

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

RT-WiFi: Uma Arquitetura para Comunicação de Tempo-Real em Redes IEEE 802.11 Infraestruturadas

Robson Costa



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Programa Doutoral em Engenharia Informática

Orientador: Paulo Portugal (Prof. Dr.)

Co-orientador: Ricardo Moraes (Prof. Dr.)

2013

RT-WiFi: Uma Arquitetura para Comunicação de Tempo-Real em Redes IEEE 802.11 Infraestruturadas

Robson Costa

Programa Doutoral em Engenharia Informática

Aprovado em provas públicas pelo Júri:

Presidente: Eugénio da Costa Oliveira (Prof. Dr.)

Vogal: Paulo Bacelar Reis Pedreiras (Prof. Dr.)

Vogal: Ana Cristina Costa Aguiar (Profa. Dra.)

Vogal: Joaquim José de Castro Ferreira (Prof. Dr.)

Vogal: Pedro Alexandre Guimarães Lobo Ferreira Souto (Prof. Dr.)

Vogal: Ricardo Alexandre Reinaldo de Moraes (Prof. Dr.)

Vogal: Paulo José Lopes Machado Portugal (Prof. Dr.)

Prof. Dr. Paulo José Lopes Machado Portugal (Orientador)

*"Para Ana Paula e Yasmin, por estarem sempre ao meu lado.
Sem vocês este trabalho não teria sido terminado."*

Agradecimentos

Inicialmente eu gostaria de agradecer a Deus, por sempre iluminar meu caminho e colocar pessoas maravilhosas nele.

Por diferentes razões eu gostaria de agradecer algumas pessoas que foram fundamentais para a conclusão deste trabalho:

ao professor da Universidade do Porto, Paulo Portugal, pelo exercício exemplar da sua função de orientador, pela disponibilidade em ouvir todas as questões e dúvidas que surgiram no decorrer deste trabalho e também pela confiança em mim depositada ao longo destes anos. Muito mais que a orientação de um trabalho de doutoramento tu me deste valiosos conselhos para a vida, não os esquecerei. Tenho a certeza que a realização deste projeto não teria sido possível sem o seu auxílio.

ao professor na Universidade Federal de Santa Catarina, Ricardo Moraes, por todo o estímulo que recebi, pelas inúmeras ideias proporcionadas, paciência em discuti-las, questionamentos e principalmente pela sua amizade e confiança em mim depositada.

ao professor da Universidade do Porto, Francisco Vasques, pela confiança em mim depositada, pelas discussões científicas que tivemos a oportunidade de realizar e também por todo o auxílio que me deste desde que cheguei ao Porto.

à minha esposa Ana Paula, por toda a compreensão e apoio. Mesmo nos momentos mais difíceis tu sempre esteve ao meu lado dando-me forças e ânimo para continuar. Tenha a certeza de que este trabalho não teria sido finalizado sem a sua ajuda.

à minha filha Yasmin, por compreender tantas horas de ausência do pai. Pelas vezes que cheguei desmotivado em casa e que o teu abraço fez meu mundo mudar instantaneamente. Sem ti eu não teria chego até o fim.

ao meus pais (Gentil e Iara) e irmãos (Átila e Ádson) que sempre estiveram ao meu lado em todas as minhas caminhadas; e nesta, uma vez mais. Foi muito difícil ficar este período de tempo fisicamente longe de vocês. Agradeço por tudo o que representam na minha vida.

à Tata e Aldori, pelo apoio e confiança depositada em mim. Cuidarei muito bem da neta do senhor Aristides e da dona Edith.

ao Silvio Sampaio, Carlos Viegas, Marcos Madruga, Luís Oliveira e Zahid Iqbal, pela partilha de diferentes momentos e pela oportunidade de convívio e aprendizado com cada um de vocês.

ao Eng. Vasques de Carvalho, pela amizade, atenção e disponibilidade. O senhor me deu a oportunidade de vivenciar com tranquilidade a minha estadia e de minha família no Porto.

aos portugueses, que acolheram à mim e a minha família. Um povo forte e batalhador, o qual passei a respeitar e admirar ainda mais.

Por fim, agradeço à Fundação para a Ciência e a Tecnologia (FCT), ao Departamento de Engenharia Mecânica e Gestão Industrial (DEMEGI) e à Unidade de Integração de Sistemas e Processos Automatizados (UISPA) do Instituto de Engenharia Mecânica (IDMEC – pólo FEUP) pelo apoio financeiro e logístico neste estudo.

Resumo

Com os recentes avanços no desenvolvimento de tecnologias de comunicação sem fios, estas tornaram-se uma alternativa credível para o suporte de aplicações distribuídas em ambientes industriais. O aumento da mobilidade e a redução dos custos de instalação e de manutenção são fatores preponderantes que tornam atrativa a sua adoção. Além disso, a possibilidade de construir sistemas de produção de rápida e fácil reconfiguração (*on-the-fly*) torna a sua utilização ainda mais interessante no domínio industrial. Neste contexto, e com base na crescente disponibilidade de soluções de redes sem fios para ambientes industriais, é provável que num futuro próximo uma parte importante destas tecnologias seja baseada na família de protocolos IEEE 802.11.

Os requisitos de comunicação em ambientes industriais são muito específicos. O tráfego transmitido neste tipo de ambiente está tipicamente associado a aplicações de controlo, para as quais os dados de controlo devem ser periodicamente transferidos entre sensores, controladores e atuadores de acordo com metas temporais bem definidas.

Um dos grandes desafios encontrados para suportar este tipo de tráfego em redes sem fios está associado ao meio físico utilizado para a transmissão das mensagens. Este meio é considerado essencialmente um ambiente de comunicação aberto, uma vez que a qualquer momento outras estações que estejam fora da esfera de controlo da arquitetura de tempo-real podem tentar acedê-lo para realizar as suas transmissões. Caso estas estações estejam a operar na mesma área de cobertura geográfica e no mesmo canal de comunicação, a transmissão destas mensagens pode resultar em atrasos não controláveis na transmissão do tráfego de tempo-real.

Logo, uma das questões fundamentais que devem ser abordadas é: "*Como garantir que os requisitos temporais do tráfego de tempo-real sejam respeitados, quando este é transmitido num ambiente de comunicação aberto?*". O principal objetivo desta tese é responder a esta questão, propondo novas soluções para a transmissão de tráfego de tempo-real em redes IEEE 802.11 a operar em ambientes de comunicação abertos.

Nesta tese é proposta uma nova arquitetura de comunicação de tempo-real denominada RT-WiFi, baseada na operação conjunta de novos mecanismos de controlo de acesso ao meio e controlo de admissão. Estes mecanismos são capazes de priorizar o tráfego de tempo-real sobre os restantes tipos de tráfego, sem ter a necessidade de controlar estes últimos diretamente. A avaliação do desempenho do RT-WiFi em cenários industriais mostrou que este é capaz de suportar a transmissão de tráfego *soft real-time*, mesmo ao operar em cenários muito exigentes. Além disso, a arquitetura proposta oferece também um atraso médio quase constante, característica esta considerada importante na transmissão de tráfego de tempo-real. Quando os seus resultados foram comparados aos obtidos por soluções tradicionais como o HCCA, o RT-WiFi mostrou que o seu comportamento é significativamente melhor no que diz respeito ao número máximo de fluxos de dados admitidos.

Abstract

The recent advances in the development of wireless communication technologies, make it a credible alternative to the support of distributed applications in industrial environments. Increased mobility and the reduction of installation and maintenance costs are factors that make attractive the use of this technology. Moreover, the possibility of building production systems with a quick and easy reconfiguration process (*on-the-fly*) makes its use even more interesting in the industrial domain. In this context, and based on the increasing availability of wireless solutions for industrial environments, it is likely that in the near future an important part of this technology will be based on the IEEE 802.11 family of protocols.

The communication requirements in industrial environments are very specific. Transmitted traffic in this environments is typically associated with control applications, for which the control data should be periodically transferred between sensors, actuators and controllers according to well-defined deadlines.

One of the major challenges encountered to support this kind of traffic in wireless networks is related with the physical medium used for the transmission. This medium is essentially considered an open communication environment, since at any time other stations that are outside of the sphere-of-control of real-time architecture may try to access it to perform their transmissions. If these stations are operating in the same geographical coverage area and in the same communication channel, the transmission of this traffic can result in uncontrollable delays for real-time traffic.

Thus, the fundamental issue that must be addressed is: "*How to ensure that the deadlines of real-time traffic are respected when it is transmitted in an open communication environment?*". The main objective of this thesis is to answer this question, proposing new solutions to the transmission of real-time traffic in IEEE 802.11 networks operating in open communication environments.

In this thesis is proposed a new architecture for real-time communication called RT-WiFi, based on the joint operation of new mechanisms for medium access and admission control. These mechanisms are able to prioritize real-time traffic over others types of traffic, without the need to control the latter directly. The evaluation of the RT-WiFi in industrial scenarios showed that it is able to fulfill all the requirements necessary to transmit real-time traffic, even when it operates in very demanding scenarios. Moreover, the proposed architecture also provides a nearly constant transmission delay, which is considered important in the transmission of real-time traffic. When comparing their results with traditional solutions as the HCCA, the RT-WiFi showed significantly better behavior regarding to the maximum number of admitted data flows.

Résumé

Suite aux récents progrès et développements des technologies de communication sans fil, celles-ci sont devenues une alternative crédible à l'appui des applications distribuées dans des environnements industriels. La mobilité accrue et les coûts réduits d'installation et d'entretien sont des facteurs prépondérants pour rendre leur adoption plus attractive. En outre, la possibilité de construire des systèmes pour la production à reconfiguration rapide et facile rend son utilisation encore plus intéressante dans le domaine industriel. Dans ce contexte, et appuyé sur la disponibilité croissante de solutions sans fil pour les environnements industriels, il est probable que dans un futur proche, une importante partie de cette technologie sera basée sur la norme 802.11 IEEE famille de protocoles.

Les exigences de performance dans les environnements industriels sont très spécifiques. Les informations transmises dans de tels environnements sont généralement associées à des applications de contrôle pour lesquels, les données de contrôle doivent être transférées périodiquement entre des capteurs, des contrôleurs et des actionneurs en fonction des objectifs temporels bien définis.

L'une des principales difficultés rencontrées pour soutenir ce genre de trafic dans les réseaux sans fil est associée au support physique utilisé pour la transmission de messages. Ce milieu est considéré comme étant essentiellement un environnement de communication ouverte. En effet, à tout moment, d'autres stations qui se trouvent en dehors du contrôle de l'architecture en temps réel, peuvent tenter d'accéder à leurs émissions. Si ces stations opèrent dans la même zone de couverture géographique et dans le même canal de communication, la transmission de ces messages peut entraîner des retards inévitables dans la transmission du trafic en temps réel.

Ainsi, l'une des questions fondamentales qui doivent être abordées est: "Comment garantir que les exigences temporelles de trafic en temps réel sont respectées quand il est transmis dans un environnement de communication ouverte?". Les objectifs principaux de cette thèse sont de répondre à cette problématique, ainsi que proposer de nouvelles solutions pour la transmission du trafic en temps réel dans les réseaux IEEE 802.11 opérant dans des environnements de communications ouvertes.

Cette thèse propose une nouvelle architecture de communication pour le temps réel dit RT-WiFi, basée sur l'exploitation commune de nouveaux mécanismes de contrôle d'accès au support et de contrôle d'admission. Ces mécanismes sont en mesure de donner la priorité au trafic en temps réel sur les autres types de trafic, sans la nécessité de contrôler directement ce dernier. L'évaluation des performances de la RT-WiFi dans les scénaris industriels a montré qu'il est capable de supporter la transmission du trafic "soft real-time", même lors de l'utilisation dans des scénaris très exigeants. En outre, l'architecture proposée prévoit également une caractéristique de retard de moyenne, à peu près constant, qui est considéré comme important dans la transmission du trafic en temps réel. Lorsque les résultats ont été comparés à ceux obtenus par les solutions traditionnelles comme HCCA, le RT-WiFi a montré que leur comportement est nettement meilleur en ce qui concerne le nombre maximum de flux de données.

Índice

Lista de Figuras	xiv
Lista de Tabelas	xv
Lista de Algoritmos	xvii
Lista de Acrónimos	xxiii
1 Introdução	1
1.1 Contexto	2
1.2 Objetivos	3
1.3 Contribuições da Tese	3
1.4 Estrutura da Tese	4
2 IEEE 802.11	5
2.1 Introdução	5
2.2 Descrição geral da arquitetura	6
2.3 Mecanismo de Controlo de Acesso ao Meio	7
2.3.1 DCF - <i>Distributed Coordination Function</i>	8
2.3.2 PCF - <i>Point Coordination Function</i>	11
2.3.3 EDCA - <i>Enhanced Distributed Channel Access</i>	13
2.3.4 HCCA - <i>HCF Controlled Channel Access</i>	15
2.4 Mecanismo de Controlo de Admissão	16
2.4.1 TSPEC - <i>Traffic Specification</i>	17
2.4.2 Mecanismo de Controlo de Admissão do EDCA	19
2.4.3 Mecanismo de Controlo de Admissão do HCCA	21
2.5 Conclusões	23
3 Trabalhos Relacionados	25
3.1 Mecanismos de Controlo de Acesso ao Meio	25
3.1.1 Abordagens Centralizadas	26
3.1.2 Abordagens Distribuídas	30
3.1.3 Síntese dos Mecanismos de Controlo de Acesso ao Meio	34
3.2 Mecanismos de Controlo de Admissão	37
3.2.1 Abordagens Baseadas em Modelos	38
3.2.2 Abordagens Baseadas em Medidas	40
3.2.3 Abordagens Híbridas	48
3.2.4 Síntese dos Mecanismos de Controlo de Admissão	58
3.3 Conclusões	61

4	A Arquitetura RT-WiFi	63
4.1	Introdução	63
4.2	Mecanismo de Controlo de Acesso ao Meio	66
4.3	Mecanismo de Controlo de Admissão	73
4.3.1	Teste de Escalonabilidade	77
4.3.2	Gestão do Tamanho dos <i>Slots</i>	83
4.3.3	Verificação de Falhas das TS	87
4.4	Suporte de Tráfego <i>Broadcast</i>	88
4.5	Conclusões	90
5	Resultados	93
5.1	Descrição dos cenários	93
5.1.1	Tráfego de Tempo-Real	96
5.1.2	Tráfego Não Tempo-Real	98
5.2	Resultados	101
5.2.1	Atraso Médio	102
5.2.2	Percentagem Média de Deadlines Perdidas	105
5.2.3	Fairness	108
5.2.4	Tamanho Médio do <i>Slot</i>	110
5.2.5	Percentagem Média de <i>Deadlines</i> Perdidas em função da Carga da Rede	112
5.3	Conclusões	114
6	Conclusões e Trabalhos Futuros	117
6.1	Conclusões	117
6.2	Trabalhos Futuros	120
	Referências	123
A	Lista de Publicações	133
B	Modelo HCCA implementado no simulador OPNET	135
B.1	Introdução	135
B.2	Implementação do HCCA	138
B.2.1	Limitações do modelo implementado	138
B.2.2	O arquivo <code>wlan_dispatch.pr.m</code>	139
B.2.3	O arquivo <code>wlan_mac_hcf.pr.m</code>	140

Lista de Figuras

2.1	Arquiteturas da rede IEEE 802.11.	6
2.2	Arquitetura MAC da norma IEEE 802.11 original (adaptada de [1]).	7
2.3	Arquitetura MAC da emenda IEEE 802.11e (adaptada de [2]).	8
2.4	Diferentes IFS das funções DCF e PCF (adaptada de [1]).	8
2.5	Relações entre os IFS e o <i>aSlotTime</i> (adaptada de [2]).	9
2.6	Transmissão DCF (adaptada de [2]).	10
2.7	Transmissão PCF (adaptada de [2]).	12
2.8	CFP encurtado (adaptada de [2]).	13
2.9	Relação entre IFS e AIFS no mecanismo EDCA (adaptada de [2]).	14
2.10	Transmissão HCCA (adaptada de [2]).	16
2.11	Fluxo das mensagens TSPEC (adaptada de [3]).	17
2.12	Formato do TSPEC (adaptada de [4]).	18
2.13	Cálculo do <i>SI</i> no mecanismo HCCA (adaptada de [4]).	22
3.1	Comparação entre as propostas de controlo de acesso ao meio apresentadas.	36
3.2	Comparação entre as propostas de controlo de admissão apresentadas.	60
4.1	Fluxo das mensagens na arquitetura RT-WiFi.	64
4.2	Arquitetura RT-WiFi.	65
4.3	Modelos de interação entre os dispositivos de tempo-real.	66
4.4	Ciclo TDMA na arquitetura RT-WiFi.	69
4.5	Sobreposição parcial dos <i>slots</i>	70
4.6	Exemplo de uma lista de escalonamento enviada na mensagem de <i>beacon</i>	71
4.7	Tempo de atraso <i>uplink</i> (b_i^{uplink}).	84
4.8	Tempo de atraso <i>downlink</i> ($b_i^{downlink}$).	84
4.9	Tempo de atraso <i>downlink</i> ($b_i^{downlink}$) para transmissão em <i>broadcast</i>	89
5.1	Ambiente de comunicação aberto.	94
5.2	Atraso médio para o cenário $P = 30ms$	103
5.3	Atraso médio para o cenário $P = 60ms$	103
5.4	Atraso médio para o cenário $P = 90ms$	104
5.5	Média de <i>deadlines</i> perdidas para o cenário $P = 30ms$	105
5.6	Média de <i>deadlines</i> perdidas para o cenário $P = 60ms$	106
5.7	Média de <i>deadlines</i> perdidas para o cenário $P = 90ms$	107
5.8	<i>Fairness</i> para o cenário $P = 30ms$	108
5.9	<i>Fairness</i> para o cenário $P = 60ms$	109
5.10	<i>Fairness</i> para o cenário $P = 90ms$	109
5.11	Tamanho médio do <i>slot</i> para o cenário $P = 30ms$	110
5.12	Tamanho médio do <i>slot</i> para o cenário $P = 60ms$	111

5.13	Tamanho médio do <i>slot</i> para o cenário $P = 90ms$	112
5.14	Percentagem média de <i>deadlines</i> perdidas para o cenário $P = 30ms$	113
5.15	Percentagem média de <i>deadlines</i> perdidas para o cenário $P = 60ms$	113
5.16	Percentagem média de <i>deadlines</i> perdidas para o cenário $P = 90ms$	114
B.1	Objetos disponíveis para modelagem de WLAN.	136
B.2	Parâmetros de comunicação sem fio configuráveis numa estação.	136
B.3	Pilha TCP/IP implementada no objeto <code>wlan_wkstn</code>	137
B.4	Processo <code>wlan_dispatch</code>	137
B.5	Processo <code>wlan_mac_hcf</code>	138

Lista de Tabelas

2.1	Valores de aCW_{min} e aCW_{max} para cada camada física.	11
2.2	Comparação entre IEEE 802.1D e IEEE 802.11e.	13
2.3	Parâmetros das Categorias de Acesso.	15
5.1	Ocupação do meio de transmissão (por mensagem).	96
5.2	Parâmetros do RT-WiFi.	97
5.3	Parâmetros do HCCA.	97
5.4	Parâmetros utilizados para o tráfego HTTP.	99
5.5	Parâmetros utilizados para o tráfego SMTP/POP3.	99
5.6	Parâmetros utilizados para o tráfego FTP.	100
5.7	Parâmetros utilizados para o tráfego de vídeo.	100
5.8	Parâmetros utilizados para o tráfego de voz.	101

Lista de Algoritmos

1	Envio da mensagem de <i>beacon</i> pelo AP_{TR}	72
2	Receção da mensagem de <i>beacon</i> pela estação de tempo-real	72
3	Função SEARCHTSID()	73
4	Interrupção SP_i	73
5	Envio da requisição ADDTS ao AP_{TR}	75
6	Processamento da requisição ADDTS pela ACU.	77
7	Função DELTS()	77
8	Função SchedTest()	79
9	Função COMPUTEVALUES()	81
10	Função COMPUTE COMPRESSIONFACTOR()	81
11	Função GETSCHEDLIST()	82
12	Interrupção de avaliação dos atrasos de <i>uplink/downlink</i>	85
13	Função ComputeDelay()	86
14	Função RedefineSlotLength()	87
15	Função CheckIdleFlows()	88

Lista de Acrónimos

AC	Access Category
ACK	Acknowledgement
ACM	Admission Control Mandatory
ACU	Admission Control Unit
ADDTS	Add Traffic Stream
AIFS	Arbitration Interframe Space
AIFSN	Arbitration Interframe Space Number
AP	Access Point
BB	Black-Burst
BE	Best-Effort
BER	Bit Error Ratio
BI	Beacon Interval
BK	Background
BP	Backoff Period
BSS	Basic Service Set
BSSID	Basic Service Set Identification
BTPS	Busy Tone Priority Scheduling
CAP	Controlled Access Phase
CBR	Constant Bit Rate
CBS	Constant Bandwidth Server
CCA	Clear Channel Assessment
CFP	Contention Free Period
CL	Controlled Load
COTS	Commercial Off-The-Shelf

CP	Contention Period
CS	Carrier Sense
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CT	Communication Type
CTS	Clear-to-Send
CW	Contention Window
DCF	Distributed Coordination Function
DELTS	Delete Traffic Stream
DFS	Distributed Fair Scheduling
DIFS	Distributed Interframe Space
EB	Energy-Burst
EDCA	Enhanced Distributed Channel Access
EDF	Earliest Deadline First
EE	Excellent Effort
EIFS	Extended Interframe Space
EMI	Electromagnetic Interference
EP	End Point
EPL	Ethernet Powerlink
FCR	Force Collision Resolution
FHCF	Fair HCF
FTDMA	Flexible TDMA
FTT	Flexible Time-Triggered
GHz	GigaHertz
HC	Hybrid Coordinator
HCCA	HCF Controlled Channel Access
HCF	Hybrid Coordination Function
HEMM	HCCA/EDCA Mixed Mode
IBSS	Independent BSS
IEEE	Institute of Electrical and Electronics Engineers
IFS	Interframe Space

MAC	Medium Access Control
Mbps	Mega bits por segundo
MHz	Mega Hertz
MMPDU	MAC Management Protocol Data Unit
MOS	Mean Opinion Score
MPDU	MAC Protocol Data Unit
MSDU	MAC Service Data Unit
NAV	Network Allocator Vector
NC	Network Control
NT	Number of Transmissions
NTR	Não Tempo-Real
OFDM	Orthogonal Frequency Division Multiplexing
PAAC	Priority Access-based Admission Control
PC	Point Coordinator
PCF	Point Coordination Function
PER	Packet Error Ratio
PHCCA	Priority HCCA
PHY	Physical
PIFS	PCF Interframe Space
PLCP	Physical Layer Convergence Procedure
PRCW	Physical Rate and Contention Window
PSQA	Pseudo-Subject Quality Assessment
QAP	QoS Access Point
QoS	Quality of Service / Qualidade de Serviço
QP-CAT	Queue size Prediction using Computation of Additional Transmission
QSTA	QoS Station
RIFS	Real-Time IFS
RM	Rate Monotonic
RTH	Real-Time HCCA
RTR	Ready-To-Receive

RTS	Request-to-Send
SCFQ	Self-Clocked Fair Queueing
SI	Service Interval
SIFS	Short Interframe Space
SNR	Signal-to-Noise Ratio
SoA	Start of Asynchronous
SoC	Start of Cycle
SP	Start Point
SRP	Stack Resource Policy
SRV	Servidor
TBTT	Target Beacon Transmission Time
TCLAS	Traffic Classification
TDMA	Time Division Multiple Access
TI	Transmission Interval
TID	Traffic Identifier
TM	Trigger Message
TR	Tempo-Real
TRT	Token Rotation Time
TS	Traffic Stream
TSID	Traffic Stream Identified
TSPEC	Traffic Specification
TTRT	Target Token Revolution Time
TXOP	Transmission Opportunity
UP	User Priority
VBR	Variable Bit Rate
VI	Video
VID	Virtual Identifier
VO	Voice
VPP	Virtual Polling Period
VPRP	Voice Packet Resolution Period

VTP-CSMA	Virtual Token Passing CSMA
WCET	Worst Cases Execution Time
WFTT	Wireless FTT
WiFi	WiFi Alliance
WLAN	Wireless Local Area Network
WRTMAC	Wireless Real-Time Medium Access Control
WTRP	Wireless Token Ring Protocol

Capítulo 1

Introdução

Nos últimos anos, as redes sem fios tornaram-se uma alternativa credível para o suporte de aplicações distribuídas em ambientes industriais. Os principais motivos para a utilização desta tecnologia estão relacionados com o aumento da mobilidade e com a redução do tempo de instalação e dos custos de manutenção. Além disso, a capacidade de construir sistemas de produção extremamente flexíveis (*on-the-fly* [5]), faz dos sistemas de comunicação sem fios um forte candidato para substituir os tradicionais sistemas de comunicação cablados. Estas características vêm de encontro às exigências atuais dos sistemas de produção, que necessitam de fáceis e rápidas reconfigurações [6].

Embora a utilização de sistemas de comunicação sem fios permita obter este conjunto de benefícios, acarretam também novos desafios. O principal desafio encontra-se no meio físico de comunicação utilizado, que é compartilhado com todos os dispositivos que estejam a operar na mesma área de cobertura geográfica e canal de comunicação. Isto faz com que o meio seja considerado um **ambiente de comunicação aberto**, em que qualquer estação pode tentar aceder a qualquer momento ao meio para enviar as respectivas mensagens. Esta característica impede que a carga total da rede possa ser prevista no momento do projeto, além de impossibilitar também que a mesma seja efetivamente controlada durante a operação da rede.

Um outro desafio imposto a estes sistemas está relacionado com os requisitos temporais exigidos pelas aplicações industriais. Estas, tipicamente exigem que as mensagens transmitidas sejam entregues respeitando um limite temporal pré-definido (*deadline*). As aplicações que possuem estes requisitos são denominadas **Aplicações de Tempo-Real** e podem ser classificadas em dois tipos¹: **i) hard real-time**, que requerem um tempo de resposta limitado e previsível, caso contrário podem ocorrer consequências severas para o funcionamento do sistema, como por exemplo, perdas económicas ou de vidas humanas; **ii) soft real-time**, que toleram perdas de algumas metas temporais, especialmente quando estas ocorrem de forma esporádica, sem ter como resultado

¹Para ambas as classificações serão utilizados termos na língua inglesa por não se encontrarem traduções compatíveis para a língua portuguesa.

consequências severas para o sistema [7].

Tradicionalmente, nas redes cabladas o tráfego de tempo-real é garantido através do controlo rígido de todos os dispositivos de comunicação. A coexistência na mesma rede de estações de tempo-real e de não tempo-real torna-se possível, por exemplo, através de mecanismos de suavização de tráfego aplicados às estações não tempo-real [8]. No entanto, esta e outras abordagens que consideram um meio de comunicação fechado não são adequadas para ambientes de comunicação sem fios, uma vez que, não é possível impor qualquer restrição ao tráfego gerado pelas estações que estão fora da esfera de controlo² da arquitetura de tempo-real³.

Neste contexto, a necessidade de sistemas de comunicação sem fios de alto desempenho e capazes de fornecer garantias temporais às aplicações industriais é um dos principais desafios de investigação neste domínio.

1.1 Contexto

Nesta tese, é proposta uma nova arquitetura para suportar a transmissão de tráfego de tempo-real em redes sem fios a operar em ambientes de comunicação abertos. Esta proposta assenta numa tecnologia de comunicação já existente e em concreto na norma IEEE 802.11⁴ [4]. A opção por esta tecnologia deveu-se ao facto desta ser considerada atualmente a norma *de facto* em conectividade para redes locais sem fio (WLAN – *Wireless Local Area Networks*). O seu baixo custo, elevado desempenho, facilidade de implementação e disponibilidade de uma grande gama de dispositivos no mercado foram os principais fatores que contribuíram para a sua ampla generalização.

Embora esta tecnologia esteja amplamente difundida, de acordo com Cena *et al.* [5], as soluções sem fios e os respectivos produtos disponíveis atualmente no mercado possuem sérias limitações no que diz respeito a transmissão de tráfego de tempo-real. Os principais motivos invocados são:

- **Ruído:** as comunicações de rádio são muito sensíveis ao ruído, podendo resultar em taxas de erro de transmissão elevadas. Neste contexto, o ruído é caracterizado principalmente por interferências eletromagnéticas (EMI – *Electromagnetic Interference*) e por problemas na propagação do sinal (por exemplo, *fading* e *multipath propagation*). Em ambientes industriais estes problemas tornam-se ainda mais sérios devido à quantidade de equipamentos elétricos geradores de EMI e à existência de estruturas metálicas no ambiente;
- **Sobreposição de redes:** uma vez que o meio de comunicação sem fios é considerado um *ambiente de comunicação aberto*, podem existir sobreposições com outras redes sem fios

²O conceito de “dentro/fora” da esfera de controlo foi definido por Kopetz [9]. Sempre que uma estação de tempo-real está dentro da esfera de controlo de um sistema, isto significa que este sistema tem autorização para mudar todos os parâmetros de comunicação desta estação. Quando a estação está fora da esfera de controlo, estes parâmetros podem ser observados mas não podem ser modificados.

³Uma arquitetura de tempo-real é composta somente por dispositivos de tempo-real.

⁴As redes IEEE 802.11 também são conhecidas como redes WiFi.

que estejam a operar na mesma área de cobertura geográfica e no mesmo canal de comunicação, independentemente da tecnologia utilizada (por exemplo, IEEE 802.11, Bluetooth e ZigBee). Além disso, estas redes estão fora da esfera de controlo da arquitetura de tempo-real, o que pode resultar em atrasos não controláveis ou erros nas transmissões, dando origem a um comportamento não determinístico por parte do sistema de comunicação de tempo-real, característica esta considerada indesejada;

- **Mecanismo de acesso ao meio:** a norma IEEE 802.11 utiliza um mecanismo de controlo de colisões (*backoff*) implementado sob a forma de um algoritmo probabilístico. Isto resulta numa imprevisibilidade acerca dos atrasos gerados para a resolução das colisões.

Como resultado destas características, foi demonstrado em trabalhos anteriores [10, 11, 12, 13] que os dispositivos que utilizam o IEEE 802.11 não garantem, à partida, um comportamento compatível com os requisitos temporais exigidos pelas aplicações de tempo-real, tornando-se para isso, necessário propor novas soluções para este domínio.

1.2 Objetivos

O principal objetivo desta tese é investigar e propor novas soluções para suportar a transmissão de tráfego do tipo *soft real-time*, em redes IEEE 802.11, a operar em ambientes de comunicação abertos. Esta tese foca-se na avaliação das características do IEEE 802.11 para a transmissão de tráfego de tempo-real, da qual, resulta a proposta para uma nova arquitetura de comunicação baseada na operação conjunta de mecanismos de controlo de acesso ao meio e controlo de admissão.

Desta forma, o principal pressuposto desta tese é que a utilização conjunta destes mecanismos, através do aumentando da prioridade do tráfego de tempo-real face os demais tipos de tráfego existentes, e à gestão dinâmica dos recursos alocados às estações de tempo-real, é adequada para suportar a transmissão de tráfego de tempo-real em ambientes de comunicação abertos. Isto é, em ambientes de comunicação onde a carga da rede não pode ser prevista em tempo de projeto e nem efetivamente controlada durante a operação do sistema.

1.3 Contribuições da Tese

As principais contribuições desta tese são:

1. A especificação da arquitetura RT-WiFi, que permite a transmissão de tráfego do tipo *soft real-time* em redes IEEE 802.11 a operar em ambientes de comunicação abertos;
2. A especificação de um mecanismo de controlo de acesso ao meio, baseado num esquema TDMA (*Time Division Multiple Access*) flexível. Este esquema permite a modificação dinâmica tanto da ordem quanto do tamanho dos *slots* alocados a cada estação de tempo-real;

3. A especificação de um mecanismo de controlo de admissão híbrido, capaz de tomar decisões acerca da admissão (ou não) de novos fluxos de dados com base num conjunto de informações fornecidas tanto pelas estações de tempo-real, quanto por medições realizadas no meio de comunicação. Adicionalmente, este mecanismo suporta a implementação de diversos tipos de algoritmos de escalonamento de tempo-real e também é responsável pelo redimensionamento do tamanho dos *slots* alocados às estações de tempo-real;
4. A implementação de um modelo de simulação da função HCCA (*HCF Controlled Channel Access*), desenvolvido com a ferramenta de simulação OPNET *Wireless Modeler*. Esta contribuição torna-se igualmente importante uma vez que não existia um modelo implementado para a ferramenta de simulação utilizada.

Os artigos publicados no desenvolvimento desta tese, até o momento da finalização deste documento, são listados no Anexo A.

1.4 Estrutura da Tese

A tese está organizada em 6 capítulos, dos quais este é o primeiro. No **capítulo 2** é apresentada uma descrição resumida das principais características da norma IEEE 802.11, com o objetivo de compreender as suas principais funcionalidades e limitações. O **capítulo 3** apresenta a revisão do estado-da-arte, que está subdividido entre propostas focadas no mecanismo de controlo de acesso ao meio e propostas focadas no mecanismo de controlo de admissão. No **capítulo 4**, é apresentada e discutida a arquitetura RT-WiFi. A avaliação do desempenho desta arquitetura, obtida através de simulação, é apresentada no **capítulo 5**. Por fim, no **capítulo 6**, são apresentadas as conclusões finais e as perspectivas de trabalhos futuros.

Capítulo 2

IEEE 802.11

Este capítulo tem como objetivo apresentar uma breve introdução à norma IEEE 802.11, nomeadamente aos mecanismos de Controlo de Acesso ao Meio e de Controlo de Admissão, que têm um papel fundamental num sistema de comunicação de tempo-real.

2.1 Introdução

Em 1990, o IEEE (*Institute of Electrical and Electronics Engineers*) estabeleceu um comité para definir uma norma para as redes locais sem fios. A primeira versão desta norma foi aprovada em 1997 sob o nome de IEEE 802.11 [1] e suportava taxas de transmissão de 1 e 2 Mbps.

Em 1999, foram aprovadas as normas IEEE 802.11b [14] e IEEE 802.11a [15], que operam nas gamas de frequência de 2.4 e 5 GHz e alcançam taxas de transmissão de 11 e 54 Mbps, respectivamente. Na prática, os dispositivos que implementavam a norma IEEE 802.11b, acabaram por conquistar uma importante quota do mercado devido ao seu baixo custo e por terem sido disponibilizados para venda antes dos dispositivos que implementam a norma IEEE 802.11a. Em 2003, o IEEE aprovou a norma 802.11g [16] que opera na gama de frequência de 2.4 GHz e consegue atingir taxas de transmissão até 54 Mbps.

Para que os dispositivos IEEE 802.11 pudessem suportar diferentes níveis de qualidade de serviço (QoS – *Quality-of-Service*), foi aprovado em 2005 a emenda IEEE 802.11e [17], a qual inclui, além de outras inovações, o suporte para a transmissão de voz e de vídeo.

Mais recentemente, em 2009, foi aprovada a norma IEEE 802.11n [18], a qual tem por objetivo fornecer maiores taxas de transmissão. Os dispositivos que implementam esta norma podem operar em ambas as gamas de frequência (2.4 GHz e 5 GHz) e são capazes de obterem taxas de transmissão até 600 Mbps.

Em 2012 foi publicada uma versão revista da norma IEEE 802.11 [4], a qual incorporou as várias emendas aprovadas ao longo dos últimos anos, bem como algumas correções às mesmas.

2.2 Descrição geral da arquitetura

A arquitetura básica de uma rede IEEE 802.11 é denominada BSS (*Basic Service Set*), que caracteriza-se por um grupo de estações que pretendem comunicar entre si. Estas comunicações ocorrem dentro de uma área comum, denominada área de serviço básico (*Basic Service Area*), definida pelas características do meio físico de comunicação. Uma estação que esteja nesta área pode comunicar com qualquer outra estação membro da mesma BSS. As BSS são divididas em dois diferentes tipos: **i)** independentes; **ii)** infraestruturadas.

Nas BSS independentes (IBSS – *Independent BSS*), as estações comunicam diretamente entre si, desde que estejam na área de cobertura uma da outra (Figura 2.1(a)). Tipicamente, as IBSS são compostas por um pequeno número de estações configuradas para um objetivo específico e por um curto período de tempo. Um exemplo é a criação de uma rede IBSS para fornecer conectividade numa reunião, permitindo aos participantes a partilha de dados. Quando a reunião termina, a IBSS é dissolvida. Devido à sua curta duração, tamanho reduzido e propósito específico, as IBSS são também conhecidas como redes *Ad Hoc*.

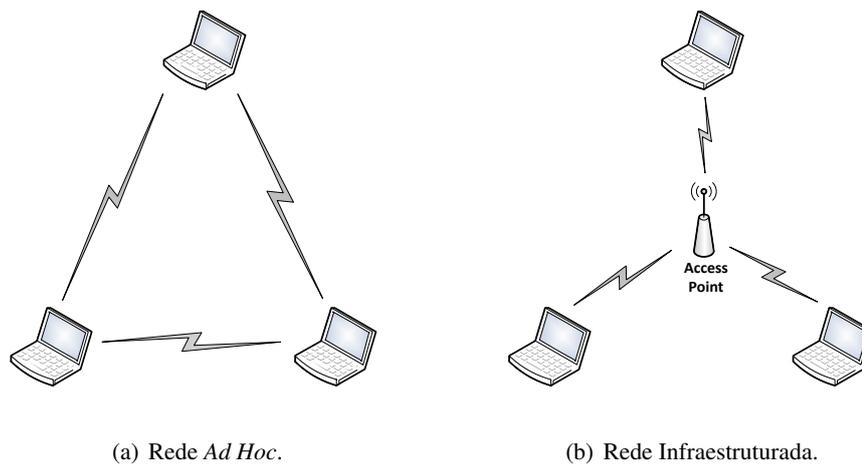


Figura 2.1: Arquiteturas da rede IEEE 802.11.

As BSS infraestruturadas (*Infrastructure BSS*), distinguem-se das IBSS pelo uso de um AP (*Access Point*) para interligar as estações (Figura 2.1(b)). Isto faz com que as estações necessitem estar dentro da área de cobertura do AP, não sendo imposta nenhuma restrição na distância existente entre cada estação. Desta forma, se uma estação pretende comunicar com outra, a comunicação será efetuada em dois passos: **a)** primeiro, a estação origem envia as suas mensagens para o AP (fluxo de dados denominado *uplink*); **b)** segundo, o AP reencaminha estas mensagens para as estações destino (fluxo de dados denominado *downlink*).

Numa rede infraestruturada, as estações devem inicialmente associar-se ao AP para obterem acesso aos serviços de rede. A associação é um processo pela qual a estação entra na rede IEEE 802.11, sendo logicamente equivalente a ligação de um cabo numa rede Ethernet. Quem inicia o processo de associação é a estação, tendo o AP a função de autorizar, ou não, o acesso. Esta decisão baseia-se no conteúdo do pedido de associação.

2.3 Mecanismo de Controlo de Acesso ao Meio

O IEEE 802.11 utiliza o protocolo CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) como base para o mecanismo de controlo de acesso ao meio. Neste caso, é utilizado um esquema de avaliação do meio para definir se o mesmo se encontra *livre* ou *ocupado*. Neste contexto, quando uma estação deseja transmitir uma mensagem, irá inicialmente “escutar” o meio por um período de tempo pré-determinado, a fim de verificar se outras estações estão a transmitir no mesmo canal de comunicação. Se o resultado desta avaliação definir que o canal está *livre*, a estação irá atrasar a transmissão, por um período de tempo aleatório, denominado de *backoff*. Se, após este tempo, o meio permanecer livre, a estação inicia imediatamente a sua transmissão. Caso contrário, a estação irá atrasar a sua transmissão, executando novamente o procedimento de *backoff*. Isto permite reduzir a probabilidade de colisões com outras estações que estejam a executar procedimentos semelhantes. Além disso, um outro *backoff*, denominado de *pos-backoff*, também é executado ao final de cada transmissão.

A subcamada MAC (*Medium Access Control*) propõe duas funções de coordenação diferentes: uma obrigatória, denominada DCF (*Distributed Coordination Function*) e outra opcional, denominada PCF (*Point Coordination Function*). A função DCF fornece um serviço de acesso ao meio com contenção, ou seja, todas as estações competem entre si pelo acesso ao meio. Já a função PCF fornece um serviço de acesso ao meio livre de contenção, ou seja, durante um período de tempo uma estação pode aceder ao meio sem competir com outras estações. Este acesso é efetuado de forma ordenada e apenas com a autorização de um controlador. A Figura 2.2 apresenta a arquitetura MAC da norma IEEE 802.11 original com as funções DCF e PCF.

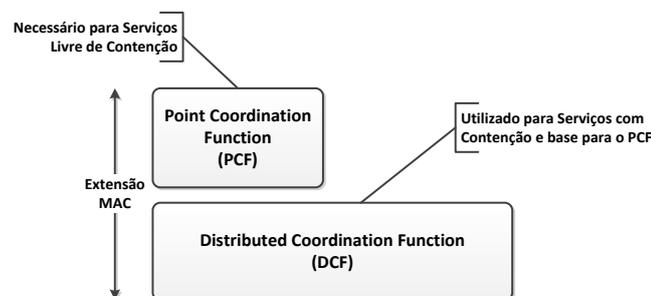


Figura 2.2: Arquitetura MAC da norma IEEE 802.11 original (adaptada de [1]).

Com o objetivo de suportar QoS em redes IEEE 802.11, foi publicado em 2005 a emenda IEEE 802.11e, que incorpora uma função de coordenação adicional denominada HCF (*Hybrid Coordination Function*), (Figura 2.3).

A função HCF fornece um serviço de acesso ao meio que aloca diferentes oportunidades de transmissão (TXOP – *Transmission Opportunities*) para cada estação. Uma TXOP é definida por um instante de tempo inicial e por uma duração máxima (ou seja, um intervalo de tempo durante o qual a estação é capaz de manter o controlo do acesso ao meio), de forma a possibilitar que

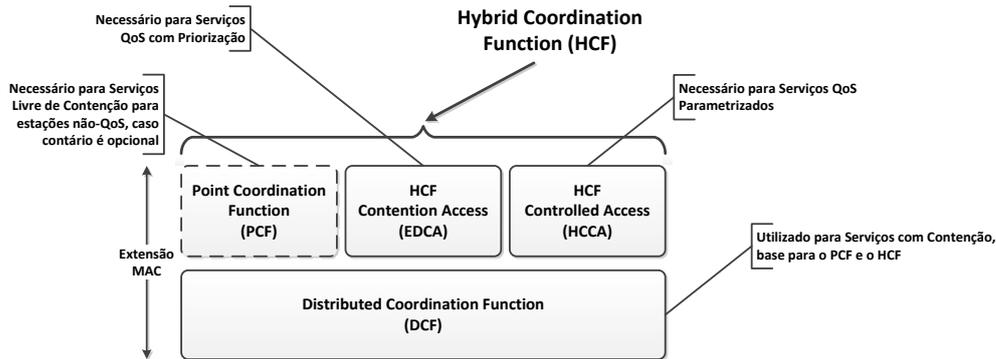


Figura 2.3: Arquitetura MAC da emenda IEEE 802.11e (adaptada de [2]).

múltiplas mensagens sejam transmitidas sem a interferência de outras estações da rede. As TXOP podem ser alocadas através de um dos dois mecanismos especificados pela função HCF: o EDCA (*Enhanced Distributed Channel Access*) e o HCCA (*HCF Controlled Channel Access*).

2.3.1 DCF - *Distributed Coordination Function*

A função DCF é definida pela norma IEEE 802.11 como a função básica de acesso ao meio. Nesta, as estações executam o procedimento de *backoff* antes de iniciarem as suas transmissões¹. A função DCF utiliza diferentes intervalos de tempo entre mensagens consecutivas, que são denominados IFS (*Interframe Spaces*) (Figura 2.4).

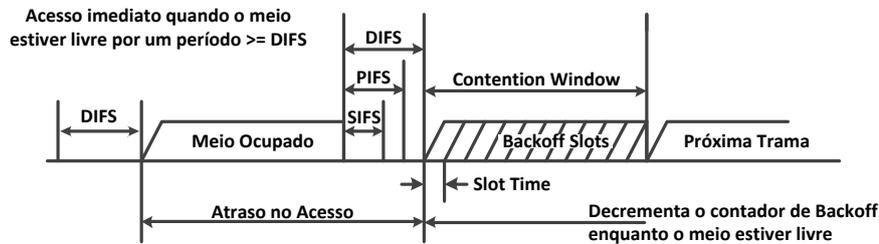


Figura 2.4: Diferentes IFS das funções DCF e PCF (adaptada de [1]).

Os diferentes valores de IFS resultam em diferentes prioridades no acesso ao meio para diversos tipos de mensagens. O SIFS (*Short Interframe Space*), o menor IFS, é utilizado para receber mensagens de confirmação (ACK – *Acknowledgment*) e na transmissão de múltiplos fragmentos de uma mensagem. O PIFS (*PCF Interframe Space*) é utilizado em operações sob a função PCF. O DIFS (*DCF Interframe Space*) é utilizado por estações que utilizam a função DCF na transmissão de mensagens de dados e de gestão. O EIFS (*Extended Interframe Space*) é utilizado em condições onde ocorram erros de comunicação.

¹É definido pela norma IEEE 802.11 um mecanismo de proteção adicional, RTS/CTS, para resolver o problema dos terminais ocultos e para gerir de forma adequada a transmissão de mensagens longas [1].

O valor do SIFS (doravante denominado $aSIFSTime$) é definido pela camada física (PHY – *Physical Layer*) e pode variar das camadas físicas utilizadas pelos dispositivos (802.11a, 802.11b, 802.11g ou 802.11n). Além disso, o seu valor é também utilizado como base para calcular os restantes IFS. O valor de $aSIFSTime$ é dado pela seguinte equação:

$$aSIFSTime = aRxRFDelay + aRxPLCPDelay + aMACProcessingDelay + aRxTxTurnaroundTime \quad (2.1)$$

onde $aRxRFDelay$ é o atraso na antena de receção, $aRxPLCPDelay$ é o atraso da camada PHY para transmitir a mensagem para a subcamada MAC, $aMACProcessingDelay$ é o atraso de processamento da subcamada MAC e $aRxTxTurnaroundTime$ é o tempo necessário para a mudança do fluxo de processamento da antena de receção para a antena de transmissão.

Outro parâmetro definido pela camada física é o $aSlotTime$. Além de ser utilizado para a definição dos intervalos de *backoff*², este parâmetro também serve como base para a definição dos diferentes IFS (com exceção do $aSIFSTime$). Seu valor é dado pela seguinte equação:

$$aSlotTime = aCCATime + aRxTxTurnaroundTime + aAirPropagationTime + aMACProcessingDelay \quad (2.2)$$

onde $aCCATime$ é o tempo necessário para se detetar (utilizando o mecanismo CCA – *Clear Channel Assessment*) que o meio de comunicação está livre, $aRxTxTurnaroundTime$ é o tempo necessário para se mudar o fluxo de processamento da antena de receção para a antena de transmissão, $aAirPropagationTime$ é o atraso de propagação no ar, e $aMACProcessingDelay$ é o atraso de processamento da subcamada MAC.

A Figura 2.5 apresenta a relação entre o $aSlotTime$ e os diferentes IFS.

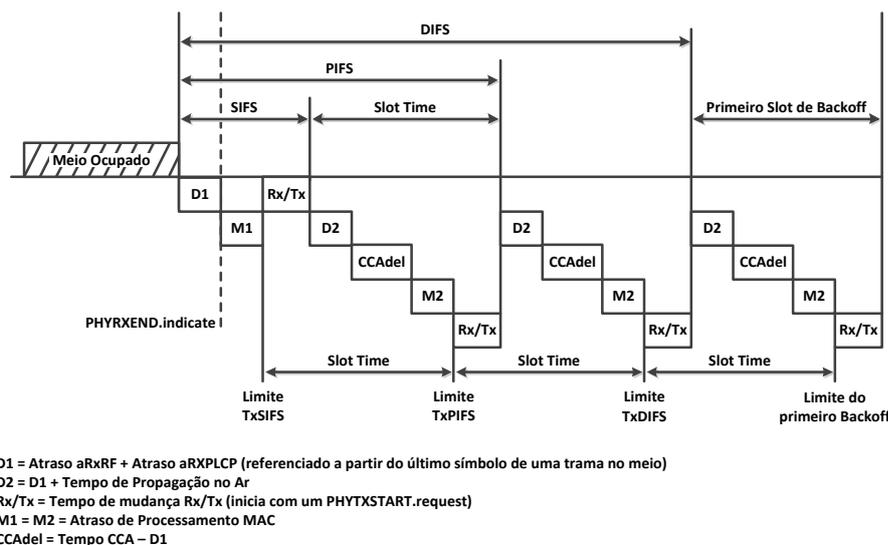


Figura 2.5: Relações entre os IFS e o $aSlotTime$ (adaptada de [2]).

²Os valores definidos pelos intervalos de *backoff* são múltiplos de $aSlotTime$.

Após descrever o $aSIFSTime$ e o $aSlotTime$, é possível apresentar os valores de DIFS, PIFS e EIFS, os quais são doravante denominados $aPIFSTime$, $aDIFSTime$ e $aEIFSTime$, respectivamente, sendo definidos pelas seguintes equações:

$$aPIFSTime = aSIFSTime + aSlotTime \quad (2.3)$$

$$aDIFSTime = aSIFSTime + (2 \times aSlotTime) \quad (2.4)$$

$$aEIFSTime = C_{ACK} + aSIFSTime + aDIFSTime \quad (2.5)$$

onde C_{ACK} é o tempo necessário para a transmissão de uma mensagem de confirmação (ACK) utilizando a taxa de transmissão base definida pela camada física.

Quando uma estação, que opera sob a função DCF, pretende realizar uma transmissão, necessita previamente de "escutar" o meio utilizando para isso o mecanismo CS (*Carrier Sense*), implementado pela primitiva CCA. Caso o meio esteja livre³ por um período de tempo maior ou igual à $aDIFSTime$, a estação pode iniciar a sua transmissão imediatamente. Caso contrário, a estação aguarda até que o meio fique livre por um período de tempo igual $aDIFSTime$ e seguidamente atribui a um contador de *backoff* um valor definido por um inteiro (múltiplo de $aSlotTime$) que toma valores no intervalo $[0, CW]$, onde CW (*Contention Window*), é inicialmente definido como aCW_{min} ⁴. Conforme o meio mantém-se livre, este contador é decrementado. Quanto o seu valor atingir zero, então a estação inicia a sua transmissão (Figura ??).

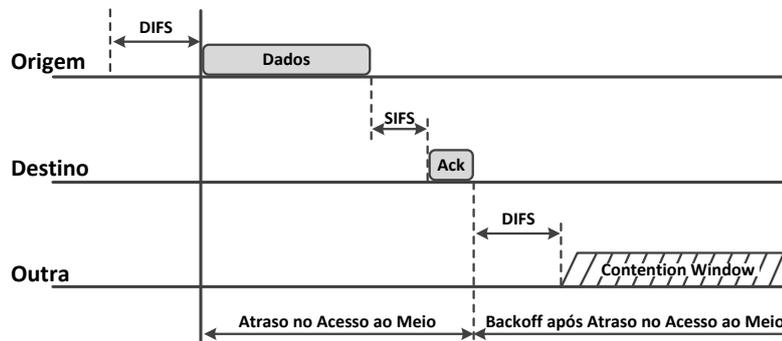


Figura 2.6: Transmissão DCF (adaptada de [2]).

Caso o meio torne-se ocupado antes do contador atingir zero, então a sua contagem é suspensa, sendo retomada apenas quando o meio torne-se livre por um período de tempo igual à $aDIFSTime$. Se após uma transmissão, a mensagem de confirmação não for recebida, então a estação irá incrementar o seu número de retransmissões efetuadas e verificar se este valor atingiu o valor máximo definido pela norma IEEE 802.11. Se o número de retransmissões for inferior ao valor máximo permitido, então é obtido um novo valor aleatório de *backoff* para uma nova tentativa de transmissão. Neste caso, o valor do CW é incrementado por $(oldCW \times 2 + 1)$, com um limite superior dado por aCW_{max} ⁵. Caso o número máximo de retransmissões tenha sido atingido, então a estação cancela a transmissão da mensagem.

³Tempo contado a partir da receção da última mensagem pela camada física.

⁴O valor do aCW_{min} é definido pela camada física.

⁵O valor de aCW_{max} é definido pela camada física.

A Tabela 2.1 apresenta os diferentes valores de aCW_{min} e aCW_{max} para cada camada física. No caso do IEEE 802.11g, o valor de aCW_{min} será 15 caso todas as estações da BSS estejam a operar em 5 GHz, caso contrário será 31.

Tabela 2.1: Valores de aCW_{min} e aCW_{max} para cada camada física.

CW	IEEE 802.11a	IEEE 802.11b	IEEE 802.11g	IEEE 802.11n
aCW_{min}	15	31	15/31	15
aCW_{max}	1023	1023	1023	1023

2.3.2 PCF - Point Coordination Function

A função PCF foi proposta na versão original da norma como uma função opcional capaz de fornecer um serviço de acesso ao meio livre de contenção. O PCF implementa um esquema de *polling* centralizado para suportar a transmissão síncrona de mensagens, onde o PC (*Point Coordinator*) opera como um coordenador central, definindo as regras de *polling*. Este coordenador é utilizado para assegurar o acesso ao meio livre de contenção através da sua restrição. Esta restrição faz com que as estações associadas ao PC possam transmitir somente após receberem uma autorização. O coordenador central encontra-se geralmente instalado no AP, o que limita a função PCF a redes infraestruturadas.

O serviço livre de contenção fornecido pela função PCF é utilizado apenas durante uma parte do tempo. Assim, quando esta função é utilizada, o tempo é dividido entre o período livre de contenção (CFP – *Contention-Free Period*) e o período com contenção (CP – *Contention Period*) os quais são alternados entre si. O acesso ao meio no CFP é controlado pela função PCF, enquanto que no CP é controlado pela função DCF.

O início do CFP é definido através do envio de uma mensagem de *beacon* pelo PC. Esta mensagem (enviada em *broadcast*) contém um campo que define a duração máxima do CFP (CFP_{max}). Todas as estações que recebem a mensagem de *beacon* atualizam seu NAV (*Network Allocator Vector*) com o valor definido no campo CFP_{max} . O objetivo é bloquear o acesso ao meio para dispositivos que estão a utilizar a função DCF.

O NAV é um mecanismo CCA virtual utilizado pelos dispositivos IEEE 802.11. Este mecanismo usa o valor do campo *duration time* existente no cabeçalho das mensagens IEEE 802.11 para inferir por quanto tempo o meio vai ser utilizado por uma estação que esteja a transmitir. Assim, como o valor do NAV é decrementado ao longo do tempo, uma estação só pode efetuar uma transmissão após o seu NAV chegar a zero.

Como segurança adicional para prevenir interferências, todas as transmissões efetuadas no CFP são separadas por $aSIFSTime$ ou $aPIFSTime$ (Figura 2.4). Como ambos os IFS são menores que o $aDIFSTime$ (utilizado pela função DCF), as estações PCF conseguem manter o controlo do acesso ao meio sem interferências de estações DCF (Figura 2.7).

