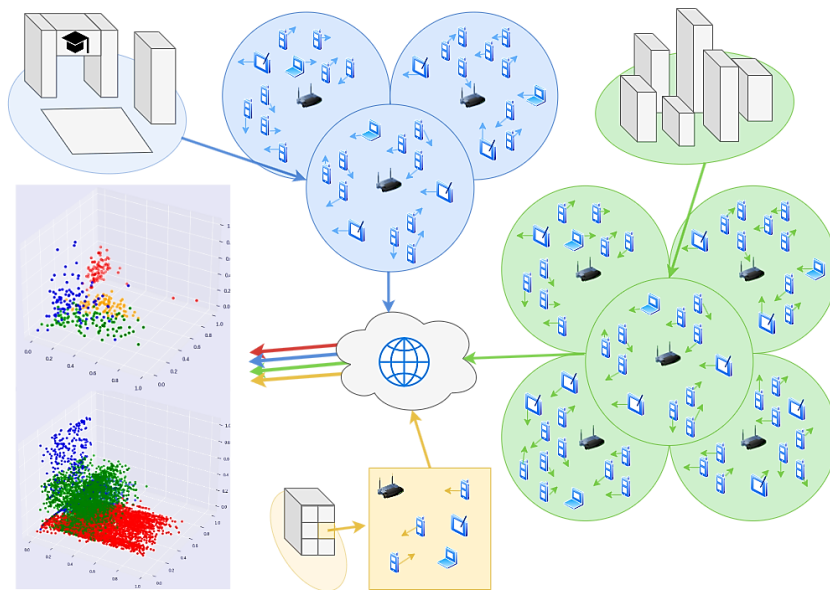


MÁSTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

TRABAJO FIN DE MASTER

ESTUDIO DE PROCEDIMIENTOS PARA LA MEJORA DE LA QOS EN REDES WLAN MEDIANTE CLUSTERING



Estudiante: Acha Alonso, Mikel

Directora: Ibarrola Armendáriz, Eva

Curso: 2022-2023

Fecha: Bilbao, 18, septiembre, 2023

Resumen

El presente proyecto comprende un estudio de técnicas de aprendizaje automático para la mejora de los procedimientos de gestión de la calidad de servicio en redes inalámbricas de área local. En concreto, se analiza la aplicabilidad de las técnicas de agrupamiento sobre las tareas de diferenciación de entornos y clasificación de perfiles.

Para dicho fin, inicialmente se ha desarrollado un extenso estudio teórico de las características de la tecnología IEEE 802.11 y sus casos de uso para generar los patrones de simulación a emplear y definir los indicadores de rendimiento clave. Asimismo, se ha realizado un análisis teórico de la aptitud de múltiples técnicas de agrupamiento para los procesos de mejora de la calidad de servicio.

Una vez cerrado el estudio conceptual, se hace uso de un simulador de redes inalámbricas para introducir múltiples casuísticas de utilización de la tecnología en base a lo definido en un inicio. Acto seguido, la salida del simulador es procesada mediante las técnicas de agrupamiento. La validación de los algoritmos se lleva a cabo mediante la aplicación de estos sobre los resultados de simulación y su posterior análisis. La síntesis final del estudio toma como soporte fundamental la comparación de las métricas de simulación correspondiente a cada caso bajo estudio con los grupos generados mediante aprendizaje automático.

Palabras Clave: Aprendizaje Automático, Agrupamiento, Red Inalámbrica de Área Local, Calidad de Servicio, Diferenciación de Entornos, Clasificación de Perfiles.

Laburpena

Proiektu honek ikasketa automatikoaren tekniken inguruko azterlana biltzen du, haririk gabeko sare lokalen zerbitzuaren kalitatearen kudeaketa prozesuen hobekuntza ahalbidetzeko. Zehazki, taldekatze tekniken aplikagarritasuna ikertzen da ingurune desberdintze eta erabiltzaile profil sailkapen eginkizunak betetzeko.

Hori xede, lehenik eta behin IEEE 802.11 teknologiaren eta haren erabilpen kasuen inguruko azterketa kontzeptual sakona garatu da simulazio ereduak diseinatu ahal izateko eta errendimendu adierazle gakoak definitzeko. Era berean, zerbitzuaren kalitatearen hobekuntzarako hainbat taldekatze tekniken gaitasunaren analisia burutu da.

Behin ikerlan kontzeptuala garatuta eta haren ondorioak oinarritzat hartuz, haririk gabeko sare simulagailu baten bidez teknologiaren kasuistika anizkunak ezarri dira. Jarraian, simulazioen emaitzak ikasketa automatikoaren algoritmoen sarrera gisa erabiltzen dira. Hori dela medio, aztertutako taldekatze teknikak balioztatu egiten dira haien aplikazioaren eta geroko analisiaren bidez. Azken azterketa prozesuak simulazioetan lortutako metrikek eragindako taldeen eta hasiera batean definitutako kasuistiken konparaketa du oinarri.

Gako Hitzak: Ikasketa Automatiko, Taldekatze, Haririk Gabeko Sare Lokal, Zerbitzuaren Kalitate, Ingurune Desberdintze, Erabiltzaile Profil Sailkapen, Anomalia Detekzio.

Abstract

The following project contains a study on machine learning techniques to enable the improvement of quality-of-service management procedures on wireless local area networks. Particularly, this project aims to analyse the applicability of clustering techniques on the processes of environment differentiation and user profile classification.

Therefore, detailed initial conceptual research has been done on the IEEE 802.11 technology characteristics and its use cases. Based on that study, several simulation patterns have been developed and key performance indicators have been defined. In the same way, a theoretical analysis of the aptitude of clustering techniques has been carried out focusing on their ability to improve quality of service management processes.

Once the conceptual research has been completed, multiple casuistries inferred from it have been implemented throughout the use of a wireless network simulator. Once the simulations have been finished, clustering techniques are used to process the output of the simulator. Finally, the analysis of the suitability of the algorithms over the results takes place. To validate the techniques for QoS improvement procedures, a comparison between the case study metrics and the final cluster results is performed.

Keywords: Clustering, Wireless Local Area Network, Quality of Service, Environment Differentiation, User Profile Classification, Anomaly Detection.

Índice

1.	Introducción	1
2.	Contexto	5
2.1.	Tecnología WLAN: 802.11	5
2.1.1.	Introducción al Estándar IEEE 802.11	5
2.1.2.	Evolución del Estándar IEEE 802.11	6
2.1.3.	Estándar IEEE 802.11 y QoS: IEEE 802.11e	9
2.2.	Aprendizaje Automático en Redes Inalámbricas	10
3.	Objetivos y Alcance	13
3.1.	Objetivo Principal	13
3.2.	Objetivos Secundarios	13
3.3.	Alcance	13
4.	Beneficios	15
4.1.	Beneficios Técnicos	15
4.2.	Beneficios Económicos	15
4.3.	Beneficios Sociales	16
5.	Estado del Arte	17
5.1.	Calidad de Servicio (QoS)	17
5.1.1.	Criterios de QoS.....	19
5.1.2.	Parámetros y requisitos mínimos de las clases de QoS	21
5.1.2.1.	Clasificación de Indicadores Clave de Rendimiento.....	25
5.1.3.	Introducción a los Procedimientos de Mejora de la QoS	27
5.2.	Resumen del Estudio de Escenarios de Aplicación del Estándar IEEE 802.11 29	
5.2.1.	Introducción a la Caracterización de Entornos de Redes Inalámbricas	29
5.2.1.1.	Definición Espacial y Topología.....	31
5.2.1.2.	Características Físicas, a Nivel de Enlace y Configuración de la Transmisión.....	33
5.2.1.3.	Caracterización del Tráfico de Red en Entornos WLAN.....	36
5.2.1.3.1.	Tipos de Tráfico por Entorno	36
5.2.1.3.2.	Características Principales de los Tipos de Tráfico	39
5.3.	Técnicas de Aprendizaje Automático	43
5.3.1.	Clasificación de los Algoritmos de Aprendizaje Automático	44
5.3.1.1.	Algoritmos de Clustering.....	46
6.	Análisis de Alternativas	52
6.1.	Alternativas Bajo Estudio	52

6.1.1.	Parametrización de los entornos WLAN en el simulador	52
6.1.1.1.	Mapeado Preliminar de los Resultados del Estudio de Escenarios de Aplicación de la Tecnología IEEE 802.11 a Parámetros de Simulación.....	52
6.1.1.2.	Modelado de Escenarios Wi-Fi con NS-3.....	56
6.1.1.3.	Definición de las Alternativas de Implementación de Parámetros de Simulación WLAN	59
6.1.2.	Indicadores de Rendimiento Clave	64
6.1.2.1.	Medida de Métricas de Calidad en Escenarios Wi-Fi con NS-3	64
6.1.2.2.	Alternativas de Indicadores de Rendimiento Clave Específicas.....	65
6.1.3.	Algoritmos y Técnicas de Clustering.....	67
6.1.3.1.	Clustering particional.....	67
6.1.3.2.	Clustering basado en densidad	68
6.1.3.3.	Clustering basado en la distribución	70
6.1.3.4.	Clustering jerárquico	71
6.2.	Criterios de Selección	73
6.2.1.	Criterios de Selección para la Parametrización de los entornos WLAN	73
6.2.2.	Criterios de Selección para los Indicadores de Rendimiento Clave.....	75
6.2.3.	Criterios de Selección para los Algoritmos y Técnicas de Clustering.....	76
6.3.	Selección de la Solución	77
6.3.1.	Selección de la Solución para la Parametrización de los entornos WLAN.....	77
6.3.2.	Selección de la Solución para los Indicadores de Rendimiento Clave.....	82
6.3.3.	Selección de la Solución para los Algoritmos y Técnicas de Clustering.....	90
7.	Análisis de Riesgos	92
7.1.	Identificación de los Riesgos Principales	92
7.2.	Evaluación y Priorización de los Riesgos	92
7.3.	Plan de Contingencia	94
8.	Metodología	96
8.1.	Implementación de los Escenarios	96
8.2.	Procesado de Datos	106
8.2.1.	Preprocesado de los Resultados de Simulación.....	107
8.2.2.	Implementación de los Algoritmos de Clustering.....	110
8.2.2.1.	Implementación de K-Means	111
8.2.2.2.	Implementación de DBSCAN.....	112
8.2.2.3.	Implementación de GMM.....	114
8.2.2.4.	Implementación de BIRCH.....	115
9.	Análisis de Resultados	117
9.1.	Resultados de Agrupación para la Diferenciación de Entornos	117

9.1.1.	Resultados de Agrupación de K-Means	118
9.1.2.	Resultados de Agrupación de DBSCAN	120
9.1.3.	Resultados de Agrupación de GMM.....	122
9.1.4.	Resultados de Agrupación de BIRCH	125
9.2.	Resultados de Agrupación para la Clasificación de Perfiles	127
9.2.1.	Resultados de Agrupación de K-Means	128
9.2.2.	Resultados de Agrupación de DBSCAN	130
9.2.3.	Resultados de Agrupación de GMM.....	131
9.2.4.	Resultados de Agrupación de BIRCH	133
9.3.	Análisis de Resultados sobre la Diferenciación de Entornos y la Clasificación de Perfiles	134
10.	Descripción del Equipo de Trabajo y Planificación.....	139
10.1.	Equipo de Trabajo.....	139
10.2.	Planificación de Trabajo.....	140
11.	Presupuesto	144
11.1.	Horas Internas.....	144
11.2.	Amortizaciones.....	144
11.3.	Gastos.....	144
11.4.	Costes Totales.....	145
12.	Conclusiones.....	146
13.	Bibliografía.....	148
14.	Anexo I: Estudio Detallado de los Escenarios Wi-Fi.....	156
14.1.	Aspectos Topológicos de los Cuatro Escenarios Bajo Estudio	156
14.2.	Características Físicas, a Nivel de Enlace y Configuración de la Transmisión de los Cuatro Escenarios Bajo Estudio	161
14.2.1.1.	Modelado de tráfico	167
14.2.1.1.1.	Caracterización de los Tipos de Tráfico por Entorno	167
14.2.1.1.2.	Modelado de los tipos de tráfico	172
15.	Anexo II: Selección de alternativas para la parametrización de los escenarios WLAN	181
16.	Anexo III: NS-3.....	192
16.1.	Introducción.....	192
16.1.1.	Modelo de abstracción de NS-3	192
16.1.2.	Limitaciones de las simulaciones Wi-Fi en NS-3	195
16.2.	Modelado de Escenarios Wi-Fi con NS-3	196
16.2.1.	Configuraciones Iniciales.....	196

16.2.2.	Función Principal: Definición Inicial y Ajuste de Parámetros	197
16.2.3.	Función Principal: Configuración de Red	201
16.2.4.	Función Principal: Movilidad.....	209
16.2.5.	Función Principal: Protocolos de Internet	210
16.2.6.	Función Principal: Aplicaciones y Generación de Tráfico.....	211
16.2.7.	Función Principal: Monitorización, Ejecución de las Simulaciones y Cálculos Finales 222	
16.2.8.	Función Principal: Volcado de Resultados a Fichero	226
16.2.9.	Función Principal: Final de Función	228
17.	Anexo IV: Procesamiento de Resultados a través de Clustering	229
17.1.	Preprocesado de los Resultados de Simulación	229
17.2.	Implementación de los Algoritmos de Clustering	230
17.2.1.	Implementación de K-Means	230
17.2.2.	Implementación de DBSCAN.....	231
17.2.3.	Implementación de GMM.....	232
17.2.4.	Implementación de BIRCH.....	233
17.3.	Representación de los Resultados	234
17.3.1.	Funciones Complementarias	235

Índice de Figuras

Figura 1: Tendencia de crecimiento global de dispositivos y conexiones 2018-2023	1
Figura 2: Punto de vista técnico y no técnico de la QoS, incluyendo la satisfacción del cliente	2
Figura 3: Arquitectura sencilla de una red WLAN.....	5
Figura 4: Relación entre la certificación de Wi-Fi Alliance y los protocolos IEEE 802.11	7
Figura 5: Arquitectura IEEE 802.11 e de la capa MAC.....	9
Figura 6: Alcance del Trabajo Fin de Máster	14
Figura 7: Esquema de contribuciones a la QoS de extremo a extremo	17
Figura 8: Relación entre QoS y NP	18
Figura 9: Matriz para la identificación de los criterios de QoS para las comunicaciones.....	19
Figura 10: Diagrama de funcionamiento de VoD.....	40
Figura 11: Diagrama de funcionamiento de tráfico de videoconferencia	41
Figura 12: Diagrama de funcionamiento de streaming multicast.....	41
Figura 13: Patrón de tráfico HTTP	42
Figura 14: Diagrama de tráfico FTP.....	42
Figura 15: Máquina de estados del tráfico VoIP	43
Figura 16: Clasificación del Aprendizaje Automático.....	45
Figura 17: Proceso de generación de conocimiento mediante clustering.....	47
Figura 18: Ejemplo de clustering basado en particiones.....	48
Figura 19: Ejemplo de clustering jerárquico	48
Figura 20: Ejemplo de clustering basado en teoría difusa	49
Figura 21: Ejemplo de clustering basado en la distribución.....	49
Figura 22: Ejemplo de clustering basado en la densidad.....	49
Figura 23: Ejemplo de clustering basado en la teoría de grafos	50
Figura 24: Ejemplo de clustering basado en cuadrículas.....	50
Figura 25: Ejemplo de clustering basado en la teoría fractal.....	50
Figura 26: Ejemplo de clustering basado en modelos de redes neuronales.....	51
Figura 27: Comparación del agrupamiento particional mediante K-Means y K-Medoids en presencia de muestras aisladas.....	68
Figura 28: Ejemplo de CFT.....	72
Figura 29: Diagrama representativo del entorno residencial.....	97
Figura 30: Diagrama representativo del entorno empresarial.....	99
Figura 31: Diagrama representativo del entorno interior de alta densidad.....	101
Figura 32: Diagrama representativo del entorno exterior de alta densidad.....	103
Figura 33: Muestra de fichero (versión reducida) con resultados de simulación a nivel de dispositivo.....	107
Figura 34: Muestra de fichero (versión reducida) con resultados de simulación a nivel de flujo de tráfico.....	107
Figura 35: Histograma de los resultados totales sin normalizar de las simulaciones en función del parámetro de medida.....	108
Figura 36: Histograma de los resultados totales preprocesados de las simulaciones en función del parámetro de medida.....	108
Figura 37: Histograma de los resultados de los AP (izq.) y STA (der.) con valores normalizados en función del parámetro de medida.....	109
Figura 38: Cruce de variables de los resultados de los APs (izq.) y STAs (der.) con valores normalizados.....	109
Figura 39: Representación tridimensional de métricas de APs (izq.) y STAs (der.) en base a distintos ejes después del preprocesado.....	110
Figura 40: Curvas de quiebre para las métricas de APs (izq.) y STAs (der.).....	111

Figura 41: Resultados de agrupación K-Means sobre las métricas preprocesadas de APs (izq.) y STAs (der.).....	112
Figura 42: Curvas de estimación de EPS para las métricas de APs (izq.) y STAs (der.).....	113
Figura 43: Resultados de agrupación DBSCAN sobre las métricas preprocesadas de APs (izq.) y STAs (der.).....	113
Figura 44: Curvas AIC y BIC para las métricas de APs (izq.) y STAs (der.).....	114
Figura 45: Resultados de agrupación GMM sobre las métricas preprocesadas de APs (izq.) y STAs (der.).....	115
Figura 46: Resultados de agrupación BIRCH sobre las métricas preprocesadas de APs (izq.) y STAs (der.).....	116
Figura 47: Gráficas tridimensionales de las muestras de APs sin etiquetar (izq.) y etiquetadas (der.).....	117
Figura 48: Gráfica tridimensional con las muestras de las métricas de los APs etiquetadas (izq.) y el resultado del agrupamiento de K-Means (der.).....	118
Figura 49: Gráficas bidimensionales del cruce de parámetros de los APs con el resultado de agrupamiento de K-Means.....	120
Figura 50: Gráfica tridimensional con las muestras de las métricas de los APs etiquetadas (izq.) y el resultado del agrupamiento de DBSCAN (der.).....	121
Figura 51: Gráficas bidimensionales del cruce de parámetros de los APs con el resultado de agrupamiento de DBSCAN.....	122
Figura 52: Gráfica tridimensional con las muestras de las métricas de los APs etiquetadas (izq.) y el resultado del agrupamiento de GMM (der.).....	123
Figura 53: Gráficas bidimensionales del cruce de parámetros de los APs con el resultado de agrupamiento de GMM.....	124
Figura 54: Gráfica tridimensional con las muestras de las métricas de los APs etiquetadas (izq.) y el resultado del agrupamiento de BIRCH (der.).....	125
Figura 55: Gráficas bidimensionales del cruce de parámetros de los APs con el resultado de agrupamiento de BIRCH.....	127
Figura 56: Gráfica tridimensional con las muestras de las métricas de las STAs sin etiquetar (izq.) y el resultado del agrupamiento de K-Means (der.).....	129
Figura 57: Gráficas bidimensionales del cruce de parámetros de las STAs con el resultado de agrupamiento de K-Means.....	129
Figura 58: Gráfica tridimensional con las muestras de las métricas de las STAs sin etiquetar (izq.) y el resultado del agrupamiento de DBSCAN (der.).....	130
Figura 59: Gráficas bidimensionales del cruce de parámetros de las STAs con el resultado de agrupamiento de DBSCAN.....	131
Figura 60: Gráfica tridimensional con las muestras de las métricas de las STAs sin etiquetar (izq.) y el resultado del agrupamiento de GMM (der.).....	132
Figura 61: Gráficas bidimensionales del cruce de parámetros de las STAs con el resultado de agrupamiento de GMM.....	132
Figura 62: Gráfica tridimensional con las muestras de las métricas de las STAs sin etiquetar (izq.) y el resultado del agrupamiento de BIRCH (der.).....	133
Figura 63: Gráficas bidimensionales del cruce de parámetros de las STAs con el resultado de agrupamiento de BIRCH.....	134
Figura 64: Gráficas tridimensionales resultantes de K-Means, DBSCAN, GMM y BIRCH para la diferenciación de entornos.....	135
Figura 65: Gráficas tridimensionales resultantes de K-Means, DBSCAN, GMM y BIRCH para la clasificación de perfiles.....	135
Figura 66: Diagrama de Gantt del proyecto.....	143
Figura 67: Disposición de escenario residencial propuesta en [46].....	157
Figura 68: Disposición de escenario empresarial propuesta en [50].....	158

Figura 69: Disposición de escenario empresarial propuesta en [46].....	158
Figura 70: Disposición de escenario indoor de alta densidad propuesta en [46].....	160
Figura 71: Disposición de escenario outdoor de alta densidad propuesta en [46].....	161
Figura 72: Diagrama de la llegada de paquetes VDI. [72].....	176
Figura 73: Diagrama de la llegada de paquetes VoIP. [72].....	178
Figura 74: Diseño en capas del modelo Wi-Fi en NS-3. [116].....	194
Figura 75: Código en NS-3 – listado de includes	196
Figura 76: Código en NS-3 – espacio de nombres y componente de log.....	196
Figura 77: Código en NS-3 – definición de variables (I).....	197
Figura 78: Código en NS-3 – definición de variables (II).....	198
Figura 79: Código en NS-3 – definición de variables (III).....	199
Figura 80: Código en NS-3 – introducción de variables por la línea de comandos (I).....	199
Figura 81: Código en NS-3 – introducción de variables por la línea de comandos (II).....	200
Figura 82: Código en NS-3 – ajustes iniciales (I).....	200
Figura 83: Código en NS-3 – ajustes iniciales (II).....	201
Figura 84: Código en NS-3 – definición de nodos y dispositivos de red	201
Figura 85: Código en NS-3 – función de generación de nodos.....	202
Figura 86: Código en NS-3 – instanciación de objetos de soporte de Wi-Fi y primeras configuraciones.....	202
Figura 87: Código en NS-3 – llamadas a la función de configuración de red.....	203
Figura 88: Código en NS-3 – función de configuración de red (I).....	203
Figura 89: Código en NS-3 – función de configuración de canal.....	204
Figura 90: Código en NS-3 – selección aleatoria de canal (I).....	204
Figura 91: Código en NS-3 – selección aleatoria de canal (II).....	205
Figura 92: Código en NS-3 – selección determinista de canal (I).....	205
Figura 93: Código en NS-3 – selección determinista de canal (II).....	206
Figura 94: Código en NS-3 – función de configuración de red (II).....	206
Figura 95: Código en NS-3 – alternativas de cálculo de pérdidas de propagación.....	207
Figura 96: Código en NS-3 – función de configuración de red (III).....	208
Figura 97: Código en NS-3 – función de configuración de red (IV).....	209
Figura 98: Código en NS-3 – configuración de movilidad (I).....	209
Figura 99: Código en NS-3 – configuración de movilidad (II).....	210
Figura 100: Código en NS-3 – configuración de movilidad (III).....	210
Figura 101: Código en NS-3 – configuración de los protocolos de internet y direccionamiento	211
Figura 102: Código en NS-3 – función para el cálculo del número de STAs por tipo de tráfico.....	211
Figura 103: Código en NS-3 – función de inicialización del array de selección de STAs.....	212
Figura 104: Código en NS-3 – función de selección de STAs.....	212
Figura 105: Código en NS-3 – función de activación y desactivación de aplicaciones.....	212
Figura 106: Código en NS-3 – implementación del tráfico VoD.....	213
Figura 107: Código en NS-3 – función de generación de tráfico desde el servidor VoD.....	213
Figura 108: Código en NS-3 – función de generación de tráfico hacia el servidor VoD.....	214
Figura 109: Código en NS-3 – implementación del tráfico HTTP.....	214
Figura 110: Código en NS-3 – función de generación de tráfico entre servidor y cliente HTTP.....	215
Figura 111: Código en NS-3 – fichero de encabezado del generador de tráfico FTP M2.....	215
Figura 112: Código en NS-3 – fichero de implementación del generador de tráfico FTP M2 (I).....	216
Figura 113: Código en NS-3 – fichero de implementación del generador de tráfico FTP M2 (II).....	217
Figura 114: Código en NS-3 – fichero de implementación del generador de tráfico FTP M2 (III).....	217
Figura 115: Código en NS-3 – fichero de implementación del generador de tráfico FTP M2 (IV).....	218

Figura 116: Código en NS-3 – fichero de implementación del generador de tráfico FTP M2 (V)	218
Figura 117: Código en NS-3 – implementación del tráfico FTP	219
Figura 118: Código en NS-3 – implementación del tráfico Gaming	219
Figura 119: Código en NS-3 – función de generación de tráfico Gaming en sentido ascendente	220
Figura 120: Código en NS-3 – función de generación de tráfico Gaming en sentido descendente	220
Figura 121: Código en NS-3 – implementación del tráfico VoIP	221
Figura 122: Código en NS-3 – función de generación de tráfico VoIP en sentido ascendente	221
Figura 123: Código en NS-3 – función de generación de tráfico VoIP en sentido descendente	222
Figura 124: Código en NS-3 – extracción de métricas de señal de los AP	222
Figura 125: Código en NS-3 – función de cálculo de métricas de señal	223
Figura 126: Código en NS-3 – declaración e instalación de FlowMonitor	223
Figura 127: Código en NS-3 – ejecución de la simulación	223
Figura 128: Código en NS-3 – cálculo de estadísticas sobre los flujos de tráfico (I)	224
Figura 129: Código en NS-3 – cálculo de estadísticas sobre los flujos de tráfico (II)	224
Figura 130: Código en NS-3 – cálculo de estadísticas sobre los flujos de tráfico (III)	225
Figura 131: Código en NS-3 – cálculo de estadísticas sobre los flujos de tráfico (IV)	225
Figura 132: Código en NS-3 – cálculo de estadísticas sobre los flujos de tráfico (V)	225
Figura 133: Código en NS-3 – volcado de datos al fichero de estadísticas de dispositivos (I)	226
Figura 134: Código en NS-3 – volcado de datos al fichero de estadísticas de dispositivos (II)	226
Figura 135: Código en NS-3 – volcado de datos al fichero de estadísticas de dispositivos (III)	227
Figura 136: Código en NS-3 – volcado de datos al fichero de de flujos de tráfico (I)	227
Figura 137: Código en NS-3 – volcado de datos al fichero de de flujos de tráfico (II)	228
Figura 138: Código en NS-3 – volcado de datos al fichero de de flujos de tráfico (III)	228
Figura 139: Código en NS-3 – final de función main	228
Figura 140: Código en Python – librerías de uso genérico para el manejo, representación y preprocesado de los resultados de simulación	229
Figura 141: Código en Python – definición del entorno de trabajo	229
Figura 142: Código en Python – preprocesado de los datos (I)	230
Figura 143: Código en Python – preprocesado de los datos (II)	230
Figura 144: Código en Python – importación de K-Means de la librería SKLearn	230
Figura 145: Código en Python – estimación del parámetro K de K-Means	231
Figura 146: Código en Python – ejecución de K-Means	231
Figura 147: Código en Python – importación de clases para la implementación de DBSCAN	231
Figura 148: Código en Python – estimación de los parámetros de DBSCAN	232
Figura 149: Código en Python – ejecución de DBSCAN	232
Figura 150: Código en Python – importación de clases para la implementación de GMM	232
Figura 151: Código en Python – estimación del número de componentes de GMM	233
Figura 152: Código en Python – ejecución de GMM	233
Figura 153: Código en Python – importación de clases para la implementación de BIRCH	233
Figura 154: Código en Python – ejecución de BIRCH	233
Figura 155: Código en Python – representación en tablas de la diversidad de grupos para el caso del etiquetado de escenarios	234
Figura 156: Código en Python – conteo de muestras de cada escenario por clúster	234
Figura 157: Código en Python – representación bidimensional de muestras	235
Figura 158: Código en Python – importación de clases para la reordenación de etiquetas	235
Figura 159: Código en Python – funciones de reordenación de etiquetas	235
Figura 160: Código en Python – ejemplo de ejecución de la reordenación de etiquetas	236

Índice de Tablas

Tabla 1: Aplicaciones estudiadas sobre la aplicación de ML en redes inalámbricas.....	3
Tabla 2: Resumen de características principales de las últimas versiones de Wi-Fi.....	7
Tabla 3: Resumen de aplicaciones de aprendizaje automático sobre redes inalámbricas.....	11
Tabla 4: Definición de clases de QoS en redes IP y objetivos de funcionamiento de red.....	23
Tabla 5: Guía para las clases de QoS IP.....	24
Tabla 6: Clasificación de KPIs con ejemplos.....	25
Tabla 7: Servicios de datos y parámetros de calidad.....	26
Tabla 8: Calificación media de opinión.....	26
Tabla 9: Ejemplos de factores por tipo de contexto.....	27
Tabla 10: Parámetros a tener en cuenta en la definición de escenarios.....	30
Tabla 11: Resumen del modelo de tráfico para entorno residencial definido por el TGax.....	37
Tabla 12: Resumen del modelo de tráfico para entorno de oficinas definido por el TGax.....	37
Tabla 13: Resumen del modelo de tráfico para entorno interior de alta densidad definido por el TGax.....	38
Tabla 14: Resumen del modelo de tráfico para entorno exterior de alta densidad definido por el TGax.....	38
Tabla 15: Resumen de los tipos de tráfico.....	39
Tabla 16: Clasificación tradicional de los algoritmos de clustering.....	47
Tabla 17: Características topológicas generales extraídas del estudio teórico.....	53
Tabla 18: Características técnicas generales extraídas del estudio teórico.....	54
Tabla 19: Resumen de tipos de tráfico por entorno contemplados en el Estado del Arte.....	54
Tabla 20: Asignación de nodos con tráfico por escenario según el TGax.....	55
Tabla 21: Alternativas espaciales a seleccionar por escenario.....	59
Tabla 22: Alternativas topológicas y espaciales de los APs.....	60
Tabla 23: Alternativas topológicas y espaciales de las STAs.....	61
Tabla 24: Alternativas de configuración frecuencial.....	61
Tabla 25: Alternativas de configuración de la transmisión.....	62
Tabla 26: Alternativas de modelos de propagación.....	63
Tabla 27: Alternativas de configuración a nivel de enlace y físico.....	63
Tabla 28: Alternativas de tráfico de red.....	64
Tabla 29: Alternativas de Métricas en NS-3.....	66
Tabla 30: Alternativas de interpretación de las medidas.....	66
Tabla 31: Criterios sobre la selección de alternativas de parametrización de red.....	74
Tabla 32: Criterios sobre la selección de alternativas de KPIs.....	75
Tabla 33: Criterios sobre la selección de alternativas de algoritmos de clustering.....	77
Tabla 34: Selección de la solución para la caracterización espacial de los escenarios.....	78
Tabla 35: Selección de la solución para la caracterización topológica, posicionamiento y movilidad de los AP.....	78
Tabla 36: Selección de la solución para la caracterización topológica, posicionamiento y movilidad de las STAs.....	79
Tabla 37: Selección de la solución de implementación a nivel frecuencial.....	79
Tabla 38: Selección de la solución de configuración de la transmisión de los dispositivos.....	80
Tabla 39: Selección de la solución de modelos de pérdidas de propagación.....	81
Tabla 40: Selección de la solución de configuración a nivel de enlace y físico.....	81
Tabla 41: Selección de la solución de los tipos de tráfico de red a implementar.....	82
Tabla 42: Selección de la solución de los flujos a establecer en la caracterización de escenarios.....	82
Tabla 43: Mapeo de los criterios para la selección de KPIs y sus identificadores.....	83

Tabla 44: Evaluación de la solución del tiempo de funcionamiento como KPI para la diferenciación de entornos.....	84
Tabla 45: Evaluación de la solución del bandwidth y throughput como KPIs para la diferenciación de entornos.....	84
Tabla 46: Evaluación de la solución del número de paquetes / bytes recibidos y transmitidos como KPIs para la diferenciación de entornos.....	85
Tabla 47: Evaluación de la solución tasa de pérdida de paquetes y errores como KPIs para la diferenciación de entornos.....	85
Tabla 48: Evaluación de la solución del retardo y jittetr percibidos como KPIs para la diferenciación de entornos.....	86
Tabla 49: Evaluación de la solución del valor SNR como KPIs para la diferenciación de entornos.....	86
Tabla 50: Evaluación de la solución del tiempo de funcionamiento como KPI para clasificación de perfiles.....	87
Tabla 51: Evaluación de la solución del bandwidth y throughput como KPIs para la clasificación de perfiles.....	87
Tabla 52: Evaluación de la solución del número de paquetes / bytes recibidos y transmitidos como KPIs para la clasificación de perfiles.....	88
Tabla 53: Evaluación de la solución tasa de pérdida de paquetes y errores como KPIs para la clasificación de perfiles.....	88
Tabla 54: Evaluación de la solución del retardo y jittetr percibidos como KPIs para la clasificación de perfiles.....	89
Tabla 55: Evaluación de la solución del valor SNR como KPIs para la clasificación de perfiles.....	89
Tabla 56: Selección de las soluciones para la implementación de KPIs a través de valores medios.....	90
Tabla 57: Mapeo de los criterios para la selección de algoritmos de clustering y sus identificadores.....	90
Tabla 58: Selección de las soluciones para la implementación algoritmos de clustering.....	91
Tabla 59: Probabilidad de ocurrencia de los riesgos.....	93
Tabla 60: Impacto de los riesgos sobre el trabajo.....	93
Tabla 61: Probabilidad de ocurrencia e impacto por riesgo.....	93
Tabla 62: Matriz probabilidad – impacto.....	93
Tabla 63: Parámetros de implementación del entorno residencial en NS-3.....	98
Tabla 64: Referencias de implementación del tráfico en el entorno residencial en NS-3.....	99
Tabla 65: Parámetros de implementación del entorno empresarial en NS-3.....	100
Tabla 66: Referencias de implementación del tráfico en el entorno empresarial en NS-3.....	101
Tabla 67: Parámetros de implementación del entorno interior de alta densidad en NS-3.....	102
Tabla 68: Referencias de implementación del tráfico en el entorno interior de alta densidad en NS-3.....	103
Tabla 69: Parámetros de implementación del entorno exterior de alta densidad en NS-3.....	104
Tabla 70: Referencias de implementación del tráfico en el entorno exterior de alta densidad en NS-3.....	104
Tabla 71: Recursos principales utilizados para la caracterización de escenarios en NS-3.....	105
Tabla 72: Recursos principales utilizados para la simulación de tráfico en NS-3.....	106
Tabla 73: Mapeo de identificadores para el seguimiento del análisis de resultados.....	117
Tabla 74: Resumen del experimento de K-Means sobre las métricas de APs.....	118
Tabla 75: Resultados de K-Means sobre las métricas de APs.....	119
Tabla 76: Resumen del experimento de DBSCAN sobre las métricas de APs.....	120
Tabla 77: Resultados de DBSCAN sobre las métricas de APs en formato tabla.....	121
Tabla 78: Resumen del experimento de GMM sobre las métricas de APs.....	122
Tabla 79: Resultados de GMM sobre las métricas de APs en formato tabla.....	124

