.11 networks with service rate based on justice index*. Folhas 108f. Dissertação (Mestrado em Gestão de Redes de Telecomunicações) – Pós-Graduação em Engenharia Elétrica, Centro de Ciências Exatas, Ambientais e de Tecnologias, Pontifícia Universidade Católica de Campinas. Campinas, 2010.

*The IEEE802.11 standard assumes that the stations (STAs) of a communications network have the same conditions during medium access, through the usage of the DCF (Distributed Coordination Function). The DCF states that all stations have the same probability of medium access, regardless of transmission rates or signal to noise ratios. Furthermore, such networks also presents an anomaly by which it is possible that a station with low transmission rate may request the channel for an extended period of time. In this context, this work proposes a QoS policy-based implementation based on the 802.11e standard, which aims to maintain a scenario of greater justice in IEEE802.11 networks. In doing so, it is proposed an algorithm to determine a Justice Index associated to a specific network, by examining the STAs connection rate, their type of service and the requirements for customer service. Thus, stations are served at rates that consider the physical characteristics of the network, in order to optimize their resources, from the viewpoint of the service provider. Results were obtained on simulated networks using the NS2 software considering the proposed policy of justice and show that the analyzed scenarios yield network throughput gain or at least result in a condition similar to that obtained without the use of the proposed justice policy.*

**Index terms:** IEEE 802.11, IEEE 802.11e, Wifi, QoS, Justice Index

## LISTA DE FIGURAS

Figura 1. Probabilidade de erros para os métodos mais comuns de modulação.....	24
Figura 2. Arquitetura da Mac 802.11.....	26
Figura 3. Transmissão de pacotes no método DCF, com o mecanismo CSMA/CA .....	28
Figura 4. Acesso ao meio através do método DCF com CSMA/CA (IEEE Std 802.11 – 2007) .....	29
Figura 5. Arquitetura da Mac 802.11e.....	33
Figura 6. Mapeamento de prioridades para as categorias de acesso .....	34
Figura 7. Modelo de implementação de uma STA no método EDCA .....	35
Figura 8. Relação entre os IFSs do IEEE802.11 .....	36
Figura 9. Sequência de funcionamento do método EDCA com controle de admissão .....	39
Figura 10. Fluxograma do algoritmo de cálculo do índice de justiça usando o MLTC .....	50
Figura 11. Fluxograma do algoritmo final de cálculo do índice de justiça usando MLTC e MLTS..	55
Figura 12. Fluxograma simplificado do algoritmo completo.....	56
Figura 13. Cenário Geral de Testes.....	58
Figura 14. Cenário de simulação da Anomalia da MAC em rede com duas STAs .....	59
Figura 15. Seqüência de Simulação da Rede para o caso 1 .....	60
Figura 16. Anomalia da MAC .....	60
Figura 17. Cenário de simulação da Priorização da Anomalia em rede com duas STAs .....	61
Figura 18. Seqüência de Simulação da Rede para o caso 2 .....	62
Figura 19. Priorização da Anomalia .....	63
Figura 20. Placa de Desenvolvimento utilizada para implementação da Política de Justiça .....	64
Figura 21. Todas as combinações possíveis para duas STAs .....	65
Figura 22. Saída do Barramento de Pinos do Processador .....	66
Figura 23. Exemplo de captura das informações transmitidas via porta serial .....	67
Figura 24. Número de combinações versus o número de STAs .....	68
Figura 25. Tempo de cálculo para uma combinação versus o número de STAs.....	69
Figura 26. Taxas médias obtidas após período de estabilização para as diferentes combinações do cenário 1.....	76
Figura 27. Análise de desempenho pela aplicação da política para o cenário 1. ....	77
Figura 28. Taxas médias obtidas após período de estabilização para as diferentes combinações do cenário 2.....	80
Figura 29. Análise de desempenho pela aplicação da política para o cenário 2. ....	81
Figura 30. Taxas médias obtidas após período de estabilização para as diferentes combinações do cenário 3.....	84
Figura 31. Taxas médias obtidas após período de estabilização para as diferentes combinações do cenário 4.....	87

## LISTA DE TABELAS

Tabela 1. Eficiência na taxa de transmissão 802.11b .....	31
Tabela 2. Valores utilizados para o fator $K_{conRate}$ .....	44
Tabela 3. Valores utilizados para o fator $K_{servType}$ .....	44
Tabela 4. Valores utilizados para a o fator $K_{servRate}$ .....	45
Tabela 5. Resultados observados durante o desenvolvimento da política.....	52
Tabela 6. Tempo de processamento para uma combinação em função do número de STAs .....	68
Tabela 7. Taxas de conexão e parâmetros de serviço para o cenário 1 .....	72
Tabela 8. Resultados do cálculo do índice de Justiça para todas as combinações do cenário 1, segundo o MLTS.....	73
Tabela 9. Resultados do cálculo do índice de Justiça para todas as combinações possíveis do cenário 1, segundo o MLTC.....	75
Tabela 10. Taxas de conexão e parâmetros de serviço para o cenário 2.....	78
Tabela 11. Resultados do cálculo do Índice de Justiça para todas as combinações do cenário 2, segundo o MLTS.....	79
Tabela 12. Taxas de conexão e parâmetros de serviço para o cenário 3.....	82
Tabela 13. Resultados do cálculo do índice de Justiça para todas as combinações do cenário 2, segundo os métodos MLTC e MLTS. ....	83
Tabela 14. Taxas de conexão e parâmetros de serviço para o cenário 4.....	85
Tabela 15. Taxas de conexão das STAs para as combinações 1 e 49 do cenário 4.....	86
Tabela 16. Taxas obtidas e Resultados do cálculo do Índice de Justiça para as combinações 1 e 49 do cenário 4.....	86

## LISTA DE ABREVIATURAS E SIGLAS

AC	=	<i>Access Category</i>
ACK	=	<i>Acknowledgment</i>
AIFSN	=	<i>Arbitration InterFrame Space Number</i>
AP	=	<i>Access Point</i>
AWGN	=	<i>Additive White Gaussian Noise</i>
BER	=	<i>Bit Error Rate</i>
CBR	=	<i>Constant Bitrate</i>
CSMA/CA	=	<i>Carrier Sense Multiple Access with Collision Avoidance</i>
CW	=	<i>Contention Window</i>
DBPSK	=	<i>Differential Binary Phase Shift Keying</i>
DCF	=	<i>Distributed Coordination Function</i>
DIFS	=	<i>Distributed Interframe Space</i>
DQPSK	=	<i>Differential Quadrature Phase Shift Keying</i>
$E_b$	=	<i>Energy per Bit</i>
EDCA	=	<i>Enhanced Distributed Channel Access</i>
FER	=	<i>Frame Error Rate</i>
FTP	=	<i>File Transfer Protocol</i>
HCCA	=	<i>Hybrid Controlled Channel Access</i>
HCF	=	<i>Hybrid Coordination Function</i>
HTTP	=	<i>Hypertext Transfer Protocol</i>
IEEE	=	<i>Institute of Electrical and Electronics Engineers</i>
IFS	=	<i>Interframe Space</i>
MAC	=	<i>Medium Access Control</i>
MPDU	=	<i>MAC Data Protocol Unit</i>
MSDU	=	<i>MAC Service Data Unity</i>
MSK	=	<i>Minimum Shift Keying</i>
$N_o$	=	<i>Spectral Noise Density</i>
OFSK	=	<i>Optical Frequency Shift Keying</i>
OOK	=	<i>On-Off Keying</i>
PC	=	<i>Point Coordinator</i>
PCF	=	<i>Point Coordination Function</i>

PLCP	= <i>Physical Layer Convergence Procedure</i>
PSK	= <i>Phase Shift Keying</i>
QoS	= <i>Quality of Service</i>
SIFS	= <i>Short Inter Frame Space</i>
SNR	= <i>Signal Noise Ratio</i>
SNR	= <i>Signal Noise Ratio</i>
STA	= <i>Station</i>
TXOP	= <i>Transmission Opportunity</i>
UDP	= <i>User Datagram Protocol</i>
UP	= <i>User Priority</i>
VoIP	= <i>Voice over IP</i>
WLAN	= <i>Wireless Local Area Network</i>
WMAN	= <i>Wireless Metropolitan Area Network</i>
WPAN	= <i>Wireless Personal Area Network</i>

# SUMÁRIO

1.	INTRODUÇÃO .....	17
1.1.	Contextualização do Problema .....	17
1.2.	Justificativa Para o Desenvolvimento do Trabalho .....	18
1.3.	Objetivo do Trabalho.....	19
1.4.	Resultados Esperados.....	19
1.5.	Delimitação da Pesquisa .....	20
1.6.	Organização da Dissertação.....	20
2.	REDES SEM FIO.....	22
2.1.	Taxas de transmissão .....	22
2.2.	A MAC IEEE 802.11 .....	26
2.3.	A MAC IEEE 802.11e .....	32
3.	Proposta de Política de Qualidade de Serviço - QoS .....	41
3.1.	Definição do Conceito de Justiça.....	41
3.2.	A Política de QoS.....	42
3.3.	Definição e Cálculo do Índice de Justiça .....	43
3.4.	Algoritmo de Cálculo do Índice de Justiça .....	47
4.	Metodologia e AMBIENTE DE Testes.....	56
4.1.	Ambiente de Simulação .....	56
4.2.	Validação do Ambiente de Simulação .....	58
4.3.	Implementação da Política de Justiça em Placa de Desenvolvimento .....	63

5.	Resultados Obtidos.....	71
5.1.	Cenário 1 - Análise de Resultados.....	71
5.2.	Cenário 2 - Análise de Resultados.....	77
5.3.	Cenário 3 – Análise de Resultados.....	82
5.4.	Cenário 4 – Análise de Resultados.....	84
6.	CONCLUSÕES.....	89
7.	REFEFÊNCIAS.....	91
8.	ANEXO A – CÓDIGO DA PLATAFORMA DE POLÍTICA.....	94
8.1.	Trecho do programa ‘ <i>main.c</i> ’, utilizado pelo PIC32.....	94
9.	ANEXO B – CÓDIGO DA PLATAFORMA DE simulação.....	101
9.1.	Trecho do <i>script</i> utilizado pelo NS2. ....	101

## 1. INTRODUÇÃO

### 1.1. Contextualização do Problema

Nos últimos anos, com o avanço da tecnologia e a popularização dos equipamentos sem fio, as redes locais sem fio (*WLAN*) se tornaram uma opção viável e economicamente acessível. Essa popularização, associada aos preços cada vez mais atrativos dos equipamentos sem fio, praticamente permitem que qualquer ponto lógico com acesso à redes cabeadas também possa ser configurado para fornecer um meio de acesso sem fio. Isso traz, como consequência, um aumento significativo no número de redes sem fio disponíveis, seja na forma de redes privadas para uso residencial e corporativo, seja na forma de redes públicas para uso geral.

Em outras palavras, à medida que as redes sem fio se tornam mais populares e disponíveis, é criada uma demanda para dispositivos móveis com capacidade de acesso sem fio. Esses, por sua vez, viabilizam e incentivam a modificação dos hábitos de acesso e navegação dos consumidores, conforme disponibilizam, cada vez mais, recursos de interatividade e multimídia.

Dessa forma, as redes sem fio, que inicialmente eram utilizadas para gerenciamento de tráfego do tipo “Melhor Esforço” (*best-effort*), devem se tornar capazes de suportar e gerenciar, cada vez mais, aplicações de áudio e vídeo em tempo real. E para tornar as redes IEEE802.11 mais robustas e capazes de suprir as necessidades desse novo tipo de tráfego, multimídia, é necessário que essas redes sejam estudadas e que novas soluções sejam propostas. É nesse contexto que esse trabalho está inserido, conforme as justificativas apresentadas a seguir.

## 1.2. Justificativa Para o Desenvolvimento do Trabalho

As redes IEEE802.11 foram implementadas considerando que existem condições de igualdade de acesso ao meio para todas as STAs, garantidas através da utilização do método *DCF* (PRASAD, 2005). Dessa forma, todas as STAs possuem a mesma probabilidade de acesso ao meio, independentemente de sua taxa de transmissão e de sua relação sinal-ruído (*SNR*), que varia em função das características do meio físico.

Entretanto, deve-se ressaltar que tais redes apresentam, como característica, uma anomalia, caracterizada em (HEUSSE, 2003), através da qual é possível que uma STA com baixa taxa de transmissão venha a ocupar o canal por um grande período de tempo, fato que, conseqüentemente, pode penalizar outras STAs, presentes na rede, que eventualmente possuam taxas de transmissão maiores.

Por outro lado, para que as redes IEEE802.11 pudessem suprir as necessidades de conteúdo multimídia, foi criado o padrão IEEE802.11e (PRASAD, 2005)(IEEE802.11e, 2005)(IEEE802.11, 2007). Esse padrão implementa o conceito de Qualidade de Serviço (*QoS*), através da classificação dos diferentes tipos de tráfego em grupos distintos, com diferentes prioridades e diferentes oportunidades de acesso ao meio quando se consideram as STAs. Todavia, a definição de prioridades do IEEE802.11e está baseada apenas na classificação das camadas superiores, sem considerar as condições de sinal e de ruído da camada física.

Porém, ao priorizar o acesso ao meio por parte de uma STA com conteúdo classificado como sendo de maior prioridade pelas camadas superiores, mas com baixa taxa de transmissão (por exemplo, em função da distância entre a STA e o ponto de acesso - *AP*), a subcamada de controle de acesso ao meio (*MAC*) do padrão IEEE802.11e penaliza todas as outras STAs que, eventualmente, possuam maior capacidade de transmissão em função de uma elevada *SNR*, mas com conteúdo de menor prioridade. Dessa forma, é criada uma segunda anomalia (GUIRARDELLO, 2008).

Com base nessas considerações, admite-se, nesse trabalho, que uma política de qualidade de serviço deve levar em consideração não apenas o tipo de serviço (como estabelecido no padrão IEEE802.11e) mas, também, as características da camada física e os requisitos necessários para atendimento desse serviço. Dessa forma, a *MAC* pode reunir condições para gerenciar eficientemente o acesso ao meio e para garantir uma utilização mais eficiente do espectro, com o maior rendimento possível.

### **1.3. Objetivo do Trabalho**

Este trabalho propõe a criação de uma política de QoS baseada na implementação do padrão IEEE802.11e. Através da análise da taxa de conexão, do tipo de serviço e dos requisitos necessários para atendimento do serviço, a *MAC* pode interagir com cada *STA*, alterando suas taxas e limitando seu acesso a fim de promover a utilização mais eficiente dos recursos da rede. A política implementada nesse trabalho prevê também a avaliação da sua implementação real em microcontroladores, como os presentes em roteadores de rede, por exemplo.

### **1.4. Resultados Esperados**

Por meio da análise da qualidade e eficiência do serviço pretende-se verificar as possíveis vantagens da política de QoS proposta frente àquela implementada no padrão IEEE802.11e. Do ponto de vista da qualidade de serviço, espera-se comprovar que várias *STAs*, conectadas com diferentes taxas a um mesmo (*AP*), possam compartilhar o canal com um mínimo de impacto na qualidade do serviço.

Deseja-se, após a implementação da política, que uma *STA* que esteja com uma condição melhor de *SNR*, mas com um tipo de tráfego de menor prioridade não seja prejudicada por uma *STA* com *SNR* pior, mas com um tráfego de alta prioridade.

## 1.5. Delimitação da Pesquisa

Neste trabalho são empregadas simulações com o software NS2 (NS2, 2009) para avaliação de diferentes cenários de rede, com número variável de STAs. Também são demonstrados testes práticos de implementação da política proposta em uma placa microcontrolada. Em função do simulador NS2 não possuir uma biblioteca oficial que implemente a MAC IEEE802.11e e, também, pela grande complexidade associada à simulação da interação entre camadas neste software, os testes realizados foram restringidos a cenários para STAs operando exclusivamente no padrão IEEE802.11g, através do método *DCF*. Essa metodologia traz como principal vantagem a possibilidade de utilização de uma biblioteca nativa do software, que permite a manipulação dos parâmetros de taxa e janela de contenção das STAs. O método *PCF* não será avaliado neste trabalho, pois é de uso opcional; já o *DCF*, por ser obrigatório, é bastante difundido em equipamentos no mercado.

## 1.6. Organização da Dissertação

O presente capítulo dessa dissertação, denominado Introdução, apresenta a contextualização do trabalho e as justificativas para seu desenvolvimento. Também estão apresentados, de forma sucinta, os objetivos do trabalho, os resultados esperados e a delimitação da pesquisa. Os demais capítulos da dissertação estão organizados como se segue.

O Capítulo 2 apresenta uma visão geral das redes IEEE802.11, descrevendo taxas e modulações e mostrando sua relação com a SNR das STAs. Também são descritas as técnicas de disputa pelo acesso ao meio aplicadas pelas STAs. Ainda neste capítulo é citado o método *PCF* e o método *DCF* é descrito com maior detalhamento, evidenciando a anomalia que é consequência da justiça por ele imposta. Também são apresentadas as características das redes IEEE 802.11e, mostrando o método *HCCA* e detalhando o método *EDCA*. Uma segunda anomalia é mostrada. Finalmente, é apresentado o método *EDCA* com controle de acesso.

No Capítulo 3 é apresentada uma definição de justiça, evidenciando a necessidade de um único índice para comparações de cenários. Também é apresentada a proposta de *Política de Justiça*, através de dois métodos de cálculo dos índices de justiça na rede, e um fluxograma de seus respectivos algoritmos.

O Capítulo 4 apresenta o ambiente de simulação, detalhando seus componentes e características, e as simulações realizadas para validação desse ambiente. Também é descrita a placa microcontrolada utilizada para aplicação prática da política de justiça, evidenciando seu desempenho durante o cálculo dos índices de justiça em diferentes cenários.

No capítulo 5 são analisados os resultados obtidos com a aplicação da política. São apresentados gráficos e tabelas, detalhando as situações em que ocorreu ganho de desempenho e as situações nas quais a aplicação de uma política resultou em uma condição não favorável.

Finalmente, no Capítulo 6 são apresentadas as conclusões desse trabalho, bem como as propostas de aplicações e trabalhos futuros.

## 2. REDES SEM FIO

O termo “rede sem fio” está associado a um conjunto de STAs que configura uma rede de dispositivos que podem se comunicar e disponibilizar serviços na ausência de cabos físicos de comunicação. Em geral, podem ser classificadas em função da sua área de abrangência: redes pessoais (*WPAN*), redes locais (*WLAN*), redes metropolitanas (*WMAN*) e redes vastas (*WWAN*).

Neste capítulo, são apresentadas as principais características de redes sem fio IEEE802.11, em especial no que diz respeito às taxas de transmissão e à sua relação com o meio físico. Também são apresentadas diferenças e características de algumas implementações da subcamada de acesso ao meio (MAC IEEE802.11) e os problemas resultantes dessas implementações.

### 2.1. Taxas de transmissão

Em redes sem fio, a taxa de transmissão das STAs depende da *SNR*. A *SNR*, por sua vez, é resultante da interação de um grande número de variáveis dinâmicas inerentes à condição de operação da rede, tais como: a posição da STA, as características físicas do ambiente, as condições de propagação do sinal. As ações concatenadas dessas variáveis produzem variações aleatórias no sinal recebido, que sofrem desvanescimento (*fading*) ao longo do tempo (RAPPAPORT, 1996). Dessa forma, diferentemente do que ocorre em redes cabeadas convencionais, onde a maneira pela qual se dá o processo de comunicação pode ser considerada determinística, em redes sem fio essa mesma consideração não pode ser efetuada. Isso porque, em redes cabeadas, o desempenho é garantido (desde que a STA esteja conectada), enquanto que em redes sem fio o desempenho é resultante de uma série de fatores, tais como as características físicas do ambiente, o posicionamento das STAs, entre outras.

Em especial, a qualidade de um enlace de rádio utilizado para transmissão de sinais digitais pode ser medida pela taxa de erros de bits, (*BER*). Nesse

contexto, considera-se que quanto menor a *BER*, melhor é o enlace pois, a princípio, um enlace com baixa taxa de erros, quando comparado a outro, com taxa de erros maior, permite que um número maior de bits por unidade de tempo seja transmitido. Conseqüentemente, enlaces com menores taxas de erros viabilizam maiores taxas de conexão.

Por outro lado, em redes sem fio, a posição da *STA* com relação ao *AP* influencia diretamente as características da *SNR*. Conforme o nível do sinal diminui (em conseqüência da sua atenuação), o número de erros aumenta. Isso requer que os dispositivos da rede façam uso de outros tipos de modulação, que apresentam maior robustez e melhores características para operação com um menor nível de *SNR*, sendo assim mais adequadas para maiores índices de *BER*.

Conforme demonstrado em (ZYREN, 1998), para diferentes técnicas de modulação, utilizadas para enlaces de rádios, obtém-se diferentes valores de probabilidades de *BER*. Esses valores são resultantes da relação entre a energia necessária para transmissão de um *bit* (*E<sub>b</sub>*) e a densidade de ruído do espectro (*N<sub>0</sub>*). A Figura 1 apresenta as curvas de *BER* (ou *BE*) em função da relação *E<sub>b</sub>/N<sub>0</sub>*, para os seguintes tipos de modulação:

- *PSK* e *MSK*;
- *DBPSK* e *DQPSK*
- *Coherent OOK* e *OFSK*
- *Incoherent OOK* e *OFSK*

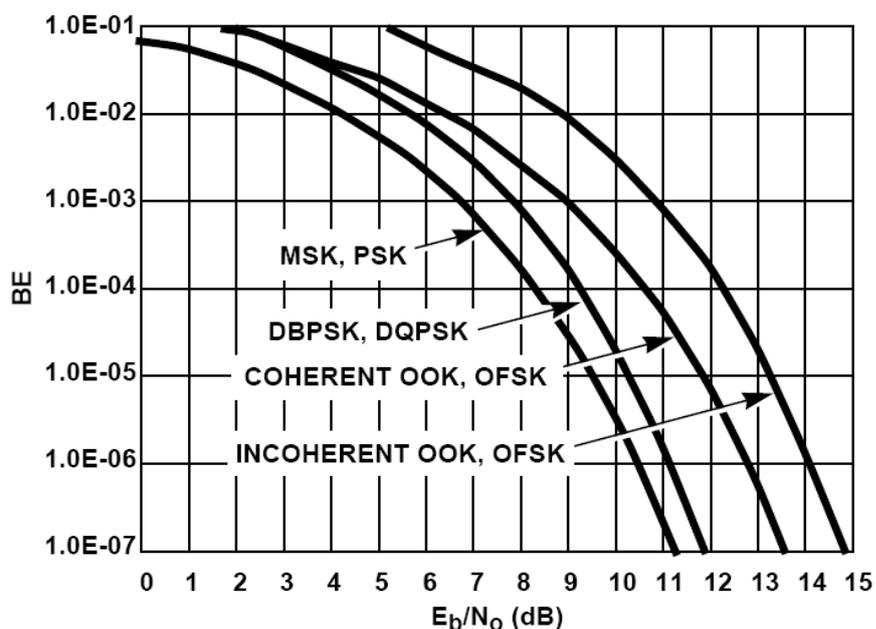


Figura 1. Probabilidade de erros para os métodos mais comuns de modulação

Fonte: (Zyren; Petrick, 1998)

O valor da *SNR* pode ser determinado através da equação:

$$SNR = \left(\frac{Eb}{No}\right) \times \left(\frac{R}{B_t}\right) \quad (1)$$

onde:

$E_b$  = Energia necessária para cada bit de informação (em joules)

$N_o$  = Ruído térmico em 1Hz de largura de banda (em W/Hz)

$R$  = Taxa de transmissão (em kbps)

$B_t$  = Largura de banda (em Hz)

De acordo com (GUIRARDELLO, 2008), o ruído é uma característica física do receptor e, por isso, está sempre presente e sempre deve ser considerado. O

ruído é definido por uma função densidade de probabilidade gaussiana com média nula e densidade espectral de potencia uniforme e, por isso, recebe o nome de *AWGN*. Por ser uma característica inerente ao receptor, este ruído é adicionado ao sinal no momento da recepção. Nas redes sem fio, na medida em que uma *STA* se distancia do *AP*, o nível do sinal recebido cai. Concomitantemente, o nível de ruído na recepção permanece constante. Dessa forma, quanto mais distante do *AP* uma *STA* estiver, menor será a sua *SNR*. Através da Equação 1, observa-se que, conforme a *SNR* de uma *STA* diminui, deve ocorrer uma diminuição proporcional no valor da relação  $E_b/N_0$ , considerando que a relação entre a taxa de transmissão  $R$  e a largura de banda  $B_t$  permaneça constante. Isso traz, como consequência direta, um aumento na taxa de erros, conforme já evidenciado Figura 1.

Uma rede pode utilizar diferentes mecanismos para compensar um aumento na taxa de erros; um deles é o aumento da potência do sinal transmitido. Porém, um aumento na potência de transmissão traz consequências financeiras, pois exige equipamentos maiores, com maior consumo de energia. Mais ainda, os níveis da potência de transmissão em redes sem fios são regulamentados na legislação e são diferentes em cada país. Portanto, o aumento da potência do sinal é um mecanismo com escopo de aplicação limitado. Um segundo mecanismo para compensação do aumento da taxa de erros é a mudança da frequência de operação. De forma análoga ao aumento da potência do sinal, as faixas de frequência passíveis de serem utilizadas são definidas em leis, o que também limita essas alterações.

Um terceiro mecanismo é a mudança do tipo de modulação. Através da Figura 1, pode-se verificar qual a melhor modulação para uma situação real, mantendo-se constante a relação  $E_b/N_0$  e visando a operação com a menor taxa de erros possível. Entretanto, é importante salientar que uma modulação mais robusta implica, necessariamente, em taxas de transmissão mais baixas.