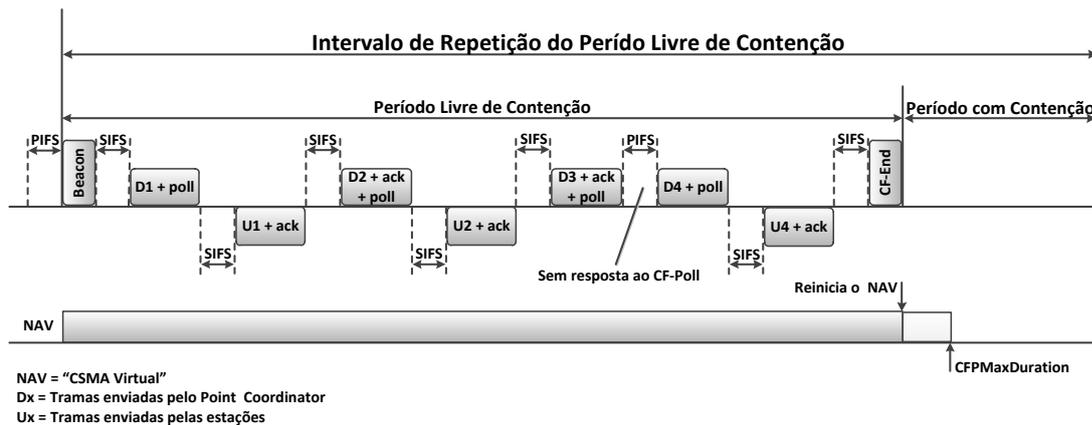


Figura 2.7: Transmissão PCF (adaptada de [2]).

Após o PC ter obtido o controlo do meio, este envia mensagens de autorização (denominadas CF-Poll – *Contention-Free Polling*) para as estações que estão na lista de *polling* (a lista de estações que solicitaram autorização para operarem durante o CFP) para que estas realizem as suas transmissões. Durante o CFP, as estações podem transmitir apenas se receberem uma mensagem CF-Poll do PC. Cada mensagem de autorização possibilita que a estação transmita apenas uma mensagem de dados, independentemente do seu tamanho. Para transmitir múltiplas mensagens de dados, a estação deve receber múltiplas mensagens CF-Poll do PC.

Para assegurar que o PC não perca o controlo do acesso ao meio, caso alguma estação autorizada não inicie sua transmissão no período de tempo alocado a si, após $aPIFS_{Time}$, é enviada pelo PC uma nova mensagem CP-Poll para a próxima estação da lista de *polling*. O CP é iniciado logo a seguir ao CFP. Este deve ter uma duração mínima igual ao tempo necessário para a transmissão (com confirmação) de uma mensagem de dados com um MPDU (*MAC Protocol Data Unit*) de tamanho máximo⁶.

A primeira transmissão dentro do CFP não ocorre necessariamente logo após o seu início. Por vezes, é possível que uma transmissão baseada num serviço com contenção, ultrapasse o final do CP. Quando isto ocorre, dizemos que o CFP foi encurtado (Figura 2.8). Caso exista uma transmissão a decorrer no momento em que a mensagem de *Beacon* deveria ser enviada, esta transmissão tem permissão para ser finalizada. Como consequência, uma vez que a mensagem de *Beacon* anuncia o início do próximo CFP, este é encurtado de acordo com o atraso sofrido. Para evitar um “efeito cascata”, este novo CFP deve ser finalizado antes do próximo instante esperado para a transmissão da mensagem de *Beacon*, referido como TBTT (*Target Beacon Transmission Time*). O PC pode terminar um CFP, antes da sua duração máxima, transmitindo uma mensagem denominada CF-End (*Contention-Free End*). Esta decisão pode ser baseada no tamanho da lista de *polling*, carga da rede, ou qualquer outro fator que o PC julgue importante.

⁶O tamanho máximo de um MPDU de acordo com a norma IEEE 802.11 é de 4095 bytes.

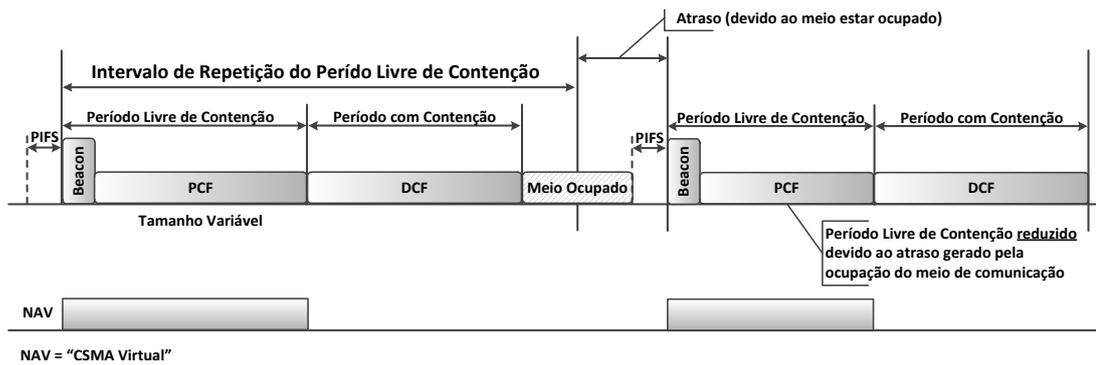


Figura 2.8: CFP encurtado (adaptada de [2]).

Diferentemente da função DCF, a função PCF opera sem o mecanismo de *backoff* nas estações durante o CFP. Neste contexto, há o risco de ocorrerem repetidas colisões, se múltiplos PC estiverem a operar no mesmo canal de comunicação. Para minimizar este risco, o PC pode opcionalmente utilizar um *aDIFSTime* mais um valor aleatório de *backoff* (onde o CW varia num intervalo de $[1, aCW_{min}]$) antes de iniciar um novo CFP, caso a mensagem de *Beacon* seja atrasada devido ao meio estar ocupado. O PC pode também escolher utilizar este valor de *backoff* durante o CFP antes de efetuar uma retransmissão.

2.3.3 EDCA - Enhanced Distributed Channel Access

O mecanismo EDCA foi desenvolvido com o objetivo de melhorar o serviço fornecido pela função DCF. A sua característica principal é a diferenciação na transmissão das mensagens através do uso de quatro categorias de acesso (*Access Categories – AC*). Cada mensagem que chega à subcamada MAC com uma prioridade pré-definida é alocada numa das quatro AC (*voz, vídeo, best-effort e background*). Todas as AC definidas pelo mecanismo EDCA são baseadas nos oito níveis de prioridade (UP – *User Priority*) definidos pela norma IEEE 802.1D (Tabela 2.2). As categorias de voz e *background* são, respectivamente, as categorias com maior e menor prioridade.

Tabela 2.2: Comparação entre IEEE 802.1D e IEEE 802.11e.

802.1D		802.11e	
UP	Designação	Categoria de Acesso	Designação
1	BK - <i>Background</i>	0	BK - <i>Background</i>
2	Reservado		
0	BE - <i>Best Effort</i>	1	BE - <i>Best Effort</i>
3	EE - <i>Excellent Effort</i>		
4	CL - <i>Carga Controlada</i>	2	VI - <i>Vídeo</i>
5	VI - <i>Vídeo</i>		
6	VO - <i>Voz</i>	3	VO - <i>Voz</i>
7	NC - <i>Controlo de Rede</i>		

A diferenciação entre os níveis de serviços fornecidos por cada uma das AC baseia-se em três mecanismos independentes: **i)** AIFS (*Arbitration Interframe Space*); **ii)** duração da TXOP; e **iii)** tamanho do CW.

Tal como o IFS, o AIFS define um intervalo de tempo mínimo pelo qual o meio deve permanecer livre antes que uma estação possa tentar efetuar sua transmissão. São atribuídos diferentes intervalos às diferentes AC. A duração de cada intervalo é dada pela seguinte equação:

$$AIFS_{[AC]} = AIFSN_{[AC]} \times aSlotTime + aSIFSTime \quad (2.6)$$

onde o $AIFSN_{[AC]}$ é um inteiro positivo que deve ser maior ou igual a 2 para todas as estações que implementem o mecanismo EDCA (denominadas pela norma como *QoS Stations* – QSTA), e igual ou maior a 1 para todos os AP que implementem o mecanismo EDCA (denominados pela norma como *QoS Access Points* – QAP). A Figura 2.9 apresenta a relação entre os diferentes AIFS do mecanismo EDCA e os IFS descritos anteriormente.

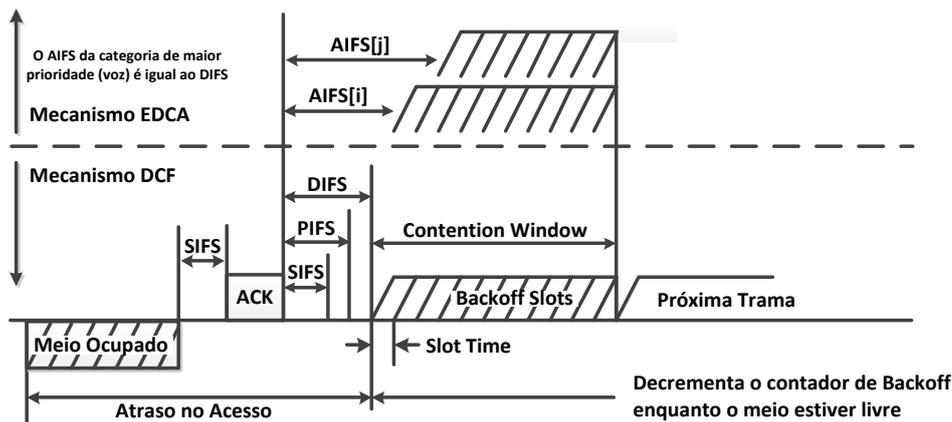


Figura 2.9: Relação entre IFS e AIFS no mecanismo EDCA (adaptada de [2]).

A TXOP é um intervalo de tempo, no qual, uma estação é capaz de manter o controlo do acesso ao meio. Isto possibilita que a estação possa enviar quantas mensagens forem possíveis dentro deste intervalo. Se uma mensagem for grande demais para ser transmitida numa única TXOP, pode ser fragmentada em mensagens menores. O uso das TXOP reduz o problema de equidade (*fairness*) encontrado na função DCF, ou seja, evita que as estações que estejam a enviar mensagens longas e que utilizem uma baixa taxa de transmissão ocupem o meio de comunicação por demasiado tempo.

Os valores das TXOP fornecidas às estações, através do mecanismo EDCA, são definidos a partir dos valores de $TXOP_{max}$ de cada AC. Estes valores variam em função da cada camada física utilizada (Tabela 2.3). Quando o valor de uma TXOP é definido como zero, significa que apenas uma MSDU (*MAC Service Data Unit*) ou uma MMPDU (*MAC Management Protocol Data Unit*) pode ser transmitida pela estação, independentemente do seu tamanho.

Tabela 2.3: Parâmetros das Categorias de Acesso.

AC	CW_{min}	CW_{max}	AIFSN	$TXOP_{max}^{a/g/n}$	$TXOP_{max}^b$
0	aCW_{min}	aCW_{max}	7	0	0
1	aCW_{min}	aCW_{max}	3	0	0
2	$(aCW_{min} + 1)/2 - 1$	aCW_{min}	2	3008 μs	6016 μs
3	$(aCW_{min} + 1)/4 - 1$	$(aCW_{min} + 1)/2 - 1$	2	1504 μs	3264 μs

Para diferenciar cada uma das AC quando o procedimento de *backoff* é invocado, são utilizados diferentes intervalos de CW. Em contraste com a função DCF, onde os valores iniciais eram aleatoriamente selecionados dentro de um intervalo de $[0, CW]$, o tamanho de CW no mecanismo EDCA é definido pelos atributos $CW_{min}[AC]$ e $CW_{max}[AC]$, que são diferentes para cada AC. Neste esquema, o procedimento de *backoff* seleciona um número aleatório num intervalo de $[0, CW]$, onde o valor do CW é inicialmente definido por $CW_{min}[AC]$. Quando uma transmissão falha, o CW é incrementado por $[(oldCW)[AC] + 1] \times PF - 1$, onde, PF é um fator de persistência (seu valor padrão é $PF = 2$) e o valor máximo do CW limitado por $CW_{max}[AC]$. Os valores padrão de CW_{min} e CW_{max} para cada AC são também apresentados na Tabela 2.3.

2.3.4 HCCA - HCF Controlled Channel Access

O mecanismo HCCA foi proposto na norma IEEE 802.11e e como uma melhoria da função PCF. É baseado num escalonador do tipo *Round-Robin*, o qual, tem como objetivo garantir um tempo de resposta limitado nas transmissões. De forma semelhante à função PCF, o mecanismo HCCA utiliza um controlador central denominado HC (*Hybrid Coordinator*), que organiza a sequência de transmissão dos fluxos de dados das estações.

O HCCA utiliza um mecanismo de especificação de tráfego, o qual, permite que as estações solicitem ao HC uma TXOP de acordo com as suas necessidades. Isto evita que todas as estações obtenham tempos de acesso ao meio iguais. O HC envia mensagens de autorização (denominadas QoS CF-Polls) para *todas* as estações na lista de *polling*, mesmo as que não tenham nenhuma mensagem para ser transmitida no CFP corrente. Quando isto ocorre, a estação irá transmitir uma mensagem (denominada QoS CF-Null) com o campo de dados vazio. Tal como a função PCF, o HCCA utiliza o mecanismo de NAV e os menores valores IFS ($aPIFS_{Time}$ e $aSIFS_{Time}$) para garantir o controlo do acesso ao meio.

O mecanismo HCCA pode operar em dois modos diferentes: um contém somente o CFP, e outro intercala o CFP com o CP. No primeiro modo, apenas estações HCCA podem transmitir, no segundo modo, denominado HEMM (*HCCA, EDCA mixed mode*), as estações HCCA compartilham o meio com estações EDCA durante o CP (Figura 2.10).

Uma grande vantagem do mecanismo HCCA sobre a função PCF é que, mesmo no CP poderão ser enviadas mensagens caso o HC ache necessário. Como é utilizado um $aPIFS_{Time}$ para o envio de uma mensagem *QoS CF-Polls*, o HC consegue obter o acesso ao meio antes das estações EDCA.

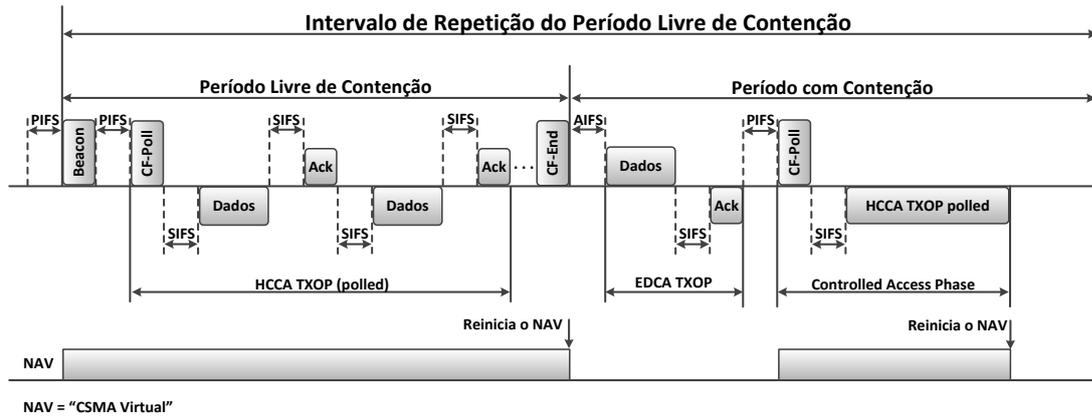


Figura 2.10: Transmissão HCCA (adaptada de [2]).

Seguidamente, o HC pode controlar o meio quanto tempo for necessário. O intervalo de tempo no qual o HC controla o acesso ao meio no CP é denominado de CAP (*Controlled Access Phase*).

Para iniciar um novo CFP, o HC envia uma mensagem de *beacon* para todas as estações, definindo o instante de início e a sua duração. Após isso, o HC autoriza o acesso ao meio oferecendo uma TXOP para cada estação que está na lista de *polling*. As estações respondem após *aSIFSTime* com uma mensagem de dados ou com uma mensagem *QoS CF-Null*, indicando que não têm mensagens na sua fila de transmissão, ou então que a TXOP alocada não é suficiente para a transmissão da mensagem pretendida. Caso após *aPIFSTime*, nenhuma transmissão seja detetada pelo HC, então este, envia uma autorização para a próxima estação na sua lista de *polling*. O CFP termina quando a sua duração expirar ou quando o HC enviar uma mensagem *QoS CF-End* (*QoS Contention-Free End*).

2.4 Mecanismo de Controlo de Admissão

O objetivo principal do mecanismo de controlo de admissão é limitar a quantidade de tráfego admitida por uma classe de serviço de forma que a QoS dos fluxos existentes não seja degradada, maximizando ao mesmo tempo a utilização dos recursos do meio de comunicação [19].

Na função HCF são definidos dois mecanismos de controlo de admissão distintos: um para serviços com contenção (EDCA) e outro para serviços livre de contenção (HCCA). Estes mecanismos possibilitam ao HC alocar de forma inteligente os recursos disponíveis no meio de comunicação e aceitar, ou rejeitar, a admissão de novos fluxos de dados. Estas decisões baseiam-se em informações enviadas pelas estações ao HC acerca das características dos fluxos de dados que a estação deseja transmitir. O envio destas informações é realizado utilizando o elemento TSPEC (*Traffic Specification*) definido na norma IEEE 802.11.

2.4.1 TSPEC - *Traffic Specification*

O objetivo do TSPEC é solicitar uma reserva de recursos ao HC, de forma a modificar o comportamento do seu escalonador. Assim, é possível evitar o congestionamento do meio de comunicação, aumentando a qualidade das transmissões. Neste contexto, o TSPEC permite que uma estação envie ao HC os requisitos de QoS que um determinado fluxo de dados (TS – *Traffic Stream*) necessita. Uma TS pode ser caracterizada por um grupo de MSDU transmitidas para uma estação específica. A sua identificação no tráfego *uplink* é dada pelo TSID (*Traffic Stream Identified*) e direção, e no tráfego *downlink* pelo TSID, direção e endereço da estação.

O uso do TSPEC pode ser imposto pelo HC ativando o campo *Mandatory Admission Control* na mensagem de *beacon*. Assim, uma estação que queira transmitir necessita primeiro de enviar uma requisição ADDTS (*Add Traffic Stream*) ao HC. Nesta mensagem são incluídos os parâmetros TSPEC da TS que a estação deseja transmitir. Ao receber a requisição ADDTS, o HC decide se admite a TS, recusa a TS, ou recusa a TS e sugere um TSPEC alternativo para ser avaliado pela estação. Esta decisão é enviada à estação pelo HC através de uma resposta ADDTS (Figura 2.11).

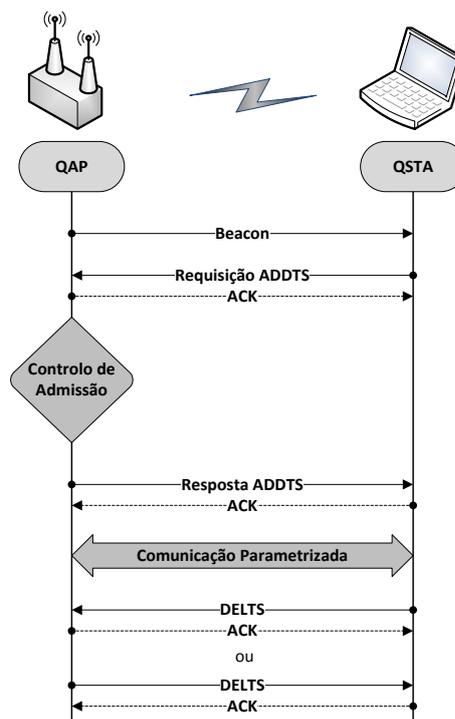


Figura 2.11: Fluxo das mensagens TSPEC (adaptada de [3]).

Caso a resposta ADDTS indique que a TS foi admitida, então esta é inserida na lista de *polling*. Caso contrário, o processo de solicitação pode ser repetido utilizando o mesmo TSID, mas com parâmetros TSPEC diferentes. Os parâmetros TSPEC das TS aceites pelo HC podem ser renegociados, através do envio de uma nova requisição ADDTS.

Uma estação pode cancelar uma TS admitida pelo HC através do envio de uma mensagem DELTS (*Delete Traffic Stream*), ou então, o próprio HC pode cancelar uma TS admitida por ele, enviando uma mensagem DELTS à estação a qual ela pertence. As TS podem ser canceladas por diversas razões, entre as quais se incluem:

- a estação decide que a TS não é necessária;
- o HC ou a estação determinam que não há mais atividade na TS;
- o tráfego foi recebido por um TSID desconhecido;
- após o tempo máximo de espera por uma resposta ADDTS.

O TSPEC (Figura 2.12) contém uma série de parâmetros que definem as características e requisitos QoS de uma TS específica. Estes parâmetros são utilizados pelo HC e pelas estações QoS para que seja possível se definir o melhor escalonamento para uma TS.

ID do Elemento	Comprimento	Info da TS	Tamanho Nominal da MSDU	Tamanho Máximo da MSDU	Intervalo Mínimo de Serviço	Intervalo Máximo de Serviço	Intervalo de Inatividade	Intervalo de Suspensão
Instante Inicial do Serviço	Taxa Mínima de Dados	Taxa Média de Dados	Taxa Máxima de Dados	Tamanho do Burst	Atraso Limite	Taxa Mínima PHY	Largura de Banda Excedente	Tempo do Meio

Figura 2.12: Formato do TSPEC (adaptada de [4]).

Seguidamente são detalhados os principais parâmetros do elemento TSPEC utilizados tanto pelas estações, quanto pelo HC para especificar os requisitos QoS e as características de uma TS:

- **Tamanho Nominal da MSDU:** tamanho nominal das MSDU (em *bytes*) de uma TS;
- **Tamanho Máximo da MSDU:** tamanho máximo das MSDU (em *bytes*) de uma TS;
- **Intervalo Mínimo de Serviço:** intervalo mínimo (em μs) entre dois serviços consecutivos;
- **Intervalo Máximo de Serviço:** intervalo máximo (em μs) entre dois serviços consecutivos;
- **Intervalo de Inatividade:** quantidade de tempo mínima (em μs) que uma TS pode permanecer sem receber ou transferir nenhum MSDU antes que esta seja excluída pelo HC;
- **Intervalo de Suspensão:** quantidade de tempo mínima (em μs) que uma TS pode permanecer sem receber ou transferir nenhum MSDU antes que o envio (pelo HC) de mensagens *CF-Poll* seja cancelado. É possível desativar este parâmetro, indicando que o *polling* para esta TS não deve ser interrompido devido a sua inatividade. O valor do Intervalo de Suspensão é sempre menor ou igual ao Intervalo de Inatividade;

- **Instante Inicial do Serviço:** instante de tempo (em μs) do início do primeiro serviço escalonado. Isto permite ao HC escalonar os serviços de modo que as MSDU sofram um menor atraso na subcamada MAC;
- **Taxa Mínima de Dados:** menor taxa de geração de dados (em bits/segundo) definida pelo AP para transmitir as MSDU pertencentes a uma TS (sem incluir os *overheads* de MAC e PHY), respeitando os limites impostos pelo seu TSPEC;
- **Taxa Média de Dados:** taxa média de geração de dados (em bits/segundo) definida pelo AP para transmitir as MSDU pertencentes a uma TS (sem incluir os *overheads* de MAC e PHY), respeitando os limites impostos pelo seu TSPEC;
- **Taxa Máxima de Dados:** taxa máxima de geração de dados (em bits/segundo) permitida para que seja possível transmitir as MSDU pertencentes a uma TS, respeitando os limites impostos pelo seu TSPEC;
- **Tamanho do Burst:** quantidade máxima de dados (em bytes) pertencentes a uma TS que podem chegar ao HC utilizando a taxa máxima de dados;
- **Atraso Limite:** quantidade máxima de tempo (em μs) permitida para se transmitir uma MSDU (com confirmação) pertencente a um TS utilizando os parâmetros TSPEC definidos anteriormente. Este valor é estimado pela diferença de tempo entre a chegada da MSDU na subcamada MAC da estação origem até a entrega (com sucesso) da MSDU na subcamada MAC da estação destino (passando pelo AP);
- **Taxa Mínima PHY:** taxa mínima de transmissão (em bits/segundo) disponível na camada PHY;
- **Largura de Banda Excedentária:** define a alocação de tempo (em μs) excedentária (além do indicado pela aplicação) para transmitir as MSDU pertencentes à TS. Este campo leva em consideração as retransmissões das mensagens;
- **Tempo do Meio:** quantidade de tempo (em múltiplos de $32 \mu\text{s}$) admitida para se aceder ao meio. Este campo é definido pelo HC nas mensagens de resposta ADDTS.

2.4.2 Mecanismo de Controlo de Admissão do EDCA

No mecanismo EDCA, o controlo de admissão é apenas suportado pelas AC de voz e vídeo. As restantes AC permitem que o tráfego seja transmitido normalmente, sem que exista a necessidade de uma aprovação prévia. Neste contexto, um QAP pode indicar às suas estações quais as AC que necessitam de controlo de admissão. Esta indicação é realizada através do campo ACM (*Admission Control Mandatory*), que permanece fixo por toda a duração da BSS. No entanto, os parâmetros CW_{min} , CW_{max} , AIFS, e TXOP podem ser ajustados ao longo do tempo pelo QAP.

Para que uma estação solicite uma TXOP para uma nova TS, esta deve enviar uma requisição ADDTS ao QAP. Esta deve conter a UP (802.1D) associada ao tráfego que deseja ser transmitido

e também a política de acesso do EDCA. O QAP deve associar a UP recebida pela requisição ADDTS com a AC apropriada (de acordo com a Tabela 2.2).

Quando o QAP recebe a requisição ADDTS, ele pode aceitá-la, ou rejeitá-la, utilizando para isso um algoritmo local. Se decidir aceitar a requisição, deve também calcular o *Tempo no Meio* das informações contidas no TSPEC. Este valor será enviado juntamente com a resposta ADDTS.

Para que seja possível efetuar o cálculo anterior, devem ser levados em consideração dois fatores: os requisitos do tráfego e os requisitos do meio. Ambos são caracterizados por parâmetros TSPEC enviados pela estação. No caso dos requisitos do tráfego, os parâmetros são: *Taxa Média de Dados* (ρ) e *Tamanho Nominal da MSDU* (L). No caso dos requisitos do meio, os parâmetros são: *Largura de Banda Excedentária* (β) e *Taxa Mínima PHY* (ϕ). Com estas informações é possível obter o valor do *Tempo no Meio* utilizando a seguinte equação:

$$\text{Tempo no Meio} = \beta \times \left\lceil \frac{(\rho/8)}{L} \right\rceil \times \text{MPDUExchangeTime} \quad (2.7)$$

onde *MPDUExchangeTime* é o tempo necessário (em μs) para se transmitir a sequência MPDU. Para o caso de uma MPDU transmitida com uma política normal de confirmação e sem a proteção RTS/CTS, este valor é igual ao tempo necessário para se transmitir a MPDU mais o tempo necessário para se receber a mensagem de confirmação (ACK). Este tempo pode ser definido pela seguinte equação:

$$\text{MPDUExchangeTime} = C_L^\phi + aSIFSTime + C_{ACK} \quad (2.8)$$

onde C é a primitiva PLME-TXTIME, a qual retorna a duração (em μs) de uma mensagem baseada no tamanho do seu campo de dados (L) e na taxa mínima PHY (ϕ) utilizada pela estação.

Quando a resposta ADDTS é recebida pela estação indicando a sua aceitação, esta irá organizar o controlo de admissão utilizando duas variáveis: *Tempo Admitido*, que é o tempo (em μs) autorizado pelo QAP para que a estação tenha acesso ao meio; e *Tempo Utilizado*, que é o tempo (em μs) decorrido de utilização do meio pela estação. Ambos são inicializados com o valor 0 no momento da (re)associação.

No caso do *Tempo Admitido*, seu valor é calculado utilizando como base o valor do campo *Tempo no Meio* enviado pelo QAP. Este processo é executado utilizando a seguinte equação:

$$\text{Tempo Admitido} = \text{Tempo Admitido} + \text{dot11EDCAveragingPeriod} \times \text{Tempo no Meio} \quad (2.9)$$

onde o *dot11EDCAveragingPeriod* indica o intervalo de tempo (em segundos) em que os parâmetros *Tempo Admitido* e *Tempo Utilizado* devem ser recalculados. A valor padrão para este parâmetro é 5 (definido pela emenda IEEE 802.11e).

Uma estação pode escolher cancelar um pedido específico a qualquer momento. Para isso, deve transmitir uma mensagem DELTS ao QAP, contendo o TSID e a direção da TS. Se a estação

enviar ou receber uma mensagem DELTS, deve recalcular o *Tempo Admitido* de acordo com a seguinte equação:

$$\text{Tempo Admitido} = \text{Tempo Admitido} - \text{dot11EDCAveragingPeriod} \times \text{Tempo no Meio} \quad (2.10)$$

O valor da variável *Tempo Utilizado* pode ser atualizado em duas diferentes situações:

- Em intervalos de cada *dot11EDCAveragingPeriod*:

$$\text{Tempo Utilizado} = \max[(\text{Tempo Utilizado} - \text{Tempo Admitido}), 0] \quad (2.11)$$

- Após cada tentativa de transmissão ou retransmissão de uma MPDU, independentemente de esta ser finalizada com sucesso ou não:

$$\text{Tempo Utilizado} = \text{Tempo Utilizado} + \text{MPDUExchangeTime} \quad (2.12)$$

Se o valor do *Tempo Utilizado* atingir ou exceder o valor do *Tempo Admitido*, a estação deve imediatamente deixar de transmitir utilizando a AC solicitada. No entanto, esta estação pode continuar suas transmissões utilizando outra AC que não necessite de aprovação prévia pelo controlo de admissão.

2.4.3 Mecanismo de Controlo de Admissão do HCCA

O mecanismo HCCA foi desenvolvido para suportar parametrização QoS do tráfego transmitido. Desta forma, uma estação pode enviar ao HC as características e os requisitos QoS de uma TS_k que deseja transmitir. Se esta TS for admitida, o HC define um escalonamento de forma a cumprir todos estes requisitos.

O escalonamento do HCCA baseia-se num esquema *Round-Robin*, que obriga as estações a transmissão dos seguintes parâmetros TSPEC: *Taxa Média de Dados* (ρ), *Tamanho Nominal da MSDU* (L), e o *Intervalo Máximo de Serviço* (SI_{max}) ou *Atraso Limite* (D). Se ambos parâmetros SI_{max} e D forem enviados, o escalonador utiliza SI_{max} para efetuar os cálculos.

Quando um novo pedido é recebido pelo HC, a unidade de controlo de admissão (ACU – *Admission Control Unit*) realiza o processo de verificação das condições solicitadas pela TS_k executando os passos seguintes:

1. Inicialmente a ACU verifica qual o menor de todos os SI_{max} pertencente às TS já admitidas pelo HC e à TS_k ;
2. Seguidamente, a ACU seleciona um novo SI , que será comum à todas as TS admitidas pelo sistema. Este valor deve ser inferior ao menor SI_{max} encontrado no passo anterior e, deve ser submúltiplo do intervalo de envio das mensagens de *beacon*. Na Figura 2.13, é

apresentado um exemplo de uma TS_i admitida. Neste caso específico, o intervalo entre o envio das mensagens de *beacon* é de 100 milissegundos e o SI_{max} para a TS_i é de 60 milissegundos. Utilizando as regras previamente apresentadas, a ACU escolhe um SI igual a 50 milissegundos.

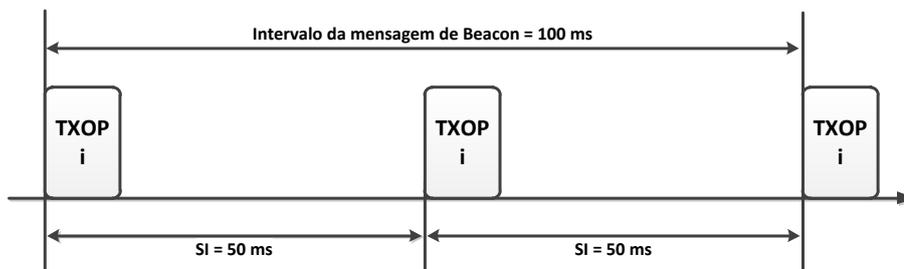


Figura 2.13: Cálculo do SI no mecanismo HCCA (adaptada de [4]).

- Em seguida a ACU calcula o número de MSDU que serão enviadas ao HC (N_k). Para isto são utilizados os valores do novo SI e da taxa média de dados (ρ) enviada pela estação. Este valor é dado pela seguinte equação:

$$N_k = \left\lceil \frac{SI \times \rho_k}{L_k} \right\rceil \quad (2.13)$$

- Seguidamente, a ACU calcula a duração da TXOP que precisa ser alocada à TS_k , dado pela seguinte equação:

$$TXOP_k = \max \left(\frac{N_k \times L_k}{R_k} + O, \frac{M}{R_k} + O \right) \quad (2.14)$$

onde R_k é a taxa mínima PHY, M é o tamanho máximo de um MSDU⁷, e O é o *overhead* em unidades de tempo. O *overhead* inclui os tempos IFS, AIFS e ACK.

- Por fim, a ACU determina se a TS_k pode ser admitida pelo sistema de comunicação caso a seguinte inequação seja satisfeita:

$$\frac{TXOP_k}{SI} + \sum_{i=1}^n \frac{TXOP_i}{SI} \leq \frac{T - T_{CP}}{T} \quad (2.15)$$

onde n é o número de TS previamente admitidas, T é o intervalo entre as mensagens de *Beacon* e T_{CP} é o tempo alocado para o tráfego EDCA no CP.

Uma observação importante neste processo é que, de acordo com a norma IEEE 802.11e, a taxa de transmissão física (R_k), utilizada pelo HCCA para o cálculo do tamanho da TXOP, deverá ser

⁷O tamanho máximo de um MSDU definido pela norma IEEE 802.11 é de 2304 bytes.

igual a menor taxa de transmissão disponibilizada pela camada física. Esta abordagem é utilizada para evitar a necessidade de se realizar novos cálculos das TXOP, caso a taxa de transmissão seja alterada pelo mecanismo de *auto-rate* em detrimento as interferências no canal de comunicação.

2.5 Conclusões

A utilização de tecnologias de comunicação sem fios cresceram progressivamente ao longo dos últimos anos. O aumento da necessidade de conectividade e de mobilidade tem impulsionado o desenvolvimento de novas tecnologias e o aperfeiçoamento de outras já existentes. Entre estas, a que ganhou maior destaque foi a definida pela norma IEEE 802.11. O seu baixo custo e facilidade de implementação conquistaram a indústria de computadores que, na sua grande maioria, acabaram por adotá-la como padrão nos seus dispositivos de comunicação.

Basicamente, a norma IEEE 802.11 define quatro mecanismos de controlo de acesso ao meio, que são implementados pela subcamada MAC. Estes mecanismos podem ser classificados entre os que disponibilizam um serviço de acesso ao meio baseado em contenção (DCF e EDCA) e os que disponibilizam um serviço de acesso ao meio livre de contenção (PCF e HCCA). No que concerne à implementação de um mecanismo de controlo de admissão e à definição de diferentes níveis de QoS, apenas o EDCA e o HCCA são capazes de fornecer este tipo de serviço.

Ao analisarmos os mecanismos disponibilizados pelo IEEE 802.11 sob a ótica da transmissão do tráfego de tempo-real, podemos observar algumas limitações (discutidas com maior detalhe no capítulo 3). Neste contexto, a que se destaca nos mecanismos DCF e PCF é a transmissão de todos os tipos de tráfego através de apenas uma categoria de acesso, ou seja, definindo assim um único nível de prioridade.

Embora o mecanismo EDCA apresente uma solução para esta limitação, estudos realizados [20] apontam que, ao operar num ambiente de comunicação aberto, o EDCA não é capaz de cumprir os requisitos temporais do tráfego de tempo-real, mesmo quando este é transmitido através da categoria de acesso de mais alta prioridade (voz) .

O mecanismo que reúne um conjunto de características mais favoráveis à transmissão do tráfego de tempo-real é o HCCA. Isto porque, com a utilização do elemento TSPEC para obter informações acerca das TS, é possível ao HC alocar de forma mais eficiente os recursos do meio de comunicação, além também de evitar sobrecargas no sistema. No entanto, estudos preliminares [21, 22] demonstraram que em algumas situações o HCCA não é capaz de garantir os requisitos temporais das mensagens de tempo-real.

Portanto, é possível concluir que, embora alguns dos mecanismos definidos pela norma IEEE 802.11 suportem a transmissão de tráfego com requisitos QoS, isto por si só não é o suficiente para a transmissão do tráfego de tempo-real. Isto evidencia a necessidade de propor novas soluções capazes de suportar um serviço de comunicação de tempo-real em redes IEEE 802.11 a operar em ambientes de comunicação abertos.

Capítulo 3

Trabalhos Relacionados

No capítulo anterior foram apresentados os mecanismos de controlo de acesso ao meio e controlo de admissão implementados pela norma IEEE 802.11, bem como, algumas das suas respectivas limitações no que concerne à transmissão de tráfego de tempo-real. Assim, evidenciou-se a necessidade de investigações nesta área, de forma a tornar possível a criação de novas propostas capazes de suportar a transmissão deste tipo de tráfego em redes IEEE 802.11. Neste contexto, o objetivo deste capítulo é, expandir de forma detalhada a discussão acerca das limitações apresentadas pela norma IEEE 802.11 e, analisar e classificar as propostas apresentadas pela comunidade científica. Como a maioria das propostas, ou focam em aspectos relacionados ao mecanismo de controlo de acesso ao meio, ou em aspectos relacionados ao mecanismo de controlo de admissão, optou-se por dividir este capítulo entre estes dois tipos de mecanismos para uma melhor organização e compreensão das características e limitações encontradas.

3.1 Mecanismos de Controlo de Acesso ao Meio

A norma IEEE 802.11 [4] define uma arquitetura para WLAN (*Wireless Local Area Networks*), que abrange as camadas física e de ligação de dados. No contexto da camada de ligação de dados, nomeadamente a subcamada MAC (*Media Access Control*), são definidos quatro mecanismos de controlo de acesso ao meio que podem ser implementados de forma centralizada ou distribuída. O acesso ao meio ocorre basicamente através da utilização de diferentes valores de IFS (*Inter-frame Space*) e do mecanismo de *backoff*, o qual é baseado num algoritmo probabilístico. Os mecanismos que fornecem um serviço de acesso ao meio livre de contenção também utilizam este algoritmo para tentar solucionar situações onde duas ou mais redes que fornecem este tipo de serviço se encontram sobrepostas. Neste contexto, e com base nas limitações da norma IEEE 802.11 apresentadas no capítulo anterior, a comunidade científica tem apresentado novas propostas para o mecanismo de controlo de acesso ao meio com o objetivo de suportar a transmissão do tráfego de tempo-real neste tipo de tecnologia.

Esta secção apresenta algumas destas propostas dividindo-as numa classificação baseada em três níveis. O *primeiro nível* está relacionado com a arquitetura da proposta, que pode adotar uma abordagem **Centralizada** ou **Distribuída**. Na abordagem centralizada é utilizado um dispositivo central para coordenar o acesso ao meio que, entre outras funções, irá definir qual a estação que terá acesso ao meio, em que instante e por quanto tempo. Na abordagem distribuída, esta decisão é tomada localmente pelas próprias estações.

O *segundo nível* de classificação baseia-se **na forma como as colisões são tratadas**, uma vez que o mecanismo de *backoff* implementado por cada estação não é capaz de fornecer garantias temporais às aplicações de tempo-real. Neste nível, classificou-se as colisões de acordo com as seguintes estratégias [23]:

- **Evitar Colisões:** o acesso ao meio é realizado através de um serviço livre de contenção que, tem por objetivo tentar evitar a ocorrência de colisões;
- **Resolver Colisões:** substitui-se o algoritmo tradicional de *backoff* (baseado num esquema probabilístico) por um algoritmo que garanta um limite temporal;
- **Reduzir Colisões:** utilizam-se algoritmos distribuídos fracamente acoplados com o objetivo de se reduzir o número de colisões.

O *terceiro nível* de classificação baseia-se na grande disseminação no mercado de dispositivos que utilizam uma tecnologia de comunicação sem fio baseada na norma IEEE 802.11. Desta forma, faz-se necessário avaliar o **grau de compatibilidade das estações de tempo-real** sob dois aspectos considerados importantes:

- **IEEE 802.11:** a prioridade no acesso ao meio do tráfego de tempo-real é mantida mesmo na presença de estações IEEE 802.11 padrão a operar no mesmo ambiente de comunicação;
- **COTS:** a implementação das estações de tempo-real é compatível com *hardware* COTS (*Commercial-Off-The-Shelf*), não necessitando a utilização de *hardware* específico.

É importante observar que os aspectos apontados no *terceiro nível* de classificação podem apresentar-se de forma independente em cada proposta, não sendo estes mutuamente exclusivos.

3.1.1 Abordagens Centralizadas

Nesta subsecção são apresentadas as propostas que utilizam uma abordagem centralizada do mecanismo de controlo de acesso ao meio, sendo estas também subdivididas de acordo com a forma como são tratadas as colisões (*segundo nível* de classificação anteriormente definido).

3.1.1.1 Evitar Colisões

O PCF (*Point Coordination Function*) foi desenvolvido como um mecanismo opcional de controlo de acesso ao meio. Implementa um esquema de *polling* centralizado para suportar transmissões síncronas de dados, onde o PC (*Point Coordinator*) desempenha o papel de mestre, fornecendo assim um serviço de acesso ao meio livre de contenção. Esta abordagem faz com que as estações associadas ao PC possam transmitir as suas mensagens apenas após a receção de uma autorização. Como o PC é executado no AP (*Access Point*), este mecanismo é restrito às redes infraestruturadas. A sua principal limitação reside no facto do PC não ser capaz de prever o tamanho das mensagens transmitidas por cada TS (*Traffic Stream*), o que pode introduzir assim um tempo de transmissão variável. Além disso, a taxa de transmissão das estações pode mudar devido a diferentes características do ambiente, impedindo assim que este serviço possa garantir tempos de resposta confiáveis.

O mecanismo HCCA (*HCF Controlled Channel Access*), proposto como uma melhoria do PCF, baseia-se também num esquema de *polling*. Tem como principal objetivo fornecer um serviço de comunicação com tempo de resposta limitado superiormente. De forma semelhante ao PCF, o HCCA implementa um coordenador central denominado HC (*Hybrid Coordinator*), que distribui autorizações de transmissão a *todas* as estações associadas a ele, mesmo quando estas não tenham nenhuma mensagem para transmitir. Quando isto ocorre, a estação irá transmitir uma mensagem com o campo de dados vazio (denominada *Null Frame*). Este processo de alocação de tempo para estações que não tenham mensagens para transmitir é considerado um *overhead* no mecanismo HCCA [24]. Além disso, estudos preliminares [21, 22] demonstraram que o HCCA pode não ser capaz de garantir os requisitos de tempo-real esperados.

Uma limitação comum a ambos os mecanismos PCF e HCCA é a sua utilização em termos práticos, uma vez que a maioria dos dispositivos WLAN comerciais nunca os implementaram em função das suas complexidades [25].

Para tentar solucionar o problema de *overhead* do HCCA diversos autores propuseram melhorias. Em [24], Son *et al.* apresentam um esquema de *polling* onde o HC pune as estações que recebem uma autorização mas que não tenham mensagens para transmitir. Assim, sempre que uma estação transmite uma mensagem vazia, esta permanecerá durante um período de tempo pré-determinado sem receber uma nova autorização. A principal limitação desta proposta é que se um fluxo de dados tiver um intervalo de serviço maior que o intervalo de serviço definido pelo HC (comum à todos os fluxos de dados admitidos), esta estação poderá ser erradamente punida. Este problema pode também ocorrer se o tráfego de tempo-real for aperiódico ou esporádico.

De forma diferente, Lo, Lee e Chen [26] definem um mecanismo denominado CP-Multipoll (*Contention Period Multipoll*) capaz de distribuir múltiplas autorizações. Para isto, foi incorporado o esquema de acesso do DCF (*Distributed Coordination Function*) ao mecanismo de *polling*, de forma a utilizar diferentes valores de *backoff* para múltiplos fluxos de dados gerados por estações associadas ao HC. Desta forma, cada estação necessita executar um procedimento de *backoff* após receber a mensagem CP-Multipoll. Além disso, com o objetivo de evitar repetidas colisões entre

estações de diferentes BSS (*Basic Service Set*) a operar no mesmo canal de comunicação, os valores atribuídos às mensagens *CP-Multipoll* da BSS vizinhas devem ser diferentes entre si.

Lee *et al.* [27] propuseram um mecanismo de *polling* baseado numa arquitetura Mestre-Escravo. Neste caso, o tempo no meio de comunicação é dividido em ciclos de transmissão definidos pelo VPP (*Virtual Polling Period*). Cada VPP é subdividido em múltiplos *slots* que podem ser alocados às estações escravo. Para definir a sequência de alocação, a estação mestre envia uma mensagem em *broadcast* contendo esta informação. Quando uma estação escravo recebe uma autorização, esta pode transmitir uma mensagem de resposta (com dados) para a estação mestre ou então diretamente para outra estação escravo.

Em [25, 28], Miorandi *et al.* apresentam também uma solução baseada numa arquitetura Mestre-Escravo para suportar comunicação de tempo-real em redes IEEE 802.11. Nesta abordagem, são entregues mensagens cíclicas às estações escravo através de requisições periódicas enviadas pela estação mestre. São apresentadas três diferentes técnicas para controlar o tráfego acíclico: a primeira consulta (no final do ciclo corrente) os escravos que sinalizam a presença de mensagens acíclicas a serem transmitidas. A segunda, permite que um escravo, após receber uma autorização, possa enviar diretamente mensagens acíclicas ao mestre. A terceira explora a natureza descentralizada do protocolo MAC da norma IEEE 802.11 onde, assim que uma mensagem acíclica é gerada, a estação escravo pode tentar efetuar a sua transmissão.

Em [29], Willig propôs um protocolo MAC denominado FTDMA (*Flexible Time Division Multiple Access*) que é baseado em um esquema de *polling*, onde em cada ciclo uma estação base envia autorizações a todas as estações registadas. Os ciclos são logicamente subdivididos em fases: *SYNC* (utilizado pela estação base para enviar as demais estações mensagens de sincronização do relógio), *Polling* (utilizado pela estação base para enviar as autorizações de transmissão), *Reservation* (utilizado pelas estações para indicarem à estação base qual o tempo necessário para sua transmissão), *Register* (utilizado pelas estações para se associarem a estação base), *Current Scheduler* (utilizado pela estação base para enviar via *broadcast* a lista de escalonamento) e *Data Transfer* (utilizado pelas estações para transmitir suas mensagens de dados). A principal vantagem do FTDMA sobre o TDMA tradicional é a possibilidade do reaproveitamento dos *slots* livres.

A principal limitação encontrada nas quatro propostas apresentadas anteriormente reside no facto destas não considerarem as suas respectivas operações num ambiente de comunicação aberto. Desta forma, a existência de transmissões provenientes de redes que estejam fora da esfera de controlo da arquitetura de tempo-real pode resultar em perdas de *deadlines*.

Hantrakoon e Phonphoem [30] propuseram um gestor para a fila de transmissão e um mecanismo de controlo de admissão denominado PHCCA (*Priority-based HCCA*). O gestor da fila de transmissão modifica o HCCA dividindo-o em três classes diferentes, organizadas pelo tipo de tráfego ou pela relevância do utilizador. O objetivo é fornecer um serviço de QoS (*Quality of Service*) com garantia mínima de recursos (*starvation protection*) para a classe de menor prioridade. Por outro lado, o mecanismo de controlo de admissão implementa um *algoritmo de empréstimo de largura de banda* onde as classes de maior prioridade podem solicitar a largura de banda das classes de menor prioridade (detalhado na seção 3.2). Apesar das melhorias realizadas nesta abordagem,

o problema de *overhead* do HCCA não foi solucionado.

Uma avaliação experimental do *framework* RTnet [31] a operar numa rede IEEE 802.11 é apresentada por Boggia *et al.* em [32]. O RTnet foi originalmente proposto para comunicações de tempo-real *hard* em redes Ethernet. Neste contexto, a sua adaptação para redes IEEE 802.11 foi dada através da utilização de um esquema TDMA e da utilização do escalonador de tempo-real Xenomai [33]. De acordo com os autores, foram utilizados dispositivos sem fio contendo o *chipset* RT2500 [34] devido ao facto de ser o único suportado pelo RTnet. Esta abordagem é baseada numa arquitetura Mestre-Escravo onde a estação mestre gere a sincronização enviando mensagens periódicas às estações escravo. Com base nesta mensagem e no número de identificação atribuído à cada estação escravo, cada uma sabe em que momento se inicia e termina o seu *slot*. A principal limitação desta proposta está no facto de não ter sido levada em consideração a existência de tráfego externo a operar no mesmo canal de comunicação da rede de tempo-real, sendo este capaz de gerar interferências no sistema de sincronização.

Seno *et al.* propuseram em [35] uma extensão do protocolo EPL (*Ethernet Powerlink*) [36] para redes IEEE 802.11. Foram utilizados os mesmos princípios do EPL original que opera de acordo com um esquema TDMA implementado na subcamada MAC e divide cada ciclo em períodos isócronos e assíncronos. No período isócrono, a estação mestre define o início do ciclo enviando uma mensagem (em *broadcast*) denominada *Start of Cycle* (SoC). Seguidamente, esta envia mensagens de autorização para cada estação escravo solicitando as suas transmissões. No final do ciclo, a estação mestre envia (em *broadcast*) uma mensagem denominada *Start of Asynchronous* (SoA) para notificar o início do período assíncrono. A principal limitação desta proposta é a possibilidade de existir uma rede sobreposta a gerar tráfego de forma não controlada. Isto faz com que o período isócrono do sistema não seja unicamente utilizado pelas estações autorizadas, mas também por estações que estejam fora da esfera de controlo do sistema de comunicação de tempo-real. Uma avaliação experimental desta proposta é apresentada por Gamba *et al.* em [13].

3.1.1.2 Resolver Colisões

Em [37], Bartolomeu *et al.* propuseram o WFTT (*Wireless Flexible Time Triggered*) inspirado no paradigma FTT [38], o qual foi aplicado com sucesso em outras tecnologias de comunicação como *Controller Area Network* (FTT-CAN) [39] e *Ethernet* (FTT-Ethernet) [40]. O WFTT é uma abordagem baseada numa arquitetura Mestre-Escravo que visa explorar a capacidade do mecanismo de *bandjacking*¹ em aumentar a prioridade no acesso ao meio e também a flexibilidade, pontualidade e eficiência do FTT em suportar comunicações de tempo-real para aplicações abrangendo requisitos estáticos e/ou dinâmicos.

Assim, o WFTT divide o tempo de acesso ao meio em ciclos que são inicializados através da transmissão de uma mensagem de *trigger* (TM) efetuada pela estação mestre. Cada ciclo é dividido em três períodos: *Protected* (onde as mensagens são transmitidas sem contenção através do

¹O funcionamento do mecanismo de *bandjacking* consiste no envio de impulsos de energia para que estações ao redor da estação transmissora atrasem as suas transmissões por considerarem o meio ocupado.

mecanismo de *bandjacking*), *Unprotected* (onde as mensagens são transmitidas através do mecanismo de contenção padrão da estação) e *Inactive* (onde todas as estações escravo mudam para o modo *sleep* – com o objetivo de economizar bateria – até momentos antes da recepção da próxima TM). A sua principal limitação é a necessidade da utilização de um *hardware* específico para a sua implementação, impossibilitando assim o uso de *hardware* COTS.

3.1.1.3 Reduzir Colisões

Não foi encontrada na literatura nenhuma proposta baseada numa abordagem centralizada que pudesse ser classificada nesta categoria.

3.1.2 Abordagens Distribuídas

Nesta subseção são apresentadas as propostas que utilizam uma abordagem distribuída para controlar o acesso ao meio, sendo estas também subdivididas de acordo com o *segundo nível* de classificação anteriormente definido.

3.1.2.1 Evitar Colisões

Em [41], Cheng *et al.* apresentam um mecanismo de controlo de acesso ao meio denominado *Ripple*. Esta proposta baseia-se num esquema *Token-Passing* e utiliza mensagens de dados com um tempo de duração fixo. Basicamente esta abordagem modifica o procedimento de transmissão do mecanismo DCF e utiliza mensagens RTS (*Request-To-Send*) e RTR (*Ready-To-Receive*) como *tokens*. Desta forma, uma estação pode transmitir apenas quando estiver na posse do *token*.

Ergen *et al.* [42] propuseram o WTRP (*Wireless Token Ring Protocol*), que é um protocolo MAC que distribui *tokens* especiais e utiliza múltiplos relógios para manter a sincronização das estações. Neste modelo, cada estação pode transmitir durante um período de tempo pré-determinado. Caso exista tempo suficiente, a estação convida outras estações que estão fora do anel (rede de tempo-real) para transmitir. Esta solução foi proposta para eliminar a ineficiência do procedimento de *backoff* e o problema das colisões numa topologia em anel lógico.

As principais limitações encontradas nas duas propostas apresentadas anteriormente decorrem da utilização do *token* para controlar o acesso ao meio. Em situações onde possa ocorrer a perda do *token* (por exemplo, uma falha na estação que o detém), isto resultaria na necessidade de uma reconfiguração de todo o sistema.

3.1.2.2 Resolver Colisões

Em [43] Sobrinho e Krishnakumar apresentam uma proposta que adapta o mecanismo EQuB (BB – *Black-Burst*) [44] para uma rede sem fio em modo Ad Hoc. Neste contexto, as estações que implementam este mecanismo enviam impulsos de energia para criarem um nível de ocupação no meio de comunicação. Desta forma, e através da utilização de valores reduzidos de IFS, estas estações conseguem obter um maior nível de prioridade no acesso ao meio quando comparado

com estações IEEE 802.11 padrão. No entanto, torna-se necessária a desativação do mecanismo padrão de transmissão nestas estações.

Um esquema semelhante é apresentado por Hwang e Cho em [45], onde as estações utilizam períodos EB (*Energy-Burst*), similares ao BB, para aumentarem a prioridade das suas mensagens em relação às restantes. Cada estação possui um identificador virtual (VID) expresso como um valor binário e determinado pelo período de resolução da mensagem (VPRP). Desta forma, a estação que possuir o VID mais elevado vence a contenção. Já o AP, transmite as suas mensagens após *aPIFSTime* sem nenhum *backoff* ou contenção.

A operar de forma similar ao DCF, o BTPS (*Busy Tone Priority Scheduling*) proposto por Yang e Vaidya em [46] envia impulsos de energia (*busy tones*) para indicar às restantes estações a existência de uma mensagem de alta prioridade numa fila de transmissão. A diferenciação entre mensagens de alta e baixa prioridade é realizada através da utilização de diferentes valores de IFS e pelo mecanismo de *backoff*. Como as mensagens de alta prioridade são enviadas utilizando RTS/CTS, quando uma estação que contém mensagens de baixa prioridade recebe um impulso de energia, esta atrasa a sua transmissão até o valor indicado pelas mensagens RTS/CTS.