Tabla 80: Resumen del experimento de GMM sobre las métricas de APs.....	125
Tabla 81: Resultados de BIRCH sobre las métricas de APs en formato tabla.....	126
Tabla 82: Mapeo de identificadores para el seguimiento del análisis de resultados de la clasificación de perfiles.....	128
Tabla 83: Resumen del experimento de K-Means sobre las métricas de STAs.....	128
Tabla 84: Resumen del experimento de DBSCAN sobre las métricas de STAs.....	130
Tabla 85: Resumen del experimento de GMM sobre las métricas de STAs.....	131
Tabla 86: Resumen del experimento de BIRCH sobre las métricas de STAs.....	133
Tabla 87: Caracterización genérica de los clústeres obtenidos a través de los algoritmos para la tarea de clasificación de perfiles.....	136
Tabla 88: Valoración resumida de los algoritmos sobre la diferenciación de entornos.....	139
Tabla 89: Relación entre los identificadores de la planificación del proyecto y el concepto que representan.....	140
Tabla 90: Planificación estructurada del proyecto.....	142
Tabla 91: Recursos humanos.....	144
Tabla 92: Amortizaciones.....	144
Tabla 93: Gastos.....	145
Tabla 94: Costes totales.....	145
Tabla 95: Modelo de tráfico para entorno residencial definido por el TGax.....	168
Tabla 96: Modelo de tráfico para entorno de oficinas definido por el TGax.....	169
Tabla 97: Modelo de tráfico para entorno interior de alta densidad definido por el TGax.....	170
Tabla 98: Modelo de tráfico para entorno exterior de alta densidad definido por el TGax.....	171
Tabla 99: Parámetros para el modelado de tráfico Wireless Display.....	172
Tabla 100: Parámetros lambda y k para tasas de bits de vídeo.....	173
Tabla 101: Características de los vídeos utilizados para el modelado de tráfico VoD.....	174
Tabla 102: Comportamiento estadístico del tráfico de vídeo en [66].....	175
Tabla 103: Parámetros para tráfico de Streaming Multicast.....	175
Tabla 104: Parámetros del tráfico VDI.....	176
Tabla 105: Parámetros de tráfico HTTP.....	177
Tabla 106: Parámetros de tráfico FTP.....	178
Tabla 107: Parámetros del tráfico VoIP.....	179
Tabla 108: Parámetros del tráfico gaming.....	180
Tabla 109: Mapeo de los criterios de selección de la parametrización WLAN y cada identificador asociado.....	181
Tabla 110: Mapeo de los entornos en estudio y sus identificadores asociados.....	181
Tabla 111: Selección de alternativas espaciales por escenario.....	182
Tabla 112: Selección de alternativas topológicas, de posicionamiento y movilidad por escenario y dispositivo.....	183
Tabla 113: Selección de alternativas en términos de frecuencia.....	184
Tabla 114: Selección del ancho de canal.....	184
Tabla 115: Selección de las alternativas de transmisión por escenario y dispositivo.....	185
Tabla 116: Selección de los modelos de propagación a implementar en cada escenario.....	187
Tabla 117: Selección de las características tecnológicas principales a nivel de protocolo.....	189
Tabla 118: Selección de los tipos de tráfico a implementar en cada escenario.....	189
Tabla 119: Selección de la densidad de tráfico por tipos y dispositivos a implementar en cada escenario.....	191

Índice de Ecuaciones

Ecuación 1: Cálculo del retardo de transferencia de paquetes IP.....	21
Ecuación 2: Cálculo de la variación de retardo de paquetes IP.....	22
Ecuación 3: Cálculo de la tasa de pérdida de paquetes IP.	22
Ecuación 4: Cálculo de la tasa de error en paquetes IP.....	22
Ecuación 5: Cálculo del ancho de banda.....	22
Ecuación 6: Cálculo del throughput.....	22
Ecuación 7: Cálculo de la disponibilidad de servicio.....	23
Ecuación 8: Inercia o criterio de la suma de los cuadrados entre clústeres.....	68
Ecuación 9: Actualización de centroides en Mean-shift.....	70
Ecuación 10: Ecuación general para el cálculo del vector mean-shift.....	70
Ecuación 11: Ponderación de los criterios para la selección de la implementación de los parámetros WLAN.....	74
Ecuación 12: Ponderación de los criterios para la selección de la implementación de los KPIs...	76
Ecuación 13: Ponderación de los criterios para la selección de los algoritmos de clustering.....	76
Ecuación 14: Criterio AIC.....	114
Ecuación 15: Criterio BIC.....	114
Ecuación 16: Función de densidad de probabilidad de una distribución Normal.....	173
Ecuación 17: Función de densidad de probabilidad de una distribución Weibull	173
Ecuación 18: Función de densidad de probabilidad de una distribución Gamma	173
Ecuación 19: Función de densidad de probabilidad para una distribución Exponencial.....	176
Ecuación 20: Función de densidad de probabilidad para la distribución Normal Bimodal del tamaño de paquete de tráfico VDI.....	176
Ecuación 21: Función de densidad de probabilidad para la distribución Pareto Truncada	176
Ecuación 22: Función de densidad de probabilidad para la distribución Lognormal Truncada	178
Ecuación 23: Función de densidad de probabilidad para la distribución Laplaciana del jitter DL en tráfico VoIP.....	179
Ecuación 24: Función de densidad de probabilidad de una distribución Largest Extreme Value	179
Ecuación 25: Función de densidad de probabilidad de una distribución Uniforme.....	180
Ecuación 26: Función de pérdidas de propagación para el entorno doméstico	207
Ecuación 27: Función de pérdidas de propagación para el entorno interior de alta densidad. .	207

Lista de Siglas y Acrónimos

Sigla / Acrónimo Significado

3GPP	3rd Generation Partnership Project
5G	Fifth Generation of Mobile Networks
A-MPDU	Aggregate MAC Protocol Data Unit
A-MSDU	Aggregate MAC Service Data Unit
aaS	As a Service
AC	Access Category / Access Class
AC_BE	Access Category Best Effort
AC_BK	Access Category Background
AC_VI	Access Category Video
AC_VO	Access Category Voice
ACK	Acknowledgement
AI	Artificial Intelligence
AIC	Akaike Information Criterion
AP	Access Point
AR/VR	Augmented Reality / Virtual Reality
ARP	Address Resolution Protocol
ART	Adaptative Resonance Theory
BA	Block ACK
BEREC	Body of European Regulators for Electronic Communications
BIC	Bayesian Information Criterion
BIRCH	Balanced Iterative Reducing and Clustering using Hierarchies
BW	Bandwidth
BSS	Basic Service Set
BV	Buffered Video
CF	Contention Free / Clustering Feature

CFP	Contention Free Period
CIDR	Classless Inter-Domain Routing
CLARA	Clustering Large Applications
CLARANS	Clustering Large Applications based upon Randomized Search
CP	Contention Period
CR	Cognitive Radio
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSV	Comma-Separated Values
CTTC	Centre Tecnològic de Telecomunicacions de Catalunya
CURE	Clustering Using Representatives
D-AMPS	Digital Advanced Mobile Phone System
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DCF	Distributed Coordination Function
DL	Downlink
DLP	Direct Link Protocol
DNS	Domain Name System
DPSK	Differential Phase Shift Keying
DSSS	Direct Sequence Spread Spectrum
EHT	Extended High Throughput
E-M	Expectation-Maximization
EDCA	Enhanced Distributed Channel Access
EPS	Epsilon
ETSI	European Telecommunications Standards Institute
FC	Fractal Clustering
FCM	Fuzzy C-Means
FHSS	Frequency Hopping Spread Spectrum
FTTH	Fiber to the Home
FTP	File Transfer Protocol

GFSK	Gaussian Frequency Shift Keying
GI	Guard Interval
GMM	Gaussian Mixture Model
HCCA	HCF Controlled Channel Access
HCF	Hybrid Coordination Function
HC	Hybrid Coordinator
HE	High Efficiency
HTTP	Hypertext Transfer Protocol
HT	High Throughput
IEEE	Institute of Electric and Electronic Engineers
IaaS	Infrastructure as a Service
InH	Indoor Hotspot
IoT	Internet of Things
IPDV	IP Packet Delay Variation
IPER	IP Packet Error Rate
IPLR	IP Packet Loss Rate
IPTD	IP Packet Time Difference
IPTV	Internet Protocol Television
ITU-D	International Telecommunication Union – Development Sector
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
IPv4	Internet Protocol version 4
KPI	Key Performance Indicator
LOS	Line of Sight
LRD	Long-Range Dependent
MAC	Medium Access Control
MCS	Modulation and Coding Scheme
MGMT	Management
MIMO	Multiple Input, Multiple Output

ML	Machine Learning
MM	Mixture Model
MOS	Mean Opinion Score
MPDU	MAC Protocol Data Unit
MPEG-TS	Moving Picture Experts Group - Transport Stream
MSDU	MAC Service Data Unit
MST	Minimum Spanning Tree
MTBF	Mean Time Between Failures
MTRS	Mean Time to Restore Service
MU-MIMO	Multiple User – Multiple Input Multiple Output
MU-OFDMA	Multiple User - Orthogonal Frequency Division Multiple Access
NLOS	Non-Line of Sight
NP	Network Performance
NS-3	Network Simulator 3
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
OPTICS	Ordering Points To Identify the Clustering Structure
OR	Opinion Rate
OSI	Open System Interconnection
P2P	Peer to Peer
PaaS	Platform as a Service
PAM	Partitioning Around Medoids
PCF	Point Coordination Function
PHY	Physical Layer
PIFS	Point Inter Frame Space
QAM	Quadrature Amplitude Modulation
QAP	QoS AP
QoE	Quality of Experience

QoS	Quality of Service
QSTA	QoS STA
RMa	Rural Macrocell
ROCK	Robust Clustering Using Links
RSSI	Received Signal Strength Indicator
RTC	Real-Time Communication
RTS/CTS	Request to Send/Clear to Send
SaaS	Service as a Service
SIFS	Short Inter Frame Space
SNR	Signal to Noise Ratio
SOM	Self-Organizing Map
STA	Station
STBC	Space-Time Block Coding
STING	Statistical Information Grid
TCP	Transmission Control Protocol
TGax	Task Group AX
TWT	Target Wake Time
TXOP	Transmission Opportunity
U-APSD	Unscheduled Automatic Power Save Delivery
UDP	User Datagram Protocol
UL	Uplink
Uma	Urban Micro-cell
Umi	Urban Macro-cell
VHT	Very High Throughput
VoIP	Voice over Internet Protocol
VoD	Video on Demand
VDI	Virtual Desktop Infrastructure
VTC	Video Teleconferencing

WAVE Wireless Access in Vehicular Environments

WD Wireless Display

Wi-Fi Wireless Local Area Network

WLAN Wi-Fi Protected Access 3

1. Introducción

La mejora continua y la innovación en el ámbito de las comunicaciones inalámbricas tratan de satisfacer la demanda de nuevas características y servicios en el sector. En la actualidad reciente, esta evolución se ha visto directamente afectada por la pandemia del Covid-19, que ha ensalzado, entre otros, el valor real de las opciones tecnológicas modernas. [1]

La tendencia en las redes de telecomunicaciones previa a los sucesos de los últimos años [2] era la de un mundo cada vez más conectado, digitalizado, distribuido y diverso (Figura 1). La expansión de los dispositivos IoT (Internet of Things) provocó que cualquier objeto de uso cotidiano fuera capaz de procesar datos y disponer de acceso a la red. En consecuencia, se generó un **nuevo abanico de escenarios** que incluía distintas densidades de dispositivos en entornos de toda clase, como el hogar, el transporte, la medicina y la salud, la industria y el ámbito empresarial. [3]

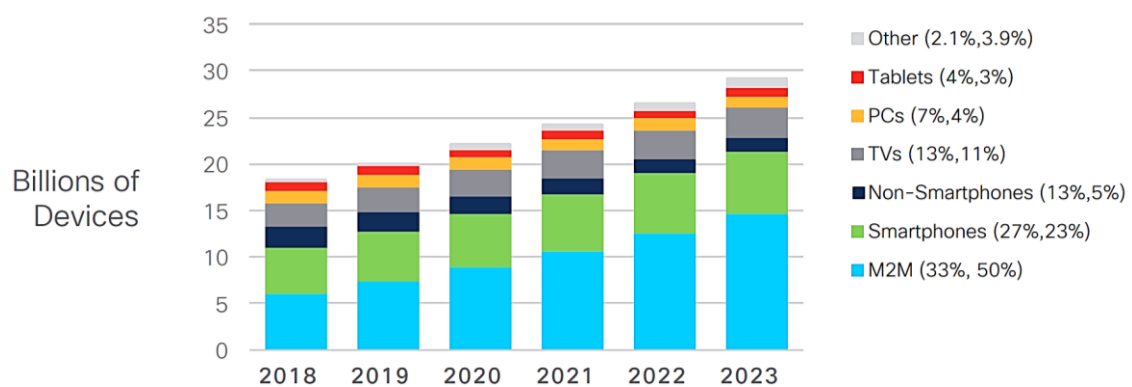


Figura 1: Tendencia de crecimiento global de dispositivos y conexiones 2018-2023 [4]

A su vez, las tecnologías de comunicaciones inalámbricas [5] se hallaban en un momento clave, cuyo progreso llevaba a la introducción de las últimas soluciones que trataban de afrontar los requerimientos de los nuevos servicios, dispositivos y entornos. Entre ellas, el inicio del despliegue de la infraestructura de red acogía la **quinta generación de tecnologías móviles (5G)**, y mientras tanto, también tenían lugar los inicios de actualización de la tecnología WLAN (Wireless Local Area Network) al **estándar IEEE 802.11ax** [6] –también denominado Wi-Fi 6 por la Wi-Fi Alliance–. Esta última, además de la mejora de múltiples funciones respecto a sus versiones previas, refuerza el apoyo a la descarga de datos (mobile data offloading) sobre las generaciones móviles. [7] De este modo, la **tecnología WLAN** adquiere un papel principal en la evolución de las redes, ya que **continúa ganando impulso** de forma paralela al desarrollo de estas [4].

No obstante, a la complejidad de los nuevos escenarios se le han de añadir también los requisitos derivados de la **irrupción de la computación en la nube y la variedad de modelos aaS** (as a Service) de los últimos años, que se deben, en gran medida, al impulso recibido en la transformación digital de los negocios. [8] Los servicios en la nube –IaaS (Infrastructure as a Service), PaaS (Platform as a Service), SaaS (Software as a Service), etc.– tienen distintas características y su provisión se diferencia en términos de control, manejo y mantenimiento principalmente. [9] Sin embargo, todos incluyen

considerables requerimientos de disponibilidad, agilidad, escalabilidad, elasticidad y tolerancia a fallos, entre otros. [8]

Es bajo esta premisa donde toma gran relevancia la **calidad de servicio (Quality of Service, QoS)**, ya que no se contemplan sólo el despliegue de nuevos dispositivos y el proceso de cambio de las redes, sino que se introducen servicios con estrictos requisitos y se produce, además, un aumento a nivel de exigencia en los servicios tradicionales. [7]

Históricamente, la QoS ha sido asociada a los requisitos mínimos para el correcto funcionamiento de distintos servicios como voz y datos [10]. La QoS se puede definir, por ende, a través de una serie de parámetros de calidad o métricas, denominados **indicadores de rendimiento (Key Performance Indicator, KPI)**. El uso de KPIs – como la medida de la pérdida de paquetes, tasa de error de bits, retardo de transmisión, variación de retardo, etc.– permite tanto el establecimiento de objetivos medibles para los servicios, como evaluar su funcionamiento (Network Performance, NP), monitorizar tendencias y realizar mejoras. [11]

Sin embargo, así como las redes y servicios han ido evolucionando, el concepto de QoS lo ha hecho también. En vez de limitarse a la perspectiva tradicional, donde se miden los niveles de degradación de la transmisión de servicios básicos, se incorporan aspectos subjetivos (Figura 2) relacionados a la **calidad de la experiencia (Quality of Experience, QoE)** [12] y una **visión global del sistema**. [11]

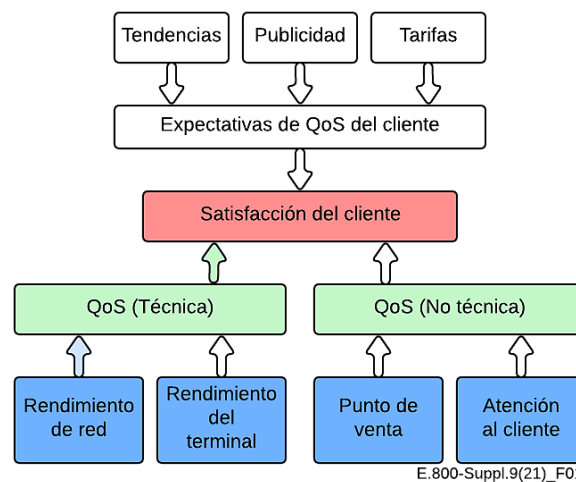


Figura 2: Punto de vista técnico y no técnico de la QoS, incluyendo la satisfacción del cliente [13]

Asimismo, se trata de ajustar los procedimientos de QoS a los nuevos **escenarios heterogéneos** –con equipos y servicios convencionales, dispositivos IoT, comunicaciones entre máquinas (Machine to Machine, M2M), etc.– y, además, satisfacer el aumento en la exigencia de los usuarios en movilidad, conectividad y calidad. [14] Por consiguiente, resulta ineludible prestar atención al desarrollo de **nuevas soluciones** que permitan una gestión de la QoS apropiada a la complejidad actual de las redes y servicios.

Ante estos nuevos retos y los grandes avances en las tecnologías de comunicaciones inalámbricas –5G, IEEE 802.11ax, etc.– surge la posibilidad de utilizar **técnicas de**

aprendizaje automático (Machine Learning, ML) para cumplir los propósitos de QoS presentes.

En los últimos años se han realizado multitud de estudios (Tabla 1) sobre la aplicabilidad de las técnicas de ML para la mejora de múltiples características de redes inalámbricas, tanto a nivel de rendimiento como de procesado de la información. [15]

Objetivo	Ámbito
Mejora de rendimiento	Análisis del espectro de radio
	Análisis de MAC
	Predicción de red
Procesamiento de información	Monitorización de infraestructura IoT
	Seguridad inalámbrica
	Posicionamiento inalámbrico
	Reconocimiento de actividad

Tabla 1: Aplicaciones estudiadas sobre la aplicación de ML en redes inalámbricas [15]

Debido al auge de las técnicas de aprendizaje automático en el sector de las redes inalámbricas y la necesidad de nuevos métodos para la gestión de la calidad de servicio que se ajusten a la actual situación del sector, [16] este estudio persigue **validar la utilización de algoritmos de ML sobre diferentes procedimientos de gestión de QoS**.

Para ello, recoge gran parte de la base teórica establecida en el **Trabajo Fin de Grado Análisis De Técnicas De Aprendizaje Automático Para La Mejora De La Calidad De Servicio En Redes Inalámbricas** [17]. Dicho primer trabajo se toma como **punto de partida** para un estudio de mayor profundidad sobre el procedimiento de diferenciación de entornos, y **se extiende notablemente su alcance** al incidir en una gran parte de las especificaciones propias de la tecnología para realizar una caracterización afín de los casos de uso actuales de la misma. Además, se persigue enriquecer también el estudio mediante la inclusión del análisis del lado de usuario, dando cabida a procedimientos de clasificación de perfiles.

En concreto, el **experimento inicial** en el que se basa este trabajo consta de la validación de dos algoritmos comunes de clustering –K-Means y DBSCAN– en la diferenciación de entornos [17] sobre entornos de simulación sencillos, suponiendo una **primera aproximación a la caracterización de escenarios y al uso de simuladores de red**.

En dicho proyecto [17] se hace uso de una **herramienta de simulación del protocolo IEEE 802.11ax** denominada Komondor [18] como resultado del análisis de alternativas, pero a pesar de introducir múltiples características WLAN y ser suficiente para justificar la **validez de las técnicas de clustering** para el ámbito del estudio, presenta algunas limitaciones en términos de caracterización de escenarios, tráfico de red y usuarios.

Por lo tanto, este estudio parte del conocimiento extraído del Trabajo Fin de Grado sobre simulación de red y tipos de algoritmos de aprendizaje automático. En base al análisis de alternativas de simuladores de red, se toma como referencia **NS-3** tras revisar su aplicabilidad a la actualidad de la tecnología (Anexo III: NS-3). Asimismo, aprovechando las bondades del aprendizaje no supervisado, y con ánimo de enriquecer la propuesta realizada entonces, se centra la atención en las **técnicas de clustering** y

su validación sobre simulaciones de red en las aplicaciones de diferenciación de entornos y clasificación de perfiles.

Sin embargo, la **contribución del trabajo** va más allá del análisis teórico de técnicas de ML, ya que comprende, además de una investigación para la búsqueda de los algoritmos óptimos para cada caso, el análisis de casos de aplicación de actualidad de la tecnología IEEE 802.11 para el diseño de simulaciones realistas y la implementación de dichas técnicas sobre programas para el tratamiento de resultados de simulación de red.

En definitiva, para el desarrollo del estudio se han de realizar un total de **cuatro grandes tareas**, en el orden que se especifica a continuación.

En primer lugar, se lleva a cabo un **análisis teórico** en profundidad de los **entornos de aplicación** principales de la **tecnología WLAN** y los **perfiles de usuario** existentes tomando como referencia el modelado de tráfico. De este modo, se permite la extracción de patrones y la definición de parámetros para el diseño de simulaciones de red.

Acto seguido, se hace uso de un **simulador de redes** que permite una cuidadosa caracterización de los escenarios de uso de las redes inalámbricas de área local e implemente los procedimientos principales de la última versión del protocolo IEEE 802.11. Entonces, se realizan múltiples simulaciones que hagan posible su posterior estudio y tratamiento con técnicas de ML.

En tercer lugar, se implementan diferentes técnicas de **aprendizaje automático basadas en clustering** que permitan ejecutar los procedimientos de gestión de QoS para la extracción del contexto del usuario y la red: **diferenciación de entornos y clasificación de perfiles**.

Por último, se procede a realizar la **validación de los algoritmos** para las tareas de gestión de QoS, realizando una comparación objetiva del desempeño de estos frente a los procedimientos establecidos, y seleccionando las técnicas más aptas como conclusión.

2. Contexto

El presente documento, como se adelantaba en la Introducción, tiene como propósito aprovechar las bondades de las técnicas de aprendizaje automático en el ámbito de la calidad de servicio de las comunicaciones inalámbricas de área local. La asociación de estas áreas de conocimiento exige una correcta contextualización –previa al desarrollo del proyecto– por lo que, a continuación, se realiza una breve presentación y se detalla su situación actual.

2.1. Tecnología WLAN: 802.11

Las **redes inalámbricas de área local o WLAN** se caracterizan por una baja potencia de transmisión y una limitada área de cobertura. Hacen uso de bandas de frecuencia sin licencia para proporcionar comunicaciones de altas capacidades de datos, dotando de conectividad a múltiples equipos y permitiendo, a su vez, portabilidad y movilidad de los usuarios. [19]

El estándar de referencia de las redes WLAN es la **familia de normas IEEE 802.11**, que consta de múltiples protocolos definidos, como su nombre indica, por el Instituto de Ingenieros Eléctricos y Electrónicos (Institute of Electric and Electronic Engineers, IEEE). Estos protocolos han ido evolucionando a lo largo de los años, incluyendo nuevas características y mejorando las ya presentes.

El estándar IEEE 802.11 cubre los niveles más bajos de la arquitectura OSI (Open System Interconnection), específicamente las capas MAC (Medium Access Control) y PHY (Physical). Por lo tanto, engloba las especificaciones de nivel físico y control de acceso al medio necesarias para la implementación y funcionamiento de las redes WLAN.

2.1.1. Introducción al Estándar IEEE 802.11

La **arquitectura** más sencilla de una red 802.11 se compone principalmente de los elementos que aparecen en la Figura 3. Un conjunto de servicios básicos (Basic Service Set, BSS) dispone de uno o más equipos terminales (Station, STA) inalámbricos y de una estación base (Access Point, AP), la cual se conecta a un dispositivo de interconexión que dé salida a Internet –el AP suele integrarse en el mismo equipo que el dispositivo de interconexión–.

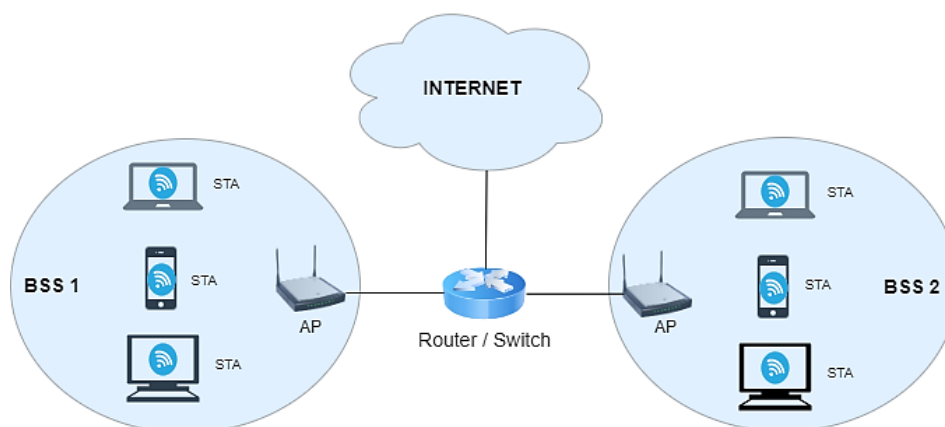


Figura 3: Arquitectura sencilla de una red WLAN.

La arquitectura anteriormente presentada se corresponde al denominado **modo infraestructura**, donde los equipos finales hacen uso de los servicios de red a través de APs. Sin embargo, también existe la posibilidad de conectar las estaciones IEEE 802.11 bajo una red ad hoc –conectándose los equipos entre sí, sin control central ni salida a otras redes– aunque no se contempla en el desarrollo de este documento, puesto que se trata de una aplicación específica de comunicaciones de naturaleza distinta al análisis de entornos y perfiles de usuario en la provisión de servicios de red del presente estudio.

Asimismo, a **nivel MAC** originalmente se especifican dos modos de funcionamiento básicos: las técnicas de acceso al medio centralizadas y distribuidas.

DCF (Distributed Coordination Function) hace uso de **CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance)** como mecanismo de control de acceso al medio, de forma que las estaciones deben conocer si el medio se encuentra disponible antes de transmitir (Request to Send /Clear to Send, RTS/CTS). Para ello, se llevan a cabo procedimientos de nivel PHY para la escucha y análisis del canal durante periodos de contención (Contention Period, CP). Sin embargo, comprende una técnica best effort que no ofrece garantías de QoS, y cuyo rendimiento se ve afectado por el aumento de equipos finales. [20]

PCF (Point Coordination Function) propone un mecanismo de encuesta (polling) de forma centralizada, de modo que el AP consulta a los terminales si desean transmitir en periodos alternados (Contention Free Period, CFP) con los de contención, donde tiene lugar el funcionamiento clásico de acceso aleatorio mediante escucha del medio. Aun así, se trata de una técnica con varias limitaciones, como la simplicidad del reparto de turnos para el acceso entre terminales, o la variabilidad de los periodos de transmisión establecidos por turnos debido a las características del tráfico.

Estos procedimientos son revisados y mejorados en la especificación IEEE 802.11e dedicada a la QoS [21], pero antes de llegar a dicha norma, se procede a presentar la evolución del estándar.

2.1.2. Evolución del Estándar IEEE 802.11

Así como las redes inalámbricas se han ido expandiendo y adaptando a las nuevas necesidades, también lo ha hecho el estándar IEEE 802.11. En la Figura 4 se puede observar la analogía entre las diferentes versiones de Wi-Fi –reconocidos bajo dicha nomenclatura de forma oficial a partir de Wi-Fi 4– y las normas IEEE 802.11.

Cada generación se ha visto afectada por el desarrollo de nuevos servicios de red y mayores exigencias para proveer tanto dichos servicios como los ya existentes de forma satisfactoria. Asimismo, con cada estándar resulta fácilmente apreciable el contraste de las especificaciones debido a la mejora tecnológica y el consecuente aumento en el rendimiento general.

En la **primera versión del protocolo** [22] sólo se disponía de una única banda frecuencial en los 2,4 GHz, y con un ancho de canal de 20 MHz y se utilizaban tecnologías de transmisión como infrarrojos, FHSS (Frequency Hopping Spread Spectrum) junto

con modulación GFSK (Gaussian Frequency Shift Keying), y DSSS (Direct Sequence Spread Spectrum) con DPSK (Differential Phase Shift Keying). En cuanto a velocidad de transmisión (throughput), se alcanzaban como máximo 2 Mbps.

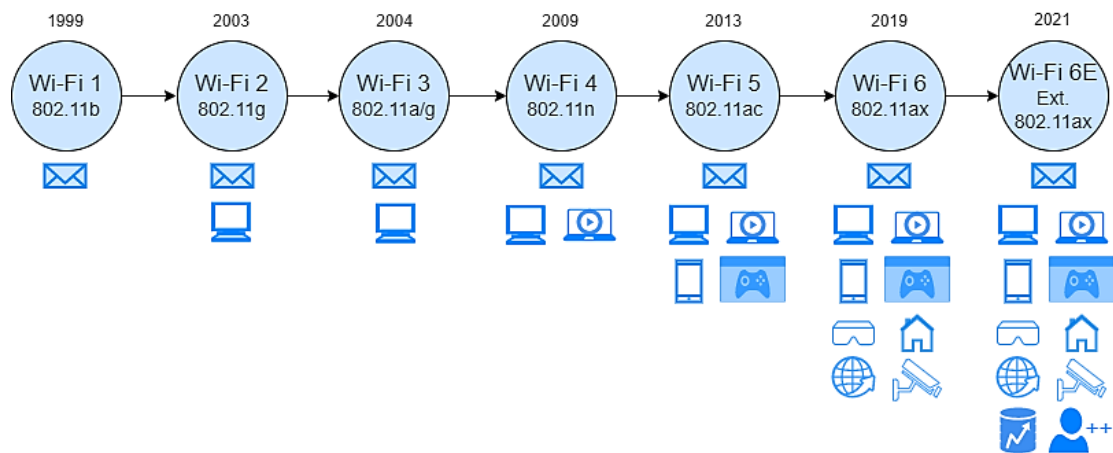


Figura 4: Relación entre la certificación de Wi-Fi Alliance y los protocolos IEEE 802.11. [23]

Las nuevas técnicas de modulación, multiplexación y gestión del espectro, tanto en las tecnologías utilizadas como en la asignación de bandas y canales de transmisión, resaltan en comparación con las propiedades anteriormente presentadas. En la Tabla 2 se muestran de forma resumida las características principales de las últimas versiones de Wi-Fi.

Características	Wi-Fi 5 (802.11ac)	Wi-Fi 6 (802.11ax)	Wi-Fi 6E (Ext. 802.11ax)
Banda de frecuencia	5 GHz	2,4 & 5 GHz	(2,4 & 5 & 6 GHz)
Anchura de canal	20, 40, 80, 80+80 160 MHz	20, 40, 80, 80+80, 160 MHz	20, 40, 80, 80+80, 160 MHz
Tasa de datos máxima	~7 Gbps	~9,6 Gbps	~9,6 Gbps
Modulación de frecuencia	256 QAM con MCS 1 a 9	1024 QAM con MCS 1 a 11	1024 QAM con MCS 1 a 11
OFDMA	-	DL/UL (MU-OFDMA)	DL/UL (MU-OFDMA)
MIMO	DL	DL & UL	DL & UL
Ahorro de energía	STBC, U-APSD	STBC, U-APSD, Target Wake Time (TWT)	STBC, U-APSD, Target Wake Time (TWT)
Eficiencia espectral	-	Celdas de color	Celdas de color

Tabla 2: Resumen de características principales de las últimas versiones de Wi-Fi. [24]

El protocolo **IEEE 802.11ax** –correspondiente a Wi-Fi 6– se basa y sirve del éxito de su antecesor (IEEE 802.11ac), pero introduce múltiples mejoras en sus técnicas y recursos. [22] [25]

Respecto a las tecnologías de transmisión, se hace uso de **OFDMA (Orthogonal Frequency Division Multiple Access)**, que permite la compartición de canales –cada cliente puede ser asignado a un segmento temporal y frecuencial en el canal– para

aumentar la eficiencia en la red y disminuir la latencia del tráfico UL (Uplink) y DL (Downlink) en entornos de alta demanda.

Con relación a las técnicas de modulación, la utilización de **1024 QAM** permite un aumento de throughput a través de una codificación más eficiente haciendo uso del mismo espectro.

De forma similar a la evolución de OFDM (Orthogonal Frequency Division Multiplexing) hacia su versión multiusuario, la tecnología **MU-MIMO (Multiple User – Multiple Input Multiple Output)** para las comunicaciones inalámbricas multicamino introduce la capacidad de transmitir a múltiples usuarios sobre la tecnología MIMO (Multiple Input Multiple Output).

De igual modo, se dispone de **canales de hasta 160 MHz**, que permiten incrementar el ancho de banda (bandwidth, BW) provisionando mejores funcionalidades a baja latencia.

Por otro lado, entre las características introducidas destaca el **beamforming** [26], que mejora la relación señal a ruido (Signal to Noise Ratio, SNR) y posibilita eliminar fuentes de interferencia indeseables y enfocar las señales transmitidas a localizaciones específicas. Esto resulta en mayores tasas de datos a un rango dado, obteniendo mayor capacidad en la red.

Además, se incluyen una serie de mejoras en diferentes planos. En términos de señalización, se utiliza una **RSSI (Received Signal Strength Indicator) inferior** a los protocolos previos, lo que proporciona una señalización robusta y eficiente. Respecto al consumo energético, se incluye un mecanismo de mejora de la eficiencia denominado **TWT (Target Wake Time)**, y a nivel de seguridad se utiliza la **tercera versión de Wi-Fi Protected Access (WPA3)**.

Por último, la **extensión del protocolo IEEE 802.11ax (2021)** –no se trata de un estándar, sino de una ampliación– introduce la banda de **6GHz**, lo que supone una gran mejora al plantear una solución sólida al aumento de la popularidad de la tecnología inalámbrica y el incremento en capacidad, que ha llevado a que el uso de las bandas de 2.4 y 5 GHz haya alcanzado sus límites. [27]

La inclusión de la banda de 6 GHz, además, permite mayor throughput, menor interferencia y un incremento considerable en la capacidad, al ofrecer más canales de transmisión: catorce canales de 80 MHz y siete canales de 160MHz. Sin embargo, al utilizar un distinto rango frecuencial y requerir OFDMA **no es del todo compatible con dispositivos legacy** (IEEE 802.11 a/b/g/n/ac), lo que resulta en un aspecto a tener en consideración a la hora de proponer redes con heterogeneidad de estándares WLAN. A su vez, también supone menor interferencia y no sufrir ralentizaciones de equipos de las generaciones anteriores. [27] [28]

De forma global, el protocolo IEEE 802.11ax trata de dar solución no sólo a la demanda de flexibilidad y escalabilidad de las aplicaciones de última generación, sino también a los requisitos de QoS de éstas y de los nuevos escenarios de red. A continuación, se

presentan de forma más extendida los procedimientos desarrollados en el estándar para la gestión y aseguramiento de la QoS.