O padrão IEEE 802.11 não possui mecanismos que permitam à *STA* transmissora saber qual a *SNR* da *STA receptora*. De forma semelhante, a *STA* transmissora não tem condições de saber a localização do receptor; não possui,

portanto, nenhuma informação que permita determinar a proximidade do receptor em relação ao transmissor. Porém, o padrão prevê que, para cada quadro recebido com sucesso, o receptor deve sinalizar seu recebimento (*ACK*). Dessa forma, através do gerenciamento dos *ACKs*, recebidos ou não, o transmissor tem condições de estabelecer uma métrica: a taxa de erros de quadros (*FER*). Em relação à *BER*, a *FER* pode ser determinada conforme descrito pela Equação 2 (GUIRARDELLO, 2008):

$$FER = 1 - (1 - BER)^N \quad (2)$$

onde: *N* é o número de bits do pacote. Conseqüentemente, a *BER* pode ser extraída da equação 2, a partir de um valor conhecido para a *FER*.

## 2.2. A MAC IEEE 802.11

O padrão IEEE 802.11 define dois métodos de acesso ao meio, através de funções de coordenação, conforme mostrado na Figura 2: *PCF* e *DCF*.

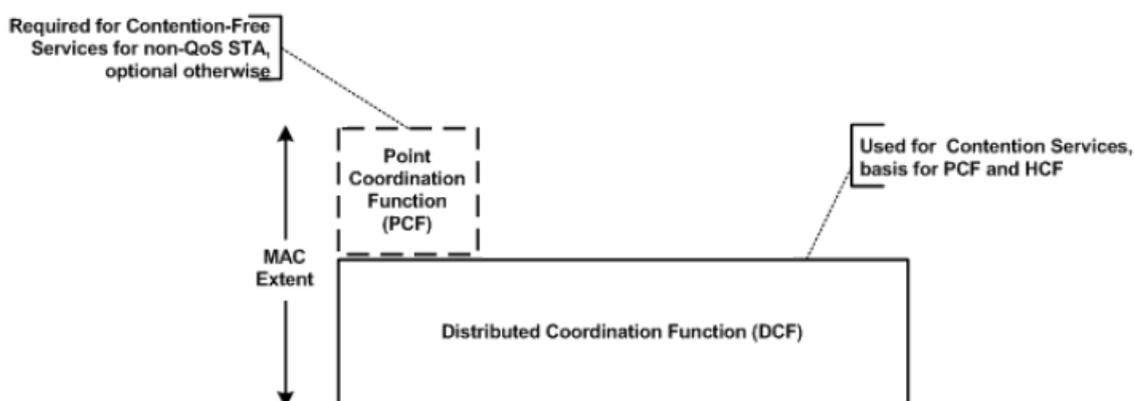


Figura 2. Arquitetura da Mac 802.11

Fonte: IEEE Std 802.11 – 2007 (adaptado)

O método *PCF* permite o acesso arbitrário das *STA* ao meio, através da utilização de um *PC* implementado no *AP*. Porém, esse método é de uso opcional, e não está implementado na grande maioria das *STAs*. Dessa forma, não será objeto de estudo nesta dissertação.

Por outro lado, o método *DCF* é obrigatório e, por isso, está presente em todas as *STAs* que trabalham no protocolo 802.11. O acesso ao meio no método *DCF* é feito através do mecanismo *CSMA/CA*. Nesse mecanismo, as *STAs* “ouvem” o meio antes de iniciar a disputa pelo canal, tentando evitar uma possível colisão de pacotes, caso o canal esteja ocupado. Porém, esse mecanismo não garante que as colisões não ocorrerão. Ele apenas tenta evitar, através de um sistema de contagem aleatória, que duas ou mais *STAs* tentem acessar o canal ao mesmo tempo.

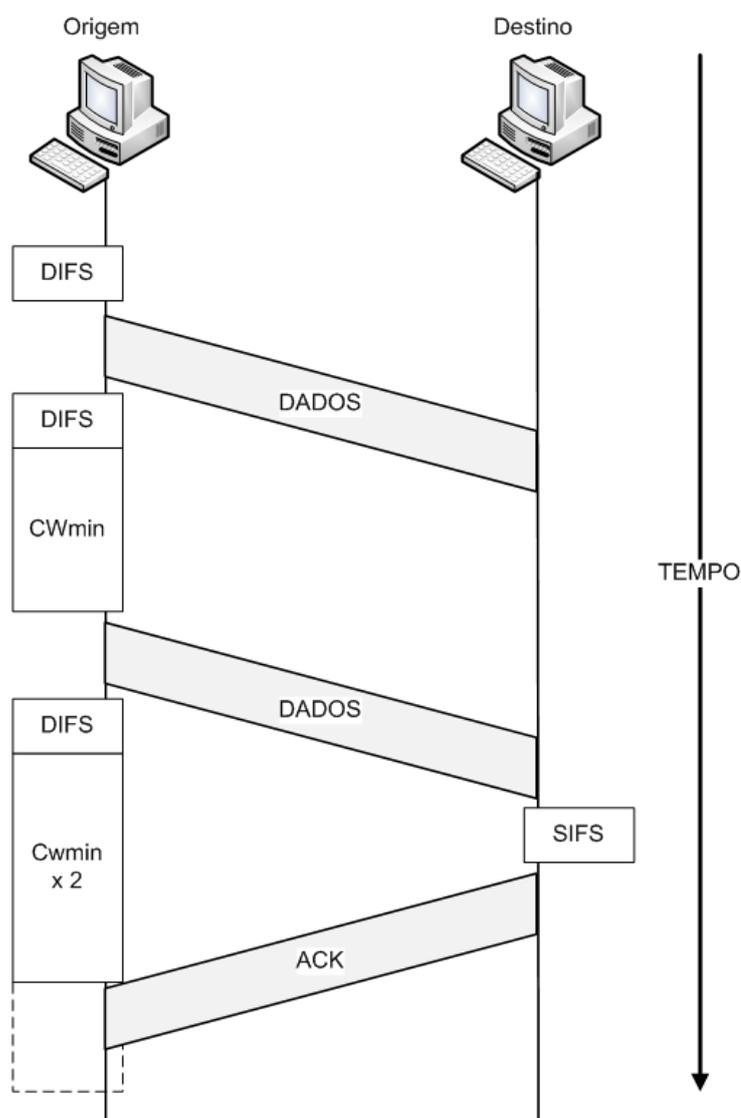
O método *DCF* prevê que, sempre que uma *STA* desejar acessar o canal pela primeira vez, ela deve “ouvir” esse canal até que o mesmo esteja livre. A partir desse momento, a *STA* deve aguardar um intervalo de tempo denominado *DIFS* e, se no final da contagem o meio ainda estiver livre, a *STA* poderá iniciar sua transmissão e passa a aguardar uma confirmação do receptor. Se o pacote transmitido pela *STA* for recebido corretamente, o receptor aguarda um intervalo de tempo denominado *SIFS* e envia um pacote *ACK*.

Se o pacote *ACK* não for recebido pela *STA* que iniciou a transmissão, a mesma considera que houve colisão e passa a aguardar um novo tempo *DIFS*, acrescido de um tempo aleatório. Esse tempo é chamado de Janela de Contenção (*CW*), e seu valor pode variar aleatoriamente, com distribuição uniforme, entre zero e um valor *CW<sub>max</sub>* na primeira transmissão. Uma vez definido na primeira transmissão, esse valor passa a ser chamado de *CW<sub>min</sub>*, e é decrementado em intervalos de tempo correspondentes a um *Slot Time* sempre que o canal estiver livre. Nas redes IEEE 802.11b, o valor do *Slot Time* é de 20 $\mu$ s.

Se, durante a contagem, o meio se tornar ocupado, a contagem é interrompida até que o meio esteja livre novamente. Ao final da contagem (quando *CW<sub>min</sub>* atinge o valor zero), a *STA* é liberada para transmitir.

Se, após a transmissão, a *STA* não receber o *ACK* de confirmação, ela assume que uma colisão ou perda de pacote pode ter ocorrido. Nesse caso, o valor de *CWmin* dobra, reiniciando a contagem. O valor máximo da contagem nunca ultrapassa *CWmax*. Esse procedimento de contagem é denominado *Backoff*.

A Figura 3 mostra a seqüência de transmissão de um pacote de dados no método *DCF*, utilizando o mecanismo *CSMA/CA*.



**Figura 3. Transmissão de pacotes no método *DCF*, com o mecanismo *CSMA/CA***

\* As caixas de tempo são apenas ilustrativas, e por isso estão fora de escala

A Figura 4 mostra o acesso ao meio realizado pelo método *DCF*, com o mecanismo *CSMA/CA*.

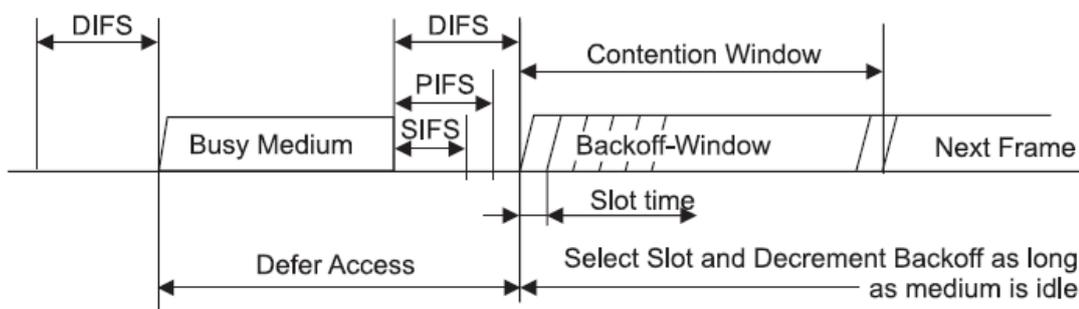


Figura 4. Acesso ao meio através do método *DCF* com *CSMA/CA* (IEEE Std 802.11 – 2007)

O método *DCF* não implementa nenhum tipo de prioridade no acesso ao canal. Todas as *STAs* possuem a mesma probabilidade de acesso, independentemente da taxa ou do tipo de pacotes. Dessa maneira, ao desconsiderar as características da camada física e proporcionar a mesma probabilidade de acesso para todas as *STAs*, o método *DCF* permite que *STAs* com menor taxa ocupem o canal por um período longo de tempo (quando se considera sua participação na rede em relação às demais *STAs*), penalizando, assim, as *STAs* com maior taxa, e diminuindo o desempenho total da rede. Esse efeito será descrito adiante, com maior detalhamento. Além disso, essa igualdade de condições na disputa pelo canal impossibilita a criação de mecanismos de garantia de serviço para aplicações sensíveis a atrasos e perdas de pacote (PRASAD, 2005).

O padrão IEEE 802.11 não define um limite para o número máximo de usuários que podem acessar o canal, simultaneamente, no método *DCF*. Porém, uma vez que esse método é baseado no mecanismo *CSMA/CA*, o atraso total da rede pode se tornar um problema, caso um número muito grande de *STAs* tente transmitir ao mesmo tempo.

Além disso, a *MAC* no padrão IEEE 802.11 não possui nenhum mecanismo para diferenciação dos tipos de tráfego e de serviços. Dessa forma, todos os dados são tratados da mesma forma, impossibilitando um tratamento diferenciado para fluxos de dados que apresentem requisitos especiais. Conseqüentemente, uma *STA* com dados de baixa prioridade pode, por exemplo, prejudicar uma *STA* trafegando vídeo (dados de alta prioridade).

### 1.1.1. A Anomalia da *MAC* IEEE802.11

O método *DCF*, através do mecanismo *CSMA/CA*, proporciona a todas as *STAs* a mesma probabilidade de acesso ao meio. Essa igualdade na probabilidade de acesso vem do fato de que todas as *STAs* seguem o mesmo procedimento na disputa pelo acesso ao meio, sem nenhuma forma de privilégio para nenhuma *STA*. Estatisticamente, espera-se que todas as *STAs* devem ganhar o canal um mesmo número de vezes, independentemente de suas condições de sinal e taxa de transmissão.

Nas redes IEEE 802.11, a taxa de transmissão é influenciada pela qualidade do enlace, que por sua vez é influenciada pelas características físicas do meio e pela posição da *STA*. Uma *STA* com baixa *SNR* e que possui, conseqüentemente, baixa taxa de transmissão, pode ocupar o canal por um período de tempo longo (em relação à qualidade do seu enlace), penalizando as outras *STAs* com melhores *SNR* e taxas de transmissão maiores. Essa característica é demonstrada como uma anomalia da *MAC* em (HEUSSE, 2003).

Em (BRANQUINHO, 2006) foi demonstrado que, devido a essa anomalia, em uma rede com duas *STAs*, com a mesma quantidade de dados para transmissão e conectadas a 11Mb/s e 1Mb/s, respectivamente, a taxa efetiva da primeira *STA* se tornará menor que 1Mb/s após um período suficiente de observação.

A título de ilustração, sejam consideradas duas *STAs* com a **mesma quantidade de dados** para transmissão e conectadas, respectivamente, a 11Mb/s e 1Mb/s; após um período suficiente de observação, em decorrência da

mesma probabilidade de acesso ao canal, a *STA* conectada a 1Mb/s terá ocupado o canal por um período de tempo 11 vezes maior que o período de tempo da *STA* conectada a 11Mb/s, para concluir satisfatoriamente o tráfego total das informações. Concomitantemente, considere-se também que, conforme demonstrado em (MORALES, 2008), a eficiência na transmissão de pacotes varia em função da taxa de conexão. Os valores de eficiência para uma rede IEEE 802.11b com preâmbulo curto são mostrados na Tabela 1.

Taxa de conexão (Mbps)	Eficiência (%)
11	59,3
5,5	69,6
2	74,3
1	76,9

**Tabela 1. Eficiência na taxa de transmissão 802.11b**

**Fonte: Morales, 2008**

Considere-se ainda, conforme descrito em (MORALES, 2008), que a capacidade de transmissão  $R_{Tn}$  de uma *STA* pode ser determinada pela equação:

$$R_{Tn} = \rho_n \cdot R_n \cdot C_n \quad (3)$$

onde:

$\rho_n$  = eficiência da transmissão (adimensional)

$R_n$  = taxa de transmissão (em Mbps ou kbps)

$C_n$  = Proporção do tempo de transmissão, considerando probabilidades de transmissão iguais para todas as *STAs*, e período de observação suficiente (adimensional).

Assim, através da aplicação da Equação 3 e considerando-se os critérios de eficiência expressos na Tabela 1, pode-se determinar a capacidade de

transmissão  $R_{T1}$ , para a *STA* conectada a 1 Mbps, e  $R_{T11}$ , para a *STA* conectada a 11 Mbps:

$$R_{T1} = 0,769 \cdot 1 \text{ Mbps} \cdot (11/12) = 704,9 \text{ kbps}$$

$$R_{T11} = 0,593 \cdot 11 \text{ Mbps} \cdot (1/12) = 543,6 \text{ kbps}$$

Portanto, nesse caso, a *STA* conectada a 11Mb/s tem um *throughput* menor que o da *STA* conectada a 1Mb/s, evidenciando os efeitos da anomalia da *MAC*.

### 2.3. A *MAC* IEEE 802.11e

Conforme apresentando anteriormente, a *MAC* do padrão IEEE 802.11 não possui mecanismos para diferenciação dos tipos de tráfego e serviços; assim, todos os dados trafegando na rede são tratados da mesma forma, o que impossibilita um tratamento diferenciado para fluxos de dados que apresentem requisitos especiais.

Para que as redes IEEE802.11 pudessem suprir as necessidades de conteúdo multimídia e em tempo real, o *IEEE* aprovou, em 2005, o padrão 802.11e (PRASAD, 2005) (IEEE802.11e, 2005)(IEEE802.11, 2007). Esse padrão, além de implementar todas as características da *MAC* 802.11 original, define a *HCF*, que, por sua vez, implementa dois novos métodos: o *EDCA* e o *HCCA*. Esses métodos possuem funcionalidades parecidas e correspondentes, respectivamente, aos métodos *DCF* e *PCF* do padrão IEEE 802.11.

A arquitetura da *MAC* IEEE 802.11e pode ser vista na Figura 5.

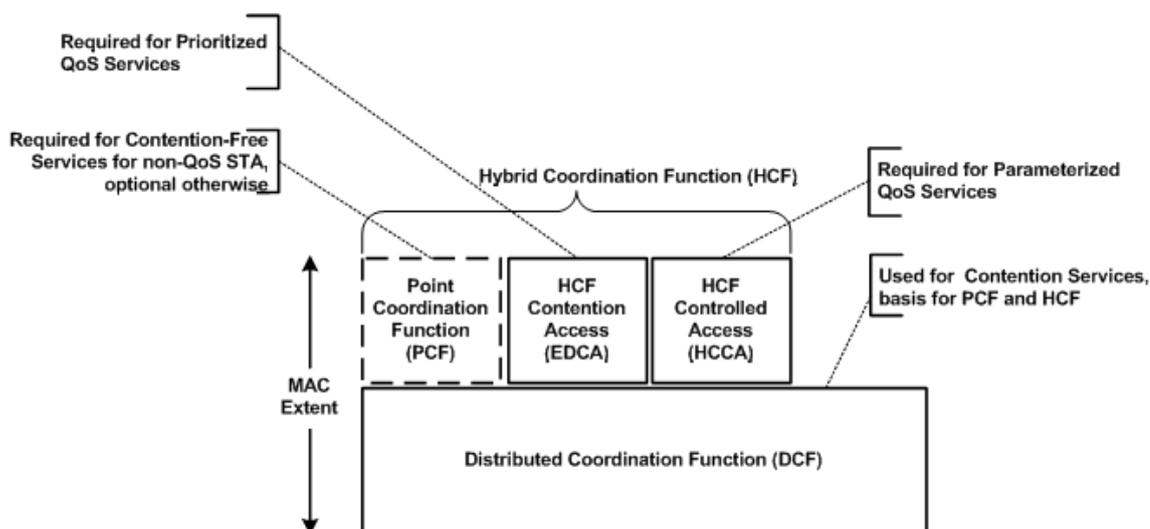


Figura 5. Arquitetura da Mac 802.11e

Fonte: IEEE Std 802.11 – 2007

O método *HCCA* não determina que as *STAs* disputem o acesso ao meio. Pelo contrário, nesse método o *AP* determina qual *STA* tem a oportunidade de acessar o canal, através da análise da priorização do tipo de serviço de cada *STA*. De forma semelhante ao método *PCF* para as redes IEEE802.11, o método *HCCA* não está implementado em todas as *STAs* IEEE802.11e, e, assim, também não será objeto de estudo nesse trabalho.

O método *EDCA*, ao contrário do *HCCA*, determina que as *STAs* disputem o acesso ao canal, assim como acontece com o método *DCF* da IEEE802.11. Entretanto, um *AP* que trabalha no método *EDCA* provê suporte às diferentes prioridades de serviços das *STAs*, através da implementação de categorias de acessos (*AC*). Cada *STA* pode gerenciar até quatro *ACs*, cada uma correspondendo a um determinado tipo de serviço, com suas próprias características e necessidades. As *ACs* são:

*AC\_VO* (Voz): Categoria com a prioridade mais alta. Corresponde a tráfegos de voz sobre pacotes, com intolerância a atrasos e *jitter*.

AC\_VI (VÍdeo): Categoria com prioridade mais baixa que a de voz, porém com prioridade mais alta sobre as demais. Corresponde a tráfegos de imagem sobre pacotes, sensíveis a atrasos e perturbações na transmissão.

AC\_BE (*Best Effort* – Melhor Esforço): Categoria com prioridade mais alta à categoria Background, mas com prioridade mais baixa com relação às demais categorias. Corresponde a tráfegos com garantia de entrega, mas tolerantes a atrasos e perturbações na transmissão. Os tráfegos dessa categoria possuem uma característica de rajadas, muito comum em tráfegos *HTTP*, por exemplo.

AC\_BK (*Background* – Segundo Plano): Categoria com a prioridade mais baixa. De forma semelhante à categoria *Best Effort*, corresponde a tráfegos com garantia de entrega, mas tolerantes a atrasos e perturbações. Porém, os tráfegos dessa categoria são constantes, com alta taxa de transferência. Um exemplo para essa categoria é o protocolo *FTP*.

Para cada categoria, é atribuído um valor de prioridade, e também um ou mais valores de prioridades de usuário (*UP*). O mapeamento das ACs e prioridades é mostrado na Figura 6.

Priority	UP (Same as 802.1D user priority)	802.1D designation	AC	Designation (informative)
Lowest ↓ Highest	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 6. Mapeamento de prioridades para as categorias de acesso

Fonte: IEEE Std 802.11 – 2007 (adaptado)

Assim, a estrutura interna de uma *STA*, com suas categorias de acesso é mostrada na Figura 7. Nessa figura, *AC\_VO*, *AC\_VI*, *AC\_BK* e *AC\_BE* correspondem às *ACs* apresentadas anteriormente; *UP* é o valor de prioridade de usuário; e *MSDU* são os dados úteis da camada *MAC*.

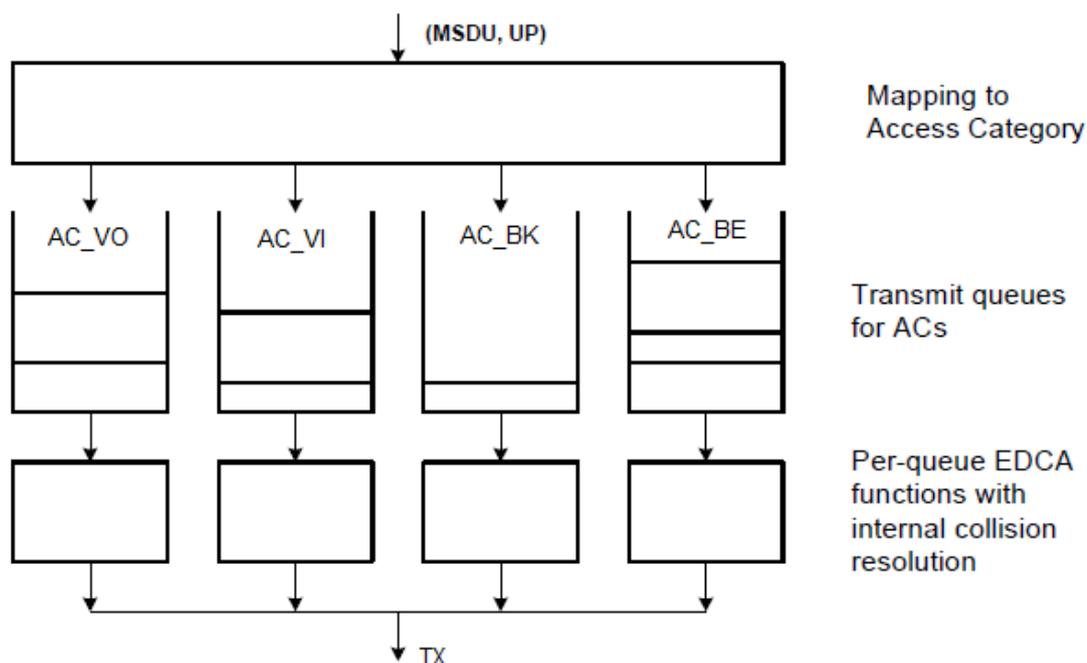


Figura 7. Modelo de implementação de uma *STA* no método *EDCA*

Fonte: IEEE Std 802.11 – 2007 (adaptado)

Cada uma das *ACs* é tratada de forma independente, contando com seu próprio gerenciamento de pacotes e filas de transmissão. Dessa forma, cada *AC* pode ser comparada a uma *STA* independente, com uma versão melhorada do método *DCF* (PRASAD, 2005). Isso ocorre porque as categorias devem sempre disputar o meio antes de iniciar uma transmissão, buscando conseguir uma oportunidade de transmissão (*TXOP*). O *TXOP* é um intervalo de tempo no qual uma determinada *STA* tem o direito de acessar o canal e transmitir seus dados.

O funcionamento do método *EDCA* é bastante parecido com o funcionamento do método *DCF*, descrito anteriormente. Porém, enquanto aquele

permitia uma igualdade de acesso a todas as *STAs*, este prioriza o acesso ao meio pelas *STAs* de maior interesse através da manipulação dos parâmetros de janela de contenção. Uma categoria com maior prioridade tem associada a si um valor menor de janela de contenção quando comparada com a janela de contenção das demais categorias. Isso faz com que a *AC* prioritária tenha uma vantagem competitiva na disputa pelo meio, aumentando sua probabilidade de acesso. Também de forma semelhante ao tempo *DIFS* do método *DCF*, o método *EDCA* implementa o tempo *AIFSN*. Esse intervalo de tempo tem seu valor no mínimo igual ao valor *DIFS*. Porém, esse valor pode ser aumentado individualmente para cada *AC*, forçando uma determinada categoria a esperar mais que a outra, antes de iniciar o procedimento de *backoff* e finalmente disputar o canal.

A Figura 8 mostra a relação entre os *IFSs* do IEEE802.11, e suas equivalências.