Em [47], Shew *et al.* propuseram um protocolo MAC baseado na abordagem de Sobrinho [43], complementado por uma árvore binária denominada árvore de contenção. Basicamente, o esquema BB é adotado para distinguir as prioridades de cada estação. A ideia é que uma estação possa obter um único ID, o qual depende de sua posição na árvore de contenção. Caso existam estações com a mesma prioridade, estas enviam mensagens utilizando um esquema *Round-Robin*.

A principal limitação das quatro propostas anteriores é a necessidade da utilização de dispositivos específicos de *hardware*, o que impossibilita as implementações utilizando *hardware* COTS. Além disso, de acordo com Yang e Vaidya [46], podem também ocorrer interferências de estações que por algum motivo (por exemplo, obstáculos) não recebam os sinais (*busy-tones*) que indicam a ocupação do meio.

Com o objetivo de gerar um comportamento mais previsível no controlo de acesso ao meio, Friedrich *et al.* apresentam em [48] o WRTMAC (*Wireless Real-Time Medium Access Control*), que é baseado no mecanismo EDCA. Esta abordagem substitui o AIFS por um tempo denominado RIFS (*Real-Time Interframe Space*). Este tempo é dado por $RIFS = aDIFSTime + i \times ST$, onde i é um inteiro que define a prioridade da mensagem e ST é igual a *aSlotTime*. A principal limitação desta proposta é a incapacidade de operar num ambiente de comunicação aberto, uma vez que os valores AIFS dos tráfegos de voz e vídeo são menores que os valores atribuídos pelo RIFS.

Em [49], Moraes *et al.* propuseram um mecanismo denominado VTP-CSMA (*Virtual Token Passing CSMA*), que tem como objetivo fornecer um serviço de comunicação de tempo-real através da passagem de um *token* virtual entre as estações de tempo-real. Como mecanismo complementar é utilizado um esquema que aumenta a prioridade do acesso ao meio de estações de tempo-real frente as demais estações [50]. Para efetuar a passagem do *token*, cada estação mantém um contador local que varia de 1 até o número de estações admitidas pelo sistema. Seu valor é incrementado por todas as estações após a transmissão de uma TXOP ou após um tempo pré-determinado (caso a estação que esteja de posse do *token* não efetua a sua transmissão). Desta

forma, este contador indicará a cada estação quem tem a posse do *token* no momento atual. Uma vez que o procedimento de passagem do *token* é implementado de forma independente em cada estação, o problema da sua perda não ocorre. No entanto, como para efetuar a sua "passagem" para outra estação o VTP-CSMA precisa, ou escutar a transmissão das mensagens da estação de posse do *token*, ou verificar que o meio permaneceu livre por um tempo pré-determinado, podem ocorrer problemas na sincronização entre as estações caso estas não consigam verificar estes requisitos devido a distâncias ou de bloqueios no ambiente de comunicação.

3.1.2.3 Reduzir Colisões

Entre as soluções que têm como objetivo principal reduzir o número de colisões, o mecanismo EDCA é a mais conhecida. Embora esta abordagem apresente uma melhoria quando comparada com o DCF, ainda são encontradas diversas limitações, nomeadamente quando se consideram ambientes de comunicação caracterizados por estações que geram uma carga de rede imprevisível [50]. Além disto, o indeterminismo do algoritmo de resolução de contenções não é capaz de fornecer um comportamento adequado às aplicações de tempo-real.

Uma avaliação do mecanismo EDCA foi apresentada por Cena *et al.* em [12]. Neste trabalho foi demonstrado que embora as mensagens enviadas ao AP sejam alocadas em diferentes filas determinadas de acordo com a sua AC (*Access Category*), a memória interna utilizada para este efeito é compartilhada. Neste contexto, em certas situações (nomeadamente carga elevada), podem ocorrer inversões de prioridades (*priority inversion*) quando mensagens de maior prioridade são descartadas pela falta de espaço em memória (*buffer overflow*). No referido trabalho, foi proposta uma alteração na gestão da memória interna do AP, fazendo com que as diferentes filas não compartilhem (de forma lógica) a mesma memória. Como consequência, isto reduziu drasticamente a taxa de mensagens perdidas em ambientes de carga elevada.

Villalón *et al.* [51] propuseram um mecanismo denominado B-EDCA, que é capaz de coexistir com estações a operar no modo DCF. Basicamente esta abordagem altera os valores de AIFS da fila de voz para $aSIFSTime + aSlotTime$ quando as estações estão em estado de *backoff*. Além disso, caso uma rede B-EDCA seja sobrepostas por outra rede HCCA, com o objetivo de manter a compatibilidade entre ambos, uma estação que implemente o mecanismo B-EDCA deve esperar por um intervalo adicional de $aSIFSTime$ quando o contador de *backoff* chegar a zero, ou seja, $2 \times aSIFSTime + aSlotTime$.

Em [52], Lopez-Aguilera *et al.* avaliaram o desempenho do mecanismo EDCA a operar de forma dessincronizada. Os autores definem o uso de valores AIFS que não sejam múltiplos de $aSlotTime$. Os resultados demonstraram que esta abordagem resolveu o problema de estrangulamento (*starvation*) do tráfego de menor prioridade.

Em [53], é proposto por Chang *et al.* o mecanismo H-EDCA (*High Performance EDCA*), o qual aplica um controlo distribuído e diferenciado das janelas de contenção. O principal objetivo é reduzir as colisões entre mensagens de uma mesma categoria de tráfego, resultantes da utilização de um mesmo valor de CW_{min} . Neste contexto, o H-EDCA controla o valor de CW_{min} através de uma cadeia de Markov que considera tanto o número de colisões, quanto o número de transmissões

efetuadas com sucesso para assim redefinir ser valor. Isto faz com que o CW_{min} de diferentes estações e diferentes categorias de acesso seja definido de forma individualizada.

Wang *et al.* [54] desenvolveram um mecanismo de resolução de tráfego denominado GDCF (*gentle DCF*). A diferença entre esta abordagem e o DCF tradicional é que o primeiro tem um comportamento mais conservador, reduzindo o valor do CW pela metade após transmitir consecutivamente com sucesso c mensagens, diferentemente do DCF, que retoma o valor mínimo do CW após uma única transmissão com sucesso. O GDCF mantém um contador de transmissões consecutivas efetuadas com sucesso, o qual retoma ao valor inicial após qualquer colisão.

Em Vittorio *et al.* [55] o mecanismo CWFC (*Contention Window Fuzzy Controller*) tenta reduzir o número de colisões através de uma adaptação dinâmica dos parâmetros de *backoff* (CW_{min} e CW_{max}) para as diferentes categorias de acesso. O controlo adaptativo é realizado através de um controlador de lógica *fuzzy*, que leva em consideração tanto o *throughput* quanto o número de retransmissões efetuadas por cada estação.

Similarmente ao CWFC, o DCC (*Dynamic Contention Control*) modifica o mecanismo EDCA de forma a ajustar dinamicamente do nível de prioridade em cada fluxo de dados. Este ajuste baseia-se na estimativa do atraso por salto (*hop*) realizado, e na geração de uma distribuição não uniforme dos tempos de *backoff* para a retransmissão de mensagens. Esta distribuição utiliza como base o atraso fim-a-fim das mensagens [56].

Em [57], Know *et al.* modificam o algoritmo distribuído SCFQ (*Self-Clocked Fair Queueing*) [58] e o combinam com o esquema de priorização proposto pelo mecanismo EDCA, para então propor o RT-FCR (*RT Fast Collision Resolution*). O algoritmo SCFQ atribui a cada mensagem uma *tag* inicial (definida pelo instante de tempo no qual a mensagem chegou na fila da transmissão) e uma *tag* final (definida como o valor da *tag* inicial mais o tempo necessário para a transmissão da mensagem). Neste contexto, as prioridades das mensagens são implementadas a partir da definição de diferentes intervalos de *backoffs* baseados no tipo de tráfego e no valor da *tag* final.

Em [59], Vaidya *et al.* propuseram um algoritmo distribuído denominado DFS (*Distributed Fair Scheduling*) o qual tem como objetivo principal realizar um escalonamento justo (*fairness*) em redes sem fio. O DFS comporta-se de forma semelhante ao mecanismo DCF, exceto no que diz respeito ao intervalo de *backoff*, o qual é calculado no início da operação com base nas *tags* finais atribuídas a cada mensagem por um algoritmo SCFQ modificado.

A principal limitação das oito propostas anteriores é que embora consigam gerir os parâmetros de forma a reduzir o número de colisões, nenhuma delas é capaz de garantir a prioridade no acesso ao meio das mensagens de tempo-real num ambiente de comunicação aberto. Isto deve-se principalmente pela variação nos valores de AIFS e *backoff* implementados pelas propostas.

Hamidian e Körner [60] apresentam uma solução capaz de fornecer garantias de QoS para o mecanismo EDCA a operar em modo *Ad Hoc*. Neste caso, as mensagens de alta prioridade (voz e vídeo) podem reservar tempo para efetuarem o acesso ao meio livre de contenção, e as demais prioridades (*best-effort* e *background*) efetuam suas transmissões com base no mecanismo EDCA tradicional. A ideia é transferir o algoritmo de escalonamento e o mecanismo de controlo de admissão do controlador HCCA (HC) para as estações.

A configuração é realizada utilizando como base a disseminação de mensagens de requisição ADDTS (*Add Traffic Stream*) pertencente ao elemento TSPEC (*Traffic Specification*). Neste contexto, as requisições são inicialmente submetidas localmente ao escalonador da estação, sendo então enviadas à todos os vizinhos (em *broadcast*). Ao receber esta requisição, as estações vizinhas verificam se o seu escalonador local pode cumprir os requisitos solicitados. Esta verificação é realizada comparando os requisitos solicitados (recebidos na mensagem) com os já admitidos pelas estações (local). Posteriormente é enviada uma resposta (em *unicast*) à estação solicitante. Caso todas as respostas recebidas pela estação solicitante sejam positivas, a requisição é aceite localmente. Ademais, para ajudar na criação de um período livre de contenção, que representa a TXOP das mensagens de alta prioridade, são utilizadas as mensagens RTS/CTS.

A principal limitação desta proposta encontra-se na necessidade de uma sincronização entre os relógios de todas as estações que trocam as mensagens de requisição e resposta, uma vez que estas precisam saber o momento inicial e final dos tráfegos admitidos para que não iniciem suas próprias transmissões. Além disso, estações vizinhas que não implementam a proposta não irão responder às requisições.

3.1.3 Síntese dos Mecanismos de Controlo de Acesso ao Meio

Embora a utilização de dispositivos de comunicação sem fio utilizando o IEEE 802.11 seja perfeitamente funcional em ambientes empresariais e domésticos, é necessário ter em conta as suas limitações quando estes são inseridos em ambientes industriais com o objetivo de fornecer serviços de comunicação de tempo-real. A principal limitação encontrada, no que diz respeito ao mecanismo de controlo de acesso ao meio, é a utilização de algoritmos probabilísticos para a resolução de contenções. Isto faz com que o tempo de transmissão das mensagens tenha um comportamento pouco previsível, característica esta considerada indesejada em sistemas de comunicação de tempo-real.

Deste forma, com base no *primeiro nível* de classificação previamente definido, podemos concluir que do ponto de vista organizacional são preferíveis propostas que utilizem como base uma arquitetura centralizada para a sua implementação. Esta abordagem possibilita ao AP ter uma visão global do estado atual do meio de comunicação onde a rede está a operar. Como consequência, a implementação de um mecanismo de controlo de admissão no AP possibilita uma visão mais precisa dos recursos disponíveis e alocados no meio de comunicação. Isto é uma consequência da unificação da área de cobertura da rede, uma vez que neste caso a premissa básica para a comunicação entre duas estações é que ambas estejam dentro da área de cobertura do AP. Por fim, a utilização de uma arquitetura centralizada possibilita também a simplificação, tanto do mecanismo de controlo de admissão, quanto do mecanismo de sincronização dos relógios das estações, que geralmente necessitam de soluções mais complexas em ambientes distribuídos.

No que diz respeito ao *segundo nível* de classificação, podemos concluir que as propostas mais adequadas são aquelas que tentam *resolver* ou *evitar* colisões, uma vez que ambas visam criar limites temporais para a sua resolução. As soluções que tentam *reduzir* o número de colisões não se apresentam adequadas para a transmissão de tráfego de tempo-real por serem incapazes de

priorizar o acesso ao meio. Além disso, é importante referir que, embora as soluções que tentam *evitar* colisões apresentem características adequadas à transmissão de tráfego de tempo-real, o respectivo mecanismo de controlo de acesso ao meio (geralmente implementado através de um esquema de *polling* ou Mestre/Escravo) pode sofrer um *overhead* significativo em resultado do processo de transmissão das autorizações às estações.

Por fim, no contexto do *terceiro nível* de classificação e ao assumirmos que o meio físico utilizado pelas tecnologias de transmissão sem fios é essencialmente um ambiente de comunicação aberto, podemos concluir que a compatibilidade das propostas com dispositivos IEEE 802.11 standard a operar na mesma área de cobertura geográfica e no mesmo canal de comunicação é um factor fundamental. Além disso, com a contínua disseminação de dispositivos IEEE 802.11, torna-se cada vez mais difícil a criação de zonas livres deste tipo de interferência. Outro factor fundamental é a compatibilidade das propostas com a sua implementação através de *hardware* COTS. Isto faz com que a proposta seja mais facilmente implementada (unicamente através de modificações de *driver* e/ou *firmware*) sem que para isto exista a necessidade de recorrer a *hardware* específicos. Do ponto de vista prático, isto simplifica o processo de implementação (somente são efetuadas atualizações de *software*) e permite reduzir os custos (não há a necessidade de compra de novos dispositivos de comunicação).

Desta forma, e com base na classificação previamente definida, são listados seguidamente os requisitos que se julgam necessários para a implementação de um mecanismo de controlo de acesso ao meio de um sistema de comunicação de tempo-real:

- preferencialmente baseado numa arquitetura centralizada;
- deve *resolver* ou *evitar* as colisões, uma vez que ambas as abordagens visam criar limites temporais para a sua resolução;
- deve ser capaz de garantir requisitos *soft real-time* mesmo quando estiver a operar num ambiente de comunicação aberto, onde o meio de comunicação é partilhado com estações IEEE 802.11 standard a operar fora da esfera de controlo da arquitetura de tempo-real;
- reduzir o *overhead* gerado pelos mecanismos de *polling* tradicionais;
- preferencialmente a sua implementação deve ser compatível com *hardware* COTS.

A Figura 3.1 resume e classifica as diferentes soluções de controlo de acesso ao meio apresentadas neste capítulo. Ao analisá-las podemos observar que, das 30 soluções apresentadas apenas 10 cumprem os requisitos considerados necessários (compatibilidade IEEE 802.11 e *resolver* ou *evitar* as colisões) e, dentre estas 10 soluções apenas 5 são compatíveis com *hardware* COTS.

Das 5 soluções encontradas, 2 são definidas pela norma IEEE 802.11, ou seja, o PCF e o HCCA. Contudo, o mecanismo de *polling* implementado por ambas as soluções acaba por gerar um *overhead* significativo nas transmissões, seja pela necessidade do envio de mensagens de autorização, seja pela obrigatoriedade de enviar tais autorizações a todas as estações da lista de *polling*, mesmo as que não tenham mensagens para transmitir.

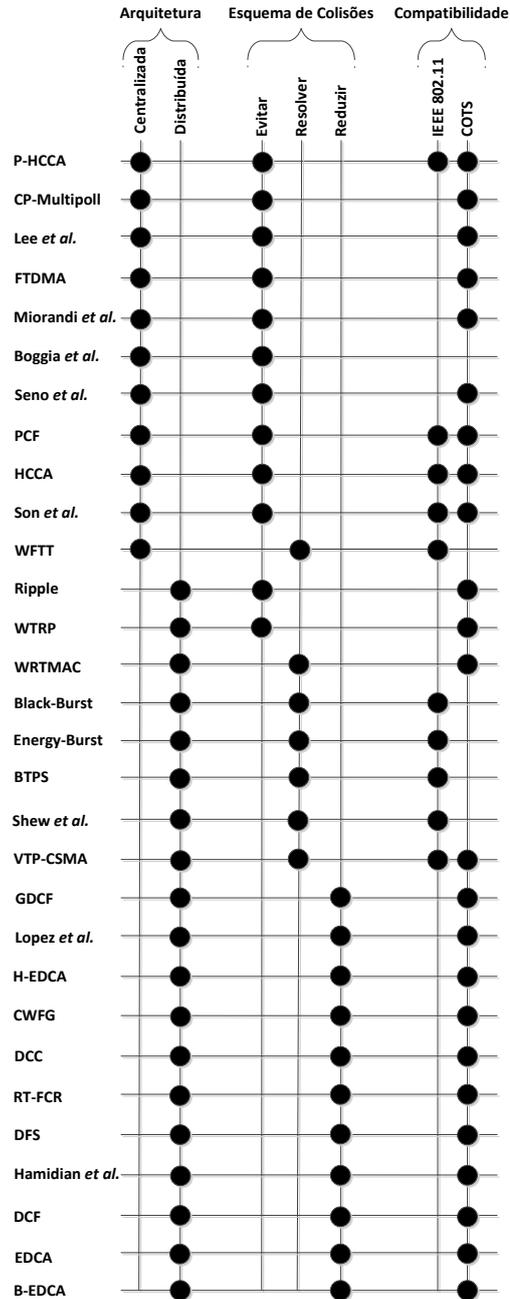


Figura 3.1: Comparação entre as propostas de controlo de acesso ao meio apresentadas.

Dois outras soluções baseiam-se no HCCA. A primeira (Son *et al.* [24]) visa diminuir o problema do *overhead* do mecanismo de *polling* punindo estações que recebam uma autorização mas não tenham mensagens para transmitir. No entanto, o método utilizado pode punir erradamente estações que tenham um fluxo de dados com um intervalo de serviço superior ao intervalo de serviço definido pelo HC, ou então estações que gerem tráfego aperiódico e/ou esporádico. A segunda solução (Hantrakoon e Phonphoem [30]) modifica o mecanismo de controlo de acesso ao meio

do HCCA para suportar três classes de prioridade diferentes, não efetuando qualquer melhoria relativa ao problema de *overhead*.

Por fim, a última solução encontrada é apresentada por Moraes *et al.* [49]. Consiste num mecanismo baseado num esquema *Token-Passing* onde a passagem do *token* é implementada de forma virtual, ou seja, localmente em cada estação. Embora isto solucione o problema de perda do *token*, a maneira como este processo é implementado pode sofrer interferências quando as estações estão a operar num ambiente de comunicação aberto. Isto porque como a passagem do *token* baseia-se numa medição dos tempos de transmissão efetuados na rede, caso alguma estação não escute (por algum motivo) estas transmissões, esta pode interpretar erradamente a passagem do *token*, resultando assim em situações onde duas ou mais estações estejam de posse do *token* ao mesmo tempo.

Torna-se assim evidente a necessidade de avançar com uma nova proposta para o mecanismo de controlo de acesso ao meio. Esta, por sua vez, deve ser capaz de favorecer o tráfego de tempo-real tanto no acesso ao meio quanto na resolução de possíveis colisões. Deve também ser capaz de garantir os requisitos temporais das mensagens mesmo na presença de outras estações IEEE 802.11 standard a operar no mesmo canal de comunicação. Por fim, a nova proposta deve permitir suportar a implementação de um mecanismo de controlo de admissão e também ser capaz de ser implementada em *hardware* COTS.

3.2 Mecanismos de Controlo de Admissão

Além de um mecanismo de controlo de acesso ao meio capaz de fornecer uma maior prioridade ao tráfego de tempo-real, tanto no acesso ao meio quanto na resolução de colisões, um sistema de comunicação de tempo-real deve também implementar um mecanismo de controlo de admissão. O principal objetivo deste mecanismo é limitar a quantidade de tráfego admitida por uma classe de serviço específica de forma que a QoS dos fluxos de dados já existentes não sejam degradadas, além de maximizar a utilização do meio de comunicação [19]. Neste contexto, são apresentadas nesta secção as principais propostas relacionadas ao mecanismo de controlo de admissão encontradas na literatura.

Uma classificação geralmente utilizada para estes mecanismos consiste em dividi-los em três diferentes abordagens: *Baseada em Modelos*, *Baseada em Medidas* e *Híbrida* [61]. As propostas que utilizam uma abordagem **baseada em modelos** utilizam tipicamente modelos analíticos como critério de decisão para a admissão (ou não) de um fluxo de dados específico. A principal vantagem é a possibilidade de otimizar o sistema como um todo, mesmo antes da sua inicialização. Entretanto, a principal desvantagem é a dificuldade de se redefinir (sempre que necessário) os parâmetros do sistema de forma a acompanhar o comportamento dinâmico da rede.

As propostas que utilizam uma abordagem **baseada em medidas** utilizam como critério as métricas obtidas do meio de comunicação e das estações comunicantes. Esta característica possibilita que os parâmetros do sistema sejam reajustados de acordo com o comportamento dinâmico da rede. No entanto, a sua principal desvantagem está na dificuldade em definir-se um limite

(*threshold*) apropriado para realizar a comparação com as métricas observadas. Além disto, dificilmente se obtém uma configuração "ótima" no momento da inicialização do sistema, uma vez que é necessário um tempo mínimo de operação para se obterem as métricas utilizadas.

Com o objetivo de evitar as limitações previamente apresentadas, algumas propostas utilizam uma abordagem **híbrida**, que realiza uma fusão das características das abordagens **baseada em modelos** e **baseada em medidas**. Neste caso, o sistema pode ser inicializado com um grupo de parâmetros "ótimo", o qual pode ser modificado dinamicamente de acordo com o comportamento da rede e/ou das estações.

3.2.1 Abordagens Baseadas em Modelos

O mecanismo HCCA possui uma coordenação centralizada que permite o acesso ao meio livre de contenção. Baseia-se num esquema de *polling* que aloca TXOP aos fluxos de dados previamente admitidos. O seu mecanismo de controlo de admissão (apresentado no capítulo 2) implementa uma ACU (*Admission Control Unit*) responsável por decidir entre aceitar (ou não) um fluxos de dados. Esta ACU posteriormente define os parâmetros que permitem ao escalonador (baseado num esquema *Round-Robin*) a gestão do mecanismo de *polling*. Tanto a ACU quanto o escalonador encontram-se numa entidade lógica denominada HC, e residente normalmente no AP.

Com o objetivo de disponibilizar um mecanismo de controlo de admissão baseado em prioridades ao HCCA, Hantrakoon e Phonphoem [30] propuseram o PHCCA (*Priority based HCCA*). Este mecanismo controla as filas de transmissão dividindo-as em 3 classes diferentes (derivadas da norma IEEE 802.3D) que podem ser organizadas pelo tipo ou pela relevância do utilizador. Para possibilitar novas admissões em classes de alta prioridade quando estas se encontrem esgotadas, o controlo de admissão implementa um algoritmo de partilha de banda (*bandwidth borrowing algorithm*). Este algoritmo possibilita que classes de maior prioridade possam solicitar banda "emprestada" de classes de menor prioridade para assim transmitir as suas mensagens. Para evitar problemas de *starvation* nas classes de menor prioridade, cada classe divide sua largura de banda em duas partes: uma passível de ser emprestada a uma classe de maior prioridade, e outra exclusiva da classe em questão. O controlo de admissão opera evitando que a soma da banda usada pelas 3 diferentes classes exceda a largura de banda máxima disponível.

Em [62], Cicconetti *et al.* apresentam o RTH (*Real-Time HCCA*) que tem como objetivo assegurar uma capacidade fixa para as TS durante um intervalo de tempo fixo. Este mecanismo baseia-se no algoritmo EDF (*Earliest Deadline First*) com SRP (*Stack Resource Policy*) e considera a transmissão de uma mensagem como uma seção crítica, uma vez que não pode interromper a sua transmissão para transmitir outra mensagem (*non-preemptive*). O escalonador subdivide-se entre atividades *online* e *offline*. A atividade *online* consiste em ler a próxima entrada $[i, t_i, TXOP_i]$ escalonada. Esta é composta pelo índice da próxima TS, o tempo de *polling* e a duração da transmissão, respectivamente. A atividade *offline* é utilizada para efetuar o controlo de admissão e gerir os recursos disponíveis no sistema. Para solicitar a admissão de uma TS, a estação envia ao HC os seguintes parâmetros: taxa média de dados (R_i), tamanho da MSDU (*MAC Service Data Unit*)(N_i), atraso máximo (D_i) e taxa de transmissão mínima da PHY (Γ_i).

Desta forma, cada TS_i é caracterizada por dois parâmetros: período (T_i) e capacidade (C_i). O período é dado por:

$$T_i = \begin{cases} D_i & \text{se } D_i < N_i/R_i \\ \left\lfloor \frac{R_i}{N_i} \times D_i \right\rfloor \times \frac{N_i}{R_i} & \text{caso contrário} \end{cases} \quad (3.1)$$

Neste caso, o período T_i é definido como o atraso máximo da mensagem (D_i), caso este seja inferior ao tempo médio entre as chegadas das mensagens na camada MAC da estação (N_i/R_i). Caso contrário, o período T_i é definido como o maior múltiplo do intervalo entre chegadas que não seja superior a D_i , respeitando assim a condição $T_i \leq D_i$.

A capacidade é definida como o tempo mínimo necessário para que a TS_i complete a sua transmissão cumprindo seus requisitos, sendo dado pela seguinte equação:

$$C_i = \left\lceil \frac{R_i \times T_i}{N_i} \right\rceil \times t_{N_i} \quad (3.2)$$

onde t_{N_i} é o tempo necessário para a transmissão de uma MSDU, incluindo o tempo necessário para a recepção da mensagem de ACK.

Assim, para admitir uma nova TS_k , inicialmente é realizado um teste de escalonabilidade levando em consideração o tempo de bloqueio que cada TS_i pode gerar. Primeiro a ACU calcula o tamanho da seção crítica (b_i) de cada TS_i . Este valor é dado pela seguinte equação:

$$b_i = t_{N_i} + t_{P_i} \quad (3.3)$$

onde t_{P_i} é o tempo necessário para efetuar o *polling* da TS_i (incluindo o IFS). Após isto é possível definir o bloqueio máximo (B_k) que uma TS_k pode sofrer. Este valor é dado pela maior seção crítica encontrada entre todas as TS_i já admitidas pelo sistema e que tenham um período maior que TS_k :

$$B_k = \max_{j>i} \{b_j\} \quad (3.4)$$

A análise de escalonabilidade produz a seguinte condição suficiente para determinar o grupo de n TS escalonáveis:

$$\frac{B_i}{T_i} + \sum_{j \leq i} \frac{C_j + \pi_j \times t_{P_j}}{T_j} \leq 1 \quad \forall i: 1 \leq i \leq n \quad (3.5)$$

onde π_j é o número máximo de vezes que o AP pode efetuar *polling* em TS_j durante o período T_j .

Em [63], Cecchetti *et al* propuseram uma alteração e adaptação para ambientes de comunicação sem fio do escalonador CBS (*Constant Bandwidth Server*) proposto por Albeni e Butazzo em [64], tendo como resultado o W-CBS (*Wireless-CBS*). Este algoritmo define de forma independente capacidades (*budgets*) para cada TS_i (Q_i). Assim, além de respeitar um escalonamento realizado por um algoritmo EDF, antes de permitir a transmissão por uma TS, a ACU verifica se esta contém um *budget* suficiente para realizá-la.

O parâmetro Q_i define a utilização máxima que a TS_i pode ter dentro do seu período P_i de geração de mensagens. Ambos, Q_i e P_i , são enviados à ACU via TSPEC. Assim, o teste de escalabilidade do W-CBS utiliza como base o período e o *budget* definido para cada TS, e para que seja aceite deve respeitar a seguinte inequação:

$$\sum_{i=0}^N \frac{Q_i}{P_i} \leq \frac{T - T_{CP}}{T} \quad (3.6)$$

onde P_i é o intervalo máximo de serviço (*Maximum Service Interval* – MSI).

O valor do *budget* de uma TS_i (Q_i) é dado por:

$$Q_i = Q_{min} + CWF(Q_{max} - Q_{min}) \quad (3.7)$$

onde F é uma função de peso de Q_{min} e Q_{max} . O parâmetro Q_{min} representa o menor *budget* necessário para que a TS_i possa transmitir (durante o período T_i) uma MSDU utilizando a taxa média de dados. O parâmetro Q_{max} representa o maior *budget* necessário para que a TS_i possa transmitir (durante o período T_i) a maior MSDU gerada utilizando a taxa máxima de dados.

Portanto, no que concerne as limitações do mecanismo de controlo de admissão do HCCA, a sua abordagem pessimista (causada pela utilização da taxa mínima de transmissão e alocação das TXOP com base numa MSDU de tamanho máximo) resulta na subutilização do meio de comunicação. Outro problema é a sua incapacidade de gerir tráfegos VBR (*Variable Bit Rate*), uma vez que ao longo do tempo o tamanho e o número de mensagens podem diferir dos valores negociados no momento da admissão². Além disso, a falta de um mecanismo de diferenciação das categorias de serviço e o facto da ACU não levar em consideração o nível de ocupação do meio de comunicação por outros dispositivos que estejam fora da esfera de controlo do mecanismo HCCA podem também serem consideradas limitações importantes.

Com relação as limitações das restantes propostas apresentadas anteriormente, embora o RTH melhore a questão do escalonamento através da utilização de um algoritmo EDF/SRP, o PHCCA proporcione níveis de prioridade diferentes ao do HCCA e o W-CBS reduza a ineficiência do cálculo da TXOP, todas estas propostas continuam a ter a maioria das limitações impostas ao HCCA original, uma vez que baseiam-se neste para a sua implementação.

3.2.2 Abordagens Baseadas em Medidas

Em [65], Toscano e Lo Bello propuseram um mecanismo de controlo de admissão baseado no EDF e num factor de redundância (ρ) atribuído à cada TS. Este factor (alterado dinamicamente) define o número máximo de retransmissões que uma TS pode efetuar. O controlo de admissão opera sobre um mecanismo de controlo de acesso ao meio baseado num esquema *Master/Slave*. É reservada uma janela inicial para que o AP envie a mensagem de *beacon* e para que as estações

²Embora os algoritmos de codificação VBR clássicos tendam a gerar variações somente no tamanho das mensagens, alguns algoritmos permitem que este tamanho seja fixo, ou seja, podem ser geradas múltiplas mensagens com um tamanho limite pré-determinado.

realizem os seus pedidos de admissão. Em seguida, o AP envia para as estações mensagens de *polling* na forma de autorizações de transmissão.

No processo de submissão de uma TS_i , a estação precisa enviar à ACU o seu período de geração de tráfego (T_i) e o tamanho da maior mensagem que pode ser gerada ($WCSize_{Data_i}$). Baseado neste último parâmetro, a ACU calcula o WCET (*Worst Case Execution Time*) para esta mensagem de dados da seguinte forma:

$$WCET_{Data_i} = \frac{WCSize_{Data_i}}{Bandwidth} + latency \quad (3.8)$$

onde *Bandwidth* é a taxa de transmissão de dados e *latency* é uma estimativa do atraso no acesso ao meio. Após isto, é somado ao $WCET_{Data_i}$ o tempo necessário para o envio da mensagem de *polling* ($WCET_{poll}$) para definir-se o tempo total $WCET_i$.

Desta forma, o tempo alocado pela ACU à TS_i (C_i) é dado pelo valor de $WCET_i$ e por um factor de redundância (ρ_i):

$$C_i = \rho_i \times WCET_i \quad (3.9)$$

O valor ρ_i é alterado dinamicamente com base na estimativa do PER (*Packet Error Ratio*) sofrido por cada TS_i . Esta estimativa leva em consideração a média de mensagens perdidas dentro de uma janela de observação.

Além do PER, a ACU utiliza também como base para a definição do valor de ρ_i a taxa máxima de perda tolerável pela TS_i ($MaxPER_i$), a qual é informada à ACU pela estação no processo de admissão. Esta taxa define a fração mínima de mensagens que devem ser entregues corretamente antes das respectivas *deadlines*. Assim, a ACU altera o valor de ρ_i com base na seguinte regra:

$$\rho_i = \begin{cases} 1 & \text{se } k = 0 \\ \rho_i + 1 & \text{se } PER_i > MaxPER_i \\ \rho_i - 1 & \text{se } EarlyCompletion_i(m) \text{ for } TRUE \\ \rho_i & \text{caso contrário} \end{cases} \quad (3.10)$$

onde $k = 0$ é o número de mensagens transmitidas na janela de observação e $EarlyCompletion_i(m)$ representa m mensagens consecutivas transmitidas com sucesso e a utilizar um número de retransmissões inferior a ρ_i .

Portanto, para que seja possível a admissão de uma nova TS_k ou a alteração do factor ρ_i de alguma TS_i já admitida, os novos parâmetros devem respeitar a seguinte inequação:

$$\sum_{i=1}^k \frac{C_i}{T_i} + \frac{B_k}{T_k} \leq 1 \quad \forall k = 1, 2, \dots, N \quad (3.11)$$

onde B_k é o bloqueio máximo que a TS_i pode sofrer. Como o mecanismo utiliza o EDF não preemptivo, B_k é igual ao maior C_i admitido pela ACU.

Desta forma, uma limitação desta proposta é a abordagem pessimista no cálculo de tamanho da TXOP. Embora o seu superdimensionamento acabe por resultar no suporte ao tráfego VBR³ isto resulta na subutilização do meio de comunicação. Outro ponto importante é a definição do valor de bloqueio B_k , o qual é definido como o maior valor C_i dentre as TS previamente admitidas, ignorando assim o bloqueio que pode ser gerado pelo tráfego proveniente de estações que estejam fora da esfera de controlo do sistema de comunicação proposto.

Em [66], Bazzi *et al.* apresentam dois diferentes mecanismos de controlo de admissão para redes IEEE 802.11 infraestruturadas. O primeiro baseia-se no nível de ocupação do meio de comunicação (C_o) observado pelo AP. Este valor é dado em função do tempo de ocupação do meio (T_B) num intervalo de observação ΔT :

$$C_o = \frac{T_B}{\Delta T} \quad (3.12)$$

Para garantir que o valor de T_B seja computado corretamente, é somado ao seu valor o tempo de transmissão gasto pelo AP (T_{AP}), ou seja, $T_B = T_{AP} + T_{SB}$.

O principal objetivo é garantir um alto *throughput*. Assim, o modelo tenta manter um nível de ocupação do meio elevado sem exceder o ponto de saturação (C_T), o qual é previamente definido pelo sistema. Quando este valor é ultrapassado, não são permitidas novas admissões até que a ocupação do meio diminua e se encontre inferior à um nível pré-determinado chamado de "descongestionamento" (D_T).

O segundo mecanismo proposto utiliza a informação sobre o tamanho da fila de transmissão do AP, ou seja, considera somente o tráfego *downlink*. Nesta abordagem o aumento da fila representa um congestionamento da rede, sendo a sua diminuição um sinal de descongestionamento. Neste caso, o *threshold* é definido pelo nível de utilização do *buffer* de transmissão. Para evitar oscilações repentinas, o mecanismo faz uso de um factor de persistência (T_P) que indica quanto tempo os níveis devem se manter acima ou abaixo dos *thresholds* pré-estabelecidos antes de modificar o estado da fila. Neste caso, são permitidas novas admissões apenas quando o mecanismo se encontra no estado de descongestionamento.

Em [67] Yong *et al.* propuseram um mecanismo de controlo de admissão para redes IEEE 802.11 infraestruturadas baseado no nível de ocupação da rede e no atraso máximo tolerável pela TS. Este atraso é definido pela estação no momento da admissão através do envio de mensagens TSPEC. No AP, um mecanismo monitoriza constantemente as diferentes i filas de receção descrevendo assim diferentes níveis de prioridade. Baseado nisto, o mecanismo define uma taxa de receção (λ_i) para cada fila. Da mesma forma, outro mecanismo monitoriza as filas de transmissão obtendo assim uma taxa de envio (μ_i) para cada uma. Como podem existir múltiplas prioridades, o problema é modelado como uma fila de prioridade preemptiva com uma única taxa de serviço.

³Uma vez que o mecanismo utiliza como base de cálculo para o controlo de admissão a maior mensagem que pode ser gerada pela aplicação, qualquer variação no tamanho das mensagens não afetará os valores calculados pela ACU.

O mecanismo de controlo de admissão admite apenas uma nova TS se esta cumprir dois requisitos. O primeiro é o atraso máximo tolerável pela TS na fila a qual as mensagens serão transmitidas ($Delay_i$). Assim, o valor estipulado de $Delay_i$ deve respeitar a seguinte inequação:

$$Delay_i < \frac{\sum_{i=1}^n \rho_i / \mu_i}{(1 - \sigma_{i-1}) \times (1 - \sigma_i)} \quad (3.13)$$

onde μ_i denota uma taxa de serviço markoviana e ρ_i a sua respectiva taxa de utilização para a fila i . Baseado nesta inequação o modelo proposto consegue estimar o tempo de espera das mensagens nas diferentes filas de transmissão.

Os valores de ρ_i e σ_n são dados pelas seguintes equações, respectivamente:

$$\rho_i = \frac{\lambda_i}{\mu_i} \quad (3.14)$$

$$\sigma_n = \sum_{i=0}^n \rho_n \quad (3.15)$$

O segundo requisito para a admissão de uma nova TS é a sua taxa de utilização ($Rate_i$), a qual deve respeitar a seguinte inequação:

$$Rate_i + CURRENT_{BW} < Admit_T \quad (3.16)$$

onde $CURRENT_{BW}$ é a estimativa da banda total utilizada e $Admit_T$ é um *threshold* definido pelo sistema para a fila i .

Em [68], Kim *et al.* propuseram um mecanismo de controlo de admissão implementado sobre um MAC capaz de criar prioridades de acesso por meio do uso de impulsos de energia (*busy-tone*). No mecanismo MAC proposto, filas com menor prioridade podem transmitir apenas quando não existirem mensagens para serem transmitidas nas filas de maior prioridade. Para que isto seja possível, as estações que desejam efetuar suas transmissões precisam aguardar que o meio fique livre por $aSIFSTime + (AIFSN \times aSlotTime) - aSlotTime$. Em seguida a estação envia um pulso de energia do tamanho de um $aSlotTime$ e logo após efetua a transmissão da mensagem utilizando o mecanismo EDCA.

Caso alguma estação que tenha mensagens na fila de transmissão receba um pulso de energia antes de finalizado o período de deteção do estado (livre/ocupado) do meio, esta suspende o envio do seu próprio pulso de energia. Assim, as filas com menor prioridade transmitem apenas quando não existirem mensagens para serem transmitidas nas filas de maior prioridade de qualquer outra estação. Os valores de AIFSN são diferentes para cada fila (variando de 0 a 3). Para diferenciar entre a transmissão de uma mensagem em um impulso de energia o mecanismo utiliza os seus respectivos tempos de duração, uma vez que a duração de um impulso de energia é inferior a 1 $aSlotTime$ e a duração de uma mensagem normalmente é superior a 3 $aSlotsTime$, dado o tamanho do cabeçalho da camada física.

O controlo de admissão deste mecanismo, denominado PAAC (*Priority Access-based Admission Control*), encontra-se no AP e utiliza informações TSPEC enviadas pelas estações que desejam submeter novas TS. Para realizar uma nova admissão, inicialmente o PAAC calcula o tempo de ocupação que a nova TS irá necessitar. Assim, calcula no final de cada intervalo de *beacon* (BI) os seguintes valores para cada AC_{*i*}: intervalo de transmissão (TI_{*i*}), tempo total de *backoff* (BP_{*i*}), número de transmissões (NT_{*i*}) e número de colisões (NC_{*i*}). Baseado nestes valores, o AP obtém a taxa de utilização do canal ($U_i^{measure}$), a probabilidade de colisão ($P_i^{measure}$) e o tempo médio de *backoff* ($B_i^{measure}$), que são dados, respectivamente, por:

$$U_i^{measure} = \frac{TI_i}{BI} \quad (3.17)$$

$$P_i^{measure} = \frac{NC_i}{NT_i} \quad (3.18)$$

$$B_i^{measure} = \frac{BP_i}{NT_i} \quad (3.19)$$

Posteriormente um mecanismo baseado em uma janela deslizante obtém os valores aproximados do tempo médio de utilização do meio (U_i), da probabilidade média de colisão (P_i) e do tempo médio de *backoff* (B_i) por tentativa de transmissão:

$$U_i = \alpha U_i + (1 - \alpha) \times U_i^{measure} \quad (3.20)$$

$$P_i = \alpha P_i + (1 - \alpha) \times P_i^{measure} \quad (3.21)$$

$$B_i = \alpha B_i + (1 - \alpha) \times B_i^{measure} \quad (3.22)$$

onde α é um factor de suavização que varia de 0 a 1. Valores próximos de 1 fazem com que as mudanças sejam mais lentas enquanto que valores próximos de 0 fazem com que as mudanças sejam mais rápidas.

Para solicitar a admissão numa AC *i*, uma TS_{*k*} envia uma mensagem ADDTS ao AP contendo os seguintes parâmetros: tamanho médio da mensagem ($L_{i,k}$), taxa média de dados ($\rho_{i,k}$) e a taxa de transmissão física ($R_{i,k}$). Ao receber a solicitação, o AP calcula o número de mensagens enviadas pela TS durante um intervalo de *beacon* ($N_{i,k}$) e o número total de mensagens transmitidas em caso de colisões ($N_{i,k,total} = N_{i,k}/(1-P_i)$).

Em seguida, o AP calcula o tempo necessário para se transmitir as $N_{i,k,total}$ mensagens ($ChTime_{i,k}$). Este valor é dado pela seguinte equação:

$$ChTime_{i,k} = N_{i,k} \times T_S + (N_{i,k,total} - N_{i,k}) \times T_C + N_{i,k,total} \times (B_i + AIFS[i]) \quad (3.23)$$

onde T_S é o tempo para se efetuar uma transmissão com sucesso e T_C é o tempo médio de ocupação do meio resultante de colisões. Assim, a taxa de utilização de uma nova TS é dada por:

$$U_{i,k,require} = \frac{ChTime_{i,k}}{BI} \quad (3.24)$$

Como a ACU do mecanismo PAAC divide o tempo total do meio (C) em 4 diferentes porções (voz, voz e vídeo, vídeo e outros tipos), cada AC_i possui uma porção exclusiva alocada para si. Portanto, é utilizada a seguinte inequação para decidir entre a admissão (ou não) de uma nova TS k :

$$U_i + U_{i,k} \leq CT_i \quad (3.25)$$

onde CT_i é a taxa de utilização livre da AC_i e que ainda pode ser alocada para novas TS.

Em [69], Shin e Schulzrinne especificam um controlo de admissão chamado QP-CAT (*Queue size Prediction using Computation of Additional Transmission*). Este mecanismo tem por objetivo prever o tamanho da fila de transmissão do fluxo *downlink* de uma TS, para que então possa ser possível prever o seu respectivo atraso e impedir (caso necessário) a admissão de novas TS evitando assim a saturação do meio de comunicação.

Inicialmente é feita a admissão da nova TS de forma virtual, levando em conta o período de geração das suas mensagens. Então, através da monitorização do meio e levando em conta a transmissão de mensagens reais, é calculado o número adicional de mensagens virtuais que poderiam ser enviadas. Por último, o número de mensagens adicionais que poderiam ser enviadas é subtraído do tamanho da fila virtual. Sendo assim, o tamanho previsto da fila é o número total de mensagens na fila atual e na fila virtual.

Para emular uma nova TS, são utilizados dois contadores especiais (*UpCounter* e *DnCounter*). Estes contadores armazenam o número de mensagens *uplink* e *downlink*, e ambos são incrementados em 1 a cada ciclo de geração da mensagem. Os respectivos valores são decrementados durante a execução baseado no número de mensagens n_p transmitidas. Desta forma pode-se afirmar que a soma da fila atual no AP com o valor de *DnCounter* será o tamanho previsto da fila de *downlink* no AP caso uma nova TS seja admitida.

Para admitir uma nova TS, o QP-CAT mede o tempo livre no meio (T_c). Caso seja utilizado o EDCA, então o valor de T_c é dado pelo tempo não utilizado de uma TXOP, ou seja, o tempo que poderia ser alocado para o envio de mensagens de outra TS. A partir disto é calculado o número de mensagens adicionais que são passíveis de serem enviadas (n_p) utilizando a seguinte equação:

$$n_p = \left\lfloor \frac{T_c}{T_t} \right\rfloor \quad (3.26)$$

onde T_t é o tempo necessário para a transmissão de uma mensagem (com sucesso) incluindo o tempo de *backoff* e os tempos IFS.

Em seguida, a ACU calcula o tempo restante da transmissão (T_r) das mensagens, uma vez que geralmente T_c e T_t não são múltiplos entre si. Assim, a seguinte equação é utilizada:

$$T_r = T_c - n_p \times T_t \quad (3.27)$$

Assim, caso $T_r > 0$, então a ACU verifica se $T_r = T_b + DIFS$, onde T_b é o tempo de *backoff*. Caso isto seja confirmado, irá ocorrer uma colisão entre a mensagem atual e a mensagem adicional. Neste caso, ambas as mensagens serão novamente retransmitidas e o tempo extra também deve ser levado em conta adicionando uma nova mensagem ao parâmetro *DnCounter*. Caso contrário, é calculado um novo valor de T_c ($T_c = T_c + T_r - DIFS$).

Assim, a ACU calcula novamente o número de mensagens adicionais (n_p) e então decrementa os valores de *DnCounter* e *UpCounter* utilizando $n_p/2$. Por fim a ACU calcula o tamanho da fila prevista (Q_p) utilizando o tamanho atual da fila no AP (Q_A) e o parâmetro *DnCounter* ($Q_p = Q_A + DnCounter$).

Em [70], Achary *et al.* propuseram o d-EDCA (*Dynamic EDCA*). Este mecanismo estima o nível de ocupação do meio de comunicação e a taxa média de colisões de uma AC específica e compara-os com níveis previamente definidos para admitir (ou não) novas TS. Para verificar estes valores, é utilizada uma janela de tempo de tamanho fixo (T). O tempo de ocupação do meio (T_{busy}) é a soma do tempo de transmissão de todas as transmissões realizadas (com ou sem sucesso) dentro da janela T :

$$T_{busy} = \sum_i t_i \quad (3.28)$$

onde t_i é o tempo de ocupação do meio da i -ésima transmissão. Baseado neste valor, a ocupação relativa do meio de comunicação é dada por:

$$BW_{occu} = \frac{T_{busy}}{T} \times 100 \quad (3.29)$$

Para definir o estado do meio de comunicação são utilizados dois níveis: um para o valor mínimo (BW_{lo}) e outro para o valor máximo (BW_{up}). Quando $BW_{occu} \geq BW_{up}$ o meio é considerado congestionado. Quando $BW_{occu} \leq BW_{lo}$ o meio é considerado subutilizado. O valor ótimo é dado por $BW_{lo} \leq BW_{occu} \leq BW_{up}$.

A taxa média de colisões (r_c) baseia-se no número de mensagens transmitidas (n_t) e no número de colisões (n_c) ocorridas em cada estação, cujo valor é dado por:

$$r_c = \frac{n_c}{n_t} \quad (3.30)$$

Da mesma forma que a taxa de ocupação, a taxa de colisão também utiliza dois níveis para definir seus limites mínimo (R_{lo}) e máximo (R_{up}). O meio é considerado congestionado quando $r_c \geq R_{up}$ e subutilizado quando $r_c \leq R_{lo}$. O valor ótimo é dado quando $R_{lo} \leq r_c \leq R_{up}$.

O controlo de admissão é baseado na ocupação do meio (BW_{occu}) e na taxa média de colisões (r_c). Neste contexto, caso o meio seja considerado congestionado, a estação suspende a

transmissão de mensagens com baixa prioridade, as quais são retomadas somente após o meio ser considerado subutilizado.

Em [71], Hiraguri *et al.* propuseram um mecanismo de controlo de admissão para o EDCA baseado no nível de ocupação do meio de comunicação e nos requisitos das TS. Além disto, também é proposto um esquema de controlo de tráfego com o objetivo de garantir a QoS das TS após a admissão. Neste contexto, ao receber um novo pedido de admissão, a ACU calcula o tempo necessário para a transmissão da nova TS levando em consideração o valor definido pelo parâmetro *Surplus Time*⁴. Baseado neste parâmetro, após receber uma requisição, a ACU calcula o tempo de ocupação no meio inicial ($T_{MT_{init}}$) gerado pela TS:

$$T_{MT_{init}} = SurplusTimeAllowance \times \left\lceil \frac{\rho}{L} \right\rceil \times T_{MPDU} \quad (3.31)$$

onde ρ é a taxa média de geração de dados, L é tamanho da MPDU e T_{MPDU} é o tempo necessário para transmitir uma MPDU (incluindo IFS e ACK).

Em seguida, a ACU calcula o montante total de ocupação no meio (T_{used}) das n TS de voz (VO) e das m TS de vídeo (VI) previamente admitidas pelo sistema:

$$T_{used} = \sum_{i=1}^n T_{MT_{VO}} + \sum_{j=1}^m T_{MT_{VI}} \quad (3.32)$$

Por fim, para que uma nova TS seja admitida, a soma dos tempos de ocupação da nova TS e das TS previamente admitidas pelo sistema deve ser menor que o tempo total disponível menos a constante d , utilizada para controlar a flutuação do tráfego. Sendo assim, uma nova TS será admitida se respeitar a seguinte inequação:

$$1 - d \geq T_{MT_{init}} + T_{used} \quad (3.33)$$

As TS são controladas de duas formas após a admissão. Na primeira, a ACU monitoriza constantemente as TS de alta prioridade (voz e vídeo). Caso o tempo de ocupação exceda o permitido, a ACU exclui repetidamente algumas TS até que o nível de ocupação seja normalizado. Da mesma forma a ACU controla o tráfego das TS de baixa prioridade (*best-effort* e *background*). No entanto, como este tipo de tráfego não submete requisições à ACU, esta simplesmente descarta as mensagens recebidas destas filas para garantir uma maior disponibilidade de recursos aos tráfegos de alta prioridade.

Em [72] Sun *et al.* é proposto um mecanismo de controlo de admissão centralizado para tráfegos multimédia, utilizando como critério de decisão diferentes níveis de QoE (*Quality-of-Experience*) para cada TS admitida. O objetivo é restringir o número de tráfegos *best-effort* afim de garantir uma maior disponibilidade de recursos para o tráfego de tempo-real.

Os níveis de QoE são obtidos através de um módulo denominado PSQA (*Pseudo-Subject Quality Assessment*). Este módulo obtém os valores de MOS (*Mean Opinion Score*) de cada

⁴O parâmetro *Surplus Time* define o nível de ocupação extra que uma TS necessitará além do valor calculado pelo seu período de geração e tamanho das mensagens. Este parâmetro considera tanto o atraso no acesso ao meio quanto as retransmissões.

TS os quais são mapeados entre diferentes níveis de QoE. Cada valor obtido é função direta do número de mensagens perdidas pela TS. O módulo PSQA grava o número de mensagens perdidas (inclusive as que perderam seus respectivos *deadlines*) dentro de uma janela deslizante de tamanho pré-determinado a qual se desloca para frente a cada T_m segundos.

Uma vez que os valores de MOS variam de 0 a 5, o modelo definiu que o valor mínimo para as TS de tempo-real deve ser igual a 3. Assim, sempre que alguma TS tiver $MOS < 3$ o controlo de admissão irá suspender uma TS *best-effort* até que o valor mínimo seja recuperado. Como a janela deslizante move-se para frente a cada T_m segundos, este será o intervalo de tempo em que o sistema irá aguardar até suspender uma nova TS *best-effort*. No entanto, como as alterações podem demorar mais tempo para serem efetivas, com o objetivo de evitar que demasiadas TS sejam suspensas, o mecanismo implementa um intervalo de tolerância (*TI*). Assim, uma TS *best-effort* somente é suspensa se a MOS do tráfego tempo-real se mantiver abaixo de 3 por um período de tempo maior que *TI*.

No que concerne às limitações dos mecanismos de controlo de admissão apresentados anteriormente, com exceção do mecanismo apresentado por Toscano e Lo Bello [65], todos os restantes tem como principal limitação a falta de um algoritmo de escalonamento para organizar as comunicações na rede de tempo-real. Isto impede que a ACU gira de forma adequada e independente as *deadlines* atribuídas a cada TS que necessita efetuar uma transmissão. Assim, podem ocorrer inversões de prioridade (quando levado em conta o *deadline* e/ou período de geração da mensagem) numa TS de uma mesma categoria de tráfego mas com requisitos de QoS diferentes.

Um outra limitação encontrada nos mecanismos apresentados por Bazzi *et al.* [66] e Yong *et al.* [67] é a dificuldade em definir um *threshold* adequado, uma vez que o ponto de saturação da rede pode variar de acordo com os parâmetros QoS requeridos por cada TS. Neste contexto, ambos os mecanismos tentam garantir um alto *throughput* nas comunicações, no entanto isto não significa a garantia de cumprimento dos requisitos temporais das mensagens.

Além disso, o cálculo de ocupação do meio de comunicação efetuado pelos mecanismos apresentados por Toscano e Lo Bello [65], Hiraguri *et al.* [71] e pelo mecanismo PAAC [68] não levam em conta o tráfego gerado por estações que estão fora da esfera de controlo da arquitetura de tempo-real. Isto pode levar a decisões erradas na admissão de novas TS, podendo em algumas situações resultar na saturação do meio de comunicação e conseqüentemente na degradação da QoS de todas as TS previamente admitidas.

Por fim, uma limitação comum ao mecanismo d-EDCA [70] e ao mecanismo apresentado por Sun *et al.* [72] é a tentativa de garantir a QoS das TS através da suspensão de tráfegos de baixa prioridade. Esta abordagem encontra problemas na sua aplicação quando os tráfegos de baixa prioridade são gerados por redes que estão fora da esfera de controlo da arquitetura de tempo-real.

3.2.3 Abordagens Híbridas

Em [73], Didi *et al.* apresentam o PRCW (*Physical Rate and Contention Window*), um mecanismo de controlo de admissão centralizado baseado na taxa de transmissão física e na janela de contenção de cada estação. A ideia principal é alterar os valores de AIFSN, CW_{min} e CW_{max}

do tráfego *best-effort* para diminuir o número de colisões e assim tentar aumentar o *throughput* total da rede. Diferentemente do HCCA (que utiliza a taxa de transmissão mínima), este modelo define para cada estação i uma taxa de transmissão (R_i) de acordo com a sua distância do AP. Neste contexto, os autores assumem a seguinte configuração: **i)** $R_i = 36$ Mbps para distância ≤ 5 metros; **ii)** $R_i = 11$ Mbps para $5 \text{ metros} < \text{distância} < 10$ metros; e **iii)** $R_i = 6$ Mbps para as demais distâncias.