2.1.3. Estándar IEEE 802.11 y QoS: IEEE 802.11e

En los anteriores apartados se ha introducido de forma resumida el funcionamiento de la tecnología IEEE 802.11 junto con sus características más significativas. La familia de normas inalámbricas incluye una versión específica para la mejora de la QoS sobre aplicaciones de WLAN: el protocolo IEEE 802.11e. [29] A través de este, se realiza la modificación de los procedimientos de acceso al medio definidos tradicionalmente para incluir mecanismos de QoS.

En origen, los métodos principales de la capa MAC comprendían la función DCF que emplea CSMA/CA y RTS/CTS, y la función PCF para la coordinación de APs. Con la introducción de IEEE 802.11e se define **la función de coordinación híbrida (Hybrid Coordination Function, HCF)**, que posibilita la provisión de servicios con QoS a través de la priorización y parametrización a nivel MAC, sobre DCF (Figura 5). El soporte de QoS a los dispositivos se ofrece tanto en los puntos de acceso (QoS AP, QAP) como en las estaciones (QoS STA, QSTA). [29]

HCF presenta múltiples mejoras frente a DCF y PCF a través de dos mecanismos: **EDCA (Enhanced Distributed Channel Access)** y **HCCA (HCF Controlled Channel Access)**. En este caso, el derecho a hacer uso del canal se gestiona a través de oportunidades de transmisión (Transmission Opportunity, TXOP), el tiempo que dispone una QSTA para comunicarse –inferior a un tiempo límite–.

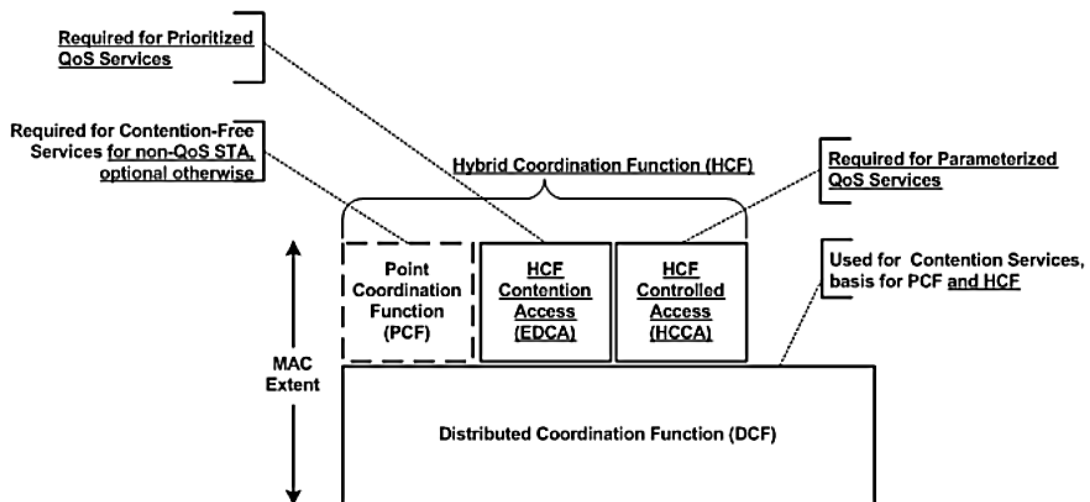


Figura 5: Arquitectura IEEE 802.11e de la capa MAC. [21]

Asimismo, se introduce el protocolo **DLP (Direct Link Protocol)**, que permite la transmisión directa entre QSTA que comparten QAP para el uso eficiente del medio. [30]

EDCA se trata de la evolución de DCF, y es un mecanismo **basado en contención** que permite diferenciar paquetes a través de clasificación y asignación de prioridades. En concreto, el soporte de QoS se provee por medio de la introducción de categorías de

acceso (Access Category, AC) en las estaciones QSTA. EDCA define un total de cuatro ACs, de forma que cada uno tiene distinto nivel de prioridad, y sirve para identificar diferentes tipos de tráfico: background (AC_BK), best effort (AC_BE), vídeo (AC_VI) y voz (AC_VO). [30]

Dicha clasificación establece la adaptación de múltiples parámetros de lo originalmente conocido como DCF a las distintas clases, de forma que el procedimiento de contienda depende de los ACs. Asimismo, la implementación de las ACs se realiza a través del establecimiento de cuatro colas separadas en cada QSTA, y a cada cola se le asignan sus propias características.

HCCA [30], por su parte, comprende el mecanismo de acceso al medio que, de forma similar a PCF, está **basado en polling**. En concreto, HCCA utiliza un coordinador híbrido (Hybrid Coordinator, HC) ubicado habitualmente en el QAP en el caso de las WLAN en modo infraestructura. El HC es el encargado de asignar los TXOPs a las QSTAs cuando adquiere el control del canal. Sin embargo, a diferencia de PCF, HCCA permite consultar a los QSTAs durante los periodos de contención, y se tienen en cuenta los requerimientos específicos de los flujos del QSTA en la planificación de los paquetes.

Cuando el HC tiene el control del canal, consulta a las QSTAs en turnos de acuerdo a su lista de polling. Para incluirse en ella, los QSTAs envían solicitudes de reserva de QoS utilizando una trama específica de gestión de QoS, necesitando cada flujo individual una solicitud de reserva particular. En la misma se incluyen parámetros de especificación de tráfico como la tasa media de datos, el retardo máximo, o el tamaño nominal de los paquetes.

En definitiva, HCCA permite un **mejor control del tráfico**, haciendo posible introducir en el mecanismo de polling los requerimientos de QoS desde las estaciones, y haciendo posible la transmisión ordenada en función de estos para asegurar una mejor QoS al disponer de información sobre el tráfico a priorizar.

De este modo, las nuevas funciones introducidas en el protocolo **IEEE 802.11e** permiten una **mejora en la QoS** al introducir mecanismos de **clasificación y priorización a nivel MAC**. Aun así, en la actualidad existen una amplia variedad de técnicas que pueden utilizarse sobre las redes inalámbricas para distintos propósitos, como la automatización de la gestión de la QoS o la mejora global de la misma. En dicho ámbito, destacan las técnicas de aprendizaje automático, sobre las que se realizan cada vez mayor cantidad de estudios para la optimización de tareas a través algoritmos que mejoran con la experiencia. [31] Por consiguiente, en el próximo apartado se presentan múltiples propuestas para la mejora de diversas características y procedimientos sobre WLAN mediante aprendizaje automático.

2.2. Aprendizaje Automático en Redes Inalámbricas

La continua evolución de las redes y su creciente complejidad [16] han dado cabida a la realización de múltiples estudios para la mejora de rendimiento en el área de las tecnologías inalámbricas. Sin embargo, el **análisis** de los múltiples **parámetros de rendimiento**—como pueden ser la tasa de pérdida de paquetes, latencia, tasa de error

de bit, throughput, etc.– y la **optimización a diferentes niveles de arquitectura** se ha vuelto un reto. Aún más, la dificultad se acentúa con la alta demanda de tráfico y la heterogeneidad presente en las redes actuales. Para afrontarlo, cada vez se estudia de forma más intensiva el uso de ML para el desarrollo de aproximaciones capaces de extraer patrones y predecir tendencias.

En la Tabla 3 se resume una gran parte de las aplicaciones sobre las que se han realizado estudios de aprendizaje automático para la mejora de tareas específicas en redes inalámbricas [31], centrándose, primero, en la mejora del rendimiento, y segundo, en el procesamiento de la información. De la misma forma, se distinguen diferentes áreas de aplicación de los dos objetivos mencionados en la segunda columna.

El análisis de los datos generados a través de infraestructura de monitorización de tráfico de red y mediante sensores inalámbricos da lugar a la posibilidad de mejorar las configuraciones de red, optimizando los ajustes para obtener, entre otras características, mayor QoS. De este modo, se presentan múltiples propuestas de estudio a nivel de espectro de radio, nivel MAC y nivel de red. [31]

Objetivo	Área	Ejemplo de Aplicación
Mejora de rendimiento	Análisis del espectro de radio	Reconocimiento automático de modulación
		Identificación de interferencia inalámbrica
	Análisis de nivel MAC	Identificación MAC
		Detección de interferencia inalámbrica a nivel de paquete
		Predicción del espectro
	Predicción de red	Predicción de rendimiento de red
Predicción de tráfico de red		
Procesamiento de la información	Monitorización de infraestructura IoT	Ecosistema inteligente: smart farming, smart mobility, smart city, smart grid
	Seguridad inalámbrica	Identificación de dispositivos mediante huellas digitales
	Localización inalámbrica	Indoor
		Outdoor
Reconocimiento de actividad	Señales inalámbricas de reconocimiento de actividad	

Tabla 3: Resumen de aplicaciones de aprendizaje automático sobre redes inalámbricas. [31]

El **análisis del espectro de radio** hace referencia al estudio de los datos inalámbricos obtenidos mediante sensores que permiten inferir el uso del espectro. El reconocimiento de tipo de modulación automático es una técnica clave en diferentes aplicaciones, ya que permite percibir el tipo de sistemas de comunicación y emisores presentes en el entorno de radio. Sobre dichas técnicas, la implementación de clasificadores basados en ML permite optimizar el proceso de clasificación y proveer mejores generalizaciones frente a conjuntos de datos que no se hayan estudiado de forma previa. Con relación a la aplicación de ML sobre la identificación de interferencia inalámbrica, se persigue reconocer el tipo de emisores inalámbricos en el entorno para la investigación efectiva de evasión de interferencia y mecanismos de coexistencia de tecnologías en el entorno bajo estudio.

A **nivel MAC**, la identificación MAC suele emplearse en aplicaciones de cognitive radio (CR) para promover la comunicación y coexistencia entre tecnologías que utilizan distintos protocolos. A través de ML, se hace posible determinar la información temporal sobre huecos libres en el espectro de radio y obtener beneficios potenciales en términos de rendimiento. Respecto a la detección de interferencia inalámbrica a nivel de paquete, se han desarrollado varias propuestas de ML para la detección del tipo de interferencia que degrada el rendimiento de red, y en relación con la predicción del espectro, el objetivo es la predicción de disponibilidad del medio para minimizar las colisiones en la transmisión de información.

En cuanto a la **capa de red**, la realización de predicciones persigue inferir el rendimiento o tráfico a nivel IP a través de históricos de datos, de forma que los métodos de ML hagan posible predecir el rendimiento o parámetros clave de los dispositivos y se puedan realizar configuraciones de red que optimicen el rendimiento general de esta. Además, la predicción de tráfico resulta crucial para la evaluación y mejora del rendimiento de los sistemas inalámbricos en múltiples aspectos, como en términos de energía, gestión de recursos y evasión de posibles casuísticas indeseables como la congestión en la red.

Por otra parte, los sensores inalámbricos y las aplicaciones móviles instaladas en diferentes dispositivos permiten el registro de datos de nivel de aplicación de forma recursiva, convirtiéndolos en recolectores de datos que además preprocesan los mismos y los almacenan en la nube para su computación mediante distintas tecnologías. [31]

Algunos ejemplos de aplicación son la **monitorización de infraestructura IoT** y los ecosistemas inteligentes –smart farming, smart mobility, smart city y smart grid– que permiten hacer un uso eficiente de los recursos básicos de cada área, como el consumo energético, aprovechamiento de recursos naturales, productividad, etc. Asimismo, se han desarrollado una gran variedad de propuestas a través de ML, como la detección de dispositivos legítimos en la red a través de huellas digitales en el ámbito de la **seguridad**, la obtención de la **ubicación de dispositivos** a través de las señales recibidas de los mismos o por medio del estudio de uso de los canales inalámbricos, o el **reconocimiento de actividad humana** –estimación de movimiento, posición o disposición de las personas en entornos dados– a través de propiedades de las señales inalámbricas.

Dentro del amplio abanico de posibles aplicaciones de ML presentados, **el presente estudio** se enmarca en el gran primer bloque, ya que el objetivo es **promover un aumento del rendimiento de red**. Para ello, se estudia la aplicabilidad de las técnicas de clustering sobre el análisis de métricas de red para la mejora de la QoS a través de la diferenciación de entornos y la clasificación de perfiles de usuario.

3. Objetivos y Alcance

La extensión del estudio comprendido en este documento exige el desglose de los objetivos y el alcance del Trabajo Fin de Máster en distintos apartados. Para ello, se introduce el propósito principal en primer lugar y se incluyen los objetivos más específicos después. Para finalizar, se define el alcance del proyecto.

3.1. Objetivo Principal

El **objetivo principal** del Trabajo Fin de Máster es realizar un estudio de la aptitud de las técnicas de clustering en los procedimientos de mejora de la calidad de servicio en redes inalámbricas de área local de diferenciación de entornos y clasificación de perfiles.

3.2. Objetivos Secundarios

Debido a la extensión de tareas a realizar implícitas en el objetivo principal, se han establecido una serie de objetivos secundarios que posibiliten, una vez sean cumplidos, alcanzar el propósito general del proyecto.

En primer lugar, se desarrolla un estudio cuyo objetivo principal es **caracterizar** de forma teórica **casos de uso actuales de la tecnología IEEE 802.11** –incluyendo distintos tipos de escenarios y tráfico generado por los usuarios– para obtener modelos reproducibles a través de herramientas de simulación.

En segundo lugar, se persigue **obtener múltiples simulaciones congruentes con el estudio inicial** mediante la implementación de los parámetros definidos en el mismo en una **herramienta de simulación de WLAN**.

En tercer lugar, el objetivo es **procesar los resultados** del simulador **mediante aprendizaje automático**, haciendo uso de técnicas de clustering para llevar a cabo los procedimientos de gestión de la QoS de **diferenciación de entornos y clasificación de perfiles**.

Por último, se busca obtener la **validación de la aptitud de los algoritmos de ML para la mejora de la QoS** a través de las tareas de gestión mencionadas con anterioridad.

3.3. Alcance

El diagrama de la Figura 6 muestra el alcance del proyecto, que se divide en **tres grandes bloques**: una etapa formativa en la que se obtienen los conocimientos requeridos para el desarrollo del trabajo, una etapa práctica en la que se aplican los recursos adquiridos previamente en la metodología, y una etapa analítica en la que se extraen las conclusiones que ponen fin al estudio.

En la primera etapa, bajo el nombre de **Formación Básica y de Profundización**, se ahonda en los conocimientos de las tecnologías de comunicación inalámbricas, prestando especial atención a la calidad de servicio, y se estudian las técnicas de aprendizaje automático y su aplicabilidad para la mejora de los procedimientos de gestión de la QoS. Con dicho propósito, se investiga sobre el estándar IEEE 802.11, los procedimientos y parámetros de QoS principales, y la evolución de ambos. Asimismo,

se estudian los algoritmos de ML no supervisado centrados en clustering, el funcionamiento de los simuladores de red y la forma de implementar escenarios WLAN con distintas características.

La segunda etapa comprende la **Metodología**, es decir, la puesta en práctica de los conocimientos adquiridos en la fase formativa. A su vez, la metodología se divide en cinco sub-bloques. En un inicio, se realiza un estudio en profundidad de los escenarios de aplicación de la tecnología WLAN, teniendo en cuenta tanto los entornos como el tráfico generado por los usuarios. A continuación, tiene lugar la familiarización con la herramienta de simulación de red y se lleva a cabo la implementación de los resultados obtenidos del estudio teórico previamente desarrollado. Entonces, se realiza la selección de algoritmos de ML que permitan el procesamiento de los datos de salida del simulador, y se adecúan a la diferenciación de entornos y clasificación de perfiles. Una vez elegidos los algoritmos, se lleva a cabo su implementación sobre los procedimientos de mejora de la QoS. Por último, se procesan los resultados de las simulaciones con ML para su posterior análisis.

La última etapa la conforma el **Análisis y Extracción de Conclusiones**. Inicialmente, se consideran y examinan los resultados del procesamiento de datos realizado con ML. Después, se procede a comparar los algoritmos utilizados y, para terminar, se obtienen las conclusiones del estudio.

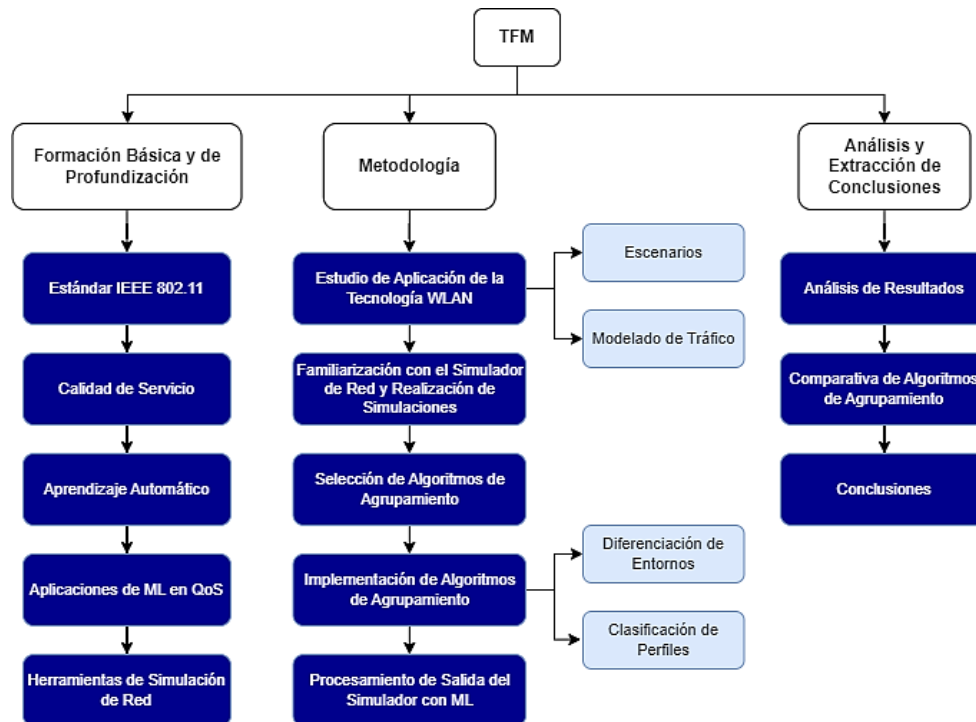


Figura 6: Alcance del Trabajo Fin de Máster

4. Beneficios

La realización de este proyecto se justifica mediante la pluralidad de beneficios que aporta en diferentes áreas. El estudio provee de una base sólida para la mejora de la calidad de servicio en redes inalámbricas, por lo que resulta de gran relevancia para múltiples grupos de interés, desde los propios usuarios de la tecnología hasta la cadena de desarrollo de esta en todos sus niveles.

El análisis de los beneficios, por tanto, se hará de forma metódica en base a **tres perspectivas** principales: los beneficios técnicos, económicos y sociales. Todas ellas suponen factores esenciales en el ámbito de la calidad de servicio.

4.1. Beneficios Técnicos

Los beneficios técnicos del estudio de técnicas de aprendizaje automático para la mejora de la calidad de servicio en redes WLAN son múltiples. Como **beneficio principal** se encuentra la obtención de las herramientas de aprendizaje automático necesarias para la correcta realización de la diferenciación de entornos de red y la clasificación de perfiles de usuario.

De este modo, se cubren los aspectos fundamentales para la **extracción del contexto de usuario y la red**—mediante ambos procedimientos de mejora de la QoS— de forma innovadora, que hace posible el conocimiento del estado de la situación y el ajuste de la tecnología al mismo.

Cada uno de los procedimientos constituye, a su vez, un beneficio técnico al realizarse de forma automatizada, ya que permite la gestión de dichos aspectos con nuevas metodologías de forma **eficaz y eficiente**, favoreciendo un **uso responsable de los recursos** físicos en la provisión de servicios y minimizando su impacto.

Asimismo, la introducción de estas técnicas de gestión de la QoS persigue la **obtención de los KPIs** que se ajusten a distintos entornos y perfiles, permitiendo prestar especial atención a la gestión de los indicadores más significativos de las situaciones que se pueden contemplar en el caso de uso de las redes inalámbricas de área local.

Además, el análisis exhaustivo de los casos de uso de la tecnología IEEE 802.11 provee una guía para la implementación de simulaciones de red realistas sobre herramientas de la mayor actualidad, por lo que también se incluyen **beneficios educativos** y favorecen la ampliación del conocimiento sobre redes inalámbricas en nuevos estudios. En este sentido, se producen también beneficios con la configuración del simulador de red, ya que se realiza de forma que sirva de conductor de **nuevos experimentos** y de manual para futuras actualizaciones del estándar.

En definitiva, el estudio aporta herramientas a múltiples niveles para el establecimiento de las bases del diseño de una **gestión automática de la QoS**.

4.2. Beneficios Económicos

El **aspecto clave** para la obtención de beneficios económicos en el presente trabajo es la **automatización**. Las técnicas de aprendizaje automático permiten ajustes de

recursos –tanto humanos como tecnológicos– más precisos, lo que conlleva a una disminución en los costes.

El **beneficio económico principal** es la **reducción en costes de recursos humanos** debida a los procedimientos de gestión que hacen uso de aprendizaje automático. La diferenciación de entornos y la clasificación de perfiles permiten obtener el contexto de cada red sin implicar un estudio individual de escenarios y usuarios a través de la iteración en la toma de medidas y el análisis de los parámetros obtenidos en las mismas. Dicha compleja y laboriosa tarea supondría una alta inversión de recursos que, gracias a la automatización, se vuelve eludible.

De igual forma, se obtiene una **reducción en los gastos** relativos a recursos **tecnológicos** debido a la introducción de los procedimientos de gestión de QoS. El mejor ajuste de las características y recursos de red resultante de la contextualización de esta permite un aprovechamiento mayor para cada caso.

Del mismo modo, la detección del entorno en base al análisis automatizado facilita el estudio y monitorización de las casuísticas propias de escenarios concretos, lo que puede dar lugar a la menores tiempos requeridos en la resolución de posibles problemas de red, y que conlleva de forma indirecta **menores tiempos de caída de servicios** y recursos, al disponer de mayor información y facilitar las tareas de operaciones de red, beneficiando tanto a los proveedores de servicios como a los de infraestructura.

4.3. Beneficios Sociales

Los **beneficios sociales principales** están relacionados con la satisfacción de los usuarios de las redes WLAN. El desarrollo de nuevos procedimientos de gestión de la QoS conlleva a la mejora de la calidad en múltiples aspectos, lo que repercute positivamente en los usuarios de los distintos servicios.

El **ajuste de las características** derivado de la diferenciación de entornos, clasificación de perfiles y extracción de KPIs permite una provisión de servicios adaptada a los usuarios en cuestión. Dicha mejora promueve también un mayor grado de **satisfacción** y una mejor **experiencia** por parte del consumidor.

Más aún, los avances en materia de calidad de servicio impulsan la utilización y expansión de las redes inalámbricas de área local, incentivando así la **fidelidad de los usuarios** a las mismas y el desarrollo tecnológico.

5. Estado del Arte

El presente apartado engloba la base teórica sobre la que se fundamenta el estudio, fraccionando la complejidad de esta en tres grandes bloques.

Para comenzar, se realiza una introducción a la **Calidad de Servicio (QoS)**, prestando especial atención a la especificación de métricas para las medidas de calidad en la red. Así, se establecen las referencias principales sobre las que centrar el foco en el posterior análisis de la tecnología bajo estudio. Además, se realiza la **Introducción a los Procedimientos de Mejora de la QoS** que entrelazan los ámbitos de calidad, aprendizaje automático y redes inalámbricas.

A continuación, se presenta el **Resumen del Estudio de Escenarios de Aplicación del Estándar IEEE 802.11**, donde se analiza en profundidad la realidad de los entornos de red inalámbricos en base a la evolución del estándar.

Por último, se estudian los conceptos principales de las **Técnicas de Aprendizaje Automático**, centrando la atención en las técnicas de clustering en base a las conclusiones obtenidas en el Trabajo Fin de Grado [17].

5.1. Calidad de Servicio (QoS)

El concepto de **calidad de servicio** (Quality of Service, QoS) ha ido evolucionando de forma paralela al desarrollo de las redes de comunicaciones y de los servicios ofrecidos en ellas. Con motivo de estandarizar la terminología relativa a la QoS la ITU-T establece la especificación E.800. [10] En concreto, se define la **QoS** como *“La totalidad de las características de un servicio de telecomunicaciones que determinan su capacidad para satisfacer las necesidades explícitas e implícitas del usuario del servicio.”*

Esta definición trata de incluir el concepto de calidad de servicio de forma global, lo que conlleva al análisis de la calidad de extremo a extremo. De forma resumida, en la Figura 7 se recogen los elementos principales que realizan contribuciones a la QoS, disponiendo cada uno de los componentes de sus particularidades a la hora de realizar dichas contribuciones.

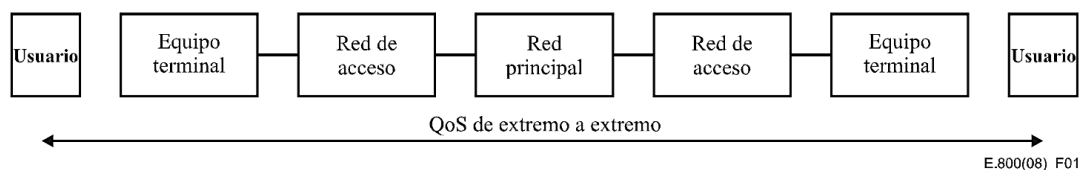


Figura 7: Esquema de contribuciones a la QoS de extremo a extremo.

Se entiende la **QoS de extremo a extremo** como la suma de todas las contribuciones de los elementos del sistema de comunicaciones entre los **usuarios** finales, o desde un proveedor de servicios al usuario final, en caso de ser estos los extremos de la comunicación. Además, como se adelantaba en la Introducción, el auge de nuevos servicios ha incorporado también nuevos condicionantes en el ámbito de las comunicaciones, y entre ellos, aparecen nuevas casuísticas en las que los usuarios finales pueden tratarse de máquinas, y no de personas.

Los **equipos terminales**, por otro lado, realizan su aportación en base a la calidad de funcionamiento del propio dispositivo. Se han de tener en cuenta características relativas al tipo de dispositivo, como las capacidades de procesamiento de este, y los protocolos de red utilizados.

Asimismo, la **red de acceso** realiza su contribución en función de la combinación de la tecnología y el medio de acceso empleados para cada servicio. En el ámbito de este estudio, se establece el medio inalámbrico como medio de acceso de referencia y se considera como tecnología la asociada al protocolo IEEE 802.11.

Por último, la **red principal** se compone por las redes de uno o más proveedores cuya aportación dependerá en gran medida de cada componente, las tecnologías empleadas y los medios de transmisión.

Sin embargo, la QoS también se ve afectada por las **condiciones operativas** específicas, como el entorno, el tráfico y el encaminamiento. La relación entre la QoS y el funcionamiento de la red (Network Performance, NP) se muestra en la Figura 8. De igual manera, se ha de poner en valor la dependencia de la QoS a cada servicio prestado y los **requerimientos específicos** en cada caso.

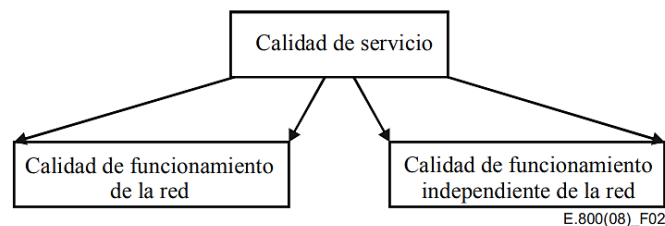


Figura 8: Relación entre QoS y NP.

Dentro de la calidad de **funcionamiento de la red** se contemplan parámetros como el throughput, la tasa de errores en los bits, la latencia y la variación de retardo, entre otros. De esta forma, afecta de directamente a las tareas de planificación, desarrollo, operaciones y mantenimiento del proveedor de red, y comprende la parte técnica de la QoS ofrecida. En la calidad de **funcionamiento independiente de la red**, en cambio, se consideran factores como el tiempo de prestación del servicio, el tiempo de reparación, la tarificación o el tiempo de resolución de quejas.

Del mismo modo, la propia ITU-T en la recomendación E.800 [10] hace la distinción entre los **parámetros objetivos** o cuantitativos, que hacen alusión a las medidas y valores cuantificables, y los **parámetros subjetivos** o cualitativos, que se refieren a juicios basados en notas de opinión.

Además, en la guía de regulación de la QoS desarrollada por la ITU-D [32] se ahonda en las influencias de diferentes factores a la percepción del usuario final sobre la calidad de servicio, haciendo también mención de aspectos subjetivos como el efecto de las tendencias sociales –popularidad de dispositivos, servicios, aplicaciones, redes sociales, etc.– o la publicidad en las expectativas de los consumidores de servicios. De esta forma,

se remarca que la percepción del usuario no depende únicamente de parámetros objetivos, sino también de la **experiencia personal** en la prestación del servicio.

En consecuencia, se introduce el término de **calidad de la experiencia (Quality of Experience, QoE)**, definido por la ITU-T como “*el grado de satisfacción o molestia del usuario de una aplicación o servicio*”. [12]

La QoE también incluye todos los efectos del sistema de comunicaciones de extremo a extremo, y toma en cuenta de forma adicional parámetros en base a las **expectativas del usuario**, el **contexto** de este, y la discrepancia entre la oferta del servicio y el conocimiento por parte del usuario del servicio y sus características.

Bajo esta premisa, la definición de un marco común que recoja los criterios a seguir en la medida de la QoS resulta ineludible. Por lo tanto, se procede a analizar la información de los criterios de QoS definida por las recomendaciones de la ITU.

5.1.1. Criterios de QoS

Como se mencionaba anteriormente, la evaluación de la QoS requiere el establecimiento de criterios y parámetros medibles. En concreto, la ITU-T recoge los términos principales para proveer un marco común de QoS en la recomendación G.1000. [33]

Se especifican siete **criterios** o grupos de características de calidad del servicio: velocidad, precisión, disponibilidad, fiabilidad, seguridad, simplicidad y flexibilidad.

		Criterios de calidad de servicio						
		Velocidad 1	Precisión 2	Disponibilidad 3	Fiabilidad 4	Seguridad 5	Simplicidad 6	Flexibilidad 7
Función de servicio								
GESTIÓN DE SERVICIO	Ventas y actividades precontractuales 1							
	Prestación 2							
	Alteración 3							
	Atención al cliente 4							
	Reparaciones 5							
	Cese 6							
CALIDAD DE LA CONEXIÓN	Establecimiento de conexión 7							
	Transferencia de información 8							
	Liberación de conexión 9							
Facturación 10								
Gestión de la red/ servicio por el cliente 11								

Figura 9: Matriz para la identificación de los criterios de QoS para las comunicaciones.

En primer lugar, la **velocidad** [10] describe el intervalo de tiempo utilizado para la realización de una función, o la velocidad a la que se ocurre la misma, tenga lugar o no con la precisión deseada. Hace referencia a todas las funciones de los servicios.

Por otro lado, la **precisión** [10] conforma la medida del grado de corrección con el que se lleva a cabo una función, pudiéndose realizar o no a la velocidad deseada. Ejemplos de medida de la precisión [32] son la calidad del habla, la tasa de éxito en llamadas y la exactitud en la facturación.

En tercer lugar, la **disponibilidad** [10] describe la situación en la que un elemento se halla en un estado en el que es capaz de ejecutar una función requerida en un instante determinado o un intervalo de tiempo dado. Por ejemplo, la disponibilidad puede ser representada por la cobertura o la disponibilidad de un servicio concreto. [32]

La **fiabilidad** [10], por su parte, es la probabilidad de que una entidad realice la función requerida cumpliendo las condiciones establecidas en un tiempo dado. Ejemplos representativos de fiabilidad [32] pueden ser la tasa de llamadas descartadas o el número de quejas relativo a facturación.

La **seguridad** [34] hace referencia a la combinación de factores que describen la integridad del sistema y sus usuarios. Incluye tanto características de autenticación como autorización, y seguridad tanto en los datos como en el acceso al sistema. Un ejemplo de ello [32] pueden ser las medidas sobre prevención de fraude.

En sexto lugar, la **simplicidad** [10] es la facilidad o ausencia de complejidad de una función del servicio, para el beneficio del usuario. Casos de aplicación de esta medida son [32] la facilidad en la actualización de software o la facilidad para la terminación de un contrato.

Por último, la **flexibilidad** puede entenderse como la tolerancia a cambios en la realización de una función del servicio. Por ejemplo, [32] la flexibilidad puede representarse por la facilidad en el cambio de contrato o la disponibilidad de diferentes métodos de pago como los pagos online, etc.

Por lo general, los siete criterios de la QoS **también aplican a la QoE**, ya que ambas están interrelacionadas. Un ejemplo de ello puede ser la velocidad, que afecta al throughput disponible y la latencia, lo que resulta de vital importancia en la percepción de un usuario de la calidad y por tanto en la QoE.

En base a los criterios definidos, se obtiene la matriz de determinación de criterios de calidad de servicio en las comunicaciones (Figura 9), que supone el paso previo a la definición de los parámetros de QoS.

De esta forma, atendiendo a cada tipo de servicio se obtienen los parámetros que permiten una solución adaptada a la medida de la calidad en cada caso.

5.1.2. Parámetros y requisitos mínimos de las clases de QoS

Los **parámetros de QoS** reciben múltiples denominaciones a lo largo de la literatura, como pueden ser métricas, indicadores, medidas o determinantes QoS, pero en todos los casos tratan de caracterizar el nivel de calidad en el que se ofrece un servicio y el nivel de satisfacción del usuario.

Como se mencionaba al comienzo del apartado Calidad de Servicio (QoS), los parámetros pueden ser obtenidos de forma objetiva o subjetiva, pudiendo estar relacionados entre sí. Los parámetros objetivos se obtienen de medidas de atributos físicos, como los circuitos, redes y señales. En cambio, los parámetros subjetivos se obtienen mediante la realización de encuestas de opinión realizadas al consumidor.

Además, las métricas pueden definirse a través de medidas directas—como pueden ser el ruido de un circuito, las pérdidas de propagación, etc.— o mediante el tratamiento matemático de parámetros primarios, como es el caso de estadísticas, ratios para la estimación de la opinión de consumidores o umbrales de decisión.

La definición de parámetros puede hacerse desde múltiples puntos de vista, realizando clasificaciones que tomen como referencia el tipo de señal, el servicio consumido, o la clase de servicio—gratuita, premium, etc.—, entre otros.

El ámbito de este estudio se ubica en el análisis de la calidad sobre servicios IP, por lo que a continuación se definen los **parámetros a nivel IP** a ser utilizados de acuerdo con la recomendación de la ITU Y.1540, [35] contando con un total de cuatro parámetros de funcionamiento principales.