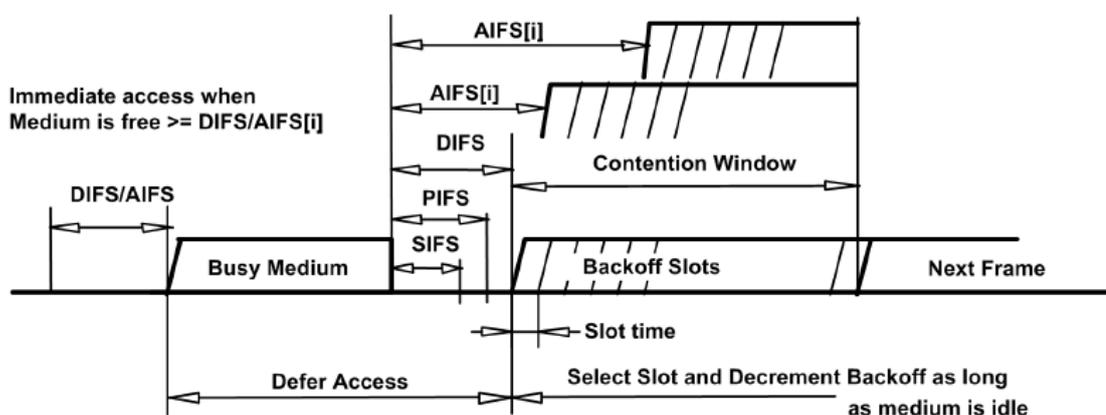


Figura 8. Relação entre os *IFSs* do IEEE802.11

Fonte: IEEE Std 802.11 – 2007

### 1.1.2. A Priorização da Anomalia

A implementação do QoS no IEEE802.11e é feita pela classificação e priorização dos pacotes unicamente pelo tipo de serviço, sem levar em consideração as condições de *SNR* das *STAs* que compõem a rede. Assim, ao levar em consideração apenas a classificação pelas camadas superiores e desconsiderar as características da camada física, o protocolo IEEE802.11e provoca um efeito que pode ser classificado como uma **priorização da anomalia MAC**, conforme demonstrado em (GUIRARDELLO, 2008). Através desse efeito, demonstra-se que uma *STA* que esteja próxima do *AP*, com uma excelente *SNR*, mas trafegando apenas pacotes *Best-Effort*, será preterida, em termos de ocupação do canal, por um *STA* mais distante e com uma *SNR* pior, mas com um tipo de serviço de prioridade mais alta. O efeito decorre do fato de que a segunda *STA* terá associada a si uma probabilidade de acesso maior, ocupando efetivamente o canal, em detrimento da primeira *STA*. Como exemplo desse efeito, em (RAMOS, 2005) descreve-se que uma *STA* com dados de alta prioridade (*AC\_VO*), mas com baixa *SNR*, pode prejudicar o *throughput* das demais *STAs* da rede, com maior *SNR*, mas com tráfego de menor prioridade.

Se *STAs* com igual probabilidade de acesso sofrem uma forte diminuição de suas taxas em função da anomalia da *MAC*, com o efeito adicional da priorização da anomalia as *STAs* com prioridade mais baixa podem ter seu acesso ao canal seriamente limitado (dependendo das condições físicas das *STAs* com prioridade mais alta), podendo, inclusive, não conseguir acesso ao canal.

Essa situação, sob o ponto de vista da distribuição de recursos, não pode ser caracterizada, neste trabalho, como uma **situação justa**, uma vez que *STAs* com excelentes condições de *SNR* podem ter seu acesso bloqueado, em função de outras *STAs* com condições físicas insatisfatórias, mas que, devido a seu tráfego priorizado, são privilegiadas na disputa pelo canal. Dessa forma, essas *STAs* com prioridade mais alta têm sua taxa limitada por suas péssimas condições de canal, mas utilizam grande parte dos recursos da rede.

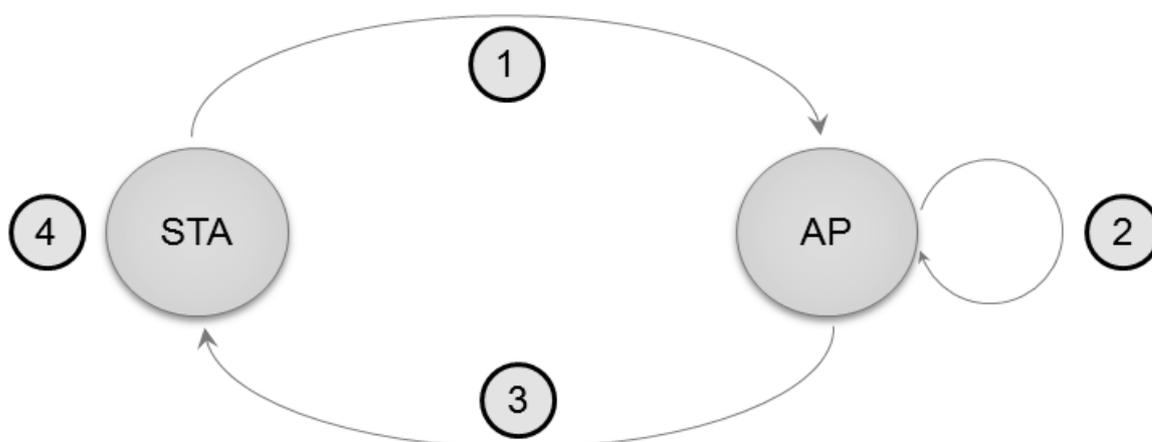
Em uma situação “mais justa”, essas *STAs* deveriam ter uma menor alocação de recursos da rede, considerando suas limitações de canal. Assim, liberariam uma maior quantidade de recursos para as outras *STAs* em melhores condições na rede. Nesse sentido, se as *STAs* pudessem informar ao *AP* suas necessidades de taxa para atendimento de seus serviços (incluindo alternativas para atendimento mínimo em caso de problemas de canal), o próprio *AP* teria condições de gerenciar os recursos disponíveis de forma a distribuí-los entre todas as *STAs*, segundo uma política de justiça pré-estabelecida. Esse cenário pode ser conseguido através da implementação de um controle de acesso para o método *EDCA*, conforme descrito no item a seguir.

### **1.1.3. EDCA com controle de admissão**

Uma alternativa para solução dos problemas causados pelas anomalias descritas anteriormente é a utilização de um mecanismo de controle de admissão para o método *EDCA*. Nas redes IEEE802.11e, o controle de admissão é mandatório para o método *HCCA*, mas opcional para o método *EDCA*.

Através desse mecanismo, o *AP* pode informar às *STAs* que trabalha com controle de admissão, e, assim, que exige dessas *STAs* uma lista de parâmetros para viabilizar o atendimento de seus serviços.

A Figura 9 representa a sequência de funcionamento do método *EDCA* com controle de admissão.



**Figura 9. Sequência de funcionamento do método *EDCA* com controle de admissão**

Inicialmente, conforme ilustrado no ítem 1 da Figura 9, cada *STA* deve enviar seus requisitos de taxa para o *AP* – incluindo-se aí taxas alternativas para atendimento mínimo de seus serviços.

Assim, de posse dos requisitos de serviço de cada *STA* e baseado nas condições físicas atuais da rede, o *AP* reúne as condições necessárias para determinar o cenário “mais justo” para a distribuição de recursos na rede, uma vez que agora possui informações relevantes sobre cada *STA* da rede para essa tomada de decisão. Esse procedimento está representado pelo ítem número 2, na Figura 9.

Finalmente, conforme representado pelo ítem 3 da Figura 9, o *AP* informa, a cada *STA*, qual das opções fornecidas por ela estará melhor adequada à condição atual da rede e que, por esse motivo, deverá ser utilizada.

Conhecidos os melhores valores para atendimento de seu serviço ou, ainda, a impossibilidade de utilização de quaisquer dos valores fornecidos, cada *STA* determinará se disputará ou não o canal. Esse procedimento está representado pelo ítem 4 da Figura 9.

Nesse cenário, as *STAs* continuam utilizando o método *EDCA* e ainda disputam o canal com a probabilidade de acesso determinada pelas prioridades

de seus fluxos. Porém, o nível de justiça na rede será maior já que algumas STAs devem abrir mão de parte de seus recursos, em favorecimento de uma distribuição mais justa entre todas as STAs, segundo uma política específica implementada no roteador da rede. A política proposta neste trabalho, para garantir essa distribuição mais justa de recursos, está explicitada no capítulo que se segue.

### 3. PROPOSTA DE POLÍTICA DE QUALIDADE DE SERVIÇO - QOS

#### 3.1. Definição do Conceito de Justiça

Em sistemas distribuídos, uma quantidade de recursos de rede, limitada, deve ser distribuída entre todos os dispositivos que fazem parte do sistema. Nessa condição, a maneira pela qual tais recursos são distribuídos tem um impacto direto no desempenho de cada dispositivo e também no desempenho do próprio sistema. Dessa forma, é sempre desejável que esses recursos sejam distribuídos da forma mais justa possível.

Segundo (JAIN, 1984), *“justiça não significa necessariamente uma distribuição igual de recursos. Em alguns casos é justificável que se dê mais recursos a determinados consumidores que a outros”*. Dessa forma, para que se possa classificar um determinado sistema como justo, ou para que se possa comparar dois ou mais cenários, segundo um critério de justiça, é necessário que se defina claramente o que significa justiça para aquele sistema.

Neste trabalho, o conceito de justiça é definido através de um sistema de pontuações. Um índice de justiça é definido para cada configuração do sistema, como o total de pontos obtidos através de um cálculo da justiça. Uma determinada configuração é mais justa que outra se obtiver um índice de justiça maior. Dessa maneira, torna-se possível definir e identificar quais são os cenários mais justos e quais são os menos justos, para uma determinada rede, através de uma comparação entre os totais de pontos obtidos por cada configuração possível para o sistema. As definições do sistema de pontuações, do índice de justiça e do cálculo desse índice estão descritas nos itens que se seguem.

### 3.2. A Política de QoS

A igualdade na probabilidade de acesso implementada pelo método *DCF* da IEEE802.11 traz como conseqüência uma anomalia, explicitada no capítulo 2, que faz com que a taxa média de tráfego de todas as STAs seja limitada pela taxa da pior STA (HEUSSE, 2003). Essa situação é agravada pelo método *EDCA* da IEEE802.11e, uma vez que, para cada tipo de serviço, existe uma diferente prioridade e, ao priorizar uma STA com baixa *SNR*, a taxa média da rede diminui ainda mais, priorizando a anomalia (GUIRARDELLO, 2008).

Portanto, uma STA com uma baixa *SNR* diminui o tráfego médio de todas as STAs que disputam os recursos fornecidos por um mesmo AP. Do ponto de vista do provedor de serviços, essa situação traz como conseqüência um prejuízo financeiro, já que a rede, inicialmente dimensionada para prover serviços a uma determinada taxa, estará sendo subutilizada.

A taxa da conexão entre uma STA e o AP é função da *SNR* do sinal recebido pela STA. Portanto, uma STA que, devido à sua localização, ou devido às características do ambiente, esteja com baixa *SNR*, terá uma taxa de conexão menor que uma STA com características físicas mais privilegiadas e com uma melhor *SNR*. Porém, a implementação da IEEE 802.11e não considera essas características físicas na priorização do acesso ao meio. Para isso, considera apenas o tipo de tráfego da STA. Entretanto, admite-se neste trabalho que uma proposta de Política de Qualidade de Serviço deve levar em consideração não apenas o tipo do serviço, mas também as características físicas das STAs.

A proposta de política apresentada neste trabalho se baseia no fato de que é possível utilizar os recursos da rede de forma mais eficiente, sem que para isso seja necessária a alteração nas prioridades dos serviços das STAs. Essa política está prevista para possibilitar sua implementação em *hardware* através de um método computacional: o algoritmo de justiça. O algoritmo de justiça busca configurar automaticamente os dispositivos, de forma a obter a situação de maior justiça na rede, segundo um critério de justiça previamente estabelecido. Para que isso seja possível, cada STA deve informar ao AP as taxas que permitem o

atendimento de seus serviços. Considerando esses valores, o número de *STAs* que disputam o meio e a taxa de conexão de cada *STA*, o *AP* tem condições de executar um algoritmo para o cálculo de pontuações e pode definir, dentre todas as combinações possíveis de taxas de serviço, aquela que resulta num melhor valor do índice de justiça.

É desejável, após a implementação da política, que uma *STA* que esteja com uma condição melhor de *SNR*, mas com um tipo de tráfego de menor prioridade, tenha suas restrições em relação aos recursos da rede minimizadas, em relação à presença de uma *STA* com pior *SNR*, mas com um tráfego de maior prioridade. Por conseguinte, esta última, que em função de suas características físicas se conecta com uma taxa inferior, deverá ter seu acesso ao meio limitado, mas de tal forma que faça com que seu serviço também seja satisfatoriamente atendido, permitindo assim um maior *throughput* a *STAs* que, de outra maneira, não teriam essa oportunidade.

### 3.3. Definição e Cálculo do Índice de Justiça

Para o cálculo do índice de justiça das *STAs* sem fio, três fatores são considerados: a taxa de conexão entre a *STA* e o *AP*, o tipo de serviço e a taxa do serviço, chamadas respectivamente de *conRate*, *servType* e *servRate*. O cálculo do índice de justiça (*Jindex*) é realizado através da seguinte equação:

$$Jindex = \sum_{i=1}^N (K_{conRate})_i \cdot (K_{servType})_i \cdot (K_{servRate})_i \quad (4)$$

onde:

$(K_{conRate})_i$  é o fator multiplicativo para a taxa de conexão para a *STA* *i*

$(K_{servType})_i$  é o fator multiplicativo para o tipo de serviço para a *STA* *i*

$(K_{servRate})_i$  é o fator multiplicativo para a taxa de serviço para a *STA* *i*

*N* é o número de *STAs* na rede

O valor do índice de justiça é calculado em função de valores especificamente selecionados para as variáveis da Equação 4. É de interesse da política de justiça proposta neste trabalho que a precedência entre as prioridades definidas pelos tipos de serviço sejam mantidas. Portanto, os valores assumidos pelas variáveis definidas na Equação 4 devem refletir esse interesse. Os valores adotados neste trabalho foram obtidos empiricamente após a realização de um conjunto de simulações em redes hipotéticas, que permitiram a verificação e a análise do funcionamento da política. Os valores utilizados para os fatores  $K_{conRate}$ ,  $K_{servType}$  e  $K_{servRate}$  estão descritos nas tabelas Tabela 2, Tabela 3 e Tabela 4 respectivamente.

Taxa de conexão [bits/s]	Valor de $K_{conRate}$
1M	100
2M	200
5,5M	300
11M	400

**Tabela 2. Valores utilizados para o fator  $K_{conRate}$**

Tipo de serviço	Valor de $K_{servType}$
Background	110
Best Effort	210
Video	310
Voice	410

**Tabela 3. Valores utilizados para o fator  $K_{servType}$**

Taxa de serviço [bits/s]	Valor de $K_{servRate}$
64k	1
100k	2
128k	3
200k	4
250k	5
256k	6
300k	7
350k	8
400k	9
500k	10
512k	11
700k	12
750k	13
1M	14
2M	15
5,5M	16
11M	17

**Tabela 4. Valores utilizados para a o fator  $K_{servRate}$**

Os valores determinados para os fatores  $K_{conRate}$ ,  $K_{servType}$  e  $K_{servRate}$  foram definidos de forma que pudessem refletir a natureza da política de justiça proposta, preservando o fato de que as STAs devem estar priorizadas em diferentes níveis durante o seu atendimento. Dessa forma, o valor do fator  $K_{servType}$  considerado para o tipo de serviço Background, por exemplo, deve ser menor que o valor do tipo de serviço Voz, e este deve ser o maior de todos os tipos de serviço. Raciocínio análogo foi utilizado na seleção dos valores do fator  $K_{conRate}$ , onde taxas de conexões maiores estão associadas a valores maiores para o fator.

A ordem de grandeza dos valores utilizados para os fatores  $K_{conRate}$  e  $K_{servType}$  foram propositalmente definidas como iguais. Isso porque a proposta de política de justiça estabelece que as taxas de conexão são definidas em função das características físicas das STAs, e são tão importantes, na definição de prioridades, quanto o tipo de serviço trafegado. Essa metodologia amplia aquela aplicada nas redes IEEE802.11e, onde as priorizações não consideram as

características físicas das *STAs*. Porém, os valores utilizados para os fatores  $K_{servType}$  foram estabelecidos com um acréscimo de 10 unidades em relação aos valores de  $K_{conRate}$ , de forma a respeitar a prioridade original do tipo de serviço.

Por fim, os valores de  $K_{servRate}$  foram definidos com a menor ordem de grandeza. Esses valores aumentam conforme aumentam as taxas do tipo de serviço, mas não possuem tanto peso quanto os outros dois fatores no cálculo do índice de justiça. Esse processo de ajuste de valores foi refinado para contemplar ordens de grandeza que pudessem produzir resultados coerentes na indicação do cenário mais justo e que, ao mesmo tempo, pudessem ser trabalhadas de forma eficiente quando da implementação real em microcontroladores de 32 bits.

Como exemplo de cálculo do índice de justiça, considere-se uma rede com três *STAs*, *STA1*, *STA2* e *STA3*, conectadas, respectivamente, a 11Mb/s, 5,5Mb/s e 1Mb/s e com taxas de serviços, respectivamente, iguais a 1Mb/s, 700kb/s e 256kb/s. A *STA1* trafega pacotes do tipo *BestEffort*, *STA2* trafega pacotes de Vídeo e a *STA3* trafega pacotes de Voz.

Aplicando-se a Equação 4 à combinação apresentada, tem-se que o índice de justiça é igual a:

Pontos da *STA1*:

$$(K_{conRate})_1 \cdot (K_{servType})_1 \cdot (K_{servRate})_1 = (210 \cdot 400 \cdot 14) = 1176000 \text{ pontos}$$

Pontos da *STA2*:

$$(K_{conRate})_2 \cdot (K_{servType})_2 \cdot (K_{servRate})_2 = (310 \cdot 300 \cdot 12) = 1116000 \text{ pontos}$$

Pontos da *STA3*:

$$(K_{conRate})_3 \cdot (K_{servType})_3 \cdot (K_{servRate})_3 = (410 \cdot 100 \cdot 6) = 246000 \text{ pontos}$$

Valor total do índice de justiça Jindex:

$$\sum_{i=1}^N (K_{conRate})_i \cdot (K_{servType})_i \cdot (K_{servRate})_i = 1176000 + 1116000 + 246000$$

Jindex = 2538000 pontos

### 3.4. Algoritmo de Cálculo do Índice de Justiça

O algoritmo para cálculo do Índice de Justiça foi desenvolvido com o objetivo de ser executado eficientemente em microcontroladores de 32 bits, como aqueles presentes atualmente no interior de um *AP* em um ambiente real.

Em condições de operação, admite-se que o *AP* deve receber a solicitação de acesso ao canal de uma nova *STA*, que por sua vez deve informar ao *AP* suas opções de taxas para o atendimento de seu serviço. De posse de informações sobre o número de *STAs* que desejam disputar o canal, a taxa de conexão de cada *STA* e os valores das taxas de serviço de cada *STA*, o *AP* pode então executar o algoritmo de cálculo do índice de justiça, conforme descrito a seguir.

#### 1.1.4. Cálculo do Índice de Justiça pelo método da limitação do acesso pelo tempo de utilização do canal (MLTC)

Inicialmente, todas as variáveis de controle presentes na memória do microprocessador são zeradas. A partir desse momento, o processador inicia um ciclo de repetição utilizando os valores passados pelas *STAs*, testando a factibilidade de cada uma das possíveis combinações para atribuição de taxas de serviço às *STAs* da rede, e, em seguida, calculando o índice de justiça para as combinações viáveis.

Assim sendo, em cada combinação, antes de realizar o cálculo de justiça, o processador verifica se a mesma está em uma condição fisicamente possível para

o atendimento das requisições das *STAs*. Para isso, calcula o tempo que deveria ser alocado a cada *STA* para que a mesma acessar o canal de forma a ter sua taxa de serviço atendida, e, a seguir, determina a somatória desses tempos para todas as *STAs* na rede. Se a soma total ultrapassar 1 segundo, a combinação é invalidada. Do contrário, o processador procede com o cálculo do índice de justiça para a combinação.

Por exemplo, considere-se uma rede com três *STAs*, denominadas *STA1*, *STA2* e *STA3*, conectadas respectivamente com taxas iguais a 5,5Mb/s, 2Mb/s e 1Mb/s, todas com taxas de serviço de 1Mb/s. Para que a combinação seja classificada como válida ou não, é preciso que seja conhecido o tempo necessário de acesso ao canal para cada *STA* ( $T_a$ ). Para isso é utilizada a Equação 5.

$$T_a = N_s / R_c \quad (5)$$

onde:

$N_s$  = número de *bits* trafegados por segundo [bits/s]

$R_c$  = taxa de conexão da *STA* [bits/s]

Aplicando-se a Equação 5 ao exemplo proposto, obtém-se os seguintes tempos de acesso:

$T_a$  para a *STA1* = 18%

$T_a$  para a *STA2* = 50%

$T_a$  para a *STA3* = 100%

Ou seja, a cada segundo, para atender ao seu serviço, a *STA1* deverá ocupar o canal por 0,18 segundos (18% do tempo); a *STA2*, por 0,5 segundos (50% do tempo); e a *STA3*, por 1 segundo (100% do tempo).

Dessa forma, para que os serviços de todas as *STAs* pudessem ser atendidos, seria necessário, a cada segundo, alocar às mesmas um tempo total de 1,68s, o que é fisicamente impossível. Essa impossibilidade fica evidente

considerando-se as condições da *STA3*. Para que seu serviço seja atendido é preciso que ela trafegue 1Mb de dados a cada segundo. Considerando-se que a taxa de conexão dessa *STA* é de 1Mb/s, para que seu serviço seja satisfeito ela deve ocupar o canal por 100% do tempo, o que impossibilita que outras *STAs* tenham condições físicas de acesso ao canal.

Sempre que um cálculo é finalizado, o processador verifica se o valor do índice calculado é o melhor dentre todas as configurações já analisadas e, em caso positivo, armazena esse valor e também o número da combinação atual em duas variáveis de controle; a partir daí, reinicia o ciclo com uma combinação diferente de taxas de serviço para as *STAs* da rede.

Ao final do processo, todas as combinações terão sido testadas, e o algoritmo é capaz de fornecer as características do cenário mais justo, que possui a maior pontuação. Esse algoritmo é denominado, nesse trabalho, de algoritmo do método da limitação do acesso pelo tempo de utilização do canal (MLTC). O fluxograma do algoritmo de cálculo, pelo método MLTC, está apresentado na Figura 10.

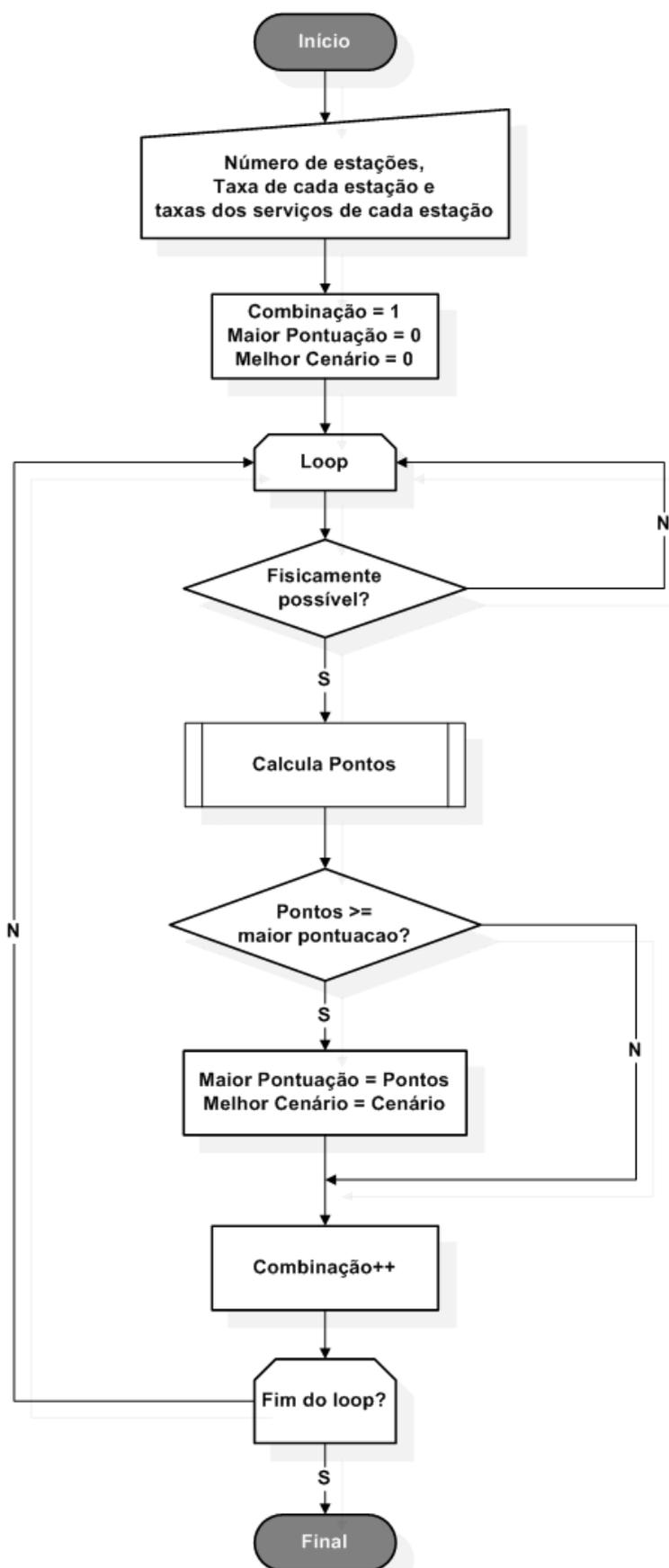


Figura 10. Fluxograma do algoritmo de cálculo do índice de justiça usando o MLTC

### 1.1.5. Cálculo do Índice de Justiça pelo método da limitação de taxas de serviço (MLTS)

O MLTC, porém, pode considerar todas as possíveis combinações de taxas de serviço em uma rede como fisicamente inactiváveis em situações onde todas as STAs requisitem pelo menos uma das taxas de serviço maior que a sua própria taxa de conexão; assim, nessas condições, não é capaz de calcular a combinação mais justa.