A admissão de uma nova TS inicia-se com o envio de uma requisição da estação ao AP. Esta requisição contém a descrição e os requisitos temporais da TS em questão. Ao recebê-la, o AP calcula a distância para definir o valor de R_i . Este valor é utilizado de base para o cálculo da $TXOP_i$, o qual é realizado utilizando os mesmos procedimentos definidos pelo cálculo do mecanismo HCCA. Em seguida, o AP calcula a carga total da rede. Caso esta seja $\leq 70\%$ da carga máxima, então o AP realiza o mesmo procedimento de escalonamento utilizado pelo mecanismo HCCA. Se os parâmetros submetidos ao escalonador passarem no teste de escalonabilidade então a TS será admitida, caso contrário será recusada.

Por outro lado, caso a carga da rede seja $> 70\%$ da carga máxima, o AP calcula o número de colisões (C) e o número de TS *best-effort* previamente admitidas (DA). Se o valor de C for maior que um nível previamente definido, o sistema aumenta o CW_{min} das TS *best-effort* definindo $CW_{min} = CW_{min} \times 2$. Este aumento é recursivamente realizado enquanto $DA > \frac{\text{total de fluxos}}{3}$ ou $C > \text{threshold}$. Caso contrário, o sistema retorna ao CW_{min} inicial. O objetivo deste aumento é manter a QoS das TS de áudio e vídeo já admitidas sem bloquear as TS *best-effort*.

Em situações onde a TS admitida possui um tráfego VBR, o número de mensagens transmitidas pela estação pode ser diferente do número calculado pelo AP, podendo assim resultar em mensagens descartadas. Neste caso, se o número de mensagens descartadas for maior um nível previamente definido, o AP irá recalculer a $TXOP$ alocada à TS. Nas estações, um mecanismo avalia constantemente o tamanho das filas das TS admitidas pelo sistema. Caso alguma fila seja menor que o tamanho informado ao AP, significa que a $TXOP$ alocada a TS é maior que o necessário para a transmissão das respectivas mensagens. Neste caso, a estação redistribui o tempo excedente entre as restantes TS existentes na estação.

A principal limitação do PRCW encontra-se na necessidade de efetuar alterações ao tráfego *best-effort* para controlar a admissão de novas TS e garantir um melhor QoS para os tráfegos de áudio e vídeo. Em ambientes onde o tráfego *best-effort* é proveniente de uma rede fora da esfera de controlo do sistema de tempo-real, não é possível realizar nenhuma modificação, impossibilitando assim o funcionamento do mecanismo.

Em [74] Cruz *et al.* definem um mecanismo de controlo de admissão que considera as variações ocorridas na taxa de transmissão física das estações. Desta forma, espera-se alcançar a utilização máxima dos recursos disponíveis diminuindo o nível de perdas das mensagens. Neste contexto, além de quantificar os recursos consumidos pelas TS admitidas, a ACU utiliza também um histórico para verificar as mudanças sofridas nas taxas de transmissão das TS admitidas.

Neste contexto, o AP armazena as seguintes informações das transmissões efetuadas pelas n TS admitidas: **i)** o tempo utilizado para a tentativa de transmissão, **ii)** a taxa de transmissão,

iii) a direção (*uplink* ou *downlink*) e **iv)** o estado final da transmissão (sucesso ou falha). Após um período de avaliação (T_m) pré-definido, estes valores são processados de forma a obter os seguintes parâmetros: **i)** grupo de taxas de transmissão (\bar{R}) utilizadas pelas transmissões efetuadas com sucesso dentro do período T_m , **ii)** tempo total consumido por transmissões efetuadas com sucesso (S_{voice}) e, **iii)** tempo total consumido por transmissões que falharam (T_{loss}). Com base nestes parâmetros, o sistema calcula o tempo total de ocupação do meio (S_{air}) durante o período de avaliação como:

$$S_{air}(n, \bar{R}) = S_{voice}(n, \bar{R}) + T_{loss}(n, \bar{R}) \quad (3.34)$$

Como o valor de T_{loss} é variável, a proposta optou por utilizar o maior valor obtido (T_{loss}^*) dentro da janela de avaliação T_m .

Em seguida a ACU compara os valores das taxas de transmissão atuais (R^{now}) de cada TS contida no grupo \bar{R} com seus respectivos valores médios armazenados no histórico, e escolhe a opção mais conservadora de ambas (R^*) e calcula um valor de ajuste das taxas de transmissão tanto do fluxo *uplink* quanto do fluxo *downlink*, dados pela seguinte equação:

$$A_j = \sum_{i=1}^n T_v(R_{i,j}^*) - T_v(R_{i,j}^{now}) + T_{ack}(R_{i,j}^*) - T_{ack}(R_{i,j}^{now}) \quad (3.35)$$

onde $T_v(R_{i,j})$ é o tempo de transmissão de uma mensagem da TS i na direção j utilizando a taxa de transmissão R . O valor total do ajuste é dado por $A = A_{downlink} + A_{uplink}$. Sendo assim, o tempo total de ocupação S_{air} pode ser alterado para:

$$S_{air}(n, \bar{R}) = S_{voice}(n, \bar{R}) + A + T_{loss}^*(n, \bar{R}) \quad (3.36)$$

Ao final, a ACU avalia o impacto que a nova TS pode causar no sistema se esta for aceita. Neste contexto, a ACU utiliza o tamanho da MSDU e a taxa mínima de transmissão enviados pela TS via TSPEC para calcular o valor extra de S_{voice} se a TS for admitida. É necessário também estimar o valor de perda para a nova TS ($T_{loss}^*(n+1)$). Este valor é derivado dos modelos matemáticos que prevêem o comportamento de fluxos bidirecionais em ambientes não saturados de redes IEEE 802.11 [75, 76]:

$$T_{loss}^*(n+1) = T_{loss}^{Measured}(n) + (T_{col}^{Model}(n+1) - T_{col}^{Model}(n)) \quad (3.37)$$

Logo, a nova TS será admitida se a seguinte inequação for respeitada:

$$T_{available}(n+1) = T_m - (S_{voice}(n+1) + A + T_{loss}^*(n+1)) > 0 \quad (3.38)$$

A principal limitação desta proposta encontra-se na utilização dos modelos matemáticos para calcular as probabilidades de colisão. Estes baseiam-se no número conhecido de estações comunicantes, situação plausível em uma BSS fechada, mas que se torna pouco realista num ambiente de comunicação aberto. Assim, o modelo probabilístico do número de colisões pode considerar

menos fluxos de transmissão do que os realmente existem, e por consequência gerar valores que não refletem a realidade.

Em [77] Andreadis e Zambon propuseram o DTXOP (*Dynamic TXOP*), um algoritmo que tem como objetivo prover *fairness* entre os fluxos *uplink* e *downlink* operados pelo mecanismo EDCA. Este modelo considera dois cenários distintos: balanceado e não-balanceado.

No cenário balanceado, o tráfego *uplink* é igual ao tráfego *downlink*, ou seja, somente existem fluxos bi-direcionais (ex.: VoIP e P2P). Neste caso o AP pode solicitar a metade do tempo alocado para uma TS, caso contrário, ele somente pode utilizar o meio por $1/(n+1)$, onde n é o número de TS admitidas. No cenário não-balanceado, o tempo alocado aos fluxos *downlink* é superior ao tempo alocado aos fluxos *uplink*. Isto reflete a situação mais comum, uma vez que na maioria dos casos o fluxo *uplink* está relacionado com mensagens de solicitação (menores) e o fluxo *downlink* com as mensagens contendo o material solicitado pelo usuário (ex.: páginas web, *streams* de voz/vídeo). Isto faz com que o AP necessite de uma maior alocação de recursos que as estações. Como resultado desta característica (principalmente em ambientes com uma carga alta de fluxos *downlink*), surge uma limitação de uso do esquema definido na norma IEEE 802.11, onde um mesmo valor $TXOP_{limit}$ para as estações e o AP é estabelecido.

Para solucionar este problema, o mecanismo DTXOP atualiza periodicamente o $TXOP_{limit}$ das filas de voz e vídeo no AP dada a condição atual de tráfego de cada uma. Em concreto, o mecanismo ajusta o valor com base no número de TS envolvidas numa fila específica e na quantidade de mensagens perdidas por cada TS. No caso da fila de voz, usualmente caracterizada por tráfegos bi-direcionais, o novo valor é dado por:

$$TX_i^{QAP} = n \times TX^{QSTA} \quad (3.39)$$

onde i é o i -ésimo intervalo de verificação dos valores no AP, TX^{QAP} é o $TXOP_{limit}$ do AP, n é o número de estações transmitindo tráfego de voz através do AP e TX^{QSTA} é o $TXOP_{limit}$ das estações. Este valor é atualizado sempre que se altera o número de estações que transmitem dados pela fila de voz. No caso da fila de vídeo, usualmente caracterizada por tráfegos unidirecionais, o $TXOP_{limit}$ é inicializado como na fila de voz, ou seja, a razão entre o seu valor padrão e o número de estações transmitindo na fila de vídeo.

Em seguida o AP calcula a diferença entre o número de mensagens perdidas nos fluxos *uplink* e *downlink*:

$$L_i = Lost_i^{downlink} - Lost_i^{uplink} \quad (3.40)$$

Com base nesta diferença o valor de $TXOP_{limit}$ para a fila de vídeo é dado por:

$$TX_i^{QAP} = \frac{n_i}{n_{i-1}} \times \left[TX_{i-1}^{QSTA} + \left(TX_{i-1}^{QSTA} \times \frac{L_{i-1}}{\alpha} \right) \right] \quad (3.41)$$

onde α é um nível previamente definido com um valor inteiro que permite regular a velocidade de variação imposta pelo algoritmo com relação aos valores obtidos. A velocidade na alteração do

valor de TX^{QAP} diminui proporcionalmente com o aumento de α .

O controlo de admissão, por sua vez, monitoriza continuamente os recursos da BSS e reserva uma largura de banda específica para cada AC. Assim, uma nova TS só pode ser admitida se existir largura de banda disponível para a AC a que se destina. Quando uma nova TS solicita sua admissão, o algoritmo começa por calcular o nível de utilização do meio de comunicação caso a nova TS seja admitida. Esta estimativa baseia-se no modelo apresentado em [78] o qual prevê o *throughput* máximo que uma WLAN pode alcançar baseado no tamanho médio das mensagens transmitidas. Desta forma, primeiramente o algoritmo calcula o tamanho médio das mensagens ($FrSz_{avg}$) dentro de uma janela de observação:

$$FrSz_{avg} = \frac{\sum_{flows} (Fr_{flow} \times FrSz_{flow})}{\sum_{flow} Fr_{flow}} \quad (3.42)$$

onde, Fr_{flow} é o número de mensagens transmitidas e $FrSz_{flow}$ é o tamanho das mensagens transmitidas. Como esta equação considera tanto as mensagens transmitidas pelas TS já admitidas quanto as mensagens que deverão ser transmitidas pela nova TS submetida, o valor $FrSz_{avg}$ corresponde ao tamanho médio das mensagens caso a nova TS seja admitida.

O *throughput* equivalente (Thp_{eq}) é dado pela seguinte equação:

$$Thp_{eq} = \frac{Thp_{ref}}{PkSz_{ref}} \times PkSz_{avg} \times \sqrt{1 + \log_2 \left(\frac{PkSz_{ref}}{PkSz_{avg}} \right)} \quad (3.43)$$

onde Thp_{ref} é o *throughput* de referência da BSS e $PkSz_{ref}$ é o tamanho médio de referência das mensagens. Como a ACU aloca diferentes percentagens de banda para cada AC, caso o valor de Thp_{eq} ultrapasse o valor alocado, a TS é rejeitada, caso contrário é admitida.

A principal limitação deste mecanismo encontra-se na sua premissa que tem como base a garantia de um alto *throughput* nas comunicações. No entanto, isto não representa necessariamente o cumprimento dos prazos de entrega das mensagens.

Em [79] Cicconetti *et al.* propuseram o WTTP (*Wireless Timed Token Protocol*) baseado no TTP (*Timed Token Protocol*) e opera sobre o mecanismo de acesso ao meio do HCCA. Nesta abordagem, um *token* é utilizado para gerir uma lista *Round-Robin*. Nesta lista, cada entrada representa um fluxo (*downlink* ou *uplink*) de uma TS. Uma entrada especial é alocada para representar o tráfego gerado por estações que utilizam um esquema de acesso ao meio baseado em contenção (EDCA e DCF).

A ACU verifica cada entrada da lista e calcula um tempo denominado *sojourn time* (equivalente ao TXOP) para a TS em questão. A ACU insere e remove TS da lista de acordo com a existência (ou não) de mensagens nas suas respectivas filas de transmissão. No caso do tráfego *downlink*, a verificação do estado da fila de transmissão torna-se transparente, uma vez que este tráfego é proveniente do AP. No entanto, como o estado das filas do tráfego *uplink* são conhecidos somente pelas estações, um mecanismo de *piggyback* é utilizado para informar o AP (através das mensagens de dados) acerca do estado da fila de cada TS. Neste caso, as TS são inseridas na lista

ou quando o AP recebe a informação de que há novas mensagens na fila de transmissão da estação ou quando o SI mínimo da TS em questão é atingido.

A rotação do *token* é regida pelo TTRT (*Target Token Revolution Time*), que é um parâmetro equivalente ao SI do sistema. Este valor é calculado pela ACU de acordo com os parâmetros TSPEC enviados pelas estações durante a fase de admissão (taxa média de geração de dados R_i , tamanho da MSDU N_i , tamanho máximo da MSDU M_i e atraso máximo D_i), sendo definido como a metade do menor D_i encontrado dentre as TS admitidas pela ACU:

$$TTRT = \frac{\min_i\{D_i\}}{2} \quad (3.44)$$

O mecanismo divide o tempo definido por TTRT em duas partes denominadas de banda simétrica e banda assimétrica. As transmissões com garantias HCCA são efetuadas apenas na banda simétrica. O *sojourn time* atribuído a cada TS é calculado de forma diferente para cada banda. Na banda simétrica, este tempo (denominado H_i) é fixo e calculado como uma percentagem do TTRT. Na banda assimétrica, este valor é uma porção variável e não reservada de TTRT. Em outras palavras, cada entrada i tem um $H_i \geq 0$ e um TRT_i (*Token Rotation Timer*) inicialmente definido com o valor de TTRT. O valor de TRT_i é decrementado a partir do último serviço provido pelo HC. Quando uma entrada i é servida, a banda assimétrica (a_i) é calculada com base na seguinte regra:

$$a_i = \begin{cases} 0 & TRT_i < 0 \\ \min\{TTRT - H_i, TRT_i\} & \text{caso contrário} \end{cases} \quad (3.45)$$

Desta forma, uma TS assimétrica pode transmitir mensagens apenas se o *token* for recebido antes do previsto, ou seja, antes de $1 \times TTRT$ após a última receção do *token*. Isto ocorre quando outra TS assimétrica consome menos recursos do que o reservado. A alocação do *sojourn time* é definida como uma fração fixa de TTRT, ou seja, este valor nunca é recalculado. No caso de recursos não utilizados, a estação realoca-os para o tráfego baseado em contenção.

O tempo H_i na banda simétrica para ambos os tráfegos CBR e VBR é dado por:

$$H_i = tx(P) \times I(i) + \left\lceil \frac{R_i \times TTRT}{N_i} \right\rceil \times tx(N_i, \Gamma_i) \quad (3.46)$$

onde $tx(P)$ é o tempo necessário para transmitir a mensagem de *polling*, $I(i)$ é uma função que indica qual a direção do fluxo de dados (1 se for *uplink* e 0 se for *downlink*) e $tx(N_i, \Gamma_i)$ o tempo necessário para se transmitir uma MSDU de tamanho N_i a uma taxa de transmissão física Γ_i .

Neste contexto, uma nova TS só pode ser admitida se respeitar a seguinte inequação:

$$\sum_i H_i + \tau \leq TTRT \quad (3.47)$$

onde τ é o tempo necessário para se retomar o controlo do meio de comunicação e iniciar uma nova CAP após uma fase EDCA.

A principal limitação desta proposta encontra-se na abordagem conservadora do escalonador em termos do atraso máximo tolerável pelas estações, implicando que uma estação possa receber o *token* mais vezes que o necessário (tal como o HCCA), o que resulta num *overhead* adicional.

Em [80], Ruscelli *et al.* propuseram um mecanismo auxiliar em cada estação denominado *Overboost*. Este possibilita a gestão de tráfegos VBR através da cooperação entre os mecanismos HCCA e EDCA. Esta cooperação baseia-se no modo HEMM (*HCCA/EDCA Mixed Mode*) definido na norma IEEE 802.11e. Neste caso, as admissões de novas TS são geridas pelo mecanismo HCCA original. No entanto, caso no fim de uma TXOP alguma TS possua ainda mensagens para serem transmitidas, o *Overboost* move as mensagens excedentes para a fila de mais alta prioridade do mecanismo EDCA (fila de voz). Após o término do CFP, o EDCA (através da intervenção do *Overboost*) envia as mensagens excedentes do mecanismo HCCA e posteriormente envia as mensagens alocadas pelo mecanismo EDCA original. Isto faz com que, principalmente em casos onde o tráfego é VBR, as TS possam utilizar o meio de comunicação mais tempo do que o alocado pela ACU através da TXOP.

A principal limitação deste mecanismo encontra-se no indeterminismo que o EDCA impõe às TS. Isto resulta principalmente de dois motivos: **i)** embora o *Overboost* transmita inicialmente as mensagens excedentes do mecanismo HCCA, estas são transmitidas pelo EDCA e assim podem sofrer colisões com mensagens oriundas de redes que estejam fora da esfera de controlo do sistema de tempo-real; e **ii)** a transmissão das mensagens excedentes, cujo número pode variar a cada ciclo, é diretamente dependente do tamanho do CP alocado ao mecanismo EDCA. Assim, caso este seja demasiado pequeno e/ou existam demasiadas mensagens para serem transmitidas, podem ocorrer perdas de *deadlines*.

Em [81], Ansel *et al.* apresentam o FHCF (*Fair HCF*) que é composto por dois escalonadores: um no AP e outro nas estações. O escalonador do AP é responsável por estimar a variação no tamanho da fila de cada estação (q_i^{est}) antes do início do próximo SI e compará-lo com o valor ideal (q_i^{ideal}). O escalonador do AP utiliza uma janela de estimativa de erro para cada TS admitida com o intuito de adaptar o cálculo da TXOP. Por outro lado, o escalonador nas estações é responsável por redistribuir entre suas TS o tempo não utilizado de uma TXOP alocada.

Inicialmente o AP calcula o tamanho ideal da fila de cada TS i no início de cada SI. Neste caso, o tamanho da fila é calculado levando em consideração que no final da sua TXOP todas as mensagens foram transmitidas, ou seja, o tamanho da fila é 0, ou seja:

$$q_i^{ideal} = \frac{\rho_i \times (SI - \sum_{j=1}^i N_j \times (\frac{M_j}{R_{eff}} + 2 \times t_{SIFS} + t_{ACK}))}{M_i} \quad (3.48)$$

onde ρ_i é a taxa média de geração de dados, N_j é quantidade de mensagens que devem ser transmitidas, M_j é o tamanho das mensagens, R_{eff} é a taxa de transmissão efetiva e M_i é o tamanho da mensagem da TS i .

Ao enviar uma mensagem de dados a estação insere no seu cabeçalho (no final da TXOP) o tamanho atual da sua fila de transmissão (q_i^e). Uma vez que o AP conhece este instante de tempo

(t_i^e), é possível então estimar o tamanho da fila (q_i^{est}) da TS_{*i*} no início do próximo SI:

$$q_i^{est} = \frac{\rho_i(SI - t_i^e)}{M_i} + q_i^e \quad (3.49)$$

Como a taxa de envio e o tamanho das mensagens podem variar ao longo do tempo, esta estimativa realizada pelo AP não é exata. Para resolver este problema o FHCF utiliza uma janela de observação w (a qual têm valores reais observados pelo AP) para melhorar as estimativas. Isto significa que ao n -ésimo SI, o AP terá a sua disposição as w últimas estimativas de erro. Estes valores são dados por:

$$\Delta_i^{n-1} = q_i^{b,real}(n-1) - q_i^{b,est}(n-1) \quad (3.50)$$

$$\Delta_i^{n-2} = q_i^{b,real}(n-2) - q_i^{b,est}(n-2) \quad (3.51)$$

$$\dots \quad (3.52)$$

$$\Delta_i^{n-w} = q_i^{b,real}(n-w) - q_i^{b,est}(n-w) \quad (3.53)$$

Assim, o AP pode calcular a diferença entre os valores estimado e real para o próximo SI n :

$$E[|\Delta_i(n)|] \simeq \frac{\sum_{j=n-w}^{n-1} |\Delta_i(j)|}{w} \quad (3.54)$$

Este valor é utilizado para melhorar a estimativa do tamanho da fila da TS i no próximo SI n :

$$q_{i,new}^{b,est}(n) = q_i^{b,est}(n) + E[|\Delta_i(n)|] \quad (3.55)$$

No início do próximo SI, o AP compara o tamanho estimado da fila com o seu tamanho ideal e calcula o número de mensagens adicionais (DN_i^{est}) que possam existir. O AP calcula o tempo de ajuste (t_i^{est}), que pode ser positivo ou negativo, que é necessário para cada TS recalculer a TXOP:

$$t_i^{est} = DN_i^{est} \times \left(\frac{M_i}{R_{eff}} + 2 \times t_{SIFS} + t_{ACK} \right) \quad (3.56)$$

Para verificar se o escalonador é capaz de realizar estas alterações o AP compara a soma de todos os valores positivos (T_P) e a soma de todos os valores negativos (T_N) com o tempo restante do mecanismo HCCA após alocar todas as TXOP (T'). Caso $T_P - T_N > T'$, significa que o escalonador não é capaz de alocar todo o tempo extra solicitado e que o mesmo deve ser reduzido. Neste caso, para garantir igualdade entre as TS, o escalonador reduz cada t_i^{est} positivo com uma percentagem β . Por outro lado, cada t_i^{est} negativo é incrementado na mesma percentagem. O valor de β é dado por:

$$\beta = \frac{(T_P - T_N) - T'}{T_P + T_N} \quad (3.57)$$

Baseado nestas alterações, o tempo adicional efetivo (t_i^{add}) aplicado a cada TS é dado por:

$$t_i^{add} = \begin{cases} (1 + \beta)t_i^{est} & t_i^{est} \geq 0 \\ (1 - \beta)t_i^{est} & t_i^{est} < 0 \end{cases} \quad (3.58)$$

onde o valor t_i^{add} é calculado juntamente com os valores das TXOP alocadas pelo HCCA.

O escalonador executado em cada estação tem como função redistribuir o tempo adicional de uma TXOP entre as suas TS. Este executa o mesmo cálculo que o escalonador do AP mas com mais exatidão, uma vez que sabe exatamente o tamanho das filas de cada TS no momento da receção da mensagem de *polling*. Assim, é capaz de estimar o tamanho da fila no final da TXOP e também o tempo adicional necessário para a TS. De acordo com o tempo T alocado por uma TXOP, a estação pode estimar o tempo restante T' que pode ser realocado entre as TS considerando o número de mensagens (N_i) para transmitir em cada TS_i e seus respectivos tempos de transmissão. O valor de T' é dado pela seguinte equação:

$$T' = T - \sum_{i=1}^p N_i \times \left(\frac{M_i}{R_{eff}} + 2 \times t_{SIFS} + t_{ACK} \right) \quad (3.59)$$

A principal limitação deste mecanismo encontra-se na generalização das prioridades do tráfego, ignorando a *deadline* de cada TS. Além disto, embora a verificação das filas de transmissão seja capaz de garantir uma gestão do tráfego aperiódico e esporádico, não é capaz de fazer o mesmo com o tráfego VBR (na variação no tamanho das mensagens), uma vez que esta informação é enviada ao AP somente no momento da admissão, e este somente estima o tamanho da fila baseado no número de mensagens e não no tempo necessário para a sua transmissão.

Toscano e Lo Bello [82] apresentam um mecanismo de controlo de admissão baseado no escalonador EDF não preemptivo. Cada TS admitida pelo sistema define uma taxa de sucesso mínima a qual esta deve operar, ou seja, a fração mínima de mensagens que devem ser entregues corretamente e antes da *deadline*. O mecanismo utiliza a taxa de perda de cada TS_i para calcular a sua probabilidade de erro (E_i). Cada TS_i é descrita pelos seguintes parâmetros: *deadline* relativo (D_i), tempo máximo de transmissão (C_i), taxa de erro (E_i) e taxa de sucesso (s_i). Caso ocorra um erro de transmissão a mensagem é reescalada com a mesma *deadline* da mensagem original.

O principal objetivo deste mecanismo é considerar no teste de escalonabilidade o número de retransmissões que permite que a taxa de sucesso seja atingida. Uma vez que as probabilidades de sucesso para cada tentativas de transmissão são estatisticamente independentes, pode-se assumir que dada a probabilidade de erro E_i , a probabilidade de uma mensagem ser perdida após M tentativas de transmissão é E_i^M . Como resultado, o número mínimo de tentativas de transmissão M_i que satisfaz os requisitos em termos de probabilidade de sucesso na transmissão s_i é dado por:

$$M_i = \lceil \log_{E_i}(1 - s_i) \rceil = \left\lceil \frac{\log(1 - s_i)}{\log(E_i)} \right\rceil \quad (3.60)$$

onde M_i é decrementado em 1 ($M_i - 1$) para se obter exclusivamente o número de retransmissões.

No entanto, considerar que cada TS_i deva ter um número de retransmissões M_i é demasiado pessimista, uma vez que somente torna-se necessária uma retransmissão quando a transmissão anterior falhou. A análise do mecanismo é realizada pela modelagem das retransmissões das mensagens como uma TS periódica. Do grupo original de TS ($\Phi = TS_1, TS_2, \dots, TS_N$) é derivado um novo grupo Φ^* o qual inclui as retransmissões de mensagens.

Para cada TS_i do grupo original, um grupo $TS_{i,1}, TS_{i,2}, \dots, TS_{i,M_i}$ é gerado, onde $TS_{i,1}$ é a primeira transmissão. Como resultado temos $\Phi^* = \bigcup_{i=1}^N \{TS_{i,1}, TS_{i,2}, \dots, TS_{i,M_i}\}$. A mesma notação é utilizada para os períodos das mensagens onde $T_{i,1}$ representa o período da primeira transmissão. Os seus valores são definidos de acordo com a frequência média de cada retransmissão. Para a primeira transmissão de cada TS_i o período é conhecido, ou seja, $T_{i,1} = T_i$. Por outro lado, para a primeira retransmissão, como a probabilidade de erro é E_i então $T_{i,2} = E_i/T_i$. Sendo assim, a segunda retransmissão é dada com a probabilidade de erro E_i^2 , o que resulta em $T_{i,3} = E_i^2/T_i$. Com base nesta consideração, os períodos $T_{i,j}$ das TS podem ser definidos por:

$$T_{i,j} = \begin{cases} T_i & j = 1 \\ \frac{T_i}{(E_i)^{j-1}} & j = 2, 3, \dots, M_i \end{cases} \quad (3.61)$$

Levando em conta que o *deadline* relativo de uma retransmissão é igual ao da mensagem original (1a tentativa), e que o escalonador EDF é não preemptivo, uma alteração no teste de escalonabilidade foi proposta. Desta forma, uma condição suficiente para alcançar a taxa de sucesso desejada em um grupo de N TS escalonadas de acordo com o algoritmo EDF é possível respeitando a seguinte inequação:

$$\sum_{i=1}^N \sum_{j=1}^{M_i} \frac{C_{i,j}}{T_i} + \frac{B_k}{T_k} \leq 1 \quad \forall k = 1, 2, \dots, N \quad (3.62)$$

onde C_i é o tempo de transmissão e B_k é o bloqueio do sistema, dado por $B_k = \max(C_{i,j})$.

No entanto, esta inequação acaba por ser pessimista, uma vez que transforma o tempo entre chegadas das retransmissões em *deadlines* relativos. Com o objetivo de propor um modelo menos pessimista, os autores utilizaram a análise de Spuri [83] que se baseia no conceito de período de ocupação (*busy period*). Desta forma é possível prover a seguinte condição suficiente:

$$d \geq \sum_{D_i \leq d} \sum_{j=1}^{M_i} \left(1 + \left\lfloor \frac{d - D_i}{T_{i,j}} \right\rfloor \right) C_{i,j} + B_{k(d)} \quad (3.63)$$

onde d é o *deadline* absoluto, D_i o *deadline* relativo e $B_{k(d)}$ é o maior bloqueio sofrido pela TS k .

Embora esta proposta otimize o cálculo da TXOP com base num mecanismo que avalia independentemente cada retransmissão realizada por uma TS, isto por si só não garante o cumprimento das *deadlines*. Esta limitação resulta do facto de que nem sempre a alocação de uma grande TXOP a uma TS significa o cumprimento da respectiva *deadline*, uma vez que esta pode ser perdida em consequência de um atraso no acesso ao meio.

Além disso, o cálculo de ocupação do meio de comunicação efetuado por este mecanismo, pelo mecanismo apresentados por Cruz *et al.* [74] e pelos mecanismos PRCW [73] e DTXOP

[77] apresentando anteriormente não levam em consideração o tráfego gerado por estações que estão fora da esfera de controlo da arquitetura de tempo-real. Isto pode levar a decisões erradas na admissão de novas TS ou na definição dos tempos de bloqueio, podendo em algumas situações resultar na saturação do meio de comunicação e consequentemente degradação da QoS de todas as TS previamente admitidas.

3.2.4 Síntese dos Mecanismos de Controlo de Admissão

A implementação dos mecanismos de controlo de admissão tem um papel fundamental na construção de uma arquitetura de comunicação de tempo-real. Através deste é possível garantir os recursos necessários e impedir a saturação do meio de comunicação, consequentemente impedindo também a perda das *deadlines* das mensagens. Neste contexto, alguns mecanismos encontrados na literatura focam os seus esforços na resolução do problema da imprecisão gerada pelo tráfego VBR, enquanto outros focam-se na limitação do tamanho imposto às TXOP, facto que impede um melhor desempenho na transmissão de fluxos contendo grandes quantidades de dados (por exemplo, vídeos de alta definição).

No entanto, é importante ter em conta que estes problemas não ocorrem na grande maioria dos sistemas de comunicação de tempo-real utilizados em ambientes industriais. Isto deve-se ao facto de, em termos gerais, as mensagens trocadas entre os dispositivos de tempo-real serem periódicas, de tamanho fixo (ou seja, caracterizadas por um tráfego CBR) e contendo uma pequena quantidade de dados (eliminando assim o problema de limitação do tamanho da TXOP).

Outros mecanismos focam os seus esforços na precisão do cálculo da TXOP atribuída a cada fluxo de dados admitido. O principal desafio encontrado por estas propostas é a variação ocorrida na taxa de transmissão de cada estação, que pode surgir devido à mobilidade ou então por possíveis interferências sofridas pelas estações. Neste contexto, algumas propostas possibilitam ao controlo de admissão recalcular tanto os recursos disponíveis no sistema quanto os recursos necessário para cada TS. Estes cálculos geralmente são efetuados com base na verificação da nova taxa de transmissão utilizada por cada estação.

No contexto do algoritmo de escalonamento, são utilizados diferentes algoritmos de tempo-real para tornar possível a definição da ordem de transmissão das TS admitidas. Esta ordenação pode tomar como base parâmetros enviados pelas estações via TSPEC (por exemplo, atraso limite), medidas efetuadas na rede (por exemplo, carga total) ou até medidas efetuadas nas estações (por exemplo, tamanho da fila de transmissão).

Portanto, ao considerarmos a transmissão de tráfego de tempo-real num ambiente de comunicação aberto, podemos concluir que além das informações TSPEC enviadas pelas estações de tempo-real, a ACU deve levar em conta também o comportamento dinâmico do meio de comunicação. Este comportamento pode ser alterado em resultado de transmissões efetuadas por estações de não tempo-real ou por ruídos no meio de comunicação.

Assim, e com base na classificação previamente definida, podemos concluir que a utilização de uma abordagem *híbrida* pelo mecanismo de controlo de admissão apresenta-se como uma opção promissora, uma vez que esta é capaz de iniciar o sistema a partir de um modelo analítico e

posteriormente alimentá-lo com informações constantemente atualizadas, sejam estas do meio de comunicação, sejam das estações. Assim é possível eliminar as limitações apresentadas pelas restantes abordagens⁵.

Outro ponto fundamental na construção de um mecanismo de controlo de admissão é a utilização do elemento TSPEC para que as estações possam informar a ACU dos requisitos e características de cada TS que desejam transmitir. Além disso, este procedimento possibilita também a definição das *deadlines* das mensagens, informação considerada fundamental para um sistema de comunicação de tempo-real. Como consequência, torna-se possível utilizar um algoritmo de escalonamento de tempo-real para organizar a sequência de transmissão das mensagens com base nas suas respectivas *deadlines*.

Por fim, outra característica importante que deve ser levada em consideração para a construção de um mecanismo de controlo de admissão é a respectiva operação num ambiente de comunicação aberto, onde o tráfego gerado pelas estações que estão fora da esfera de controlo da arquitetura de tempo-real pode resultar em atrasos e bloqueios diferentes dos calculados num ambiente fechado.

Em jeito de conclusão, são listados seguidamente os requisitos julgados necessários para a implementação de um mecanismo de controlo de admissão de um sistema de comunicação de tempo-real:

- obrigatoriamente utilizar uma abordagem híbrida, obtendo tanto informações das TS (via TSPEC), quanto do meio de comunicação para efetuar a admissão (ou não) de uma nova TS;
- deve ser capaz de gerir variações repentinas que possam ocorrer nos valores medidos no meio de comunicação e/ou nas estações;
- preferencialmente ser implementado sobre uma arquitetura centralizada;
- caracterizar as TS através do uso de informações TSPEC;
- implementar um algoritmo de escalonamento de tempo-real para organizar a sequência de transmissão das mensagens com base nas respectivas *deadlines*;
- identificar e remover TS que possam se encontrar num estado de "falha", ou seja, que estejam com recursos alocados mas que não estejam em operação.

A Figura 3.2 resume e classifica os diferentes mecanismos de controlo de admissão apresentados previamente. Além de apresentar uma classificação quanto à abordagem utilizada (definida no início desta secção), apresenta também algumas das principais características de cada mecanismo avaliado. A primeira divide os mecanismos quanto à sua arquitetura (centralizada ou distribuída). De forma complementar são apresentados os tipos de tráfegos (CBR, VBR e aperiódico⁶) suportados por cada mecanismo, suporte às informações TSPEC, utilização de informações provenientes

⁵As limitações das abordagens utilizadas para implementar o mecanismo de controlo de admissão são discutidas no início desta secção.

⁶O suporte ao tráfego esporádico não é caracterizado uma vez que este pode ser modelado como um tráfego periódico com período igual ao seu intervalo mínimo de geração (MIT – *Minimum Interarrival Time*).

do ambiente de comunicação aberto (ACA), utilização da *deadline* como referência para a transmissão, tipo de algoritmo de escalonamento utilizado (quando existir) e tipo do mecanismo de controlo de acesso ao meio sobre o qual a proposta foi construída.

Desta forma, podemos observar que das 19 soluções apresentadas, apenas 7 implementam uma abordagem *híbrida*. Destas 7 soluções, apenas 4 implementam um algoritmo de escalonamento de tempo-real e utilizam a *deadline* das mensagens como referência para organizar a transmissão das TS. Por fim, ao analisarmos as soluções que consideram a sua operação num ambiente de comunicação aberto, este número acaba por ser reduzido para apenas uma solução, apresentada por Toscano e Lo Bello [82].

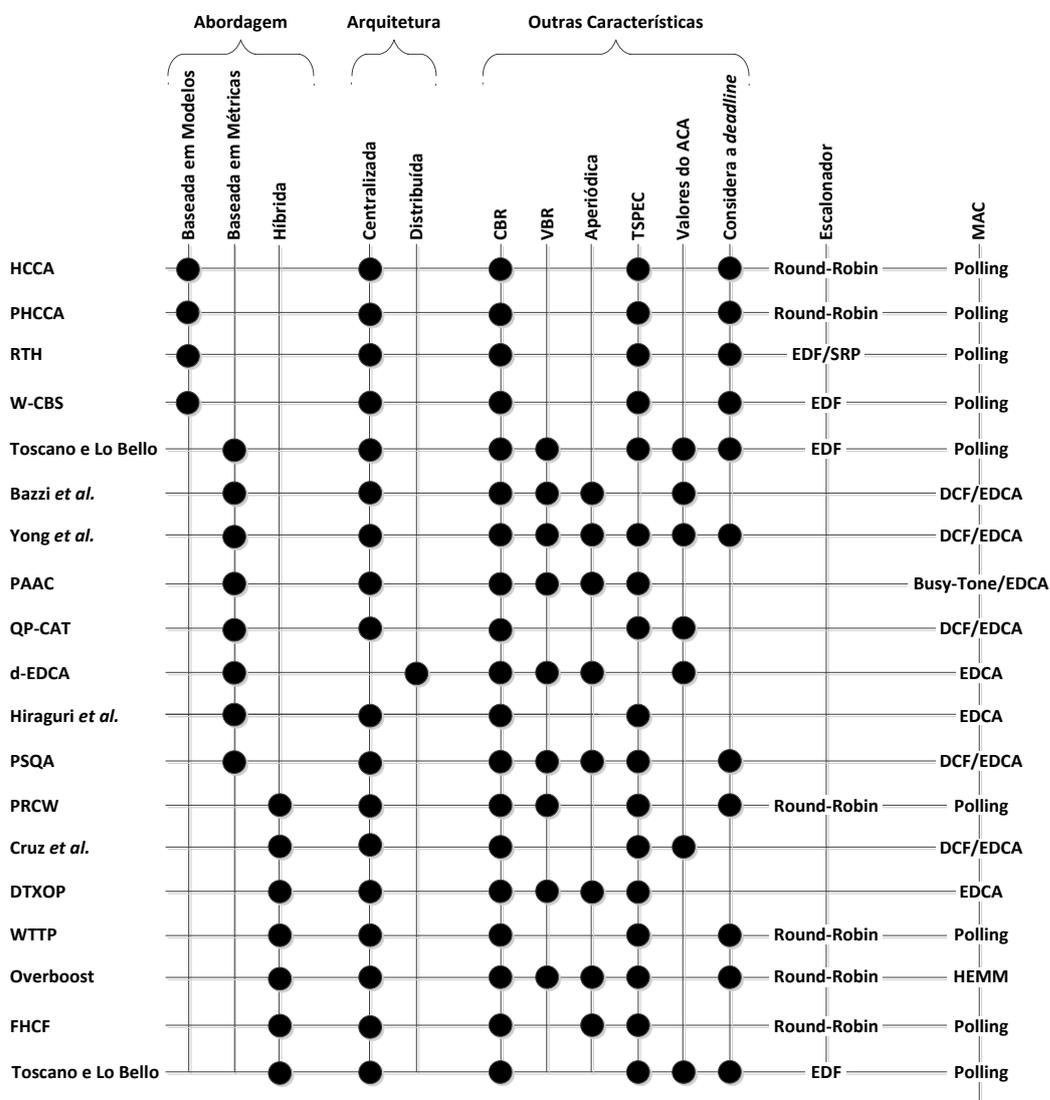


Figura 3.2: Comparação entre as propostas de controlo de admissão apresentadas.

No entanto, esta solução não satisfaz completamente o critério relacionado com os valores observados num ambiente de comunicação aberto. Isto porque, embora o mecanismo considere que as colisões possam ocorrer com qualquer outra estação, o cálculo utilizado para a definição do bloqueio não considera que as ocupações no meio de comunicação podem ser geradas pelo tráfego proveniente de estações que estão fora da esfera de controlo da arquitetura de tempo-real. Como resultado, o bloqueio real pode ser superior ao bloqueio calculado pelo mecanismo.

Torna-se assim evidente a necessidade de avançar com uma nova proposta para o mecanismo de controlo de admissão. Esta, por sua vez, deve ser capaz de considerar interferências provenientes de tráfego transmitido por estações que estão fora da esfera de controlo da arquitetura de tempo-real. Para isto, esta deve ser construída sobre uma abordagem *híbrida*, que permita atualizar dinamicamente o comportamento do algoritmo de escalonamento de tempo-real com base tanto nas informações enviadas pelas estações via TSPEC, quanto em informações obtidas da rede ou das próprias estações de tempo-real.

3.3 Conclusões

Da análise dos mecanismos de controlo de acesso ao meio e controlo de admissão apresentados neste capítulo torna-se evidente a necessidade de uma integração entre ambos para que seja possível propor uma arquitetura de comunicação de tempo-real para redes IEEE 802.11.

Um factor fundamental que deve ser considerado no momento da concepção deste tipo de arquitetura é a premissa de que o ambiente de comunicação é aberto, ou seja, pode existir tráfego a ser transmitido a partir de redes que estão fora de esfera de controlo desta arquitetura e que irá interferir com a transmissão do tráfego de tempo-real. Assim, uma nova solução deve ser capaz de operar neste tipo de ambiente, sem que para isto seja necessário controlar as estações que não fazem parte da arquitetura de tempo-real.

Neste contexto e, no que diz respeito aos mecanismos de controlo de acesso ao meio, constatou-se que a maioria é incapaz de fornecer tal característica. E das soluções que suportam este tipo de operação, apenas algumas têm a sua implementação compatível com *hardware* COTS.

Por outro lado, ao analisarmos esta mesma característica sob a ótica dos mecanismos de controlo de admissão, podemos concluir que a maioria das soluções não leva em conta os efeitos que as estações localizadas fora da esfera de controlo da arquitetura de tempo-real podem gerar para a transmissão do tráfego de tempo-real. Isto resulta no dimensionamento errado dos recursos disponíveis para alocação de recursos e, conseqüentemente, em decisões (por parte da ACU) que podem comprometer todos os fluxos previamente admitidos.

Desta forma, podemos concluir que se torna evidente a necessidade de avançar com uma nova proposta, e que esta se deve focar na integração dos mecanismos de controlo de acesso ao meio com o controlo de admissão, bem como na proposta de novos mecanismos deste tipo, dado que os existentes não são totalmente adequados para o suporte de tráfego de tempo-real.

Capítulo 4

A Arquitetura RT-WiFi

*Neste capítulo é apresentada a proposta de uma arquitetura de comunicação de tempo-real para redes IEEE 802.11 infraestruturadas denominada **RT-WiFi**, cujo principal objetivo é suportar a transmissão de tráfego soft real-time em ambientes de comunicação abertos. Inicialmente é apresentada uma visão geral do seu funcionamento, sendo posteriormente detalhados os mecanismos de controlo de acesso ao meio e controlo de admissão.*

4.1 Introdução

No capítulo anterior foram analisadas diversas soluções propostas pela comunidade científica para a transmissão de tráfego de tempo-real em redes IEEE 802.11, tendo-se definido no final do capítulo um conjunto de pré-requisitos necessários para que uma nova proposta.

Com base nestes pré-requisitos, este capítulo apresenta o RT-WiFi, uma arquitetura de comunicação de tempo-real que oferece um conjunto de mecanismos capaz de controlar o acesso ao meio das mensagens de tempo-real, a admissão de novas TS (*Traffic Stream*) e o escalonamento das respectivas transmissões. O seu objetivo é fornecer um serviço de comunicação de tempo-real em redes IEEE 802.11 infraestruturadas a operar em ambientes de comunicação abertos. Neste contexto, uma das suas principais vantagens é a capacidade de fornecer este tipo de serviço controlando apenas as estações de tempo-real.

A arquitetura RT-WiFi baseia-se no mecanismo EDCA (*Enhanced Distributed Channel Access*) e utiliza uma topologia de rede infraestruturada. Esta topologia permite um melhor controlo do algoritmo de escalonamento e do mecanismo de controlo de admissão, uma vez que assim é possível ao coordenador central (localizado no AP – *Access Point*) ter uma visão global de todo o tráfego transmitido no ambiente de comunicação onde o RT-WiFi está a operar.

A Figura 4.1 apresenta o fluxo das mensagens de tempo-real. Neste caso, as mensagens trocadas entre duas estações pertencentes à mesma BSSID (*Basic Service Set Identifier*) são inicialmente enviadas pela estação origem ao AP (fluxo de dados *uplink*) e posteriormente reencaminhada para a estação destino (fluxo de dados *downlink*).

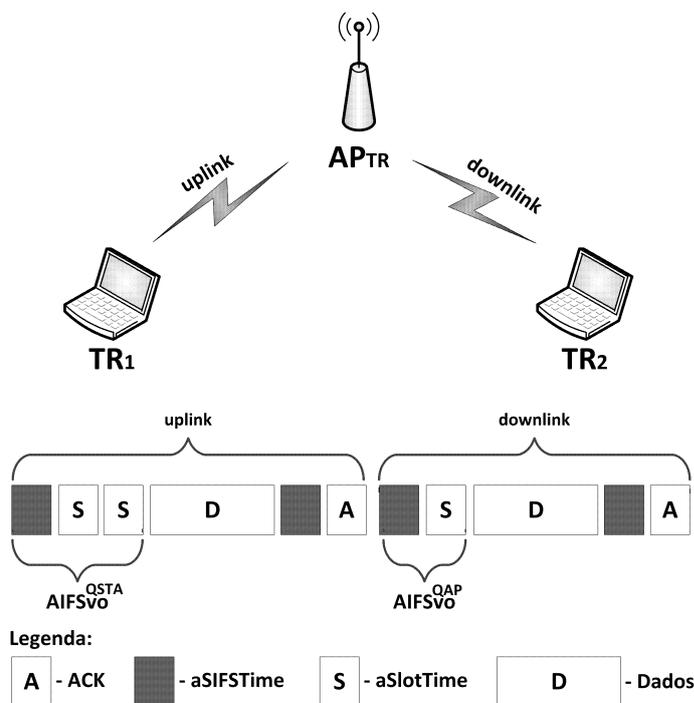


Figura 4.1: Fluxo das mensagens na arquitetura RT-WiFi.

O RT-WiFi faz uso de uma arquitetura organizada em duas camadas (Figura 4.2). Na camada inferior encontra-se o *Mecanismo de Controlo de Acesso ao Meio* que tem como objetivo principal aumentar a prioridade no acesso ao meio do tráfego de tempo-real, nomeadamente em situações onde ocorram colisões. Desta forma é possível diminuir o indeterminismo gerado pelo mecanismo CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) intrínseco da norma IEEE 802.11. Na camada superior encontra-se o *Mecanismo de Controlo de Admissão*, que tem como objetivo principal controlar a admissão de novas TS na rede de tempo-real de forma a evitar a sua sobrecarga e consequentemente a degradação da QoS (*Quality of Service*) das TS previamente admitidas. Como esta camada é responsável pela gestão da arquitetura RT-WiFi, encontra-se apenas implementada no coordenador central, ou seja, no AP.

Para que o *Mecanismo de Controlo de Acesso ao Meio* possa aumentar a prioridade do tráfego de tempo-real é utilizado um mecanismo denominado FCR (*Force Collision Resolution*) [50] que desabilita o procedimento de *backoff*. Assim, para evitar múltiplas colisões entre as mensagens de tempo-real e também reduzir o *overhead* gerado pelo esquema de *polling* tradicional, as transmissões de tempo-real são sequenciadas através de um esquema TDMA (*Time Division Multiple Access*) que aloca *slots* para as estações transmitirem.

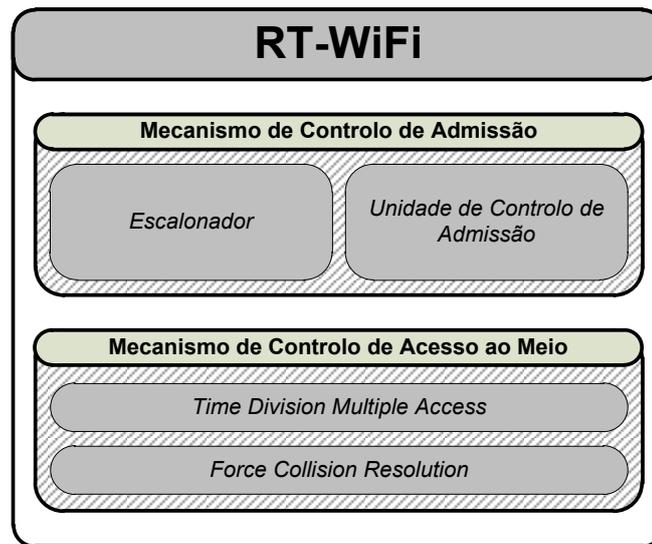


Figura 4.2: Arquitetura RT-WiFi.

Cada ciclo TDMA é inicializado com a transmissão de uma mensagem de *beacon*. Esta mensagem contém uma lista de escalonamento que é difundida para as estações de tempo-real. Nesta lista são inseridas a ordem e o tamanho dos *slots* alocados a cada estação. Tanto a ordem quanto o tamanho dos *slots* podem ser modificados dinamicamente em cada ciclo TDMA. Neste caso, a ordem é gerida por um algoritmo de escalonamento de tempo-real e o tamanho por um mecanismo capaz de estimar (com base num histórico) o tempo necessário para se efetuar a transmissão de uma mensagem com sucesso. Isto possibilita uma maior flexibilidade ao RT-WiFi. O mecanismo de controlo de acesso ao meio é detalhado na secção 4.2.

No caso do *Mecanismo de Controlo de Admissão*, o uso de uma topologia infraestruturada permite que o nível de ocupação do meio de comunicação possa ser estimado de forma mais precisa, quando comparado a uma topologia *Ad Hoc*. Com o objetivo de evitar que as prioridades/características das TS sejam generalizadas numa única prioridade, a ACU (*Admission Control Unit*) exige que as estações TR enviem ao AP informações TSPEC (*Traffic Specification*) acerca de cada TS que desejam transmitir. A ACU tomará a decisão de aceitar (ou não) a nova TS com base nestas informações e em valores previamente estimados no meio de comunicação, caracterizando-se por uma abordagem *híbrida* na sua implementação. Além disso, é também implementado um mecanismo capaz de identificar e eliminar as TS que foram admitidas e que por algum motivo pararam de transmitir tráfego. O mecanismo de controlo de admissão é detalhado na secção 4.3.

Para a implementação da arquitetura RT-WiFi levou-se em conta dois modelos de interação utilizados para a troca das mensagens entre a estação origem e a estação destino. O primeiro assume um modelo de comunicação ponto-a-ponto. Um exemplo deste modelo é uma planta industrial contendo sensores que, após efetuarem suas medidas, enviam mensagens para um controlador, que pode também enviar mensagens para os atuadores (Figura 4.3(a)). O segundo modelo baseia-

se na difusão de dados (*broadcast*). Neste caso o emissor envia a mesma informação para todos os restantes dispositivos da rede. Um exemplo é um grupo de robôs móveis que precisam trocar informações entre si de forma que todos possam ter uma visão global do sistema (Figura 4.3(b)).

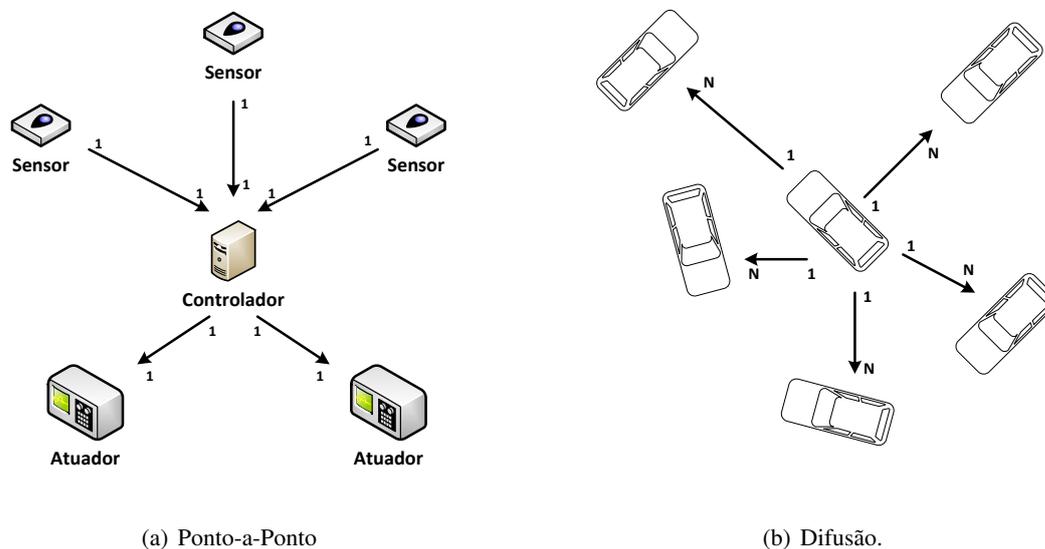


Figura 4.3: Modelos de interação entre os dispositivos de tempo-real.

No contexto da arquitetura RT-WiFi, decidiu-se focar os esforços no suporte as aplicações que utilizam um modelo de interação ponto-a-ponto, uma vez que este é o mais geralmente utilizado pelas aplicações de controlo. No entanto, na secção 4.4, são apresentadas e discutidas as alterações necessárias para que a arquitetura RT-WiFi possa suportar aplicações que utilizam o modelo de interação baseado na difusão de dados.

4.2 Mecanismo de Controlo de Acesso ao Meio

O objetivo do *Mecanismo de Controlo de Acesso ao Meio* é aumentar a prioridade do tráfego de tempo-real no seu acesso ao meio. Este, por sua vez, está estruturado em duas subcamadas (Figura 4.2): a subcamada inferior implementa um mecanismo de separação de tráfego denominado FCR, e a subcamada superior implementada um mecanismo de coordenação baseado num esquema TDMA.

O mecanismo FCR, implementado tanto nas estações TR como no AP TR (doravante denominado AP_{TR}), tem como objetivo aumentar a prioridade no acesso ao meio das mensagens provenientes destes dispositivos em caso de colisão. Basicamente o funcionamento do FCR consiste na desativação do mecanismo de *backoff* e na utilização da fila de voz. Assim, sempre que existir uma colisão entre uma estação TR (ou AP_{TR}) e uma ou mais estações (ou AP) NTR, todos os envolvidos (com exceção dos dispositivos TR) irão executar o procedimento de *backoff* utilizando os valores padrões definidos pelos seus respectivos mecanismos de acesso ao meio.

Neste contexto, os dispositivos TR desativam o mecanismo de *backoff* através da atribuição de um valor igual à zero aos parâmetros aCW_{min} e aCW_{max} . Assim, sempre que ocorrer uma colisão, os dispositivos TR tentarão retransmitir suas mensagens imediatamente após seu IFS (*Interframe Space*). Por outro lado, os dispositivos NTR deverão aguardar um intervalo de tempo equivalente ao $backoff + IFS$ da mensagem para tentarem iniciar as suas retransmissões. Além disto, os dispositivos TR transmitem as suas mensagens através da utilização de valores IFS equivalentes aos valores definidos para o AIFS da categoria de mais alta prioridade da função EDCA, ou seja, a categoria de voz (VO). Desta forma, os valores IFS utilizados pelo tráfego *uplink* ($AIFS_{VO}^{QSTA}$) e pelo tráfego *downlink* ($AIFS_{VO}^{QAP}$) são dados respectivamente por:

$$AIFS_{VO}^{QSTA} = aSIFSTime + 2 \times aSlotTime \quad (4.1)$$

$$AIFS_{VO}^{QAP} = aSIFSTime + aSlotTime \quad (4.2)$$

onde o valor de $AIFS_{VO}^{QSTA}$ equivale ao valor de $aDIFSTime$ utilizado pela função DCF e o valor de $AIFS_{VO}^{QAP}$ equivale ao valor de $aPIFSTime$ (Figura 4.1).