Primero, el **retardo en la transferencia de paquetes IP** (IP packet transfer delay, IPTD), mide la diferencia entre los tiempos de ocurrencia de dos eventos de referencia consecutivos de paquetes IP (Ecuación 1), es decir, la diferencia de tiempos entre dos transmisiones de paquetes entre puntos de medida en la red predefinidos. Esto aplica tanto a los resultados de paquetes satisfactorios como con errores.

$$IPTD(s) = t_{e1}(s) - t_{e2}(s)$$

Ecuación 1: Cálculo del retardo de transferencia de paquetes IP.

Asimismo, existen diferentes tipos de IPTD derivados del parámetro original. El IPTD mínimo ($IPTD_{\min}$) es el mínimo retardo de paquetes IP entre todos los retardos de paquetes IP (IPTD). Por otro lado, el IPTD mediano es el percentil 50 de la distribución de frecuencias de los retardos de transferencia de paquetes IP. También se define el IPTD medio, que se obtiene mediante la media aritmética de los retardos de transferencia de paquetes IP.

En segundo lugar, además de los parámetros derivados directamente del IPTD, existe otra métrica de uso común que utiliza información extraída de la primera. Se trata de la **variación de retardo de paquetes IP** (IP delay variation, IPDV), también conocida como jitter. La variación de retardo comprende la diferencia entre una instancia del

retardo de un paquete IP y el retardo de transferencia de paquetes IP de referencia (Ecuación 2), que puede ser, por ejemplo, el IPTD medio.

$$IPDV (s) = IPTD_1(s) - IPTD_2(s)$$

Ecuación 2: Cálculo de la variación de retardo de paquetes IP.

En tercer lugar, la **tasa de pérdida de paquetes IP** (IP Packet Loss Ratio, IPLR) es el ratio del total de paquetes IP perdidos respecto al total de paquetes IP transmitidos en una medida dada (Ecuación 3). Dentro del total de paquetes IP transmitidos se incluyen tanto los paquetes satisfactorios como los paquetes con error.

$$IPLR (\%) = \frac{\text{número de paquetes perdidos}}{\text{número de paquetes totales}} \times 100$$

Ecuación 3: Cálculo de la tasa de pérdida de paquetes IP.

Por último, la **tasa de error en paquetes IP** (IP Paquet Error Ratio, IPER) se define como el ratio del total de paquetes IP con error respecto al total de paquetes IP transmitidos en una medida dada (Ecuación 4). Al igual que en el caso anterior, dentro del total de paquetes IP transmitidos se incluyen tanto los paquetes satisfactorios como los paquetes con error.

$$IPER (\%) = \frac{\text{número de paquetes con error}}{\text{número de paquetes totales}} \times 100$$

Ecuación 4: Cálculo de la tasa de error en paquetes IP.

Entre los parámetros anteriores, se obvia el parámetro asociado a la capacidad ya que se sobreentiende que atienden a contratos especificados entre el usuario y proveedor de forma previa a la provisión de servicios.

Este parámetro se denomina **ancho de banda** (bandwidth, BW), y define la capacidad o número de bits máximos de transmisión utilizables o disponibles (Ecuación 5). Es un parámetro decisivo para la percepción de funcionamiento del usuario final, siendo mejores, por lo general, mayores tasas de bits.

$$BW (bps) = \frac{\text{volumen de datos a transferir (bit)}}{\text{tiempo especificado (s)}}$$

Ecuación 5: Cálculo del ancho de banda. [36]

Otro parámetro fundamental es el **throughput** (Ecuación 6), la medida de la cantidad de datos transferidos en un tiempo específico. Puede definirse para un enlace o para un sistema completo. De forma habitual, se mide en bits por segundo o paquetes por segundo, aunque en ambos casos se pueden aplicar transformaciones en caso de conocer el tamaño del paquete.

$$\text{Throughput (bps)} = \frac{\text{número de paquetes enviados (bit)}}{\text{tiempo de entrega (s)}}$$

Ecuación 6: Cálculo del throughput.

Además de la especificación de parámetros definida por la ITU-T para paquetes IP, también resulta habitual la medida de calidad en base a **parámetros a nivel de servicio**.

Entre estos destaca la medida de la **disponibilidad** como la viabilidad del servicio en cada solicitud realizada, y cuyo cálculo se expresa en la Ecuación 7.

$$Disponibilidad (\%) = 100 - \frac{t_{no\ disponibilidad} + t_{desconexión}}{t_{total}} \times 100$$

Ecuación 7: Cálculo de la disponibilidad de servicio. [36]

Otros parámetros típicos a nivel de servicio son el **tiempo medio entre fallos** (Mean Time Between Failure, MTBF), [36] que se obtiene como promedio del tiempo entre fallos que ocurren durante la provisión de un servicio, y el **tiempo medio de recuperación del servicio** (Mean Time to Restore Service, MTRS), [37] que es el promedio del tiempo necesario desde que un servicio se encuentra en estado de no disponibilidad hasta que es restablecido completamente y vuelve a funcionar en su estado habitual.

En base a los parámetros de calidad definidos en el apartado anterior, la ITU-T establece en su recomendación Y.1541 [38] los **requisitos mínimos por clase de QoS**. El alcance definido para los valores de referencia de la Tabla 4 es entre las interfaces de red de los equipos finales, por lo que incluye todo el sistema de comunicaciones intermedio.

De esta forma, la categorización en clases permite definir los límites superiores de los parámetros de funcionamiento clave para servicios IP extremo a extremo, cubriendo los requerimientos de aplicaciones como la telefonía, aplicaciones basadas en TCP, o televisión digital.

Parámetro de Funcionamiento de Red	Naturaleza del Objetivo de Funcionamiento de Red	Clases de QoS					
		Clase 0	Clase 1	Clase 2	Clase 3	Clase 4	Clase 5 (Sin Especificar)
IPTD	Límite superior del IPTD medio.	100 ms	400 ms	100 ms	400 ms	1 s	-
IPDV	Límite superior del cuantil 1-10 ⁻³ de IPTD menos el IPTD mínimo.	50 ms	50 ms	-	-	-	-
IPLR	Límite superior de la probabilidad de pérdida de paquetes.	1 x 10 ⁻³	1 x 10 ⁻³	1 x 10 ⁻³	1 x 10 ⁻³	1 x 10 ⁻³	-
IPER	Límite superior.	1 x 10 ⁻⁴					-

Tabla 4: Definición de clases de QoS en redes IP y objetivos de funcionamiento de red.

Las características asociadas a cada clase de QoS presente en la Tabla 4 se recogen en la Tabla 5, donde se especifican ejemplos de aplicación, los mecanismos utilizados en los nodos para gestionar dichas clases, y las técnicas de red asociadas.

Resulta conveniente aclarar que las clases de QoS definidas en la recomendación Y.1541 no hacen referencia a las clases de tráfico mencionadas en el apartado Estándar IEEE

802.11 y QoS: IEEE 802.11e, ya que estas últimas son específicas de los métodos de gestión de QoS implementados en los protocolos de redes inalámbricas.

Clase de QoS	Ejemplos de Aplicación	Mecanismos de Nodo	Técnicas de Red
0	Tiempo real, sensible a jitter, alta interacción (VoIP, VTC)	Colas separadas con servicio preferencial, traffic grooming	Rutado y distancias más limitadas
1	Tiempo real, sensible al jitter, interactivo (VoIP, VTC)		Rutado y distancias menos limitadas
2	Datos de transacción, alta interacción (señalización)	Colas separadas, prioridad de descarte	Rutado y distancias más limitadas
3	Datos de transacción. Interactivo		Rutado y distancias menos limitadas
4	Sólo bajas pérdidas (transacciones cortas, grandes volúmenes de datos, streaming de vídeo)	Colas largas, prioridad de descarte	Cualquier ruta o camino
5	Aplicaciones tradicionales de las redes IP por defecto	Colas separadas, mínima prioridad	Cualquier ruta o camino

Tabla 5: Guía para las clases de QoS IP.

Dentro del amplio abanico de parámetros de QoS habitualmente se hace referencia a los **indicadores clave de rendimiento** (Key Performance Indicator, KPI) [32] como los parámetros clave que resultan decisivos para la medida de la calidad, y pueden ser tanto técnicos como no técnicos.

Los KPIs están basados en mediciones de red y resultan esenciales para las tareas de operaciones, mantenimiento y modelos de negocio, además de generar aportaciones a las tareas de reporte y auditoría.

Como se menciona en Calidad de Servicio (QoS), la QoS es experimentada, requerida y esperada por los usuarios finales. El nivel planificado por cada proveedor de servicio puede no coincidir con el nivel de QoS entregado, y a pesar de los esfuerzos de estos por obtener altos niveles, la experiencia de usuarios finales puede resultar insuficiente, e incluso muy inferior a lo inicialmente planteado.

En la **definición de KPIs**, se recomienda evitar la especificación de estos con objetivos implícitos. [32] Un ejemplo de ello puede ser la definición del porcentaje de fallos en un sistema reparados en un día, donde el objetivo sería ese día. Una expresión más apropiada sería la medida del tiempo en el que el 80 por ciento de fallos han sido solucionados con mayor velocidad, ya que se elimina la dependencia al día concreto.

Por otra parte, el **nivel de funcionamiento alcanzable** está estrechamente relacionado al **tipo de acceso a la red**, lo que afecta también a la definición de KPIs. Puede resultar necesario definir objetivos distintos para accesos móviles y fijos, así como para inalámbricos y cableados. Además, los requerimientos por parte del usuario también varían a lo largo del tiempo, del mismo modo que la propia tecnología.

La distinción en la forma de acceso puede influir tanto sobre la propia naturaleza del servicio – un ejemplo de ello puede ser la telefonía – como sobre los valores esperados de los KPIs y los resultados obtenidos en las medidas. Sin embargo, esta diferenciación se va reduciendo junto con la convergencia de redes y la migración de servicios al modelo de paquetes IP. [32]

A continuación, se procede a describir los KPIs más habituales para diferentes tipos de aplicaciones realizando una distinción inicial que separa los KPIs técnicos de los que no lo son.

5.1.2.1. Clasificación de Indicadores Clave de Rendimiento

Existen múltiples criterios para la clasificación de **KPIs técnicos**, pero de acuerdo con los objetivos del presente estudio, se realiza una distinción en base a la naturaleza del servicio, y también se hace alusión a KPIs independientes del servicio, como se puede apreciar en la Tabla 6.

Dentro de los servicios de datos principales, el manual de regulación de QoS [32] contempla el protocolo FTP, el correo electrónico a través de protocolos estandarizados y los servicios web sobre HTTP. Además, el protocolo HTTP también puede ser utilizado en la provisión de ciertos servicios donde la gestión de QoS resulta de gran importancia, aunque no se consideren como servicio de datos tradicionales. [32] Ejemplo de ello son servicios propietarios como Skype, Netflix o Twitter, y servicios públicos como IPTV.

KPIs técnicos		Independientes del servicio Transversales	KPIs no técnicos
Dependientes del servicio			Satisfacción del cliente, información de facturación, claridad de los planes y tarifas, exactitud en la tarificación, facilidades de pago, calidad y provisión del servicio, tiempos de activación / desactivación / restauración del servicio, disponibilidad de red, tiempo medio de servicio caído.
Genéricos	Específicos del Servicio		
Ancho de banda, retardo de propagación, retardo de encolado, pérdidas de paquetes.	Accesibilidad del servicio, tiempo de configuración del servicio, tasa de datos del servicio, tasa de transferencias incompletas, tiempo de sesión.	Accesibilidad de red, tasa de fallo y tiempo de conexión a la red, tasa de fallo y tiempo de resolución DNS, tasa de fallo y tiempo en acceso a llamadas de datos.	

Tabla 6: Clasificación de KPIs con ejemplos.

De acuerdo con las recomendaciones de la ITU-T G.1011 [39] y E.800 [10], los parámetros principales de calidad para los servicios de datos habituales se recogen en la Tabla 7. En ella se indica la relevancia de cada parámetro con símbolos positivos y negativos, desde el menos relevante (-) al más relevante (+++).

En base a la Tabla 7, por tanto, se pueden extraer como **KPIs** [32] **genéricos** aplicables a todos los servicios el ancho de banda, el retardo de propagación, el retardo de encolado –el tiempo que un paquete debe esperar antes de ser transmitido en cada nodo– y las pérdidas de paquetes.

Estos KPIs se aproximan a los parámetros definidos por ETSI y recomendados por BEREC, [40] que son la velocidad de subida y bajada, el retardo, la variación de retardo, la tasa de pérdida de paquetes y la tasa de errores en paquetes.

Aplicación	Velocidad de transmisión		Retardo	Jitter	Pérdidas de paquete
	DL	UL			
Búsqueda Web (texto)	++	-	++	-	+++
Búsqueda Web (media)	+++	-	++	+	+++
Descarga de ficheros	+++	-	+	-	+++
Transacciones	-	-	++	-	+++
Streaming	+++	-	+	-	+
VoIP	+	+	+++	+++	+
Gaming	+	+	+++	++	+++

Tabla 7: Servicios de datos y parámetros de calidad.

Sin embargo, cada aplicación concreta puede disponer de **KPIs específicos del servicio** y de la definición de sus funcionalidades. En concreto, la ITU-T incluye en su recomendación E.804 varios ejemplos de métricas específicas, [41] como pueden ser, en el caso del servicio FTP, el tiempo de configuración del servicio FTP, la tasa de transferencias incompletas FTP o el tiempo de sesión FTP.

Por otro lado, en la recomendación E.804 también se recogen [41] una serie de **KPIs transversales**, que son independientes del servicio, y hacen énfasis en procedimientos propios de la red. Entre ellos, se contemplan aspectos como la disponibilidad y accesibilidad de red, los tiempos y errores en resolución DNS o las tasas de fallo en conexión o asociación a la red.

Además de los anteriores, con la introducción de la QoE se incide cada vez más en parámetros subjetivos, lo que da lugar a la definición de **KPIs no técnicos**. [41] Este tipo de métricas se recoge en la recomendación de la ITU-T E.803 [42], y se centra en cómo se provee el servicio a nivel usuario, teniendo en cuenta la calidad e información conocida del mismo, y el desempeño del servicio. Algunos de los ejemplos principales se muestran en la Tabla 6.

Gran parte de las medidas de los KPIs no técnicos son difícilmente cuantificables en términos de unidades estándar, por lo que se siguen las aproximaciones utilizadas en la medida de la QoE en base a la **calificación de opiniones** (Opinion Rating, OR).

MOS	Calidad
1	Mala
2	Pobre
3	Media
4	Buena
5	Excelente

Tabla 8: Calificación media de opinión.

Una medida de la calidad ampliamente extendida, que inicialmente se definió para los servicios de voz [10] es la denominada calificación media de opinión (Mean Opinion

Score, MOS). Dicho procedimiento establece un valor entre 1 y 5 (Tabla 8) para la valoración de la opinión, desde la peor a la mayor calidad experimentada, respectivamente.

5.1.3. Introducción a los Procedimientos de Mejora de la QoS

Hasta ahora se han presentado los criterios y parámetros principales en la evaluación de la QoS. Tomando dicha base como referencia, a continuación, se introducen las técnicas para la **mejora de la gestión de la QoS** de las que consta el estudio: la diferenciación de entornos y la clasificación de perfiles.

Dentro del ámbito de la QoS existen infinidad de procedimientos que pueden ser automatizados y mejorados. [43] En este caso, se centra la atención en la gestión de información relevante para la QoS en términos de contexto de uso de la tecnología – localización, movimiento, temporalidad, uso, etc.– y de las métricas –throughput, retardo, pérdidas, seguridad, etc.– presentadas en el apartado anterior.

Se entiende por contexto la descripción de entidades como usuarios, dispositivos, situaciones o los propios entornos, y pueden clasificarse en base a múltiples criterios. En la Tabla 9 se recogen algunos ejemplos. [44]

Contexto	Factores
Físico	Tiempo, localización, disposición física
Usuario	Identidad, preferencias, edad y género, emociones
Dispositivo	Batería, CPU y RAM, tamaño de pantalla
Actividad	Entretenimiento, compras, trabajo

Tabla 9: Ejemplos de factores por tipo de contexto.

La **diferenciación de entornos** es el procedimiento de análisis mediante el que se obtiene una distinción de escenarios de red a través de la extracción de datos de monitorización y su posterior procesado estadístico.

El objetivo final de la diferenciación de entornos es la **extracción del contexto** [43] **de red** para permitir un **ajuste adaptado de la QoS** a los requerimientos de cada escenario, de forma que se facilite una gestión automatizada que favorezca los requisitos de QoS específicos.

Para poder realizar la diferenciación se deben tomar entornos de red significativos que cubran casos de uso de una tecnología con características distintivas. En la práctica, dicho procedimiento se basa en la selección de escenarios y la monitorización mediante analizadores de tráfico.

Sin embargo, en el ámbito del presente estudio se adopta el proceso contrario. Se trata de proveer herramientas de mejora de la QoS partiendo de simulaciones de red, por lo que inicialmente se debe realizar una parametrización exhaustiva de los escenarios, teniendo en cuenta tanto factores propios de la tecnología como del entorno y sus usuarios. Dicha tarea se halla descrita en el apartado Resumen del Estudio de Escenarios de Aplicación del Estándar IEEE 802.11.

El proceso de desarrollo de la diferenciación de entornos, por tanto, se hace a través de ingeniería inversa a lo que resultaría en escenarios reales. En lugar de clasificar escenarios mediante la toma de medidas, se definen inicialmente sus parámetros característicos para su simulación y procesado posterior.

Bajo esta premisa aparecen las tareas de parametrización de entornos y el modelado del tráfico de red que, a su vez, abren la puerta a la extensión del estudio a diferentes tipos de usuario en base a patrones de movilidad, clases de tráfico utilizadas y patrones de uso sobre las aplicaciones.

De forma paralela a la diferenciación de entornos, la **clasificación de perfiles** trata del procedimiento de diferenciación de perfiles de usuario mediante el análisis de datos obtenidos mediante monitorización, y se puede hacer en base a distintas perspectivas.

Un criterio común es la clasificación de perfiles de usuario en función de los **hábitos de consumo** de aplicaciones, que suscita gran interés en la correcta caracterización de las redes inalámbricas.

Existen varias propuestas sencillas para la clasificación de usuarios en función de varios parámetros. Por ejemplo, en una aproximación del comportamiento de usuarios de Internet en redes municipales con acceso FTTH [45] se establecieron un total de tres perfiles mediante la utilización de clustering para la separación de los tipos de perfiles en función de la cantidad de datos consumidos y el número de aplicaciones utilizadas.

Ejemplos más complejos pueden atender a los diferentes contextos presentados en la Tabla 9, como puede ser el caso del estudio multidimensional de la información contextual extraída mediante la combinación de perfiles de usuario y situaciones concretas [44], donde se definen perfiles de usuarios en base a factores estáticos –datos personales, conocimiento, características, capacidades, estado físico y cognitivo– y dinámicos –intereses, preferencias, actividades, localización– o dependientes del comportamiento. Además, se introduce la casuística en la que cada usuario puede ser asociado a múltiples perfiles.

Realizar clasificaciones de los usuarios en función de la utilización de la red resulta de gran utilidad a distintos niveles. Por un lado, los proveedores de servicios pueden conocer los hábitos de consumo, en qué medida los usuarios consumen ancho de banda, y otros aspectos para **ajustar las características de la red**.

De forma similar a la diferenciación de entornos, la naturaleza del estudio conlleva a adoptar el proceso contrario al que se aplicaría en la realidad, y, en este caso, el análisis se fundamenta en el ajuste óptimo de los algoritmos a los escenarios y su similitud a los perfiles de tráfico definidos en el Resumen del Estudio de Escenarios de Aplicación del Estándar IEEE 802.11.

Una vez introducidos los aspectos principales para tener en cuenta en el ámbito de la QoS, y tomando dicho conocimiento como criterio en el análisis de la documentación sobre la aplicación de la tecnología IEEE 802.11, se presenta el estudio que fundamenta las simulaciones WLAN.

5.2. Resumen del Estudio de Escenarios de Aplicación del Estándar IEEE 802.11

Como se adelanta en la Introducción, la situación de las redes actuales se puede describir como compleja y heterogénea. En consecuencia, poder obtener una base sólida para el desarrollo de simulaciones de redes que utilizan la tecnología IEEE 802.11 requiere la realización de un análisis documental afín a las características de dichas redes.

Por lo tanto, a continuación, se presenta de forma resumida la información recogida y analizada sobre el estado del arte de la tecnología WLAN. De este modo, se reserva el Anexo I: Estudio Detallado de los Escenarios para cubrir la descripción completa y justificación de los datos extraídos de las múltiples fuentes, además de los modelos estadísticos representativos de los tipos de tráfico.

5.2.1. Introducción a la Caracterización de Entornos de Redes Inalámbricas

Con el objetivo de establecer los parámetros del simulador de red de la forma más apropiada a los diferentes entornos disponibles en la red, se debe realizar una extensa revisión documental.

Para ello, se analizan y extraen las conclusiones de múltiples investigaciones e informes de fuentes internacionalmente reconocidas, contemplando los patrones descriptivos a distintos niveles, entre los que se incluyen la disposición y movilidad de equipos –tanto APs como STAs– de acuerdo con cada escenario estudiado, las configuraciones de red más utilizadas, el tipo de tráfico en base a diferentes casos de uso, y el rendimiento general observado por las muestras tomadas sobre casos reales.

La caracterización de escenarios resulta una tarea complicada. El tráfico en Internet es en gran medida dinámico y a veces incluye cambios rápidos. [4] Estas variaciones pueden ser provocadas por motivos sociales, culturales y tecnológicos entre otros. Situaciones como los cambios en los hábitos de consumo de servicios, la irrupción de nuevos servicios o la regulación de uso de la red pueden modificar drásticamente las características de uso de esta, y, por lo tanto, también las de los escenarios de aplicación WLAN.

Del mismo modo, la heterogeneidad presente en las redes hace inevitable la convivencia de no solo distintos tipos de usuarios y dispositivos, sino también de las versiones de los estándares de la tecnología.

En consecuencia, para obtener una vista completa no se ha de limitar el estudio únicamente al análisis de entornos de interés sobre una versión del protocolo IEEE 802.11, sino que ha de contemplar tanto escenarios de última generación como modelos más tradicionales.

Sin embargo, el objetivo es generar **modelos reproducibles** que reflejen la realidad más actual posible, y tratar de cubrir todos los escenarios como casos particulares en base a escenarios concretos y en función de cada versión del estándar resultaría una

tarea excesivamente minuciosa para el ámbito de aplicación de este trabajo. Por lo tanto, en base a la selección de artículos estudiados, y con ánimo de enriquecer la propuesta realizada en el Trabajo Fin de Grado [17], se han considerado, desde un principio, un total de cuatro **entornos característicos** aplicables a los escenarios más comunes: **entorno residencial, empresarial, interior de alta densidad, y exterior de alta densidad.**

Aspecto Estudiado	Característica	Descripción
Definición espacial y topología	Plano	Características físicas del entorno
	Posicionamiento	Localización exacta de los AP y STA
	Movilidad	Modelo de movilidad de los AP y STA
Características físicas y configuración de la transmisión	Banda frecuencial	Banda de frecuencias utilizada
	Configuración de canales	Selección y uso de canales por dispositivo en función de la banda frecuencial
	Ancho de canal	Tamaño de canal utilizado
	MIMO	Disponibilidad de tecnología MIMO
	Número de antenas	Configuración de las antenas TX/RX utilizadas en los AP y STA
	OFDM	Disponibilidad de tecnología OFDM
	Potencia TX	Potencia de transmisión de los AP y STA en dBm
Características MAC	Modelo de propagación	Modelo de propagación utilizado para la caracterización del entorno en términos de pérdidas y retardo
	EDCA	Protocolo de acceso empleado y características
	Agregación	Características de agregación a distintos niveles (A-MPDU, A-MSDU)
	Retries	Número de repeticiones máximas
Modelo de tráfico	RTS/CTS	Uso de RTS/CTS y su umbral de aplicación
	Aplicación simulada	Selección de la aplicación utilizada para la generación de tráfico y los parámetros específicos para cada caso de uso
Interferencia	Tipo de tráfico	UL, DL, P2P, tráfico de gestión
	Interferencia	Su caracterización se puede realizar de forma explícita o implícita, a través de la propia definición física y espacial, el modelo de tráfico y los parámetros de red.

Tabla 10: Parámetros a tener en cuenta en la definición de escenarios. [46]

Para definirlos, resulta indispensable definir una plantilla que reúna los parámetros necesarios para la realización de las simulaciones. Los parámetros requeridos de forma habitual para la definición de dichos escenarios se extraen de las características resumidas en la Tabla 10.

El estudio de las características a diferentes niveles de la tecnología IEEE 802.11 y sus casos de uso en diferentes entornos, por tanto, cubre múltiples parámetros determinantes a la hora de diseñar simulaciones realistas, pero han de tomarse en consideración algunos **factores de influencia** de forma previa al desarrollo de la información recogida de las múltiples fuentes.

Por un lado, la **disponibilidad de** gran parte de las **características** a nivel de sistema depende directamente de la versión del estándar contemplada en cada caso. Esto afecta directamente a las configuraciones a nivel físico y de enlace de la tecnología empleada.

Además, el **interés por algunos escenarios** se ve influenciado por las características clave de cada versión del estándar. Un ejemplo de esto es el alto interés actual por los escenarios de alta densidad de usuarios, ya que la última versión de Wi-Fi – correspondiente al protocolo IEEE 802.11ax– hace hincapié en la eficiencia frente a casuísticas de este tipo.

Del mismo modo, se debe considerar que existe un **tiempo intermedio de implantación** de nuevas versiones de la tecnología WLAN desde que se presenta el estándar hasta que se despliega y se vuelve objeto de estudio sobre casos reales, lo que puede afectar a la disponibilidad de información sobre los distintos estándares.

Poniendo en valor dichos aspectos, los próximos subapartados describen en mayor detalle los parámetros seleccionados como referencia para la realización de las simulaciones.

En base a una primera lectura de la documentación que comprende este estudio, y tomando como referencia las propuestas realizadas por el grupo de desarrollo de la IEEE TGax (Task Group AX) [46] se han definido tres grandes grupos de características, que recogen la gran mayoría¹ de los aspectos mencionados en la Tabla 10. Con motivo de enriquecer los parámetros con casos reales y actuales, se analizan múltiples configuraciones utilizadas en diferentes estudios sobre redes inalámbricas para cada uno de los cuatro escenarios definidos.

En general, cabe mencionar que se aprecia simplicidad en la definición de los entornos en la documentación analizada, donde, en su mayoría, se establecen las configuraciones mínimas para caracterizar la red en base a la propiedad en la que se centra cada estudio o investigación. En consecuencia, se favorecen fuentes que proporcionen datos –y cifras objetivas– de casos de aplicación y estudio reales frente a propuestas de simulación.

5.2.1.1. Definición Espacial y Topología

En relación con la dimensión espacial y la topología en los casos de uso de la tecnología IEEE 802.11, se estudian las **dimensiones** de los entornos, las **características específicas** a tener en cuenta en cada caso –por ejemplo, elementos que degraden la señal, como puede ser la existencia de paredes en entornos de interior–, el **posicionamiento** tanto de los AP como las STA, y la **movilidad** de los mismos.

En el caso de los **entornos residenciales**, se recogen artículos donde se plantean despliegues de red con APs estáticos y STAs de distinto tipo sobre hogares –en su

¹ Debido a la extensión del estudio, no se modela la interferencia de forma explícita, sino que se caracteriza implícitamente en el modelado completo de la red, el tráfico, los usuarios, y el escenario.

mayoría apartamentos con varias habitaciones– donde se toma como referencia el análisis de muestras de tráfico obtenidas desde los APs instalados.

De este modo, en cuanto a las dimensiones de los escenarios, se describen entornos domésticos con áreas que oscilan entre los 80 m² [47] y los 100 m² [46], considerando alturas de los techos aproximadas de 3 m [46].

Por lo general, el posicionamiento de los APs y STAs varía en cada caso, estableciendo APs fijos –en su mayoría un AP por apartamento– en localizaciones específicas al apartamento y STAs que pueden ser tanto fijos como móviles –tomando como máximo 16 STAs [47] pertenecientes a cada residencia– debido a la heterogeneidad de dispositivos presente en los hogares, todos ellos a una altura estimada de 1,5 m [46].

En el **ámbito empresarial**, se analizan despliegues de red con AP y STA estáticos sobre entornos de oficina, en su mayoría centrados en la toma de muestras de señal para el estudio de sistemas de posicionamiento Wi-Fi que incluyen casos reales de aplicación de este tipo de escenarios.

En términos de las dimensiones de las oficinas, se recoge información de tipos de empresas muy distintas en función de su tamaño, lo que afecta en gran medida a la definición de las capacidades de las oficinas. De esta forma, se encuentran espacios que varían entre los 30 m² [48] y los 600 m² [49][50], estableciendo alturas máximas –al igual que en el entorno doméstico– aproximadas de 3 m [50] [46].

El posicionamiento de los APs y STAs es específico de cada oficina. Se instalan APs fijos –de forma que puede haber varios por oficina, uno único o ser compartidos por oficinas adyacentes, incluso ubicándose en el exterior de la habitación– y STAs que dependen de cada puesto de trabajo definido y el número de estos, y se supone que, en general, no hay movilidad [46]. Los APs se establecen a una altura de 3 m y las STAs a 1,5 m [50] [46].

Con relación a **los entornos de interior de alta densidad**, se realiza una búsqueda de información relativa a escenarios donde se establecen múltiples BSSs distribuidos de forma uniforme, como pueden ser, entre otros, aeropuertos, estaciones de tren, espacios comerciales de interior o despliegues para dar acogida a eventos de diferentes características. Se asume, por tanto, alta densidad tanto de APs como de STAs.

Entre los artículos estudiados, se recoge información en base a dimensiones muy variables dependientes del tipo de escenario, que en general parten de los 600 m² [51]. Para su análisis, se establecen celdas de tamaño medio –a partir de 150 m² [51]– con una altura de hasta 3m, donde se toma como eje central el AP, de forma que tan sólo se considera uno –aunque las bajas distancias ofrecen cobertura de distintos APs por localización– por celda –realizando la aproximación en base a la división del área de cobertura entre el número de APs estimado –, y se incluyen cantidades variables de STAs que pueden alcanzar un total de 50 [52]. Los APs se configuran con una altura de 3 m y se presupone que la altura de las STAs es 1,5 m [46].

Por último, respecto a los **entornos de exterior de alta densidad**, se estudian entornos de grandes dimensiones de exterior donde se aprecia gran densidad de STAs y una disposición de APs de tal forma que haya una distancia considerable entre ellos. Entre los ejemplos de casos reales de estos escenarios destacan los despliegues de red en ciudad y los campus universitarios abiertos.

Los espacios abiertos considerados parten de tamaños superiores a 1.000.000 m² [53], por lo que, al igual que los entornos de interior de alta densidad, requiere un análisis basado en celdas de tamaño inferior al total. Se consideran celdas a partir de 1400 m² [54] tomando como área la relación del espacio total entre el número de APs, y no se define una altura máxima debido a la naturaleza del escenario.

De este modo, se toman como referencia celdas cuyo centro dispone de un único AP a una altura de 10 m [46] y un número variable de STAs representativo de la alta densidad, que puede ser de hasta 60 terminales [55]. Se presupone que las STAs son dispositivos móviles y están a una altura de 1,5 m. [46]

5.2.1.2. Características Físicas, a Nivel de Enlace y Configuración de la Transmisión

En el apartado Introducción a la Caracterización de Entornos de Redes Inalámbricas, se anticipa la existencia de múltiples consideraciones a poner en valor en el estudio de las características físicas, de enlace y las configuraciones aplicables a la transmisión en el uso de la tecnología IEEE 802.11 en diferentes entornos.

Debido a la interrelación entre las versiones del estándar analizadas y las características específicas de las mismas tanto para la capa física como de enlace, se presentan de forma conjunta los parámetros principales de dichos niveles.

A continuación, se desarrolla la información resumida para los cuatro casos bajo estudio en términos de la **banda frecuencial** a utilizar, la **configuración de canales** en base a la selección y uso de los canales sobre las bandas estudiadas, el **ancho de** aplicable a cada caso, el uso de **características MIMO** y el **número de antenas** a utilizar tanto en APs como en STAs, la **potencia de transmisión** de estos, las **configuraciones OFDM** que se ajustan a la tecnología bajo estudio, y los **modelos de propagación** afines a los entornos analizados.

Asimismo, se presenta la información específica de **nivel de enlace** como el uso de **EDCA**, la aplicación de métodos de **agregación**, la configuración de **retransmisiones**, y el empleo de **RTS/CTS**.

De este modo, y con motivo de definir los parámetros especificados siguiendo el mismo planteamiento que el apartado anterior, se recoge la información disponible en una amplia selección de documentos que describen los cuatro escenarios de interés. En consecuencia, el análisis pertinente se realiza clasificado por cada entorno en cuestión.

En primer lugar, se extraen las características físicas y configuraciones del **ámbito residencial**.

La información recogida en base al uso de bandas frecuenciales muestra ejemplos de entornos domésticos sobre las tres bandas disponibles: 2,4 GHz, 5 GHz y 6 GHz. Sin embargo, en la gran mayoría de estudios se menciona la banda de 2,4 GHz –llegando, en algunos casos concretos, a multiplicar por cuatro su utilización frente a la de 5 GHz cuando ambas bandas están disponibles [56]– con canales de 20 MHz. Sobre la banda de 5 GHz se utilizan mayormente canales de 40 MHz.

Gran parte de los casos de uso incluyen MIMO con configuraciones de antenas distintas para APs y STAs. En el caso de los APs, oscilan entre las 2 y 4 antenas de transmisión y recepción, mientras que las STAs disponen de entre 1 y 2 antenas para cada función. Las potencias de transmisión de las antenas varían entre 17 dBm y 21 dBm para los APs, y entre 14 dBm y 18 dBm para las STAs.

Además, en las últimas versiones se incluye OFDM con intervalos de guarda bajos, y principalmente se hacen dos propuestas para el cálculo de pérdidas de propagación. Primero, se define un modelo de pérdidas por espacio libre [47] al que se le añaden pérdidas específicas por atravesamiento de paredes. Por otra parte, se define un modelo [46] que atiende a las pérdidas logarítmicas en base a la distancia incluyendo dos factores distintos para paredes y suelos.

A nivel de enlace, se considera el uso de EDCA con los parámetros por defecto por clase de acceso, agregación para el caso de MPDUs, número de repeticiones máximo de 10 por defecto y no se hace uso de RTS/CTS [46].

A continuación, se recogen las características físicas y configuraciones del **ámbito empresarial**.