Isso acontece porque, no MLTC, o cálculo da justiça é realizado utilizando apenas os valores teóricos de taxa de serviço, desconsiderando as possíveis inconsistências físicas da conexão. Por exemplo, uma STA conectada a 1Mb/s claramente não tem condições físicas de trafegar serviços a 11Mb/s; entretanto, o MLTC, ao considerar os valores teóricos originalmente propostos, invalida a combinação. Por outro lado, deve-se considerar também que a taxa de conexão é função de características físicas do ambiente, e varia sem que a camada de aplicação tenha qualquer informação ou interferência. Dessa forma, é possível que uma STA informe que necessita de determinada taxa, para seu serviço, maior que sua própria taxa de conexão e, nesses casos, a STA não poderia ser impedida de acessar o canal.

A fim de determinar uma solução para o problema, foi realizada uma simulação com o software NS2 em uma rede com três STAs (STA1, STA2 e STA3), conectadas respectivamente a 1Mb/s, 5,5Mb/s e 11Mb/s, em duas diferentes combinações. A combinação (A) considera que a STA1 requisita tráfego a uma taxa de 11Mb/s, STA2 requisita tráfego a uma taxa de 1Mb/s e STA3 requisita tráfego a uma taxa de 512Kb/s. Já a combinação (B) apresenta as mesmas requisições de tráfego com exceção da requisição da STA1, que nessa combinação, requer uma taxa de 1Mb/s.

Através da simulação, na combinação (A), observou-se que as taxas de serviço efetivamente alocadas para as STA1, STA2 e STA3 foram, respectivamente, 55200 bytes/s, 127696 bytes/s e 65412 bytes/s.

Já na combinação (B), também utilizando o software NS2, observou-se que as taxas de serviço efetivamente alocadas para as *STA1*, *STA2* e *STA3* foram mantidas idênticas às da combinação (A). A Tabela 5 sintetiza os resultados da simulação para as duas combinações.

Comb.	<i>STA1</i>	<i>STA2</i>	<i>STA3</i>	<i>STA1</i>	<i>STA2</i>	<i>STA3</i>	média da soma das taxas das STAs (bytes)	Índice de justiça calc.
	taxa de conexão			taxa média				
	1Mb/s	5.5Mb/s	11Mb/s	(bytes/s)	(bytes/s)	(bytes/s)		
A	11	1	0,512	55200	127696	65412	248308	3400000
B	1	1	0,512	55200	127696	65412	248308	3400000

**Tabela 5. Resultados observados durante o desenvolvimento da política**

Através desses dados, pode-se observar que a *STA1* está conectada a uma taxa de 1Mb/s. Ao tentar trafegar com uma taxa de 11Mb/s (correspondente à combinação A), sua taxa efetivamente alocada foi de 55200 bytes/s. Por outro lado, ao tentar trafegar seus dados com uma taxa de 1Mb/s, recebeu também dados na mesma taxa. O mesmo efeito pode ser observado com as outras duas STAs. Isso ocorre porque a taxa de conexão é o fator limitante para o tráfego dos pacotes. Dessa forma, todas as taxas maiores que a própria taxa de conexão serão sempre limitadas pela própria taxa de conexão. Assim sendo, sem prejuízo do que realmente ocorre em conexões reais, nos casos onde a *STA* solicita uma taxa de serviço maior que a taxa de conexão, pode-se considerar, para efeito de pontuação, o valor da taxa de serviço ao invés do valor da taxa de conexão.

Com base nessas considerações, o algoritmo para o MLTC foi modificado, dando origem a um segundo algoritmo de cálculo, que foi chamado de método da limitação de taxas de serviço (MLTS). Esse novo algoritmo calcula o índice de justiça para todas as combinações, porém o faz ajustando os valores das taxas de serviços das STAs quando esses ultrapassam os valores da taxa de conexão da *STA*. Portanto, ao invés de descartar uma combinação fisicamente impossível,

esse método utiliza uma aproximação para determinar um possível valor do índice de justiça.

Como exemplo de cálculo do índice de justiça pelo MLTS, considere-se uma rede com três *STAs* (*STA1*, *STA2* e *STA3*) conectadas respectivamente a 1Mb/s, 5,5Mb/s e 11Mb/s e com taxas de serviços respectivamente iguais a 11Mb/s, 1Mb/s e 512kb/s. A *STA1* trafega pacotes do tipo *BestEffort*, *STA2* trafega pacotes de Vídeo e a *STA3* trafega pacotes de Voz.

Aplicando-se o método MLTS às três *STAs*, tem-se que a *STA1* possui uma taxa de serviço maior que sua própria taxa de conexão. Dessa forma, o valor de taxa de serviço a ser utilizado para o cálculo será o próprio valor da taxa de conexão.

Assim, aplicando-se a Equação 4 à combinação apresentada, tem-se que o índice de justiça é igual a:

Pontos da *STA1*:

$$(K_{conRate})_1 \cdot (K_{servType})_1 \cdot (K_{servRate})_1 = (210 \cdot 100 \cdot 14) = 294000 \text{ pontos}$$

Pontos da *STA2*:

$$(K_{conRate})_2 \cdot (K_{servType})_2 \cdot (K_{servRate})_2 = (310 \cdot 300 \cdot 14) = 1302000 \text{ pontos}$$

Pontos da *STA3*:

$$(K_{conRate})_3 \cdot (K_{servType})_3 \cdot (K_{servRate})_3 = (410 \cdot 400 \cdot 11) = 1804000 \text{ pontos}$$

Valor total do índice de justiça Jindex:

$$\sum_{i=1}^N (K_{conRate})_i \cdot (K_{servType})_i \cdot (K_{servRate})_i = 294000 + 1302000 + 1804000$$

Jindex = 3400000 pontos

Dessa forma, o algoritmo para cálculo deve ser alterado para calcular o índice de justiça para todas as combinações utilizando os dois métodos (MLTC e MLTS), e excluir as combinações descartadas pelo método MLTC antes de apresentar o resultado. Se, no final, todas as combinações forem descartadas pelo método MLTC, a combinação com maior índice de justiça segundo o método MLTS será a utilizada. O fluxograma do novo algoritmo proposto é apresentado na Figura 11.

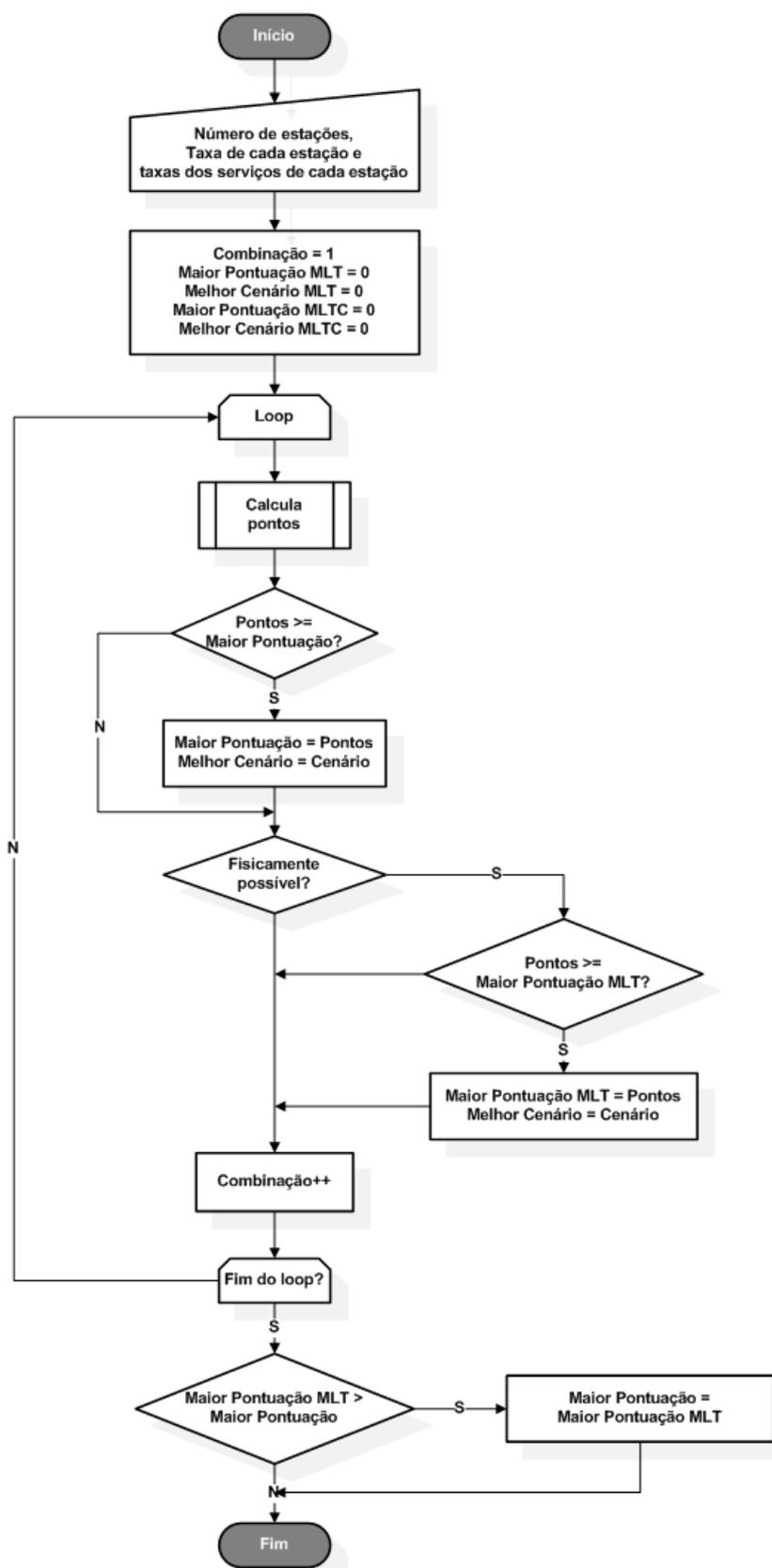


Figura 11. Fluxograma do algoritmo final de cálculo do índice de justiça usando MLTC e MLTS

O fluxograma simplificado do algoritmo completo é mostrado na Figura 12.

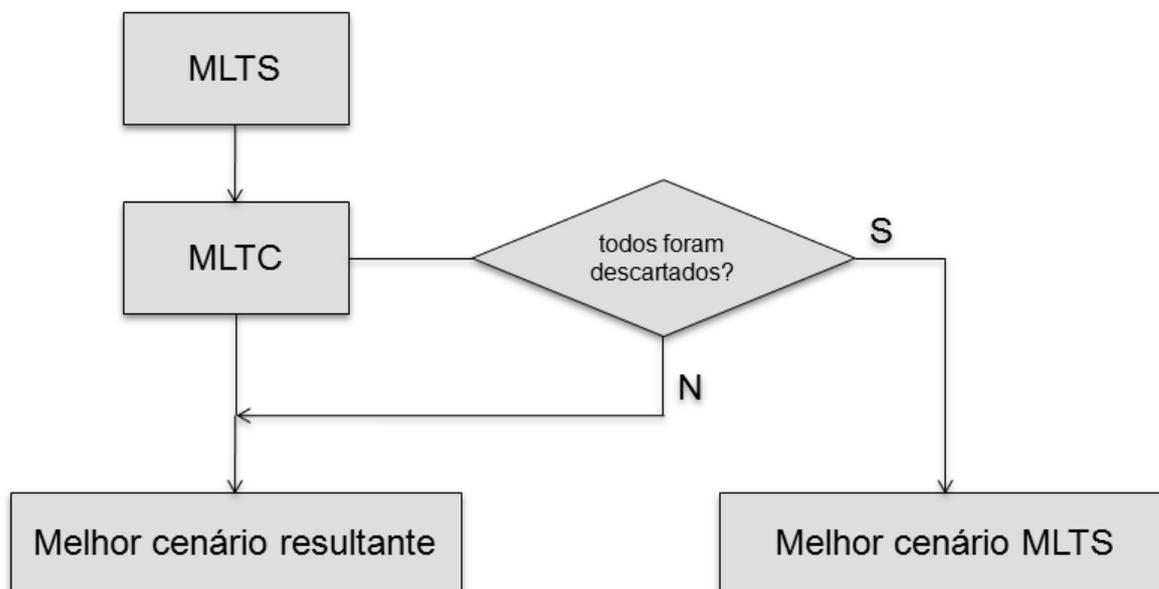


Figura 12. Fluxograma simplificado do algoritmo completo

## 4. METODOLOGIA E AMBIENTE DE TESTES

Nesse capítulo estão descritos os métodos e o ambiente de testes utilizados para a simulação dos diferentes cenários de rede, bem como as ferramentas de *software* que foram desenvolvidas para a aplicação e a verificação da política de justiça. Todas as simulações foram realizadas utilizando o simulador de redes NS2 (NS2, 2009).

### 4.1. Ambiente de Simulação

Para a verificação do desempenho de várias *STAs* conectadas a um mesmo *AP* e dos efeitos resultantes das variações das taxas em função da alteração dos parâmetros de *CW*, foi criado um *script* para o software NS2 capaz de simular um cenário virtual de testes, com três *STAs* (*STA1*, *STA2* e *STA3*), arranjadas conforme a Figura 13.

Todas as *STAs* sem fio foram configuradas como *STAs* fixas, desconsiderando, conseqüentemente, a possibilidade de variação de taxa em função de mobilidade no cenário de simulação. Assim, cada simulação corresponde a um estado estático de uma rede sem fio convencional; a dinâmica do comportamento de uma rede sem fio, quanto à mobilidade das *STAs*, pode ser obtida através da realização de uma seqüência de simulações estáticas, cada uma representando um estado temporal da rede, com o posicionamento específico das *STAs* em cada instante de tempo.

As *STA1*, *STA2* e *STA3* foram configuradas de forma a permitir a variação de suas taxas de conexão e dos parâmetros *CWmin* e *CWmax*. Ao *AP* foram adicionadas 3 fontes de tráfego *UDP* denominadas *udp1*, *udp2* e *udp3*. Em cada uma das três *STAs* foram adicionados drenos de tráfego, chamados de *sink1*, *sink2* e *sink3*. A função de cada um desses drenos é receber o tráfego gerado pelas fontes de tráfego e gerar os dados estatísticos resultantes da simulação. As fontes *udp1*, *udp2* e *udp3* geram tráfegos *CBR*, cada um com pacotes de 900 bytes. Esses pacotes são enviados através do *AP* para os drenos *sink1*, *sink2* e *sink3*, respectivamente.

Para análise e comparação dos cenários foram criadas 4 variáveis: *mediaSink1*, *mediaSink2*, *mediaSink3* e *mediaSomaSinks*, atualizadas automaticamente a cada 10 segundos de simulação. As três primeiras variáveis correspondem respectivamente à média aritmética dos 10 últimos valores de bytes recebidos pelo *sink1*, pelo *sink2* e pelo *sink3* (tomados a cada segundo nos últimos 10 segundos de simulação). A quarta variável corresponde à média aritmética dos últimos 10 valores da soma dos bytes recebidos pelos três drenos (*sink1*, *sink2* e *sink3*), tomados a cada segundo nos últimos 10 segundos de simulação.

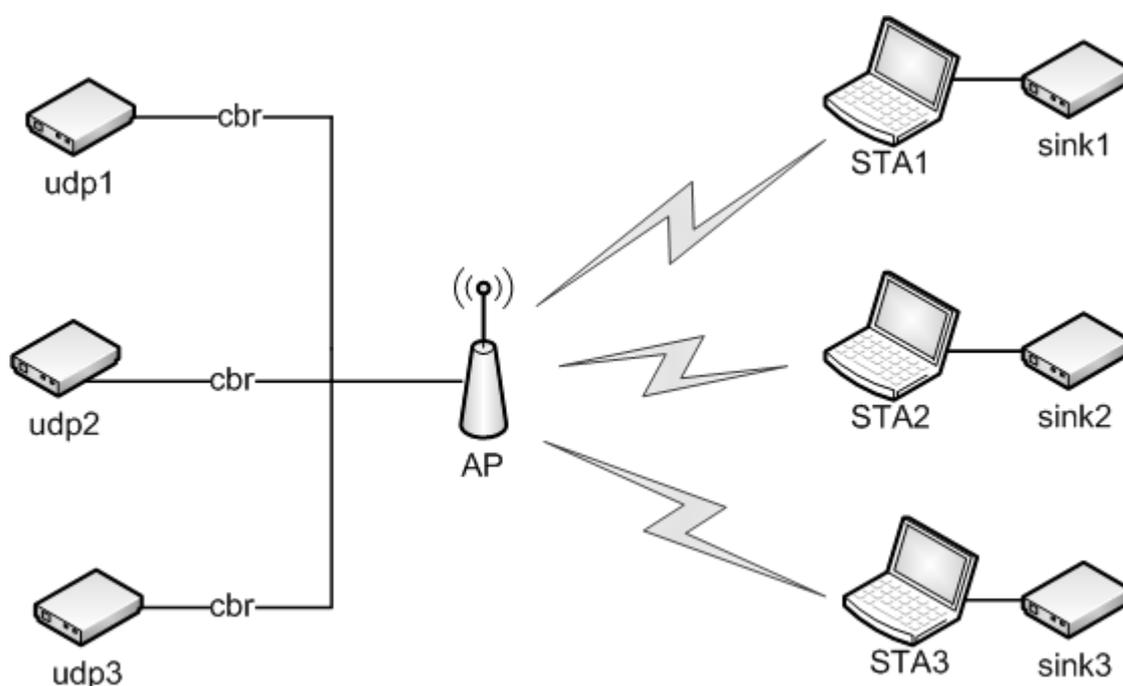


Figura 13. Cenário Geral de Testes

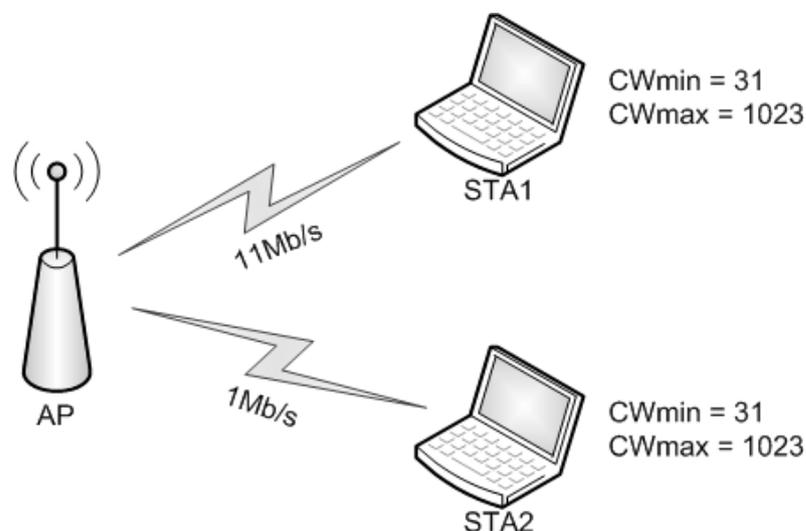
## 4.2. Validação do Ambiente de Simulação

A fim de apresentar as características da rede e de validar os *scripts* de teste do simulador NS2 duas simulações foram realizadas, com o objetivo de validar o comportamento da rede quanto ao aparecimento da Anomalia da *MAC* e a sua priorização.

### 1.1.6. Simulação 1 – A Anomalia da *MAC*

Essa simulação teve como propósito evidenciar os efeitos da Anomalia da *MAC* na rede sob teste. Consiste no instanciamento de duas *STAs* sem fio (*STA1* e *STA2*) conectadas a um mesmo *AP*, ambas com valores mínimos e máximos de Janela de Contenção definidos, respectivamente, como  $CW_{min} = 31$  e  $CW_{max} = 1023$ . A conexão entre a *STA1* e o *AP* foi definida com uma taxa de 11Mb/s, e a

conexão entre a *STA2* e o *AP* foi definida com uma taxa de 1Mb/s. O cenário de simulação está representado na Figura 14.



**Figura 14. Cenário de simulação da Anomalia da MAC em rede com duas STAs**

A seqüência de simulação está representada na Figura 15. Nessa figura, observa-se que, inicialmente, ambas as *STAs* estavam desligadas. Em  $t = 10$  segundos, a *STA1* iniciou uma transmissão. Em  $t = 30$  segundos, a segunda *STA* iniciou sua transmissão, permanecendo assim até o instante  $t = 50$  segundos, quando foi desligada. Finalmente, em  $t = 70$  segundos, a *STA1* também foi desligada.

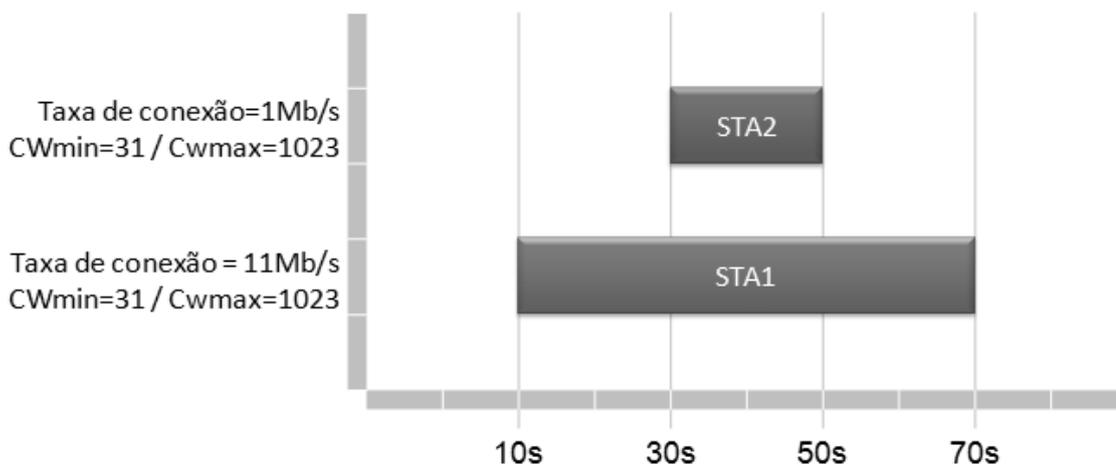


Figura 15. Seqüência de Simulação da Rede para o caso 1

Os resultados da simulação estão representados no gráfico apresentado na Figura 16.

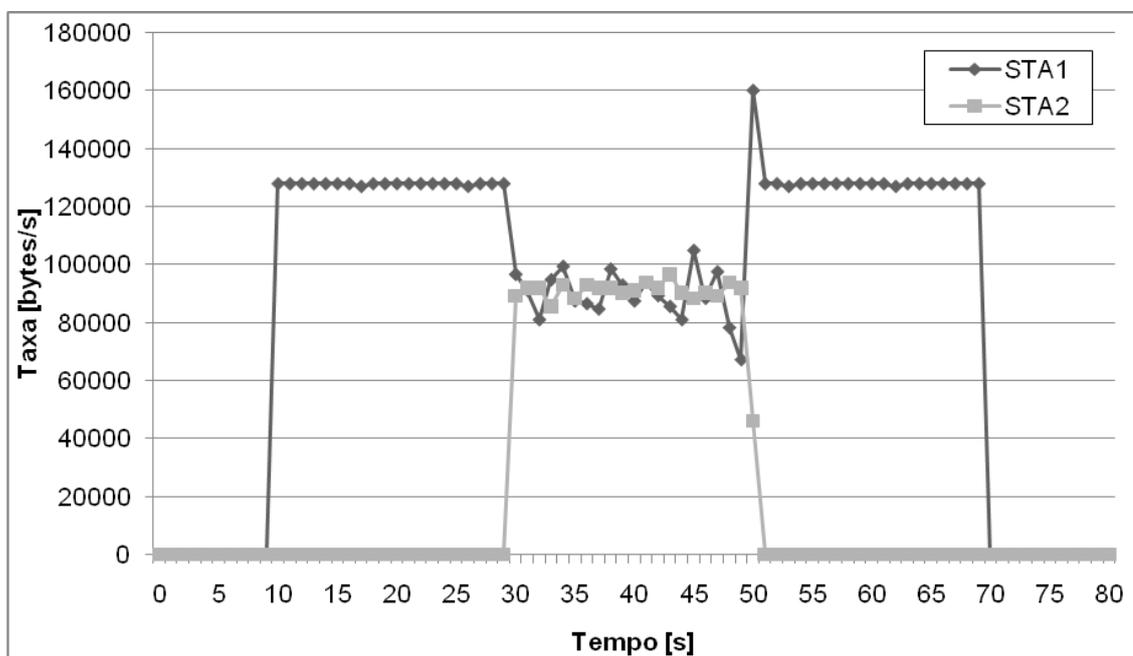


Figura 16. Anomalia da MAC

Essa simulação evidencia a Anomalia da MAC ao explicitar que o tráfego da STA2 prejudicou o tráfego da STA1, reduzindo drasticamente sua taxa média em

aproximadamente 30% (de 130kbytes/s para 90kbytes/s). Essa redução se manteve enquanto a *STA2* permaneceu em estado de transmissão de dados (iniciando em  $t = 30$  segundos), com uma taxa média de 90kbytes/s. A partir do instante em que a *STA2* é desligada da rede (em  $t = 50$  segundos) *STA1* recupera seu nível original de taxa de transmissão.