A principal limitação deste mecanismo é que, se duas ou mais estações TR tentarem aceder ao meio simultaneamente, vão ocorrer repetidas colisões que conseqüentemente resultarão no descarte das mensagens após ser atingido o número máximo de retransmissões. Para solucionar este problema é utilizado um mecanismo de coordenação que tem por objetivo sequenciar as transmissões TR. Este sequenciamento é realizado através de um esquema TDMA que separa as transmissões TR em *slots* independentes.

Cada *slot* é definido pelos limites inicial SP (*Start Point*) e final EP (*End Point*), e deve ter um tamanho suficiente para a transmissão de ambos os fluxos de dados (*uplink* e *downlink*), além de possíveis retransmissões que ambos os fluxos possam necessitar para suportar erros de comunicações. Neste contexto, como a transmissão do fluxo *downlink* ocorre imediatamente após a receção do fluxo *uplink*, o esquema TDMA é implementado apenas nas estações TR, não sendo necessária a sua implementação no AP_{TR} .

O tamanho e a sequência dos *slots* são geridos dinamicamente pelo *Mecanismo de Controlo de Admissão* (ver secção 4.3) por forma a garantir uma maior flexibilidade ao algoritmo de escalonamento da arquitetura RT-WiFi. Neste contexto, duas situações podem ser consideradas críticas. A primeira é a admissão de múltiplas TS quando a ocupação do meio de comunicação é baixa. Neste caso, geralmente são necessário *slots* pequenos. Quando a ocupação do meio de comunicação aumenta, há necessidade de alocar *slots* maiores, o que pode levar à exclusão de alguma TS por forma de garantir a QoS da rede TR. Esta situação é considerada crítica pois obriga à realização de um novo teste de escalonabilidade a cada alteração no tamanho dos *slots*.

Para evitar estas situações, no momento da admissão de uma TS_k , a ACU da arquitetura RT-WiFi define um tamanho máximo C_k^{max} para os *slots* alocados a esta TS. Assim, independentemente do tamanho que os *slots* possam assumir durante a operação da rede, a ACU sempre utilizará o parâmetro C^{max} das TS admitidas para efetuar o teste de escalonabilidade de uma nova

TS. Assim, além de evitar que variações no tamanho dos *slots* resultem em exclusões de TS previamente admitidas, esta abordagem permite também que o teste de escalabilidade da TS seja executado somente no momento da sua admissão, reduzindo assim o custo computacional desta operação na ACU.

Neste contexto, durante a operação da rede a dimensão atual dos *slots* pode ser inferior ao definido pelo seu respectivo parâmetro C^{max} . Esta diferença relativa entre o tamanho máximo e o tamanho atual dos *slots* é considerado o **factor de compressão** do sistema. Com o objetivo de otimizar a utilização dos recursos disponíveis na rede de tempo-real, a arquitetura RT-WiFi utiliza este factor de compressão para possibilitar que as estações TR transmitam dois tipos diferentes de tráfego TR (**alta e baixa** prioridade) além do tráfego NTR.

Num ambiente industrial, isto permite que além das mensagens de tempo-real de alta prioridade, as estações possam também transmitir mensagens de tempo-real de menor prioridade, como por exemplo as mensagens de gestão do *software* SCADA (*Supervisory Control and Data Acquisition*). Além disso, a compatibilidade com a transmissão de tráfego NTR permite que as estações RT-WiFi transmitam mensagens de gestão, como por exemplo o *beacon* (no caso do AP_{TR}), requisições e respostas ADDTS e também pedidos de associação e autenticação.

A diferença entre os tráfegos TR de **alta e baixa** prioridade reside no facto do primeiro receber uma garantia da ACU de que terá uma alocação de *slots* de acordo com o seu período de geração de mensagens. Por outro lado, esta mesma garantia não é dada ao tráfego TR de baixa prioridade. Este receberá apenas uma alocação de *slots* se existirem recursos disponíveis no sistema. A definição da disponibilidade de recursos é dada a partir da soma dos recursos disponíveis (ou seja, não alocados por nenhuma TS) e dos recursos disponíveis do **factor de compressão** (ou seja, a quantidade de recursos que embora sejam reservados ao tráfego TR de alta prioridade, no momento atual, não estão sendo utilizados). Importa referir que, embora o tráfego TR de baixa prioridade não receba uma garantia de alocação de *slots* por parte da ACU, mantêm-se as configurações do mecanismo FCR e do esquema TDMA.

No caso de uma estação RT-WiFi desejar transmitir mensagens pertencentes a uma TS NTR, esta utiliza o mecanismo DCF ou EDCA definido pela norma IEEE 802.11, não havendo assim a necessidade de aprovação prévia pelo mecanismo de controlo de admissão. Caso contrário, a estação utiliza as definições da arquitetura RT-WiFi, sendo obrigatória a aprovação prévia pelo mecanismo de controlo de admissão. Neste contexto, a estação pode alternar entre modos RT-WiFi e DCF/EDCA conforme a necessidade de transmissão de suas TS, sendo que o modo DCF/EDCA será ativado apenas durante os intervalos de tempo em que não existam *slots* alocados para as TS TR, independentemente da sua prioridade.

Como todas as mensagens TR são transmitidas através da fila de voz, é possível que mensagens TR de diferentes prioridades e mensagens NTR estejam misturadas na mesma fila. Neste contexto, cada estação TR implementa um escalonador local que permite distinguir as mensagens TR e NTR alocadas na fila de voz evitando assim inversões de prioridade nas suas transmissões. Esta diferenciação ocorre através do uso de uma lista local contendo todas as TS admitidas pela ACU e também os seus respectivos tipos de tráfego. Ao cruzar as informações desta lista com o parâmetro

TID (*Traffic Identifier*) contido no campo *QoS Control* pertencente ao cabeçalho MAC (*Medium Access Control*) de cada mensagem, o escalonador da estação consegue identificar qual o tipo de tráfego e, se necessário, modificar o seu posicionamento.

Tal como descrito anteriormente, as comunicações da arquitetura RT-WiFi são organizadas em ciclos TDMA (Figura 4.4). O tamanho de cada ciclo é dado pelo parâmetro SI (*Service Interval*) que é definido pela ACU com base no período de geração de mensagens das TS admitidas. O início de cada ciclo TDMA é definido pelo envio de uma mensagem de *beacon* pelo AP. Esta mensagem é utilizada para sincronizar o relógio das estações com o relógio do AP e também difundir uma lista de escalonamento. Esta lista informa as estações acerca do instante inicial (*SP - Start Point*) e final (*EP - End Point*) dos *slots* alocados a cada TS que tem permissão para transmitir no ciclo atual.

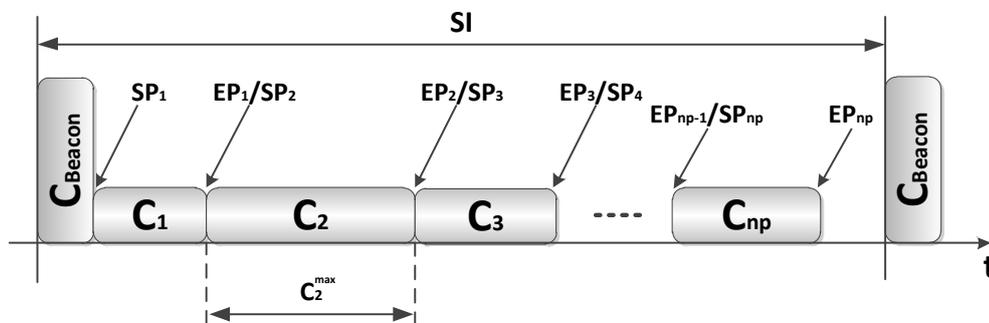


Figura 4.4: Ciclo TDMA na arquitetura RT-WiFi.

Uma TS_i só pode tentar aceder ao meio de comunicação durante o intervalo de tempo compreendido entre SP_i e EP_i (C_i), e para a transmissão de uma única mensagem de dados, independentemente do número de mensagens que possam existir na fila de transmissão. Esta característica simplifica o algoritmo de escalonamento e mantém-se compatível com o modelo de tráfego assumido pela arquitetura RT-WiFi. Para garantir que as estações TR transmitam uma única mensagem por vez, o respectivo parâmetro TXOP (*Transmission Opportunity*) é definido como 0. Além disso, para evitar transmissões desnecessárias, o escalonador local é também responsável por eliminar mensagens que eventualmente possam ter perdido suas respectivas *deadlines* e não tenham iniciado uma transmissão.

Uma observação importante é que, mesmo com a utilização do mecanismo FCR para aumentar a prioridade no acesso ao meio de uma estação TR (ou AP_{TR}), há a possibilidade de existirem colisões subsequentes com outros dispositivos NTR (por exemplo: uma segunda colisão com outra estação NTR). Estas novas colisões são tratadas da mesma maneira que as primeiras. Assim, para que uma estação TR mantenha uma alta prioridade no acesso ao meio mesmo diante de tais colisões, as estações de TR podem efetuar quantas retransmissões forem necessárias até que o *slot* atribuído à TS_i termine ou então que a mensagem de dados seja enviada com sucesso.

Numa situação normal, a transmissão de qualquer mensagem de TR irá terminar antes do EP_i . No entanto, em certas condições (por exemplo, altas cargas da rede) este instante de tempo pode ser excedido. Neste caso específico, a transmissão corrente não é cancelada, mas não serão permitidas novas retransmissões. Esta situação não leva a qualquer conflito com a transmissão que deverá ser efetuada no *slot* seguinte, uma vez que para este efeito o meio é considerado ocupado, impedindo assim o início de qualquer transmissão.

Cada estação TR pode transmitir múltiplas TS com diferentes características. Para cada TS admitida pela ACU é atribuído um índice i (TS_i), sendo $i \in \mathbb{N}^*$. Desta forma, considera-se um grupo G contendo np membros representados por $G = \{TS_1, TS_2, \dots, TS_{np}\}$, onde TS_{np} representa a i -ésima TS admitida. Cada estação somente pode iniciar suas transmissões dentro do período de tempo compreendido entre SP e EP. Caso alguma estação não receba a mensagem de *beacon*, esta não iniciará nenhuma transmissão por considerar que não existem *slots* alocados para si.

Como cada estação TR pode ter uma ou mais TS a transmitir mensagens, a ACU aloca os *slots* de forma independente para cada TS admitida pelo sistema. Além disso, como a arquitetura RT-WiFi considera a sua operação num ambiente de comunicação aberto, o tamanho de cada *slot* é definido pela soma dos tempos necessários para que sejam efetuadas as transmissões *uplink* e *downlink* e também de um tempo extra para que sejam realizadas retransmissões em casos de falha e para que sejam suportados atrasos no início das transmissões em consequência da ocupação do meio de comunicação por estações NTR.

Mesmo com este "superdimensionamento" dos *slots*, em algumas situações, as transmissões podem iniciar-se dentro dos seus respectivos *slots* mas serem finalizadas sobre o *slot* subsequente (Figura 4.5). Esta sobreposição parcial dos *slots* não interfere no correto funcionamento do mecanismo de controlo de acesso ao meio. Isto porque, como não são permitidas retransmissões após EP e, como o AIFS (*Arbitrary Interframe Space*) do AP é menor que o das estações de tempo-real (Figura 4.1), uma estação que tenha o seu *slot* sobreposto irá detetar o meio de comunicação como ocupado e, atrasará o início da sua transmissão evitando assim colisões.

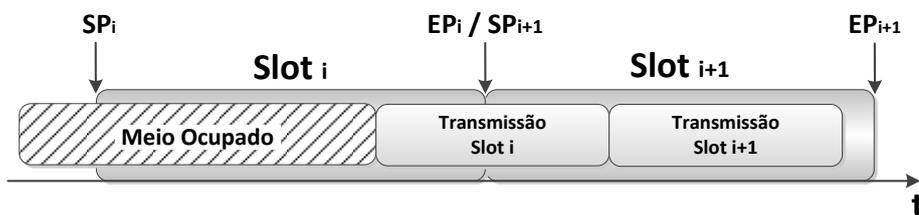


Figura 4.5: Sobreposição parcial dos *slots*.

Se por um lado o "superdimensionamento" dos *slots* auxilia a evitar perdas de *deadlines* em situações onde a ocupação do meio de comunicação é considerada alta, por outro ele pode ser considerado um *overhead* caso a ocupação do meio de comunicação seja considerada baixa. Para amortizar esta situação, a ACU ajusta dinamicamente o tamanho de cada *slot* com base numa

média ponderada do atraso sofrido pela TS para entregar a sua mensagem com sucesso. Este atraso, contado a partir do SP, é a soma do tempo em que o meio de comunicação manteve-se ocupado mais o tempo gasto por transmissões que resultaram em falha. O mecanismo de gestão do tamanho dos *slots* é detalhado na secção 4.3.2.

O "superdimensionamento" dos *slots* permite também que estações que estejam fora de esfera de controlo da arquitetura de tempo-real possam efetuar suas transmissões nos "intervalos" não utilizados dos *slots*. Este "compartilhamento" do meio de comunicação acaba por gerar uma maior equidade (*fairness*) dos dispositivos da rede TR com dispositivos de outras redes NTR.

A lista de escalonamento ($Sched_{List}$) inserida na mensagem de *beacon* é criada pelo algoritmo de escalonamento com base nas informações obtidas pela ACU e contém, para cada TS_i a permissão para transmitir no ciclo corrente, o endereço MAC, o ID da própria estação¹ e os respectivos limites SP e EP (Figura 4.6). O seu conteúdo pode ser alterado a cada ciclo TDMA, garantindo assim uma elevada flexibilidade à arquitetura RT-WiFi.

Lista de Escalonamento			
Addr_STA	TSID	SP	EP
00 : 1b : b1 : 4c : ec : 23	2	0.000150	0.000250
00 : 24 : 54 : ce : c6 : b9	3	0.000250	0.000400
58 : 98 : 35 : 7c : 75 : 3b	2	0.000400	0.000600
00 : 1b : b1 : 4c : ec : 23	1	0.000600	0.000700
08 : 76 : ff : 99 : d0 : 56	3	0.000700	0.000850

Figura 4.6: Exemplo de uma lista de escalonamento enviada na mensagem de *beacon*.

Assim, ao receber uma mensagem de *beacon*, cada estação inicialmente sincroniza seu relógio local com o relógio do AP_{TR} e então realiza uma busca na $Sched_{List}$ (utilizando o seu endereço MAC como índice) para obter os valores de SP_i e EP_i referentes às suas TS. Seguidamente, a estação agenda interrupções com os valores recebidos para sinalizar início e fim de cada *slot*. Uma vez que antes de agendar novas interrupções os relógios foram sincronizados, os valores definidos nas variáveis SP_i e EP_i são descritos em termos absolutos, ou seja, os mesmos do AP_{TR} . Caso alguma estação não receba a mensagem de *beacon*, esta não aloca nenhum *slot*, passando assim um ciclo TDMA sem transmitir. Caso a estação receba a mensagem de *beacon* mas não tenha nenhum *slot* alocado para o ciclo corrente, realiza mesmo assim a sincronização do relógio.

¹O ID local (na estação) de uma TS admitida pode ser diferente do ID atribuído pela ACU. Neste contexto, a nível das estações, as identificações são realizadas sempre pelo tuplo [endereço MAC / ID local].

O algoritmo 1 formaliza a função **SENDBEACON**, invocada pela ACU quando a interrupção periódica que agenda o envio das mensagens de *beacon* é executada. Inicialmente a ACU verifica se entre as TS previamente admitidas, alguma se encontra num estado de "falha". Em caso afirmativos, remove-as da lista de admissão (linha 2). Seguidamente, a ACU calcula os atrasos sofridos pelos fluxos *uplink* e *downlink* de cada TS (linha 3). Estes atrasos são utilizados como base para a redefinição do tamanho do *slot* que será alocado à cada TS (linha 4). Após atualizar o tamanho dos *slots*, a ACU atribui à variável $Sched_{List}$ a nova lista de escalonamento para o ciclo corrente (linha 5) e insere-a na mensagem de *beacon* (linha 6). As funções descritas entre as linhas 2 e 5 são específicas do controlo de admissão e serão detalhadas na secção 4.3.

Por fim, a ACU agenda (no AP_{TR}) as interrupções de início e fim do *slot* para cada TS da $Sched_{List}$ (linhas 8 e 9) e envia a mensagem de *beacon* (linha 11). O objetivo das interrupções é sinalizar à ACU os limites dos *slots* atribuídos à cada TS no ciclo corrente, de forma a permitir que esta possa estimar os atrasos nos fluxos de dados.

Algoritmo 1 Envio da mensagem de *beacon* pelo AP_{TR}

```

1: function SENDBEACON()
2:   CHECKIDLEFLOWS()
3:   COMPUTEDELAY()
4:   REDEFINESLOTLENGTH()
5:    $Sched_{List} \leftarrow$  GETSCHEDLIST()
6:    $BEACON.SCHEDLIST \leftarrow Sched_{List}$ 
7:   for ( $i = 0 \rightarrow Sched_{List}.SIZE - 1$ ) do
8:     SETINTRPTSP[ $Sched_{List}[i].TSID$ ] =  $SchedList[i].SP$ 
9:     SETINTRPTEP[ $Sched_{List}[i].TSID$ ] =  $SchedList[i].EP$ 
10:  end for
11:  SEND.BEACON
12: end function

```

Por outro lado, o algoritmo 2 formaliza os procedimentos realizados quando a mensagem de *beacon* é recebida por uma estação TR. Inicialmente, a estação verifica se a mensagem de *beacon* é proveniente do AP_{TR} (linha 2). Em caso afirmativo, então esta bloqueia a transmissão de novas mensagens (linha 3) e cancela qualquer interrupção agendada que possa ainda não ter sido executada (linha 4).

Algoritmo 2 Receção da mensagem de *beacon* pela estação de tempo-real

```

1: if ( $RXMSG == BEACON$ ) then
2:   if ( $BEACON.BSSID == StationBSSID$ ) then
3:      $RightToSend \leftarrow FALSE$ 
4:     CANCELALLINTRPT()
5:     SYNC( $BEACON.TSF$ )
6:     SEARCHTSID( $BEACON.SCHEDLIST$ )
7:   end if
8: end if

```

Por fim, a estação realiza a sincronização dos relógios (linha 5) e invoca a função **SEARCHTSID** passando como parâmetro a lista de escalonamento recebida na mensagem de *beacon* (linha

6). Esta função têm como objetivo pesquisar na $Sched_{List}$ entradas para as TS previamente submetidas pela estação e aprovadas pela ACU e, caso encontradas, agendar suas respectivas interrupções SP e EP.

A função **SEARCHTSID** é formalizada pelo algoritmo 3, e recebe como parâmetro de entrada a lista de escalonamento enviada pela mensagem de *beacon*. Esta função realiza uma pesquisa na lista de escalonamento até encontrar uma entrada correspondente ao endereço MAC da estação (linha 3). Seguidamente agenda as interrupções SP e EP para a TS em questão (linhas 4 e 5).

Algoritmo 3 Função **SEARCHTSID()**

```

1: function SEARCHTSID(SchedList)
2:   for ( $i = 0 \rightarrow SchedList.SIZE - 1$ ) do
3:     if ( $SchedList[i].Addr\_STA == LocalAddr$ ) then
4:       SETINTRPTSP[ $SchedList[i].TSID$ ]  $\leftarrow SchedList[i].SP$ 
5:       SETINTRPTEP[ $SchedList[i].TSID$ ]  $\leftarrow SchedList[i].EP$ 
6:     end if
7:   end for
8: end function

```

Por fim, o algoritmo 4 formaliza os procedimentos executados pela estação quando a interrupção SP_i é executada. Neste caso, o parâmetro TSID é utilizado para identificar qual TS deve iniciar a transmissão (linha 1). Assim, inicialmente a estação atualiza a variável *RightToSend* (linha 3) para permitir que sejam efetuadas repetidas tentativas de transmissão de uma única mensagem da TS_i (linha 4) até que uma das três seguintes situações ocorra (linha 4): **i**) a estação receba a mensagem de ACK, **ii**) a interrupção do final do *slot* (EP_i) seja executada, **iii**) a variável *RightToSend* seja atualizada para *False*.

Algoritmo 4 Interrupção SP_i

```

1: if ( $EXEC.INTRPTSP[TSID]$ ) then
2:   repeat
3:      $RightToSend \leftarrow TRUE$ 
4:      $SEND.DATAMESSAGE(TSID, 1)$ 
5:   until ( $RECEIPT.ACK[TSID]$  or  $EXEC.INTRPTEP[TSID]$  or  $!RightToSend$ )
6: end if

```

4.3 Mecanismo de Controlo de Admissão

Para evitar que haja uma sobrecarga de tráfego TR, e consequentemente a sua degradação, a arquitetura RT-WiFi implementa um *Mecanismo de Controlo de Admissão* que obriga as estações que queiram transmitir mensagens de TR a solicitarem previamente à ACU a admissão de uma TS. Esta TS identifica um fluxo de dados específico entre uma estação origem e uma estação destino, ou seja, na arquitetura RT-WiFi uma TS é composta pelos fluxos de dados *uplink* e *downlink* das estações comunicantes.

Para realizar o pedido de admissão de uma nova TS_k , a estação deve enviar os respectivos requisitos temporais e características específicas. Com base nestas informações, a ACU efetua

o teste de escalabilidade do grupo G previamente admitido juntamente com a TS_k . Este teste (detalhado na secção 4.3.1) é dado pelo algoritmo de escalonamento em utilização, e seu resultado irá definir entre a admissão (ou não) da TS_k .

O pedido de admissão da TS_k é realizado através do envio de uma requisição ADDTS (*Add Traffic Stream*) ao AP_{TR} contendo as seguintes informações TSPEC:

- período de geração das mensagens (P_k);
- tamanho (L_k) da MPDU (*MAC Protocol Data Unit*);
- intervalo de inatividade (II_k);
- tipo de requisição (CT_k);
- tempo extra de alocação ($SurplusTime_k$).

A arquitetura RT-WiFi implementa três tipos de requisições que podem ser efetuadas pelas estações para a transmissão de tráfegos de TR de **alta** ou **baixa** prioridades:

- **Alta**: solicita a alocação de recursos para a transmissão de tráfegos de **alta** prioridade;
- **Alta/Baixa**: solicita a alocação de recursos para a transmissão de tráfegos de **alta** prioridade. No entanto, caso não seja possível alocar tais recursos nesta categoria, solicita então a alocação de recursos como sendo para a transmissão de tráfegos de **baixa** prioridade;
- **Baixa**: solicita a alocação de recursos para a transmissão de tráfegos de **baixa** prioridade;

O tempo extra de alocação (definido pelo parâmetro $SurplusTime_k$) solicitado pela estação refere-se ao tempo adicional (além do necessário para a transmissão com sucesso de uma única mensagem de dados) que é solicitado à ACU alocar para a TS_k . Para simplificar os cálculos, este valor é baseado no número de retransmissão que cada fluxo de dados (*uplink* e *downlink*) pode efetuar caso não ocorram atrasos no acesso ao meio. Desta forma, o valor de $SurplusTime_k$ é dado pela seguinte equação:

$$SurplusTime_k = (C_{attempt}^{uplink} \times RN_k^{uplink}) + (C_{attempt}^{downlink} \times RN_k^{downlink}) \quad (4.3)$$

onde RN_k^{uplink} e $RN_k^{downlink}$ representam o número de retransmissões utilizada pela TS_k como base de cálculo para cada fluxo de dados, respectivamente². Os parâmetros $C_{attempt}^{uplink}$ e $C_{attempt}^{downlink}$ são os tempos necessários para efetuar as transmissões com sucesso de uma única mensagem de dados nos fluxos *uplink* e *downlink*, respectivamente. Os respectivos valores são dados pelas seguintes equações:

$$C_{attempt}^{uplink} = AIFS_{VO}^{QSTA} + C_{DATA}[L_k] + SIFS + C_{ACK} \quad (4.4)$$

²Os parâmetros RN_k^{uplink} e $RN_k^{downlink}$ podem ser configurados pelo utilizador.

$$C_{attempt}^{downlink} = AIFS_{VO}^{QAP} + C_{DATA}[L_k] + SIFS + C_{ACK} \quad (4.5)$$

onde $AIFS_{VO}^{QSTA}$ é o AIFS da categoria de voz na estação TR, $AIFS_{VO}^{QAP}$ é o AIFS da categoria de voz no AP_{TR} , C_{DATA} é o tempo necessário para se realizar a transmissão de uma única mensagem de dados com MPDU de tamanho L_k e C_{ACK} é o tempo necessário para se realizar a transmissão da mensagem de ACK. Todos estes tempos são baseados nos valores definidos pela camada física.

Neste contexto, o envio da requisição ADDTS é formalizado pelo algoritmo 5. Esta função recebe como parâmetros o endereço MAC da estação TR ($Addr_STA$), o id da TS_k ($TSID$), o período de geração das mensagens (P_k), o tamanho do MPDU (L_k), o intervalo de inatividade (Π_k) e o tipo de requisição (CT_k), sendo os cinco primeiros enviados pelo $TSPEC$ e o último pelo elemento $TCLAS$ (*Traffic Classification*), ambos definidos na norma IEEE 802.11.

Algoritmo 5 Envio da requisição ADDTS ao AP_{TR} .

```

1: function SEND_ADDTSREQUEST( $Addr\_STA$ ,  $TSID$ ,  $P_k$ ,  $L_k$ ,  $\Pi_k$ ,  $CT_k$ )
2:    $C_{attempt}^{uplink} = AIFS_{VO}^{QSTA} + C_{DATA}[L_k] + SIFS + C_{ACK}$ 
3:    $C_{attempt}^{downlink} = AIFS_{VO}^{QAP} + C_{DATA}[L_k] + SIFS + C_{ACK}$ 
4:    $SurplusTime_k = (C_{attempt}^{uplink} \times RN_k^{uplink}) + (C_{attempt}^{downlink} \times RN_k^{downlink})$ 
5:   SENDMSG.ADDTSRequest[ $Addr\_STA$ ,  $TSID$ ,  $P_k$ ,  $L_k$ ,  $\Pi_k$ ,  $CT_k$ ,  $SurplusTime_k$ ]
6:   while (!TIMEOUT.ADDTSRequest) do
7:     if (RECEIPT.ADDTSResponse) then
8:       if (ADDTSResponse.STATUS = Accepted) then
9:         INSERT.AdmittedList( $TSID$ , ADDTSResponse.CT)
10:      else
11:        DELTS( $TSID$ )
12:      end if
13:    EXIT()
14:  end if
15: end while
16: end function

```

Inicialmente a estação calcula o tempo necessário para a transmissão de uma mensagem de dados com sucesso para ambos os fluxos de dados (*uplink* e *downlink*) (linhas 2 e 3, respectivamente) e calcula também o valor da variável *SurplusTime* (linha 4). Seguidamente, a estação insere a variável *SurplusTime* juntamente com os restantes parâmetros recebidos pela função na requisição ADDTS, a qual é então enviada ao AP_{TR} (linha 5). Quando recebe a mensagem de resposta ADDTS, a estação verifica se a sua solicitação foi aceite ou não (linha 8). Em caso afirmativo, a estação insere a TS na sua lista de admissão utilizando o tipo de classificação definida pela ACU na mensagem de resposta ADDTS (linha 9), caso contrário a TS é eliminada (linha 11).

Ao receber a requisição ADDTS da TS_k , a ACU calcula o tempo necessário para a transmissão de uma única mensagem de dados de tamanho L_k nos fluxos *uplink* (equação 4.4) e *downlink* (equação 4.5). Seguidamente, a ACU define um tempo de atraso (*Interf_k*) que a estação pode sofrer antes de efetivamente iniciar suas tentativas de acesso ao meio de comunicação. Este atraso pode ser resultante ou da ocupação do meio de comunicação por estações que se encontram fora

da esfera de controlo da arquitetura RT-WiFi, ou pela sobreposição parcial de *slots*³. O valor do parâmetro *Interf* é dado pela seguinte equação:

$$Interf_k = C_{DATA}[MPDU^{max}] + SIFS + C_{ACK} \quad (4.6)$$

onde $C_{DATA}[MPDU^{max}]$ é o tempo necessário para a transmissão de uma mensagem contendo um MPDU de tamanho máximo⁴.

Uma vez que a ACU pode modificar dinamicamente o tamanho dos *slots* alocados para a TS_k (procedimento detalhado na secção 4.3.2), para evitar o seu crescimento excessivo, esta define um tamanho máximo que os *slots* podem atingir (C_k^{max}), que é dado pela seguinte equação:

$$C_k^{max} = (2 \times Interf_k) + C_{attempt}^{uplink} + C_{attempt}^{downlink} + SurplusTime_k \quad (4.7)$$

onde o valor de *Interf* é multiplicado por 2 para considerar as interferências que ambos os fluxos de dados *uplink* e *downlink* possam sofrer.

O algoritmo 6 formaliza o processamento de uma requisição ADDTS recebida pela ACU. Esta função recebe como parâmetros de entrada os valores TSPEC enviados pela estação para a admissão de uma TS_k . Inicialmente a ACU calcula o tempo necessário para a transmissão de uma mensagem de dados (com sucesso) para ambos os fluxos de dados *uplink* e *downlink* (linhas 2 e 3, respectivamente) e também o valor da variável $Interf_k$ (linha 4). Seguidamente, a ACU calcula o tamanho máximo do *slot* (linha 5) e invoca a função **SCHEDTEST** para realizar o teste de escalonabilidade (linha 6).

Caso os parâmetros inseridos no teste sejam escalonáveis (linha 7), então a ACU define como tamanho atual do *slot* ($C_k^{current}$) o tamanho utilizado no teste de escalonabilidade (linha 8) e atualiza a variável CT_k para qual a prioridade a TS_k foi admitida (linha 9). Se esta foi admitida como sendo de **alta** prioridade (linha 10), então a ACU agenda interrupções subsequentes com um intervalo de tempo definido pela variável P_k para a sua inserção na "lista de pronto" de TS de **alta** prioridade (linha 11). Caso contrário, agenda interrupções subsequentes com um intervalo de tempo definido pela variável P_k para a sua inserção na "lista de pronto" de TS de **baixa** prioridade (linha 13).

Uma vez definida a respectiva prioridade, a ACU insere a TS_k e seus respectivos parâmetros numa lista de TS admitidas (linha 15) e envia uma mensagem de resposta ADDTS à estação contendo a informação acerca da admissão e para qual prioridade esta foi admitida (linha 16). Caso o resultado do teste de escalonabilidade seja negativo, então a ACU envia uma mensagem de resposta ADDTS informando a estação acerca da impossibilidade de admissão da TS_k (linha 18).

Uma vez admitida, a remoção de uma TS (seja na ACU ou na estação) pode ocorrer em duas situações: **i**) por solicitação da própria estação, **ii**) por imposição da ACU. Em ambos os casos a função **DELTS** é invocada.

³A sobreposição parcial de *slots* ocorre quando a TS_i de uma estação inicia a sua transmissão instantes antes do final do seu *slot* (EP_i) e esta é prolongada até uma parte inicial do *slot* da TS_{i+1} .

⁴O tamanho máximo de um MPDU definido pela norma IEEE 802.11 é de 2304 bytes.

Algoritmo 6 Processamento da requisição ADDTS pela ACU.

```

1: function PROCESS_ADDTSREQUEST(Addr_STA, TSID, Pk, Lk, Πk, CTk, SurplusTimek)
2:   Cattemptuplink ← AIFSVOQSTA + CDATA[Lk] + SIFS + CACK
3:   Cattemptdownlink ← AIFSVOQAP + CDATA[Lk] + SIFS + CACK
4:   Interfk ← CDATA[MSDUmax] + SIFS + CACK
5:   Ckmax ← (2 × Interfk) + Cattemptuplink + Cattemptdownlink + SurplusTimek
6:   SCHEDTEST(Ckmax, Pk, CTk)
7:   if (SCHEDTEST.STATUS == SCHEDULABLE) then
8:     Ckcurrent ← Ckmax
9:     CTk ← SCHEDTEST.CT
10:    if (CTk == HIGH) then
11:      SETINTRPTLOOPINSERTREADYLIST(ReadyListHigh, Pk)
12:    else
13:      SETINTRPTLOOPINSERTREADYLIST(ReadyListLow, Pk)
14:    end if
15:    AdmittedList.INSERT(Addr_STA, TSID, Pk, Πk, CTk, Ckcurrent, Ckmax, Cattemptuplink, Cattemptdownlink)
16:    SENDMSG.ADDTSResponse[Accepted, CTk]
17:  else
18:    SENDMSG.ADDTSResponse[Denied, NULL]
19:  end if
20: end function

```

No primeiro caso, a estação invoca a função **DELTS** localmente (removendo a TS da sua lista local) e envia uma requisição DELTS ao AP_{TR} que, ao receber esta mensagem, invoca também a função **DELTS** (removendo a TS da lista de admissão da ACU).

No segundo caso, inverte-se a situação, a ACU invoca a função **DELTS** localmente (removendo a TS da lista de admissão da ACU) e envia uma requisição DELTS à estação da TS impondo a sua remoção. Esta, por sua vez, invoca a função **DELTS** para remover a TS da sua lista local.

A função **DELTS** (formalizada pelo algoritmo 7) recebe como parâmetro de entrada o identificador local da TS (TSID). Com base neste valor é feita uma pesquisa na lista de TS admitidas (linha 2) até que seja encontrada a entrada correspondente (linha 3) e então é removida a entrada para a TS (linha 4).

Algoritmo 7 Função DELTS()

```

1: function DELTS(TSID)
2:   for (i = 0 → AdmittedList.SIZE - 1) do
3:     if (AdmittedList[i].TSID == TSID) then
4:       AdmittedList[i].REMOVE
5:     end if
6:   end for
7: end function

```

4.3.1 Teste de Escalonabilidade

A arquitetura RT-WiFi foi implementada de forma a possibilitar a utilização de diferentes algoritmos de escalonamento sem a necessidade de uma reestruturação completa. Assim, é apenas

necessário alterar o algoritmo em questão e conseqüentemente, o respectivo teste de escalonabilidade, uma vez que este está intrinsicamente ligado ao algoritmo utilizado. No modelo proposto, foi adotado o algoritmo de escalonamento *Rate Monotonic* (RM) [84].

O RM é um algoritmo de escalonamento de tempo-real pertencente à classe de algoritmos de prioridade estática, em que a prioridade é dada com base no período de geração da tarefa que deve ser escalonada. Assim, quanto menor este período maior será a prioridade da tarefa.

A opção pelo algoritmo RM está relacionada com o seu baixo *overhead* de processamento no dispositivo escalonador. Desta forma, a criação de uma nova lista de escalonamento é apenas necessária quando uma nova TS é admitida. A utilização de um algoritmo de escalonamento de prioridade dinâmica, como o EDF (*Earliest Deadline First*), resultaria na necessidade de recriar uma nova lista de escalonamento a cada ciclo TDMA executado, o que conseqüentemente elevaria os requisitos computacionais do dispositivo escalonador.

Neste contexto, após receber uma requisição ADDTS e calcular o tamanho do *slot* solicitado pela TS_k , a ACU calcula o nível de utilização (U_k) que esta TS irá impor a rede se for admitida. Seu valor é dado pela seguinte equação:

$$U_k = \frac{C_k^{max}}{P_k} \quad (4.8)$$

onde C_k^{max} é o tamanho máximo do *slot* definido pela ACU ao receber a requisição ADDTS e P_k é o período de geração das mensagens da TS_k .

Seguidamente, a ACU calcula o nível de utilização gerado pela mensagem de *beacon* (U_{beacon}) caso a TS_k seja aceite. Desta forma, leva-se em consideração a inserção de uma nova TS na $Sched_{List}$. O valor de U_{beacon} é dado pela seguinte equação:

$$U_{beacon} = \frac{C_{beacon}[Sched_{List}^{np+1}]}{BI} \quad (4.9)$$

onde $C_{beacon}[Sched_{List}^{np+1}]$ é o tempo necessário para a transmissão da mensagem de *beacon* com uma lista de escalonamento ($Sched_{List}$) contendo $np + 1$ elementos e BI (*Beacon Interval*) é o período de geração da mensagem de *beacon*.

Assim, é realizado um teste de escalonabilidade baseado na utilização do sistema. Como no caso específico deste trabalho foi utilizado o algoritmo de escalonamento RM, antes da aplicação do teste de escalonabilidade a ACU verifica se o período de geração das mensagens das TS previamente admitidas, da TS_k e da mensagem de *beacon* são harmônicos. Em caso afirmativo, o limite para a utilização é dado pelo valor 1 (equivalente a 100%), caso contrário, este limite é dado por $n \times (2^{1/n} - 1)$ [84], onde n é o número total de TS que precisam ser transmitidas, ou seja, todas as np TS previamente admitidas pela ACU, mais a TS_k e a mensagem de *beacon* ($n = np + 2$). Desta forma, a nova TS_k somente será admitida se a seguinte inequação for respeitada:

$$U_{beacon} + \sum_{i=1}^{np} U_i + U_k \leq \begin{cases} 1 & \text{se o conjunto G for harmônico} \\ n \times (2^{1/n} - 1) & \text{caso contrário} \end{cases} \quad (4.10)$$

O teste de escalabilidade é formalizado no algoritmo 8. Esta função é invocada pela ACU ao receber uma requisição ADDTS. São utilizados como parâmetros de entrada o tamanho máximo do *slot* (C_k^{max}), o período de geração das mensagens (P_k) e o tipo de requisição que a TS_k foi submetida (CT_k).

O primeiro passo consiste em inicializar uma variável auxiliar denominada *CheckLow* (linha 2), que é utilizada para controlar o fluxo de execução do algoritmo para **baixa** prioridade quando uma requisição do tipo **alta/baixa** não é aceite como sendo de *alta* prioridade.

Seguidamente, a ACU executa uma função que, para uma dada prioridade, retorna o número de TS admitidas e a soma das seus respectivos níveis de utilização (linhas 3 – 5). É importante notar que o nível de utilização retornado por esta função para as TS de **alta** prioridades baseia-se no tamanho máximo do *slot* (C_k^{max}), diferentemente do nível de utilização retornado para as TS de **baixa** prioridade, que se baseia no tamanho atual do *slot* ($C_k^{current}$).

Algoritmo 8 Função SchedTest()

```

1: function SCHEDTEST( $C_k^{max}$ ,  $P_k$ ,  $CT_k$ )
2:   CheckLow  $\leftarrow$  FALSE
3:   COMPUTEVALUES(HIGH)
4:    $n^{High}$   $\leftarrow$  COMPUTEVALUES.N
5:    $Sum^{High}$   $\leftarrow$  COMPUTEVALUES.U
6:   if (( $CT_k$  == HIGH) or ( $CT_k$  == HIGH/LOW)) then
7:      $n \leftarrow 2 + n^{High}$ 
8:      $Sum \leftarrow$  (Beacon.SIZE / BI) +  $Sum^{High}$  + ( $C_k^{max}$  /  $P_k$ )
9:     if (ISHARMONIC( $P_k$ ) and ( $Sum \leq 1$ )) then
10:      return (SCHEDULABLE, HIGH)
11:     else if (!ISHARMONIC( $P_k$ ) and ( $Sum \leq n \times (2^{1/n} - 1)$ )) then
12:      return (SCHEDULABLE, HIGH)
13:     else
14:       if ( $CT_k$  == HIGH) then
15:         return (NONSCHEDULABLE, NULL)
16:       else
17:         CheckLow  $\leftarrow$  TRUE
18:       end if
19:     end if
20:   end if
21:   if (( $CT_k$  == LOW) or (CheckLow == TRUE)) then
22:     COMPUTEVALUES(LOW)
23:      $n^{Low}$   $\leftarrow$  COMPUTEVALUES.N
24:      $Sum^{Low}$   $\leftarrow$  COMPUTEVALUES.U
25:      $n \leftarrow 2 + n^{Low}$ 
26:      $Sum \leftarrow$  (Beacon.SIZE / BI) +  $Sum^{Low}$  + ( $C_k^{max}$  /  $P_k$ )
27:     CF  $\leftarrow$  COMPUTECOMPRESSIONFACTOR()
28:     if CHECKHARMONIC( $P_k$ ) and ( $Sum \leq 1 - (CF \times Sum^{High})$ ) then
29:       return (SCHEDULABLE, LOW)
30:     else if !CHECKHARMONIC( $P_k$ ) and ( $Sum \leq (n \times (2^{1/n} - 1) - (CF \times Sum^{High}))$ ) then
31:       return (SCHEDULABLE, LOW)
32:     else
33:       return (NONSCHEDULABLE, NULL)
34:     end if
35:   end if
36: end function

```

Após calcular o nível de utilização das TS de **alta** prioridade, a ACU verifica o tipo de requisição solicitada. Se esta for do tipo **alta** ou **alta/baixa** (linha 6), então é definido o número total de TS e o nível total de utilização que devem ser considerados pelo teste de escalabilidade, incluindo a mensagem de *beacon* e a nova TS (linhas 7 e 8). Posteriormente é executado o teste de escalabilidade levando em consideração se o conjunto de TS têm períodos de geração harmônicos, ou não (linhas 9 e 11, respectivamente). Caso o resultado do teste seja positivo, a função retorna o estado de "escalonável" e a categoria para qual a TS foi aceite (linhas 10 e 12). Caso contrário, é verificado qual o tipo de requisição que foi solicitada. No caso desta ser de **alta** prioridade (linha 14), retorna o estado de "não-escalonável" (linha 15), caso contrário, a ACU redefine o valor da variável *CheckLow* (linha 17).

O passo seguinte consiste em verificar se a requisição é do tipo **baixa** ou se a variável *CheckLow* está definida como *True* (linha 21). Em caso afirmativo, então a ACU executa novamente a função que retorna o número de TS admitidas e a soma das seus respectivos níveis de utilização (linhas 22 – 24). Seguidamente, a ACU define o número total de TS e o nível total de utilização que devem ser considerados pelo teste de escalabilidade, incluindo a mensagem de *beacon* e a nova TS (linhas 25 e 26) e executa uma função que retorna o fator de compressão dos *slots* pertencentes às TS de **alta** prioridade já admitidas pelo sistema (linha 27). Este fator define, em termos relativos, a diferença entre o tamanho máximo e o tamanho atual dos *slots* de **alta** prioridade. Por fim é executado o teste de escalabilidade levando em consideração se o conjunto de TS têm períodos de geração harmônicos e descontando (através do fator de compressão) o nível de utilização atualmente imposto pelas TS de **alta** prioridade (linhas 28 e 30, respectivamente). Caso o resultado do teste seja positivo, a função retorna o estado de "escalonável", a categoria para qual a TS foi aceita e o tamanho do *slot* (linhas 29 e 31). Caso contrário, a função retorna o estado de "não-escalonável" (linha 33).

A função **COMPUTEVALUES** é formalizada no algoritmo 9, e tem como objetivo obter o número e o nível de utilização das TS admitidas numa dada prioridade. Importa referir que o nível de utilização retornado por esta função para as TS de **alta** prioridade baseia-se no tamanho máximo do *slot* (C^{max}), enquanto que para as TS de **baixa** prioridade este valor baseia-se no tamanho atual do *slot* ($C^{current}$). Neste contexto, são inicializadas duas variáveis auxiliares (linhas 2 e 3), e em seguida é realizada uma pesquisa na lista de TS admitidas utilizando como base a prioridade passada como parâmetro de entrada (linhas 4 e 5). Caso seja encontrada alguma entrada correspondente, então a variável n é incrementada (linha 6) e a variável U somada ao nível de utilização da TS encontrada utilizando como base a regra previamente descrita para as diferentes prioridades (linhas 7 – 11). Por fim, os valores de ambas as variáveis são retornados à função anterior (linha 14).

Algoritmo 9 Função COMPUTEVALUES()

```

1: function COMPUTEVALUES(Priority)
2:    $n \leftarrow 0$ 
3:    $U \leftarrow 0$ 
4:   for ( $i = 0 \rightarrow AdmittedList.SIZE - 1$ ) do
5:     if ( $AdmittedList[i].CT == Priority$ ) then
6:       INC( $n$ )
7:       if ( $Priority == HIGH$ ) then
8:          $U \leftarrow U + (AdmittedList[i].C^{max} / AdmittedList[i].P)$ 
9:       else
10:         $U \leftarrow U + (AdmittedList[i].C^{current} / AdmittedList[i].P)$ 
11:      end if
12:    end if
13:  end for
14:  return ( $n, U$ )
15: end function

```

A função **COMPUTECOMPRESSIONFACTOR** é formalizada no algoritmo 10 e tem como objetivo definir o fator de compressão (CF) dos *slots* alocados às TS de **alta** prioridade. Neste contexto, são inicializadas duas variáveis auxiliares (linhas 2 e 3) e posteriormente é realizada uma pesquisa na lista de admissão (linha 4) pelas TS de **alta** prioridade (linha 5). Para cada entrada encontrada, os valores do tamanho máximo do *slot* (C^{max}) e o tamanho corrente do *slot* ($C^{current}$) são acumulados (linhas 6 e 7, respectivamente). Por fim é realizado o cálculo do fator de compressão (linha 8) para verificar a diferença entre o tamanho máximo e o tamanho corrente dos *slots*. Este valor é retornado pela função (linha 11).

Algoritmo 10 Função COMPUTECOMPRESSIONFACTOR()

```

1: function COMPUTECOMPRESSIONFACTOR
2:    $Sum\_C^{max} \leftarrow 0$ 
3:    $Sum\_C^{current} \leftarrow 0$ 
4:   for ( $i = 0 \rightarrow AdmittedList.SIZE - 1$ ) do
5:     if ( $AdmittedList[i].CT == HIGH$ ) then
6:        $Sum\_C^{max} \leftarrow Sum\_C^{max} + AdmittedList[i].C^{max}$ 
7:        $Sum\_C^{current} \leftarrow Sum\_C^{current} + AdmittedList[i].C^{current}$ 
8:        $CF \leftarrow Sum\_C^{current} / Sum\_C^{max}$ 
9:     end if
10:  end for
11:  return ( $CF$ )
12: end function

```

Além de realizar o teste de escalonabilidade, a ACU é também responsável por manter duas listas de TS que se encontram no estado de "pronto" para serem transmitidas. Estas listas, para TS de **alta** e **baixa** prioridade (*ReadyListHigh* e *ReadyListLow*, respectivamente), são utilizadas pela função **GETSCHEDLIST** invocada no algoritmo 1.

Neste contexto, a função **GETSCHEDLIST** é formalizada no algoritmo 11 e tem como finalidade retornar a lista de escalonamento (para o ciclo TDMA atual) que será inserida na mensagem de *beacon*. Neste contexto, a ACU inicializa algumas variáveis auxiliares (linhas 2 – 6), em especial a variável *Budget* (que define quanto tempo há disponível no ciclo atual para transmissão)

e a variável *ComputingList* (que define qual a lista de "pronto" utilizada no momento). Seguidamente, a ACU inicia um ciclo de repetição (linha 7) que irá primeiramente avaliar as entradas na lista de **alta** prioridade (*ReadyListHigh*), para posteriormente avaliar as entradas na lista de **baixa** prioridade (*ReadyListLow*).

Para cada lista de "pronto" avaliada, é verificado se existe pelo menos 1 entrada registada (linha 8). Em caso afirmativo, a lista em questão é percorrida (linha 9) e para cada entrada é calculado o tamanho da mensagem de *beacon* tendo em conta a inserção de uma nova entrada na lista de escalonamento (linha 10) para então calcular o seu tempo de transmissão (linha 11).

Posteriormente, é verificado se o recurso disponível do sistema (descontando o tempo utilizado para a transmissão da mensagem de *beacon*) é suficientemente grande para a inserção do novo *slot* (linha 12). Se existirem recursos suficientes, então a ACU define os instantes de tempo SP e EP (linhas 13 e 14) e insere as informações necessárias na lista a enviar na mensagem de *beacon* (linha 15). Uma vez terminada a verificação da lista de **alta** prioridade, o processo é repetido para a lista de **baixa** prioridade.

Algoritmo 11 Função **GETSCHEDLIST()**

```

1: function GETSCHEDLIST()
2:    $N \leftarrow 1$ 
3:    $SlotSum \leftarrow 0$ 
4:    $Time \leftarrow \text{CURRENTTIME}()$ 
5:    $Budget \leftarrow BI$ 
6:    $ComputingList \leftarrow ReadyListHigh$ 
7:   for ( $i = 0 \rightarrow 1$ ) do
8:     if ( $ComputingList.SIZE > 0$ ) then
9:       for ( $i = 0 \rightarrow ComputingList.SIZE - 1$ ) do
10:         $BeaconSize \leftarrow \text{DEFAULTBEACONSIZE} + (N \times \text{ONERTBEACONINFOSIZE})$ 
11:         $C^{beacon} \leftarrow AIFS_{VO}^{QAP} + SIFS + C_{MNG}[BeaconSize]$ 
12:        if ( $(Budget - C^{beacon}) \geq ComputingList[i].C^{current}$ ) then
13:           $SP \leftarrow Time + SlotSum$ 
14:           $EP \leftarrow SP + ComputingList[i].C^{current}$ 
15:           $List.INSERT(ComputingList[i].Addr\_STA, ComputingList[i].TSID, SP, EP)$ 
16:           $SlotSum \leftarrow SlotSum + ComputingList[i].C^{current}$ 
17:           $Budget \leftarrow Budget - ComputingList[i].C^{current}$ 
18:           $INC(N)$ 
19:        end if
20:      end for
21:    end if
22:     $ComputingList \leftarrow ReadyListLow$ 
23:  end for
24:   $BeaconSize \leftarrow \text{DEFAULTBEACONSIZE} + ((N - 1) \times \text{ONERTBEACONINFOSIZE})$ 
25:   $C^{beacon} \leftarrow AIFS_{VO}^{QAP} + SIFS + C_{MNG}[BeaconSize]$ 
26:  for ( $i = 0 \rightarrow List.SIZE - 1$ ) do
27:     $List[i].SP \leftarrow List[i].SP + C^{beacon}$ 
28:     $List[i].EP \leftarrow List[i].EP + C^{beacon}$ 
29:  end for
30:  return ( $List$ )
31: end function

```

É importante referir que só no final deste processo a ACU saberá quantas entradas ficarão

inseridas na lista de escalonamento. Assim, é necessário realizar um ajuste nos instantes SP e EP previamente definidos para que estes parâmetros considerem também o tempo de transmissão da mensagem de *beacon* já com seu tamanho final (linhas 24 – 29).

4.3.2 Gestão do Tamanho dos Slots

Ao solicitar a admissão de uma TS_k , a estação TR deve enviar ao AP_{TR} (via TSPEC) o tempo extra ($SurplusTime_k$ - equação 4.3) que deve ser alocado pela ACU para que sejam suportadas eventuais retransmissões e/ou atrasos no acesso ao meio devido a sua ocupação. Neste contexto, pode-se concluir que em algumas situações (dependendo do nível de utilização e ruído do meio de comunicação) o tamanho do *slot* alocado é superior ao realmente necessário para o envio (com sucesso) de uma mensagem de dados.

Para "otimizar" o tamanho dos *slots* alocados por cada TS_i admitida, a ACU realiza o seu redimensionamento com base num estimador que, a cada ciclo TDMA, mede o atraso sofrido pelos fluxos de dados *uplink* e *downlink* (b_i^{uplink} e $b_i^{downlink}$, respectivamente) para finalizarem as suas respectivas transmissões com sucesso. Este atraso consiste, para cada fluxo de dados, no tempo adicional gasto pela estação além do tempo necessário para se transmitir com sucesso uma única mensagem de dados. Este valor é obtido pela soma de dois atrasos: **i**) atraso no acesso ao meio (contado a partir do início do *slot*) resultante da ocupação deste por outras estações, e **ii**) atraso gerado por retransmissões, sejam estas resultantes de colisões ou de erros de transmissão gerados por problemas no canal de comunicação (ex: *fading*, *multi-path* e EMI – *Electromagnetic Interference*).

É importante ressaltar que como o estimador utilizado pela ACU mede o atraso sofrido pelas TS no ciclo TDMA anterior, situações onde possam ocorrer variações repentinas nestas medições podem resultar num comportamento oscilatório do mecanismo de redimensionamento. *red*Isto resultaria no simples reflexo de comportamentos anteriores, e não na criação de uma tendência futura das transmissões. Como estratégia para evitar estas variações, faz-se necessário o uso de um mecanismo capaz de suavizá-las. Além disso, a necessidade de avaliar o atraso sofrido por ambos os fluxos de dados deve-se a dois factores: **i**) diferenças de configuração entre as estações TR e o AP_{TR} (nomeadamente os valores de IFS), e **ii**) localização de cada dispositivo no ambiente de comunicação, o qual pode resultar em diferentes atrasos, sejam estes resultantes da sobreposição de diferentes redes (a gerar atrasos de acesso ao meio e colisões) ou do próprio ambiente físico onde o dispositivo está localizado (podendo este sofrer diferentes níveis de ruído no canal).

Como a arquitetura RT-WiFi é baseada numa topologia infraestruturada, isto permite que a ACU possa verificar a origem das transmissões efetuadas dentro dos limites de cada *slot*. Uma vez que a ACU conhece também o tempo necessário para que cada TS admitida pelo sistema possa efetuar a transmissão (com sucesso) de uma única mensagem de dados, então esta é capaz de calcular o tempo b_i^{uplink} de uma TS_i medindo o tempo entre o início do seu respectivo *slot* (SP_i) e o início da transmissão de uma mensagem de dados finalizada com sucesso (Figura 4.7).

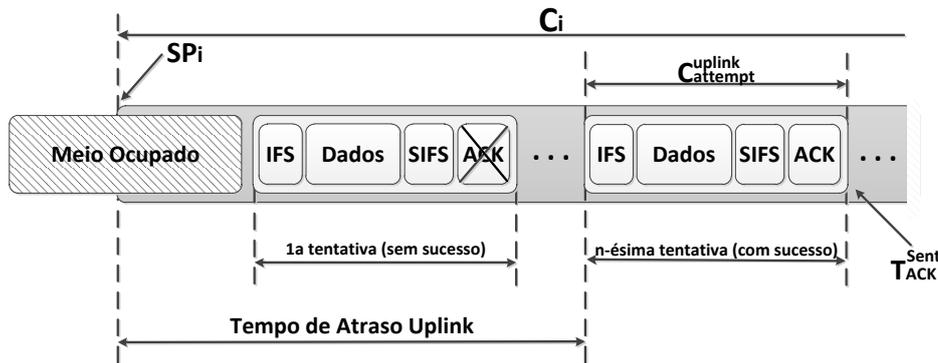


Figura 4.7: Tempo de atraso *uplink* (b_i^{uplink}).

Neste contexto, o valor de b_i^{uplink} é dado pela seguinte equação:

$$b_i^{uplink} = T_{ACK}^{Sent} - SP_i - C_{attempt}^{uplink} \quad (4.11)$$

onde T_{ACK}^{Sent} é o instante de tempo onde a mensagem de ACK é enviada pelo AP_{TR} à estação origem, SP_i é o instante de tempo descrito como o início do *slot* atribuído à TS_{*i*} e $C_{attempt}^{uplink}$ o tempo necessário para se realizar a transmissão de uma mensagem de dados (com sucesso) da estação origem ao AP_{TR}.