En este ámbito también se hallan casos de aplicación sobre las tres bandas frecuenciales utilizadas en WLAN: 2,4 GHz, 5 GHz y 6 GHz. A diferencia del entorno residencia, la banda predominante entre la información analizada es la de 5 GHz, con un uso habitual de canales de 80 MHz. [46] [57] [58] [59]

Las configuraciones MIMO utilizadas [46] [60] [61] en términos del número de antenas, oscilan entre 2 x 2 y 10 x 8 [59] para los APs –definiéndose valores de potencia de transmisión [46] [59] de alrededor de 24 dBm–, y entre 1 x 1 y 2 x 2 para STAs –con potencias de transmisión entre 15 dBm y 21 dBm–.

Asimismo, se hace uso de OFDM con intervalos de guarda bajos, y con relación a los modelos de propagación se definen varios específicos para oficinas: la propuesta para entornos de interior de la ITU [49], el modelo InH de 3GPP [59] y un modelo indoor para LOS y NLOS [57].

A nivel MAC, se utiliza EDCA con los parámetros específicos para cada clase de acceso, agregación para el caso de MPDUs, número de repeticiones máximo de 10 por defecto y no se hace uso de RTS/CTS [46].

En el **ámbito interior de alta densidad** sucede lo mismo que para los dos casos anteriores, pero se intuye un mayor interés en su estudio para los protocolos más actuales.

La banda frecuencial más estudiada [62] [60] [63] [46] es la de 5 GHz con tamaños de canal de 80 MHz, aunque se recogen ejemplos para las tres bandas disponibles: 2,4 GHz, 5 GHz y 6 GHz.

Gran parte de los casos de uso con alta densidad de dispositivos también incluyen MIMO con configuraciones de antenas distintas para APs y STAs, de forma similar a los dos entornos anteriores. Por lo general [46] [60] [61], se definen entre las 2 y 4 antenas de transmisión y recepción para APs, mientras que las STAs disponen de entre 1 y 2 antenas para transmisión y recepción. Algunos estudios [60] [63] [46] contemplan potencias de transmisión menores que en casos anteriores, partiendo de rangos entre 4 dBm y 23 dBm para los APs, y estableciendo 15 dBm para las STAs.

Además, la aplicación de las últimas versiones del estándar incluye el uso de OFDM con intervalos de guarda bajos, y se definen dos modelos principales para el cálculo de las pérdidas de propagación en este tipo de entornos. En primer lugar, se propone el modelo TMB del TGax [60], diseñado para escenarios de alta densidad de interior. En segundo lugar, se propone el uso de pérdidas logarítmicas [46] en función de la distancia. Además de las anteriores, otro estudio emplea un modelo realista mediante ray tracing [63] para definir las pérdidas de obstáculos presentes en el escenario.

A nivel de enlace [46], al igual que en los entornos anteriores, se considera el uso de EDCA con parámetros por defecto por clase de acceso, agregación para el caso de MPDUs, número de repeticiones máximo de 10 por defecto y se añade el mecanismo RTS/CTS en casos concretos de alta densidad de dispositivos [63].

Para terminar con el análisis de las características físicas y configuración de la transmisión, se estudia el **ámbito exterior de alta densidad**. Al igual que en el entorno interior de alta densidad, se percibe un mayor interés en su análisis con las versiones más actuales del estándar.

Entre la información recogida, se hallan casos de uso de la banda de 2,4 GHz [46] [54] [53] [64] como de 5 GHz [46] [53] [62] [54] de igual manera, aunque se favorece la segunda en las últimas versiones de Wi-Fi. Sobre 2,4 GHz se utilizan canales de 20 MHz de tamaño principalmente. En el caso de la banda de 5 GHz, en cambio, se tiende a utilizar 80 MHz.

Las configuraciones MIMO son equivalentes al entorno anterior, oscilando entre 2 x 2 y 4 x 4 para los APs –con potencias de transmisión de hasta 30 dBm–, y entre 1 x 1 y 2 x 2 para las STAs –con potencias de transmisión de alrededor de 15 dBm–. Sin embargo, en el caso de OFDM también se hacen usos de intervalos de guarda entre símbolos de diferentes tamaños, tanto bajos –800 ns– como altos –3200 ns–.

Para la caracterización de las pérdidas de propagación en entornos urbanos, TGax propone los modelos UMi (Urban Micro-cell) y UMa (Urban Macro-cell) debido a su

planteamiento de escenario que se fundamenta en el despliegue de APs en celdas de forma similar a las redes móviles [46]. Sin embargo, también se ha de considerar que los entornos de exterior disponen de características a nivel de espectro muy distintas, por lo que también se deben contemplar modelos simples como las pérdidas por espacio libre. [65]

Finalmente, a nivel MAC se configura EDCA con los parámetros específicos para cada clase de acceso, agregación para el caso de MPDUs, número de repeticiones máximo de 10 por defecto y no se hace uso de RTS/CTS [46].

5.2.1.3. Caracterización del Tráfico de Red en Entornos WLAN

Para llevar a cabo la definición de los modelos de tráfico a implementar en el simulador de red, se deben estudiar tanto las aplicaciones más habituales y sus parámetros específicos como las características relativas al tipo de tráfico –ascendente, descendente, simétrico, asimétrico, tráfico de gestión, etc–.

Existe una gran variedad de aproximaciones para la caracterización del tráfico de forma realista, ya sea en base a las características estadísticas generales del tráfico IP o en base a la caracterización de servicios de red concretos.

Además, en el presente estudio se trata de cubrir la aplicabilidad de dichos modelos a una serie de entornos Wi-Fi específicos, por lo que los patrones de uso de las distintas aplicaciones pueden diferir en cada caso. En consecuencia, se deben estudiar, por un lado, las aplicaciones a implementar en cada escenario, y por otro, las herramientas que permitan realizar una caracterización del tráfico por aplicación.

5.2.1.3.1. Tipos de Tráfico por Entorno

Entre la documentación analizada, varios estudios [66] [67] que hacen distinción de tipos de tráfico en base a aplicaciones toman como referencia para la construcción diferenciada en entornos Wi-Fi la propuesta del grupo de estudio de IEEE 802.11ax para diferentes escenarios [46], en la que también se definen los tipos de tráfico a modelar en cada caso.

A continuación, se recoge de forma resumida la información relativa al **tráfico de red en base a aplicaciones comunes para cada entorno de interés**.

En el **entorno residencial**, el grupo TGax [46] propone las transmisiones caracterizadas en la Tabla 11. Se ha de tener en cuenta que los servicios considerados desde o hacia cada STA deben ser asignados de forma aleatoria sobre el número total de dispositivos de dicho rol.

Otro ejemplo de tráfico del entorno residencial es el tráfico IoT [68], que se define como tráfico en tiempo real asimétrico –de subida– en tiempo real sin pérdidas.

En estudios sobre casos reales de entornos domésticos [69] también se hace referencia a diferentes modelos de tráfico, como el acceso a aplicaciones web, VoD y audio y vídeo RTC. Además, en el análisis de tráfico sobre hogares se llega a detectar [45] hasta 20 protocolos distintos con sus aplicaciones asociadas.

Modelo de tráfico	ID de tráfico	Características	AC
Buffered Video Streaming	BV3, BV6	Asimétrico Bidireccional	VI
FTP	FTP	Asimétrico Bidireccional	BE
HTTP	HTTP	Asimétrico Bidireccional	BE
VoIP	VOIP	Simétrico Bidireccional	VO
MGMT: Beacon	Beacon de 220 octetos a 1 Mbps (2,4 GHz) o 6 Mbps (5 GHz) cada 100 TUs	Unidireccional	VI
Gaming	GMG	Asimétrico Bidireccional	VI
MGMT: Probe Req	TBD	Unidireccional	VI

Tabla 11: Resumen del modelo de tráfico para entorno residencial definido por el TGax. [46]

En el **entorno de oficinas**, el grupo TGax [46] realiza una propuesta similar al residencial, pero acotando los tipos de tráfico y aplicando un ajuste proporcional al número de nodos –STAs y APs– contemplados.

También se hace alusión a los diferentes tipos de tráfico sobre este tipo de escenarios en múltiples estudios de casos reales. Entre ellos, se contemplan casos de uso como transmisión de vídeo para teleconferencias, servicios cloud y streaming de audio y vídeo [27].

Además, cabe destacar el crecimiento drástico de las aplicaciones de tráfico de vídeo –incluyendo el vídeo bajo demanda, las teleconferencias y VR/AR– en los últimos años [70] sobre servicios como Skype, Hangouts o FaceTime en entornos empresariales.

Modelo de tráfico	ID de tráfico	Características	AC
Buffered Video Streaming	BV3, BV6	Asimétrico Bidireccional	VI
VDI	VDI	Asimétrico Bidireccional	VI
VoIP	VOIP	Simétrico Bidireccional	VO
MGMT: Beacon	Beacon de 280 octetos a 1 Mbps (2,4 GHz) o 6 Mbps (5 GHz) cada 100 Tus	Unidireccional	VI
MGMT: Probe Req	TBD	Unidireccional	VI

Tabla 12: Resumen del modelo de tráfico para entorno de oficinas definido por el TGax. [46]

Por otro lado, en el **entorno de interior de alta densidad**, el grupo TGax [46] propone un set de configuraciones de tráfico similar al entorno residencial, pero de mayor dimensión en términos de dispositivos.

Múltiples casos de uso [27] sobre entornos inalámbricos asignan a este tipo de escenarios flujos derivados de tráfico de streaming y AR/VR, como puede ser el caso de eventos sociales, o acceso web o transmisión de vídeo para entretenimiento en el caso de espacios de interior que se puedan asemejar a lugares turísticos. En general, destaca la presencia del tráfico de vídeo, como puede ser [66] VoD (Video over Demand).

Del mismo modo, también se propone para la simulación de escenarios de alta densidad tanto de interior como de exterior [71] la utilización de tráfico de tres tipos de uso común. Estos servicios son la transferencia de ficheros, streaming de vídeo sobre HTTP, y acceso a páginas web recurrentes.

Modelo de tráfico	ID de tráfico	Características	AC
Buffered Video Streaming	BV3, BV6	Asimétrico Bidireccional	VI
FTP	FTP	Asimétrico Bidireccional	BE
HTTP	HTTP	Asimétrico Bidireccional	BE
Gaming	GMG	Asimétrico Bidireccional	VI
VoIP	VOIP	Simétrico Bidireccional	VO
MGMT: Beacon	Beacon de 280 octetos a 1 Mbps (2,4 GHz) o 6 Mbps (5 GHz) cada 100 Tus	Unidireccional	VI
MGMT: Probe Req	-	Unidireccional	VI

Tabla 13: Resumen del modelo de tráfico para entorno interior de alta densidad definido por el TGax. [46]

Para finalizar la caracterización de las aplicaciones comunes a cada escenario, en relación con el **entorno exterior de alta densidad**, el grupo TGax [46] propone una serie de aplicaciones igual al entorno anterior, pero con un ligero aumento en los dispositivos.

Modelo de tráfico	ID de tráfico	Dirección	AC
Buffered Video Streaming	BV3, BV6	Asimétrico Bidireccional	VI
Multicast Video Streaming	MC2	Unidireccional	VI
FTP	FTP	Asimétrico Bidireccional	BE
HTTP	HTTP	Asimétrico Bidireccional	BE
Gaming	GMG	Asimétrico Bidireccional	VI
VoIP	VOIP	Simétrico Bidireccional	VO
MGMT: Beacon	Beacon de 280 octetos a 1 Mbps (2,4 GHz) o 6 Mbps (5 GHz) cada 100 Tus	Unidireccional	VI
MGMT: Probe Req	-	Unidireccional	VI

Tabla 14: Resumen del modelo de tráfico para entorno exterior de alta densidad definido por el TGax. [46]

Además de los anteriores, se mencionan aplicaciones como AR/VR en el ámbito de la educación [27], como también ocurría en los entornos de interior de alta densidad. En este caso, se aplica a los espacios universitarios abiertos. Asimismo, al ser un entorno susceptible de cubrir eventos, también se correspondería con la aplicación de transmisión de vídeo [27].

Al igual que para los entornos de interior, en la caracterización de tráfico sobre entornos de alta densidad tanto de exterior como interior [71] se propone la utilización de transferencia de ficheros, streaming de vídeo sobre HTTP y acceso a páginas web recurrentes.

5.2.1.3.2. Características Principales de los Tipos de Tráfico

Con relación al **modelado del tráfico**, TGax también define las características estadísticas de distintos tipos de tráfico de forma clasificada [72], de forma que se permite la implementación de estos sobre los entornos definidos [46].

Para ello, se sigue el mismo procedimiento que en el desarrollo de los escenarios, y se parte de una primera propuesta de metodología de evaluación para IEEE 802.11ax sobre la que iteran los participantes de las diferentes compañías que colaboran en el grupo.

En una primera aproximación [72], se caracterizan varios modelos de tráfico de vídeo, y también el tráfico de gestión de la red. Dicha propuesta se va nutriendo a lo largo de las revisiones hasta disponer de múltiples modelos de tráfico definidos en consenso por el grupo, como son el tráfico VoIP, FTP o HTTP. [72]

A continuación, se describen los tipos de tráfico principales, incluyendo sus características y una **introducción funcional** para el modelado de tráfico. La abstracción del tráfico basada en modelos estadísticos y sus parámetros concretos aplicables se recoge en el apartado Modelado de los tipos de tráfico.

Tráfico de gestión	Tráfico de vídeo	Otros tipos de tráfico comunes
Transmisión de probe requests y beacons	Wireless display, streaming con buffer, videoconferencias, streaming multicast, VDI	HTTP, FTP, VoIP, Gaming

Tabla 15: Resumen de los tipos de tráfico. [46]

TGax realiza una definición sencilla para los mecanismos de descubrimiento Wi-Fi en la que se diferencian las STAs en función de la situación de asociación a algún AP, donde se describen los aspectos principales del **tráfico de gestión**.

Para clientes no asociados a ningún AP, se definen periodos de sondeo para el envío de probe request, y en caso de no hallarse asociado un cliente tras varios intentos sobre todos los canales, se repite el mismo procedimiento duplicando dicho periodo.

Para clientes asociados, en cambio, se define un periodo de sondeo fijo, canales iguales a los que cada cliente está asociado siempre que el RSSI de los beacons del AP asociado sean superiores a una potencia mínima y tamaños de frame para las probe request predefinidos.

En otros estudios [45], sin embargo, cuando se trata de caracterizar entornos en función de uso y consumo de servicios de red, se tiende a excluir este tipo de tráfico del análisis, debido a que, en términos totales, supone porcentajes de tráfico ínfimos, inferiores al 1%.

Dentro del ámbito del **tráfico de vídeo** se recogen múltiples casuísticas, como se puede apreciar en la Tabla 15.

Para comenzar, se define el modelo de tráfico **Wireless Display**, que se corresponde con vídeo ligeramente comprimido, y comprende una aplicación de vídeo de salto único (single-hop) unidireccional, como puede ser la transmisión de vídeo desde una computadora a un monitor.

Los fragmentos o slices de vídeo se generan en un intervalo fijo definido por el equipo transmisor, y a pesar de que, por lo general, los fragmentos de vídeo se paquetizan en MPEG-TS para este tipo de aplicaciones, en el modelo de simulación propuesto por TGax [72] se obvia dicho procedimiento y se presupone una entrega directa al nivel MAC para su transmisión.

Otra aplicación de vídeo de uso habitual es la relacionada al tráfico **streaming con buffer** (Figura 10), y comprende el caso de servicios de vídeo de uso común como YouTube o Netflix. También se le denomina **vídeo bajo demanda**.

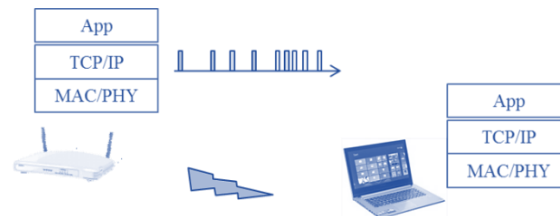


Figura 10: Diagrama de funcionamiento de VoD. [72]

Este tráfico tiene lugar en sentido unidireccional desde el servidor al equipo cliente, y se define su transmisión sobre TCP, realizando fragmentación en origen y presuponiendo variabilidad en múltiples aspectos como en delay, tiempo entre llegadas, etc.

A nivel de transporte se definen segmentos TCP de 1500 bytes y se fragmentan los paquetes de vídeo en segmentos TCP. Se añade latencia a los paquetes cuando los segmentos llegan al AP.

A lo largo de las revisiones del documento de la TGax [73], se amplía como especificación para este tipo de tráfico la transmisión de los ACK de TCP recíprocos desde el receptor y el tiempo entre llegadas respecto a la recepción de vídeo en destino.

Entre los estudios utilizados en el modelado de escenarios anterior, también se realiza el modelado correspondiente a este tipo de tráfico sobre tres topologías de red [66], basado en un estudio sobre tráfico real en el que se consumen tres vídeos distintos en tamaño a diferentes velocidades.

Además de los anteriores, otro modelo de tráfico de vídeo de interés es el de **videoconferencias** (Figura 11), que, a diferencia del anterior, es bidireccional y no depende fuertemente de un buffer sobre el que se almacena el vídeo de forma temporal. En este caso el tráfico se puede generar en cualquier estación, y existen requisitos de latencia debido al tipo de servicio en cuestión.

En la transmisión desde el AP se presupone la existencia de jitter, mientras que en sentido contrario se puede definir el tráfico de forma periódica a nivel de aplicación desde la STA.

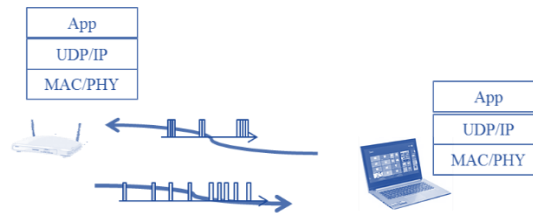


Figura 11: Diagrama de funcionamiento de tráfico de videoconferencia. [72]

Con relación a los patrones de eventos a nivel de aplicación –cuándo comienza la ejecución, cuánto dura el uso de una aplicación, etc.– se considera que para diferentes escenarios se pueden emplear distintos modelos.

A pesar de que funcionalmente tengan similitudes, una clase de tráfico de vídeo distinta a la anterior es el **streaming multicast** (Figura 12), que supone la transmisión unidireccional desde un AP a múltiples STA. El tráfico es generado, a diferencia del caso de videoconferencias, en un servidor de vídeo, y atraviesa múltiples saltos a través de internet antes de llegar al AP para alcanzar a las STAs.

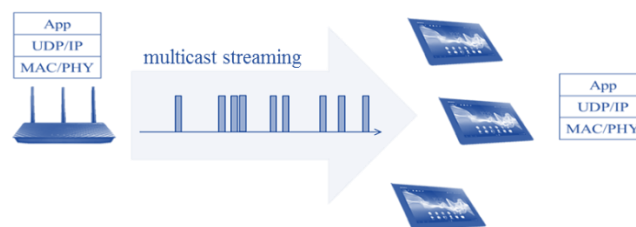


Figura 12: Diagrama de funcionamiento de streaming multicast. [72]

Como el tráfico entre el AP y las estaciones sufre jitter a lo largo de la red, se puede modelar del mismo modo que el streaming de vídeo. El tráfico se transporta sobre UDP y se transmite a MAC, en este caso, como tramas con direcciones RA multicast. La generación de vídeo, por tanto, sigue también un procedimiento similar que en el caso de streaming, pero asumiendo tasas de bit distintas.

Además de los anteriores, se incluye una última categoría de tráfico de vídeo, el tráfico **VDI (Virtual Desktop Infrastructure)**. En este tipo de aplicaciones se genera tráfico desde un servidor que atraviesa múltiples saltos antes de llegar al AP para alcanzar las STAs. La transmisión desde el AP a la STA es asimétrica, bidireccional, y de salto único (single-hop).

El tráfico VDI se transporta sobre TCP, y su modelado describe la atribución de tráfico desde el AP a la STA, la navegación de la aplicación VDI y el tráfico de vuelta desde la STA al AP. La generación de dicho tráfico se modela como paquetes de tamaño variables que se crean en intervalos que también son variables.

Entre las aplicaciones comunes distintas al tráfico de vídeo, se define el tráfico relativo a la **navegación web HTTP**. El tráfico HTTP está gobernado por la estructura de las páginas web, y por lo general tiene un perfil a ráfagas debido a las características de la interacción humana, donde los patrones de tráfico HTTP se corresponden con la Figura 13.

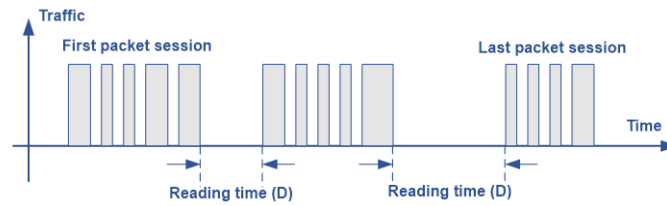


Figura 13: Patrón de tráfico HTTP.

Se trata de tráfico asimétrico, bidireccional y con frames formados de bloques de tamaño fijo en una dirección, de un servidor HTTP a un cliente HTTP sobre transporte TCP. En sentido contrario tienen lugar las confirmaciones de recepción ACK a las solicitudes HTTP.

Las sesiones representan la transferencia de páginas web desde el servidor al cliente, y el intervalo de lectura representa el tiempo que un cliente dedica a la lectura de la página. La cantidad de información transferida desde el servidor al cliente está gobernada por la estructura de la página web. Las páginas web se suelen componer de un objeto principal y múltiples objetos embebidos. El tráfico total transferido de un servidor a un cliente es equivalente al tamaño total del objeto principal y los embebidos,

El cliente HTTP envía una solicitud HTTP que tiene un tamaño constante para iniciar el evento en el servidor. Dependiendo de la versión de HTTP el tráfico relativo a la descarga de los objetos del servidor puede seguir un patrón en ráfagas o persistente.

Además, los ACK recíprocos se generan con tamaño fijo y con un tiempo entre llegadas establecido desde la recepción del frame HTTP en el cliente.

Otra aplicación de uso común es la transferencia de ficheros mediante **FTP (File Transfer Protocol)** (Figura 14). El protocolo FTP se utiliza para la transmisión de ficheros desde una fuente FTP a un cliente FTP a través de una red basada en TCP. Se trata de tráfico asimétrico y bidireccional con frames de datos compuestos por bloques de datos de tamaño fijo en la dirección de la fuente al cliente, y con respuestas ACK del cliente a la fuente.

El tráfico FTP se modela como una secuencia de transferencias separadas por el tiempo de lectura, que se define como el tiempo desde el final de la transmisión de un fichero hasta el comienzo de la siguiente transmisión de ficheros.

El tráfico relativo a los ACKs recíprocos a las transferencias se genera como transmisiones de tamaño fijo con tiempo entre llegadas establecido desde la recepción del frame FTP en el cliente.

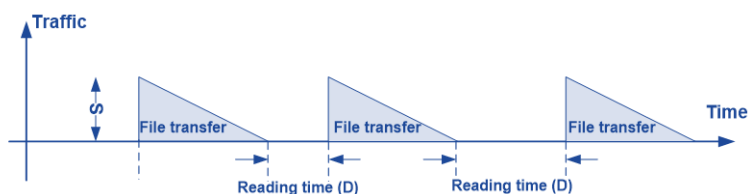


Figura 14: Diagrama de tráfico FTP. [72]

Otra aplicación de amplia utilización es la relativa al servicio **VoIP**, que permite la comunicación de voz en tiempo real a través de redes de paquetes. El tráfico VoIP comprende periodos de habla activa y silencios, que se puede entender como una actividad describable en dos estados según un modelo de Markov (Figura 15). Se trata de un modelo de tráfico bidireccional asimétrico.

El usuario VoIP siempre se encuentra en estado de silencio o en habla activa, y se define a como la probabilidad de transición de estado de silencio a habla, y b como la probabilidad de transición de estado de habla a silencio. Por lo tanto, la probabilidad de mantenerse en estado de silencio es $1-a$, y de quedarse en estado de habla $1-b$.

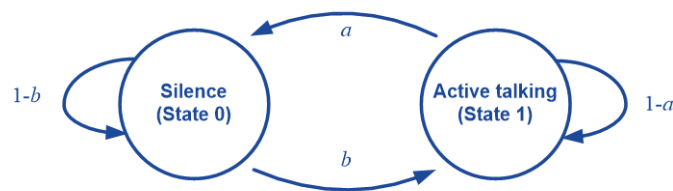


Figura 15: Máquina de estados del tráfico VoIP. [72]

Por último, también se introduce el modelado de **tráfico gaming**, que toma como referencia un modelo de juego multijugador online. Se trata de un tipo de tráfico específico en doble sentido, asimétrico, con llegadas de tráfico irregulares y requisitos adicionales, como limitación de delay para tráfico en tiempo real.

Por otro lado, existen alternativas para la generación de tráfico genérico de internet [74], a través de herramientas de generación de tráfico realista, como la propuesta basada en PPBP (Poisson Pareto Burst Process), un modelo para el tráfico de red LRD (Long-Range Dependent) que provee una solución simple y precisa para generar tráfico afín a las características estadísticas de las redes IP. Para ello, este tipo de herramientas se basan en parámetros como la longitud media de una ráfaga de tráfico, la tasa de bits de cada flujo, o la tasa de llegadas de paquetes, tomando como referencias métricas reales para su validación o reproduciendo modelos de procesos estocásticos que reproducen características concretas del tráfico.

Sin embargo, el presente estudio trata no solo de definir distintos escenarios, o de realizar una aproximación genérica del tráfico en la red, sino que tiene como fin la implementación final de procedimientos de QoS, que requieren la diferenciación de tipos de tráfico, por lo que se focaliza en propuestas que cumplan dichos requerimientos.

Una vez presentados los fundamentos teóricos en términos de QoS y desarrollado el estudio de aplicación de la tecnología IEEE 802.11, se han de introducir las herramientas de procesamiento de información que posibilitan las tareas de mejora de la QoS.

5.3. Técnicas de Aprendizaje Automático

Este trabajo tiene como objetivo sentar las bases para la automatización de procedimientos de mejora de la QoS basándose en las bondades que las técnicas de

aprendizaje automático presentan frente a conjuntos de datos complejos, y la aplicación de estas adquiere aún más sentido si se proyecta sobre el procesado de volúmenes de datos cada vez mayores [16] provenientes de redes inalámbricas reales.

Esto se debe a que el **aprendizaje automático** surge como la disciplina de la inteligencia artificial (Artificial Intelligence, AI) que permite solventar los problemas que las grandes cantidades de información y la complejidad de los datos en entornos prácticos supone para la programación convencional. [75] Por definición, está dedicado al estudio de procesos que posibilitan el aprendizaje en sistemas informáticos guiado principalmente por colecciones de datos, más que la propia programación explícita. [76]

El proceso de aprendizaje [77] es llevado a cabo mediante iteraciones en las que se utilizan algoritmos que favorecen la mejora de sistemas, la descripción de los datos y la predicción de resultados. Para ello, se incluye un procedimiento inicial de recopilación de datos con el que se construye un modelo estadístico que permite entrenar al algoritmo.

5.3.1. Clasificación de los Algoritmos de Aprendizaje Automático

La clasificación habitual (Figura 16) del aprendizaje automático [76] se compone por el **aprendizaje supervisado** (supervised learning), que hace uso de datos –con información de entrada y salida del algoritmo– previamente etiquetados, el **aprendizaje no supervisado** (unsupervised learning), que utiliza datos de entrada sin etiquetar, y el **aprendizaje por refuerzo** (reinforcement learning), guiado por la retroalimentación del análisis de datos para la obtención del mejor resultado posible.

En primer lugar, el **aprendizaje supervisado** [77] hace referencia a cualquier proceso de aprendizaje en el que se obtiene una función que relaciona un tipo de entrada con un tipo de salida utilizando muestras que constan de datos que tienen valores relativos tanto a la una como la otra. En consecuencia, es una técnica utilizada para la realización de predicciones a partir de los datos empleados en el entrenamiento del algoritmo, donde los datos suelen ser pares de vectores compuestos por los valores de la entrada y la salida.

En términos matemáticos [78], se define un set de datos $D = \{(x^n, y^n), n = 1, \dots, N\}$ donde x es la entrada e y la salida. El objetivo, entonces, es aprender la relación entre ambas mediante una función que sea capaz de describir la salida en función de la entrada y de forma que, si se añade una nueva entrada (x'), la predicción de la salida (y') sea precisa. Estadísticamente, la salida está condicionada por la entrada y el set de datos disponible ($p(y|x, D)$).

Dos ejemplos típicos [79] son el aprendizaje de clasificación y la regresión. Si tomando una entrada x la salida y es una categoría o resultado discreto de una serie de posibles clases, se considera problema de **clasificación**. Esta consiste en la segmentación de un dominio en regiones que pueden ser asociadas a las clases de interés de una aplicación concreta.

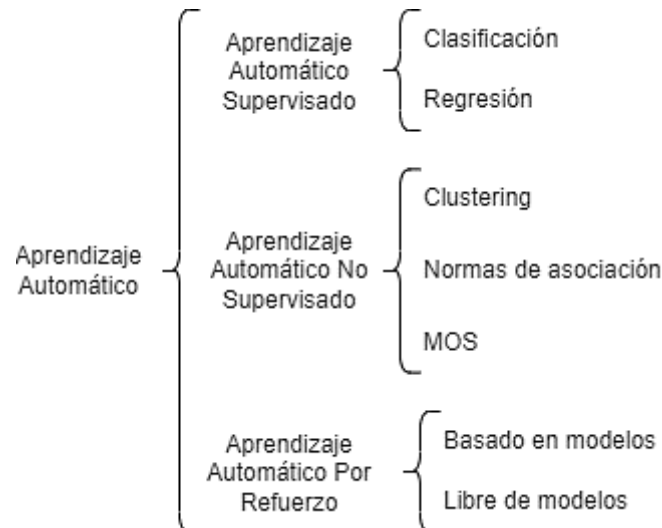


Figura 16: Clasificación del Aprendizaje Automático.

Sin embargo, si la salida y es numérica o continua, se denomina problema de **regresión**. La regresión se fundamenta en la obtención mediante inferencia de una función de valor real –denominada función de regresión– cuyo valor se corresponde a la media de una variable dependiente –la respuesta o salida– condicionada por una o más variables –o entradas– independientes.

En segundo lugar, se define el **aprendizaje automático no supervisado** [79] como el proceso capaz de aprender estructuras en la ausencia de una salida identificada –datos etiquetados– o retroalimentación. De este modo, sus técnicas facilitan el análisis de conjuntos de datos sin procesar, y la generación de conocimiento analítico frente a datos sin etiquetas.

En términos matemáticos [78], dado un set de datos $D = \{x^n, n = 1, \dots, N\}$ donde x es la entrada, el objetivo una descripción compacta de dicha entrada. A diferencia del aprendizaje supervisado, no hay una variable de predicción, y el modelado a realizar de forma estadística es la distribución de probabilidad de la entrada ($p(x)$).

Tres ejemplos típicos de aprendizaje no supervisado son las técnicas de agrupamiento, las normas de asociación y los mapas autoorganizados. El **clustering** o agrupamiento consiste en la asignación de una serie de objetos dentro de grupos denominados clusters, o lo que es lo mismo, la partición de un set de ejemplos en grupos. [79] De forma intuitiva, los objetos del mismo cluster tienen mayores similitudes en algún aspecto entre ellos que los objetos pertenecientes a clusters distintos.

Por otro lado, las **normas de asociación** [79] se basan en la extracción de normas de la forma $X \rightarrow Y$ donde tanto X como Y son sets de ítems de forma que, si un set X ocurre en una muestra, es probable que también lo haga Y . Además, la norma se asocia a dos estadísticas medidas en el conjunto de datos. La frecuencia o soporte de una norma ($fr(X \rightarrow Y)$) es el número de muestras donde $X \cup Y$, y la confianza (confidence) la probabilidad condicionada observada $P(Y|X) = fr(X \cup Y)/fr(X)$.

Los **mapas autoorganizados** (self-organizing map, SOM) [79] son un método de análisis de datos computacional proveniente de la computación neuronal que produce mapeos de datos no lineales a dimensiones inferiores. Se pueden entender como un algoritmo de clustering que produce un conjunto de grupos organizados en una cuadrícula regular.

En tercer lugar y, a diferencia de los tipos de aprendizaje automático anteriores, el **aprendizaje por refuerzo** [79] es un aprendizaje conductual. Describe una larga clase de problemas característicos de agentes autónomos que interactúan con un entorno y de esta forma, se plantean problemas secuenciales de toma de decisiones con recompensas. El objetivo es alcanzar una política de funcionamiento que maximice la recompensa según avanza el tiempo.

Existen varias clasificaciones del aprendizaje por refuerzo. Una distinción común es la definición de los **algoritmos basados en modelos**, que utilizan la función de transición y recompensa para la estimación de la política óptima, y los **algoritmos libres de modelo**, que tratan de encontrar la política óptima con un conocimiento muy limitado de las dinámicas del entorno.

La naturaleza de los procedimientos de mejora a implementar en el caso de la mejora de la QoS definidos anteriormente se adecúan a un **problema de clasificación**, ya que en ambos se trata de obtener el contexto del usuario y la red mediante inferencia, que es resultado directo del análisis de datos extraídos del entorno y la diferenciación de grupos –entornos y perfiles– tras procesar la información.

Las tareas de clasificación tienen una estrecha relación con el **modelado de datos**. Mediante el desarrollo de esta se permite identificar grupos similares entre sí, lo que facilita la tarea de creación de modelos, definiendo una serie de propiedades simplificadas que permiten la provisión de explicaciones intuitivas sobre los aspectos relevantes de la colección de datos, teniendo en cuenta la correlación entre las variables utilizadas y la generación de grupos mediante clasificación.

Específicamente, se centra la atención sobre el **aprendizaje automático no supervisado**, ya que la información a procesar obtenida de la red en la realidad no dispone de etiquetas que referencien al escenario en la diferenciación de entornos o al usuario en la clasificación de perfiles.

5.3.1.1. Algoritmos de Clustering

Los problemas de clasificación sobre datos sin etiquetar se realizan mediante **clustering o agrupamiento**. Estas técnicas, como se adelantaba en la Técnicas de Aprendizaje Automático, conforman el área del aprendizaje no supervisado dedicado a la búsqueda de patrones ocultos sobre datos de entrada sin etiquetar, aprendiendo así sobre la estructura de los datos y extrayendo características en común para la formación de grupos.

Este proceso requiere la organización de los datos de forma que se obtenga alta similitud intra-cluster y baja similitud inter-cluster, para lo que se hace uso de medidas

de distorsión, también denominadas **medidas de similitud**. El proceso de clustering resulta más sencillo cuantos menos atributos sean necesarios para la definición de un ejemplo perteneciente a un grupo. [80]

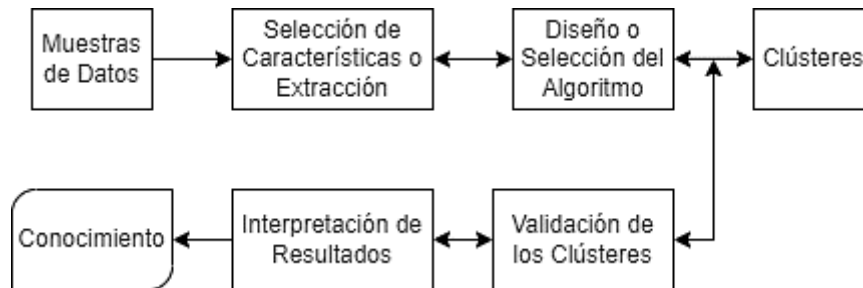


Figura 17: Proceso de generación de conocimiento mediante clustering.