### 1.1.7. Simulação 2 – Priorização da Anomalia da MAC

A segunda simulação tem a finalidade de apresentar a Priorização da Anomalia da *MAC*, e a rede simulada consiste novamente em duas *STAs* sem fio conectadas a um mesmo *AP*. Como anteriormente, a primeira *STA* (*STA1*) foi configurada para se conectar a uma taxa de 11 Mb/s, e teve seus parâmetros mínimos e máximos de janela de contenção definidos respectivamente como  $CW_{min} = 31$  e  $CW_{max} = 1023$ . Entretanto, a segunda *STA* (*STA2*), embora também configurada para se conectar a uma taxa de 1Mb/s, tem, nesse caso, seus parâmetros mínimos e máximos de Janela de Contenção definidos respectivamente como  $CW_{min} = 1$  e  $CW_{max} = 31$ . O cenário de simulação proposto é mostrado na Figura 17.

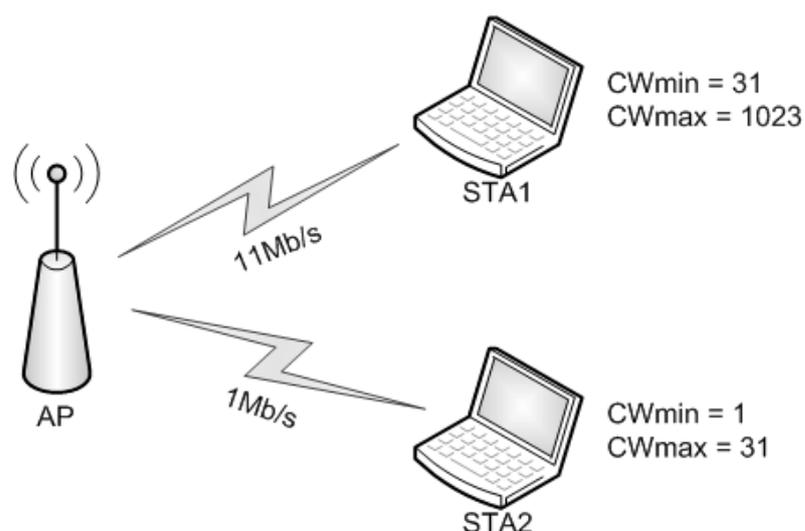
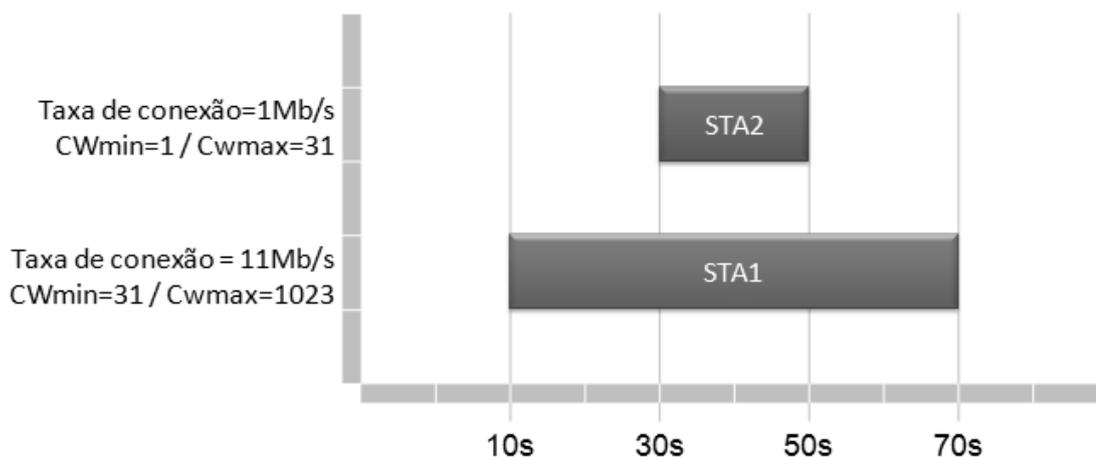


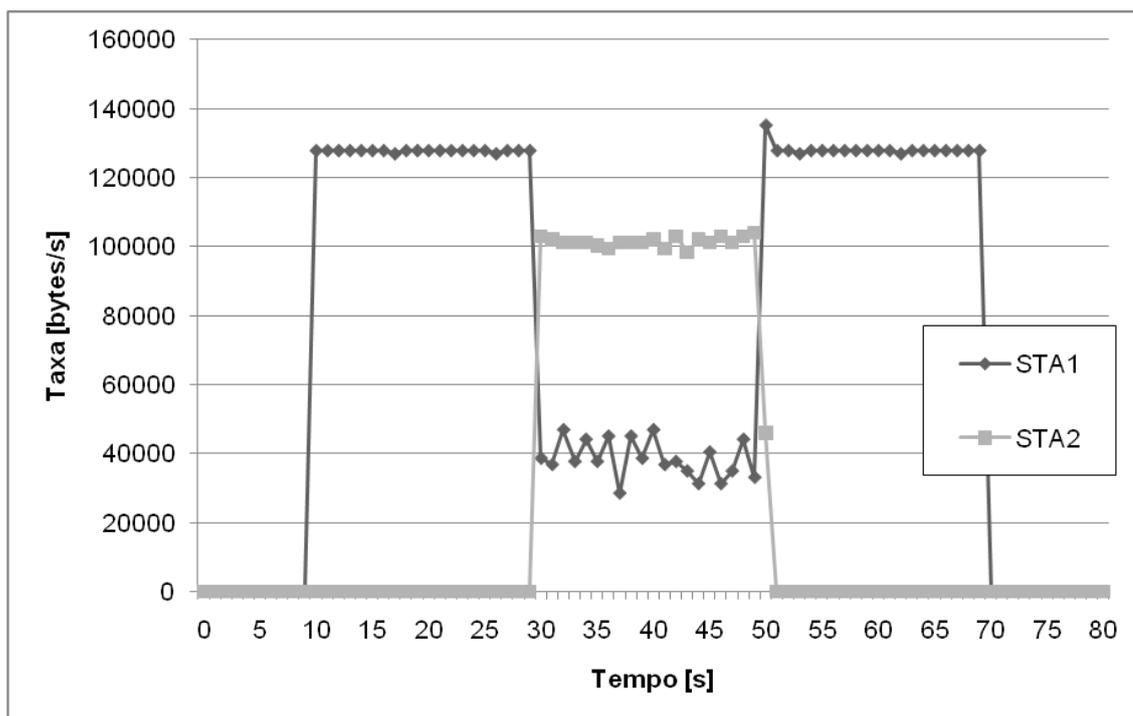
Figura 17. Cenário de simulação da Priorização da Anomalia em rede com duas *STAs*

A seqüência dessa simulação está representada na Figura 18. Nessa seqüência, inicialmente ambas as *STAs* estavam desligadas. Em  $t = 10$  segundos a *STA1* inicia sua transmissão. Em  $t = 30$  segundos a segunda *STA* inicia sua transmissão, permanecendo assim até o instante  $t = 50$  segundos, quando foi desligada. Finalmente, em  $t = 70$  segundos, a *STA1* também foi desligada.



**Figura 18. Seqüência de Simulação da Rede para o caso 2**

Os resultados da simulação estão representados no gráfico apresentado na Figura 19.



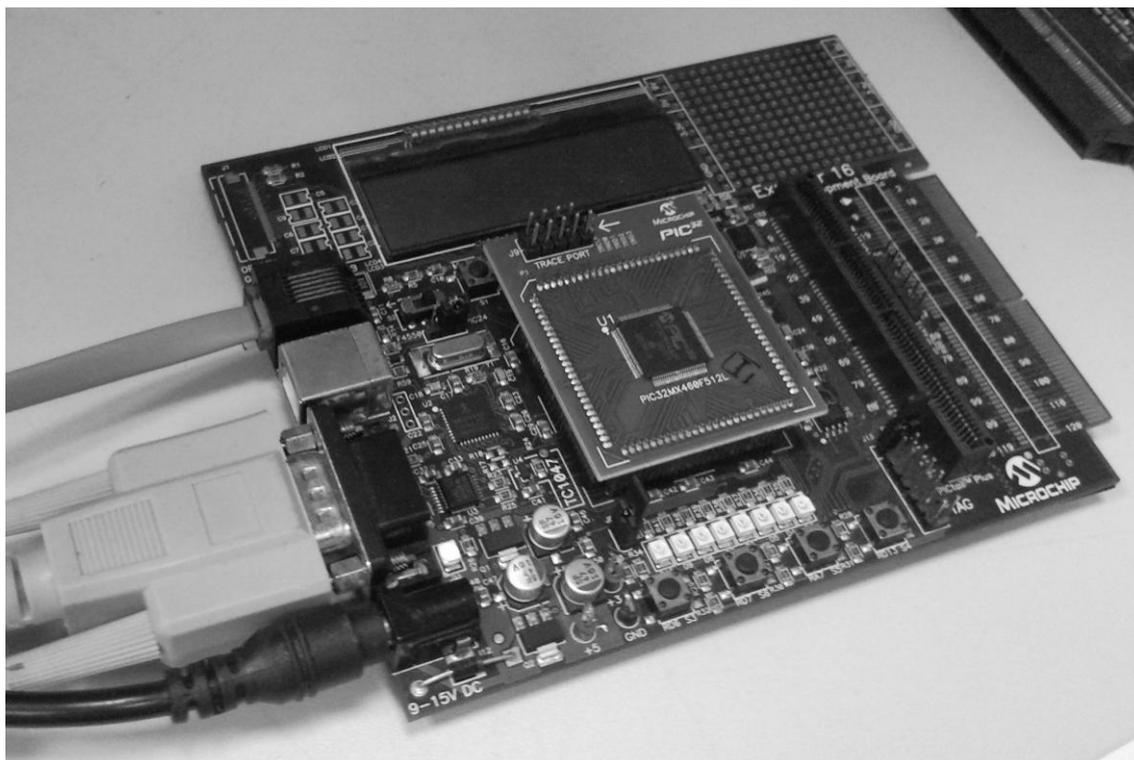
**Figura 19. Priorização da Anomalia**

Essa simulação evidenciou a Priorização da Anomalia da *MAC*, uma vez que a *STA2*, com menor taxa, foi priorizada e, assim, teve a sua probabilidade de acesso ao canal aumentada. Dessa forma, a taxa média das *STAs*, quando simultaneamente conectadas, foi menor que a da simulação anterior, com uma redução de 45% para a *STA1* (que passou de uma taxa de 90kbytes/s no caso 1 para 40kbytes/s no caso 2).

### 4.3. Implementação da Política de Justiça em Placa de Desenvolvimento

A fim de testar o desempenho e a viabilidade da política em um ambiente real foi utilizada uma placa de desenvolvimento *Explorer 16*, da *Microchip*, montada com um processador *PIC32* de 32 *bits*, mostrada na Figura 20. Foi desenvolvido um software em linguagem C com o propósito de calcular todas as

combinações possíveis para o estado da rede, a partir da entrada do número de *STAs* e das configurações de serviços de cada *STA*, e informar, através da porta serial, a configuração com o melhor Índice de Justiça. Esse *software* está descrito no Anexo A.

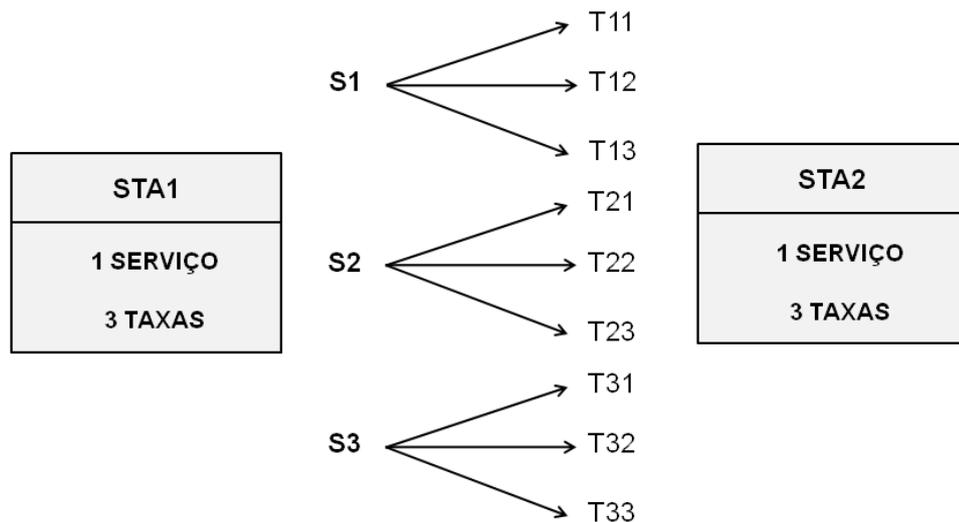


**Figura 20. Placa de Desenvolvimento utilizada para implementação da Política de Justiça**

A implementação em *hardware* tem o propósito de testar a política de justiça, proposta nesse trabalho, em relação à sua implementação futura em roteadores de redes, uma vez que tais equipamentos têm seu funcionamento fundamentado em aplicações com microcontroladores. Portanto, a implementação do *software* (descrito no item a seguir) em um microcontrolador de 32 *bits* permite a avaliação da viabilidade da aplicação da política em um ambiente real.

### 1.1.8. Implementação da Política de Justiça

Para implementação da política foi considerado que cada *STA* deve trafegar apenas um tipo de serviço, embora sempre disponibilizando ao *AP* três opções de taxas para atendimento desse serviço. Uma vez definido o número de *STAs* e as opções de serviço, o processador pode avaliar o índice de justiça em todas as combinações possíveis, armazenando o maior em memória ou registrador. Ao final do processo, o processador tem condições de apresentar a combinação que apresentou o melhor índice de justiça. Entretanto, nas condições descritas anteriormente, o cálculo do índice de justiça é realizado de forma exaustiva, testando todas as combinações possíveis. Como exemplo, a Figura 21 ilustra todas as combinações possíveis para um cenário com 2 *STAs*.



**Figura 21. Todas as combinações possíveis para duas *STAs***

Como cada *STA* possui apenas um tipo de serviço, e três opções de atendimento, o número total de combinações a serem analisadas é dado por:

$$C_t = 3^N \quad (5)$$

onde  $N$  é o número de *STAs* que disputam o canal

Duas variáveis foram utilizadas como métrica para avaliar o desempenho da política na implementação em hardware. A primeira é o número total de combinações possíveis para a rede (dada pela equação 4), e a segunda é o tempo total de processamento para calcular todas as combinações. A medida de tempo total de processamento foi realizada com o auxílio de um osciloscópio, conectado ao barramento dos pinos do processador. O programa foi colocado em loop e implementado de forma que o barramento de pinos inverteria seu sinal sempre que uma nova seqüência de teste fosse iniciada. Assim, a partir da captura de medidas seqüenciais, os pinos produziam uma onda quadrada, cujo período completo corresponde ao tempo requerido para dois testes. A Figura 22 apresenta a tela do osciloscópio no momento do cálculo do índice de justiça para uma rede com 3 STAs. Nesse caso o tempo médio necessário para o cálculo de uma das combinações de configuração da rede é 404us.

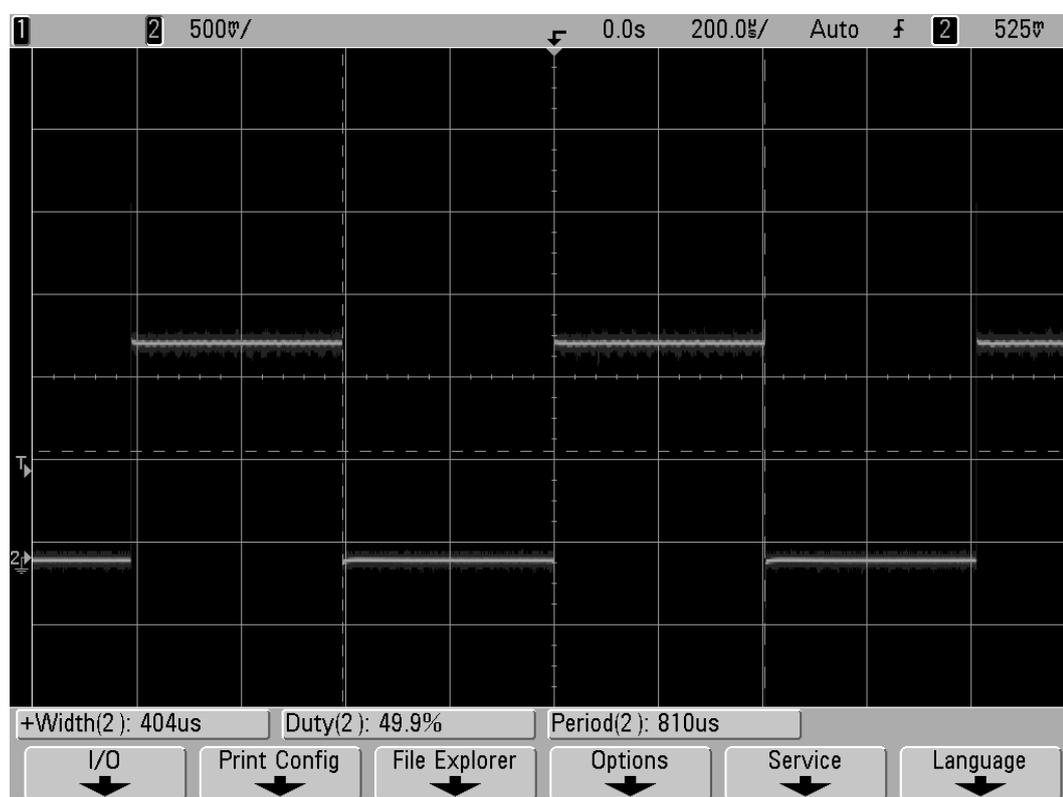
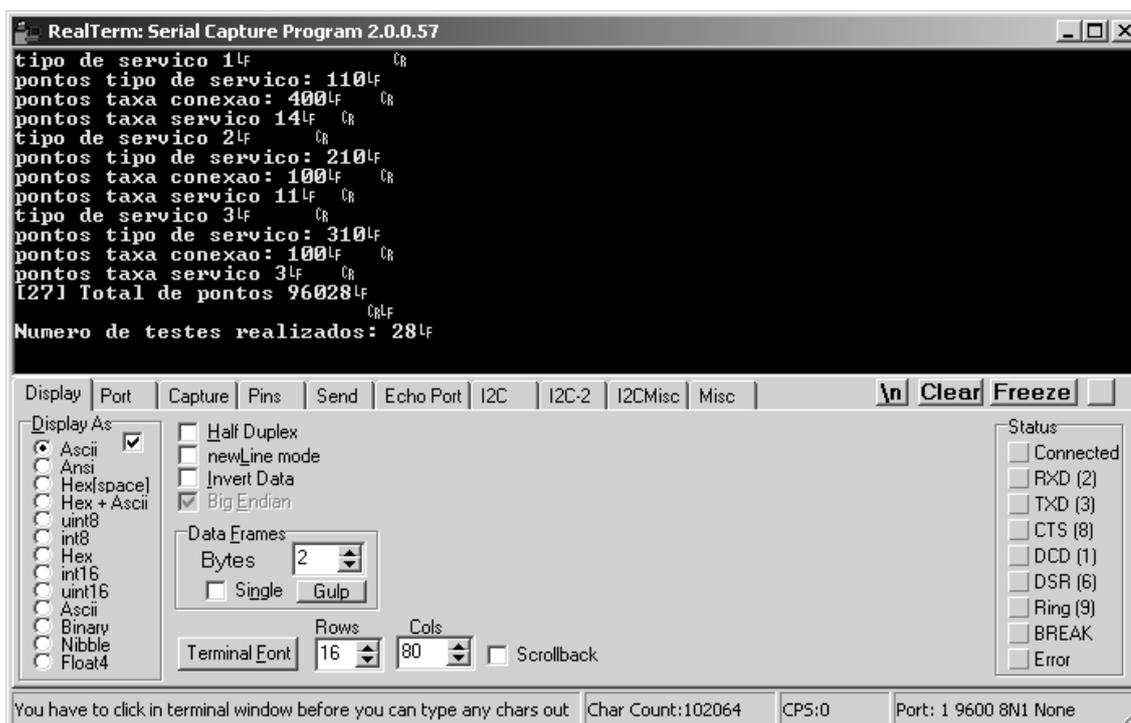


Figura 22. Saída do Barramento de Pinos do Processador

Durante a realização dos cálculos, o microcontrolador envia, através de uma porta serial, as informações sobre a combinação atual e, ao final do processo iterativo completo, o número de testes realizados e o melhor cenário calculado. A Figura 23 apresenta um exemplo de captura das informações transmitidas via porta serial.



**Figura 23. Exemplo de captura das informações transmitidas via porta serial**

Esse método de medição de tempos pode ser estendido para redes com um número maior de STAs. A Tabela 6 mostra o número de combinações e os tempos de processador necessários para o cálculo da Justiça (para uma combinação) em função do número de STAs. A Figura 24 apresenta um gráfico com o número total de combinações necessárias versus o número total de STAs e a Figura 25 apresenta um gráfico com o tempo de cálculo para uma das combinações versus o número total de STAs.

STAs	Combinações	Tempo
3	27	404us
4	81	630us
5	243	5.9ms
6	729	19ms
7	2187	71ms
8	6561	242ms
9	19683	785ms
10	59049	2.6s

Tabela 6. Tempo de processamento para uma combinação em função do número de STAs

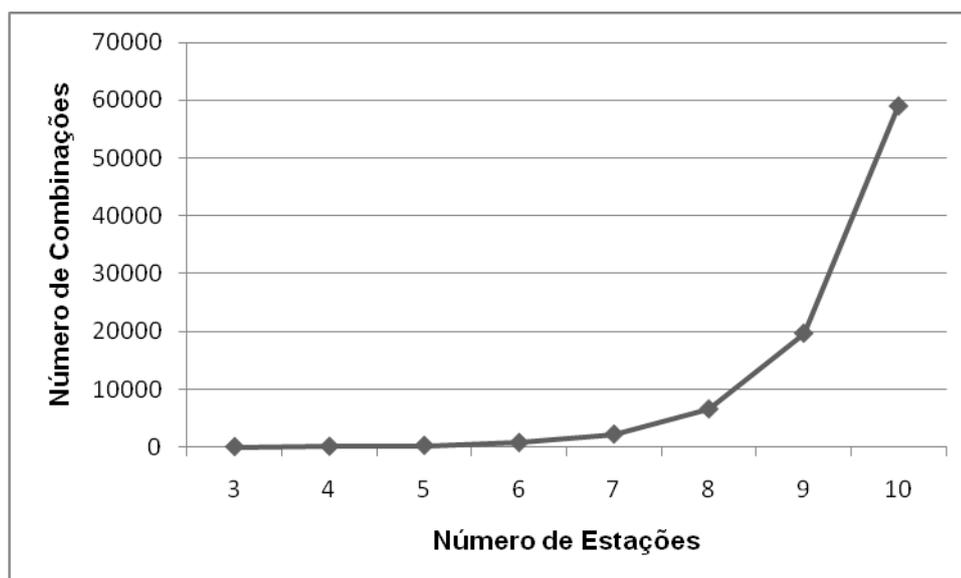
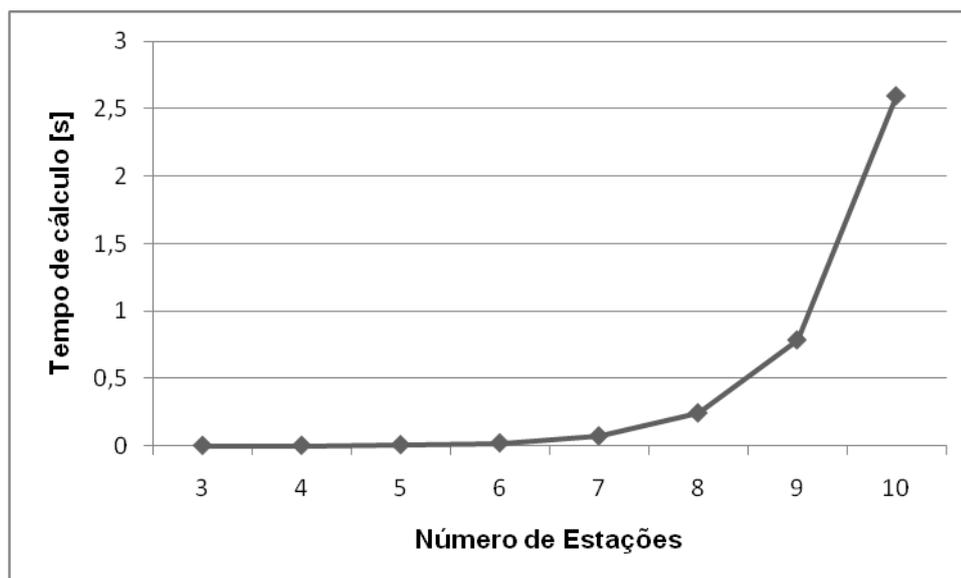


Figura 24. Número de combinações versus o número de STAs



**Figura 25. Tempo de cálculo para uma combinação versus o número de STAs**

Como esperado, o tempo de processamento é proporcional ao número de combinações a serem avaliadas. Mais ainda, dessa relação, fica evidente que, conforme o número de *STAs* na rede aumenta, o número de configurações a serem avaliadas também aumenta exponencialmente. Em redes reais, o tempo de máquina necessário para a avaliação do índice de justiça de todas as combinações possíveis pode tornar inviável a aplicação da política de justiça, a partir de um número suficiente de *STAs* conectadas na rede.