A topologia infraestruturada permite também que o tempo $b_i^{downlink}$ de uma TS_{*i*} possa ser calculado pela ACU com base no tempo necessário para que o AP_{TR} reencaminhe (com sucesso) a mensagem recebida à estação destino. Este tempo inicia-se logo após o envio da mensagem de ACK para a estação origem e termina no início de uma transmissão com sucesso para a estação destino (Figura 4.8). É importante referir que, embora a transmissão do fluxo de dados *downlink* seja procedida logo após a recepção da mensagem pelo AP_{TR}, a sua transmissão para a estação destino pode sofrer atrasos devido a estações ocultas (*hidden-nodes*), ou seja, estações que estão fora da área de cobertura da estação origem.

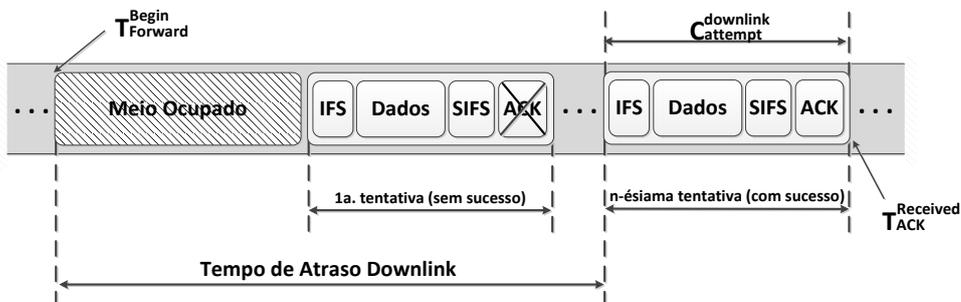


Figura 4.8: Tempo de atraso *downlink* ($b_i^{downlink}$).

Neste caso, o valor de $b_i^{downlink}$ é dado pela seguinte equação:

$$b_i^{downlink} = T_{ACK}^{Received} - T_{Forward}^{Begin} - C_{attempt}^{downlink} \quad (4.12)$$

onde $T_{ACK}^{Received}$ é o instante de tempo onde a mensagem de ACK da estação destino é recebida pelo AP_{TR}, $T_{Forward}^{Begin}$ é o instante de tempo onde o AP_{TR} inicia o reencaminhamento da mensagem para a estação destino (ou seja, logo após o envio da mensagem de ACK à estação origem) e $C_{attempt}^{downlink}$ o tempo necessário para se realizar a transmissão de uma única mensagem de dados do AP_{TR} à estação destino.

O algoritmo 12 formaliza a função **MEASUREDELAY**, que é invocada pela ACU quando a interrupção agendada no instante SP_i de cada *slot* é activada. Esta função recebe como parâmetro de entrada o identificador da TS que deve monitorizar (TSID) e tem como objetivo estimar os atrasos sofridos em ambos os fluxos de dados.

Algoritmo 12 Interrupção de avaliação dos atrasos de *uplink/downlink*

```

1: function MEASUREDELAY(TSID)
2:   repeat
3:     if (SENT.ACK[TSID]) then
4:        $b^{uplink} \leftarrow T_{ACK}^{Sent} - T^{SP} - AdmittedList[TSID].C_{attempt}^{uplink}$ 
5:        $AdmittedList[TSID].MsgUpRcvd \leftarrow TRUE$ 
6:        $AdmittedList[TSID].IdleTime \leftarrow 0$ 
7:       EXIT()
8:     end if
9:     until (INTRPTEP[TSID])
10:    if (! $AdmittedList[TSID].MsgUpRcvd$ ) then
11:       $b^{uplink} \leftarrow AdmittedList[TSID].C^{current}$ 
12:      INC( $AdmittedList[TSID].IdleTime$ )
13:    else
14:      repeat
15:        if (RECEIPT.ACK.MSGDEST[TSID]) then
16:           $b^{downlink} \leftarrow T_{ACK}^{Receipt} - T_{Forward}^{Begin} - AdmittedList[TSID].C_{attempt}^{downlink}$ 
17:           $AdmittedList[TSID].MsgDownSent \leftarrow TRUE$ 
18:          EXIT()
19:        end if
20:      until (INTRPTEP[TSID])
21:      if (! $AdmittedList[TSID].MsgDownSent$ ) then
22:         $b^{downlink} \leftarrow AdmittedList[TSID].C^{current} - (b^{uplink} + AdmittedList[TSID].C_{attempt}^{uplink})$ 
23:      end if
24:    end if
25:    COMPUTEDELAY(TSID,  $b^{uplink}$ ,  $b^{downlink}$ )
26:  end function

```

Após o início da interrupção SP_i , a ACU monitoriza a sua fila de transmissão até ao envio da mensagem de ACK para a estação da TSID em questão, ou até a execução da interrupção EP_i . Caso a mensagem de ACK seja enviada (linha 3), então a ACU calcula o atraso sofrido pela estação para o envio do tráfego *uplink* (linha 4) e atualiza duas variáveis contidas na lista de TS admitidas pelo sistema (linhas 5 e 6). A primeira variável (*MsgUpRcvd*), indica ao sistema que o fluxo *uplink* foi recebido com sucesso (linha 5) enquanto que a segunda (*IdleTime*) reinicializa o contador que

controla o tempo máximo ao qual uma TS pode ficar sem transmitir (linha 6). Este contador será utilizado posteriormente para controlar a exclusão das TS que se encontrem num estado de "falha" (detalhado na seção 4.3.3). Caso a mensagem do fluxo *uplink* não tenha sido recebida com sucesso, significa que a interrupção EP_i foi executada e que as variáveis previamente referidas não foram atualizadas (linha 10). Neste caso a ACU define como atraso do fluxo *uplink* o tamanho atual do seu *slot* (linha 11) e incrementa o valor da variável *IdleTime* (linha 12).

Caso a mensagem do fluxo *uplink* tenha sido recebida corretamente, então a ACU inicia a monitorização da sua fila de receção até receber a mensagem de ACK da estação destino ou, até à execução da interrupção EP_i . Caso a mensagem de ACK seja recebida (linha 15), é calculado o atraso sofrido pelo fluxo de dados *downlink* (linha 16) e atualizada a variável *MsgDownSent* (linha 17). A variável *MsgDownSent* tem por objetivo sinalizar o encaminhamento com sucesso da mensagem para a estação destino. Caso constate-se que esta não tenha sido atualizada (linha 21), significa então que a mensagem não foi reencaminhada com sucesso. Neste caso, a ACU calcula o atraso sofrido pelo fluxo *downlink* (linha 22) como sendo o tamanho atual do *slot* menos o tempo utilizado pelo fluxo de dados *uplink* (inclusive pelo seu atraso).

Por fim, a ACU invoca a função **COMPUTEDELAY()** passando como parâmetros de entrada o identificador da TS e os atrasos *uplink* e *downlink* previamente medidos. Esta função tem como objetivo suavizar o impacto que variações repentinas possam gerar sobre a redefinição do tamanho do *slot*. Neste contexto, é utilizada uma janela deslizante baseada numa média exponencial ponderada. Sendo assim, os valores de atraso previamente medidos (b_i^{uplink} e $b_i^{downlink}$) são utilizados como base para gerar valores aproximados dos tempos de atraso *uplink* (B_i^{uplink}) e *downlink* ($B_i^{downlink}$) que, posteriormente serão utilizados como base na redefinição do tamanho do *slot*. Os valores de B_i^{uplink} e $B_i^{downlink}$ são dados pelas seguintes equações, respectivamente:

$$B_i^{uplink} = (1 - \alpha) \times B_i^{uplink} + \alpha \times b_i^{uplink} \quad (4.13)$$

$$B_i^{downlink} = (1 - \alpha) \times B_i^{downlink} + \alpha \times b_i^{downlink} \quad (4.14)$$

onde α é o fator de suavização e varia de [0...1]. O valor de α utilizado pela arquitetura RT-WiFi é o mesmo sugerido por Jacobson em [85] e utilizado pela RFC 6298 [86], ou seja, $\alpha = 0.125$.

Neste contexto, a função **COMPUTEDELAY()** é formalizada no algoritmo 13 a qual atualiza os valores de B_i^{uplink} e $B_i^{downlink}$ para uma dada TSID.

Algoritmo 13 Função **ComputeDelay()**

```

1: function COMPUTEDELAY(TSID,  $b^{uplink}$ ,  $b^{downlink}$ )
2:    $AdmittedList[TSID].B^{uplink} \leftarrow ((1 - \alpha) \times AdmittedList[TSID].B^{uplink}) + (\alpha \times b^{uplink})$ 
3:    $AdmittedList[TSID].B^{downlink} \leftarrow ((1 - \alpha) \times AdmittedList[TSID].B^{downlink}) + (\alpha \times b^{downlink})$ 
4: end function

```

Uma vez que os atrasos dos fluxos de dados *uplink* e *downlink* são calculados durante a execução do ciclo TDMA, a redefinição do tamanho dos *slots* ocorre posteriormente, isto é, momentos antes da criação de uma nova mensagem de *beacon* (a qual conterá os novos valores) através da

função **REDEFINESLOTLENGTH**. O tamanho atual do *slot* de uma TS_i ($C_i^{current}$) é redefinido utilizando a seguinte equação:

$$C_i^{current} = \min[(B_i^{uplink} + C_{attempt}^{uplink} + B_i^{downlink} + C_{attempt}^{downlink}), C_i^{max}] \quad (4.15)$$

onde C_i^{max} é o tamanho máximo que o *slot* pode alcançar.

A função **REDEFINESLOTLENGTH** é formalizada no algoritmo 14, a qual atualiza o tamanho dos *slots* de todas as TS contidas na lista de admissão da ACU. Para cada entrada na lista (linha 2) a ACU inicializa algumas variáveis auxiliares (linhas 3 – 7) e, com base nestas variáveis, é calculado o novo tamanho do *slot* (linha 8), que tem um limite superior definido por C^{max} .

Algoritmo 14 Função **RedefineSlotLength()**

```

1: function REDEFINESLOTLENGTH()
2:   for ( $i = 0 \rightarrow AdmittedList.SIZE - 1$ ) do
3:      $C_{attempt}^{uplink} \leftarrow AdmittedList[i].C_{attempt}^{uplink}$ 
4:      $C_{attempt}^{downlink} \leftarrow AdmittedList[i].C_{attempt}^{downlink}$ 
5:      $B^{uplink} \leftarrow AdmittedList[i].B^{uplink}$ 
6:      $B^{downlink} \leftarrow AdmittedList[i].B^{downlink}$ 
7:      $C^{max} \leftarrow AdmittedList[i].C^{max}$ 
8:      $C^{current} \leftarrow \min[(B^{uplink} + C_{attempt}^{uplink} + B^{downlink} + C_{attempt}^{downlink}), C^{max}]$ 
9:   end for
10: end function

```

É importante referir que os valores de B_i^{uplink} e $B_i^{downlink}$ são calculados no ciclo TDMA anterior, diferentemente dos valores de b_i^{uplink} e $b_i^{downlink}$, que são calculados no ciclo TDMA corrente. Assim, quando o sistema é inicializado ou quando uma nova TS é admitida, ambos os valores de B_i^{uplink} e $B_i^{downlink}$ ainda não estão definidos. No entanto, isto não reflete um problema uma vez que ao enviar a requisição ao AP_{TR} as estações definem previamente um valor extra que deve ser alocado para retransmissões (parâmetro *SurplusTime*).

4.3.3 Verificação de Falhas das TS

Numa situação normal, ao excluir uma TS a ACU liberta os recursos a ela alocados, independentemente deste processo ter sido iniciado pela estação ou pela própria ACU. No entanto, quando ocorre uma "falha" numa TS ou então na estação que a detém (por exemplo, desligamento inesperado da estação), estes recursos podem ficar bloqueados caso não exista um mecanismo capaz de identificá-las.

Desta forma, a ACU considera que uma TS_i entrou num estado de "falha" quando esta para de efetuar transmissões por um período de tempo maior que o estipulado pelo parâmetro de intervalo de inatividade (II_i) enviado por ela (via TSPEC) no momento da sua admissão. Assim, após identificá-la a ACU pode excluí-la para libertar os recursos a ela alocados.

Para realizar esta verificação, a ACU insere na lista de admissão uma variável (*IdleTime*) que define o tempo máximo que uma TS_i pode permanecer sem efetuar qualquer transmissão. Cada vez que a ACU identificar a receção de uma mensagem de dados (independentemente de

esta ser recebida com sucesso ou não) proveniente da TS_i e com destino ao AP_{TR} dentro do seu respectivo *slot*, então a variável *IdleTime* é reinicializa ($IdleTime = 0$). Caso contrário, o seu valor é incrementado com base no período de geração de mensagens (P_i) estipulado pela TS_i no momento da admissão.

Assim, antes de criar uma nova mensagem de *beacon*, a ACU invoca a função **CHECKIDLE-FLOWS** (formalizada no algoritmo 15) que tem como objetivo verificar se alguma TS previamente admitida pela ACU encontra-se num estado de "falha". Neste contexto, é executada uma verificação para cada TS contida na sua lista de admissão (linha 2). Se alguma destas TS tiverem o valor da sua variável *IdleTime* maior ou igual ao definido pelo parâmetro Π (linha 3) então é invocada a função **DELTS** (linha 4), a qual irá remover a TS da sua lista de admissão, liberar os recursos por ela alocados e enviar uma mensagem DELTS à estação origem.

Algoritmo 15 Função **CheckIdleFlows()**

```

1: function CHECKIDLEFLOWS()
2:   for ( $i = 0 \rightarrow AdmittedList.SIZE - 1$ ) do
3:     if ( $AdmittedList[i].IdleTime \geq AdmittedList[i].\Pi$ ) then
4:       DELTS[ $AdmittedList[i].TSID$ ]
5:     end if
6:   end for
7: end function

```

4.4 Suporte de Tráfego *Broadcast*

Tal como discutido na secção 4.1, inicialmente foi utilizado como base para a construção da arquitetura RT-WiFi o suporte a aplicações baseadas num modelo de comunicação *peer-to-peer*. Assim, assumindo os conceitos apresentados anteriormente, o objetivo desta secção é apresentar as alterações necessárias para que a arquitetura RT-WiFi possa suportar aplicações que utilizem um modelo de comunicação baseado em *broadcast*. Neste contexto, é importante lembrar que numa rede IEEE 802.11 infraestruturada, o envio de uma mensagem em *broadcast* ocorre somente no fluxo de dados *downlink*, sendo a comunicação no fluxo de dados *uplink* efetuada através de uma transmissão *unicast* (ou seja, com confirmação).

Assim, se por um lado a utilização do modelo de comunicação baseado em *broadcast* torna as comunicações menos confiáveis, uma vez que não é utilizada a mensagem de confirmação (ACK), por outro lado esta abordagem reduz o tempo necessário para que sejam efetuadas as transmissões e, no âmbito da arquitetura RT-WiFi, consequentemente o tamanho dos *slots*.

Neste contexto, são necessárias três modificações para que a arquitetura RT-WiFi possa suportar aplicações que utilizem este modelo de comunicação. A primeira é a maneira como o parâmetro $SurplusTime_k$ (Equação 4.3) é definido pelas estações de tempo-real. Este parâmetro, enviado para a ACU com o objetivo de definir o tempo extra de alocação de uma TS_k , passa a ser

calculado levando em consideração somente o número de retransmissões que podem ser efetuadas no fluxo de dados *uplink*, uma vez que não são efetuadas retransmissões no fluxo de dados *downlink*. Portanto, o parâmetro $SurplusTime_k$ passa a ser definido pela seguinte equação:

$$SurplusTime_k = (C_{attempt}^{uplink} \times RN_k^{uplink}) + C_{attempt}^{downlink} \quad (4.16)$$

onde $C_{attempt}^{uplink}$ e $C_{attempt}^{downlink}$ são, respectivamente, os tempos necessários para efetuar as transmissões com sucesso de uma única mensagem de dados nos fluxos de dados *uplink* e *downlink* e RN_k^{uplink} é o número de retransmissões utilizada pela TS_k como base de cálculo para o fluxo de dados *uplink*.

A segunda modificação ocorre no cálculo do tempo de transmissão do fluxo de dados *downlink* (Equação 4.5). Neste caso, como esta transmissão é efetuada em *broadcast*, são removidos da equação os tempos referentes a recepção da mensagem de confirmação. Assim, o novo valor é dado pela seguinte equação:

$$C_{attempt}^{downlink} = AIFS_{VO}^{QAP} + C_{DATA}[L_k] \quad (4.17)$$

A última modificação é a maneira como o atraso no fluxo de dados *downlink* ($b_i^{downlink}$) é calculado para cada TS_i (Equação 4.12). Como neste caso em específico não é transmitida uma mensagem de confirmação, o novo atraso deve ser medido entre o início do reencaminhamento da mensagem pelo AP_{TR} (dado pelo instante de tempo $T_{Forward}^{Begin}$) e o final da sua respectiva transmissão (dado pelo instante de tempo T_{DATA}^{Sent}). A Figura 4.9 apresenta como é medido o tempo de atraso *downlink* para transmissões em *broadcast*.

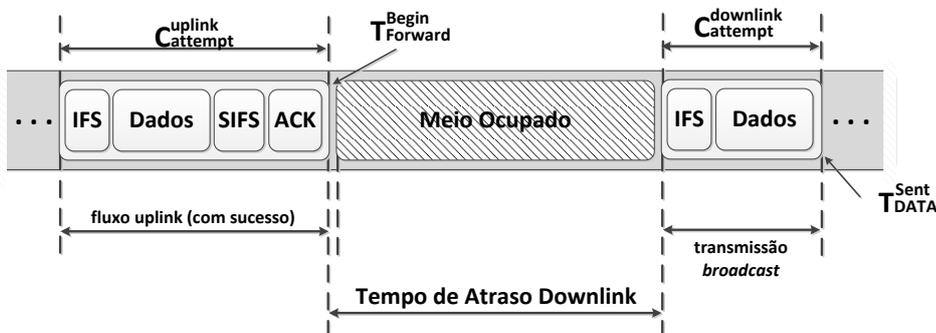


Figura 4.9: Tempo de atraso *downlink* ($b_i^{downlink}$) para transmissão em *broadcast*.

Neste contexto, o valor de $b_i^{downlink}$ passa a ser dado pela seguinte equação:

$$b_i^{downlink} = T_{DATA}^{Sent} - T_{Forward}^{Begin} - C_{attempt}^{downlink} \quad (4.18)$$

Uma vez que o método original de análise dos tempos de atraso *downlink* no AP_{TR} muda em função da transmissão de tráfego *broadcast*, faz-se necessária a indicação deste tipo de tráfego no momento da submissão de uma nova TS a ACU. Esta diferenciação é realizada através do campo

de destino contido no elemento TCLASS enviado juntamente com a requisição ADDTS. Desta forma, como este elemento é naturalmente enviado neste tipo de requisição para identificar a TS, nenhuma alteração adicional precisa ser realizada.

É importante observar que embora a transmissão de tráfego *broadcast* seja realizada de forma diferente ao originalmente proposto, a operação destes dois modos não é mutuamente exclusiva, ou seja, é perfeitamente possível que numa mesma rede RT-WiFi podem existir algumas TS a transmitir tráfego *broadcast* e outras a transmitir tráfego *unicast* (*peer-to-peer*).

4.5 Conclusões

Neste capítulo foi apresentada a arquitetura de comunicação RT-WiFi, a qual tem como objetivo suportar a transmissão de tráfego *soft real-time* em ambientes de comunicação abertos, ou seja, partilhando o meio de comunicação com estações que estão fora da sua esfera de controlo.

Uma das suas principais características é a capacidade de fornecer um serviço de comunicação de tempo-real controlando apenas as estações TR, sem a necessidade de uma atualização de todos os dispositivos de comunicação que estejam a operar na mesma área de cobertura e canal de comunicação. Esta característica por si só já demonstra uma vantagem quando comparada com as propostas apresentadas no capítulo anterior.

Outra vantagem é a utilização de um mecanismo de controlo de acesso ao meio baseado num esquema TDMA. Esta abordagem possibilita uma redução no *overhead* das comunicações (quando comparado com os mecanismos de *polling* tradicionais) e impede colisões entre transmissões de tempo-real. Como cada ciclo TDMA se inicia com a transmissão de uma mensagem de *beacon*, que contém uma lista de escalonamento, além de garantir uma sincronização frequente dos relógios das estações, possibilita também uma elevada flexibilidade ao algoritmo de escalonamento, uma vez que a cada ciclo TDMA a ordem de alocação dos *slots* pode ser alterada em função das necessidades existentes.

Esta flexibilidade é estendida à capacidade que a arquitetura tem de ajustar dinamicamente o tamanho de cada *slot* alocado pela ACU. Este ajuste é realizado tendo como base os atrasos que os fluxos de dados *uplink* e *downlink* sofrem para transmitirem as respectivas mensagens. Tais atrasos levam em consideração tanto o atraso no acesso ao meio (devido à ocupação do canal de comunicação por outra estação) quanto o tempo gasto em retransmissões (sejam estas resultantes de colisões ou de erros de transmissão causados por ruídos no canal de comunicação).

Neste contexto, para evitar que seja necessário executar um novo teste de escalabilidade cada vez que é realizada uma alteração no tamanho dos *slots*, no momento da admissão de uma TS_k a ACU define o tamanho máximo que os *slots* alocados para esta TS podem atingir. Assim, independentemente do tamanho que estes *slots* possam assumir durante a operação da rede, o teste de escalabilidade será sempre realizado com o tamanho máximo definido pela ACU. Como consequência, esta abordagem reduz o custo computacional imposto à ACU, uma vez que o teste é apenas executado no momento da admissão de uma nova TS.

A gestão dinâmica no tamanho dos *slots* permite uma melhor utilização dos recursos disponíveis na rede. Com o objetivo de otimizar a utilização deste recursos e, ao levar em consideração que em algumas situações o tamanho atual do *slot* é menor que o seu tamanho máximo, a arquitetura possibilita a transmissão de dois tipos de tráfego de tempo-real: alta e baixa prioridade; além de tráfego não tempo-real. Desta forma, o tráfego tempo-real de baixa prioridade é transmitido nos espaços alocados pelo tráfego tempo-real de alta prioridade, mas que no momento, devido a redução no tamanho atual destes *slots*, não estão a ser utilizados.

No que diz respeito ao mecanismo de controlo de admissão, este foi construindo tendo como principal objetivo as *deadlines* das mensagens de cada TS admitida pela ACU. O objetivo é evitar que as características de algumas TS possam ser generalizadas, tal como algumas propostas apresentadas no capítulo anterior. Para isto, previamente à admissão de uma TS_k , a respectiva estação deve enviar ao AP_{TR} uma requisição ADDTS contendo algumas informações de TSPEC. Após analisar este pedido a ACU decide entre a admissão ou não desta TS. Caso seja admitida, o algoritmo de escalonamento realizará a ordenação das transmissões com base na *deadline* de todas as TS admitidas pela ACU. Ressalva-se que, a arquitetura RT-WiFi foi desenvolvida de forma a facilitar a migração de um algoritmo de escalonamento para outro, sem que seja necessária uma reestruturação completa dos mecanismos da arquitetura.

Para auxiliar na gestão dos recursos disponíveis, a arquitetura RT-WiFi implementa um mecanismo capaz de identificar e remover TS admitidas pela ACU e que por algum motivo não estejam em operação. Isto permite que os recursos alocados por estas TS possam ser libertados para outros fins.

Outra vantagem da arquitetura RT-WiFi é a sua fácil implementação, sendo necessárias apenas pequenas modificações no *driver/firmware* dos dispositivos IEEE 802.11 padrão, o que torna esta proposta compatível com *hardware* COTS. Esta é uma característica muito importante do ponto de vista econômico, uma vez que não implica no desenvolvimento de novos dispositivos de comunicação para a implantação de uma rede RT-WiFi.

Por fim, como todas as estruturas e formatos de mensagens utilizadas pela arquitetura RT-WiFi estão previstas na norma IEEE 802.11, esta mantém a sua compatibilidade com os dispositivos IEEE 802.11 standard. A única alteração significativa foi a introdução de uma lista de escalonamento na mensagem de *beacon*. No entanto, a norma IEEE 802.11 define que esta mensagem tem um tamanho variável e também existem campos reservados para uso futuro. Assim as alterações propostas não implicam nenhuma modificação no formato original desta mensagem.

Capítulo 5

Resultados

O principal objetivo deste capítulo é avaliar o comportamento da arquitetura RT-WiFi num ambiente de comunicação aberto. Para isso, definiu-se um conjunto de cenários representativos deste tipo de ambiente e avaliou-se a arquitetura proposta através de um conjunto de métricas relevantes. Além disso, compara-se o desempenho do RT-WiFi com soluções já existentes, nomeadamente com a função HCCA (HCF Controlled Channel Access), uma vez que esta é a única solução definida pela norma IEEE 802.11 que suporta um mecanismo de controlo de admissão e um escalonador. Os resultados foram obtidos a partir de modelos de simulação do RT-WiFi e do HCCA¹ implementados utilizando a ferramenta OPNET Wireless Modeler [87].

5.1 Descrição dos cenários

Em consequência da crescente comercialização de dispositivos de comunicação baseados na norma IEEE 802.11 e na limitação do número de canais disponibilizados pela respectiva camada física, torna-se cada vez mais difícil a criação de um ambiente de comunicação fechado, ou seja, livre de interferências provenientes da sobreposição de outras redes que possam estar a operar na mesma zona geográfica.

Esta situação torna-se ainda mais crítica em regiões metropolitanas, isto porque, com a redução de custos desta tecnologia, além das WLAN (*Wireless Local Network*) normalmente encontradas em escritórios, lojas, instituições de ensino e ambientes públicos, tem-se tornando cada vez mais frequente a utilização deste tipo de redes em ambientes domésticos.

Neste contexto, optou-se por criar cenários de simulação que modelem a operação da rede de tempo-real (seja esta RT-WiFi ou HCCA) num **ambiente de comunicação aberto**. Assim, considerou-se que a rede de tempo-real (TR) tem a sua área de cobertura sobreposta por uma rede não tempo-real (NTR) a operar no mesmo canal de comunicação (Figura 5.1).

A rede NTR gera diferentes tipos de tráfegos que estão fora da esfera de controlo do sistema de comunicação de tempo-real. Neste sentido, embora a maioria das redes IEEE 802.11 atuais utilizem como mecanismo de controlo de acesso ao meio a função DCF (*Distributed Coordination*

¹A descrição do modelo HCCA é detalhada no anexo B.

Function), optou-se por implementar a rede NTR sobre a função EDCA (*Enhanced Distributed Channel Access*). O objetivo é avaliar o comportamento das redes de tempo-real num ambiente de comunicação aberto mais crítico, uma vez que assim, o tráfego NTR será transmitido utilizando configurações de QoS (*Quality of Service*). Como consequência, o nível de interferência (nomeadamente o atraso no acesso ao meio) imposto pelo tráfego NTR sobre o tráfego TR é superior.

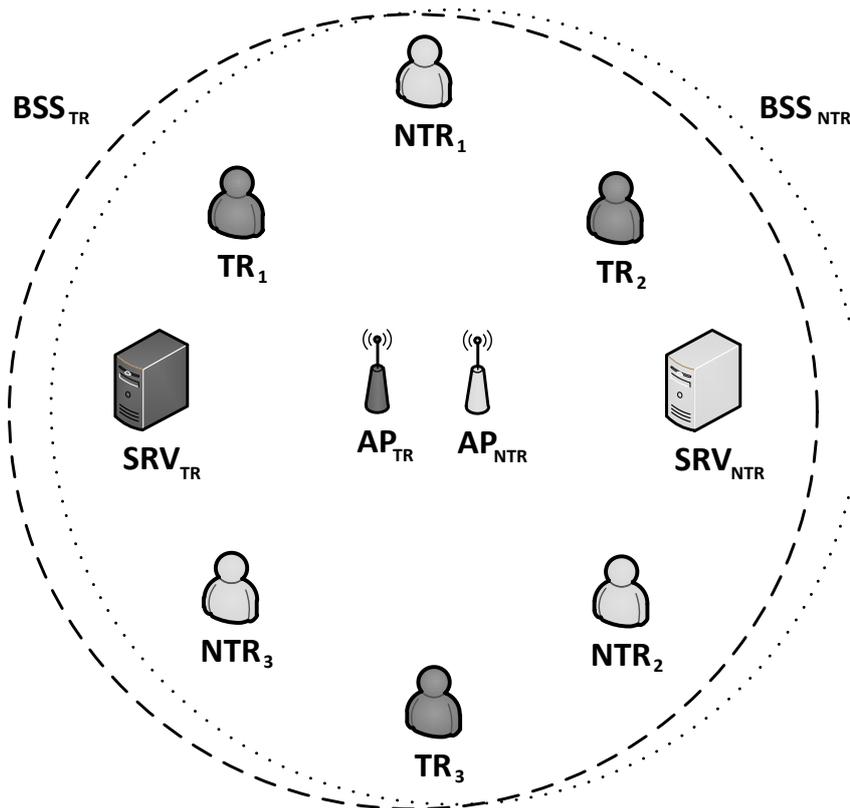


Figura 5.1: Ambiente de comunicação aberto.

Portanto, assumiu-se uma aplicação de tempo-real caracterizada por fluxos de dados *peer-to-peer*, ou seja, onde uma estação origem envia dados para uma estação destino. Uma vez que ambos os sistemas de comunicação de tempo-real avaliados assumem uma topologia infraestruturada, devemos levar em conta que serão criados dois fluxos de dados (*uplink* e *downlink*). Portanto, optou-se por criar cenários onde a rede de tempo-real é composta por múltiplas estações TR que enviam mensagens para um servidor (SRV_{TR}) através do *Access Point* (AP_{TR}). A utilização de um único destino (SRV_{TR}) possibilita a simplificação no processo de análise dos resultados sem implicar na modificação da característica do tráfego, uma vez que para a sua transmissão continuam a ser utilizados fluxos de dados *uplink* e *downlink*.

As estações TR e o SRV_{TR} estão geograficamente posicionados de forma aleatória num raio de 55 metros em torno do AP_{TR} [88]. Cada estação TR realiza a submissão de uma única TS (*Traffic*

Stream). Portanto, para cada cenário avaliado, o número de estações de tempo-real varia de 1 até o máximo admitido pelo controlo de admissão em questão.

A rede não tempo-real utiliza também uma topologia infraestruturada sendo composta por 10 estações, um servidor (SRV_{NTR}) e um AP_{NTR} . Assim como feito na rede de tempo-real, a utilização de uma única estação destino (SRV_{NTR}) facilita a análise dos resultados. As estações NTR e o SRV_{NTR} estão geograficamente posicionadas de forma aleatória num raio de 55 metros em torno do AP_{NTR} . A troca de mensagens entre as estações NTR e o SRV_{NTR} é realizada através do AP_{NTR} . Neste contexto, as estações e o SRV_{NTR} impoem três níveis diferentes de carga de rede sobre o meio de comunicação.

O uso de 10 estações NTR na criação dos cenários deve-se ao facto deste número possibilitar que a carga de rede imposta pelo tráfego NTR (detalhado na secção 5.1.2) possa atingir níveis próximos do ponto de saturação do meio de comunicação. Como o impacto sobre o comportamento da rede de tempo-real está mais relacionado com a quantidade de carga gerada pelas estações NTR do que com o número de estações NTR em operação, variações no número de estações NTR não resultam em mudanças de tendências nos resultados apresentados.

No contexto da camada física, optou-se por utilizar em ambas as redes (TR e NTR) a norma IEEE 802.11a [15]. Assim, cada estação opera em modo OFDM (*Orthogonal Frequency Division Multiplexing*). Na rede NTR e na arquitetura RT-WiFi as mensagens de controlo são transmitidas utilizando a taxa base (6 Mbits/s) e as mensagens de dados são transmitidas utilizando a taxa máxima (54 Mbits/s). No caso da função HCCA, ambas as mensagens, de controlo e dados, são transmitidas utilizando a taxa base (6 Mbits/s), tal como exige a norma IEEE 802.11.

A utilização da norma IEEE 802.11a deve-se ao facto desta apresentar duas características que, numa implementação real, seriam consideradas vantajosas para as redes de tempo-real. A primeira é o funcionamento na faixa de frequência de 5 GHz. Isto evita algumas interferências² que podem ocorrer em transmissões realizadas numa faixa de frequência de 2.4 GHz. A outra característica é a maior disponibilidade de canais não sobrepostos (12 canais), quando comparada aos 3 canais fornecidos pela faixa de frequência de 2.4 GHz.

Outro aspecto importante, que deve ser levado em consideração, para a definição dos cenários de simulação é o nível de ruído que as redes TR e NTR estão submetidas. Este ruído, normalmente é resultante de interferências eletromagnéticas (EMI – *Electromagnetic Interferences*) ou da recepção (em instantes diferentes) de múltiplas cópias de uma mesma onda de rádio (*fading*). Em ambientes industriais, o nível de ruído nas transmissões pode ser agravado quer pela existência de dispositivos elétricos de alta potência, quer pelo número de equipamentos metálicos instalados no ambiente onde as comunicações são realizadas. Além disso, caso os dispositivos da rede de tempo-real sejam móveis e/ou existam movimentações no ambiente de comunicação (por exemplo: carros a serem movimentados numa linha de montagem), o nível de ruído sofrido por cada estação torna-se variável.

²Aparelhos de telefone sem fio, microondas e dispositivos *Bluetooth* e *ZigBee* podem gerar interferências em redes IEEE 802.11 a operar na faixa de frequência de 2.4 GHz.

Neste contexto, para que fosse possível obter um comportamento mais realista, foi utilizado um modelo de erros para o canal de comunicação disponibilizado pela ferramenta de simulação. Este modelo avalia dinamicamente o valor do BER (*Bit Error Ratio*) de cada transmissão baseado nos valores médios produzidos pelo SNR (*Signal-to-Noise Ratio*). Desta forma, é possível obter valores independentes de BER para cada transmissão realizada. Assim, para os cenários avaliados, o valor do BER varia de 10^{-4} a 10^{-3} [89].

5.1.1 Tráfego de Tempo-Real

O tráfego de tempo-real para ambos os sistemas de comunicação avaliados (RT-WiFi e HCCA) é gerado pelas estações de tempo-real que submetem ao coordenado central uma única TS cada. O número de estações de tempo-real varia de 1 até o máximo admitido pelo controlo de admissão. Cada TS_i possui um período fixo de geração de mensagens (P_i). Neste contexto, para os cenários de comunicação avaliados, foram considerados períodos de geração de mensagens de 30, 60 e 90 milisegundos. Optou-se por estes períodos por acreditarmos que são compatíveis com a realidade, embora seus valores possam variar de acordo com o tipo de planta industrial [90, 65].

Cada MSDU (*MAC Service Data Unit*) gerado por uma TS_i possui um tamanho fixo (L_i) de 73 bytes³, ou seja, uma MPDU (*MAC Process Data Unit*) de 103 bytes. Isto segue as características do tráfego de tempo-real geralmente encontrado em plantas industriais onde, na sua grande maioria, é caracterizado por mensagens trocadas entre sensores, controladores e atuadores.

Tendo em conta as taxas de transmissão previamente apresentadas, podemos calcular o nível de ocupação no meio de transmissão para cada TS. Estes valores são apresentados na Tabela 5.1.

Uma vez que o RT-WiFi utiliza o RM (*Rate Monotonic*) como algoritmo de escalonamento,

Tabela 5.1: Ocupação do meio de transmissão (por mensagem).

Parâmetro	RT-WiFi		HCCA	
	Uplink	Downlink	Uplink	Downlink
AIFS / Poll	34 μs	25 μs	64 μs	-
Dados	36 μs	36 μs	164 μs	164 μs
SIFS	16 μs	16 μs	16 μs (x2)	16 μs (x2)
ACK	28 μs	28 μs	44 μs	44 μs
Total do fluxo	114 μs	105 μs	304 μs	240 μs
Total geral	219 μs		544 μs	

foi considerado que a *deadline* relativa (D_i) das mensagens produzidas por cada TS_i é igual ao seu respectivo período de geração ($D_i = P_i$). Para situações onde o $D_i \neq P_i$, pode-se utilizar o parâmetro *Atraso Limite* do elemento TSPEC. Este parâmetro define o tempo máximo permitido pela TS_i para a transmissão com sucesso de uma MSDU. Como este tempo é contado a partir do momento de chegada da MSDU à subcamada MAC (dado pelo seu *timestamp*), pode ser associado ao limite temporal para a execução da transmissão, ou seja, a *deadline* da mensagem.

³Equivalente a 45 bytes de dados na camada de aplicação quando esta é encapsulada numa pilha UDP/IP.

No contexto da arquitetura RT-WiFi, todo o tráfego de tempo-real foi definido como sendo de *alta prioridade*. A tabela 5.2 apresenta os valores utilizados pelo mecanismo de controlo de admissão para os seguintes parâmetros: período de geração de mensagens (P), tamanho da MSDU (L), intervalo de inatividade (II)⁴, factor de suavização (α), tamanho máximo de uma MSDU ($MSDU_{max}$), número máximo de retransmissões para o fluxo de dados *uplink* (RN_{uplink}), número máximo de retransmissões para o fluxo de dados *downlink* ($RN_{downlink}$) e intervalo da mensagem do *beacon* (BI).

Tabela 5.2: Parâmetros do RT-WiFi.

Parâmetro	Valor(es)
P (em milisegundos)	30 / 60 / 90
L (em bytes)	73
II (em milisegundos)	90 / 180 / 270
α	0.125
$MSDU_{max}$ (em bytes)	2304
RN_{uplink}	2
$RN_{downlink}$	2
BI (em milisegundos)	30 / 60 / 90

No contexto da função HCCA, a tabela 5.3 apresenta os valores utilizados pelo seu mecanismo de controlo de admissão para os seguintes parâmetros: taxa média de dados (ρ), tamanho da MSDU (L), intervalo máximo de serviço (SI_{max}), atraso máximo (D), tamanho máximo da MSDU ($MSDU_{max}$), duração do intervalo livre de contenção (CFP), o qual foi definido como o máximo permitido pela norma IEEE 802.11 e intervalo da mensagem de *beacon* (BI).

Tabela 5.3: Parâmetros do HCCA.

Parâmetro	Valor(es)
ρ (em bytes/s)	2434 / 1217 / 812
L (em bytes)	73
SI_{max} (em milisegundos)	30 / 60 / 90
D (em milisegundos)	30 / 60 / 90
$MSDU_{max}$ (em bytes)	2304
CFP	$CFP_{MaxDurationLimit}$
BI (em milisegundos)	30 / 60 / 90

Para garantir que as *deadlines* das mensagens geradas pela função HCCA sejam iguais aos seus respectivos períodos de geração (P), os parâmetros SI_{max} e D assumem o mesmo valor definido no parâmetro P, ou seja, $P = D = SI_{max}$.

No contexto do valor apresentado pelo parâmetro ρ , este é definido pela RFC 2215 [91] e, refere-se a carga (em bytes por segundo) recebida pela camada MAC das camadas superiores, sendo o respectivo valor dado pela seguinte equação:

$$\rho = \left[\frac{1}{SI_{max}} \times L \right] \quad (5.1)$$

⁴O intervalo de inatividade de uma TS_i foi definido como $3 \times P_i$.

5.1.2 Tráfego Não Tempo-Real

O tráfego não tempo-real é gerado pelas estações NTR e pelo SRV_{NTR} . Como o objetivo principal das simulações apresentadas neste capítulo é avaliar o comportamento da rede de tempo-real a operar num ambiente de comunicação aberto, devemos considerar que a rede NTR é capaz de impor diferentes cargas na rede.

Assim, para avaliar o comportamento da rede de tempo-real sob diferentes níveis de ocupação do meio de comunicação e levando em conta que o ponto de saturação de uma rede IEEE 802.11 é aproximadamente 60% do *throughput* teórico máximo [92, 93], foram definidos três níveis de carga de rede impostos pelas estações NTR e SRV_{NTR} :

- **Baixo:** representa 10% do *throughput* teórico máximo;
- **Médio:** representa 30% do *throughput* teórico máximo;
- **Alto:** representa 50% do *throughput* teórico máximo.

Além da carga de rede, outro factor importante que deve ser considerado no tráfego NTR é o tipo de serviço responsável pela geração das mensagens que estão a ser transmitidas. Isto irá definir, entre outros aspectos, o tamanho das mensagens, a periodicidade de geração⁵, o tipo do fluxo de transmissão⁶ e o protocolo de transporte utilizado. Estas quatro características são as principais responsáveis pelo comportamento do tráfego gerado pela rede NTR.

Logo, com o objetivo de criar cenários onde o tráfego NTR fosse o mais realista possível, foram analisados diversos trabalhos relativos à sua caracterização [94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106]. Após esta análise, concluiu-se que grande parte do tráfego transmitido poderia ser modelado por cinco tipos diferentes:

1) Tráfego HTTP (*Hypertext Transfer Protocol*)

Este tráfego foi modelado como requisições enviadas das estações NTR a um serviço HTTP alocado no SRV_{NTR} . Assim, logo após processar as requisições, o SRV_{NTR} envia para a estação cliente mensagens contendo a página solicitada. Como o tráfego HTTP é transmitido sobre o protocolo de transporte TCP, isto resulta na transmissão de mensagens adicionais (além das requisições e respostas) para que seja possível efetuar o controlo da conexão. Todo o tráfego HTTP é transmitido através da fila *best-effort* da função EDCA e representa 40% da carga total imposta na rede [96, 97].

Deste modo, o serviço HTTP disponibiliza o acesso a dois tipos de páginas: **i)** páginas comuns, contendo um número variado de elementos (por exemplo: texto, figuras, áudio e vídeo), e; **ii)** páginas que possuem *streams* de vídeo, ou seja, semelhante as páginas comuns mas contendo um vídeo de maior dimensão. Para cada 10 requisições processadas pelo SRV_{NTR} , 7 são para páginas comuns e 3 são para *streams* de vídeo [106].

⁵A geração das mensagens pode assumir um comportamento CBR (*Constant Bit Rate*) ou VBR (*Variable Bit Rate*).

⁶A conexão entre cliente e servidor pode ser caracterizada por um fluxo de transmissão de mensagens unidirecional (por exemplo: *stream* de vídeo) ou bidirecional (por exemplo: video-conferência).

Neste contexto, as requisições são caracterizadas por mensagens pequenas, de tamanho fixo (500 bytes) e com um intervalo de geração modelado por uma distribuição de *Poisson* [98, 101]. Por outro lado, as respostas têm um tamanho que varia de acordo com a página solicitada. Desta forma, com o objetivo de modelar o acesso a diferentes páginas de Internet, para cada requisição processada, o SRV_{NTR} utiliza uma distribuição uniforme para variar o tamanho do conteúdo enviado como resposta. Assim, para as páginas comuns, este valor varia de 400 *Kbytes* a 2 *Mbytes* [101] e, para as páginas que possuem *stream* de vídeo, este valor varia de 2 a 18 *Mbytes* [103]. A Tabela 5.4 resume os parâmetros utilizados para o tráfego HTTP.

Tabela 5.4: Parâmetros utilizados para o tráfego HTTP.

Parâmetro	Carga		
	Baixo	Médio	Alto
Request Size (bytes)	500	500	500
Page Interarrival Time (segundos)	poisson(5)	poisson(3)	poisson(1)
Comum Object Size (Mbytes)	uniform_int(0.4,2)	uniform_int(0.4,2)	uniform_int(0.4,2)
Comum Number of Objects	uniform_int(3,10)	uniform_int(3,10)	uniform_int(3,10)
Vídeo Object Size (Mbytes)	uniform_int(2,18)	uniform_int(2,18)	uniform_int(2,18)
Vídeo Number of Objects	constant(1)	constant(1)	constant(1)
Type of Service (ToS)	Best-Effort(0)	Best-Effort(0)	Best-Effort(0)

2) Tráfegos SMTP (*Simple Mail Transfer Protocol*) e POP3 (*Post Office Protocol version 3*)

Este tráfego foi modelado como o envio e a recepção de e-mails através dos serviços SMTP e POP3, respectivamente, alocados no SRV_{NTR} . Ambos os tráfegos são transmitidos sobre o protocolo TCP, resultando assim na transmissão de mensagens adicionais para controle da conexão. Juntos, os tráfegos SMTP e POP3 representam 15% da carga total imposta na rede [96, 97]. A sua transmissão é efetuada através da fila *best-effort* da função EDCA.

O tamanho de cada e-mail é modelado por uma distribuição Gaussiana⁷ tendo como valor médio 22,7 *Kbytes* [96]. Os intervalos, de envio de e-mails ao servidor SMTP e, requisições feitas ao servidor POP3, são modelados por uma distribuição de *Poisson* [98]. A Tabela 5.5 resume os parâmetros utilizados para o tráfego SMTP/POP3.

Tabela 5.5: Parâmetros utilizados para o tráfego SMTP/POP3.

Parâmetro	Carga		
	Baixo	Médio	Alto
Send Interarrival Time (segundos)	poisson(5)	poisson(3)	poisson(1)
Send Group Size	3	3	3
Receive Interarrival Time (segundos)	poisson(5)	poisson(3)	poisson(1)
Receive Group Size	6	6	6
E-mail Size (bytes)	normal($\mu=22.7, \sigma^2=0.1$)	normal($\mu=22.7, \sigma^2=0.1$)	normal($\mu=22.7, \sigma^2=0.1$)
Type of Service (ToS)	Best-Effort(0)	Best-Effort(0)	Best-Effort(0)

⁷A distribuição de Gauss (Gaussiana) também é conhecida como distribuição normal.

3) Tráfego FTP (*File Transfer Protocol*)

Este tráfego foi modelado como o envio e a recepção de arquivos através de um serviço FTP alocado no SRV_{NTR} . Utiliza o protocolo de transporte TCP para gerir suas conexões, e é transmitido através da fila *best-effort* da função EDCA, representando 5% da carga total imposta na rede [96, 97].

De todo o tráfego FTP processado pelas estações NTR, 80% é caracterizado pela recepção de arquivos do SRV_{NTR} e apenas 20% é caracterizado pelo envio de arquivos ao SRV_{NTR} . O tamanho dos arquivos transferidos é modelado por uma distribuição Gaussiana tendo como valor médio 500 *Kbytes* [94]. Os intervalos de envio e requisições de arquivos são modelados por uma distribuição de *Poisson* [97]. A Tabela 5.6 resume os parâmetros utilizados para o tráfego FTP.

Tabela 5.6: Parâmetros utilizados para o tráfego FTP.

Parâmetro	Carga		
	Baixo	Médio	Alto
Command Mix (Get/Total)	80%	80%	80%
Inter-request Time (segundos)	poisson(1)	poisson(1)	poisson(1)
File Size (Kbytes)	normal($\mu=84, \sigma^2=0.1$)	normal($\mu=250, \sigma^2=0.1$)	normal($\mu=500, \sigma^2=0.1$)
Type of Service (ToS)	Best-Effort(0)	Best-Effort(0)	Best-Effort(0)

4) Tráfego de Vídeo

Este tráfego foi modelado como chamadas de vídeo-conferência entre as estações NTR e o SRV_{NTR} . Utiliza o protocolo UDP e representa 25% da carga total imposta na rede [96, 97]. As suas mensagens são transmitidas através da fila de vídeo da função EDCA.

As chamadas de vídeo-conferência utilizam o *codec*⁸ de vídeo H.264 [107] com o perfil *CBP Constrained Baseline Profile* [105]. Para possibilitar o aumento da carga de rede imposta pelas estações NTR, a resolução das chamadas de vídeo-conferência é aumentada. A Tabela 5.7 resume os parâmetros utilizados para o tráfego de vídeo.

Tabela 5.7: Parâmetros utilizados para o tráfego de vídeo.

Parâmetro	Carga		
	Baixo	Médio	Alto
Frame Interarrival Time Information (segundos)	constant(0.05)	constant(0.05)	constant(0.05)
Frame Size Information (bytes)	constant(730)	constant(2175)	constant(3625)
Type of Service (ToS)	Interactive Multimedia (5)	Interactive Multimedia (5)	Interactive Multimedia (5)

⁸Um *codec* é um dispositivo ou programa de computador capaz de codificar e decodificar um fluxo de dados ou um sinal digital.

5) Tráfego de Voz

Este tráfego foi modelado como chamadas de VoIP (*Voice over IP*) efetuadas entre as estações NTR e o SRV_{NTR} . Utiliza o protocolo UDP e representa 15% da carga total imposta na rede [96, 97]. As suas mensagens são transmitidas através da fila de voz da função EDCA.

As chamadas de VoIP utilizam o *codec* de voz G.711 [108] com um *Codec Bit Rate* que varia de acordo com a carga de rede imposta pelas estações NTR. Desta forma, como são transmitidas mensagens com um tamanho fixo de 20 milissegundos, o tamanho final da mensagem irá variar de acordo com o *Codec Bit Rate* utilizado [100]. Para possibilitar o aumento da carga de rede imposta pelas estações NTR, o *Codec Bit Rate* das chamadas de VoIP é aumentado. A Tabela 5.8 resume os parâmetros utilizados para o tráfego de voz.

Tabela 5.8: Parâmetros utilizados para o tráfego de voz.

Parâmetro	Carga		
	Alto	Médio	Alto
Silence Length (segundos)	exponential(0.01)	exponential(0.01)	exponential(0.01)
Talk Spurt Length (segundos)	exponential(0.992)	exponential(0.992)	exponential(0.992)
Encoder Scheme	G.711	G.711	G.711
Coding Rate (bits/segundo)	40 Kbps	135 Kbps	240 Kbps
Frame Size (segundos)	0.02	0.02	0.02
Frame Size (bytes)	100	337.5	600
Voice Frames per Packet	1	1	1
Type of Service (ToS)	Interactive Voice (6)	Interactive Voice (6)	Interactive Voice (6)
Copression delay (segundos)	0.02	0.02	0.02
Decompression delay (segundos)	0.02	0.02	0.02

5.2 Resultados

As métricas de desempenho analisadas são: atraso médio, percentagem média de *deadlines* perdidas, *fairness* e tamanho médio dos *slots*. O atraso médio das transmissões representa o atraso fim-a-fim das mensagens que foram transmitidas com sucesso das estações de tempo-real para o SRV_{TR} . É importante observar que este valor não deve ultrapassar a *deadline* relativa estipulada para cada TS, caso contrário as mensagens serão entregues fora do prazo limite. Esta métrica permite também a avaliação do comportamento (variável ou constante) das transmissões. Neste contexto, para algumas aplicações de tempo-real, é desejável que o atraso nas transmissões tenha um comportamento constante, uma vez que isto torna as comunicações mais previsíveis e também possibilita a redução do *jitter*.

Outra métrica analisada é a percentagem média de *deadlines* perdidas que, representa a percentagem de mensagens geradas pelas estações de tempo-real e que perderam as suas respectivas *deadlines*. É importante observar que nesta métrica são contabilizadas tanto as mensagens que foram entregues corretamente, mas fora do prazo limite, quanto as mensagens que não puderam ser entregues por erros e/ou colisões durante as suas respectivas transmissões. Esta métrica pode

ser considerada uma das mais importantes na análise dos sistemas de comunicação de tempo-real, uma vez que quantifica de forma direta o limite crítico que o sistema poderá atingir.

No contexto dos resultados do RT-WiFi, outra métrica considerada relevante é o tamanho médio dos slots. Este valor representa o tamanho médio dos *slots* alocados às TS admitidas pela ACU (*Admission Control Unit*). A sua análise possibilita observar o comportamento dinâmico do mecanismo responsável por gerir o tamanho dos *slots* sob diferentes cargas de rede e número de TS admitidas.

Por fim, com o objetivo de avaliar o impacto que as comunicações de tempo-real têm sobre a rede não tempo-real, foi analisado também o impacto sobre o *fairness* das transmissões NTR. Faz-se observar que esta não é uma métrica clara e bem definida, tendo sido proposta na literatura diversas abordagens para a sua quantificação [109, 110, 111, 112, 113]. No entanto, com o objetivo de simplificar a sua avaliação, optou-se por analisá-la com base no *throughput* agregado da rede NTR, quando esta opera sozinha no meio de comunicação e, quando este é compartilhado com a rede TR.

Para que fosse possível obter diferentes tendências e comportamentos, cada simulação foi repetida 5 vezes utilizando-se uma semente (*seed*) diferente para cada simulação. Logo, os resultados apresentados neste capítulo representam a média dos valores obtidos nestas simulações. Todos os resultados são apresentados com um intervalo de confiança de 95%, com uma largura relativa de 5%. O tempo de simulação utilizado representa o tempo necessário para que os resultados obtidos respeitassem o intervalo de confiança selecionado.

5.2.1 Atraso Médio

A primeira avaliação considera um cenário onde cada estação TR admitida pela ACU gera mensagens num intervalo periódico de 30 milissegundos ($P = 30ms$). Estas estações operam na mesma área de cobertura e canal de comunicação que outras 10 estações NTR que impõem diferentes cargas de rede (*Baixa, Média e Alta*).

A Figura 5.2 ilustra o atraso médio das transmissões para um número de estações TR, que varia de 1 até o máximo suportado pelo mecanismo de controlo de admissão em questão (eixo X). A primeira característica que é possível observar é a grande diferença entre o número máximo de estações admitidas pelo RT-WiFi (19 estações) e pelo HCCA (apenas 2 estações). O número reduzido de estações admitidas pelo HCCA é o reflexo das regras pessimistas implementadas pelo seu respectivo mecanismo de controlo de admissão. Com relação ao atraso médio sofrido pelas transmissões do RT-WiFi, é possível observar que este é ligeiramente menor ($\approx 15ms$, ou seja, metade do SI) ao apresentado pelo HCCA ($\approx 17ms$).

Uma tendência semelhante à anterior pode ser observada na segunda avaliação, onde o cenário passa a considerar um período de geração de mensagens TR definido por $P = 60ms$ (Figura 5.3). Neste caso, devido à diminuição do número de mensagens geradas por cada TS a cada segundo, houve um acréscimo no número máximo de estação que ambos os sistemas de comunicação, RT-WiFi e HCCA, são capazes de admitir.

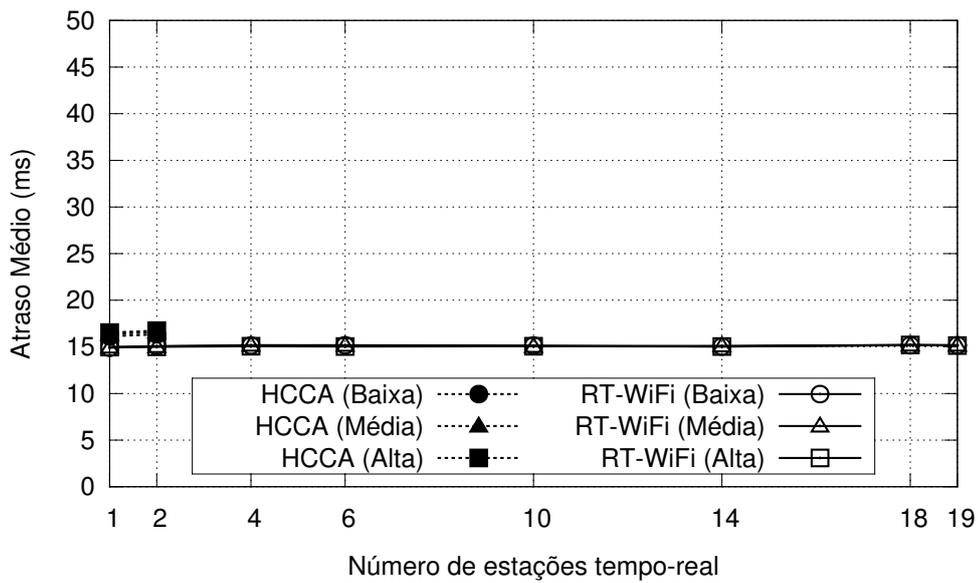


Figura 5.2: Atraso médio para o cenário $P = 30ms$.