El **procedimiento habitual** [16] para la obtención de los clústeres se muestra en la Figura 17. En un inicio, se dispone de las muestras de datos de las cuales se han de extraer características. El proceso de selección de características se fundamenta en la creación de métricas de similitud, en base a la que se diseña el algoritmo de agrupamiento. Tras aplicar el algoritmo sobre el conjunto de datos, se obtienen los clusters, que deben ser validados en una primera fase de diseño. Después, se interpretan los resultados para finalmente obtener conocimiento mediante la inferencia de los patrones presentes en los datos.

Categoría	Algoritmos típicos
Algoritmos de clustering basados en particiones	K-Means, K-Medoids, PAM, CLARA, CLARANS
Algoritmos de clustering jerárquicos	BIRCH, CURE, ROCK, Chameleon
Algoritmos de clustering basados en la teoría difusa	FCM, FCS, MM
Algoritmos de clustering basados en la distribución	DBCLASD, GMM
Algoritmos de clustering basados en la densidad	DBSCAN, OPTICS, Mean-shift
Algoritmos de clustering basados en la teoría de grafos	CLICK, MST
Algoritmos de clustering basados en cuadrículas	STING, CLIQUE
Algoritmos de clustering basados en la teoría fractal	FC
Algoritmos de clustering basados en modelos	COBWEB, GMM, SOM, ART

Tabla 16: Clasificación tradicional de los algoritmos de clustering. [81]

Existen varios enfoques para la **categorización de algoritmos de agrupamiento**. En general, se proponen clasificaciones que toman como eje la base para la medida de la similitud entre muestras y la formación de los clústeres. Un modelo de clasificación tradicional con ejemplos de algoritmos ampliamente utilizados se recoge en la Tabla 16.

El **agrupamiento en clústeres basado en particiones** [82] realiza una organización de datos en grupos no jerárquicos. La idea original [81] es establecer el centro de los

puntos de datos como el centro del clúster, como se puede apreciar en la Figura 18. Resultan técnicas notoriamente eficientes y de baja complejidad, pero sensibles a las condiciones y valores atípicos iniciales. Además, requieren del conocimiento previo del número de clústeres, que a su vez tiene un alto impacto en el resultado. Los algoritmos de este tipo más populares son **K-Means** y **K-Medoids**.

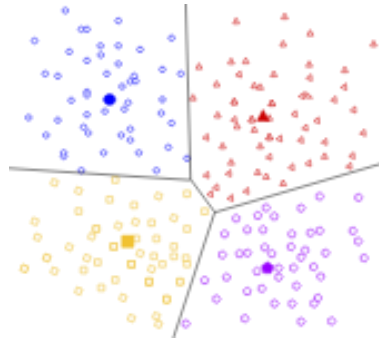


Figura 18: Ejemplo de clustering basado en particiones. [82]

El **agrupamiento en clústeres jerárquico** [82], también denominado agrupamiento en clústeres basado en la conectividad, genera árboles de clústeres mediante el establecimiento de relaciones de jerarquía entre muestras. El resultado de un árbol que separa las muestras en dos ramas se puede observar en la Figura 19. Este tipo de agrupamiento puede tomar dos enfoques distintos [81], aglomerativo y divisivo. Los agrupamientos aglomerativos desarrollan el proceso de formación de clústeres en base a muestras vecinas de forma recursiva. En cambio, los agrupamientos divisivos realizan el proceso contrario. Son técnicas de clustering que resultan de especial interés para datos jerárquicos, y permiten seleccionar cualquier cantidad de clústeres si se establecen límites en los niveles de ramificación del árbol de forma adecuada. Los algoritmos jerárquicos más conocidos son **BIRCH**, **ROCK** y **CURE**.



Figura 19: Ejemplo de clustering jerárquico. [82]

El **agrupamiento en clústeres basado en la teoría difusa** [81] toma como idea principal que el valor discreto -0 o 1 relativo a la pertenencia de una muestra del conjunto de datos a un grupo es cambiada en un intervalo continuo $[0, 1]$ para describir dicha relación de pertenencia de forma razonable. En la Figura 20 se puede apreciar la superposición de grupos formados a través de este modelo de clustering. Este tipo de agrupamiento tiene como bondades el realismo a la hora de asignar la probabilidad de pertenencia de muestras a grupos y una exactitud relativamente alta. Por el contrario, en general no muestran alta escalabilidad, disponen de alta sensibilidad a los datos iniciales, y requieren conocer el número de clústeres con antelación. Algoritmos típicos de este tipo son **FCM**, **FCS** y **MM**.

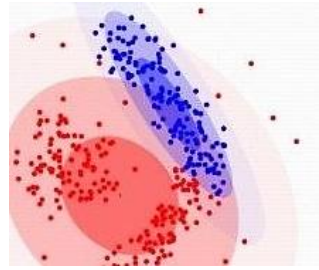


Figura 20: Ejemplo de clustering basado en teoría difusa. [83]

El **agrupamiento en clústeres basado en la distribución** [82] es el modelo más próximo al análisis estadístico, y es que el procedimiento de formación de grupos se realiza en función de la distribución seguida por las muestras. Por ejemplo, los clústeres pueden ser compuestos por distribuciones gaussianas, como es el caso de la Figura 21. Según aumenta la distancia al centro de la distribución, disminuye la probabilidad de que una muestra pertenezca a esa distribución. Tienen como ventaja [81] una definición de probabilidad de pertenencia a clústeres realista y relativamente alta escalabilidad. Como limitación, existe la necesidad de conocer el tipo de distribución seguida por los datos y conforma una metodología con alta complejidad temporal. De hecho, necesitar conocer la distribución con anterioridad hace de la hipótesis de clustering algo que no resulta completamente correcto. Dos algoritmos ampliamente utilizados son **DBCLASD** y **GMM**.

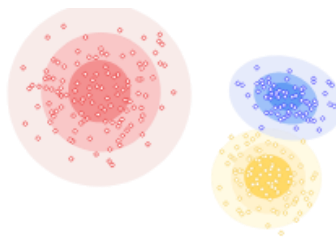


Figura 21: Ejemplo de clustering basado en la distribución. [82]

El **agrupamiento en clústeres basado en la densidad** [81], en cambio, trata de conectar espacios con alta densidad de muestras en forma de grupos. Este tipo de algoritmos resultan métodos de alta eficiencia adecuados para conjuntos de datos con formas arbitrarias, pero presentan dificultades con conjuntos de datos donde se encuentran distintas densidades y dimensiones. Asimismo, esta clase de técnicas descartan muestras atípicas de los clústeres, como se puede apreciar en la Figura 22. Un ejemplo habitual de algoritmo es **DBSCAN**.

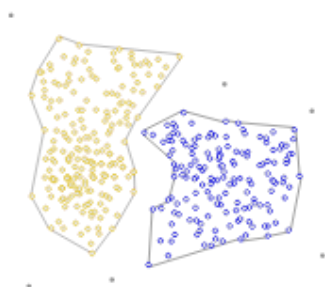


Figura 22: Ejemplo de clustering basado en la densidad. [82]

De acuerdo con el **agrupamiento en clústeres basado en la teoría de grafos** [81], el proceso de agrupación se realiza en el grafo donde el nodo es considerado como el punto de datos y las aristas la relación entre los datos. Entre las bondades de este tipo de clustering se destaca la alta eficiencia y precisión. Por el contrario, el aumento de complejidad en el grafo produce un dramático aumento de la complejidad temporal del algoritmo. Dos algoritmos de clustering basados en la teoría de grafos son **CLICK** y **MST**.

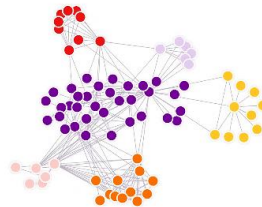


Figura 23: Ejemplo de clustering basado en la teoría de grafos. [84]

El **agrupamiento en clústeres basado en cuadrículas** [81] se fundamenta en la modificación del espacio de datos original en una estructura de cuadrícula con tamaño fijo. Entre sus ventajas se encuentra una baja complejidad temporal, alta escalabilidad y adecuación al procesado paralelo y actualización incremental. Sin embargo, cuenta con la desventaja de depender directamente de la granularidad y de tener una baja calidad y precisión en la formación de clusters a costa de mantener una alta eficiencia en los cálculos de agrupamiento. Los algoritmos típicos de esta clase son **STING** y **CLIQUE**.

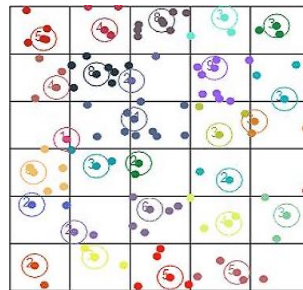


Figura 24: Ejemplo de clustering basado en cuadrículas. [85]

El **agrupamiento en clústeres basado en la teoría fractal** se fundamenta en la división de la geometría en múltiples partes que comparten características comunes con el total de muestras. Resulta una técnica altamente eficiente y escalable, que es válida para conjuntos de datos de forma arbitraria y alta dimensión. Sin embargo, la hipótesis no es completamente correcta, ya que su resultado depende directamente de los parámetros utilizados. El algoritmo más utilizado de este tipo es **FC**.



Figura 25: Ejemplo de clustering basado en la teoría fractal. [86]

El **agrupamiento en clústeres basado en modelos** parte de la idea inicial de definir un modelo para cada clúster y encontrar el que mejor se adecúe a este. Hay dos clases principales de algoritmos, los basados en el aprendizaje estadístico y los basados en redes neuronales. En general, cuentan con la ventaja de resultar modelos diversos y bien desarrollados que proveen descripciones adecuadas de los datos. Aun así, disponen de una complejidad temporal alta, y la hipótesis puede no resultar del todo correcta por la predisposición a utilizar un modelo en concreto y la fuerte dependencia de los resultados a los parámetros utilizados. Ejemplos de algoritmos de agrupamiento en clústeres basados en modelos de **aprendizaje estadístico** son **COBWEB** y **GMM**, mientras que ejemplos de los basados en **redes neuronales** son **SOM** y **ART**.

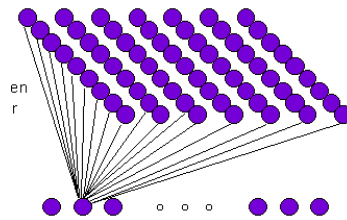


Figura 26: Ejemplo de clustering basado en modelos de redes neuronales. [87]

En la actualidad, de los enfoques anteriormente desarrollados destacan cuatro, que reúnen los algoritmos más ampliamente utilizados. [82] Estos se tratan del **agrupamiento basado en particiones**, donde se destaca K-Means, el **agrupamiento basado en densidad**, donde se destaca DBSCAN, el **agrupamiento basado en la distribución**, donde destaca GMM, y el **agrupamiento jerárquico**, donde se adoptan aproximaciones tanto de aglomeración como de división.

6. Análisis de Alternativas

La consecución de los objetivos con los mejores resultados posibles requiere la toma de decisiones técnicas que guíen el desarrollo del estudio. En el ámbito de este trabajo, se consideran como problemáticas de ingeniería que requieren un Análisis de Alternativas parametrización del entorno WLAN, la selección de indicadores de rendimiento clave, y los algoritmos cuya aptitud se ha de evaluar en base a su ajuste al estudio.

6.1. Alternativas Bajo Estudio

En este apartado se exponen y describen los objetos de decisión principales del Trabajo Fin de Máster, con el propósito de ser introducidos debidamente para su posterior selección.

6.1.1. Parametrización de los entornos WLAN en el simulador

Una vez realizado el Resumen del Estudio de Escenarios de Aplicación del Estándar IEEE 802.11, se pueden extraer valores aproximados que sirvan como referencia para la caracterización de los entornos de red en las simulaciones.

En consecuencia, a continuación, se presenta el mapeado de la información recogida en el Estado del Arte a parámetros de simulación, realizando un acercamiento a los valores a implementar en base a las características disponibles en el simulador de red.

Para ello, se parte de las conclusiones generales del estudio teórico, ampliando, cuando sea necesario, la propuesta de la Tabla 10 tomada como referencia durante el análisis de la documentación. Una vez presentados los **parámetros y sus valores preliminares**, se introducen las **herramientas de modelado** de red disponibles en el simulador, de forma que se expongan las distintas alternativas para llevar a cabo la implementación final de los escenarios. Por último, se cruzan los resultados de ambos apartados para obtener un **resumen de las alternativas** definidas.

6.1.1.1. Mapeado Preliminar de los Resultados del Estudio de Escenarios de Aplicación de la Tecnología IEEE 802.11 a Parámetros de Simulación

En términos de **topología** y **definición espacial** del entorno, se definen el plano y la configuración de los dispositivos. Para convertir los datos del estudio teórico en parámetros de simulación, en la Tabla 17 se recoge una abstracción de la realidad de los cuatro entornos contemplados en base a 8 datos.

El **tipo de plano** hace referencia a la entidad abstraída en la simulación, que puede ser la representación de habitaciones o espacios con particularidades del tipo de escenario. Relacionado al tipo de plano se hallan la **forma** –que indica qué figura se toma como referencia para la definición del plano– y el **área** a estudiar para representar el entorno. Además, se define la **altura máxima** sobre la que se pueden encontrar los dispositivos, que puede variar tanto para entornos interiores como exteriores.

Por otro lado, las características a definir sobre los dispositivos son el tipo de **posicionamiento** –si está predeterminado sobre las dimensiones del plano o se establece de forma aleatoria–, el **número de dispositivos** de cada clase y la **altura** a la que se ubican, y el modelo de **movilidad** para los APs y STAs.

Característica	Entorno residencial	Entorno empresarial	Entorno interior de alta densidad	Entorno exterior de alta densidad
Tipo de plano	Apartamento	Oficina	Genérico de interior	Genérico de exterior
Forma	Cuadrada	Rectangular	Celda	Celda
Área	Poco variable: 80m ² – 100m ²	Muy variable: en función del tipo de empresa: 30m ² – 600m ²	Tamaños totales: a partir de 600m ² . Celdas medianas: a partir de 150m ²	Tamaños totales: a partir de 100000m ² . Celdas grandes: a partir de 1400m ²
Altura máxima	3m	3m	3m	-
Posicionamiento	APs aleatorio STAs aleatorio	APs aleatorio STAs aleatorio	APs céntrico STAs aleatorio	APs céntrico STAs aleatorio
Número de dispositivos	APs: fijo (1) STAs: variable (máx 16)	APs: variable (máx 5) STAs: variable (máx 4 por puesto)	APs: fijo (1) STAs: variable (máx 50)	APs fijo (1) STAs: variable (máx 60)
Altura de dispositivos	APs: 1,5m STAs: 1,5m	APs: 3m STAs: 1,5m	APs: 3m STAs: 1,5m	APs: 10m STAs: 1,5m
Movilidad	APs: no STAs: modelo mixto	APs: no STAs: no	APs: no STAs: sí	APs: no STAs: sí

Tabla 17: Características topológicas generales extraídas del estudio teórico.

Con relación a las **características específicas de configuración de la tecnología**, se definen múltiples parámetros a simular a nivel físico y de enlace, y se recogen los datos extraídos del apartado Características Físicas, a Nivel de Enlace y Configuración de la de forma resumida en la Tabla 18.

Este tipo de parámetros parte de la definición del uso del espectro, para lo que en primer lugar se define la **banda de frecuencias** a emplear, el tipo de **asignación de canales**, y el **ancho de canal**.

Después, se incluye la configuración específica de las antenas, incluyendo la disponibilidad de **MIMO** y el **número de antenas**, además de la **potencia de transmisión** para APs y STAs.

A continuación, se define si se considera el uso de **OFDM** y el intervalo de guarda entre símbolos en caso afirmativo, y se especifican los **modelos de propagación** utilizados habitualmente en cada escenario.

Una vez establecido lo anterior, se determina la utilización de **EDCA** –junto con sus parámetros específicos– y se concretan una serie de configuraciones a nivel de enlace como el uso de **agregación** –para MPDU o MSDU–, el **número de repeticiones** y la utilización de **RTS/CTS**.

Característica	Entorno residencial	Entorno empresarial	Entorno interior de alta densidad	Entorno exterior de alta densidad
Banda frecuencial	2,4 GHz / 5 GHz / 6 GHz	2,4 GHz / 5 GHz / 6 GHz	2,4 GHz / 5 GHz / 6 GHz	2,4 GHz / 5 GHz / 6 GHz
Configuración de canales	Estática: compartición de canal.	Estática: reutilización de frecuencias.	Estática: reutilización de frecuencias.	Estática: reutilización de frecuencias.
Ancho de canal	20 MHz / 80 MHz / 80 MHz	20 MHz / 80 MHz / 80 MHz	20 MHz / 80 MHz / 80 MHz	20 MHz / 80 MHz / 80 MHz
Características MIMO	Sí	Sí	Sí	Sí
Número de antenas AP y STA	APs 2x2 / 4x4 STAs 1x1 / 2x2	APs 2x2 / 4x4 / 10x8 STAs 1x1 / 2x2	APs 2x2 / 4x4 STAs 1x1 / 2x2	APs 2x2 / 4x4 STAs 1x1 / 2x2
Características OFDM	Sí (800ns)	Sí (800ns)	Sí (800ns)	Sí (800ns / 3200ns)
Potencia TX AP y STA	APs: 17dBm-21dBm STAs: 14dBm-18dBm	APs: 24dBm STAs: 15dBm-21dBm	APs: 4dBm-23dBm STAs: 15dBm	APs: -1dBm-30dBm STAs: 15dBm
Modelo de propagación	Espacio libre + paredes Logarítmicas (d) + paredes + suelos	ITU Indoor InH 3GPP (N)LOS Indoor	TGax TMB Logarítmicas (d)	UMi UMa Logarítmicas (d)
EDCA	Sí (valores por defecto por AC)	Sí (valores por defecto por AC)	Sí (valores por defecto por AC)	Sí (valores por defecto por AC)
Agregación	A-MPDU (64MPDU) Tamaño de ventana BA	A-MPDU (tam máx) Tamaño de ventana BA	A-MPDU (tam máx) Tamaño de ventana BA	A-MPDU (64MPDU) Tamaño de ventana BA
Repeticiones (retries)	Valor por defecto: 10	Valor por defecto: 10	Valor por defecto: 10	Valor por defecto: 10
RTS/CTS	No	No	No/Sí	No/Sí

Tabla 18: Características técnicas generales extraídas del estudio teórico.

Tipo de tráfico	Entorno residencial	Entorno empresarial	Entorno interior de alta densidad	Entorno exterior de alta densidad
Gestión	✓	✓	✓	✓
Wireless Display				
Buffered Video	✓	✓	✓	✓
Videoconferencia		✓		
Streaming Multicast				✓
VDI		✓		
HTTP	✓		✓	✓
FTP	✓		✓	✓
VoIP	✓	✓	✓	✓
Gaming	✓		✓	✓
Nuevos Servicios (IoT, AR/VR...)	✓		✓	✓

Tabla 19: Resumen de tipos de tráfico por entorno contemplados en el Estado del Arte.

Por último, respecto a la **simulación de tráfico**, se resume la información recogida en base a dos aspectos. Por un lado, se precisan los **tipos de tráfico** destacados por escenario (Tabla 19), basándose en la información extraída de los apartados Tipos de Tráfico por Entorno y Caracterización de los Tipos de Tráfico por Entorno, y por otro, se presenta la asignación específica de la **tasa de dispositivos** por tipo de tráfico en base a la aproximación de caracterización de tráfico del TGax.

Como se puede observar en la Tabla 20, el TGax hace una propuesta de implementación de tráfico en función del tipo de tráfico, la tasa de dispositivos de cada tipo para los entornos bajo estudio y el sentido del tráfico. De este modo, se entiende por cada dato numérico presente en la tabla la combinación de APs y STAs implicados en los flujos de tráfico, ya sea en sentido ascendente (STA/AP) o descendente (AP/STA).

Características de Tráfico		Entorno residencial	Entorno empresarial	Entorno interior de alta densidad	Entorno exterior de alta densidad
Tipo de tráfico	Dirección del flujo	(Máx: 10)	(Máx: 64)	(Máx: 30)	(Máx: 30)
Gestión	AP → STA	1/0	1/0	1/0	1/0
	STA → AP				
Buffered Video	AP → STA	6/6	8/8	20/20	25/25
	STA → AP	1/1		4/4	10/10
Videoconferencia	AP → STA				
	STA → AP				
Streaming multicast	AP → STA				1/12
	STA → AP				
VDI	AP → STA		48/48		
	STA → AP				
HTTP	AP → STA	4/4		12/12	15/15
	STA → AP				
FTP	AP → STA	2/2		4/4	4/4
	STA → AP	2/2		4/4	4/4
VoIP	AP → STA	2/2	10/10	12/12	20/20
	STA → AP				
Gaming	AP → STA	3/3		16/16	25/25
	STA → AP	3/3		16/16	25/25

Tabla 20: Asignación de nodos con tráfico por escenario según el TGax.

Sin embargo, cabe mencionar que, además de la propuesta del TGax, durante el Resumen del Estudio de Escenarios de Aplicación del Estándar IEEE 802.11 también se han observado otras definiciones de tráfico relativo tanto a los tipos de tráfico contemplados por dicho grupo como de tipos de tráfico IoT y AR/VR, aunque comprenden aproximaciones genéricas donde se destaca su uso en algunos entornos. En consecuencia, aunque se contemplan como alternativa de tipo de tráfico (Tabla 19) no se incluyen entre las propuestas específicas de la Tabla 20.

En definitiva, los diferentes parámetros definidos como conclusión del estudio teórico conforman el resumen de una propuesta de caracterización a múltiples niveles a tomar en cuenta en el momento de diseño de los escenarios, pero se han de valorar también

las posibilidades que ofrece el simulador de red para dicha implementación, y el enfoque propio del estudio y sus objetivos.

6.1.1.2. Modelado de Escenarios Wi-Fi con NS-3

El simulador de red NS-3 presenta múltiples clases de C++ que sirven de soporte para la integración de características tecnológicas de comunicaciones. Entre las mismas, se han de analizar las alternativas de implementación de acuerdo con el Mapeado Preliminar de los Resultados del Estudio de Escenarios de Aplicación de la Tecnología IEEE 802.11 a Parámetros de Simulación expuesto en el apartado anterior. De este modo, se persigue introducir los recursos principales proporcionados por el simulador de forma que se guíe la definición concreta de las alternativas reales de implementación.

Una vez seleccionadas las herramientas a utilizar, se podrá recurrir al apartado Metodología para encontrar un desarrollo más detallado y práctico de la selección final de las clases y objetos que a continuación se presentan. Además, en el apartado Anexo III: NS-3 se realiza una ampliación de la información relativa al simulador con enfoque didáctico, para que, de este modo, permita futuras ampliaciones y extensiones del presente estudio.

En este caso, se centra la atención en los recursos que proporciona para el modelado de escenarios Wi-Fi en base a la topología de red, definición espacial del entorno, configuración de nodos y movilidad.

En relación con la **topología de red**, **NodeContainer** es una clase que permite realizar el seguimiento de una serie de punteros a nodos. De esta forma, se trata del soporte fundamental para la definición tanto de APs como STAs –nodos agrupados en función de sus características en común– que componen la red. En relación con los parámetros de la Tabla 17, NodeContainer permite caracterizar el número de equipos por tipo, de forma que se generen objetos separados por tipo de dispositivo. En base a la documentación oficial de NS-3 [88], resulta una práctica habitual definir un NodeContainer para STAs y otro para APs por cada red.

Por otro lado, para cada nodo se definen los dispositivos de red a instalar, de modo que se les dote de la capacidad de comunicarse con otros nodos. De forma similar a la agrupación de nodos, la manera común de proceder es llevar su gestión en contenedores a través de **NetDeviceContainer**. Así, se definen los dispositivos sobre los que se realiza la instalación de la tecnología y se asignan los flujos de tráfico.

En términos espaciales, el simulador NS-3 cuenta con una clase de soporte de movilidad denominada **MobilityHelper** que resulta clave para la caracterización de espacios tridimensionales y de la localización y movilidad de cada nodo en dicho espacio.

De este modo, permite definir el **posicionamiento** de los APs y STAs en base a distintos modelos, tomando como referencia valores fijos o aleatorios sobre planos en forma de disco, rectangulares o en forma de cuadrícula, entre otros.

Una vez establecida la ubicación de los nodos, MobilityHelper también permite modelar la **movilidad** de los estos de forma similar a la definición del posicionamiento, aplicando

modelos como puede ser la localización estática, velocidad o aceleración constantes, o nodos que simulan caminar de forma aleatoria por el plano definido.

Por otro lado, así como se definen los componentes principales de la red en base a clases de soporte que los alojan en contenedores, también se hace uso de otras herramientas para la caracterización de aspectos fundamentales de la tecnología Wi-Fi a **nivel PHY y MAC**.

Para comenzar, la clase **WifiHelper** permite crear **WifiNetDevices** –la implementación de Wi-Fi sobre los dispositivos de red de cada nodo– y configurar gran parte de sus parámetros durante el proceso de creación. Entre ellos, destaca la posibilidad de seleccionar la versión del estándar Wi-Fi a implementar, lo que configura implícitamente los valores por defecto de parte de las características a nivel PHY y MAC.

En concreto, cada capa de abstracción de la red tiene sus propias clases de soporte. A nivel físico, **SpectrumWifiPhyHelper** permite caracterizar los aspectos esenciales de la transmisión, como la **potencia de transmisión**, el **número de antenas** y los **flujos espaciales** a aplicar en las comunicaciones. A diferencia de los otros dos parámetros, la potencia de transmisión puede definirse de forma estática o en base a unos márgenes –valores mínimos y máximos– recorridos en saltos, de modo que la potencia aumenta de forma progresiva.

Sobre el mismo nivel, se pueden realizar modificaciones de los parámetros por defecto para la simulación de escenarios específicos. Entre dichas características, se puede modificar, por ejemplo, el intervalo de guarda entre símbolos, favoreciendo así especificaciones concretas del empleo de **OFDM**.

Relacionado de forma directa con los parámetros anteriores, así como con los dispositivos de red, se hallan los **canales de comunicación**. NS-3 los define en su modelo de abstracción de forma separada a la capa física, pero se comunican y relacionan de forma directa a esta. La clase **SpectrumChannelHelper** facilita la definición de la **banda de frecuencias** en uso, el **tamaño de canal** a emplear, y la **asignación de los canales** –esta ha de modelarse de forma manual, especificando cada canal en uso o mediante programación de métodos de asignación de frecuencias– sobre la banda en uso. Además, permite implementar distintos modelos de canal, haciendo posible la **caracterización de pérdidas** mediante modelos predefinidos –LogDistance, Friis, etc.– o incluso pudiendo modificar sus valores, añadiéndoles datos de referencia para los cálculos a realizar en la propagación de la señal por el canal.

A nivel MAC se dispone de **WifiMacHelper**, que permite implementar el modelo de nivel de enlace específico para STAs o APs, incluyendo parámetros como el **SSID**, el **soporte de QoS**, y el **tráfico de gestión** para la asociación entre estaciones y puntos de acceso. Además, contempla la utilización de **EDCA**, dando posibilidad de especificar sus valores de referencia, y también permite asignar **tamaños de agregación** en función de las clases de tráfico.

Asimismo, dentro de WifiMac se incluyen también se incluyen otras características de bajo nivel, que permiten, entre otros, la definición de los umbrales de **RTS/CTS** en caso

de habilitar dicho mecanismo, los algoritmos de gestión de ACKs, e incluso los valores de parámetros como **TXOP**, **PIFS** o **SIFS**.

Una vez especificada la implementación de la topología y las características a nivel de físico y de enlace, se ha de definir la configuración de cada nodo para posibilitar las comunicaciones. Para llevar a cabo dicha tarea se dispone de **InternetStackHelper**, la clase dedicada a la instalación de la pila de protocolos de internet.

Para posibilitar la comunicación a niveles superiores a los anteriormente mencionados también se dispone de clases de soporte adicionales.

A nivel de red, **Ipv4AddressHelper** permite realizar asignaciones de direcciones IP sobre los nodos de forma sencilla.

La utilización de los distintos protocolos a nivel de transporte viene condicionada por las aplicaciones utilizadas en la generación de tráfico en base a distintos modelos. **ApplicationContainer** permite realizar el seguimiento de este tipo de aplicaciones, cuya implementación se realiza mediante los siguientes módulos disponibles: BulkSendApplication, OnOffApplication, PacketSink, ThreeGppHttpClientServer, UdpClientServer y UdpEcho.

BulkSendApplication se fundamenta en el envío simple de datos a la máxima velocidad definida hasta alcanzar un número máximo de bytes o hasta que la aplicación es detenida. Cuando se llena el buffer de envío de paquetes, se espera hasta que se dispone de espacio para nuevos envíos, tratando de mantener un flujo constante de datos.

OnOffApplication sigue un patrón de generación de tráfico On-Off, que en esencia supone alternar entre dos estados donde uno simula el caso de transmisión de tráfico y el segundo define el tiempo en el que un dispositivo no está transmitiendo. El tráfico generado se basa en la tasa de datos y el tamaño de paquete definidos.

PacketSink, a diferencia de las dos anteriores, es una aplicación que simula la recepción de tráfico, y aunque resulta de especial interés en la transmisión multicast porque hace posible la recepción de tramas multicast a nivel de enlace y posibilita su análisis, también sirve para transmisiones unicast como ejemplo de cómo se reciben los paquetes a nivel de aplicación.

ThreeGppHttpClientServer es un módulo dedicado a la simulación de tráfico web basado en el protocolo HTTP (Hypertext Transfer Protocol). Consiste en una o más aplicaciones cliente que se conectan a una aplicación servidor, de forma que el cliente realiza peticiones de objetos, y dependiendo del tipo de solicitud, el servidor responde con objetos principales o embebidos para simular la descarga de una página web.

UdpClientServer cubre las aplicaciones de cliente y servidor UDP, de forma que se simula el tráfico a nivel de transporte donde el cliente envía paquetes UDP con número de secuencia y marca de tiempos en sus datos, y el servidor los recibe. Permite analizar

las pérdidas de paquetes y el retardo en base a la información que transporta y los paquetes enviados y recibidos.

UdpEcho simula el envío de ida y vuelta de paquetes UDP entre cliente y servidor, de forma que el cliente envía un paquete que espera recibir de vuelta, y el servidor realiza el procedimiento opuesto.

Las aplicaciones anteriores ofrecen una variedad considerable de posibilidades de implementación, pero también se ha de valorar como alternativa la definición propia de implementaciones de tráfico e incluso la utilización o adaptación de librerías externas, que pueden aportar un alto valor añadido al presente trabajo.

6.1.1.3. Definición de las Alternativas de Implementación de Parámetros de Simulación WLAN

En base a los dos apartados anteriores, a continuación, se resumen las alternativas de implementación de los parámetros de simulación WLAN. De este modo, se vinculan las conclusiones teóricas de la revisión realizada en el Estado del Arte a las posibilidades de implementación reales, obteniendo las alternativas definitivas a analizar con la misma fórmula de separación de características en tres grandes grupos.

El primero está dedicado a la parametrización de la **topología y definición espacial** del entorno. Dentro de esta, inicialmente se caracterizan los aspectos físicos del escenario. Para ello, se contempla la definición de la **forma** del plano según las diferentes opciones presentes en NS-3 a través de la configuración de MobilityHelper. A través de dicha clase de soporte también se establecen diferentes niveles de **variabilidad de la dimensión** de los entornos partiendo de las conclusiones del estudio teórico, así como los valores de referencia del **área** del plano.

Aspecto bajo análisis	Parámetro característico	Alternativas de implementación
Definición física del escenario	Forma (MobilityHelper)	Rectangular Circular Por cuadrículas
	Variabilidad de la dimensión del escenario (MobilityHelper)	Baja (hasta 40m ²) Intermedia (hasta 100m ²) Alta (más de 100m ²)
	Área (MobilityHelper)	Pequeña (hasta 50m ²) Mediana (50m ² – 100m ²) Grande (a partir de 100m ²)

Tabla 21: Alternativas espaciales a seleccionar por escenario.

Después, se definen las características específicas de los nodos. En el caso de los APs, se define como alternativa a seleccionar el **modelo de definición del número de APs por red**, que podrá ser determinista –un valor fijo definido– o aleatorio –partiendo de una semilla aleatoria que definirá el valor en cada caso partiendo de una función de distribución de probabilidad–. Por consiguiente, resultará necesario contemplar la **densidad de APs por red**, para lo que se han definido tres niveles en base al estudio teórico que comprenderán un número concreto en el caso del modelo determinista, o

un rango en el caso de aplicar modelos estadísticos en los datos que se utilizarán sobre NodeContainer.

Según la abstracción de NS-3, se pueden definir diferentes **interfaces de red** por nodo apoyándose en NetDeviceContainer. Por lo tanto, se ha de contemplar como alternativa el uso de una única interfaz de red por AP o más.

Además, se deben definir las características de posicionamiento y movilidad de los APs. Se establece como alternativa la selección del **tipo de posicionamiento** en base a modelos deterministas, uniformes o aleatorios, que son las opciones disponibles en la clase MobilityHelper.

En términos de **altura**, se ha de seleccionar si se concreta de forma determinista o aleatoria, tomando como referencia los valores del estudio teórico de escenarios e introduciendo dicha información en el objeto de la clase MobilityHelper.

Por último, apoyándose en lo anterior, se debe establecer si los APs disponen de **movilidad**, y qué modelo siguen en caso de haberla. MobilityHelper ofrece cuatro opciones distintivas: sin movilidad, velocidad constante, aceleración constante y Random Walk.

Aspecto bajo análisis	Parámetro característico	Alternativas de implementación
Características de los APs	Modelo de definición del número de APs por red (NodeContainer)	Determinista Aleatorio
	Densidad de APs por red (NodeContainer)	Baja (1 – 2) Media (2 – 10) Alta (10 – 20)
	Interfaces de Red por AP (NetDeviceContainer)	Única (1) Múltiples (2, 3, 4)
	Tipo de posicionamiento de APs (MobilityHelper)	Determinista Uniforme Aleatorio
	Modelo de altura de los APs (MobilityHelper)	Determinista Aleatorio
	Modelo de movilidad de los APs (MobilityHelper)	Estático (sin movilidad) Velocidad constante Aceleración constante Random Walk

Tabla 22: Alternativas topológicas y espaciales de los APs.

La definición de **alternativas topológicas y espaciales de las STAs** siguen la misma estructura que los APs, pero varían en los valores aplicables.

Dichos valores parten de las conclusiones del estudio de escenarios en el que se aprecian diferencias significativas, como en el caso del número de APs y STAs presentes en cada red.