Assim, torna-se desejável, para a implementação real da política, a utilização de um método heurístico, onde algumas combinações possam ser automaticamente descartadas, diminuindo consideravelmente o tempo de teste. Neste trabalho, essa característica é contemplada pelo método MLTC, onde o descarte de combinações classificadas como fisicamente impossíveis configura-se como uma aplicação de uma regra heurística que contribui para redução do esforço de processamento requerido para a avaliação da justiça na rede. Deve-se salientar, entretanto, que devem ser tomadas as devidas precauções na implementação desse método, uma vez que o mesmo pode considerar ineficazes todas as combinações para redes onde as *STAs* com solicitação de taxas de serviço maiores que suas taxas de conexão. Uma possível solução para essa

situação é a de considerar a alocação convencional de recursos prevista originalmente pelo padrão IEEE 802.11 caso o MLTC considere todas as configurações da rede sem solução.

## 5. RESULTADOS OBTIDOS

Nesse capítulo são apresentados os resultados obtidos, tomando como base o cenário geral de testes delineado no capítulo 4. A partir desse cenário geral, foram definidos quatro cenários específicos para a realização das simulações com o software NS2.

Em todos os cenários as três *STAs* da rede simulada foram ajustadas para receber dados originados de três fontes diferentes, provenientes de um mesmo *AP*. Após um intervalo de estabilização de 10 segundos, foram registradas a média aritmética do número de bytes recebidos por cada *STA* e a média aritmética da soma total dos bytes recebidos pelas *STAs*. Ambas as médias foram calculadas em função dos dados dos últimos 10 períodos de simulação.

Para a verificação do nível de justiça na rede, quanto à distribuição de recursos, todas as possíveis combinações de taxas de conexão e parâmetros de serviço para as *STAs* da rede foram verificadas. Os resultados assim obtidos são comparados com os resultados da combinação de número 1, tomada como caso base em cada cenário. Isso porque essa combinação corresponde àquela que seria normalmente utilizada pelo padrão IEEE802.11 caso a política não fosse aplicada, e, portanto, é a que apresenta os valores que caracterizam o padrão atual para o atendimento dos serviços.

A seguir, são apresentados os cenários simulados, sua caracterização e os resultados obtidos em cada um deles.

### 5.1. Cenário 1 - Análise de Resultados

Nesse cenário, os valores de taxas de conexão e parâmetros de serviço para cada *STA* foram determinados conforme a Tabela 7.

STA 1	STA 2	STA 3
Taxa de conexão: 1Mbps	Taxa de conexão: 11Mbps	Taxa de conexão: 5,5Mbps
Tipo de serviço: BEST_EFFORT	Tipo de serviço: VIDEO	Tipo de serviço: VOICE
Taxas de serviço: 1Mbps	Taxas de serviço: 700kbps	Taxas de serviço: 256kbps
750kbps	500kbps	128kbps
500kbps	250kbps	64kbps

**Tabela 7. Taxas de conexão e parâmetros de serviço para o cenário 1**

Os resultados do cálculo do índice de Justiça, ordenados em ordem crescente de média da soma das taxas (*throughput* total da rede), estão apresentados na Tabela 8. Inicialmente os índices de justiça foram determinados usando o método MLTS, para todas as combinações do cenário 1.

	STA1	STA2	STA3	STA1	STA2	STA3	média da soma das taxas das STAs	Índice de justiça calc.
	taxa de conexão			taxa média				
comb.	1Mb/s	11Mb/s	5.5Mb/s	(bytes/s)	(bytes/s)	(bytes/s)	(bytes/s)	(MLTS)
27	.500	.250	.064Mb	63848	31924	8188	103960	953000
26	.500	.250	.128Mb	63848	31924	16376	112148	1199000
25	.500	.250	.256Mb	63848	31924	32752	128524	1568000
18	.750	.250	.064Mb	95864	31924	8188	135976	1016000
24	.500	.500	.064Mb	63848	63940	8188	135976	1573000
9	1Mb	.250	.064Mb	103316	31924	8188	143428	1037000
17	.750	.250	.128Mb	95864	31924	16284	144072	1262000
23	.500	.500	.128Mb	63848	63940	16376	144164	1819000
8	1Mb	.250	.128Mb	98992	31924	16284	147200	1283000
16	.750	.250	.256Mb	95864	31924	32660	160448	1631000
22	.500	.500	.256Mb	63848	63940	32752	160540	2188000
7	1Mb	.250	.256Mb	95864	32108	32844	160816	1652000
21	.500	.700	.064Mb	63848	89516	8188	161552	1821000
6	1Mb	.500	.064Mb	95128	63848	8188	167164	1657000
15	.750	.500	.064Mb	96140	63940	8188	168268	1636000
20	.500	.700	.128Mb	63940	89516	16376	169832	2067000
1	1Mb	.700	.256Mb	61272	89608	19228	170108	2520000
5	1Mb	.500	.128Mb	93380	63940	16284	173604	1903000
14	.750	.500	.128Mb	93380	63940	16284	173604	1882000
2	1Mb	.700	.128Mb	76912	89424	9292	175628	2151000
3	1Mb	.700	.064Mb	84180	89516	5704	179400	1905000
4	1Mb	.500	.256Mb	87952	63848	32752	184552	2272000
13	.750	.500	.256Mb	88320	64032	32752	185104	2251000
19	.500	.700	.256Mb	63940	89516	32752	186208	2436000
12	.750	.700	.064Mb	89148	89516	8188	186852	1884000
11	.750	.700	.128Mb	85744	89424	16284	191452	2130000
10	.750	.700	.256Mb	79764	89700	32752	202216	2499000

**Tabela 8. Resultados do cálculo do índice de Justiça para todas as combinações do cenário 1, segundo o MLTS.**

Nessas condições, a política apresentou como mais justa a combinação de número 1, salientada na Tabela 8 com os dados em negrito. Porém, observando

os resultados da tabela, verifica-se que existem outras combinações mais justas, como a de número 10, por exemplo. Essa combinação, embora tenha tido um índice de justiça menor que a combinação de número 1 deve ser considerada mais justa, pois apresenta taxas médias maiores para as três STAs e, em especial, para a STA3, que apresenta o tráfego de maior prioridade.

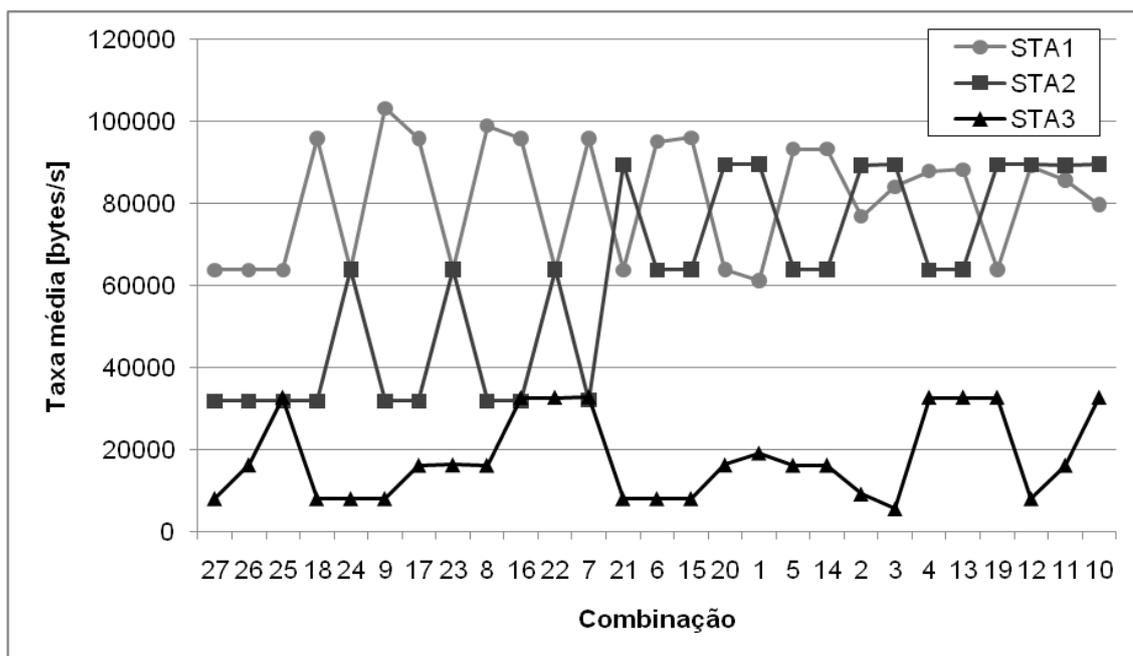
Esse equívoco na determinação da combinação mais justa decorre justamente da utilização de valores teóricos, pelo MLTS, para o cálculo do índice de justiça. Consequentemente, não são consideradas as taxas reais das STAs, já que as mesmas só podem ser mensuradas após o ajuste e aplicação dos parâmetros das STAs.

Por outro lado, os resultados do cálculo do índice de Justiça, usando o método MLTC estão apresentados na Tabela 9, para todas as combinações do cenário 1 classificadas pelo método como fisicamente possíveis, e também ordenados em ordem crescente de média da soma das taxas (*throughput* total da rede). Pode-se observar que aplicando-se apenas o método MLTC, a política proposta apresenta como mais justa a combinação número 10. Nesse caso, inclusive, a política apresenta como combinação mais justa também a de maior *throughput*.

	STA1	STA2	STA3	STA1	STA2	STA3	média da soma das taxas das STAs	Índice de justiça calc.
	taxa de conexão			taxa média				
comb.	1Mb/s	11Mb/s	5.5Mb/s	(bytes/s)	(bytes/s)	(bytes/s)	(bytes/s)	(MLTC)
27	.500	.250	.064Mb	63848	31924	8188	103960	953000
26	.500	.250	.128Mb	63848	31924	16376	112148	1199000
25	.500	.250	.256Mb	63848	31924	32752	128524	1568000
18	.750	.250	.064Mb	95864	31924	8188	135976	1016000
24	.500	.500	.064Mb	63848	63940	8188	135976	1573000
17	.750	.250	.128Mb	95864	31924	16284	144072	1262000
23	.500	.500	.128Mb	63848	63940	16376	144164	1819000
16	.750	.250	.256Mb	95864	31924	32660	160448	1631000
22	.500	.500	.256Mb	63848	63940	32752	160540	2188000
21	.500	.700	.064Mb	63848	89516	8188	161552	1821000
15	.750	.500	.064Mb	96140	63940	8188	168268	1636000
20	.500	.700	.128Mb	63940	89516	16376	169832	2067000
14	.750	.500	.128Mb	93380	63940	16284	173604	1882000
13	.750	.500	.256Mb	88320	64032	32752	185104	2251000
19	.500	.700	.256Mb	63940	89516	32752	186208	2436000
12	.750	.700	.064Mb	89148	89516	8188	186852	1884000
11	.750	.700	.128Mb	85744	89424	16284	191452	2130000
10	.750	.700	.256Mb	79764	89700	32752	202216	2499000

**Tabela 9. Resultados do cálculo do índice de Justiça para todas as combinações possíveis do cenário 1, segundo o MLTC.**

Por fim, aplicando-se o algoritmo de Justiça completo a combinação apresentada como a **mais justa** também foi a de **número 10**. As taxas médias obtidas após período de estabilização, em bytes/s, para as diferentes combinações do cenário 1, estão representadas na Figura 26.



**Figura 26. Taxas médias obtidas após período de estabilização para as diferentes combinações do cenário 1.**

Em relação ao ganho de desempenho, a Figura 27 apresenta a comparação entre as taxas obtidas para o caso base (combinação 1) e o caso apontado como o mais justo pela política proposta.

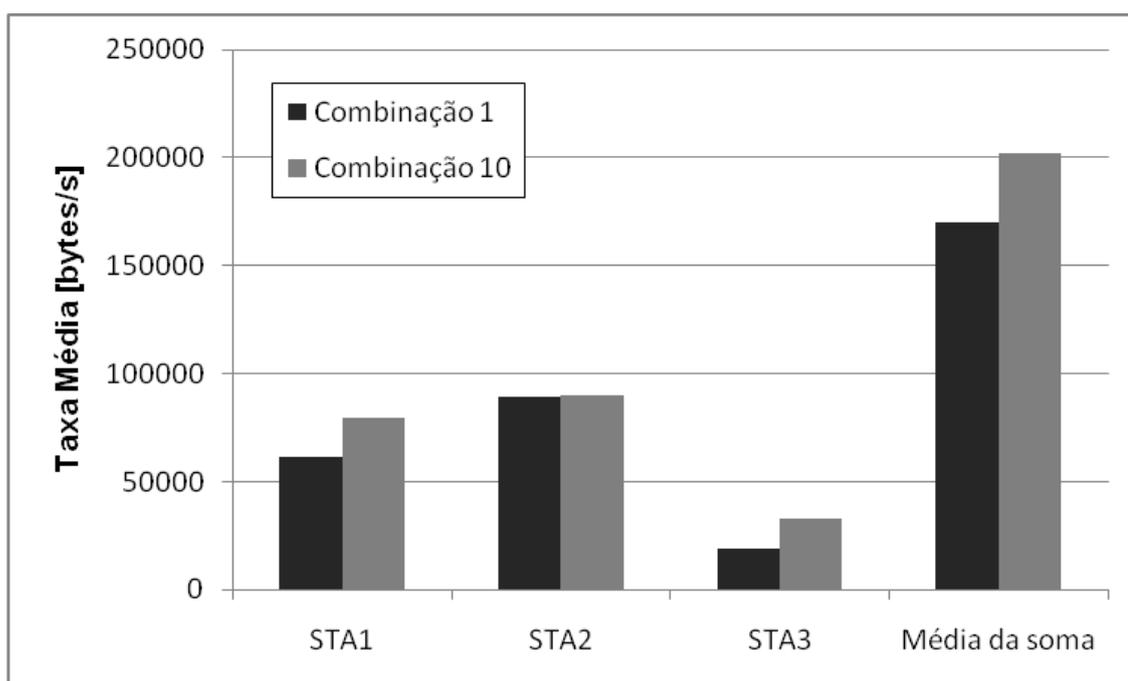


Figura 27. Análise de desempenho pela aplicação da política para o cenário 1.

Da análise dessa figura, pode-se observar que a combinação apontada como mais justa apresentou ganhos significativos quando comparada com o caso base (combinação 1). A taxa média da *ST1* passou de 61272 bytes/s para 79764 bytes/s, um aumento de aproximadamente 30%. Para a *STA2* a taxa ficou praticamente estável.

O melhor resultado foi obtido com a *STA3*, que teve um aumento de aproximadamente 70% na taxa, passando de 19228 bytes/s para 32752 bytes/s. A média da soma das taxas de todas as *STAs* passou de 170108 bytes/s para 202216 bytes/s, um aumento de aproximadamente 18%.

## 5.2. Cenário 2 - Análise de Resultados

O cenário 2, também foi simulado com base no caso geral de uma rede com três *STAs* apresentado no capítulo 4, tem como taxas de conexão e parâmetros de serviço para cada *STA* os valores apresentados na Tabela 10.

STA 1	STA 2	STA 3
Taxa de conexão: 1Mbps	Taxa de conexão: 5,5Mbps	Taxa de conexão: 11Mbps
Tipo de serviço: BEST_EFFORT	Tipo de serviço: VIDEO	Tipo de serviço: VOICE
Taxas de serviço: 11Mbps 5,5Mbps 1Mbps	Taxas de serviço: 2Mbps 1Mbps 512kbps	Taxas de serviço: 512kbps 256kbps 128kbps

**Tabela 10. Taxas de conexão e parâmetros de serviço para o cenário 2.**

Nesse cenário, explicita-se uma das situações problemáticas enfrentadas no desenvolvimento da política de justiça: A *STA1* está conectada a uma taxa de 1Mb/s, mas informa ao *AP* que duas das três opções de atendimento de seu serviço exigem uma taxa maior que a própria taxa de conexão da *STA*.

Quando aplicada apenas com o método MLTC, a política não é capaz de determinar um cenário com a maior justiça. Isso decorre do fato de todos os cenários serem eliminados ao serem classificados como “fisicamente impossíveis” pois, de acordo com o MLTC, tendem a ocupar o canal por um tempo maior que o disponível (1 segundo), por definição, pela aplicação do critério de justiça.

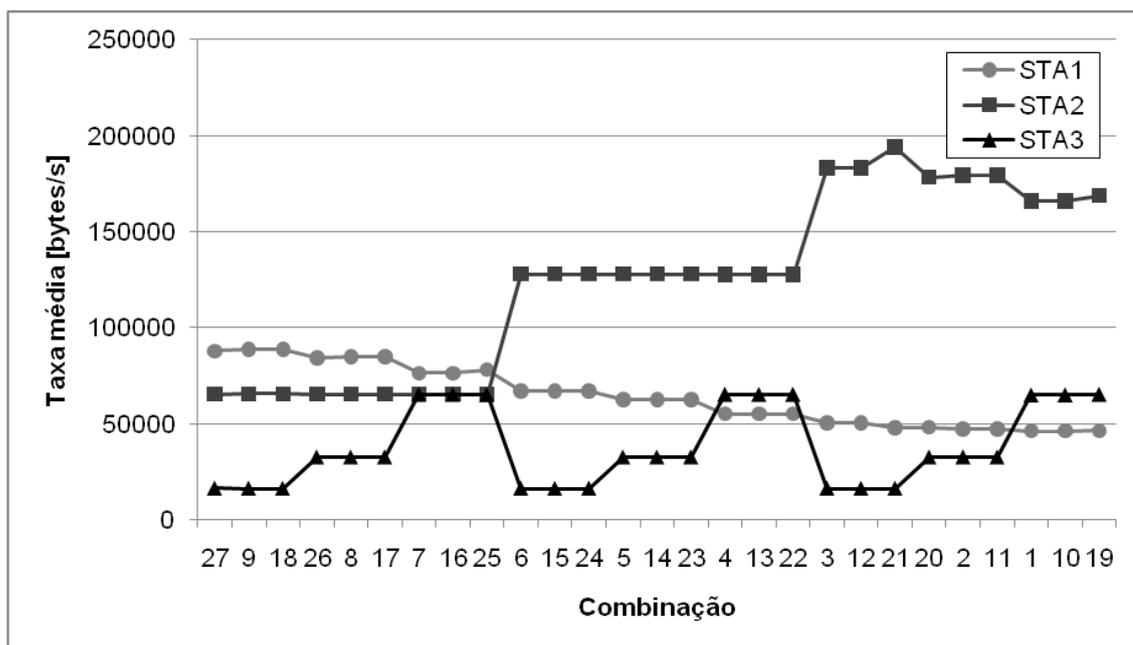
Já em relação ao MLTS, os resultados da aplicação da política para todas as combinações do cenário 2 estão representados na Tabela 11, ordenados em ordem crescente de média da soma das taxas. Nesse caso, a política apresentou a combinação de número 19 como a mais justa. Mais ainda, a combinação escolhida como a mais justa foi também a combinação com maior *throughput*. Através da análise da Tabela 11 pode-se perceber que o número de *bytes* recebidos pela *STA1* foi um dos menores de todas as combinações. De forma semelhante, podemos perceber que situação bastante parecida aconteceu com a *STA2*. Com relação à *STA3*, o número de *bytes* recebidos foi um dos maiores entre todas as combinações. Isso explica o porquê da combinação número 19 ter sido escolhida como a melhor. Nesse caso, o cálculo do índice de justiça dá um maior peso à relação entre a priorização das *STAs*.

Conseqüentemente, a combinação apresentada como a mais justa no final da aplicação do algoritmo completo de justiça também foi a combinação de número 19.

comb.	STA1	STA2	STA3	STA1	STA2	STA3	média da soma das taxas das STAs (bytes)	Índice de justiça calc. (MLTS)
	taxa de conexão			taxa média				
	1Mb/s	5.5Mb/s	11Mb/s	(bytes/s)	(bytes/s)	(bytes/s)		
27	1	0,512	0,128	87860	65412	16376	169648	1809000
9	11	0,512	0,128	88688	65504	16284	170476	1809000
18	5,5	0,512	0,128	88688	65504	16284	170476	1809000
26	1	0,512	0,256	84272	65320	32752	182344	2301000
8	11	0,512	0,256	85008	65320	32660	182988	2301000
17	5,5	0,512	0,256	85008	65320	32660	182988	2301000
7	11	0,512	0,512	76452	65320	65412	207184	3121000
16	5,5	0,512	0,512	76452	65320	65412	207184	3121000
25	1	0,512	0,512	78200	65320	65320	208840	3121000
6	11	1	0,128	67160	127880	16284	211324	2088000
15	5,5	1	0,128	67160	127880	16284	211324	2088000
24	1	1	0,128	67160	127880	16284	211324	2088000
5	11	1	0,256	62468	127880	32660	223008	2580000
14	5,5	1	0,256	62468	127880	32660	223008	2580000
23	1	1	0,256	62468	127880	32660	223008	2580000
4	11	1	0,512	55200	127696	65412	248308	3400000
13	5,5	1	0,512	55200	127696	65412	248308	3400000
22	1	1	0,512	55200	127696	65412	248308	3400000
3	11	2	0,128	50324	183356	16284	249964	2181000
12	5,5	2	0,128	50324	183356	16284	249964	2181000
21	1	2	0,128	48024	194212	16284	258520	2181000
20	1	2	0,256	48116	178296	32660	259072	2673000
2	11	2	0,256	47196	179308	32660	259164	2673000
11	5,5	2	0,256	47196	179308	32660	259164	2673000
1	11	2	0,512	46184	166152	65136	277472	3493000
10	5,5	2	0,512	46184	166152	65136	277472	3493000
19	1	2	0,512	46368	168728	65412	280508	3493000

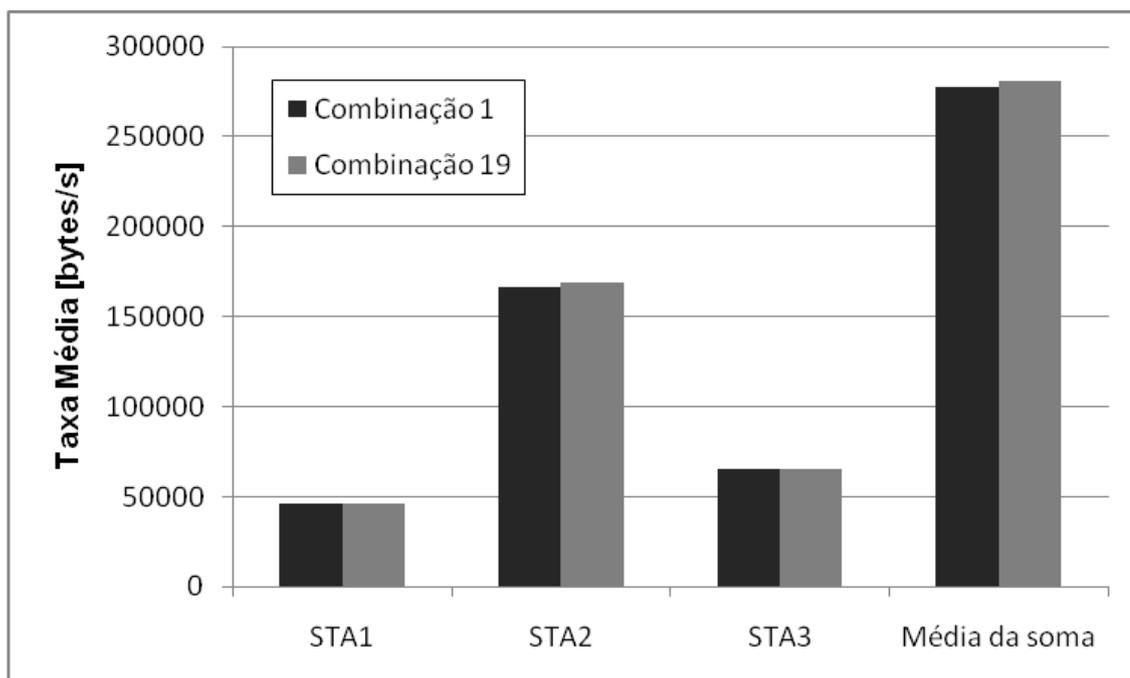
**Tabela 11. Resultados do cálculo do Índice de Justiça para todas as combinações do cenário 2, segundo o MLTS.**

As taxas médias obtidas após período de estabilização, em bytes/s, para as diferentes combinações do cenário 2, estão representadas na Figura 28.



**Figura 28. Taxas médias obtidas após período de estabilização para as diferentes combinações do cenário 2.**

Em relação ao ganho de desempenho, a Figura 29 apresenta a comparação entre as taxas obtidas para o caso base (combinação 1) e o caso apontado como o mais justo pela política proposta.



**Figura 29. Análise de desempenho pela aplicação da política para o cenário 2.**

Da análise dessa figura, pode-se observar que a combinação apontada como mais justa apresentou um ligeiro ganho quando comparada com o caso base (combinação 1). A taxa média da *ST1* ficou praticamente estável, com um aumento de aproximadamente 0,39%, passando de 46184 bytes/s para 46368 bytes/s. Situação semelhante aconteceu com a *STA3*, com um aumento de aproximadamente 0,42%, passando de 65136 bytes/s para 65412 bytes/s.

O melhor resultado foi obtido com a *STA2*, que teve um pequeno aumento de 1,55% na taxa, passando de 166152 bytes/s para 168728 bytes/s. A média da soma das taxas de todas as *STAs* passou de 277472 bytes/s para 280508 bytes/s, um aumento de aproximadamente 1%.

### 5.3. Cenário 3 – Análise de Resultados

O cenário 3, tem como taxas de conexão e parâmetros de serviço para cada STA os valores apresentados na Tabela 12. A rede simulada nesse cenário também está baseada no caso geral apresentado no capítulo 4.