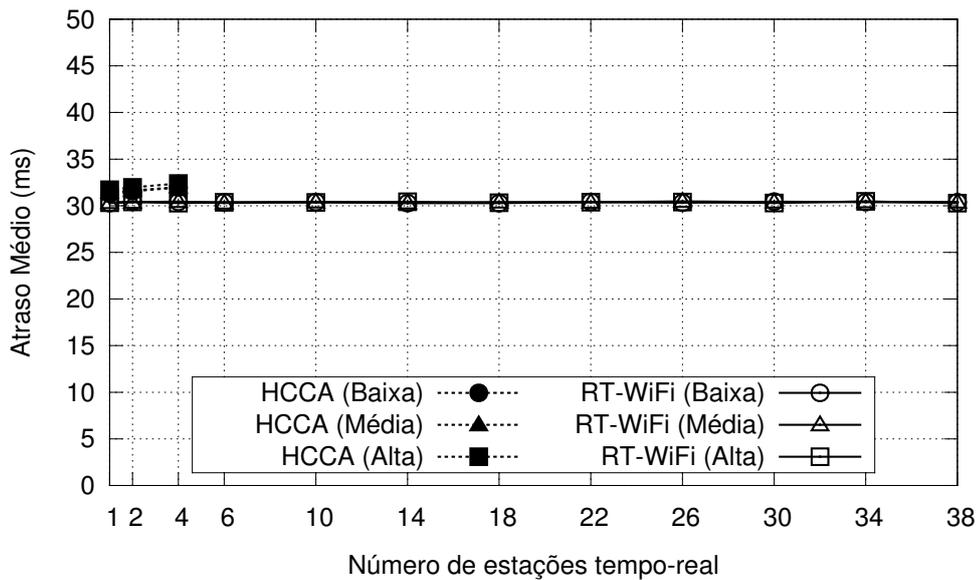


Figura 5.3: Atraso médio para o cenário $P = 60ms$.

No caso do RT-WiFi, o número de estações admitidas pela ACU duplicou, seguindo assim um comportamento já esperado, uma vez que o número de mensagens geradas por segundo em cada TS foi reduzido pela metade, quando comparado ao cenário anterior. O mesmo ocorreu como o HCCA, que passou a admitir 4 estações.

Tal como no caso anterior, o atraso médio sofrido pelas transmissões do RT-WiFi ($\approx 30ms$) é ligeiramente inferior ao apresentado pelo HCCA que, se inicia em $\approx 31ms$ (para 1 estação admitida) indo até $\approx 33ms$ (para 4 estações admitidas).

Por fim, a última avaliação realizada considera um cenário onde as estações TR têm um período de geração de mensagens de $P = 90ms$ (Figura 5.4). Assim como esperado, com a diminuição do número de mensagens geradas por cada TS a cada segundo, ocorreu um acréscimo no número de estações admitidas tanto pelo HCCA (que passou a admitir até 6 estações), quanto pelo RT-WiFi (que passou a admitir até 58 estações).

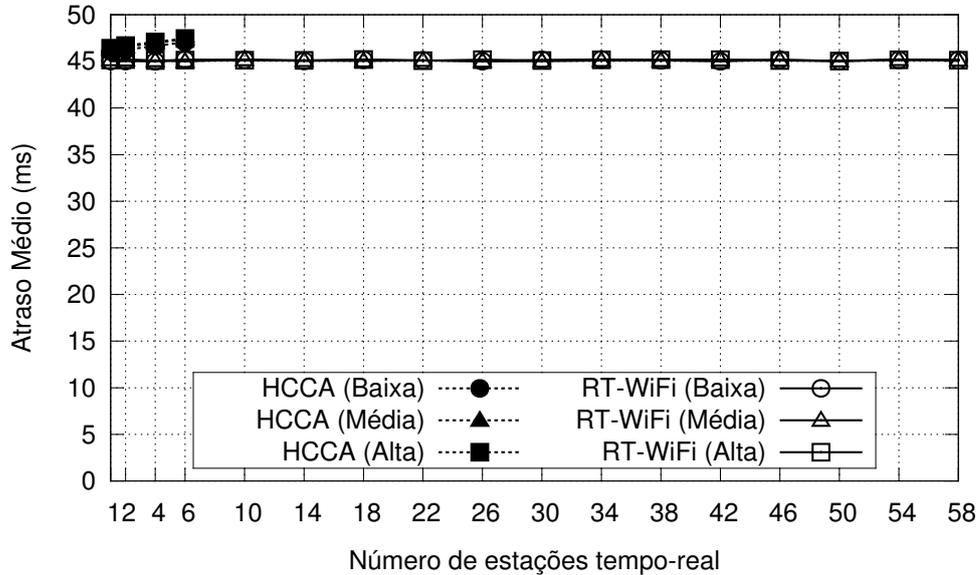


Figura 5.4: Atraso médio para o cenário $P = 90ms$.

Seguindo a tendência dos resultados anteriores, é possível observar que o atraso médio sofrido pelas transmissões do RT-WiFi ($\approx 45ms$) é ligeiramente inferior ao apresentado pelo HCCA, que se inicia em $\approx 46ms$ (para 1 estação admitida) indo até $\approx 48ms$ (para 6 estações admitidas).

Para todos os resultados apresentados anteriormente, foram também avaliados os valores do respectivo desvio padrão. No entanto, devido a sua reduzida expressão (tipicamente $< 5\%$), não foram representados graficamente. Pode-se assim, concluir que, o *jitter* das transmissões de ambos os sistemas de comunicação, RT-WiFi e HCCA, é considerado pequeno.

Com base na análise comparativa dos resultados, apresentados anteriormente, é possível concluir que o atraso médio da arquitetura RT-WiFi é quase constante e previsível (tendendo para $SI/2$ tal como esperado). Além disso, seu valor não sofre variações, nem com o aumento da carga de rede imposta pelas estações NTR, nem como o aumento do número de estações TR admitidas pela arquitetura.

Este comportamento deve-se basicamente a dois factores. O primeiro é o uso do mecanismo FCR (*Force Collision Resolution*). Como as transmissões TR são efetuadas utilizando a fila de voz e para além disto não executam o procedimento de *backoff*, a sua probabilidade de aceder ao meio antes das estações NTR é maior. Desta forma é possível suportar diferentes cargas provenientes da rede NTR sem sofrer alterações significativas no atraso médio.

O segundo factor é o uso do esquema TDMA (*Time Division Multiple Access*). Isto evita que as estações TR concorram entre si para aceder ao meio de comunicação, permitindo assim um

aumento no número de estações TR admitidas sem sofrer variações no atraso médio das transmissões. Além disso, como este esquema distribui o acesso ao meio das estações TR dentro de um SI (*Service Interval*) e, como as estações TR foram inicializadas de forma aleatória e em momentos diferentes, isto faz com que os resultados do atraso médio tendam para $SI/2$.

Na análise do atraso médio apresentado pelo HCCA, é possível observar que, tal como no RT-WiFi, este não sofre grandes variações devido ao aumento da carga de rede imposta pela rede NTR. Isto deve-se ao CFP (*Contention Free Period*) criado pelo seu mecanismo de *polling*, que bloqueia o meio de comunicação para a transmissão das mensagens de tempo-real.

No entanto, este mecanismo impõe o envio de uma mensagem de autorização para que uma estação TR possa iniciar a transmissão das suas mensagens. O envio destas mensagens de autorização pode ser considerado um *overhead* que pode ser observado no comportamento ascendente do atraso médio do HCCA sempre que uma nova estação TR é admitida.

Por fim, com base na análise comparativa dos resultados apresentados, verifica-se que a arquitetura RT-WiFi é capaz de admitir quase 10 vezes mais estações TR que o HCCA. Esta é uma observação importante e que demonstra a escalabilidade da proposta.

5.2.2 Percentagem Média de Deadlines Perdidas

A percentagem média de *deadlines* perdidas para o cenário $P = 30ms$ é apresentada na Figura 5.5. No que diz respeito ao RT-WiFi é possível verificar que, quando a carga imposta pelas estações NTR é considerada *Baixa*, esta métrica situa-se por volta de $\approx 0,5\%$. Quando a carga de rede aumenta até o nível considerado *Médio*, a percentagem média de *deadlines* perdidas também cresce para $\approx 2\%$. Por fim, quando a rede NTR impõe uma carga considerada *Alta*, a percentagem média de *deadlines* perdidas sobe para $\approx 4,25\%$.

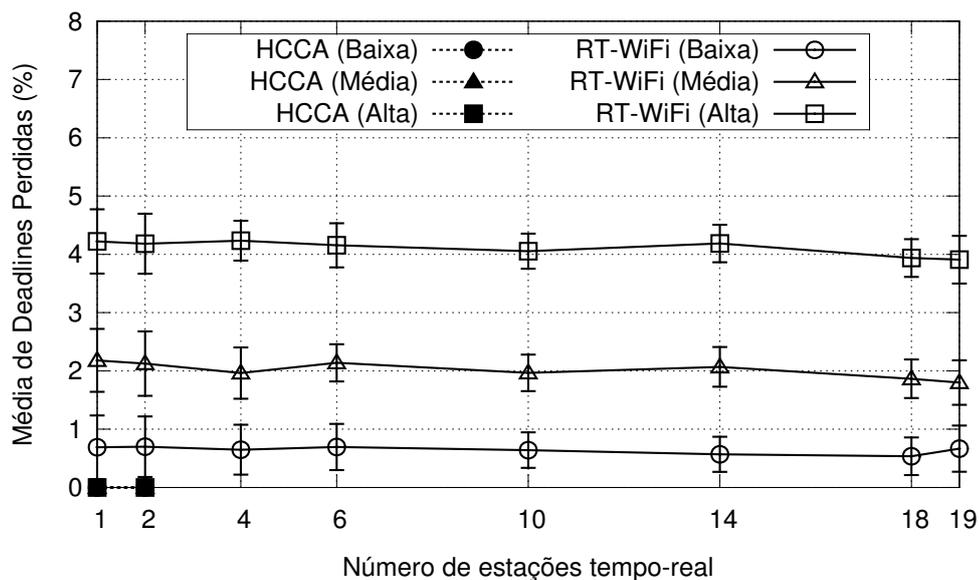


Figura 5.5: Média de *deadlines* perdidas para o cenário $P = 30ms$.

No que diz respeito ao HCCA, é possível verificar que as estações TR não perderam praticamente nenhuma *deadline*. Isto demonstra um comportamento esperado, uma vez que o HCCA bloqueia o acesso ao meio para poder efetuar as suas transmissões sem nenhuma contenção e/ou colisão. Embora possam ocorrer erros devido a ruídos nas transmissões, estes são resolvidos através de retransmissões.

Para estes resultados foram também avaliados os valores do respectivo desvio padrão, os quais são representados em cada ponto do gráfico. Assim, para cada um, é possível observar os níveis de variação superior e inferior em relação à média. Neste contexto, verifica-se que, mesmo ao considerarmos o índice mais alto atingido por uma variação superior (1 estação RT-WiFi sob uma carga de rede *Alta*), o resultado apresentado mantém-se abaixo de 5%.

A mesma métrica é apresentada na Figura 5.6 para o cenário onde $P = 60ms$. No caso do RT-WiFi é possível verificar que, para os níveis de carga impostos pela rede NTR considerados *Baixo* e *Médio*, os resultados para este cenário foram muito semelhantes aos apresentados anteriormente ($\approx 0,5\%$ e $\approx 2\%$, respectivamente). Quando comparados com os resultados anteriores, observa-se uma pequena diminuição na percentagem média de *deadlines* perdidas quando a carga de rede imposta pela rede NTR é considerada *Alta*, situando-se à volta de $\approx 4\%$. Isto deve-se ao aumento no intervalo de geração das mensagens, fazendo com que os requisitos temporais sejam reduzidos.

No que diz respeito ao HCCA, e tal como esperado, é possível verificar o mesmo comportamento apresentado no resultado anterior.

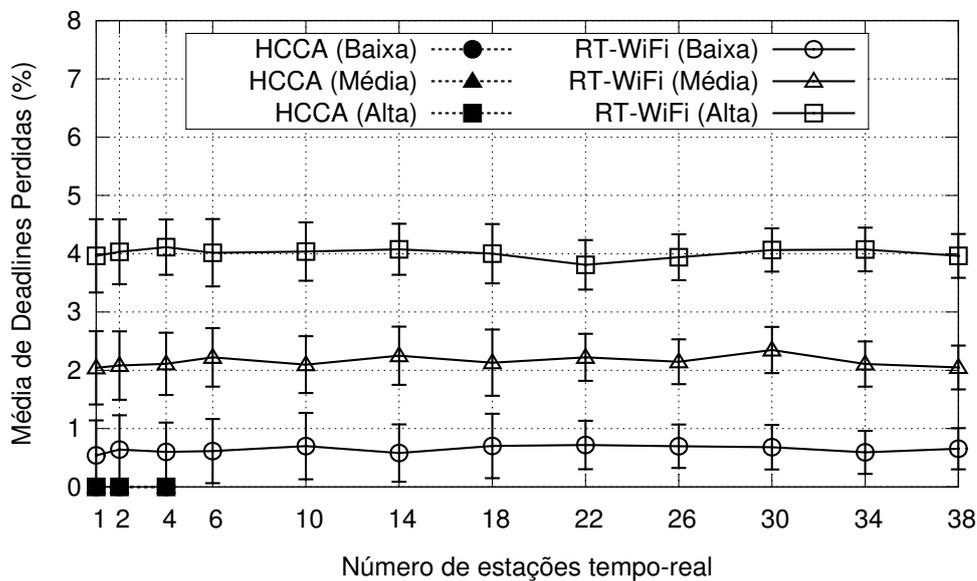


Figura 5.6: Média de *deadlines* perdidas para o cenário $P = 60ms$.

Por fim, são apresentados os resultados para o cenário de $P = 90ms$ (Figura 5.7). Com relação ao RT-WiFi verifica-se que, para o nível de carga de rede considerado *Baixo*, a percentagem média de *deadlines* perdidas não ultrapassa 0,5%. Quando a carga aumenta para o nível considerado *Médio*, a percentagem média de *deadlines* perdidas passa a ser $\approx 1,5\%$. Por fim, quando a carga é considerada *Alta*, a percentagem média de *deadlines* perdidas sobe para $\approx 3,5\%$.

De forma análoga, os resultados obtidos pelo HCCA mantêm a mesma tendência apresentada nos resultados anteriores.

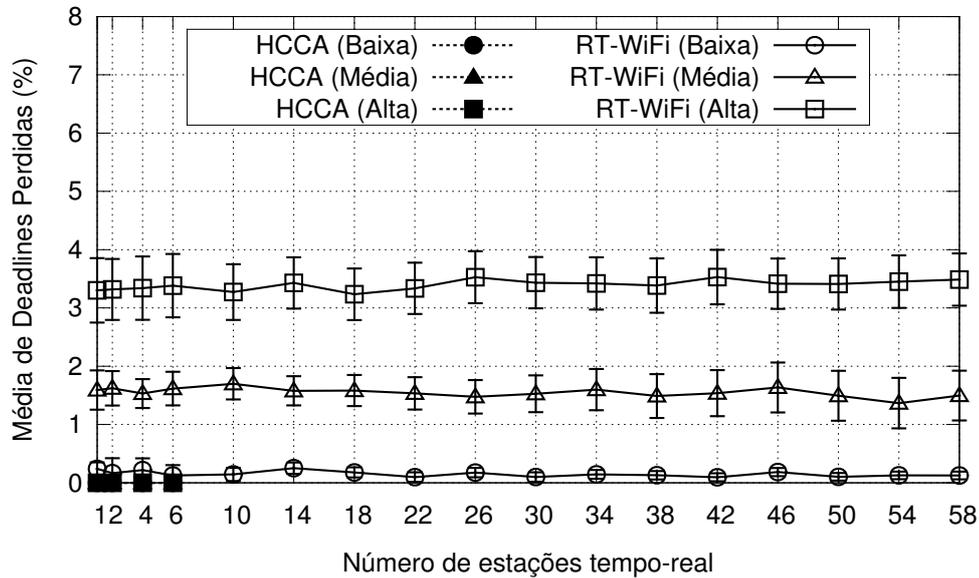


Figura 5.7: Média de *deadlines* perdidas para o cenário $P = 90ms$.

Ao compararmos os resultados obtidos pelo RT-WiFi no cenário de $P = 90ms$ com os obtidos nos cenários anteriores, é possível verificar uma queda na percentagem média de *deadlines* perdidas para todos os 3 tipos de cargas consideradas. Isto deve-se ao aumento do intervalo de geração das mensagens que, por consequência, aumenta também o tempo máximo disponível para a entrega das mesmas.

Com base na análise comparativa dos resultados previamente apresentados, pôde-se concluir que o RT-WiFi apresenta um comportamento semelhante nos três cenários avaliados, não sendo afetado pelo aumento do número de estações TR. Além disso, a percentagem média de *deadlines* perdidas (mesmo ao considerarmos os valores do desvio padrão) encontra-se abaixo do limite de 5% normalmente imposto por sistemas de controlo [114].

Uma análise mais aprofundada destes resultados mostrou que, a principal causa da perda das *deadlines* no RT-WiFi está relacionada com a perda da mensagem de *beacon*. Como esta mensagem é utilizada para sincronizar o ciclo TDMA e difundir a lista de escalonamento, com a sua perda, as estações não iniciam as suas respectivas transmissões.

Um comportamento com leves variações pode ser observado nos resultados apresentados pelo RT-WiFi. Estas variações devem-se ao posicionamento aleatório das estações TR no ambiente de comunicação, uma vez que, para um número diferente de estações em posições geográficas diferentes, poderá incidir à cada uma diferentes níveis de interferência. No entanto, mesmo assim é possível observar que estas variações encontram-se dentro do desvio padrão apresentado.

No que diz respeito aos resultados obtidos pelo HCCA, embora este não tenha perdido praticamente nenhuma *deadline*, o seu mecanismo de controlo de admissão impõe uma grave limitação no que diz respeito ao número máximo de estações TR que podem ser admitidas.

5.2.3 Fairness

Como previamente discutido, embora na comunidade científica exista um consenso acerca do significado do *fairness* nas comunicações (ou seja, um equilíbrio dos direitos de transmissão entre as estações em operação numa determinada rede ou local), o mesmo não ocorre acerca da sua metodologia de quantificação, existindo diversas abordagens propostas [109, 110, 111, 112, 113]. No entanto, com o objetivo de simplificar a sua avaliação, neste trabalho optou-se por analisar o *fairness* com base no *throughput* agregado da rede NTR, quando esta opera sozinha no meio de comunicação, comparando estes resultados com os valores obtidos quando o meio de comunicação é compartilhado com a rede TR.

Desta forma, o *fairness* do RT-WiFi para o cenário de $P = 30ms$ é apresentado na Figura 5.8. A primeira coluna apresenta o *throughput* agregado da rede NTR quando a carga de rede é considerada *Baixa* e sem a presença da rede RT-WiFi. As 3 colunas apresentam o *throughput* agregado da rede NTR, quando uma rede RT-WiFi com 2, 10 e 19 estações está a operar sobreposta, respectivamente. É possível verificar que não ocorre nenhuma variação significativa.

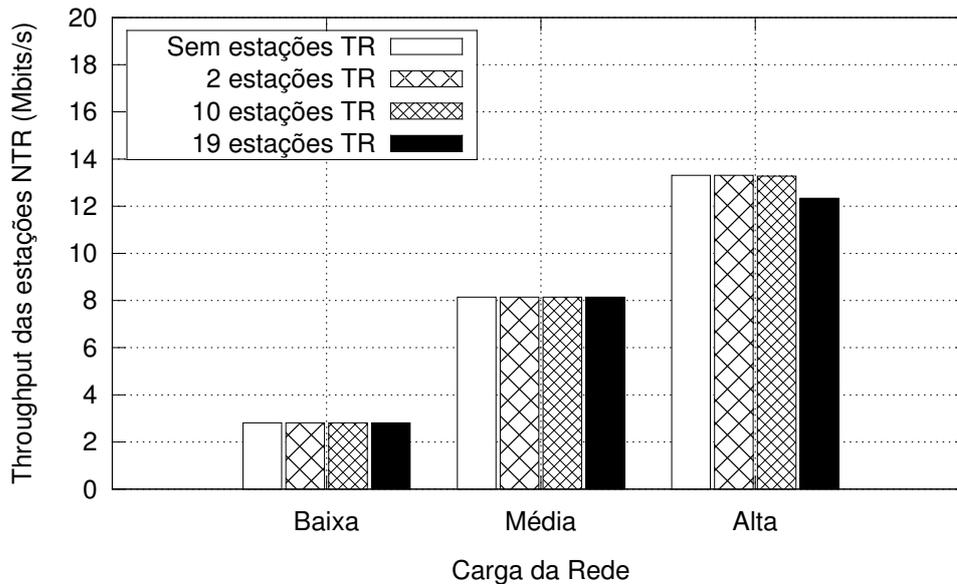


Figura 5.8: *Fairness* para o cenário $P = 30ms$.

A mesma comparação é realizada para as cargas consideradas *Média* e *Alta*. É possível verificar uma pequena variação apenas quando a carga de rede imposta é considerada *Alta* e quando a arquitetura RT-WiFi opera com 19 estações, ou seja, o limite para este cenário. Mesmo assim, a variação apresentada ($< 0,5$ Mbits/s) pode ser considerada pequena.

O *fairness* para o cenário de $P = 60ms$ é apresentado na Figura 5.9. Ao observarmos os resultados obtidos é possível verificar que estes seguem a mesma tendência dos anteriores.

Neste caso em específico, o *throughput* agregado da rede NTR sofre apenas uma pequena diminuição no seu nível, quando o nível de carga de rede é considerado *Alto*. Neste contexto, é possível verificar uma variação mínima, quando o RT-WiFi opera com 22 estações e uma variação maior quando o RT-WiFi opera com 38 estações. No entanto, esta última variação ($< 0,5$ Mbits/s) continua a ser considerada pequena.

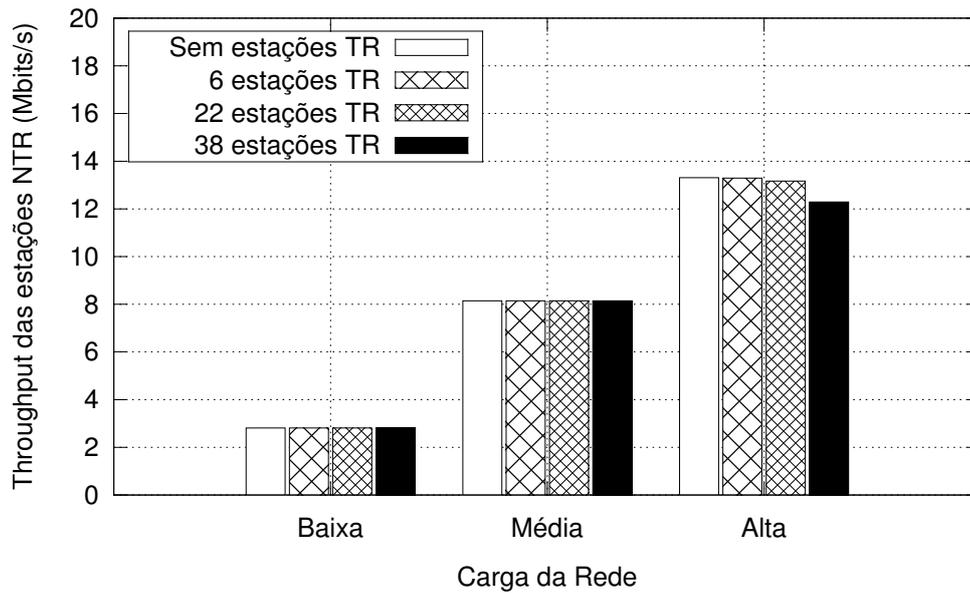


Figura 5.9: *Fairness* para o cenário $P = 60ms$.

No cenário de $P = 90ms$ (Figura 5.10) é possível observar a mesma tendência apresentada nos resultados anteriores, onde as variações no *throughput* agregado da rede NTR ocorrem apenas quando se impõe uma carga considerada *Alta*. A principal diferença é que neste caso a variação sofrida pela rede NTR se inicia quando o RT-WiFi está a operar com 18 estações. Esta variação aumenta conforme ocorre o aumento do número de estações TR. No entanto, embora a variação máxima apresentada seja de 0,9 Mbits/s, esta ainda pode ser considerada pequena, tendo em conta o número de estações TR que foram admitidas pelo RT-WiFi (58 estações).

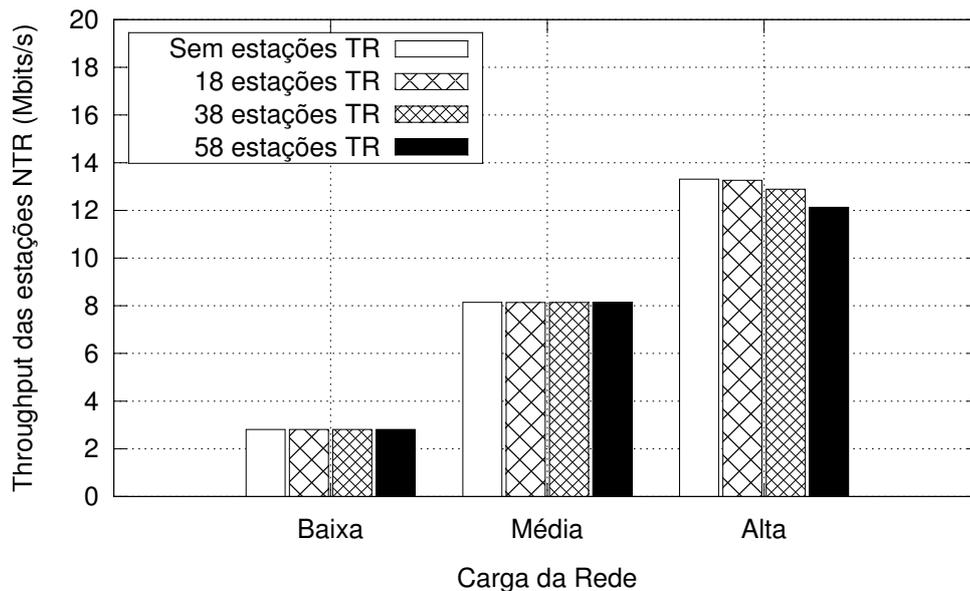


Figura 5.10: *Fairness* para o cenário $P = 90ms$.

Uma análise comparativa dos resultados apresentados anteriormente demonstra que, independentemente de operar no mesmo canal de comunicação que uma rede NTR, o RT-WiFi impõe uma variação mínima sobre o *throughput* agregado desta rede. Desta forma, é possível concluir que, a arquitetura RT-WiFi possui um nível de *fairness* adequado para a sua coexistência com redes NTR. Esta característica deve-se ao facto do RT-WiFi, diferentemente do HCCA, não bloquear o acesso ao meio das estações NTR. Neste contexto, o RT-WiFi permite que as estações NTR transmitam as suas mensagens nos intervalos de tempo entre as transmissões das estações TR.

5.2.4 Tamanho Médio do Slot

Esta métrica tem como objetivo demonstrar o comportamento do mecanismo de redimensionamento dos *slots* sob a dinâmica imposta pelas diferentes cargas de rede. Os resultados desta métrica são apresentados num gráfico com dois eixos Y: o eixo da esquerda apresenta o tamanho do *slot* e o eixo da direita a carga de rede atual. São apenas apresentados os 5 primeiros segundos de simulação, uma vez que a tendência mantém-se no restante período de tempo.

A Figura 5.11 apresenta os resultados para o cenário de $P = 30ms$. Nesta figura o primeiro, segundo e terceiro gráficos representam as cargas de rede *Baixa*, *Média* e *Alta*, respectivamente. É possível verificar que as estações são inicializadas com um *slot* largo (consequência dos valores iniciais definidos pelo RT-WiFi), o qual é rapidamente ajustado para valores mais reduzidos baseados no nível de atraso sofrido pelas suas TS. Neste caso em específico, o tamanho médio dos *slots* para as cargas de rede *Baixa*, *Média* e *Alta* é de 0,45ms, 0,55ms e 0,7ms, respectivamente.

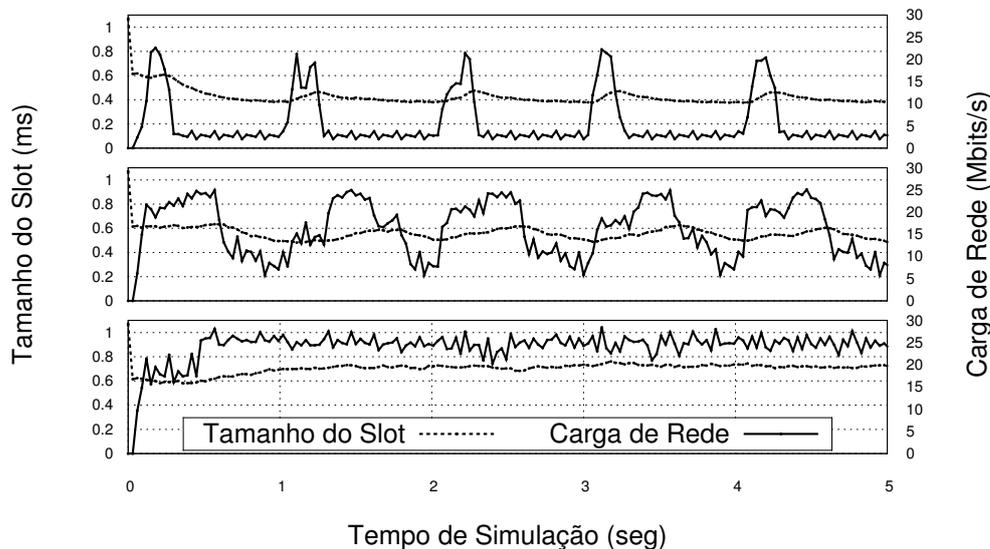


Figura 5.11: Tamanho médio do *slot* para o cenário $P = 30ms$.

Como esperado, cargas da rede mais elevadas resultam em maiores atrasos nas transmissões TR, seja pelo atraso no acesso ao meio, seja pelo aumento da probabilidade de colisões, e consequentemente num aumento no tamanho médio dos *slots* alocados. Além disto, é possível verificar

que este tamanho muda dinamicamente de acordo com as variações da carga da rede. Este comportamento é mais evidente nas cargas de rede *Baixa* e *Média*, onde os picos da carga imposta são mais acentuados.

A mesma tendência pode ser observada no cenário de $P = 60ms$ (Figura 5.12). Neste caso em específico, o tamanho médio dos *slots* para as cargas de rede *Baixa*, *Média* e *Alta* é de 0,5ms, 0,65ms e 0,8ms, respectivamente. Tal como apresentado no resultado anterior, neste cenário é também possível observar que o tamanho médio dos *slots* utiliza como base de ajuste o atraso imposto às TS pelas transmissões NTR, normalmente resultante de um maior nível de ocupação do meio de comunicação.

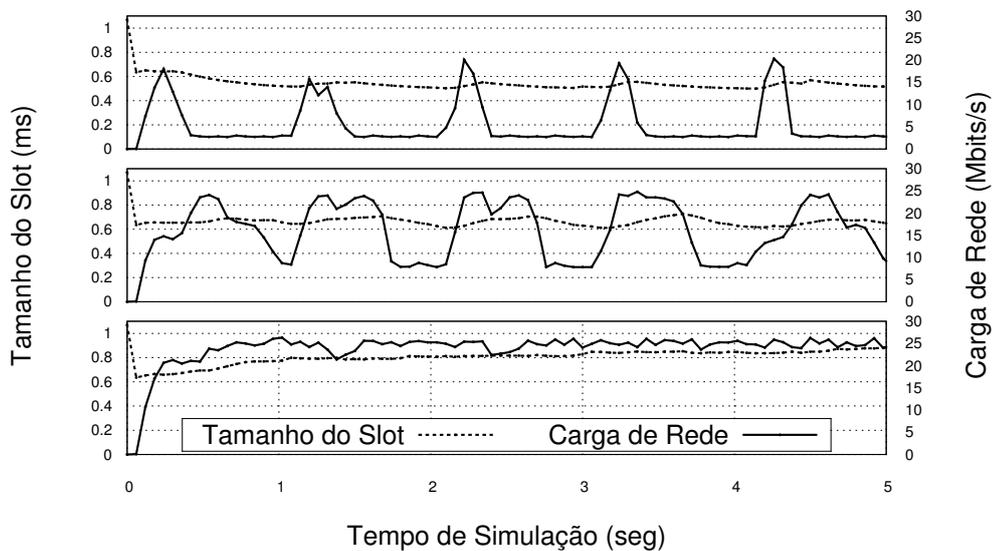


Figura 5.12: Tamanho médio do *slot* para o cenário $P = 60ms$.

A mesma tendência é observada para o cenário de $P = 90ms$ (Figura 5.13). Neste caso específico, o tamanho médio dos *slots* para as cargas *Baixa*, *Média* e *Alta* é de 0,6ms, 0,75ms e 0,85ms, respectivamente.

Uma análise comparativa dos resultados anteriores mostra que, o mecanismo de redimensionamento dinâmico dos *slots* implementado pelo RT-WiFi segue a tendência das diferentes cargas de rede, ajustando os tamanhos dos *slots*, para assim, melhorar o sucesso nas transmissões e otimizar o nível de alocação de recursos efetuado pelo escalonador.

Outra observação importante é a existência de pequenas variações na carga da rede imposta pelas estações NTR, quando comparada a mesma carga nos 3 diferentes cenários. Isto deve-se ao facto de que, os resultados obtidos em cada cenário avaliado foram realizados tendo em conta o número máximo de estações TR que o RT-WiFi poderia admitir. Assim, para os cenários onde o intervalo entre a geração das mensagens TR é de 30, 60 e 90 milissegundos, foram consideradas 19, 38 e 58 estações TR, respectivamente. Esta situação faz com que a carga efetiva da rede num ambiente com uma maior concentração de dispositivos seja menor do que num ambiente onde este número é reduzido. Pode-se observar ainda que para a carga considerada *Alta* praticamente

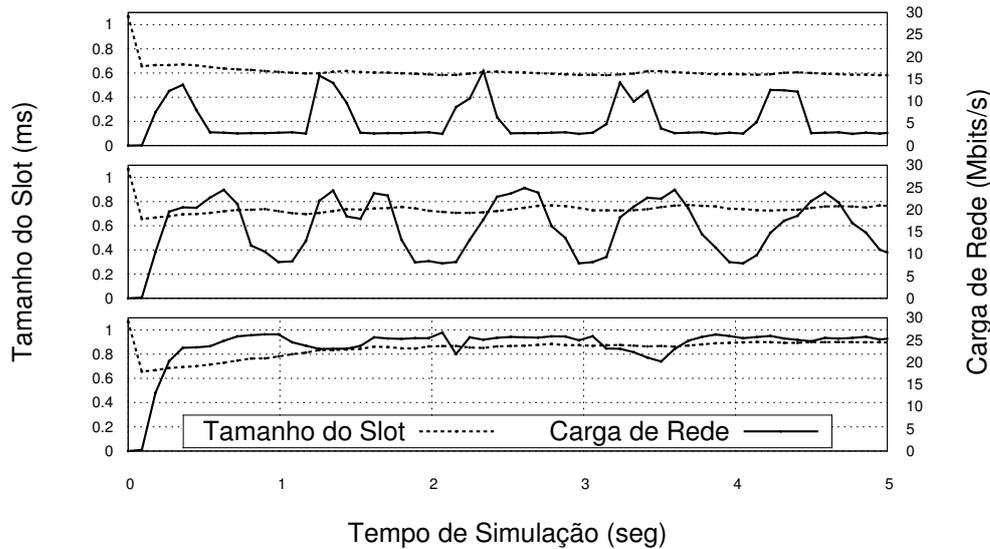


Figura 5.13: Tamanho médio do *slot* para o cenário $P = 90ms$.

não houve variação. Isto deve-se ao facto de que, independentemente do número de estações TR existentes na rede, esta carga encontra-se próxima do ponto de saturação da rede.

No que diz respeito ao tamanho dos *slots* atribuídos à cada TS, é possível observar um aumento crescente no seu valor quando comparados os 3 diferentes cenários. Tal como verificado com a carga de rede imposta pelas estações NTR, esta variação deve-se ao facto de serem considerados para cada cenário diferentes números de estações TR. Embora as restrições temporais sejam mais exigentes num cenário onde o intervalo de geração de mensagens seja menor (por exemplo, $P = 30ms$), o número de dispositivos a operar no mesmo canal de comunicação também é menor (quando comparado aos demais cenários). Isto possibilita uma melhor transmissão das mensagens, diminuindo o nível de colisões entre estações TR e NTR, uma vez que não ocorrem colisões entre estações TR.

5.2.5 Percentagem Média de *Deadlines* Perdidas em função da Carga da Rede

Esta secção tem por objetivo analisar o comportamento da percentagem média de *deadlines* perdidas pelo RT-WiFi em situações onde possam ocorrer mudanças repentinas na carga de rede imposta pelas estações NTR. Esta análise foca principalmente no tempo necessário para que a ACU adapte o tamanho dos *slots* a nova carga de rede imposta. Assim, para analisar este comportamento, é realizado um acréscimo de 10% na carga atual da rede NTR aos 30 segundos de simulação. Para verificar o comportamento do mecanismo de redimensionamento dos *slots* neste instante, são apresentados os 10 segundos anteriores e posteriores ao ponto de mudança. Os resultados apresentados por esta métrica têm dois eixos Y: o eixo da esquerda mostra a percentagem média de *deadlines* perdidas e o da direita a carga atual de rede imposta pelas estações NTR.

A Figura 5.14 apresenta os resultados para o cenário de $P = 30ms$, sendo que o primeiro, segundo e terceiro gráficos representam as cargas de rede *Baixa*, *Média* e *Alta*, respectivamente. Como esperado, é possível observar um pequeno aumento na percentagem média de *deadlines*

perdas logo após o aumento da carga de rede imposta pelas estações NTR (aos 30 segundos). No entanto, para todas as cargas de rede, esta variação é inferior a 1%. Além disso, observa-se ainda que mesmo com o aumento do número de estações TR, a percentagem média de *deadlines* perdidas mantém-se estável (as linhas estão sobrepostas), seguindo assim as tendências apresentadas na secção 5.2.2.

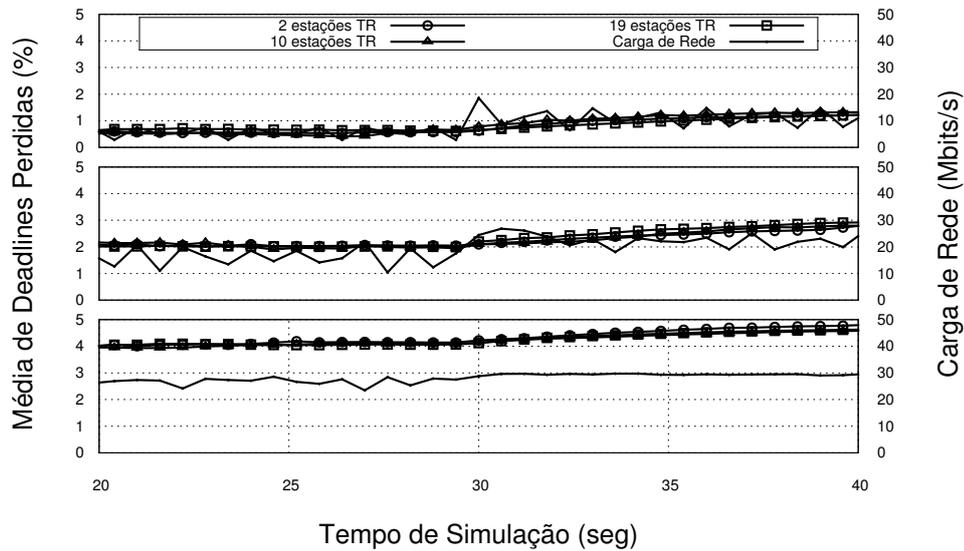


Figura 5.14: Percentagem média de *deadlines* perdidas para o cenário $P = 30ms$.

A Figura 5.15 apresenta os resultados para o cenário de $P = 60ms$. Verificam-se as mesmas tendências dos resultados anteriores. Como esperado, há um ligeiro aumento ($< 1\%$) na percentagem média de *deadlines* perdidas após o aumento na carga da rede NTR. Observa-se também que estes valores não são afetados de forma significativa pelo aumento do número de estações TR.

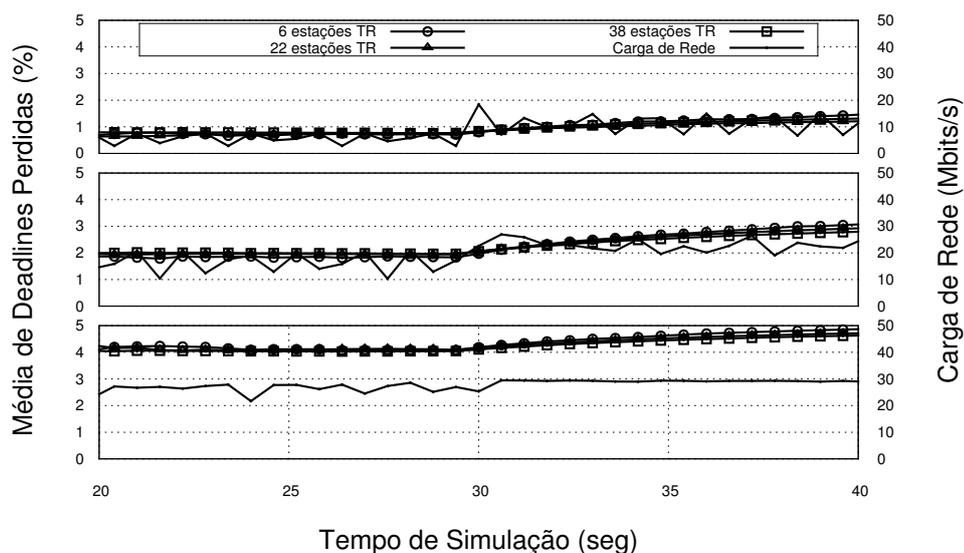


Figura 5.15: Percentagem média de *deadlines* perdidas para o cenário $P = 60ms$.

Por fim, a Figura 5.16 apresenta os resultados para o cenário de $P = 90ms$. Estes, por sua vez, reforçam as tendências apresentadas anteriormente. Neste contexto observa-se que, o aumento da percentagem média de *deadlines* perdidas após o aumento na carga da rede NTR mantém-se abaixo de 1%, e que estes resultados não sofrem variações significativas em detrimento do aumento do número de estações TR.

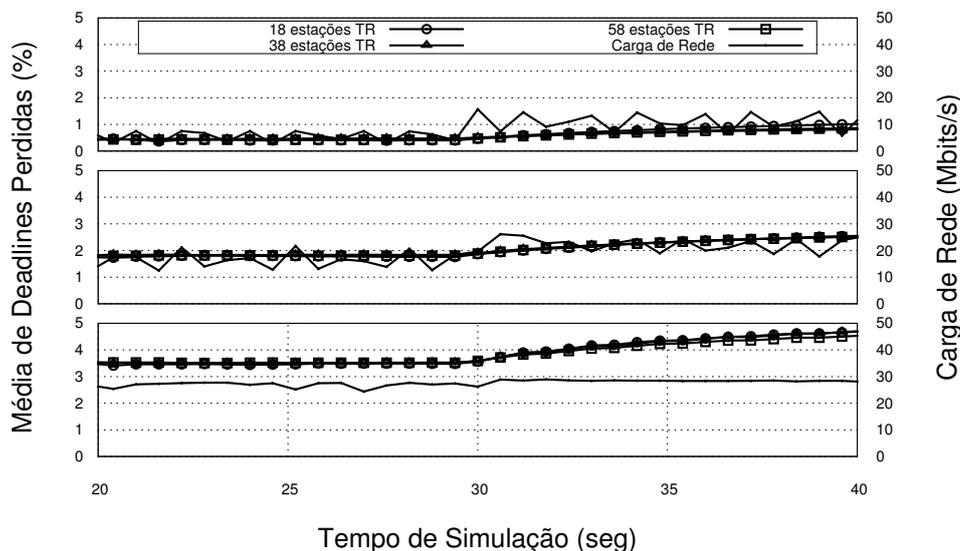


Figura 5.16: Percentagem média de *deadlines* perdidas para o cenário $P = 90ms$.

A análise comparativa dos resultados anteriores demonstra que, embora a percentagem média de *deadlines* perdidas sofra um ligeiro aumento em decorrência do aumento da carga de rede imposta pelas estações NTR, esta não é suficiente para prejudicar o funcionamento do RT-WiFi como um todo, uma vez que os seus valores são inferiores ao limite de 5%, geralmente admissíveis pelos sistemas de controlo. Além disso, também é possível afirmar que o aumento no número de estações de TR admitidas na rede RT-WiFi não influencia de forma significativa o aumento da percentagem média de *deadlines* perdidas.

Esta característica deve-se basicamente ao facto de que, com o aumento da carga de rede imposta pelas estações NTR, as estações TR também sofrerão um aumento no atraso das suas transmissões (seja pelo atraso no acesso ao meio, seja pelo aumento do número de colisões). Em contrapartida, para compensar esta situação, a ACU redimensiona o tamanho dos *slots* alocados a cada TS admitida, de forma a aumentar a probabilidade de sucesso das respectivas transmissões. Este aumento no tamanho dos *slots* evita um aumento demasiado na percentagem média de *deadlines* perdidas, estabilizando este valor dentro de uma faixa admissível para os sistemas de controlo.

5.3 Conclusões

Neste capítulo avaliou-se a arquitetura RT-WiFi a operar num ambiente de comunicação aberto. Para possibilitar uma base de comparação foi também avaliada a função HCCA (definida pela

norma IEEE 802.11), sendo ambos os modelos implementados na ferramenta de simulação OP-NET *Modeler Wireless*. Assim, para avaliar o impacto do tráfego NTR sobre o tráfego TR, foram considerados cenários onde a rede de tempo-real está a operar na mesma área de cobertura e canal de comunicação de uma rede NTR. Além disso, para todos os cenários avaliados foi também considerada a possibilidade de existirem erros no canal de comunicação.

A análise dos resultados demonstrou que:

- O atraso médio do RT-WiFi é praticamente constante e independente da carga de rede imposta pelas estações NTR ou do número de estações TR admitidas;
- O atraso médio do RT-WiFi é previsível, tendendo para $SI/2$;
- O *jitter* das comunicações do RT-WiFi é reduzido, podendo ser este observado de forma indireta através do desvio padrão ($< 5\%$) apresentado nos resultados do atraso médio;
- A percentagem média de *deadlines* perdidas pelo RT-WiFi é estável e sofre variações mínimas com o aumento da carga de rede imposta pelas estações NTR. Além disso, mesmo em situações onde a carga de rede é considerada *Alta*, a média de *deadlines* perdidas é inferior ao limite de 5% geralmente admissível em sistemas de controlo;
- A percentagem média de *deadlines* perdidas pelo RT-WiFi não sofre alteração com o aumento do número de estações de TR admitidas, de modo que a expansão da rede não degrada a qualidade das transmissões (do ponto de vista do cumprimento das *deadlines*);
- O RT-WiFi foi capaz de admitir quase 10 vezes mais estações de TR do que o HCCA, nos cenários avaliados;
- Devido a utilização de um esquema TDMA, onde as estações NTR podem efetuar as suas transmissões entre os *slots* alocados pela arquitetura RT-WiFi, esta por sua vez, possui um nível adequado de *fairness*, que torna a sua operação compatível com dispositivos IEEE 802.11 num mesmo ambiente de comunicação.

Através da análise destes pontos, é possível concluir que a arquitetura RT-WiFi reúne as condições necessárias para suportar a transmissão de tráfego de tempo-real num ambiente de comunicação aberto. O comportamento quase constante e previsível do atraso médio das suas transmissões é uma característica desejável em serviços de comunicação que suportam aplicações de tempo-real.

Além disso, o mecanismo de redimensionamento de *slots* garante uma melhor utilização dos recursos da rede e também acompanha o comportamento dinâmico dos atrasos impostos a cada TS, sejam estes provenientes de interferências ou da ocupação do meio de comunicação por transmissões da rede NTR.

O esquema TDMA utilizado pela arquitetura RT-WiFi garante também um *fairness* adequado entre transmissões da rede RT-WiFi e de outras redes NTR que estejam a operar na mesma área de cobertura e canal de comunicação. Esta abordagem evita o bloqueio no acesso ao meio (*starvation*) das estações NTR.

Por fim, é possível observar que o conjunto de mecanismos implementados pela arquitetura RT-WiFi consegue, garantir uma alta taxa de cumprimento das *deadlines* (superior a 95%), além de ser capaz de admitir quase 10 vezes mais estações de TR que a função HCCA sem que para isto seja necessário o bloqueio do acesso ao meio pela rede TR.

Capítulo 6

Conclusões e Trabalhos Futuros

Este capítulo resume os principais resultados obtidos nesta tese, destacando as contribuições resultantes da investigação, assim como também apresentada algumas perspectivas para trabalhos de investigação futuros que possam surgir a partir deste trabalho.

6.1 Conclusões

O objetivo principal desta tese foi propor uma solução que permitisse a transmissão de tráfego de tempo-real em redes sem fios a operar em ambientes de comunicação abertos, nomeadamente industriais. Neste contexto, optou-se por dar ênfase à soluções compatíveis com a norma IEEE 802.11, uma vez que esta se tornou um standard *de facto* para a implementação de WLAN (*Wireless Local Network*).

Baseada na análise da norma IEEE 802.11 foi possível identificar várias limitações no que diz respeito à sua utilização para a transmissão de tráfego de tempo-real. A compreensão detalhada dos seus modos de funcionamento, bem como das respectivas limitações, serviram como base para a definição dos pré-requisitos que a solução proposta deveria contemplar. Complementarmente, foram também analisadas, classificadas e comparadas diversas soluções propostas para a transmissão de tráfego de tempo-real sobre redes IEEE 802.11.

No que diz respeito às soluções focadas no *Mecanismo de Controlo de Acesso ao Meio*, foi possível definir uma classificação em três eixos. O primeiro define a arquitetura (centralizada ou distribuída) utilizada. Neste contexto, concluiu-se que a utilização de uma arquitetura centralizada seria capaz de fornecer melhores resultados no que diz respeito à gestão das transmissões de tempo-real, uma vez que a solução teria uma visão global do ambiente de comunicação.

O segundo eixo de classificação define a forma como as soluções tratam as colisões. Neste contexto, foram definidas 3 categorias: soluções que evitam colisões, que resolvem as colisões e que reduzem a ocorrência de colisões. Concluiu-se que as soluções mais adequadas são aquelas

que tentam resolver ou evitar as colisões, uma vez que é possível gerar um comportamento mais previsível na transmissão das mensagens de tempo-real.

Por fim, o terceiro eixo de classificação define a forma como as soluções se comportam na presença de estações IEEE 802.11 standard a operar em redes sobrepostas à rede de tempo-real, além de analisar também, a possibilidade de implementação das soluções em *hardware* COTS (*Commercial Off-The-Shelf*).

Através desta avaliação, ficou claro que existem poucas soluções que possibilitam a coexistência de dispositivos de tempo-real e dispositivos IEEE 802.11 standard no mesmo ambiente de comunicação. A maioria das soluções analisadas necessita de um controlo completo do ambiente de comunicação, ou seja, todas as estações que estejam a operar no mesmo canal de comunicação e área de cobertura devem obrigatoriamente estar dentro da esfera de controlo da arquitetura de tempo-real. Porém, com a crescente utilização de dispositivos de comunicação sem fio, não é realista assumir-se atualmente a possibilidade de criar um ambiente de comunicação fechado, uma vez que o meio de comunicação utilizado por este tipo de tecnologia é compartilhado. Desta forma, as soluções que garantem as características de tempo-real através do controlo de todos os dispositivos de comunicação existentes no ambiente não são consideradas aplicáveis.

No que diz respeito ao *Mecanismo de Controlo de Admissão*, foi definida uma classificação em 2 eixos para a análise das soluções existentes. O primeiro eixo define a abordagem utilizada pelo modelo. Esta, por sua vez, é classificada em 3 diferentes categorias: **a)** baseadas em modelos, **b)** baseadas em métricas e, **c)** híbridas. Neste contexto, concluiu-se que a utilização de uma abordagem híbrida seria a melhor opção, uma vez que, permite um comportamento dinâmico por parte do mecanismo de controlo de admissão que, além de utilizar informações TSPEC para decidir entre a admissão (ou não) de uma TS (*Traffic Stream*), pode também utilizar dados provenientes do meio de comunicação e também das estações.

O segundo eixo define a arquitetura (centralizada ou distribuída) utilizada pelas soluções. Concluiu-se que a utilização de uma arquitetura centralizada seria capaz de fornecer melhores resultados no que diz respeito à gestão das transmissões de tempo-real, uma vez que a ACU (*Admission Control Unit*) possui uma visão global do ambiente de comunicação. Esta conclusão, por sua vez, vem de encontro a obtida na análise das soluções relativas aos mecanismos de controlo de acesso ao meio.

Além destes dois eixos de classificação, da análise das soluções focadas no *Mecanismo de Controlo de Admissão* resultou numa lista de características apresentadas por cada um: suporte aos tráfegos CBR (*Constant Bit Rate*), VBR (*Variable Bit Rate*) e aperiódico, utilização (pela ACU) de informações TSPEC (*Traffic Specification*) e de medidas do meio de comunicação para o auxílio à tomada de decisão, utilização de um escalonador baseado nas *deadlines* das mensagens e tipo de mecanismo de controlo de acesso ao meio sobre o qual as soluções foram construídas.

Partindo da análise de todas estas propostas, foi possível definir uma lista de pré-requisitos necessários numa nova solução capaz de suportar a transmissão de tráfego de tempo-real em redes IEEE 802.11:

- ser capaz de garantir requisitos *soft real-time*, mesmo quando estiver a operar num ambiente de comunicação aberto, onde o meio de comunicação é compartilhado com estações que estão fora da esfera de controlo da arquitetura de tempo-real;
- implementação compatível com *hardware* COTS;
- mecanismo de controlo de acesso ao meio capaz de:
 - *resolver* ou *evitar* as colisões, uma vez que ambas as abordagens visam garantir limites temporais para a sua resolução;
 - reduzir o *overhead* gerado pelos mecanismos de *polling* tradicionais;
- mecanismo de controlo de admissão capaz de:
 - utilizar uma abordagem *híbrida*, obtendo tanto informações das TS (via TSPEC), quanto do meio de comunicação para analisar o pedido de admissão de uma nova TS;
 - gerir as variações que possam ocorrer nos valores medidos no meio de comunicação e/ou nas estações;
 - caracterizar as TS através do uso de informações TSPEC;
 - implementar um algoritmo de escalonamento de tempo-real para organizar a sequência de transmissão das mensagens com base nas respectivas *deadlines*;
 - identificar e remover as TS que possam se encontrar num estado de "falha", ou seja, que estejam com recursos alocados mas que não estejam em operação.