Aspecto bajo análisis	Parámetro característico	Alternativas de implementación
Características de las STAs	Modelo de definición del número de STAs por red (NodeContainer)	Determinista Aleatorio
	Densidad de STAs por red (NodeContainer)	Baja (0 – 10) Media (10 – 30) Alta (30 – 50)
	Interfaces de Red por STA (NetDeviceContainer)	Única (1) Múltiples (2)
	Tipo de posicionamiento de STAs (MobilityHelper)	Determinista Uniforme Aleatorio
	Modelo de altura de las STAs (MobilityHelper)	Determinista Aleatorio
	Modelo de movilidad de las STAs (MobilityHelper)	Estático (sin movilidad) Velocidad constante Aceleración constante Random Walk

Tabla 23: Alternativas topológicas y espaciales de las STAs.

Las **características específicas de configuración de red** conforman los múltiples parámetros a simular sobre las configuraciones tecnológicas, incluyendo el nivel físico y de enlace.

Primero, la Tabla 24 define la configuración frecuencial. En ella, se define la **banda de frecuencias** a utilizar, precisando como alternativas las tres bandas características de la tecnología IEEE 802.11ax. Además, se condensan las posibilidades de **configuración de canales** en base a tres técnicas, la selección aleatoria, la compartición de canal y la asignación determinista de los canales. Asimismo, los **anchos** contemplados se corresponden con las posibilidades ofrecidas entre todas las versiones y bandas de Wi-Fi, y es dependiente de ambas.

Parámetro característico	Alternativas de implementación
Selección de la banda frecuencial (SpectrumChannelHelper)	2,4 GHz 5 GHz 6 GHz
Modelo de configuración de canales (SpectrumChannelHelper)	Selección aleatoria Compartición de canal Asignación determinista
Ancho de canal (SpectrumChannelHelper)	20 MHz 40 MHz 80 MHz 160 MHz

Tabla 24: Alternativas de configuración frecuencial.

Después, la Tabla 25 recoge las **configuraciones de antenas en términos de cantidad y potencia** disponibles haciendo uso de SpectrumWifiPhyHelper. Por un lado, se definen las **configuraciones de antenas** extraídas del Estado del Arte tanto para APs como STAs, incluyendo el uso de **MIMO** de forma implícita para configuraciones de múltiples antenas por nodo. Se definen por separado el número de

antenas y de flujos debido a que la disponibilidad de las primeras no implica su utilización conjunta por defecto.

Asimismo, se proponen distintos **valores de potencia** para APs y STAs, así como **modelos de potencia** fijos –donde la potencia adquiere un valor constante a lo largo de la simulación– y variables o de forma gradual.

Parámetro característico	Alternativas de implementación
Configuración de antenas de los AP (SpectrumWifiPhyHelper)	1x1 / 2x2 / 4x4 / 8x8
Configuración de antenas de las STA (SpectrumWifiPhyHelper)	1x1 / 2x2 / 3x3
Número de flujos espaciales AP (SpectrumWifiPhyHelper)	Único (1) Múltiple con bajo valor (2) Múltiples con valor intermedio (4) Múltiple con alto valor (8)
Número de flujos espaciales STA (SpectrumWifiPhyHelper)	Único (1) Múltiple con bajo valor (2) Múltiples con valor intermedio (3) Múltiple con alto valor (4)
Modelo de potencia TX APs (SpectrumWifiPhyHelper)	Estático Variable
Potencia TX APs (SpectrumWifiPhyHelper)	Baja (0dBm -10dBm) Media (10dBm – 20 dBm) Alta (20 dBm – 30 dBm)
Modelo de potencia TX STAs (SpectrumWifiPhyHelper)	Estático Variable
Potencia TX STAs (SpectrumWifiPhyHelper)	Baja (0dBm -10dBm) Media (10dBm – 20 dBm) Alta (20 dBm – 30 dBm)

Tabla 25: Alternativas de configuración de la transmisión.

Por otro lado, se presentan como alternativas los **modelos de propagación** disponibles en el simulador y contemplados en el estudio teórico. La descripción breve de los modelos propuestos se puede observar en la Tabla 26, donde se toma como referencia la clase SpectrumWifiPhyHelper.

Una vez especificadas las configuraciones principales del espectro, en la Tabla 27 se pueden encontrar las alternativas de parámetros relacionados al **nivel de enlace y físico** disponibles en las últimas versiones del protocolo IEEE 802.11.

Inicialmente se definen la posibilidad de utilizar **EDCA** y la definición de sus **valores principales** en caso de que la primera alternativa sea afirmativa. A continuación, se definen las alternativas sobre uso de **agregación** y el **tamaño de las agrupaciones**.

Después, se configuran valores específicos con relación al número de **repeticiones** máximo, el **intervalo de guarda** entre símbolos, los **espacios entre tramas** (Interframe Space, IFS), y el uso de **RTS/CTS** junto con su umbral de aplicación.

Parámetro característico	Alternativas de implementación	Descripción breve
Modelo de propagación (SpectrumWifi PhyHelper)	Introducción fija	Valores predefinidos de pérdidas y delay
	Introducción aleatoria	Valores aleatorios de pérdidas y delay
	LogDistance	Pérdidas logarítmicas en función de la distancia
	Three LogDistance	Pérdidas logarítmicas en función de la distancia en base a tres distancias
	Friis	Modelo Friis para las pérdidas de propagación.
	ConstantSpeed	Modelo de propagación de la señal a velocidad constante
	ITU-R 1411 LOS (outdoor)	Modelo para comunicaciones outdoor de bajo rango para frecuencias entre 300 MHz y 100 GHz
	Nakagami	Modelo de pérdidas Nakagami para fast fading.
	3GPP Indoor Mixed Office	Modelo de pérdidas para oficinas mixtas entre 0.5 GHz y 100 GHz
	3GPP Indoor Office	Modelo de pérdidas para oficinas abiertas entre 0.5 GHz y 100 GHz
	3GPP Uma	Modelo de pérdidas para entornos rurales macrocelulares.
	3GPP Umi	Modelo de pérdidas para entornos urbanos microcelulares.
	3GPP RMa	Modelo de pérdidas para entornos urbanos macrocelulares.
	Modelo mixto	Combinación de los modelos predefinidos con pérdidas adicionales para representación de entornos específicos (Indoor Hotspot, (N)LOS Indoor, TGax TMB)

Tabla 26: Alternativas de modelos de propagación.

Parámetro característico	Alternativas de implementación
EDCA (WifiMacHelper)	Sí No
Definición de valores EDCA (WifiMacHelper)	Estándar Específicos
Agregación (WifiMacHelper)	A-MPDU A-MSDU
Tamaño de agregación (WifiMacHelper)	64 Máx
Repeticiones (retries)	Estándar Específicos
GI	800ns 1600ns 3200ns
IFS	Estándar Específicos
RTS/CTS	Sí No
Umbral RTS/CTS	Estándar Específicos

Tabla 27: Alternativas de configuración a nivel de enlace y físico.

Por último, en cuanto al **tráfico de red**, las alternativas han de seleccionarse mediante un proceso de dos fases. En la primera fase, se han de definir los **tipos de tráfico** a simular por cada tipo de escenario. En la segunda, una vez establecida la selección anterior, se han de especificar la **cantidad de dispositivos** afectados **por cada flujo** – tanto de subida como de bajada– para cada tipo de tráfico y entorno.

Parámetro característico	Alternativas de implementación
Tipo de tráfico	Tráfico de gestión, VoD, videoconferencia, streaming multicast, VDI, HTTP, FTP, VoIP, gaming, VR/AR, IoT
Tasa de flujos STA → AP por tipo de tráfico	% de STAs / % de APs
Flujos AP → STA por tipo de tráfico	% de APs / % de STAs

Tabla 28: Alternativas de tráfico de red.

6.1.2. Indicadores de Rendimiento Clave

En el apartado Parámetros y requisitos mínimos de las clases de QoS se introducían los **parámetros principales de QoS**, cuyo enfoque genérico destacaba la medida del retardo, la variación de retardo, la pérdida de paquetes, la tasa de error en paquetes, el ancho de banda y el throughput.

Sin embargo, posteriormente, en la Clasificación de Indicadores Clave de Rendimiento, se hacía una primera distinción sobre los tipos de parámetros para la medida de calidad, incluyendo, a parte de los anteriores, KPIs transversales y no técnicos (Tabla 6).

Asimismo, se hacía hincapié en los **servicios de datos** habituales, presentando la relevancia de la medida de algunas métricas en la definición de los indicadores clave de rendimiento para cada servicio (Tabla 7) específico.

El presente estudio se fundamenta en la realización de simulaciones con métricas cuantificables directamente extraídas del entorno, por lo que se centra la atención en los **KPIs técnicos** asociados a los servicios a implementar. De este modo, a continuación, se hace una breve introducción a los recursos y posibilidades proporcionados por el simulador en la medida de la QoS, y una vez contemplado el potencial de este, se definen las posibles alternativas específicas a implementar como parámetros de calidad.

6.1.2.1. Medida de Métricas de Calidad en Escenarios Wi-Fi con NS-3

NS-3 hace posible la medida de la calidad de servicio mediante la provisión de herramientas y métricas que permiten evaluar los aspectos principales contemplados en el Estado del Arte.

De forma general, al disponer de acceso a todos los módulos relacionados con la simulación de escenarios Wi-Fi se puede hacer uso de **callbacks**, que son métodos que facilitan el manejo de eventos específicos ocurridos durante el proceso de simulación. Gracias a ellos, se puede llevar a cabo la recopilación de datos y estadísticas, haciendo posible su análisis posterior.

MonitorSnifferRx es un método de NS-3 ampliamente empleado en la captura y análisis de paquetes sobre un nodo concreto, y es un ejemplo de uso de callbacks. Para poder realizar el registro de información, requiere la configuración previa del dispositivo de red sobre el que se va a realizar la monitorización y la asociación de la instancia del monitor al dispositivo.

Una vez establecido, proporciona parámetros fundamentales como el contenido de los paquetes capturados, las direcciones IP origen y destino, los protocolos empleados, el número y tamaño de los paquetes, etc. Del mismo modo, también permite la extracción de métricas con relación al canal y la frecuencia utilizadas, el tiempo de actividad, o la potencia de transmisión. En consecuencia, permite su posterior procesamiento para el cálculo de métricas de rendimiento como el throughput, la latencia o la tasa de pérdida de paquetes, e incluso medidas a nivel espectral para poder obtener información relativa a los niveles de señal y ruido.

Sin embargo, a nivel de tráfico de red, la herramienta principal que proporciona NS-3 es **FlowMonitor**, cuyo enfoque se centra en la monitorización y análisis de flujos de tráfico. Su funcionamiento se basa en la recogida de información de los diferentes flujos de paquetes durante las simulaciones y la clasificación de estos en función de tuplas compuestas por las direcciones IP origen y destino, los puertos origen y destino y el protocolo de transporte utilizado.

Mediante la identificación de flujos, permite la extracción de datos de los paquetes, como las métricas relativas al número de paquetes y bytes enviados y recibidos por cada dispositivo en los flujos, la duración de las transmisiones y recepciones o métricas sobre los retardos y sus variaciones a nivel de paquete.

Por lo tanto, el cruce de las dos herramientas anteriores y su empleo para obtener parámetros de medida de calidad desde distintas perspectivas posibilita múltiples planteamientos para la definición de KPIs.

En resumen, procedimiento para la medida de los parámetros de calidad requiere, primero, la **utilización de las herramientas de NS-3** para la extracción de datos concretos—como el número de paquetes y bytes transmitidos y recibidos por dispositivo o flujo, tiempos de envío y recepción de paquetes, la latencia o el jitter— y segundo, el **procesado de dichos datos mediante operaciones de cálculo** para la obtención de valores medios, o métricas más elaboradas— como el tiempo medio de transmisión, la estimación del nivel SNR en un dispositivo, el valor medio de la variación de jitter, o el throughput total experimentado durante un tiempo concreto. De este modo, se abre la posibilidad de medir parámetros de forma adaptada a cada caso específico.

6.1.2.2. Alternativas de Indicadores de Rendimiento Clave Específicas

Una vez presentadas las posibilidades de obtención de indicadores en el apartado anterior, y recogiendo las recomendaciones generales de los Criterios de QoS, a continuación, se presentan de forma resumida las métricas a definir como alternativas de indicadores de rendimiento clave para los casos de uso propuestos.

La libertad de definición de parámetros en NS-3 permite la introducción de **múltiples posibilidades para el cálculo** de métricas, ya sea de forma directa o mediante el procesamiento estadístico de las mismas.

Debido al ámbito del estudio –entornos de red Wi-Fi 6 y extracción del contexto del usuario– se recogen en la Tabla 29 los posibles parámetros de medida de calidad a implementar en NS-3 para la realización de las simulaciones.

Alternativa de KPI	Descripción breve
Tiempo de funcionamiento	Tiempo de disponibilidad del servicio en segundos
Bandwidth	Capacidad de transmisión en Mbps
Throughput	Tasa de transmisión en Mbps
Número de paquetes / bytes transmitidos	Número de paquetes / bytes transmitidos
Número de paquetes / bytes recibidos	Número de paquetes / bytes recibidos
Retardo	Tiempo de retardo en milisegundos
Jitter	Tiempo de variación del retardo en milisegundos
Tasa de pérdida de paquetes	Porcentaje de pérdida de paquetes
Tasa de errores en paquetes	Porcentaje de paquetes recibidos con error
SNR	Diferencia señal a ruido en dB

Tabla 29: Alternativas de Métricas en NS-3.

Las propuestas se extraen en su mayoría de las especificaciones del manual de la IEEE sobre QoS [32] para los servicios de red bajo análisis. No obstante, se complementan con los **parámetros básicos** accesibles mediante NS-3 –número de paquetes/bytes transmitidos, número de paquetes/bytes recibidos– para su cálculo, y se incluyen también aspectos de medida de señal debido a su relevancia en la eficacia de la tecnología [89][90].

Perspectiva	Descripción	Alternativas
Ámbito de aplicación	Uso de valores específicos por dispositivo o utilización de valores globales de la red	Valor individual, medio o total
	Aplicación del parámetro en estudio sobre cada tipo de equipo	AP, STA o ambos
Tiempo de análisis	Tiempo tomado como referencia para la toma de medidas del parámetro en estudio.	Medida instantánea, duración total de la simulación o duración de flujo

Tabla 30: Alternativas de interpretación de las medidas.

Por último, se ha de determinar el planteamiento específico sobre cada medida (Tabla 30), ya que los parámetros obtenidos pueden ser analizados según diferentes

perspectivas. En términos del **enfoque de la métrica**, se puede adquirir un punto de vista concreto sobre el dispositivo analizado –ya sea un AP o STA– o una perspectiva global de la red. Asimismo, en caso de obtener parámetros extraídos de los nodos, se debe distinguir de qué tipo de dispositivo se trata, ya que la interpretación de la métrica puede verse afectada en función de si se toma como referencia una estación o un punto de acceso.

Del mismo modo, se pueden extraer KPI en función de diferentes valores calculados sobre los parámetros extraídos, como cada valor individual, el valor medio o la suma total de las muestras tomadas.

Además, se ha de valorar el **tiempo de análisis** a tomar como referencia para la evaluación de los indicadores en cada caso, ya sea una medida instantánea, el total de la duración de la simulación o la duración del flujo.

6.1.3. Algoritmos y Técnicas de Clustering

En la sección dedicada a los Algoritmos de Clustering se adelantaba la existencia de una variedad de categorías de agrupamiento basada en el aspecto fundamental empleado para la medida de similitud entre muestras y formación de grupos.

En consecuencia, en el presente apartado se describen con mayor detalle el funcionamiento de los algoritmos principales de los enfoques más utilizados [82], facilitando así la selección de algoritmos a implementar sobre los resultados de simulación.

6.1.3.1. Clustering particional

Dentro del enfoque del agrupamiento particional, que realiza el proceso de organización mediante la separación de muestras en grupos que no se basan en jerarquías, se consideran los algoritmos K-Means y K-Medoids.

K-Means es un algoritmo basado en la distancia que realiza la agrupación de n observaciones en k clústeres, evaluando la pertenencia de cada muestra al grupo más cercano. Se trata de una técnica basada en centroides, que se corresponden con los centros de gravedad de cada clúster, y en este caso concreto representa el valor medio de los puntos que forman el clúster.

El funcionamiento de K-Means se basa en la iteración sobre la asignación de los objetos a los clústeres en base a la distancia euclídea al centroide más cercano, de forma que las diferencias dentro de cada grupo formado son reducidas en cada ejecución del algoritmo. De este modo, el centroide es actualizado repetidas veces hasta que se alcanza un estado de convergencia, es decir, cuando la asignación de muestras a clústeres se vuelve estables, manteniendo la misma asignación entre iteraciones.

En concreto, siguiendo la Ecuación 8, el algoritmo divide un conjunto de N muestras X en un total de k clústeres C , descritos por la media de muestras μ_j en el clúster.

$$\sum_{i=0}^n \min_{\mu_j \in C} (\|x_i - \mu_j\|^2)$$

Ecuación 8: Inercia o criterio de la suma de los cuadrados entre clústeres. [91]

Cabe considerar que la influencia de la posición inicial de los centroides resulta decisiva para la posición final de estos, y que se requiere el conocimiento previo del número de clústeres a obtener. En general, [91] comprende un algoritmo de propósito general escalable en número de muestras que requiere el conocimiento previo del número total de clústeres, y resulta adecuado para clústeres de tamaño equilibrado y geometría plana, para un número no excesivamente alto de grupos.

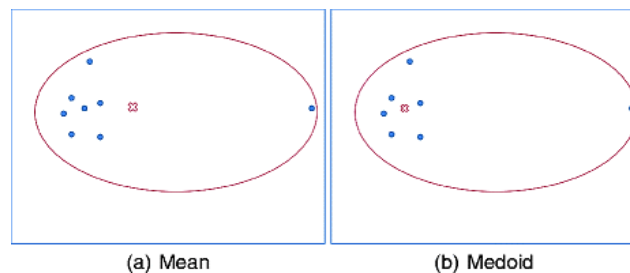


Figura 27: Comparación del agrupamiento parcial mediante K-Means y K-Medoids en presencia de muestras aisladas.

El algoritmo **K-Medoids** [79], por otro lado, es una variante de K-Means que trata de proveer robustez al ruido y casos aislados de pertenencia. La diferencia principal está en que, en lugar de tomar como centro del clúster el valor medio directamente, utiliza una muestra actual del clúster como centro, denominado medoide. Este es el objeto ubicado en la posición más céntrica del grupo, es decir, con la mínima suma de distancias a las demás muestras del clúster.

De forma similar a K-Means, se basa en la inicialización mediante la selección de k muestras aleatorias como medoides iniciales – equivalentes a los centroides – y el procedimiento iterativo de asignación de objetos a los clústeres en base a la distancia al medoide y actualización de centros en base a la suma de distancias de cada clúster formado hasta la convergencia.

6.1.3.2. Clustering basado en densidad

Respecto al agrupamiento basado en densidad, que toma como eje de desarrollo la concentración de muestras del set de datos en espacios reducidos, se contemplan los algoritmos DBSCAN, OPTICS y Mean-shift.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [77] es un algoritmo de agrupamiento que realiza la configuración de clústeres a través de la estimación de la distribución de densidad de las muestras del conjunto de datos.

A diferencia de los algoritmos presentados anteriormente, no parte del conocimiento del número de clústeres, sino que se basa en la definición de dos parámetros ϵ y n , el umbral de distancia y el umbral de densidad, respectivamente. Asimismo, su

funcionamiento resulta de mayor complejidad a las técnicas presentadas hasta el momento.

En un inicio, se selecciona una muestra x del set de datos y se asigna al primer clúster. Acto seguido, se mide la cantidad de muestras cuya distancia a x es inferior o igual a ϵ . Si dicha cuantía es igual o mayor al umbral de densidad, se asocian esa cantidad de muestras al mismo clúster, y se les denomina puntos del núcleo.

Una vez se cumple dicha condición, se examina cada miembro del grupo y se estudian sus vecinos del mismo modo en el que se procedió para la muestra x . Si algún participante del grupo tiene n o más vecinos, se expande el clúster original incluyendo a dichos vecinos, y se itera de este modo hasta que no queden datos que introducir bajo las condiciones de los umbrales de distancia y densidad.

Cuando se finaliza la formación de un grupo se procede a repetir el método anterior con una muestra distinta no perteneciente a ningún clúster. Dicho procedimiento se repite hasta alcanzar una condición de finalización, ya sea que todas las muestras pertenezcan a un clúster o una condición específica. Los datos excluidos son considerados ruido.

De forma global, [91] resulta un algoritmo escalable en número de muestras y a un número considerable de clústeres, que permite tamaños desiguales entre los grupos a formar y facilita el descarte de datos muy lejanos a los grupos. Requiere disponer de nociones sobre el tamaño de los vecindarios (grupos de puntos cercanos) para el establecimiento de los umbrales como parámetros de entrada.

OPTICS (Ordering Points To Identify the Clustering Structure) [91] es un algoritmo que guarda gran similitud con DBSCAN, y puede considerarse una generalización de este donde se relaja la condición del umbral de distancia a un rango de valores en lugar de uno único.

A diferencia de DBSCAN, proporciona una representación visual de la estructura de clustering en forma de gráfico de alcance (reachability graph), que asigna cada muestra a dos atributos: el primero, que se corresponde a la distancia de alcance, mide la conectividad local y la densidad del conjunto de datos; el segundo, denominado orden, asocia la muestra a un punto en el ordenamiento lineal del clúster (desde la distancia de alcance más pequeña a la más grande).

Una vez realizados el cálculo de la distancia y el ordenamiento, se realiza el gráfico de alcance partiendo de este último, que representa la relación de alcance entre vecinos. Los puntos de inflexión en el gráfico permiten la identificación de clústeres.

En general, es una técnica altamente escalable [91] en términos de muestras y muy escalable en número de clústeres a formar, que permite trabajar con geometrías no uniformes, tamaños de clúster distintos entre sí, diferentes densidades, y posibilitando la eliminación de datos muy lejanos a los grupos. Requiere disponer de información sobre los mínimos de membresía del clúster.

Mean-shift [91] es un algoritmo basado en densidad que, como su nombre indica, se basa en el concepto de desplazamiento de la media. De forma sencilla, se fundamenta en la actualización de candidatos a centroides sobre un conjunto de muestras que sean la media de los puntos en una región dada. Estos candidatos se filtran posteriormente en una fase de procesado para eliminar duplicados cercanos que formen el conjunto final de centroides.

La posición de los candidatos se ajusta iterativamente utilizando una técnica denominada hill climbing, que busca el máximo local de la densidad de probabilidad estimada. La Ecuación 9 describe el proceso de actualización de un candidato a centroide x en una iteración t .

$$x^{t+1} = x^t + m(x^t)$$

Ecuación 9: Actualización de centroides en Mean-shift. [91]

Asimismo, m es el vector mean-shift que se calcula para cada centroide que apunta hacia una región del máximo crecimiento en densidad de puntos. Para calcular el valor de m , se define $N(x)$ como el conjunto de muestras vecinas en una distancia dada alrededor de x .

$$m(x) = \frac{\sum_{x_j \in N(x)} K(x_j - x)x_j}{\sum_{x_j \in N(x)} K(x_j - x)} - x$$

Ecuación 10: Ecuación general para el cálculo del vector mean-shift.

La ecuación general para el cálculo de m depende de un kernel (núcleo) utilizado para la estimación de densidad, es decir, mide la densidad de puntos en función de su proximidad.

Resulta un algoritmo no escalable [91] en número de muestras, que se puede utilizar para un alto número de clústeres de tamaño variable y geometría no uniforme. Requiere el conocimiento del ancho de banda, parámetro que define el tamaño de la región espacial sobre la que realizar búsqueda de clústeres.

6.1.3.3. Clustering basado en la distribución

En relación con el agrupamiento basado en la distribución, que parte de aproximaciones muy cercanas al análisis estadístico, se estudian DBCLASD y GMM.

DBCLASD (Distribution Based Clustering of Large Spatial Databases) [81] es un algoritmo incremental y dinámico, es decir, en el que la asignación de puntos se basa únicamente en los puntos procesados sin considerar el clúster completo o el set de datos en su conjunto. De este modo, se trata de agrupar las muestras en función de la proximidad y distribución en el espacio.

DBCLASD itera aumentando el clúster inicial de forma incremental mediante sus puntos vecinos, siempre que la distancia al vecino más cercano del clúster resultante satisfaga la distribución de distancia esperada. Por tanto, establece la generación de candidatos

con cada nuevo miembro del clúster mediante la evaluación de los vecinos de este último a una distancia inferior definida, y añadiéndolos como nuevos candidatos.

Esto supone la construcción de un grafo de vecindad, y da lugar a la estimación posterior de la densidad en base a los vecinos cercanos y la formación de grupos basada en dicho parámetro.

En general, como su nombre indica, destaca por la eficiencia y escalabilidad ante grandes conjuntos de datos, y la posibilidad de detectar estructuras no uniformes debido al procedimiento incremental de formación de clústeres.

GMM (Gaussian Mixture Models) [91] [92] comprende una serie de modelos probabilísticos que asumen que todas las muestras a analizar son generadas mediante la mezcla de un número finito de distribuciones gaussianas con parámetros desconocidos.

GMM requiere conocer el dato de entrada del número de componentes gaussianas presentes de forma latente en el conjunto de datos. Una vez seleccionada dicha premisa, se inicializan los parámetros –media, matriz de covarianza, pesos, etc.– del modelo estadístico para cada componente, que también requieren ser definidos.

Un algoritmo de uso habitual en GMM es EM (Expectation-Maximization), que permite dar solución al problema del desconocimiento de qué puntos del set de datos vienen de qué componente gaussiana o grupo. De esta forma, se trata de estimar parámetros como la media, la matriz de covarianza o los pesos de cada componente gaussiana.

El procedimiento resulta similar a K-Means, ya que se realizan estimaciones iniciales para la localización y forma de los grupos, y después se realiza un proceso de repetición hasta la convergencia en un procedimiento basado en dos pasos. El primero, E (Expectation), asigna los pesos para representar la probabilidad de pertenencia de las muestras a los grupos, y el segundo, M (Maximization), actualiza la localización, normalización y forma en cada clúster basándose en los pesos de todos los puntos.

De modo global, [91] se trata de un algoritmo no escalable que requiere de múltiples parámetros a definir de forma previa a su utilización. Sin embargo, resulta adecuado para la estimación de densidad y geometrías planas.

6.1.3.4. Clustering jerárquico

Por último, entre los algoritmos de agrupamiento jerárquico, basados en la conectividad, se analizan BIRCH, CURE y ROCK.

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) [91] es un algoritmo basado en jerarquías que permite el manejo de grandes grupos de datos mediante una única lectura del set de muestras.

El algoritmo se basa en una estructura de árbol CFT (Clustering Feature Tree) para representar y resumir los datos de forma compacta. Para ello, se definen nodos CF en los que se comprime la información con pérdidas. Los nodos tienen un número de

subgrupos denominados subclústeres CF, y los subclústeres ubicados en los nodos no terminales pueden tener, a su vez, nodos CF hijo.

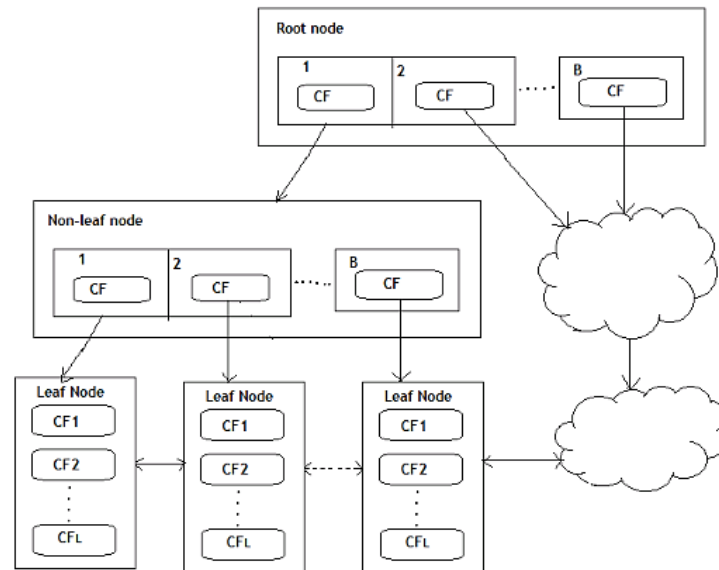


Figura 28: Ejemplo de CFT. [93]

Los subclústeres contienen la información necesaria –número de muestras en un subclúster, suma lineal de las muestras, suma de la norma euclidiana al cuadrado de las muestras, los centroides, y la norma al cuadrado de los centroides– para el agrupamiento, lo que evita tener que disponer de toda la información inicial en memoria.

El algoritmo dispone de dos parámetros principales, el umbral y el factor de ramificación. El umbral define el número máximo de muestras a considerar en un subclúster, y el factor de ramificación limita la distancia entre la muestra de entrada y los subclústeres existentes.

Por otro lado, **CURE** (Clustering Using Representatives) [94] es un algoritmo jerárquico que se basa en la distancia entre puntos, y abarca un ámbito intermedio entre los métodos basados en centroides y el enfoque de puntos extremos, que trata de seleccionar puntos representativos máximos o mínimos para formar los clústeres. La estrategia empleada es el muestreo aleatorio para seleccionar un conjunto de puntos representativos como muestras iniciales.

Su funcionamiento de forma simplificada parte de la toma de dichas muestras para realizar un proceso de compresión que reduce la complejidad del conjunto de datos y asigna representantes (centroides para la agrupación inicial) a cada punto comprimido. Una vez realizado un primer proceso de formación de grupos, se utilizan técnicas basadas en distancias para recalcularlos centroides de forma iterativa. Resulta adecuado para clustering a gran escala.

En resumen, se trata de un algoritmo adecuado para trabajar con clústeres de diferente forma y tamaño –incluyendo variabilidad entre los clústeres para un mismo estudio de

agrupamiento– y cuyo funcionamiento es escalable a largos sets de datos sin sacrificar la calidad de los grupos a formar.

En tercer lugar, **ROCK** (RObust Clustering using linKs) [81] [95] es un algoritmo aglomerativo jerárquico que presenta algunas mejoras respecto a CURE para trabajar con tipos de datos enumerados. Para ello, hace uso de los aspectos de similitud con los datos alrededor del clúster de referencia.

Fundamentalmente, hace uso de la relación de proximidad entre puntos, utilizando el coeficiente Jaccard [79] para la definición de enlaces entre los objetos de datos. Una vez construidas las vecindades de cada objeto y las vecindades compartidas (vecinos cercanos en común) se genera un gráfico con los enlaces para convertirlo en un problema de agrupamiento gráfico.

Resulta un algoritmo que, de forma general, muestra buenos resultados para conjuntos de datos grandes, y permite una buena caracterización para clústeres desiguales en tamaño y forma. Además, recoge en gran medida las bondades de CURE. [81] Sin embargo, se dispone de sensibilidad a los parámetros de entrada definidos, y complejidad computacional derivada de la construcción de grafos de vecindad y cálculo de densidades.

6.2. Criterios de Selección

La metodología seguida para la definición de los criterios de selección se fundamenta en el establecimiento de criterios específicos en función de la problemática, asignando valores enteros entre 0 y 5 sobre cada aspecto a contemplar. Cada uno de los criterios desarrollados dispone de un peso porcentual que se corresponde con la relevancia de este sobre la puntuación final de la alternativa en estudio.

6.2.1. Criterios de Selección para la Parametrización de los entornos WLAN

De cara a tomar las decisiones de selección sobre las alternativas de parametrización especificadas en la Definición de las Alternativas de Implementación de Parámetros de Simulación WLAN, a continuación, se describen de forma ordenada–de mayor a menor peso en términos de la ponderación– los criterios de selección que guían las simulaciones Wi-Fi de acuerdo con los objetivos del proyecto:

1. **Referencia del estudio teórico:** la extensa revisión documental expuesta en el Estado del Arte recoge los casos de uso más habituales en los escenarios de interés para la tecnología bajo estudio. En consecuencia, resulta de vital importancia poner en valor los resultados de dicho estudio a la hora de guiar las simulaciones, por lo que se asigna un valor del 30% de la nota otorgada a cada alternativa de parametrización.
2. **Ajuste a la actualidad de la tecnología:** debido al carácter innovador del proyecto, un factor clave a la hora de decidir sobre la aplicación de configuraciones es la representación de estas de acuerdo con las últimas versiones del estándar. De esta forma, se favorece la obtención de simulaciones representativas de los procedimientos actuales a nivel tecnológico, posibilitando al mismo tiempo la reutilización de las propuestas generadas y favoreciendo su actualización regular.

Por lo tanto, se valora con un peso del 25% la adaptación de alternativas de última generación.

3. **Innovación:** uno de los propósitos transversales del proyecto es la generación de contribuciones al ámbito de la investigación de la QoS en redes inalámbricas, por lo que es imprescindible la puesta en valor de la aportación de nuevas herramientas en dicho campo. De este modo, se valorará con un 15% las propuestas metodológicas originales sobre la ponderación total.
4. **Complejidad de implementación:** la complejidad de las alternativas seleccionadas también resulta de cierta importancia, ya que la naturaleza del proyecto trata de cubrir los máximos aspectos posibles a la hora de representar las redes, y una inversión de tiempo excesiva en la caracterización de aspectos concretos de la tecnología puede provocar carencias en el desarrollo general del trabajo. Por ello, se establece un valor del 15% sobre la complejidad de las soluciones a seleccionar.
5. **Consumo de recursos:** además de las anteriores, es indispensable poner en valor los recursos tecnológicos disponibles para la realización del trabajo, ya que no resultan ilimitados. En consecuencia, se valora con un 15% la carga añadida en términos de procesamiento y memoria a los sistemas en uso por cada una de las alternativas.

La valoración final de cada posible implementación de los parámetros en el simulador WLAN se calcula, por tanto, de acuerdo con la siguiente ecuación:

$$\text{Valoración} = \text{Criterio}_1 \cdot 0,3 + \text{Criterio}_2 \cdot 0,25 + \text{Criterio}_3 \cdot 0,15 + \text{Criterio}_4 \cdot 0,15 + \text{Criterio}_5 \cdot 0,15$$

Ecuación 11: Ponderación de los criterios para la selección de la implementación de los parámetros WLAN.

En la Tabla 31 se resumen las implicaciones de la asignación de puntuaciones máximas y mínimas para cada criterio.

Criterio	Ponderación	Implicaciones de la puntuación máxima (5)	Implicaciones de la puntuación mínima (0)
1. Referencia del estudio teórico	30%	Representación fiel de los resultados del estudio teórico	Propuesta contraria a las conclusiones del estudio teórico
2. Ajuste a la actualidad de la tecnología	25%	Implementación de acuerdo con la última versión del estándar	Ajuste incompatible con la actualidad de la tecnología
3. Innovación	15%	Propuestas originales de integración de métodos sobre el simulador	Implementación básica y escueta
4. Complejidad de implementación	15%	Manejo de estructuras sencillas	Representaciones complejas y de alto coste temporal
5. Consumo de recursos	15%	Consumo mínimo de recursos del sistema	Alto consumo de recursos del sistema

Tabla 31: Criterios sobre la selección de alternativas de parametrización de red.