STA 1	STA 2	STA 3
Taxa de conexão: 11Mbps	Taxa de conexão: 5,5Mbps	Taxa de conexão: 1Mbps
Tipo de serviço: BEST_EFFORT	Tipo de serviço: VIDEO	Tipo de serviço: VOICE
Taxas de serviço: 1Mbps	Taxas de serviço: 700kbps	Taxas de serviço: 256kbps
750kbps	500kbps	128kbps
500kbps	250kbps	64kbps

**Tabela 12. Taxas de conexão e parâmetros de serviço para o cenário 3.**

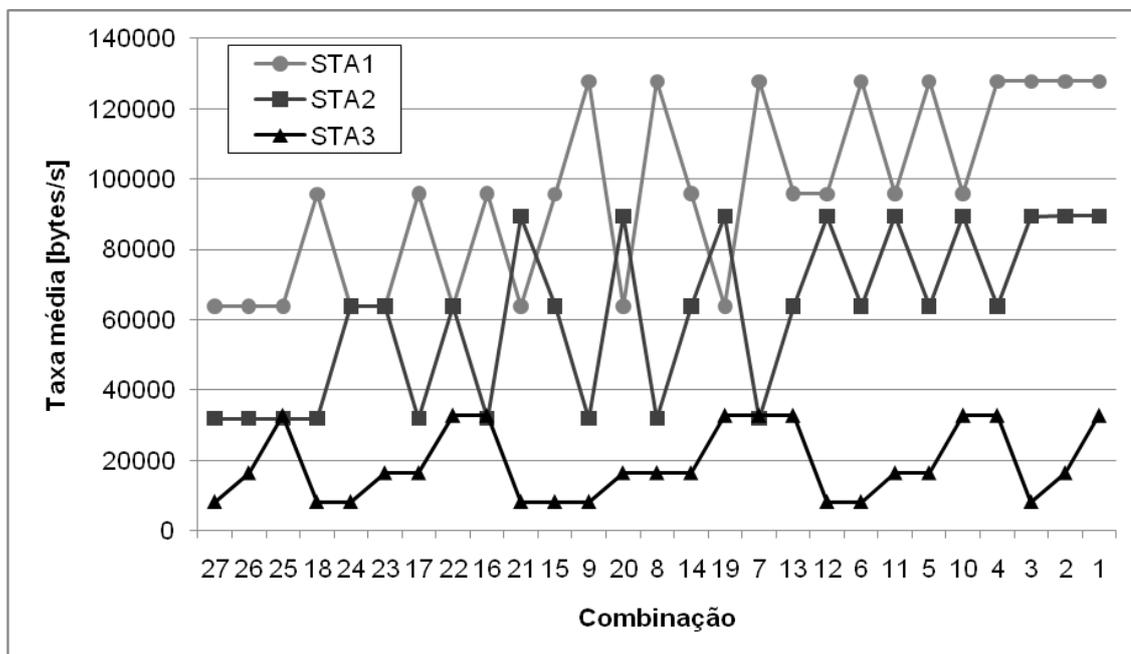
Os resultados obtidos para todas as combinações estão mostrados na Tabela 13. Nesse caso, a combinação escolhida como a mais justa, quando aplicada apenas com o método MLTS, foi a combinação de número 1, que é a combinação originalmente utilizada quando a política não é implementada.

Resultados semelhantes foram obtidos quando o índice de justiça foi calculado exclusivamente pelo método MLTC. Isso ocorreu devido ao fato de todas as STAs estarem conectadas com taxas suficientemente boas para atendimento máximo de seus serviços, eliminando, dessa forma, qualquer necessidade de alteração de parâmetros para melhora da justiça.

	STA1	STA2	STA3	STA1	STA2	STA3	média da soma das	Índice de justiça calc.
	taxa de conexão			taxa média			taxas das STAs	
comb.	11Mb/s	5.5Mb/s	1Mb/s	(bytes/s)	(bytes/s)	(bytes/s)	(bytes)	
27	.500Mb	.250Mb	.064Mb	63848	31924	8188	103960	1305000
26	.500Mb	.250Mb	.128Mb	63848	31924	16376	112148	1428000
25	.500Mb	.250Mb	.256Mb	63848	31924	32752	128524	1551000
18	.750Mb	.250Mb	.064Mb	95772	31924	8188	135884	1557000
24	.500Mb	.500Mb	.064Mb	63848	63848	8188	135884	1770000
23	.500Mb	.500Mb	.128Mb	63848	63848	16376	144072	1893000
17	.750Mb	.250Mb	.128Mb	95864	31924	16376	144164	1680000
22	.500Mb	.500Mb	.256Mb	63848	63848	32752	160448	2016000
16	.750Mb	.250Mb	.256Mb	95864	31924	32752	160540	1803000
21	.500Mb	.700Mb	.064Mb	63848	89424	8188	161460	1956000
15	.750Mb	.500Mb	.064Mb	95772	63848	8188	167808	2022000
9	1Mb	.250Mb	.064Mb	127788	31924	8188	167900	1641000
20	.500Mb	.700Mb	.128Mb	63848	89516	16376	169740	2079000
8	1Mb	.250Mb	.128Mb	127788	31924	16376	176088	1764000
14	.750Mb	.500Mb	.128Mb	95864	63848	16376	176088	2145000
19	.500Mb	.700Mb	.256Mb	63848	89516	32752	186116	2202000
7	1Mb	.250Mb	.256Mb	127788	31924	32752	192464	1887000
13	.750Mb	.500Mb	.256Mb	95864	63848	32752	192464	2268000
12	.750Mb	.700Mb	.064Mb	95772	89424	8188	193384	2208000
6	1Mb	.500Mb	.064Mb	127788	63848	8188	199824	2106000
11	.750Mb	.700Mb	.128Mb	95864	89516	16376	201756	2331000
5	1Mb	.500Mb	.128Mb	127788	63848	16376	208012	2229000
10	.750Mb	.700Mb	.256Mb	95864	89516	32752	218132	2454000
4	1Mb	.500Mb	.256Mb	127788	63848	32752	224388	2352000
3	1Mb	.700Mb	.064Mb	127788	89424	8188	225400	2292000
2	1Mb	.700Mb	.128Mb	127788	89516	16376	233680	2415000
1	1Mb	.700Mb	.256Mb	127788	89516	32752	250056	2538000

**Tabela 13. Resultados do cálculo do índice de Justiça para todas as combinações do cenário 2, segundo os métodos MLTC e MLTS.**

As taxas médias obtidas após período de estabilização, em bytes/s, para as diferentes combinações do cenário 3, estão representadas na Figura 30.



**Figura 30. Taxas médias obtidas após período de estabilização para as diferentes combinações do cenário 3.**

Nesse cenário, a combinação apontada pela política como a mais justa, além de ser a combinação base (aquela que seria aplicada caso a política de justiça não fosse implementada), também é a de maior *throughput*, motivo pelo qual a política não resultou em nenhum ganho de performance na rede.

#### 5.4. Cenário 4 – Análise de Resultados

Esse cenário apresenta uma situação onde 6 *STAs* disputam o canal. Devido ao maior número de *STAs*, esse foi o cenário que apresentou o maior número de combinações e, também, a maior complexidade para o cálculo do índice de justiça. As configurações de taxa de conexão e de tipos de serviços das *STAs* estão apresentadas na Tabela 14.

STA 1	STA 2	STA 3
<b>Taxa de conexão:</b> 5,5Mbps	<b>Taxa de conexão:</b> 2Mbps	<b>Taxa de conexão:</b> 1Mbps
<b>Tipo de serviço:</b> BEST_EFFORT	<b>Tipo de serviço:</b> VIDEO	<b>Tipo de serviço:</b> VOICE
<b>Taxas de serviço:</b> 1Mbps	<b>Taxas de serviço:</b> 750kbps	<b>Taxas de serviço:</b> 256kbps
750kbps	500kbps	128kbps
500kbps	250kbps	64kbps

STA 4	STA 5	STA 6
<b>Taxa de conexão:</b> 11Mbps	<b>Taxa de conexão:</b> 5,5Mbps	<b>Taxa de conexão:</b> 11Mbps
<b>Tipo de serviço:</b> BEST_EFFORT	<b>Tipo de serviço:</b> VIDEO	<b>Tipo de serviço:</b> VOICE
<b>Taxas de serviço:</b> 11Mbps	<b>Taxas de serviço:</b> 2Mbps	<b>Taxas de serviço:</b> 512kbps
5,5Mbps	1Mbps	256kbps
1Mbps	512kbps	128kbps

**Tabela 14. Taxas de conexão e parâmetros de serviço para o cenário 4.**

Para esse cenário foram calculados, no total, 729 combinações. O cálculo do índice de justiça, quando efetuado apenas com o método MLTS, resultou como sendo a mais justa a combinação de número 1, que é a combinação originalmente utilizada quando a política não é aplicada. Por outro lado, o cálculo do Índice de Justiça pelo método MLTC resultou como mais justa a combinação de número 49. A Tabela 15 e Tabela 16 mostram as taxas de conexão, as taxas médias obtidas e os resultados obtidos para essas 2 combinações através de simulação e do cálculo do índice de justiça.

	STA1	STA2	STA3	STA4	STA5	STA6
	taxa de conexão (Mbps)					
comb.	5,5Mb/s	2Mb/s	1Mb/s	11Mbps	5,5Mbps	11Mbps
1	1	0,7	0,256	11	2	0,512
49	1	0,7	0,128	1	1	0,512

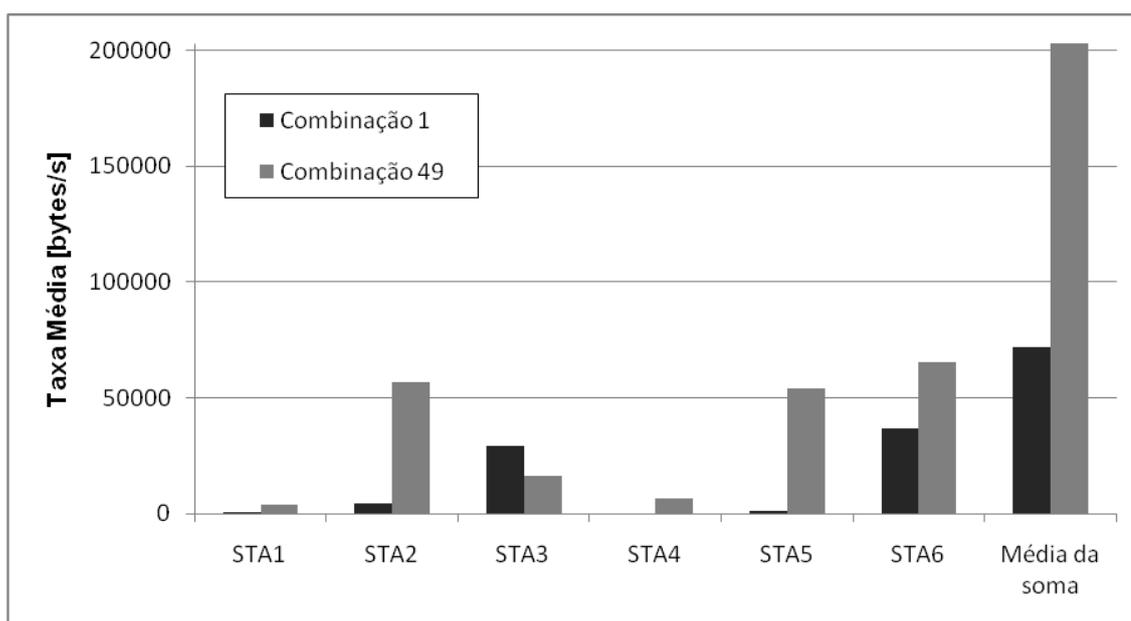
**Tabela 15. Taxas de conexão das STAs para as combinações 1 e 49 do cenário 4.**

	STA1	STA2	STA3	STA4	STA5	STA6	média da soma das taxas	Índice de Justiça calc.
	taxa média							
comb.	(bytes/s)	(bytes/s)	(bytes/s)	(bytes/s)	(bytes/s)	(bytes/s)	(bytes)	
1	92	4140	29348	0	1380	36892	71852	6499000
49	3956	56672	16376	6716	54004	65228	202952	6031000

**Tabela 16. Taxas obtidas e Resultados do cálculo do Índice de Justiça para as combinações 1 e 49 do cenário 4.**

A combinação apresentada como a mais justa no final da aplicação do algoritmo completo de justiça foi a combinação de número 49. Isso porque a combinação 1, apesar de ter apresentado um Índice de Justiça maior, é classificada como fisicamente impossível pelo algoritmo MLTC.

Em relação ao ganho de desempenho, a Figura 31 apresenta a comparação entre as taxas obtidas para o caso base (combinação 1) e o caso apontado como o mais justo pela política proposta.



**Figura 31. Taxas médias obtidas após período de estabilização para as diferentes combinações do cenário 4.**

A análise desses resultados demonstra que a aplicação da política para um ambiente com 6 STAs resultou em um ganho considerável de performance para a rede. O gráfico da Figura 31 evidencia o ganho obtido com a aplicação da política, demonstrando a vantagem de sua utilização.

A taxa média da STA1 apresentou um ganho de aproximadamente 4200%, passando de 92 bytes/s para 3956 bytes/s. De forma semelhante, a taxa média da STA2 apresentou um ganho de aproximadamente 1269%, passando de 4140 bytes/s para 56672 bytes/s.

Por outro, a taxa média da STA3 apresentou uma perda de aproximadamente 42%, passando de 29348 bytes/s para 16376 bytes/s.

A STA4, que antes da aplicação da política não recebia pacotes de dados passou a receber com uma taxa média de 6716 bytes/s. A taxa média da STA5 apresentou um ganho de aproximadamente 3813%, passando de 1380 bytes/s para 54004 bytes/s. Finalmente, a taxa média da STA6 apresentou um ganho de 76%, passando de 36892 bytes/s para 65228 bytes/s.

A média da soma das taxas das STAs teve um aumento de aproximadamente 182%, passando de 71852 bytes/s para 202952 bytes/s, evidenciando o ganho obtido com a aplicação da política.

## 6. CONCLUSÕES

Nesta dissertação foi apresentada uma proposta de política qualidade de serviço baseada na implementação do IEEE802.11e. O desenvolvimento dessa política foi realizado buscando uma solução que pudesse ser implementada em um ambiente real como, por exemplo, em um microcontrolador montado no interior de um roteador. Dessa forma, o algoritmo resultante foi implementado e programado em uma placa de desenvolvimento microcontrolada, comprovando a possibilidade de execução em um ambiente real.

Através de análises e simulações, foi comprovado que as redes IEEE802.11 realmente sofrem de uma anomalia devido ao pressuposto de igualdade na oportunidade de acesso ao meio, implementado pelo método *DCF*, que desconsidera as características físicas das *STAs*. Foi demonstrado também que as redes IEEE802.11e se tornam ineficientes, quando se considera seu potencial no tocante aos recursos totais disponíveis na rede, na priorização por tipo de serviço, sem considerar as características do meio físico, causando uma segunda anomalia.

Os resultados obtidos demonstram que através de um controle eficiente do acesso das *STAs* no método *EDCA* é possível obter uma maior justiça na rede, sem que para isso sejam necessárias a despriorização de qualquer *STA*.

Entretanto, é importante salientar que um problema observado na implementação da política proposta é que a mesma foi concebida para ser aplicada no momento da disputa da *STA* pelo canal. Assim, os algoritmos propostos levam em consideração apenas parâmetros teóricos para o cálculo do índice de justiça. Por conseguinte, não está sendo considerada a taxa real das *STAs*, pois essa taxa é resultante dos parâmetros configurados nas *STAs* e que resultaram da aplicação da própria política. Assim sendo, é possível que, em algumas situações, a política possa interpretar uma situação como sendo a “mais justa”, quando, na verdade, outras situações poderiam apresentar um índice de justiça ainda maior.

Por outro lado, os resultados obtidos nesse trabalho também demonstram que a utilização da política não traz nenhuma desvantagem à rede; pelo contrário, todos os cenários analisados resultaram em um ganho ou, no mínimo, uma condição igual à condição original, sem utilização da política.

Como sugestão de trabalhos futuros nessa área, pode-se destacar o estudo e a implementação de métodos e técnicas que possibilitem a inclusão, entre as variáveis consideradas para a elaboração de um índice de justiça, da taxa real de atendimento às *STAs* durante a operação real da rede.

## 7. REFEFÊNCIAS

BRANQUINHO, O. C.; REGGIANI, N.; FERREIRA, D.M. *Mitigating 802.11 Mac Anomaly Using SNR to Control Backoff Contention Window*. In: IEEE Computer Society, v. 4, p. 55-61, 2006.

GUIRARDELLO, M. Política de QoS com Priorização de Acesso ao Meio para Redes IEEE 802.11. 2008. 104f. Dissertação (Mestrado profissional em Gestão de Redes de Telecomunicações) – Pós-Graduação em Engenharia Elétrica, Centro de Ciências Exatas, Ambientais e de Tecnologias, Pontifícia Universidade Católica de Campinas, Campinas, 2008.

HEUSSE, M. et al. *Performance Anomaly of 802.11b*. Em: IEEE INFOCOM 2003, Março 2003.

IEEE802.11e. *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 8: Medium Access Control (MAC) Quality of Service Enhancements* - Novembro, 2005.

IEEE802.11. *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* – Jun, 2007.

JAIN, R. K.; CHIU, D. W.; HAWKES, W. *A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems* – Set 1984.

MORALES, M. A. Política de priorização de acesso de STAs com taxas diferentes para redes 802.11 baseada na SNR. 2008. 75f. Dissertação (Mestrado profissional em Gestão de redes Telecomunicações) – Pós-Graduação em Engenharia Elétrica, Centro de Ciências Exatas, Ambientais e de Tecnologias, Pontifícia Universidade Católica de Campinas, Campinas, 2008.

NS2. Disponível em <http://www.isi.edu/nsnam/ns/>. Acesso em Dezembro de 2009.

PRASAD, A. R.; PRASAD, N. R. *802.11 WLANs and IP Networking*. Ed. Artech House, 2005.

RAMOS, N.; PANIGRAHI, D.; DEY, S. *Quality of Service Provisioning in 802.11e Networks: Challenges, Approaches, and Future Directions*. *IEEE Network* Jun/Ago 2005.

RAPPAPORT, T. S. *Wireless Communications: Principles and Practice*. New Jersey: Prentice Hall.

ZYREN, J.; PETRICK, A. *Tutorial on Basic Link Budget Analysis*. In: Application Note, AN9804.1, INTERSIL, 1998.

## **ANEXO A**

## 8. ANEXO A – CÓDIGO DA PLATAFORMA DE POLÍTICA

### 8.1. Trecho do programa 'main.c', utilizado pelo PIC32.

```

#include <p32xxxx.h>
#include <plib.h>
#include "stdio.h"
#include "defines.h"
#include "calculation.h"

// Config settings
// POSCMOD = HS, FNOSC = PRIPLL, FWDTEN = OFF
// PLLDIV = DIV_2, PLLMUL = MUL_16
// PBDIV = 8 (default)
// Main clock = 8MHz /2 * 16 = 80MHz
// Peripheral clock = 80MHz /8 = 10MHz

// Configuration Bit settings
// SYSCLK = 80 MHz (8MHz Crystal/ FPLLDIV * FPLLMUL / FPLLODIV)
// PBCLK = 10 MHz
// Primary Osc w/PLL (XT+,HS+,EC+PLL)
// WDT OFF
// Other options are don't care
//
#pragma config FPLLMUL = MUL_20, FPLLDIV = DIV_2, FPLLODIV = DIV_1, FWDTEN = OFF
#pragma config POSCMOD = HS, FNOSC = PRIPLL, FPBDIV = DIV_8

// Define o baudrate da serial
#define BAUDRATE 9600

// Set the right UART for stdio output: UART1 for MPLAB SIM, UART2 for Explorer 16
#define OUT_BUSY BusyUART2
#define OUT_WRITE WriteUART2
#define IN_READY DataRdyUART2
#define IN_READ ReadUART2
#define IO_INIT OpenUART2

stSta sStas[NUM_STAS];
stMelhorCenario sMelhoresCenarios[30];
long lContadorMelhoresCenarios;

/*****//*!
 @brief Redirect stdio calls to our UART
 @return
 *****/
int _mon_putc( int c)
{
    while(OUT_BUSY())
    {
    }

    OUT_WRITE(c);
    return c;
}

/*****//*!
 @brief
 @return
 *****/
char _mon_getc( void)
{
    while (!IN_READY())

```

```

    {
    }
    IN_READ();
}

/*****//*/
@brief
@return
void InitSTAs(void)
{
    int i;

    i=0;
    while(i<NUM_STAS)
    {
/*
// CENARIO 1 DA DISSERTACAO
// STA_A
    sStas[i].lTaxaConexao = TAXA_1Mbps;
    sStas[i].sServico.iTypeOfService = PRIORITY_BEST_EFFORT;
    sStas[i].sServico.lServiceRate[0] = 1000000L;
    sStas[i].sServico.lServiceRate[1] = 750000L;
    sStas[i].sServico.lServiceRate[2] = 500000L;
    i++;

// STA_B
    sStas[i].lTaxaConexao = TAXA_11Mbps;
    sStas[i].sServico.iTypeOfService = PRIORITY_VIDEO;
    sStas[i].sServico.lServiceRate[0] = 700000L;
    sStas[i].sServico.lServiceRate[1] = 500000L;
    sStas[i].sServico.lServiceRate[2] = 250000L;
    i++;

// STA_C
    sStas[i].lTaxaConexao = TAXA_5M5bps;
    sStas[i].sServico.iTypeOfService = PRIORITY_VOICE;
    sStas[i].sServico.lServiceRate[0] = 256000L;
    sStas[i].sServico.lServiceRate[1] = 128000L;
    sStas[i].sServico.lServiceRate[2] = 64000L;
    i++;
*/

/*
// CENARIO 2 DA DISSERTACAO
// STA_A
    sStas[i].lTaxaConexao = TAXA_1Mbps;
    sStas[i].sServico.iTypeOfService = PRIORITY_BEST_EFFORT;
    sStas[i].sServico.lServiceRate[0] = 11000000L;
    sStas[i].sServico.lServiceRate[1] = 5500000L;
    sStas[i].sServico.lServiceRate[2] = 1000000L;
    i++;

// STA_B
    sStas[i].lTaxaConexao = TAXA_5M5bps;
    sStas[i].sServico.iTypeOfService = PRIORITY_VIDEO;
    sStas[i].sServico.lServiceRate[0] = 2000000L;
    sStas[i].sServico.lServiceRate[1] = 1000000L;
    sStas[i].sServico.lServiceRate[2] = 512000L;
    i++;

// STA_C
    sStas[i].lTaxaConexao = TAXA_11Mbps;
    sStas[i].sServico.iTypeOfService = PRIORITY_VOICE;
    sStas[i].sServico.lServiceRate[0] = 512000L;
    sStas[i].sServico.lServiceRate[1] = 256000L;
    sStas[i].sServico.lServiceRate[2] = 128000L;
    i++;
*/

/*
// CENARIO 3 DA DISSERTACAO
// STA_A
    sStas[i].lTaxaConexao = TAXA_11Mbps;
    sStas[i].sServico.iTypeOfService = PRIORITY_BEST_EFFORT;
    sStas[i].sServico.lServiceRate[0] = 1000000L;

```

```

sStas[i].sServico.lServiceRate[1] = 750000L;
sStas[i].sServico.lServiceRate[2] = 500000L;
i++;

// STA_B
sStas[i].lTaxaConexao = TAXA_5M5bps;
sStas[i].sServico.iTypeOfService = PRIORITY_VIDEO;
sStas[i].sServico.lServiceRate[0] = 700000L;
sStas[i].sServico.lServiceRate[1] = 500000L;
sStas[i].sServico.lServiceRate[2] = 250000L;
i++;

// STA_C
sStas[i].lTaxaConexao = TAXA_1Mbps;
sStas[i].sServico.iTypeOfService = PRIORITY_VOICE;
sStas[i].sServico.lServiceRate[0] = 256000L;
sStas[i].sServico.lServiceRate[1] = 128000L;
sStas[i].sServico.lServiceRate[2] = 640000L;
i++;

*/

// CENARIO 4 DA DISSERTACAO (6 estacoes)
// STA_A
sStas[i].lTaxaConexao = TAXA_5M5bps;
sStas[i].sServico.iTypeOfService = PRIORITY_BEST_EFFORT;
sStas[i].sServico.lServiceRate[0] = 1000000L;
sStas[i].sServico.lServiceRate[1] = 750000L;
sStas[i].sServico.lServiceRate[2] = 500000L;
i++;

// STA_B
sStas[i].lTaxaConexao = TAXA_2Mbps;
sStas[i].sServico.iTypeOfService = PRIORITY_VIDEO;
sStas[i].sServico.lServiceRate[0] = 700000L;
sStas[i].sServico.lServiceRate[1] = 500000L;
sStas[i].sServico.lServiceRate[2] = 250000L;
i++;

// STA_C
sStas[i].lTaxaConexao = TAXA_1Mbps;
sStas[i].sServico.iTypeOfService = PRIORITY_VOICE;
sStas[i].sServico.lServiceRate[0] = 256000L;
sStas[i].sServico.lServiceRate[1] = 128000L;
sStas[i].sServico.lServiceRate[2] = 64000L;
i++;

// STA_D
sStas[i].lTaxaConexao = TAXA_11Mbps;
sStas[i].sServico.iTypeOfService = PRIORITY_BEST_EFFORT;
sStas[i].sServico.lServiceRate[0] = 11000000L;
sStas[i].sServico.lServiceRate[1] = 5500000L;
sStas[i].sServico.lServiceRate[2] = 1000000L;
i++;

// STA_E
sStas[i].lTaxaConexao = TAXA_5M5bps;
sStas[i].sServico.iTypeOfService = PRIORITY_VIDEO;
sStas[i].sServico.lServiceRate[0] = 2000000L;
sStas[i].sServico.lServiceRate[1] = 1000000L;
sStas[i].sServico.lServiceRate[2] = 512000L;
i++;

// STA_F
sStas[i].lTaxaConexao = TAXA_11Mbps;
sStas[i].sServico.iTypeOfService = PRIORITY_VOICE;
sStas[i].sServico.lServiceRate[0] = 512000L;
sStas[i].sServico.lServiceRate[1] = 256000L;
sStas[i].sServico.lServiceRate[2] = 128000L;
i++;

}

/*
// STA_D