Neste contexto, foi proposta uma nova arquitetura para comunicação de tempo-real em redes IEEE 802.11, denominada RT-WiFi. Esta arquitetura, organizada em duas camadas, propõe dois novos mecanismos de *Controlo de Acesso ao Meio* e de *Controlo de Admissão* com o objetivo de garantir o cumprimento das *deadlines* específicas de cada TS admitida pela ACU, mesmo quando o meio de comunicação é partilhado com estações que estejam fora da sua esfera de controlo.

O *Mecanismo de Controlo de Acesso ao Meio* implementa um esquema de separação de tráfego que permite a priorização do tráfego de tempo-real sobre os restantes tipos de tráfego sem que para isto seja necessário controlar todas as estações que estejam a operar no ambiente de comunicação. Além disso, implementa também um esquema TDMA para evitar colisões entre mensagens de tempo-real. A utilização deste esquema possibilita também a redução do *overhead* da transmissão, resultante do envio das mensagens de autorização para as estações, quando comparado ao mecanismo tradicional de *polling* utilizado pela grande maioria das soluções existentes.

Sobre este mecanismo é implementado o *Mecanismo de Controlo de Admissão*. A sua implementação assume uma abordagem híbrida, onde, além de utilizar os requisitos temporais especificados por cada TS (através do envio de mensagens TSPEC), permite também alterar dinamicamente o seu comportamento, através da constante análise dos atrasos sofridos por cada transmissão de tempo-real. Isto permite à ACU redimensionar os *slots* alocados a cada TS de forma a otimizar

a alocação dos recursos do sistema. Outra função implementada pela ACU é a de identificar a possibilidade de uma TS se encontrar num estado de "falha" e, então removê-la da lista de alocação de forma a libertar os recursos por ela reservados. O mecanismo implementa também um algoritmo de escalonamento que reordena, a cada ciclo TDMA, a sequência de alocação dos *slots* de forma a cumprir as *deadlines* das mensagens de tempo-real.

Os resultados obtidos pela arquitetura RT-WiFi mostram claramente que esta é capaz de garantir uma elevada probabilidade de sucesso nas transmissões do tráfego de tempo-real. Uma característica importante é que, para os cenários avaliados, independentemente da carga de rede imposta pelas estações que estão fora da esfera de controlo da arquitetura de tempo-real, ou do número de estações existentes na rede de tempo-real, o atraso médio das transmissões no RT-WiFi é aproximadamente constante e tem um comportamento previsível (tendendo para $SI/2$).

Outra característica importante é a sua capacidade de suportar um número elevado de estações de tempo-real. Os resultados obtidos mostram claramente que mesmo para o cenário mais exigente (onde a carga de rede imposta pelas estações não tempo-real se encontra perto do ponto de saturação e onde é atingido o limite máximo de estações de tempo-real admitidas pelo controlo de admissão), a arquitetura RT-WiFi é capaz de suportar a transmissão do tráfego de tempo-real e manter uma percentagem média de perdas de *deadlines* inferior aos 5% geralmente admissíveis nos sistemas de controlo.

A arquitetura RT-WiFi apresenta também um nível adequado de *fairness*, pois a sua operação não resulta em interferências significativas sobre as restantes estações que estão fora da esfera de controlo da arquitetura de tempo-real. Embora a arquitetura RT-WiFi tenha sido concebida com o foco em ambientes industriais, esta última característica em particular possibilita expandir a sua operação para ambientes domésticos e empresariais. Adicionalmente, a implementação da arquitetura RT-WiFi pode ser realizada através de pequenas modificações no *driver/firmware* dos dispositivos IEEE 802.11 já existentes, tornando-a assim compatível com *hardware* COTS.

Por fim, importa também referir, como contribuição desta tese, a implementação de um modelo de simulação da função HCCA (*HCF Controlled Channel Access*) para a ferramenta OPNET. Este modelo teve como objetivo a geração de resultados comparativos com a arquitetura RT-WiFi. Além disso, também foi possível entender as limitações do HCCA, quando este é utilizado para suportar tráfego de tempo-real. A principal conclusão obtida foi que, embora o HCCA possa ser compatível com a transmissão de alguns tipos de tráfego de tempo-real (nomeadamente CBR), este sofre uma grave limitação no que diz respeito ao número máximo de estações que podem ser admitidas pelo seu mecanismo de controlo de admissão.

6.2 Trabalhos Futuros

Os trabalhos futuros que surgem a partir desta tese estão intimamente relacionados com a arquitetura RT-WiFi. Em primeiro lugar, a utilização da mensagem de *beacon* para o envio da lista de escalonamento deve ser cuidadosamente avaliada. Como a sua transmissão é realizada em *broadcast* (ou seja, sem confirmação), estações que estejam fora da área de cobertura geográfica

do AP (*Access Point*) ou que recebam a mensagem de *beacon* corrompida não poderão iniciar as suas transmissões. Como consequência, as TS alocadas nestas estações perderão as respectivas *deadlines*. Neste contexto, devem ser propostas técnicas que efetuem a transmissão da mensagem de *beacon* de forma mais eficiente, como por exemplo *reliable broadcast* ou *reliable multicast*. Outra possibilidade seria desenvolver um mecanismo de recuperação (por parte das estações) da lista de escalonamento quando a mensagem de *beacon* é perdida.

Como o AP_{TR} pode ser considerado um ponto único de falha, é interessante analisar a arquitetura RT-WiFi do ponto de vista de tolerância a falhas. Neste contexto, poderiam ser propostos mecanismos de recuperação do AP_{TR} (por exemplo, *shadowing*) e analisar o seu comportamento em diferentes situações de falha de modo a avaliar o seu impacto na perda das *deadlines* das mensagens.

A proposta realizada nesta tese apresenta a arquitetura RT-WiFi a operar numa única BSS (*Basic Service Set*). No entanto, poderiam ser propostas soluções onde múltiplas redes RT-WiFi poderiam trocar mensagens entre si através do DS (*Distribution System*). Neste contexto, para garantir o correto escalonamento das mensagens com origem numa das redes e transmitida para a outra, seria necessário propor um mecanismo capaz de realizar um escalonamento centralizado destas transmissões. De forma análoga, poderiam ser propostas soluções para situações onde duas (ou mais) redes RT-WiFi sem qualquer ligação lógica entre si estejam a operar na mesma área de cobertura geográfica e canal de comunicação.

Outra avaliação que poderia ser realizada seria a análise de uma forma mais robusta da arquitetura RT-WiFi. Neste contexto, e de forma semelhante ao mecanismo de *polling*, a mensagem de *beacon* criaria um período livre de contenção (CFP – *Contention Free Period*) para o envio da lista de escalonamento e também onde os *slots* das estações de tempo-real seriam alocados. De forma a garantir o controlo do meio de transmissão, todas as mensagens de tempo-real transmitidas neste período transportam no respectivo campo *duration* do seu cabeçalho um valor condizente com o fim do CFP. A vantagem desta abordagem está na redução do número de *deadlines* perdidas, e também numa redução dos recursos alocados a cada TS, possibilitando uma expansão do número máximo de TS admitidas.

No contexto da análise do atraso sofrido pelas estações de tempo-real para iniciarem suas respectivas transmissões, torna-se interessante realizar testes com diferentes valores de α para otimizar a janela deslizante utilizada para suavizar as variações na rede. Além disso, outros mecanismos de caracterização de tendência comportamental poderiam ser propostos e avaliados.

Por fim, podemos concluir que a transmissão de tráfego de tempo-real (nomeadamente *soft real-time*) em redes sem fios apresenta-se como uma tendência natural a ser adotada pelas indústrias. O nível de exigência das aplicações que utilizarão estes sistemas de comunicação crescerá acompanhando os níveis de desempenho que estes possam proporcionar. O aumento da confiabilidade e disponibilidade destas soluções pode ser atingido através da coordenação de esforços empregues tanto na camada física (reduzindo o número de erros de transmissão) quanto na subcamada MAC (coordenando o acesso ao meio e a alocação de recursos). Porém, outro aspecto que se aponta como fundamental para a adoção destas soluções em larga escala diz respeito à sua

segurança. Uma vez que estas soluções operam num ambiente de comunicação aberto estão mais suscetíveis às tentativas de ataque. Desta forma, garantir requisitos de confidencialidade, integridade, autenticidade e disponibilidade dos fluxos de dados será tão importante quanto a garantia dos seus requisitos temporais.

Referências

- [1] IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Junho 1997.
- [2] IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.
- [3] Tomoko Miyano, Shinya Otsuki, Makoto Umeuchi, e Mamoru Ogasawara. Admission and Traffic Control Techniques for WLANs. Special Feature: QoS Control Techniques for Quality Improvement in Wireless Local Area Networks 11, NTT Access Network Service Systems Laboratories, Novembro 2007.
- [4] IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2012.
- [5] G. Cena, I. C. Bertolotti, A. Valenzano, e C. Zunino. Evaluation of response times in industrial WLANs. *IEEE Transactions on Industrial Informatics*, 3(3):191–201, 2007.
- [6] T. Sauter. The continuing evolution of integration in manufacturing automation. *IEEE Industrial Electronics Magazine*, 1(1):10–19, 2007.
- [7] P. Ramanathan. Overload management in real-time control applications using (m, k)-firm guarantee. *IEEE Transactions on Parallel and Distributed Systems*, 10(6):549–559, 1999.
- [8] L. Lo Bello, G. A. Kaczynski, e O. Mirabella. Improving the real-time behavior of ethernet networks using traffic smoothing. *IEEE Transactions on Industrial Informatics*, 1(3):151–161, 2005.
- [9] K. Kopetz. The time-triggered model of computation. Em *Proceedings of the 19th IEEE Real-Time Systems Symposium (RTSS)*, pp. 168–177, 1998.
- [10] R. Moraes, P. Portugal, F. Vasques, e R. F. Custódio. Assessment of the IEEE 802.11e EDCA Protocol Limitations when Dealing with Real-Time Communication. *EURASIP Journal on Wireless Communications and Networking*, 2010.
- [11] G. Cena, L. Seno, A. Valenzano, e C. Zunino. On the Performance of IEEE 802.11e Wireless Infrastructures for Soft-Real-Time Industrial Applications. *IEEE Transactions on Industrial Informatics*, 6(3):425–437, 2010.

- [12] G. Cena, A. Valenzano, C. Zunino, e L. Seno. Evaluation of real-time communication performance in QoS-enabled infrastructure WLANs. Em *Proceedings of the 14th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1149–1155, 2009.
- [13] G. Gamba, L. Seno, e S. Vitturi. Theoretical and experimental evaluation of polling times for wireless industrial networks using commercially available components. Em *Proceedings of the 15th IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–8, 2010.
- [14] Supplement to 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: Higher-Speed Physical Layer Extension in the 2.4 GHz Band, Setembro 1999.
- [15] Supplement to 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: High-speed Physical Layer in the 5 GHz Band, Setembro 1999.
- [16] IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band, Junho 2003.
- [17] 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.
- [18] 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 5: Enhancements for Higher Throughput, Outubro 2009.
- [19] D. Gao, J. Cai, e K. N. Ngan. Admission control in IEEE 802.11e wireless LANs. *IEEE Network*, 19(4):6–13, 2005.
- [20] R. Moraes, P. Portugal, e F. Vasques. Simulation analysis of the IEEE 802.11e EDCA protocol for an industrially-relevant real-time communication scenario. Em *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 202–209, 2006.
- [21] C. Casetti, C. F. Chiasserini, M. Fiore, e M. Garetto. Notes on the Inefficiency of 802.11e HCCA. Em *Proceedings of the 62nd IEEE Vehicular Technology Conference*, volume 4, pp. 2513–2517, Setembro 2005.
- [22] P. Garg, R. Doshi, R. Greene, M. Baker, M. Malek, e X. Cheng. Using IEEE 802.11e MAC for QoS over Wireless. Em *IEEE International Performance, Computing, and Communications Conference*, pp. 537–542, 2003.

- [23] J. D. Decotignie. Ethernet-Based Real-Time and Industrial Communications. *Proceedings of the IEEE*, 93(6):1102–1117, 2005.
- [24] J. Son, I. G. Lee, H. J. Yoo, e S. C. Park. An effective polling scheme for IEEE 802.11e. *IEICE Transactions on Communications*, E88.B(12):4690–4693, 2005.
- [25] D. Miorandi e S. Vitturi. Analysis of master-slave protocols for real-time-industrial communications over IEEE 802.11 WLANs. Em *Proceedings of the 2nd IEEE International Conference on Industrial Informatics (INDIN)*, pp. 143–148, Junho 2004.
- [26] S.C. Lo, G. Lee, e W.T. Chen. An efficient multipolling mechanism for IEEE 802.11 wireless LANs. *IEEE Transactions on Computers*, pp. 764–778, 2003.
- [27] S. Lee, K.N. Ha, J.H. Park, K.C. Lee, G.S. Byun, e H.K. Lee. NDIS-based virtual polling algorithm for IEEE 802.11b for guaranteeing the real-time requirements. *Computer Standards & Interfaces*, 29(3):316–324, 2007.
- [28] F. De Pellegrini, D. Miorandi, S. Vitturi, e A. Zanella. On the use of wireless networks at low level of factory automation systems. *IEEE Transactions on Industrial Informatics*, 2(2):129–143, 2006.
- [29] A. Willig. A MAC protocol and a scheduling approach as elements of a lower layers architecture in wireless industrial LANs. Em *Proceedings of the IEEE International Workshop on Factory Communication Systems (WFCS)*, pp. 139–148, 1997.
- [30] S. Hantrakoon e A. Phonphoem. Priority Based HCCA for IEEE 802.11e. Em *Proceedings of the International Conference on Communications and Mobile Computing (CMC)*, volume 3, pp. 485–489, Abril 2010.
- [31] J. Kiszka, B. Wagner, Y. Zhang, e J. Broenink. RTNet - A flexible hard real-time networking framework. Em *Proceedings of the 10th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 19–22, 2005.
- [32] G. Boggia, P. Camarda, L.A. Grieco, e G. Zacheo. An experimental evaluation on using TDMA over 802.11 MAC for Wireless Networked Control Systems. Em *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1157–1160, 2008.
- [33] Xenomai: Real-Time Framework for GNU/Linux, 2010. URL: <http://www.xenomai.org>.
- [34] RaLink Technologies, 2010. URL: <http://www.ralinktech.com/>.
- [35] L. Seno, S. Vitturi, e C. Zunino. Analysis of Ethernet Powerlink Wireless Extensions Based on the IEEE 802.11 WLAN. *IEEE Transactions on Industrial Informatics*, 5(2):86–98, Maio 2009.
- [36] Ethernet Powerlink Standardization Group: Ethernet Powerlink Communication Profile Specification, 2004.
- [37] P. Bartolomeu, J. Ferreira, e J. Fonseca. Enforcing Flexibility in Real-Time Wireless Communications: A Bandjacking Enabled Protocol. Em *Proceeding of the 14th IEEE Conference on Emerging Technologies Factory Automation (ETFA)*, pp. 1–4, Setembro 2009.

- [38] FTT. FTT Paradigm., Janeiro 2011. URL: <http://www.ieeta.pt/lse/ftt/>.
- [39] L. Almeida, P. Pedreiras, e J.A.G. Fonseca. The FTT-CAN protocol: Why and how. *IEEE Transactions on Industrial Electronics*, 49(6):1189–1201, 2002.
- [40] P. Pedreiras, L. Almeida, e P. Gai. The FTT-Ethernet Protocol: Merging Flexibility, Timeliness and Efficiency. Em *Proceedings of the 14th Euromicro Conference on Real-Time Systems*, pp. 152. IEEE Computer Society, 2002.
- [41] R. G. Cheng, C. Y. Wang, L. H. Liao, e J. S. Yang. Ripple: a wireless token-passing protocol for multi-hop wireless mesh networks. *IEEE Communications Letters*, 10(2):123–125, Fevereiro 2006.
- [42] M. Ergen, D. Lee, R. Sengupta, e P. Varaiya. WTRP - Wireless Token Ring Protocol. *IEEE Transactions on Vehicular Technology*, 53(6):1863–1881, Novembro 2004.
- [43] J. L. Sobrinho e A. S. Krishnakumar. Quality-of-Service in Ad Hoc Carrier Sense Multiple Access Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1353–1368, Agosto 1999.
- [44] J. L. Sobrinho e A. S. Krishnakumar. EQUB - Ethernet Quality-of-Service using Black Bursts. Em *Proceedings of the 23rd Annual Conference on Local Computer Networks (LCN)*, pp. 286–296, Outubro 1998.
- [45] G. Hwang e D. Cho. New access scheme for VoIP packets in IEEE 802.11e wireless LANs. *IEEE Communications Letters*, 9(7):667–669, 2005.
- [46] X. Yang e N. Vaidya. Priority scheduling in wireless ad hoc networks. *Wireless Networks*, 12(3):273–286, 2006.
- [47] J. P. Sheu, C. H. Liu, S. L. Wu, e Y. C. Tseng. A priority MAC protocol to support real-time traffic in ad hoc networks. *Wireless Networks*, 10(1):61–69, 2004.
- [48] G. R. Friedrich, O. R. Alimenti, e G. H. Reggiani. WRTMAC: A MAC Proposal for 802.11 Networks in Factory Automation. Em *Proceedings of the 15th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2010.
- [49] R. Moraes, F. Vasques, P. Portugal, e J. A. Fonseca. VTP-CSMA: A Virtual Token Passing Approach for Real-Time Communication in IEEE 802.11 Wireless Networks. *IEEE Transactions on Industrial Informatics*, 3(3):215–224, Agosto 2007.
- [50] R. Moraes, F. Vasques, P. Portugal, e P. F. Souto. A forcing collision resolution approach able to prioritize traffic in CSMA-based networks. *Computer Communications*, 33(1):54–64, 2010.
- [51] J. Villalón, P. Cuenca, L. O. Barbosa, e A. Garrido. B-EDCA: A QoS mechanism for multimedia communications over heterogeneous 802.11/802.11e WLANs. *Computer Communications*, 31(17):3905–3921, 2008.
- [52] E. Lopez-Aguilera, J. Casademont, J. Cotrina, e A. Rojas. Enhancement proposal for WLAN IEEE 802.11e: desynchronization of its working procedure. Em *Proceedings of the 14th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN)*, pp. 1–6, Setembro 2005.

- [53] B.J. Chang e J.W. Chu. H-EDCA: QoS-based High Performance Media Access Control for IEEE 802.11 e WLANs. Em *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1–5, 2006.
- [54] C. Wang, B. Li, e L. Li. A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF. *IEEE Transactions on Vehicular Technology*, 53(4):1235–1246, Julho 2004.
- [55] S. Vittorio, E. Toscano, e L. Lo Bello. CWFC: A contention window fuzzy controller for QoS support on IEEE 802.11e EDCA. Em *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1193–1196, 2008.
- [56] Y. Wu, J. Chiu, e T. Sheu. A modified EDCA with dynamic contention control for real-time traffic in multi-hop ad hoc networks. *Journal of Information Science and Engineering*, 24(4):1065–1079, 2008.
- [57] Y. Kwon, Y. Fang, e H. Latchman. Design of MAC protocols with fast collision resolution for wireless local area networks. *IEEE Transactions on Wireless Communications*, 3(3):793–807, 2004.
- [58] S.J. Golestani. A self-clocked fair queueing scheme for broadband applications. Em *Proceedings of 13th IEEE Conference on Computer Communications Networking for Global Communications*, volume 2, pp. 636–646, 1994. doi:10.1109/INFCOM.1994.337677.
- [59] N. Vaidya, A. Dugar, S. Gupta, e P. Bahl. Distributed fair scheduling in a wireless LAN. *IEEE Trans. Mob. Comput*, 4(6):616–629, 2005.
- [60] A. Hamidian e U. Körner. An enhancement to the IEEE 802.11e EDCA providing QoS guarantees. *Telecommunication Systems*, 31(2–3):195–212, 2006.
- [61] M. Masri, G. Juanole, e S. Abdellatif. Hybrid Admission Control Algorithm for IEEE 802.11e EDCA: Analysis. Em *7th International Conference on Networking (ICN)*, pp. 93–98, Abril 2008.
- [62] C. Cicconetti, L. Lenzini, E. Mingozzi, e G. Stea. Design and performance analysis of the Real-Time HCCA scheduler for IEEE 802.11e WLANs. *Computer Networks*, 51(9):2311–2325, 2007.
- [63] G. Cecchetti, A. L. Ruscelli, e F. Checconi. W-CBS: a scheduling algorithm for supporting QoS in IEEE 802.11e. Em *The Fourth International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, pp. 1–7, 2007.
- [64] L. Abeni e G. Buttazzo. Integrating Multimedia Applications in Hard Real-Time Systems. Em *Proceedings of the IEEE Real-Time Systems Symposium*, 1998.
- [65] E. Toscano e L. Lo Bello. A middleware for reliable soft real-time communication over IEEE 802.11 WLANs. Em *6th IEEE International Symposium on Industrial Embedded Systems (SIES)*, pp. 115–122, Junho 2011.
- [66] A. Bazzi, M. Diolaiti, e G. Pasolini. Measurement based call admission control strategies in infrastructured IEEE 802.11 WLANs. Em *16th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, volume 3, pp. 2093–2098, Setembro 2005.

- [67] C. Yong, B. Ali, S. Khatun, e K. Abdullah. An adaptive admission control mechanism for infrastructure based WLAN. Em *7th IEEE Malaysia International Conference on Communication & 13th IEEE International Conference on Networks*, volume 1, Novembro 2005.
- [68] Sunmyeng Kim, Young-Jong Cho, e Yong K. Kim. Admission control scheme based on priority access for wireless LANs. *Computer Networks*, 54(1):3–12, 2010.
- [69] S. Shin e H. Schulzrinne. Call Admission Control in IEEE 802.11 WLANs Using QP-CAT. Em *The 27th Conference on Computer Communications (INFOCOM)*, pp. 726–734, Abril 2008.
- [70] R. Achary, V. Vityanathan, P. Raj, e N. Srinivasan. Measurement based Admission Control for QoS Enhancement of IEEE 802.11 WLAN. *European Journal of Scientific Research*, 85(4):541–550, 2012.
- [71] T. Hiraguri, T. Kimura, T. Ogawa, H. Takase, A. Kishida, e K. Nishimori. Admission and Traffic Control Schemes Suitable for QoS Applications in WLAN Systems. *American Journal of Operations Research*, 2(3):382–390, 2012.
- [72] X. Sun, K. Piamrat, e C. Viho. QoE-based dynamic resource allocation for multimedia traffic in IEEE 802.11 wireless networks. Em *EEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, Julho 2011.
- [73] F. Didi, H. Labiod, G. Pujolle, e M. Feham. Physical rate and contention window based admission control (PRCW) for 802.11 WLANs. Em *IEEE Symposium on Computers and Communications (ISCC)*, pp. 1–7, Junho 2010.
- [74] I. Cruz, O. Lee, e S. Choi. Measurement-based Call Admission Control algorithm for multi-rate VoWLANs. Em *9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pp. 100–102, Junho 2012.
- [75] D.P. Hole e F.A. Tobagi. Capacity of an IEEE 802.11b wireless LAN supporting VoIP. Em *IEEE International Conference on Communications*, volume 1, pp. 196–201, Junho 2004.
- [76] J. Yu, M. Choi, D. Qiao, e S. Choi. Supporting VoIP Services in IEEE 802.11e WLANs. Em *Quality of Service in Heterogeneous Networks*, volume 22, pp. 493–509. Springer Berlin Heidelberg, 2009.
- [77] A. Andreadis e R. Zambon. Improving QoS Performance in IEEE 802.11e Under Heavy Traffic Loads. *International Journal of Wireless Information Networks*, 19:49–61, 2012.
- [78] A. Andreadis, G. Benelli, e R. Zambon. An admission control algorithm for QoS provisioning in IEEE 802.11e EDCA. Em *3rd International Symposium on Wireless Pervasive Computing (ISWPC)*, pp. 298–302, Maio 2008.
- [79] C. Cicconetti, L. Lenzini, E. Mingozzi, e G. Stea. An efficient cross layer scheduler for multimedia traffic in wireless local area networks with IEEE 802.11e HCCA. *ACM Mobile Computing Communications Reviews (SIGMOBILE)*, 11(3):31–46, Julho 2007.
- [80] A. L. Ruscelli, G. Cecchetti, A. Alifano, e G. Lipari. Enhancement of QoS support of HCCA schedulers using EDCA function in IEEE 802.11e networks. *Ad Hoc Networks*, 10(2):147–161, 2012.

- [81] P. Ansel, Q. Ni, e T. Tulletti. FHCF: A Simple and Efficient Scheduling Scheme for IEEE 802.11e Wireless LAN. *Mobile Networks and Applications*, 11:391–403, 2006.
- [82] E. Toscano e L. Lo Bello. Bandwidth-efficient admission control for EDF-based wireless industrial communication. Em *IEEE International Symposium on Industrial Electronics (ISIE)*, pp. 1186–1193, Junho 2011.
- [83] M. Spuri. Analysis of deadline scheduled real-time systems. Relatório técnico, INRIA – Institut National de Recherche en Informatique et en Automatique, 1996.
- [84] C. L. Liu e James W. Layland. Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. *Journal of ACM*, 20(1):46–61, Janeiro 1973.
- [85] V. Jacobson. Congestion avoidance and control. Em *ACM SIGCOMM Computer Communication Review*, volume 18, pp. 314–329. ACM, 1988.
- [86] RFC 6298: Computing TCP’s Retransmission Timer.
- [87] OPNET, Janeiro 2011. URL: <http://www.opnet.com>.
- [88] D. Brevi, D. Mazzocchi, R. Scopigno, A. Bonivento, R. Calcagno, e F. Rusina. A Methodology for the Analysis of 802.11a Links in Industrial Environments. Em *IEEE International Workshop on Factory Communication Systems (WFCS)*, pp. 165–174, 2006.
- [89] A. Willig, M. Kubisch, C. Hoene, e A. Wolisz. Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer. *IEEE Transactions on Industrial Electronics*, 49(6):1265–1282, 2002.
- [90] H. Trsek, L. Wisniewski, E. Toscano, e L. Lo Bello. A flexible approach for real-time wireless communications in adaptable industrial automation systems. Em *IEEE 16th Conference on Emerging Technologies Factory Automation (ETFA)*, pp. 1–4, 2011. doi:10.1109/ETFA.2011.6059152.
- [91] RFC 2215 - General Characterization Parameters for Integrated Service Network Elements, Setembro 1997.
- [92] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, 2000.
- [93] A. Kamerman e G. Aben. Throughput performance of wireless LANs operating at 2.4 and 5 GHz. Em *The 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, volume 1, pp. 190–195, 2000.
- [94] V. Paxson. Empirically derived analytic models of wide-area TCP connections. *IEEE/ACM Transactions on Networking*, 2(4):316–336, 1994.
- [95] B.A. Mah. An empirical model of HTTP network traffic. Em *Proceedings Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pp. 592–600, Abril 1997.
- [96] P. Tran-Gia, D. Staehle, e K. Leibnitz. Source Traffic Modeling of Wireless Applications. *AEU - International Journal of Electronics and Communications*, 55(1):27–36, 2001.
- [97] S. Luo e G.A. Marin. Realistic Internet traffic simulation through mixture modeling and a case study. Em *Proceedings of the Winter Simulation Conference*, Dezembro 2005.

- [98] A. Dainotti, A. Pescapé, e G. Ventre. A packet-level characterization of network traffic. Em *11th International Workshop on Computer-Aided Modeling, Analysis and Design of Communication Links and Networks (CAMAD)*, pp. 38–45, 2006.
- [99] Y. C. Chehadeh, A. Z. Hatahet, A. E. Agamy, M. A. Bamakhrama, e S. A. Banawan. Investigating Distribution of Data of HTTP Traffic: An Empirical Study. Em *Innovations in Information Technology*, pp. 1–5, Novembro 2006.
- [100] L. X. Cai, X. Shen, J. W. Mark, L. Cai, e Y. Xiao. Voice Capacity Analysis of WLAN with Unbalanced Traffic. *IEEE Transactions on Vehicular Technology*, 55(3):752–761, Maio 2006.
- [101] L. Shuai, G. Xie, e J. Yang. Characterization of HTTP behavior on access networks in Web 2.0. Em *International Conference on Telecommunications (ICT)*, pp. 1–6, Junho 2008.
- [102] X. Chen, S. Xu, J. Nie, e W. Wang. Research of HTTP Traffic Modeling for Future Networks. Em *4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, pp. 1–6, Outubro 2008.
- [103] Y. Chen, W. Wang, L. Fu, e X. Zhang. Traffic model for HTTP video page. Em *Third International Conference on Communications and Networking in China (ChinaCom)*, pp. 432–436, Agosto 2008.
- [104] R. Alturki, K. Nwizege, R. Mehmood, e M. Faisal. End to End Wireless Multimedia Service Modelling over a Metropolitan Area Network. Em *11th International Conference on Computer Modelling and Simulation (UKSIM)*, pp. 532–537, Março 2009.
- [105] R. Raghavendra e E.M. Belding. Characterizing high-bandwidth real-time video traffic in residential broadband networks. Em *Proceedings of the 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, pp. 597–602, 2010.
- [106] A. Rao, A. Legout, Y. Lim, D. Towsley, C. Barakat, e W. Dabbous. Network characteristics of video streaming traffic. Em *Proceedings of the Seventh Conference on emerging Networking EXperiments and Technologies, CoNEXT '11*, pp. 25:1–25:12. ACM, 2011.
- [107] ITU-T Recommendation H.264 - Advanced video coding for generic audiovisual services, Maio 2003.
- [108] ITU-T. Recommendation G.711: Pulse code modulation (PCM) of voice frequencies, Novembro 1988.
- [109] R. Jain, D. Chiu, e W. R. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer system. Relatório técnico, Eastern Research Laboratory, Digital Equipment Corporation, 1984.
- [110] A.V. Babu e L. Jacob. Performance analysis of IEEE 802.11 multirate WLANs: time based fairness vs throughput based fairness. Em *International Conference on Wireless Networks, Communications and Mobile Computing*, volume 1, pp. 203–208, 2005.
- [111] A.V. Babu e L. Jacob. Fairness Analysis of IEEE 802.11 Multirate Wireless LANs. *IEEE Transactions on Vehicular Technology*, 56(5):3073–3088, 2007.

- [112] Y. Weibo, M. Zhichao, N. Dawei, e D. Chao. PCAP: Proportional Channel Access Probability Fairness in Multi-rate IEEE 802.11 DCF. Em *Second International Conference on Intelligent System Design and Engineering Application (ISDEA)*, pp. 448–451, 2012.
- [113] H. Shi, R. Prasad, E. Onur, e I. Niemegeers. Fairness in Wireless Networks: Issues, Measures and Challenges. *IEEE Communications Surveys Tutorials*, (99):1–20, 2013.
- [114] Feng Xia, Youxian Sun, e Yu-Chu Tian. Feedback Scheduling of Priority-Driven Control Networks. *Computer Standards & Interfaces*, 31(3):539–547, Março 2009.
- [115] R. Costa, P. Portugal, F. Vasques, e R. Moraes. A TDMA-based Mechanism for Real-Time Communication in IEEE 802.11e Networks. Em *Proceedings of the 15th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2010.
- [116] R. Costa, P. Portugal, F. Vasques, R. Moraes, e R. Custódio. RT-WiFi: um mecanismo TDMA para suportar comunicações de tempo real em redes IEEE 802.11. Em *Proceedings of the 29th Simpósio Brasileiro de Telecomunicações. (SBrT)*, 2011.
- [117] R. Costa, P. Portugal, F. Vasques, R. Moraes, e R. Custódio. A Coordination Layer to Handle Real-Time Communication in Wi-Fi Networks with Uncontrolled Traffic Sources. Em *Proceedings of the 36th IEEE Conference on Local Computer Networks (LCN)*, 2011.
- [118] R. Costa, P. Portugal, R. Moraes, e F. Vasques. An admission control mechanism to handle real-time traffic in IEEE 802.11 networks in open communication environments. Em *Proceedings of the 9th IEEE International Workshop on Factory Communication Systems (WFCS)*, pp. 63–66, 2012.
- [119] R. Costa, P. Portugal, F. Vasques, e R. Moraes. Comparing RT-WiFi and HCCA approaches to handle real-time traffic in open communication environments. Em *Proceedings of the 17th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–8, 2012.
- [120] R. Costa, P. Portugal, F. Vasques, e R. Moraes. A comparative analysis of IEEE 802.11 MAC layer mechanisms to handle real-time traffic in open communication environments. Em *Proceedings of the 13th Conferência de Redes de Computadores (CRC)*, pp. 85–90, 2013.

Anexo A

Lista de Publicações

Lista de artigos publicados no âmbito desta tese:

- R. Costa, P. Portugal, F. Vasques, e R. Moraes. A TDMA-based Mechanism for Real-Time Communication in IEEE 802.11e Networks. Em *Proceedings of the 15th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2010. [115]
- R. Costa, P. Portugal, F. Vasques, R. Moraes, e R. Custódio. RT-WiFi: um mecanismo TDMA para suportar comunicações de tempo real em redes IEEE 802.11. Em *Proceedings of the 29th Simpósio Brasileiro de Telecomunicações. (SBrT)*, 2011. [116]
- R. Costa, P. Portugal, F. Vasques, R. Moraes, e R. Custódio. A Coordination Layer to Handle Real-Time Communication in Wi-Fi Networks with Uncontrolled Traffic Sources. Em *Proceedings of the 36th IEEE Conference on Local Computer Networks (LCN)*, 2011. [117]
- R. Costa, P. Portugal, R. Moraes, e F. Vasques. An admission control mechanism to handle real-time traffic in IEEE 802.11 networks in open communication environments. Em *Proceedings of the 9th IEEE International Workshop on Factory Communication Systems (WFCS)*, pp. 63–66, 2012. [118]
- R. Costa, P. Portugal, F. Vasques, e R. Moraes. Comparing RT-WiFi and HCCA approaches to handle real-time traffic in open communication environments. Em *Proceedings of the 17th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–8, 2012. [119]
- R. Costa, P. Portugal, F. Vasques, e R. Moraes. A comparative analysis of IEEE 802.11 MAC layer mechanisms to handle real-time traffic in open communication environments. Em *Proceedings of the 13th Conferência sobre Redes de Computadores (CRC)*, pp. 85–90, 2013. [120]

Anexo B

Modelo HCCA implementado no simulador OPNET

Este apêndice descrever o modelo HCCA (*HCF Controlled Channel Access*) implementado através da ferramenta de simulação OPNET *Modeler Wireless Suite*. Este modelo (disponível em http://paginas.fe.up.pt/~pportuga/hcca_model_opnet) foi utilizado para a obtenção de resultados comparativos com os obtidos pela arquitetura RT-WiFi proposta nesta tese.

B.1 Introdução

O OPNET (*Optimised Network Engineering Tools*) é um simulador de eventos discretos (*Discret Event Simulation – DES*) que utiliza uma modelagem baseada em objetos e um fluxo de execução baseado numa máquina de estados. Desta forma, todos os dispositivos de rede, *links* de ligação e protocolos são modelados graficamente como um objeto, que possui uma série de características e propriedades. Assim, cada objeto por ser representado por uma lista de outros sub-objetos (por exemplo: um computador pode ser representado por uma lista de objetos que compõem a pilha de rede TCP/IP), ou então por processo modelado através de uma máquina de estados (por exemplo: a implementação funcional do objeto que representa a camada de rede).

No caso específico de redes WLAN (*Wireless Local Area Network*), o OPNET disponibiliza uma lista de protocolos previamente implementados no pacote original¹:

Camada Física

Standard 802.11a, IEEE 802.11b, IEEE 802.11g e IEEE 802.11n.

Camada de Enlace

Mecanismos DCF (*Distributed Coordination Function*), PCF (*Point Coordination Function*) e EDCA (*Enhanced Distributed Channel Access*).

Além dos pacotes originais, é possível encontrar na comunidade de utentes do OPNET, diferentes implementações de protocolos não suportados pelo pacote original.

¹Ao levar em consideração a versão 16.1 do OPNET Modeler Wireless Suite.

Como previamente mencionado, cada dispositivo utilizado nas simulações é referenciado com um objeto. A Figura B.1 apresenta uma lista de objetos disponíveis para redes WLAN.



Figura B.1: Objetos disponíveis para modelagem de WLAN.

Cada objeto possui uma lista de parâmetros para a configuração das suas características específicas. A Figura B.2 apresenta a lista de parâmetros passíveis de configuração de um objeto que representa uma estação (referenciado no OPNET como o modelo `wlan_wkstn`).

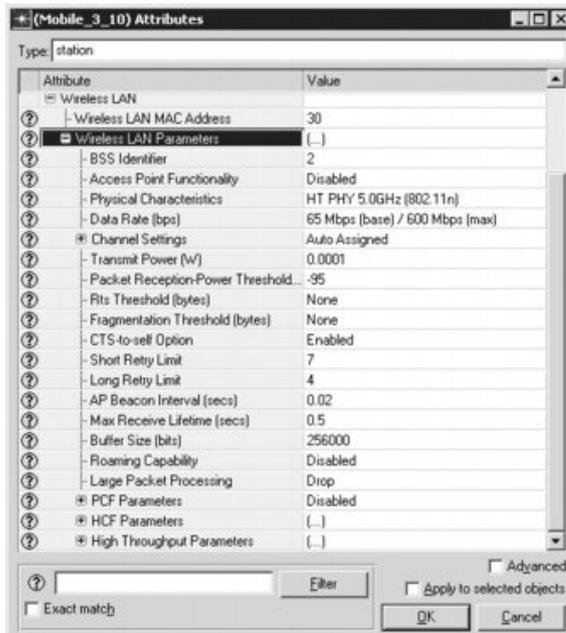


Figura B.2: Parâmetros de comunicação sem fio configuráveis numa estação.

O objeto `wlan_wkstn` implementa a pilha TCP/IP para estações que utilizam o standard IEEE 802.11 como tecnologia de comunicação. A Figura B.3 apresenta os sub-objetos utilizados para a sua implementação.

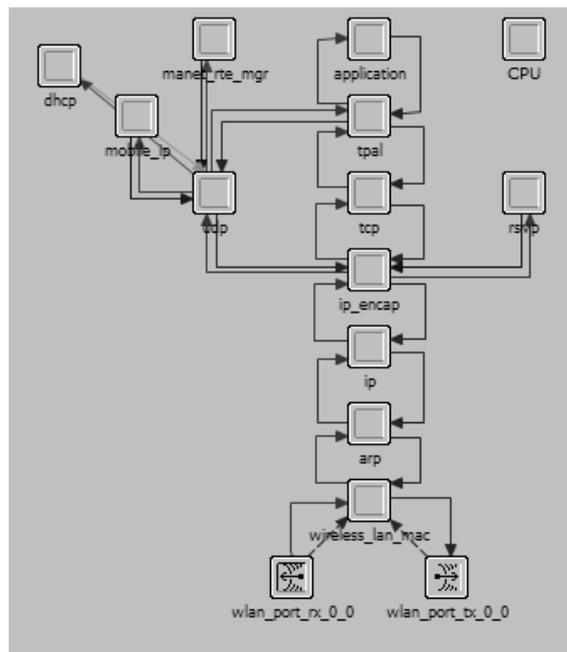


Figura B.3: Pilha TCP/IP implementada no objeto wlan_wkstn.

No caso específico do objeto que modela a camada de enlace (*wireless_lan_mac*), este é implementado pelo processo *wlan_dispatch* (Figura B.4), o qual contém um único estado denominado *spawn*. No OPNET, cada estado possui um fluxo de execução de entrada e outro de saída nos quais são executadas funções e realizadas leituras e definições de parâmetros.

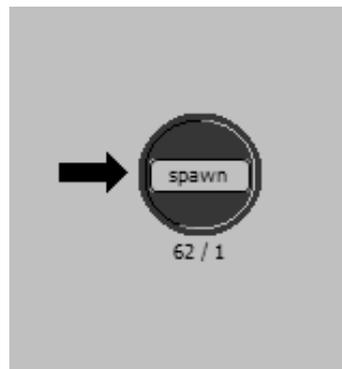


Figura B.4: Processo wlan_dispatch.

Após realizar as leituras dos parâmetros, as funções definidas no código de entrada deste estado detectam se a estação está a utilizar uma implementação comum da sub-camada MAC (*Media Access Control*), ou seja, contendo os mecanismos DCF ou PCF; ou então está a utilizar uma implementação baseada em QoS (*Quality-of-Service*), ou seja, o mecanismo EDCA. Caso seja detectada a utilização da sub-camada MAC comum, então o sistema instancia o processo *wlan_mac*, caso contrário será instanciado o processo *wlan_mac_hcf*.

Com o objetivo de seguir a mesma distribuição lógica definida pelo modelo original, o modelo HCCA implementado nesta tese foi construído sobre o processo `wlan_mac_hcf` (Figura B.5).

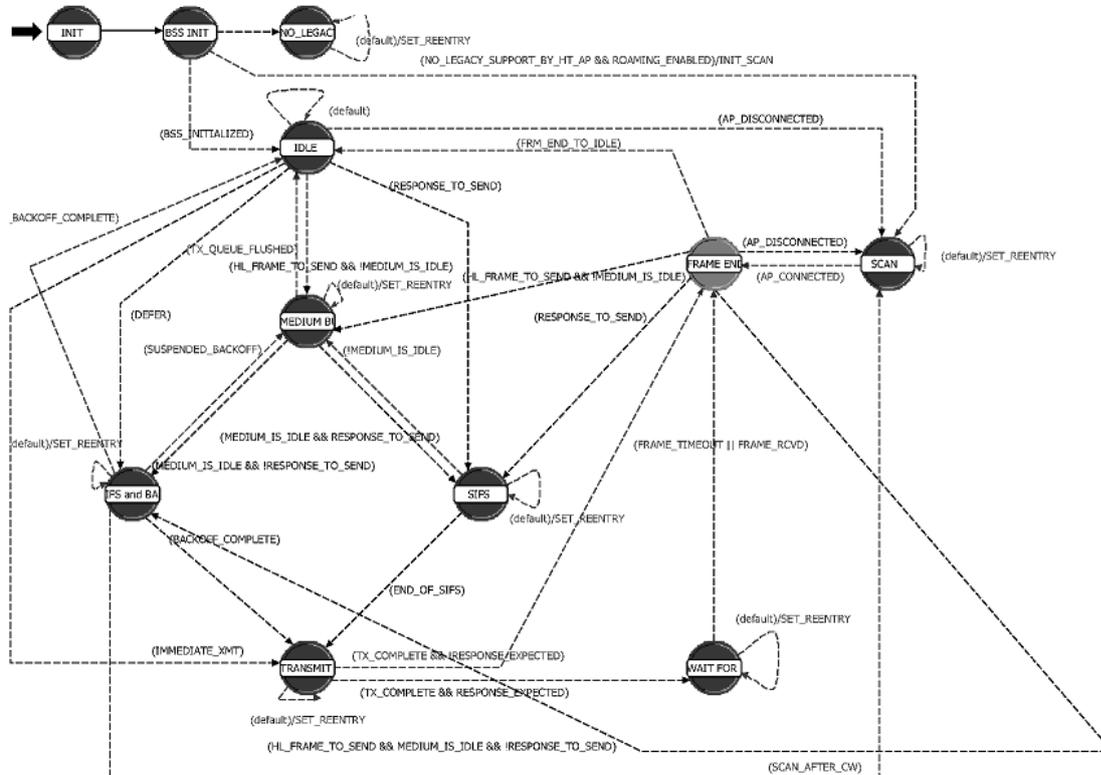


Figura B.5: Processo `wlan_mac_hcf`.

B.2 Implementação do HCCA

Como previamente apresentado, o modelo HCCA foi implementado no formato de um mecanismo adicional disponível no processo `wlan_mac_hcf`. Para isto, foram realizadas alterações nos seguintes arquivos:

- `wlan_dispatch.pr.m`: responsável pela definição e leitura dos parâmetros iniciais do HCCA, tanto das estações quanto do HC (*Hybrid Coordinator*);
- `wlan_mac_hcf.pr.m`: responsável pela implementação das funções operacionais do HCCA.

B.2.1 Limitações do modelo implementado

A principal limitação do modelo implementado encontra-se na ausência de um protocolo responsável pela troca de mensagens TSPEC (*Traffic Specification*) entre as estações e o HC. Para

solucionar este problema, ao instanciar o objeto referente a implementação de cada estação, os parâmetros TSPEC de cada uma (detalhados na subsecção B.2.2) são compartilhados como variáveis globais e inseridos em uma lista acessível pelo objeto referenciado como HC.

Este procedimento permite abstrair a troca de mensagens, de forma que, após instanciar todos os objetos, o HC têm disponível todas as informações TSPEC de todas as estações que desejam efetuar transmissões utilizando o mecanismo HCCA. Com base nesta lista, o HC pode realizar os procedimentos pertinentes ao controle de admissão.

B.2.2 O arquivo `wlan_dispatch.pr.m`

As modificações realizadas neste arquivo dizem respeito aos atributos do modelo, nomeadamente relacionados a interface. Para facilitar a configuração dos parâmetros TSPEC das estações e de parâmetros do HC, foi criada uma entrada denominada "HCCA Parameters" no menu de configuração (Figura B.2).

Neste contexto, foram inseridos os seguintes parâmetros:

- **HCCA Functionality:** variável booleana que define se a estação opera (ou não) sobre a função HCCA;
- **Type:** variável inteira que define se o dispositivo é uma estação (1), um AP (2) ou um servidor (3);
- **CFP Beacon Multiple:** variável inteira que define o número de TBTT (*Target Beacon Transmission Times*) entre dois CFP (*Contention Free Period*). Por exemplo, se o valor deste atributo for 2, e o TBTT for 20 msec (definido pelo parâmetro "AP Beacon Interval") então a BSS irá entrar em CFP a cada 40 msec. O primeiro CFP irá iniciar em "Beacon Interval" \times ("CFP Offset" + 1).
- **CFP Offset:** variável inteira que define o número de TBTT como offset do CFP a partir do seu ponto inicial.
- **CFP Interval:** variável de ponto flutuante que define a duração máxima do CFP (em segundos). Um AP pode não utilizar todo o tempo definido neste parâmetro, caso este finalize o polling de todas as estações contida na lista de polling e esvazie o seu *buffer* para transmissões CFP antes deste tempo; então o tempo restante pode ser utilizado pelo CP (*Contention Period*).
- **Max Failed Polls:** variável inteira que define o número máximo de polls consecutivos que podem ser enviados pelo AP sem uma mensagem acknowledgment válida. Se o número de falhas para um determinado MAC exceder este parâmetro o AP irá enviar o poll para a próxima estação da lista de polling.
- **Mean Data Rate:** variável de ponto flutuante que define o valor médio de dados (ρ) da especificação TSPEC (em bits/segundo). Este valor é utilizado pelo controle de admissão do HCCA para definir o número de MSDU enviados pela estação (N_i).

- **Nominal MSDU Size:** variável inteira que define o tamanho nominal da MSDU (L) da especificação TSPEC (em bytes). Este valor é utilizado pelo controlo de admissão do HCCA para definir o parâmetro N_i e o tamanho da TXOP alocada para cada estação.
- **Physical Transmission Rate:** variável de ponto flutuante que define a taxa de transmissão física (R) da especificação TSPEC (em bits/segundo). Este valor é utilizado pelo controlo de admissão do HCCA para se definir o tamanho da TXOP alocada para cada estação.
- **Maximum Allowable MSDU Size:** variável inteira que define o tamanho máximo permitido para uma MSDU (M) da especificação TSPEC (em bytes). Este valor é utilizado pelo controlo de admissão do HCCA para se definir o tamanho da TXOP alocada para cada estação.
- **Overheads:** variável de ponto flutuante que define os valores de *overhead* (O) da especificação TSPEC (em segundos). Neste parâmetro são incluídos os tempos para a transmissão da mensagem CF-Poll, para a transmissão da mensagem de ACK e também do SIFS.
- **Maximum Service Interval:** variável de ponto flutuante que define o intervalo máximo de serviço (SI_{max}) que uma TS (*Traffic Stream*) pode ser transmitida (em segundos).
- **Delay Bound:** variável de ponto flutuante que define o atraso máximo (D) que uma TS pode sofrer (em segundos) para entregar com sucesso as suas mensagens ao destino.

B.2.3 O arquivo `wlan_mac_hcf.pr.m`

No contexto deste arquivo, foram definidas novas variáveis de estado ("*State Variables*") e variáveis temporárias ("*Temporary Variables*"), criadas novas estruturas de dados e flags ("*Header Block*") bem como criadas novas funções e alteradas algumas já existentes ("*Function Block*"). Todas as inserções de código estão comentadas e indentificadas com a frase "*HCCA Code*".

Na máquina de estados (Figura B.5), alguns estados também tiveram parte do seu código alterado, nomeadamente:

BSS INIT

- **Fluxo de saída:** criada uma lista com todas as estações HCCA contendo os seus respectivos parâmetros TSPEC, realizado o teste de escalabilidade e caso este tenha um resultado positivo então é criada a lista de polling;

IDLE

- **Fluxo de saída:** alteradas as variáveis condicionais para permitir a transmissão de mensagens pelas estações HCCA dentro do CFP;

IFS and BACKOFF

- **Fluxo de entrada:** alteração do tempo IFS utilizado pelas transmissões de mensagens utilizando o mecanismo HCCA;
- **Fluxo de saída:** controlo das interrupções de CF-End e CF-Poll;

TRANSMIT

- **Fluxo de entrada:** alteração de uma variável de estado para controlo de TXOP em estações HCCA;

WAIT FOR RESPONSE

- **Fluxo de entrada:** agendamento da interrupção de *timeout* para transmissões utilizando o mecanismo HCCA. Assim como definido no standard IEEE 802.11, este tempo deve ser igual a PIFS;
- **Fluxo de saída:** processamento da interrupção de *timeout* caso a mesma não seja previamente cancelada pelo recebimento da mensagem antes do tempo PIFS.

Para implementar o modelo HCCA aproveitou-se grande parte do código já existente no simulador, de tal forma que algumas funções foram alteradas para tornar o código compatível com o novo modelo. No entanto, como esperado, foi necessário implementar novas funções para contemplar o objetivo. Segue abaixo uma lista contendo as funções que foram alteradas/criadas e suas respectivas funcionalidades:

wlan_hcf_sv_init

Esta função é responsável por inicializar as variáveis do sistema e também ler parâmetros locais (dos dispositivos) e globais (do ambiente). As modificações efetuadas nesta função caracterizam-se basicamente pela inicialização de algumas variáveis e a leitura de parâmetros locais (nomeadamente os parâmetros do menu "*HCCA Parameters*") e do parâmetro global "*Relative Deadline*" (definido como atributo global do processo `wlan_mac_hcf`).

wlan_hcf_frame_transmit

Esta função é responsável por preparar e seleccionar os tipos apropriados de mensagens que precisam ser enviadas. Neste caso, as alterações somente efetuam o controlo da fila de transmissão que está sendo utilizada para a transmissão das mensagens HCCA.

wlan_hcf_control_frame_send

Esta função é responsável por criar e enviar as mensagens de controlo. As alterações nesta função envolvem as mensagens ACK (alteração do tipo de mensagem que uma estação espera após o seu envio) e CF-End (alteração de variáveis para manter a compatibilidade com o restante do código).

wlan_hcf_cf_poll_send

Esta função (implementada para o modelo HCCA) é responsável por criar e enviar a mensagem CF-Poll, utilizada pelo HC para dar permissão de transmissão para uma estação.

wlan_hcf_insert_cf_poll

Esta função (implementada para o modelo HCCA) é responsável por inserir a mensagem CF-Poll na fila de voz para a sua transmissão.

wlan_hcf_cf_null_send

Esta função (implementada para o modelo HCCA) é responsável por criar e enviar a mensagem CF-Null, utilizada pelas estações que receberam uma mensagem CF-Poll mas que não têm mensagens de dados na fila para efetuar uma transmissão.

wlan_hcf_beacon_send

Esta função é responsável por criar e enviar a mensagem de *beacon*. As alterações nesta função envolvem a definição dos parâmetros no campo "*CF-Parameters*", responsável por criar um CFP e estipular sua duração.

wlan_hcf_interrupts_process

Esta função é responsável por processar as interrupções previamente agendadas no sistema. As alterações nesta função ocorrem nas interrupções responsáveis por sinalizar o instante de envio de uma nova mensagem de *beacon* (`wlanC_Beacon_Tx_Time`) e por sinalizar o final de um CFP (`wlanC_Cfp_End`). Na primeira interrupção são tratadas algumas variáveis de estado, agendada a interrupção referente ao envio da mensagem CF-End e inseridas todas as mensagens CF-Poll para todas as estações contidas na lista de polling. Na última interrupção são tratadas algumas variáveis de estado e invocada a função responsável pela criação e envio da mensagem CF-End.

wlan_hcf_physical_layer_data_arrival

Esta função é responsável por processar as mensagens recebidas da camada física. As alterações nesta função ocorrem basicamente na identificação e processamento das mensagens CF-Poll e CF-Null (antes não existentes no modelo), o processamento dos novos campos da mensagem de *beacon* e a atualização de variáveis de estado no recebimento de uma mensagem CF-End.

wlan_hcf_accepted_frame_stats_update

Esta função é responsável por atualizar as estatísticas de atraso fim-a-fim e *throughput*. As alterações nesta função foram realizadas de forma a contemplar as estatísticas específicas do tráfego HCCA.

wlan_hcf_frame_discard

Esta função é responsável por descartar (e posteriormente atualizar as estatísticas) as mensagens que tenham atingido o seu limite de retransmissão. Assim como na função anterior,

as alterações efetuadas nesta função foram realizadas de forma a contemplar as estatísticas específicas do tráfego HCCA.

Para melhor analisar os resultados obtidos pelo mecanismo HCCA, foi necessário criar um conjunto específico de estatísticas locais (independentes para cada dispositivo) e globais (conjunto de todas as estatísticas globais). Segue abaixo a lista contendo estas estatísticas:

Estatísticas locais:

- CF-Poll Traffic Sent (packets/sec);
- CF-Poll Traffic Rcvd (packets/sec);
- CF-Null Traffic Sent (packets/sec);
- CF-Null Traffic Rcvd (packets/sec);
- Deadline Missed by Drop (bits/sec);
- Deadline Missed by Drop (packets/sec);

Estatísticas globais:

- Real-Time Load (bits/sec);
- Real-Time Load (packets/sec);
- Real-Time Traffic Received (packets/sec);
- Real-Time Throughput (bits/sec);
- Real-Time End-to-End Delay (sec);
- Deadline Missed by Drop (bits/sec);
- Deadline Missed by Drop (packets/sec);
- Deadline Missed by Time (bits/sec);
- Deadline Missed by Time (packets/sec).