```

```

sStas[0].lTaxaConexao = TAXA_1Mbps;
sStas[0].sServico.iTypeOfService = PRIORITY_VIDEO;
sStas[0].sServico.lServiceRate[0] = 2000000L;
sStas[0].sServico.lServiceRate[1] = 1000000L;
sStas[0].sServico.lServiceRate[2] = 5120000L;

// STA_E
sStas[1].lTaxaConexao = TAXA_1Mbps;
sStas[1].sServico.iTypeOfService = PRIORITY_BEST_EFFORT;
sStas[2].sServico.lServiceRate[0] = 512000L;
sStas[2].sServico.lServiceRate[1] = 256000L;
sStas[2].sServico.lServiceRate[2] = 128000L;

// STA_F
sStas[0].lTaxaConexao = TAXA_5M5bps;
sStas[0].sServico.iTypeOfService = PRIORITY_BEST_EFFORT;
sStas[0].sServico.lServiceRate[0] = 11000000L;
sStas[0].sServico.lServiceRate[1] = 5500000L;
sStas[0].sServico.lServiceRate[2] = 1000000L;
*/
}

/*****//*/
@brief
@return
*****/
int main(void)
{
    int pbFreq;
    int i;
    int iServiceCounter;
    int iNumeroDeEstacoes;
    int iInteracoes = 0;
    int iLoop;
    long lMaiorPontuacao;
    long lPontuacaoAtual;
    int iMelhorCenario;
    float fTempoAcesso;

    //
    // INICIALIZACAO DO MICROCONTROLADOR
    //

    // disable JTAG port
    DDPCONbits.JTAGEN = 0;

    // This function automatically configures the device for maximum
    // performance for a given system clock frequency
    pbFreq = SYSTEMConfig(8000000L, SYS_CFG_WAIT_STATES | SYS_CFG_PCACHE);

    //
    TRISA = 0; // all PORTA as output
    T1CON = 0x8030; // TMR1 on, prescale 1:256 PB

    // Inicializa a UART
    IO_INIT
    (
        UART_EN|UART_NO_PAR_8BIT|UART_1STOPBIT,UART_RX_ENABLE|UART_TX_ENABLE,
        (pbFreq/16/BAUDRATE)-1
    );

    PORTA = 0x00;

    //
    // LOOP PRINCIPAL
    //
    while(1)
    {
        //
        #if PRINTF_ENABLED == 1
            printf("\n\r", NUM_STAS);
            printf("Numero de estacoes: %i\n\r", NUM_STAS);
            printf("Numero de taxas por servico: %i\n\r\n\r", NUM_SERVICES_RATES);
        #endif

        // Inicializa as estacoes

```

```

InitSTAs();
lMaiorPontuacao = 0x00000000;
iMelhorCenario = 0x0000;
lContadorMelhoresCenarios = 0x00000000;
iInteracoes = 1;

// Inicializa os contadores de teste de todas as estacoes
for(i=0;i<NUM_STAS;i++)
{
    sStas[i].iServiceRateIndex = 0;
}

// Realizacao dos testes
iLoop = 1;
while(iLoop)
{
    fTempoAcesso = 0;

    //
    // FAZ O CALCULO DO TEMPO DE UTILIZACAO DO CANAL
    // (MLTC - METODO DA LIMITACAO DO ACESSO PELO TEMPO
    // DE UTILIZACAO DO CANAL)
    //
    for( i=0; i< NUM_STAS;i++)
    {
        // verifica se o cenario atual pode ser fisicamente obtido
        fTempoAcesso+=
            (float)((float)sStas[i].sServico.lServiceRate_
                _[sStas[i].iServiceRateIndex] /
                (float)sStas[i].lTaxaConexao);
    }
    /*
    //
    // FAZ A VERIFICACAO DA TAXA DE CONEXAO
    // (MLTS - METODO DA LIMITACAO DAS TAXAS DE SERVICO)
    //
    // Verifica se em alguns dos casos a taxa de servico eh maior
    // que a taxa de conexao. Se for, limita a taxa de servico
    // ao valor da taxa de conexao
    fTempoAcesso = 0;
    for( i=0; i< NUM_STAS;i++)
    {
        if( sStas[i].sServico.lServiceRate[sStas[i].iServiceRateIndex] _
            _> sStas[i].lTaxaConexao )
        {
            sStas[i].sServico.lServiceRate_
                [sStas[i].iServiceRateIndex] = sStas[i].lTaxaConexao;
        }
    }
    */

    if( fTempoAcesso < 1 )
    {
        // calcula o indice de justica para todas as estacoes,
        // considerando o cenario atual
        lPontuacaoAtual = 0x00000000;
        for( i=0; i< NUM_STAS;i++)
        {
            PORTA = 0xFF;
            lPontuacaoAtual += lCalculaPontosSta(&sStas[i],_
                _sStas[i].iServiceRateIndex);
            PORTA = 0x00;
        }
        #if PRINTF_ENABLED == 1
        printf("[%i] Total de pontos %ld\n\r\n\r", iInteracoes,
            lPontuacaoAtual);
        #endif

        if(lPontuacaoAtual >= lMaiorPontuacao)
        {
            lMaiorPontuacao = lPontuacaoAtual;
            iMelhorCenario = iInteracoes;
            sMelhoresCenarios[lContadorMelhoresCenarios]_
                .lCombinacao = iInteracoes;
            sMelhoresCenarios[lContadorMelhoresCenarios]_

```

```

        .lPontuacao = lPontuacaoAtual;

        sMelhoresCenarios[lContadorMelhoresCenarios]_
        _lTaxa[0] = sStas[0].sServico.lServiceRate
        _[sStas[0].iServiceRateIndex];
        sMelhoresCenarios[lContadorMelhoresCenarios]_
        _lTaxa[1] = sStas[1].sServico.lServiceRate
        _[sStas[1].iServiceRateIndex];
        sMelhoresCenarios[lContadorMelhoresCenarios]_
        _lTaxa[2] = sStas[2].sServico.lServiceRate
        _[sStas[2].iServiceRateIndex];
        sMelhoresCenarios[lContadorMelhoresCenarios]_
        _lTaxa[3] = sStas[3].sServico.lServiceRate
        _[sStas[3].iServiceRateIndex];
        sMelhoresCenarios[lContadorMelhoresCenarios]_
        _lTaxa[4] = sStas[4].sServico.lServiceRate
        _[sStas[4].iServiceRateIndex];
        sMelhoresCenarios[lContadorMelhoresCenarios]_
        _lTaxa[5] = sStas[5].sServico.lServiceRate
        _[sStas[5].iServiceRateIndex];
        lContadorMelhoresCenarios++;
    }
}

//
iInteracoes++;

// incrementa os contadores
for( i=(NUM_STAS-1);i>=0;i--)
{
    sStas[i].iServiceRateIndex++;

    if( sStas[0x00].iServiceRateIndex >= NUM_SERVICES_RATES )
    {
        // todas as interacoes ja foram realizadas
        iLoop = 0x00;
        break;
    }

    if(sStas[i].iServiceRateIndex >= NUM_SERVICES_RATES)
    {
        sStas[i].iServiceRateIndex = 0;
    }
    else
    {
        break;
    }
}

//
#if PRINTF_ENABLED == 1
    printf("Numero de testes realizados: %i\n\r", iInteracoes);
#endif

lMaiorPontuacao = 0x00000000;
}

} // main

```

## **ANEXO B**

## 9. ANEXO B – CÓDIGO DA PLATAFORMA DE SIMULAÇÃO

### 9.1. Trecho do *script* utilizado pelo NS2.

```

set val(chan)           Channel/WirelessChannel    ;# Channel Type
set val(prop)           Propagation/Shadowing      ;# radio-propagation model
set val(netif)          Phy/WirelessPhy           ;# network interface type
set val(mac)            Mac/802_11                ;# MAC type
set val(ifq)            Queue/DropTail/PriQueue   ;# interface queue type
set val(ll)            LL                          ;# link layer type
set val(ant)            Antenna/OmniAntenna       ;# antenna model
set val(ifqlenSTA)      50                        ;# max packet in ifq
set val(ifqlenBSS)     1000000                   ;# max packet in ifq
set val(nn)             4                         ;# number of mobilenodes
#set val(rp)            DSDV                      ;# routing protocol
set val(rp)            AODV                       ;# routing protocol

# =====
# Ajuste dos parametros da rede sem fio
# =====
Propagation/Shadowing set pathlossExp_          2
Propagation/Shadowing set std_db_              0
Propagation/Shadowing set dist0_              1.0
Propagation/Shadowing set seed_               0

Antenna/OmniAntenna set X_                    0.0
Antenna/OmniAntenna set Y_                    0.0
Antenna/OmniAntenna set Z_                    1.5
Antenna/OmniAntenna set Gt_                   1
Antenna/OmniAntenna set Gr_                   1

Phy/WirelessPhy set L_                        1.0
Phy/WirelessPhy set freq_                     2.4e9
Phy/WirelessPhy set bandwidth_               11Mb
Phy/WirelessPhy set Pt_                      0.2818 ;# Potencia em W
#Phy/WirelessPhy set Pt_                     0.00009 ;# Potencia em W
Phy/WirelessPhy set CPTthresh_               10 ;# limiar de colisao dado em dB
Phy/WirelessPhy set CSTthresh_              3.1622777e-14
Phy/WirelessPhy set RXTthresh_              3.1622777e-13
Phy/WirelessPhy set debug_                  false

Mac/802_11 set dataRate_                     11Mb ;#definida durante o programa
Mac/802_11 set basicRate_                    1Mb
Mac/802_11 set CWMin_                        31
Mac/802_11 set CWMax_                       1023
Mac/802_11 set SlotTime_                    0.000020 ;# 20us
Mac/802_11 set SIFS_                        0.000010 ;# 10us
Mac/802_11 set PreambleLength_              72 ;# preambulo curto
Mac/802_11 set PLCPHeaderLength_           48 ;# 48 bits
Mac/802_11 set PLCPDataRate_               1.0e6 ;# 1Mbps
Mac/802_11 set RTSThreshold_               4000 ;# bytes
Mac/802_11 set ShortRetryLimit_            7 ;# retransmissions
Mac/802_11 set LongRetryLimit_             4 ;# retransmissions
Mac/802_11 set bandwidth_                  11.0Mb

set f0 [open out0.tr w]
set f1 [open result.txt w]

#
# configura os valores de testes
#
set best_effort_rate 1Mb
set video_rate       1Mb

```

```

set voice_rate          .100Mb

# Initialize Global Variables
set ns_                  [new Simulator]
set tracefd              [open wireless_mitf.tr w]
$ns_ trace-all $tracefd

# Ajusta a topologia da simulacao
set val(x)                500
set val(y)                10000
set topo                  [new Topography]
$topo load_flatgrid $val(x) $val(y)

# Create God
create-god $val(nn)

# New API to config node:
# 1. Create channel (or multiple-channels);
# 2. Specify channel in node-config (instead of channelType);
# 3. Create nodes for simulations.

# Create channel #1
set chan_1_ [new $val(chan)]

# CRIA AS STAS
# Todas as STAs sao criadas com as mesmas configuracoes, com excessao das taxas
$ns_ node-config -adhocRouting $val(rp) \
                -llType $val(ll) \
                -macType $val(mac) \
                -ifqType $val(ifq) \
                -ifqLen $val(ifqlenSTA) \
                -antType $val(ant) \
                -propType $val(prop) \
                -phyType $val(netif) \
                -topoInstance $topo \
                -agentTrace ON \
                -routerTrace ON \
                -macTrace ON \
                -movementTrace OFF \
                -channel $chan_1_

# Uma vez que as configuracoes comuns de todas as STAs foram definidas, define as
# configuracoes individuais e cria cada uma das STAs

# NO 0
Mac/802_11 set CWMin_          31
Mac/802_11 set CWMax_         1023
Mac/802_11 set dataRate_      11Mb
set node_(0) [$ns_ node]
$node_(0) random-motion 0

# NO 1
Mac/802_11 set CWMin_          15
Mac/802_11 set CWMax_         1023
Mac/802_11 set dataRate_      1Mb
set node_(1) [$ns_ node]
$node_(1) random-motion 0

# NO 2
Mac/802_11 set CWMin_          3
Mac/802_11 set CWMax_         7
Mac/802_11 set dataRate_      11Mb
set node_(2) [$ns_ node]
$node_(2) random-motion 0

# CRIA A BS
$ns_ node-config -adhocRouting $val(rp) \
                -llType $val(ll) \
                -macType $val(mac) \
                -ifqType $val(ifq) \
                -ifqLen $val(ifqlenBSS) \
                -antType $val(ant) \
                -propType $val(prop) \
                -phyType $val(netif) \
                -topoInstance $topo \

```

```

        -agentTrace ON \
        -routerTrace ON \
        -macTrace ON \
        -movementTrace OFF \
        -channel $chan_1_

Mac/802_11 set CWMin_                1
Mac/802_11 set CWMax_                31
Mac/802_11 set dataRate_             11Mb
set node_(3) [$ns_ node]
$node_(3) random-motion 0

#
# Provide initial (X,Y, for now Z=0) co-ordinates for mobilenodes
#
$node_(0) set X_ 100
$node_(0) set Y_ 150.0
$node_(0) set Z_ 0.0

$node_(1) set X_ 140.0
$node_(1) set Y_ 150.0
$node_(1) set Z_ 0.0

$node_(2) set X_ 180.0
$node_(2) set Y_ 150.0
$node_(2) set Z_ 0.0

$node_(3) set X_ 160.0
$node_(3) set Y_ 200.0
$node_(3) set Z_ 0.0

# define as configuracoes do script

#Cria e inicializa as variaveis globais
array set arrayCbrRate {1 10Mb 2 9Mb 3 8Mb 4 7Mb 5 6Mb 6 5Mb 7 4Mb 8 3Mb 9 2Mb 10 1Mb \
    11 0.9Mb 12 0.8Mb 13 0.7Mb 14 0.7Mb 15 0.5Mb 16 0.4Mb 17 0.3Mb \
    18 0.2Mb 19 0.1Mb 20 0}

array set arrayBestEffortRates {1 1Mb 2 .750Mb 3 .500Mb}
array set arrayVideoRates {1 .700Mb 2 .500Mb 3 .250Mb}
array set arrayVoiceRates {1 .256Mb 2 .128Mb 3 .064Mb}

# indica quais as medias que deverao ser calculadas
array set cbrAverageEnabled {1 0 2 0 3 0 4 0}

# ponteiro que indica qual a posicao da matriz de medias que deve ser escrita
array set cbrAverageIndex {1 1 2 1 3 1 4 1}

# matriz de medias
array set cbr1AverageValues {1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0}
# matriz de medias
array set cbr2AverageValues {1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0}
# matriz de medias
array set cbr3AverageValues {1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0}
# matriz de medias (contem a media das somas)
array set averageSumValues {1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0}

set cbr1RateIndex 1
set cbr2RateIndex 1
set cbr3RateIndex 1

set udp1 [new Agent/UDP]
set udp2 [new Agent/UDP]
set udp3 [new Agent/UDP]
set udp4 [new Agent/UDP]

# set udp2 [new Agent/UDP]
# $udp2 set packetSize_ 1000; #(max=1000)
set cbr1 [new Application/Traffic/CBR]
set cbr2 [new Application/Traffic/CBR]
set cbr4 [new Application/Traffic/CBR]
set cbr3 [new Application/Traffic/CBR]

```

```

set crbrate 10Mb

# BEST EFFORT
$cbr1 set packetSize_ 900
$cbr1 set rate_ $best_effort_rate
$cbr1 set random_ 0
$cbr1 attach-agent $udp1

# VIDEO
$cbr2 set packetSize_ 900
$cbr2 set rate_ $video_rate
$cbr2 set random_ 0
$cbr2 attach-agent $udp2

# VOZ
$cbr3 set packetSize_ 900
$cbr3 set rate_ $voice_rate
$cbr3 set random_ 0
$cbr3 attach-agent $udp3

set UdpSink1 [new Agent/LossMonitor]
set UdpSink2 [new Agent/LossMonitor]
set UdpSink3 [new Agent/LossMonitor]

$ns_ attach-agent $node_(0) $udp1
$ns_ attach-agent $node_(1) $udp2
$ns_ attach-agent $node_(2) $udp3
$ns_ attach-agent $node_(3) $UdpSink1
$ns_ attach-agent $node_(3) $UdpSink2
$ns_ attach-agent $node_(3) $UdpSink3

$ns_ connect $udp1 $UdpSink1
$ns_ connect $udp2 $UdpSink2
$ns_ connect $udp3 $UdpSink3

#
# aqui estao descritos todos os procedimentos da simulacao
#
$ns_ at 0.0 "record"

set timeOffset 0

#for {set beCounter 1} {$beCounter < 3} {incr beCounter} {

$ns_ at [expr $timeOffset + 0.1] "$cbr1 stop"
$ns_ at [expr $timeOffset + 0.1] "$cbr2 stop"
$ns_ at [expr $timeOffset + 10.0] "$cbr1 start"
$ns_ at [expr $timeOffset + 30.0] "$cbr2 start"

set i 0

$ns_ at [expr $timeOffset + 50.0] "$cbr2 stop"
$ns_ at [expr $timeOffset + 70.0] "$cbr1 stop"

for {set i 0} {$i < $val(nn) } {incr i} {
    $ns_ at [expr $timeOffset + 75.0] "$node_($i) reset";
}

set timeOffset [expr $timeOffset + 80]

#}

$ns_ at [expr $timeOffset + 00.0] "puts \"NS EXITING...\" ; $ns_ halt"
$ns_ at [expr $timeOffset + 00.1] "stop"

#
# Tell nodes when the simulation ends
#
for {set i 0} {$i < $val(nn) } {incr i} {
    $ns_ at [expr $timeOffset + 02.0] "$node_($i) reset";
}

#}

```

```

#-----
#
#-----
proc fnAjustaTaxa {} {
    global arrayCbrRate cbr3RateIndex cbr1 cbr2 cbr3
    global cbrAverageEnabled

    #Get an instance of the simulator
    set ns [Simulator instance]

    #Set the time after which the procedure should be called again
    set time 10
    set now [$ns now]
    puts "index:$cbr3RateIndex valor:$arrayCbrRate($cbr3RateIndex)"
    $cbr3 set rate_ $arrayCbrRate($cbr3RateIndex)
    set cbr3RateIndex [expr ($cbr3RateIndex + 1)]

    ## $ns at [expr $now+$time] "fnAjustaTaxa"
}

#-----
#
#-----
proc fnCalculaMediacbr1 {} {
    global cbrAverageEnabled(1)
    set cbrAverageEnabled(1) 1
}

#-----
#
#-----
proc fnCalculaMediacbr2 {} {
    global cbr2AverageEnabled
    set cbr2AverageEnabled 1
}

#-----
#
#-----
proc fnCalculaMediacbr3 {} {
    global cbr3AverageEnabled
    set cbr3AverageEnabled 1
}

#-----
#
#-----
proc record {} {
    global udp1 udp2 udp3 UdpSink1 UdpSink2 UdpSink3 f0 f1 cbr1 cbr2 cbr3
    global cbrAverageIndex cbrAverageEnabled cbr1AverageValues cbr2AverageValues
    cbr3AverageValues averageSumValues
    global best_effort_rate video_rate voice_rate

    #Get an instance of the simulator
    set ns [Simulator instance]
    #Get the current time
    set now [$ns now]
    #Set the time after which the procedure should be called again
    set time 1

    set media1      0
    set media2      0
    set media3      0
    set mediaSoma   0

    #How many bytes have been received by the traffic sinks?
    set bytes1 [$UdpSink1 set bytes_]

```

```

set pacotes1 [$UdpSink1 set npkts_]
set pacotes_perdidos1 [$UdpSink1 set nlost_]

set bytes2 [$UdpSink2 set bytes_]
set pacotes2 [$UdpSink2 set npkts_]
set pacotes_perdidos2 [$UdpSink2 set nlost_]

set bytes3 [$UdpSink3 set bytes_]
set pacotes3 [$UdpSink3 set npkts_]
set pacotes_perdidos3 [$UdpSink3 set nlost_]

if { $cbrAverageEnabled(1) > 0 } {

    #puts "calculando a media cbr1"

    # Armazena o valor para contagem da media
    set cbr1AverageValues($cbrAverageIndex(1)) $bytes1
    #puts "valor $bytes1 armazenado na posicao $cbrAverageIndex(1) "
    set cbrAverageIndex(1) [expr ($cbrAverageIndex(1)+1)]

    # Faz o calcula da media a cada dez medidas
    if { $cbrAverageIndex(1) > 11 } {

        # soma todos os valores da matriz de medias
        for {set i 1} {$i < 11} {incr i} {
            set medial [expr ($medial + $cbr1AverageValues($i) )]
        }

        # divide o valor somado pelo numero de amostras, e calcula a media
        set medial [expr ($medial / 10)]
        #puts "media cbr1: $medial"

        # volta o ponteiro para posicao inicial do buffer, e recomeca a contage
        set cbrAverageIndex(1) 1
        #
        set cbrAverageEnabled(1) 0
    }
}

if { $cbrAverageEnabled(2) > 0 } {

    #puts "calculando a media cbr2"

    # Armazena o valor para contagem da media
    set cbr2AverageValues($cbrAverageIndex(2)) $bytes2
    set cbrAverageIndex(2) [expr ($cbrAverageIndex(2)+1)]

    # Faz o calcula da media a cada dez medidas
    if { $cbrAverageIndex(2) > 11 } {

        # soma todos os valores da matriz de medias
        for {set i 1} {$i < 11} {incr i} {
            set media2 [expr ($media2 + $cbr2AverageValues($i) )]
        }

        # divide o valor somado pelo numero de amostras, e calcula a media
        set media2 [expr ($media2 / 10)]
        #puts "media cbr2: $media2"

        # volta o ponteiro para posicao inicial do buffer, e recomeca a contage
        set cbrAverageIndex(2) 1
        #
        set cbrAverageEnabled(2) 0
    }
}

if { $cbrAverageEnabled(3) > 0 } {

    #puts "calculando a media cbr3"

    # Armazena o valor para contagem da media
    set cbr3AverageValues($cbrAverageIndex(3)) $bytes3
    set cbrAverageIndex(3) [expr ($cbrAverageIndex(3)+1)]

    # Faz o calcula da media a cada dez medidas
    if { $cbrAverageIndex(3) > 11 } {

        # soma todos os valores da matriz de medias

```

```

        for {set i 1} {$i < 11} {incr i} {
            set media3 [expr ($media3 + $cbr3AverageValues($i) )]
        }

        # divide o valor somado pelo numero de amostras, e calcula a media
        set media3 [expr ($media3 / 10)]
        #puts "media cbr3: $media3"

        # volta o ponteiro para posicao inicial do buffer, e recomeca a contage
        set cbrAverageIndex(3) 1
        #
        set cbrAverageEnabled(3) 0
    }
}

if { $cbrAverageEnabled(4) > 0 } {

# Calcula a media das somas
# Armazena o valor para contagem da media
set averageSumValues($cbrAverageIndex(4)) [expr $bytes1+$bytes2+$bytes3]
set cbrAverageIndex(4) [expr ($cbrAverageIndex(4)+1)]

# Faz o calcula da media a cada dez medidas
if { $cbrAverageIndex(4) > 11 } {

    # soma todos os valores da matriz de medias
    for {set i 1} {$i < 11} {incr i} {
        set mediaSoma [expr ($mediaSoma + $averageSumValues($i) )]
    }

    # divide o valor somado pelo numero de amostras, e calcula a media
    set mediaSoma [expr ($mediaSoma / 10)]
    #puts "mediaSoma: $mediaSoma"

    # volta o ponteiro para posicao inicial do buffer, e recomeca a contage
    set cbrAverageIndex(4) 1
    #
    set cbrAverageEnabled(4) 0
}

}

set total [expr ($bytes1+$bytes2+$bytes3)]
puts "$now\t$best_effort_rate\t$video_rate\t\t[expr ($bytes1/1)]\t[expr
($bytes2/1)]\t$total\t"

#      puts $f2 "$now [expr $bw2/$time*8/1000000]"

#Reset the bytes_ values on the traffic sinks
$UdpSink1 set bytes_ 0
$UdpSink1 set npkts_ 0
$UdpSink1 set nlost_ 0
$UdpSink2 set bytes_ 0
$UdpSink2 set npkts_ 0
$UdpSink2 set nlost_ 0
$UdpSink3 set bytes_ 0
$UdpSink3 set npkts_ 0
$UdpSink3 set nlost_ 0

#Re-schedule the procedure
$ns at [expr $now+$time] "record"
}

#-----
#
#-----
proc stop {} {
    global ns_ tracefd f0
    $ns_ flush-trace
    close $tracefd
}

```

```
        close $f0;
        exit 0;
    }

    puts "Starting Simulation..."
    $ns_ run
```