6.2.2. Criterios de Selección para los Indicadores de Rendimiento Clave

A la hora de realizar la selección de los parámetros de medida de calidad, se ha de poner en valor el ajuste de estos a los casos en estudio, es decir, de acuerdo con la tecnología empleada, los tipos de tráfico a simular, y el ámbito de aplicación de las simulaciones y posterior procesamiento. A continuación, se enumeran y describen de forma breve la selección de criterios establecida para la selección de KPIs.

1. **Adecuación a la tecnología:** la tecnología Wi-Fi se presenta como tecnología de acceso inalámbrica para múltiples casos de uso, y cuenta con unas características específicas como se ha recogido en el apartado introductorio Tecnología WLAN: 802.11. Por consiguiente, resulta fundamental poner en valor que las métricas en estudio sean válidas y representativas para las propiedades de dichas redes, por lo que se valora con un 30% de la nota.
2. **Ajuste a los tipos de tráfico:** en la Clasificación de Indicadores Clave de Rendimiento se indicaba la importancia de algunos parámetros en la medida de la calidad de distintos servicios. Por tanto, dicho conocimiento ha de ser tomado en consideración para la selección de indicadores, de forma que se ajusten a los tipos de tráfico implementados en el simulador. Este aspecto fundamental se pondera respecto a la valoración total con un 30%.
3. **Innovación:** como en la serie de criterios anteriores, se debe tener en cuenta que uno de los propósitos transversales del proyecto es la generación de contribuciones al ámbito de la investigación de la QoS en redes inalámbricas, por lo que es indispensable valorar la aportación de nuevas herramientas en dicho campo. De este modo, se valorará con un 20% las propuestas metodológicas innovadoras sobre la ponderación total.
4. **Complejidad de implementación:** debido a que el proyecto trata de abarcar el máximo número de aspectos posibles, se ha de poner en valor la inversión de tiempo adicional requerida en el desarrollo de métodos para el procesamiento de los KPIs. Por lo tanto, se establece para este criterio un valor del 20% sobre el peso total de la valoración.

Criterio	Ponderación	Implicaciones de la puntuación máxima (5)	Implicaciones de la puntuación mínima (0)
1. Adecuación a la tecnología	30%	Representación fiel del funcionamiento de la tecnología WLAN	Incompatible con las características de la tecnología WLAN
2. Ajuste a los tipos de tráfico	30%	Alta influencia en el desempeño de los tráficos bajo estudio	Desajuste respecto a los parámetros medibles de los tráficos en estudio
3. Innovación	20%	Propuestas originales de obtención de métricas	Implementaciones estándar o ya existentes
4. Complejidad de implementación	20%	Implementación sencilla y eficaz	Uso de métodos de cálculo complejos y de alto coste temporal

Tabla 32: Criterios sobre la selección de alternativas de KPIs.

En definitiva, la valoración final de cada posible implementación de los indicadores de calidad en el simulador WLAN se calcula, siguiendo la Ecuación 12.

$$\text{Valoración} = \text{Criterio}_1 \cdot 0,3 + \text{Criterio}_2 \cdot 0,3 + \text{Criterio}_3 \cdot 0,2 + \text{Criterio}_4 \cdot 0,2$$

Ecuación 12: Ponderación de los criterios para la selección de la implementación de los KPIs.

En la Tabla 32 se muestra de forma resumida la serie de implicaciones que tiene la obtención de puntuaciones máximas y mínimas para la selección de KPIs.

6.2.3. Criterios de Selección para los Algoritmos y Técnicas de Clustering

La selección de algoritmos a utilizar para el procesamiento de la salida del simulador resulta clave para el desarrollo de este estudio. Para poder llevarlo a cabo, se han definido una serie de criterios que provean las indicaciones necesarias para su selección:

1. **Compatibilidad de requisitos con el procedimiento de mejora de la QoS:** cada algoritmo de aprendizaje automático requiere un conocimiento previo del ámbito de aplicación para la definición de sus parámetros de entrada. En consecuencia, se debe tener en cuenta que dichos aspectos sean compatibles con los procedimientos de mejora de la QoS. Como resulta un aspecto decisivo para el estudio, se asigna un valor del 30% de la nota total.
2. **Eficiencia y escalabilidad:** el objetivo de la implementación de los algoritmos de clustering es realizar agrupaciones sobre sets de datos extraídos de las simulaciones. Por consiguiente, es indispensable poner en valor la eficiencia del algoritmo para ejecutarse sobre colecciones de distinta dimensión, y su correcto ajuste al número de clústeres y muestras en estudio. Por ello, se pondera con un peso del 25% sobre la valoración final.
3. **Adaptabilidad en la forma de los clústeres:** debido al desconocimiento inicial de las formas de los clústeres derivados de las métricas de red, se debe considerar la adaptabilidad de los algoritmos a estas. Como la adecuación de estos sobre las formas de los conjuntos de datos resulta de gran relevancia, se establece un valor del 25% sobre la valoración de las soluciones a seleccionar.
4. **Popularidad y reconocimiento:** además de las anteriores, resulta importante tener en cuenta la expansión o popularidad de los métodos a emplear, ya que, por lo general, facilitan la implementación en las herramientas disponibles y suponen soluciones adecuadas para múltiples planteamientos de problemas. En consecuencia, se valora con un 20% la utilización de algoritmos ampliamente utilizados.

Así, la valoración final de cada algoritmo de agrupamiento se obtiene mediante la siguiente ecuación:

$$\text{Valoración} = \text{Criterio}_1 \cdot 0,3 + \text{Criterio}_2 \cdot 0,25 + \text{Criterio}_3 \cdot 0,25 + \text{Criterio}_4 \cdot 0,2$$

Ecuación 13: Ponderación de los criterios para la selección de los algoritmos de clustering.

En la Tabla 33 se muestra de forma resumida la serie de implicaciones que tiene la obtención de puntuaciones máximas y mínimas para la selección de KPIs.

Criterio	Ponderación	Implicaciones de la puntuación máxima (5)	Implicaciones de la puntuación mínima (0)
1. Compatibilidad de requisitos con el procedimiento de mejora de la QoS	30%	Ajuste óptimo de los parámetros de entrada a la información disponible sobre el ámbito de aplicación	Incompatibilidad de los parámetros de entrada con la información conocida del ámbito de aplicación
2. Eficiencia y escalabilidad	25%	Alta eficiencia y ajuste a conjuntos de datos y clústeres de distinto tamaño	Baja eficiencia y poca escalabilidad frente a conjuntos de datos y clústeres de distinto tamaño
3. Adaptabilidad en la forma de los clústeres	25%	Aptitud del algoritmo frente a clústeres con formas regulares e irregulares	Falta de capacidad del algoritmo de resultar efectivo con distintas formas de clúster
4. Popularidad y reconocimiento	20%	Solución ampliamente utilizada	Algoritmo de uso específico y poco común

Tabla 33: Criterios sobre la selección de alternativas de algoritmos de clustering.

6.3. Selección de la Solución

Una vez establecidas las alternativas y definidos los criterios para su selección, el presente apartado recoge la aplicación de los criterios para elegir las herramientas y características más adecuadas para el estudio.

6.3.1. Selección de la Solución para la Parametrización de los entornos WLAN

Debido a la gran variedad de parámetros a definir en la caracterización de las simulaciones y a la extensión de las tablas resultantes, se traslada la información que recoge el cruce entre los criterios y las alternativas anteriormente expuestas con las ponderaciones correspondientes al Anexo II: Selección de alternativas para la parametrización de los escenarios WLAN.

A continuación, por tanto, se exponen los **resultados de la selección de alternativas** de la parametrización de escenarios.

En primer lugar, la Tabla 34 recoge la solución para la **definición espacial de los escenarios**. En los mismos se consideran entornos similares a los encontrados en el Resumen del Estudio de Escenarios de Aplicación del Estándar IEEE 802.11.

No obstante, se ajustan a las características específicas del estudio en base a los criterios de selección, definiendo así cuatro escenarios de tamaño medio, que en el caso de entornos suponen una muestra o porción en la que se recogen características propias de los escenarios.

Además, al tratarse de casuísticas de alta densidad, se delega la concentración a los nodos por red y la cercanía entre redes.

Parámetro característico	Entorno residencial	Entorno empresarial	Entorno interior de alta densidad	Entorno exterior de alta densidad
Forma	Rectangular	Rectangular	Circular	Circular
Variabilidad de la dimensión del escenario	Baja (hasta 40m ²)	Intermedia (hasta 100m ²)	Alta (más de 100m ²)	Alta (más de 100m ²)
Área	Mediana (50m ² – 100m ²)	Mediana (50m ² – 100m ²)	Mediana (50m ² – 100m ²)	Mediana (50m ² – 100m ²)

Tabla 34: Selección de la solución para la caracterización espacial de los escenarios.

Por consiguiente, una vez definidas las características espaciales del entorno, se procede a definir la **topología, posicionamiento y movilidad de los dispositivos** (Tabla 35 para los AP, y Tabla 36 para las STA).

Parámetro característico	Entorno residencial	Entorno empresarial	Entorno interior de alta densidad	Entorno exterior de alta densidad
Modelo de definición del número de APs por red (NodeContainer)	Determinista	Determinista	Determinista	Determinista
Densidad de APs por red (NodeContainer)	Baja (1)	Baja (1)	Baja (1)	Baja (1)
Interfaces de Red por AP (NetDeviceContainer)	Única (1)	Única (1)	Única (1)	Única (1)
Tipo de posicionamiento de APs (MobilityHelper)	Aleatorio	Aleatorio	Determinista	Determinista
Modelo de altura de los APs (MobilityHelper)	Determinista	Determinista	Aleatorio	Aleatorio
Modelo de movilidad de los APs (MobilityHelper)	Estático (sin movilidad)	Estático (sin movilidad)	Estático (sin movilidad)	Estático (sin movilidad)

Tabla 35: Selección de la solución para la caracterización topológica, posicionamiento y movilidad de los AP.

Como se puede observar, centrando la atención en los **puntos de acceso**, se define un único AP por red y una sola interfaz por dispositivo, de modo que todo el tráfico pasa por dicho nodo a través de la misma interfaz de red. En cuanto al posicionamiento y movilidad, cabe destacar que se utilizan modelos de movilidad estáticos para todos los casos (el AP se basa en una instalación no móvil de un punto de acceso Wi-Fi) y se definen posicionamientos aleatorios para modelos domésticos y de oficina, caracterizando la diversidad de los espacios de dicho tipo. Sin embargo, en el caso de los entornos de alta densidad, se realiza un posicionamiento determinista, con la intención de ubicar cada AP en el centro del área definida, de forma similar a la formación de celdas de cobertura de redes móviles.

Parámetro característico	Entorno residencial	Entorno empresarial	Entorno interior de alta densidad	Entorno exterior de alta densidad
Modelo de definición del número de STAs por red (NodeContainer)	Aleatorio	Aleatorio	Aleatorio	Aleatorio
Densidad de STAs por red (NodeContainer)	Baja (alrededor de 10)	Media (alrededor de 25)	Alta (alrededor de 50)	Alta (alrededor de 50)
Interfaces de Red por STA (NetDeviceContainer)	Única (1)	Única (1)	Única (1)	Única (1)
Tipo de posicionamiento de STAs (MobilityHelper)	Aleatorio	Aleatorio	Aleatorio	Aleatorio
Modelo de altura de las STAs (MobilityHelper)	Aleatorio	Aleatorio	Aleatorio	Aleatorio
Modelo de movilidad de las STAs (MobilityHelper)	Random Walk	Estático (sin movilidad)	Random Walk	Random Walk

Tabla 36: Selección de la solución para la caracterización topológica, posicionamiento y movilidad de las STAs.

Por otro lado, los escenarios se enriquecen con múltiples **STAs** que generan y reciben tráfico, y cuya densidad y patrones de movimiento varían para cada entorno. Esta selección se realiza siguiendo las conclusiones extraídas del estudio teórico y valorando márgenes de dispositivos realizables con equipos estándar, debido a que la generación de grandes cantidades de flujos de tráfico sobre múltiples nodos puede dar lugar a altos tiempos de procesamiento en los equipos sobre los que se ejecuta NS-3.

Después de definir las características más generales de cada escenario a simular, se realiza la selección de los **aspectos frecuenciales**, como se recoge en la Tabla 37.

Parámetro característico	Entorno residencial	Entorno empresarial	Entorno interior de alta densidad	Entorno exterior de alta densidad
Selección de la banda frecuencial (SpectrumChannelHelper)	5 GHz	5 GHz	5 GHz	5 GHz
Modelo de configuración de canales (SpectrumChannelHelper)	Aleatorio	Determinista	Compartición de canal	Compartición de canal
Ancho de canal (SpectrumChannelHelper)	80 MHz	80 MHz	80 MHz	80 MHz

Tabla 37: Selección de la solución de implementación a nivel frecuencial.

Con motivo de hacer uso en todos los casos de los mismos recursos espectrales –a excepción de la selección de canal, que es distintiva de cada tipo de escenario– y debido a la amplia utilización de la banda de 5 GHz como se mencionaba en el estudio teórico,

y se selecciona esta para la realización de las simulaciones, haciendo uso de canales de 80 MHz.

En cuanto a la **configuración de canales**, se define una selección aleatoria para las simulaciones en entornos domésticos, determinista para entornos de oficina ya que se presupone un mayor conocimiento y capacidad de configuración por la disponibilidad de técnicos con conocimiento de la tecnología, y de compartición de canal para representar escenarios donde la alta densidad de dispositivos hace recurrente el empleo de los mismos canales para obtener acceso a servicios de red.

Tras seleccionar las características frecuenciales, se procede a definir los **parámetros de transmisión** seleccionados para APs y STAs, como se indica en la Tabla 38.

Parámetro característico	Entorno residencial	Entorno empresarial	Entorno interior de alta densidad	Entorno exterior de alta densidad
Configuración de antenas de los AP (SpectrumWifiPhyHelper)	2x2	2x2	4x4	4x4
Configuración de antenas de las STA (SpectrumWifiPhyHelper)	1x1	1x1	2x2	2x2
Número de flujos espaciales AP (SpectrumWifiPhyHelper)	Múltiple con bajo valor (2)	Múltiple con bajo valor (2)	Múltiples con valor intermedio (4)	Múltiples con valor intermedio (4)
Número de flujos espaciales STA (SpectrumWifiPhyHelper)	Único (1)	Único (1)	Múltiple con bajo valor (2)	Múltiple con bajo valor (2)
Modelo de potencia TX APs (SpectrumWifiPhyHelper)	Variable	Variable	Variable	Variable
Potencia TX APs (SpectrumWifiPhyHelper)	Media (10dBm – 20 dBm)	Alta (20 dBm – 30 dBm)	Alta (20 dBm – 30 dBm)	Alta (20 dBm – 30 dBm)
Modelo de potencia TX STAs (SpectrumWifiPhyHelper)	Variable	Variable	Variable	Variable
Potencia TX STAs (SpectrumWifiPhyHelper)	Media (10dBm – 20 dBm)	Media (10dBm – 20 dBm)	Media (10dBm – 20 dBm)	Media (10dBm – 20 dBm)

Tabla 38: Selección de la solución de configuración de la transmisión de los dispositivos.

Como se puede observar, la definición para los dispositivos es similar para APs y STAs, pero con características de mayor nivel en el caso de los primeros. Se consideran, por tanto, **configuraciones de antenas** 2x2 y 4x4 para los APs con hasta 4 flujos espaciales, partiendo de los valores aproximados del estudio teórico, y configuraciones de 1x1 y 2x2 para las STAs, permitiendo un único flujo espacial simultáneo.

Asimismo, la **potencia** se define en forma variable, presuponiendo que la intensidad transmitida no resulta constante y asignada mediante rangos basados en las métricas recogidas de los diferentes estudios analizados en el Estado del Arte.

La definición de los **modelos de propagación** a implementar en el simulador se realiza de forma separada, y la selección final se resume en la Tabla 39. En la aplicación de los criterios de decisión previamente establecidos el aspecto más determinante ha resultado, de forma similar a los parámetros anteriores, las conclusiones del estudio teórico.

En su mayoría, los modelos seleccionados se corresponden con los propuestos por el TGax, que resultan aplicables a la tecnología más actual. Para ello, se han de configurar los modelos específicos definidos por estos para el cálculo de pérdidas, empleándose, en caso de ser posible, configuraciones que aprovechen las bondades de los métodos ya implementados en el simulador.

Parámetro característico	Entorno residencial	Entorno empresarial	Entorno interior de alta densidad	Entorno exterior de alta densidad
Modelo de pérdidas de propagación (SpectrumWifiPhyHelper)	Modelo específico (3DLogDistance + Obstáculos)	3GPP Indoor Office	Modelo específico (TMB TGax)	LogDistance

Tabla 39: Selección de la solución de modelos de pérdidas de propagación.

No obstante, el modelo seleccionado para el entorno exterior de alta densidad difiere del resto al tomar como referencia una aproximación sencilla mediante el empleo de pérdidas logarítmicas en función de la distancia en lugar de redes celulares (modelos UMi y UMa), debido a que los escenarios a contemplar en este caso no son únicamente urbanos, y no se trata de entornos que no siguen la estructura celular propuesta por el TGax ni satisface las especificaciones para su uso en NS-3.

Parámetro característico	Entorno residencial	Entorno empresarial	Entorno interior de alta densidad	Entorno exterior de alta densidad
EDCA (WifiMacHelper)	Sí	Sí	Sí	Sí
Definición de valores EDCA (WifiMacHelper)	Estándar	Estándar	Estándar	Estándar
Agregación (WifiMacHelper)	A-MPDU	A-MPDU	A-MPDU	A-MPDU
Tamaño de agregación (WifiMacHelper)	64 Máx	64 Máx	64 Máx	64 Máx
Repeticiones (retries)	Estándar	Estándar	Estándar	Estándar
GI	800ns	800ns	800ns	3200ns
IFS	Estándar	Estándar	Estándar	Estándar
RTS/CTS	No	No	Sí	Sí
Umbral RTS/CTS	Estándar	Estándar	Estándar	Estándar

Tabla 40: Selección de la solución de configuración a nivel de enlace y físico.

Para finalizar con los aspectos principales de los escenarios, en la Tabla 40 se especifican los **parámetros a nivel de protocolo** –tanto de la capa física como de enlace– para la configuración del protocolo IEEE 802.11. La solución obtenida es igual para todos los escenarios, a excepción del uso del mecanismo RTS/CTS para entornos de alta densidad, ya que permite la evasión de colisiones y mejora la eficiencia en casuísticas de dicho tipo. En general, se hace uso de parámetros de configuración estándar.

Para terminar con la selección de soluciones de la parametrización WLAN, en la Tabla 41 se muestran los **tipos de tráfico** a implementar según el análisis realizado, recogiendo una selección de 5 tipos –con ánimo de acotar la extensión del estudio– cuya asignación de flujos se puede observar en la Tabla 42.

Definición de tráfico	Entorno residencial	Entorno empresarial	Entorno interior de alta densidad	Entorno exterior de alta densidad
VoD	✓	✓	✓	✓
HTTP	✓	✓	✓	✓
FTP	✓	✓	✓	✓
VoIP	✓	✓	✓	✓
Gaming	✓		✓	✓

Tabla 41: Selección de la solución de los tipos de tráfico de red a implementar.

En relación con los flujos de tráfico, se toma como referencia principal la definición del reparto de servicios sobre nodos del TGax para sus propuestas de escenarios. De este modo, se establecen **porcentajes alcanzables de los dispositivos** –tanto APs como STAs– para tráfico en sentido ascendente (STA→AP) y descendente (AP→STA), de forma que la aplicación real oscile entre porcentajes nulos y los definidos en la tabla.

Definición de tráfico		Entorno residencial	Entorno empresarial	Entorno interior de alta densidad	Entorno exterior de alta densidad
Tipo de tráfico	Flujo	% AP / % STA	% AP / % STA	% AP / % STA	% AP / % STA
VoD	DL	60% / 60%	20% / 20%	60% / 60%	80% / 80%
	UL	10% / 10%	10% / 10%	10% / 10%	20% / 20%
HTTP	DL	40% / 40%	60% / 60%	40% / 40%	50% / 50%
	UL	0% / 0%	0% / 0%	0% / 0%	0% / 0%
FTP	DL	20% / 20%	20% / 20%	10% / 10%	10% / 10%
	UL	0% / 0%	0% / 0%	0% / 0%	0% / 0%
VoIP	DL	20% / 20%	20% / 20%	40% / 40%	60% / 60%
	UL	20% / 20%	20% / 20%	40% / 40%	60% / 60%
Gaming	DL	30% / 30%	0% / 0%	50% / 50%	80% / 80%
	UL	30% / 30%	0% / 0%	50% / 50%	80% / 80%

Tabla 42: Selección de la solución de los flujos a establecer en la caracterización de escenarios.

6.3.2. Selección de la Solución para los Indicadores de Rendimiento Clave

La selección de la solución sobre KPIs requiere un **enfoque distinto** a los otros dos procedimientos de selección de soluciones, ya que estos deben ser **específicos del caso de uso**, y este proyecto consta de dos aplicaciones distintas.

En concreto, este trabajo tiene como objetivo la extracción del contexto del usuario para la realización de las tareas de diferenciación de entornos y clasificación de perfiles.

Por consiguiente, resulta indispensable distinguir las perspectivas de ambos procedimientos. Para ello, se toman como referencia los dos ejes definidos en el apartado Alternativas de Indicadores de Rendimiento Clave Específicas: el **ámbito de aplicación** (a nivel de dispositivo o global), y el **tiempo de análisis**.

Del mismo modo, la evaluación de parámetros en base a los ejes descritos genera múltiples posibilidades, por lo que se acota la selección por grupos de KPIs en tablas separadas, y se establece como condición de selección la superación de una puntuación de tres sobre el máximo.

Además, tan solo se contempla el uso de un valor concreto (total, medio o instantáneo) de cada métrica en cada aplicación, y entre las posibilidades resultantes se aceptan como KPI las mayores puntuaciones obtenidas, siendo múltiples en caso de haber empates.

Criterio	Identificador
1. Adecuación a la tecnología	C.2.1
2. Ajuste a los tipos de tráfico	C.2.2
3. Innovación	C.2.3
4. Complejidad de implementación	C.2.4

Tabla 43: Mapeo de los criterios para la selección de KPIs y sus identificadores.

En primer lugar, se presenta la selección de **soluciones para la diferenciación de entornos**, que toma una perspectiva global de la red para tratar de extraer la información latente en los parámetros de medida y caracterizar así los escenarios.

Por lo tanto, se presta especial atención a los AP y a las métricas globales, tomando como referencia temporal total la duración de la simulación. Cabe mencionar que la obtención de métricas globales exige la combinación de información obtenida tanto de dispositivos terminales como puntos de acceso, por lo que, de forma general, aporta complejidad al conjunto de datos, pero también a la extracción y cálculo de los indicadores. Teniendo dichos aspectos en consideración, las tablas siguientes recogen la selección de la solución de KPIs para el propósito de diferenciación de entornos.

Para comenzar, se evalúa la posibilidad de añadir como parámetro el **tiempo de funcionamiento** de los dispositivos y a nivel global de la red. Como muestra la Tabla 44, ninguna de las métricas derivadas obtiene una puntuación relevante, en su valor instantáneo, valor medio o total. Esto se debe a que las simulaciones están acotadas en el tiempo, lo que hace que no tenga gran efecto sobre el tráfico ni la disponibilidad general del servicio de red.

De este modo, no adquiere altas puntuaciones debido a la falta de correspondencia con los criterios establecidos y su aplicación sobre el procedimiento de diferenciación de entornos. Sin embargo, a pesar de que no resulte una propuesta innovadora, es una métrica sencilla de implementar mediante NS-3.

Parámetro	Alternativa		C.2.1	C.2.2	C.2.3	C.2.4	Resultado
Tiempo de funcionamiento	AP	Valor inst.	1	0	0	4	1.1
		Valor medio	3	0	0	3	1.5
		Valor total	3	0	0	3	1.5
	STA	Valor inst.	1	0	0	4	1.1
		Valor medio	1	0	0	3	0.9
		Valor total	1	0	0	3	0.9
	Global	Valor inst.	0	0	0	4	0.8
		Valor medio	0	0	0	3	0.6
		Valor total	0	0	0	3	0.6

Tabla 44: Evaluación de la solución del tiempo de funcionamiento como KPI para la diferenciación de entornos.

A continuación, se valoran los parámetros de **ancho de banda** (bandwidth) y **tasa de datos o rendimiento** (throughput), haciendo referencia el primero a la capacidad máxima teórica sobre las conexiones entre dispositivos, y el segundo al rendimiento real o efectivo de estas.

Como muestra la Tabla 45, adquiere mayor sentido la implementación del parámetro de throughput, ya que la capacidad de los enlaces se establece manualmente en el simulador de forma global, y no aporta más información.

No obstante, la medida de la tasa efectiva de transmisión toma un alto valor a nivel de punto de acceso y global, ya que, según las configuraciones establecidas para los escenarios, ambas proporcionan información sobre los flujos totales en las redes de diferentes entornos. Además, la obtención e implementación de dicha información desde APs en entornos reales resulta de menor complejidad. Por otro lado, las STAs proporcionan valores individuales de los usuarios, por lo que en caso de considerarse como representantes de un escenario resulta conveniente estudiar su valor medio o total.

Parámetro	Alternativa		C.2.1	C.2.2	C.2.3	C.2.4	Resultado
Bandwidth	AP	Valor inst.	0	0	0	0	0
		Valor medio	0	0	0	0	0
		Valor total	0	0	0	0	0
	STA	Valor inst.	0	0	0	0	0
		Valor medio	0	0	0	0	0
		Valor total	0	0	0	0	0
	Global	Valor inst.	2	2	0	0	1.2
		Valor medio	3	3	0	0	1.8
		Valor total	3	3	0	0	1.8
Throughput	AP	Valor inst.	2	3	2	3	2.5
		Valor medio	4	4	3	3	3.6
		Valor total	3	3	2	3	2.8
	STA	Valor inst.	2	3	2	2	2.3
		Valor medio	4	3	3	2	3.1
		Valor total	3	3	2	2	2.6
	Global	Valor inst.	2	3	2	3	2.5
		Valor medio	4	4	3	2	3.4
		Valor total	3	3	2	3	2.8

Tabla 45: Evaluación de la solución del bandwidth y throughput como KPIs para la diferenciación de entornos.

Para continuar con la información sobre la transmisión de datos, la Tabla 46 muestra la evaluación de las métricas de **número de paquetes o bytes transmitidos y recibidos**, que proporcionan información sobre los niveles de tráfico en la red.

Parámetro	Alternativa		C.2.1	C.2.2	C.2.3	C.2.4	Resultado
Número de paquetes / bytes transmitidos	AP	Valor inst.	1	2	1	3	1.7
		Valor medio	4	4	2	3	3.4
		Valor total	3	3	2	3	2.8
	STA	Valor inst.	1	3	1	2	1.8
		Valor medio	4	3	2	2	2.9
		Valor total	3	3	2	2	2.6
	Global	Valor inst.	1	2	1	3	1.7
		Valor medio	4	4	2	2	3.2
		Valor total	3	3	2	3	2.8
Número de paquetes / bytes recibidos	AP	Valor inst.	1	2	1	3	1.7
		Valor medio	4	4	2	3	3.4
		Valor total	3	3	2	3	2.8
	STA	Valor inst.	1	3	1	2	1.8
		Valor medio	4	3	2	2	2.9
		Valor total	3	3	2	2	2.6
	Global	Valor inst.	1	2	1	3	1.7
		Valor medio	4	4	2	2	3.2
		Valor total	3	3	2	3	2.8

Tabla 46: Evaluación de la solución del número de paquetes / bytes recibidos y transmitidos como KPIs para la diferenciación de entornos.

De forma relacionada a los parámetros anteriores, la Tabla 47 recoge las **tasas de pérdidas y errores en paquetes** como KPIs. La primera se obtiene a partir de la diferencia entre los paquetes enviados y recibidos, mientras que la segunda se calcula mediante el procesamiento de los paquetes e inspección de parámetros que indiquen la existencia de errores, como la verificación de integridad a través de checksums.

Parámetro	Alternativa		C.2.1	C.2.2	C.2.3	C.2.4	Resultado
Tasa de pérdida de paquetes	AP	Valor inst.	1	1	0	3	1.2
		Valor medio	4	4	1	2	3
		Valor total	4	3	1	2	2.7
	STA	Valor inst.	1	1	0	3	1.2
		Valor medio	4	3	1	2	2.7
		Valor total	4	3	1	2	2.7
	Global	Valor inst.	1	1	0	3	1.2
		Valor medio	4	4	1	1	2.8
		Valor total	4	2	1	2	2.4
Tasa de errores en paquetes	AP	Valor inst.	1	1	0	2	1
		Valor medio	4	4	1	1	2.8
		Valor total	4	3	1	1	2.5
	STA	Valor inst.	1	1	0	2	1
		Valor medio	4	4	1	1	2.8
		Valor total	4	3	1	1	2.5
	Global	Valor inst.	1	1	0	2	1
		Valor medio	4	3	1	1	2.5
		Valor total	4	2	1	1	2.2

Tabla 47: Evaluación de la solución tasa de pérdida de paquetes y errores como KPIs para la diferenciación de entornos.

Se destaca la facilidad de implementación del primero, aunque se debe de tener en cuenta la relación con los KPIs de la Tabla 46, ya que la pérdida de paquetes puede calcularse como la diferencia entre transmisión y recepción, lo que puede introducir redundancia en las medidas.

Además de los anteriores, también se valoran como solución de KPIs para la diferenciación de entornos el empleo de parámetros temporales como el **retardo** y el **jitter**. En la Tabla 48 se obtiene como resultado de la selección el retardo medio a nivel de AP, así como el jitter medio percibido en el mismo. Esto se debe a que se ha considerado que aporta más información a nivel de dispositivo que global, y resultan métricas sencillas de obtener como valor medio.

Parámetro	Alternativa	C.2.1	C.2.2	C.2.3	C.2.4	Resultado	
Retardo	AP	Valor inst.	2	2	1	3	2
		Valor medio	4	4	2	2	3.2
		Valor total	3	3	2	2	2.6
	STA	Valor inst.	2	3	1	3	2.3
		Valor medio	4	4	2	1	3
		Valor total	3	3	2	2	2.6
	Global	Valor inst.	2	2	1	3	2
		Valor medio	3	3	2	2	2.6
		Valor total	3	3	2	2	2.6
Jitter	AP	Valor inst.	1	1	0	3	1.2
		Valor medio	4	4	2	2	3.2
		Valor total	3	3	2	2	2.6
	STA	Valor inst.	1	1	0	3	1.2
		Valor medio	4	4	2	1	3
		Valor total	3	3	2	2	2.6
	Global	Valor inst.	1	1	0	3	1.2
		Valor medio	3	3	2	2	2.6
		Valor total	3	3	2	2	2.6

Tabla 48: Evaluación de la solución del retardo y jitter percibidos como KPIs para la diferenciación de entornos.

Por último, se considera como KPI la medida de la diferencia señal a ruido **SNR** percibida desde los puntos de acceso, como se muestra en la Tabla 49, ya que esta puede, principalmente, ser un indicador clave del funcionamiento general de la red.

Parámetro	Alternativa	C.2.1	C.2.2	C.2.3	C.2.4	Resultado	
SNR	AP	Valor inst.	2	1	2	4	2.1
		Valor medio	4	3	3	2	3.1
		Valor total	4	2	3	3	3
	STA	Valor inst.	2	1	2	4	2.1
		Valor medio	4	3	3	1	2.9
		Valor total	4	2	3	3	3
	Global	Valor inst.	2	1	2	4	2.1
		Valor medio	4	2	3	2	2.8
		Valor total	4	2	3	3	3

Tabla 49: Evaluación de la solución del valor SNR como KPIs para la diferenciación de entornos.

En segundo lugar, se recoge la selección de **soluciones para la clasificación de perfiles**, que toma un enfoque más específico al dispositivo y los flujos de tráfico, por

lo que se presta mayor atención a los parámetros de medida sobre equipos finales de los usuarios. Asimismo, en este caso se considera como tiempo total el de transmisión de un flujo de datos, lo que afecta a la selección de alternativas.

En un inicio, de forma paralela a las evaluaciones anteriores, se valora la posibilidad de añadir como parámetro el **tiempo de funcionamiento** de los dispositivos y a nivel global de la red. Del mismo modo que en la diferenciación de entornos, el planteamiento de las simulaciones hace que no tenga una puntuación alta a nivel tecnológico, de análisis de tráfico y de innovación. Esto mismo se recoge en la Tabla 50.

Parámetro	Alternativa		C.2.1	C.2.2	C.2.3	C.2.4	Resultado
Tiempo de funcionamiento	AP	Valor inst.	1	0	0	4	1.1
		Valor medio	3	0	0	3	1.5
		Valor total	3	0	0	3	1.5
	STA	Valor inst.	1	0	0	4	1.1
		Valor medio	1	0	0	3	0.9
		Valor total	1	0	0	3	0.9
	Global	Valor inst.	0	0	0	4	0.8
		Valor medio	0	0	0	3	0.6
		Valor total	0	0	0	3	0.6

Tabla 50: Evaluación de la solución del tiempo de funcionamiento como KPI para clasificación de perfiles.

Después, se valoran los parámetros de **bandwidth** y **throughput**, y como muestra la Tabla 51, se destaca la medida del throughput medio, que adquiere mayor sentido a nivel conceptual en la evaluación de los flujos de datos para distinguir perfiles de usuario.

Parámetro	Alternativa		C.2.1	C.2.2	C.2.3	C.2.4	Resultado
Bandwidth	AP	Valor inst.	0	0	0	0	0
		Valor medio	0	0	0	0	0
		Valor total	0	0	0	0	0
	STA	Valor inst.	0	0	0	0	0
		Valor medio	0	0	0	0	0
		Valor total	0	0	0	0	0
	Global	Valor inst.	2	3	0	0	1.5
		Valor medio	3	4	0	0	2.1
		Valor total	3	4	0	0	2.1
Throughput	AP	Valor inst.	2	3	2	2	2.3
		Valor medio	4	3	3	2	3.1
		Valor total	3	3	2	2	2.6
	STA	Valor inst.	2	3	2	3	2.5
		Valor medio	4	4	3	3	3.6
		Valor total	3	3	2	3	2.8
	Global	Valor inst.	2	3	2	3	2.5
		Valor medio	4	3	3	3	3.3
		Valor total	3	3	2	3	2.8

Tabla 51: Evaluación de la solución del bandwidth y throughput como KPIs para la clasificación de perfiles.

A continuación, la Tabla 52 muestra las ponderaciones otorgadas a las métricas de **número de paquetes o bytes transmitidos y recibidos**, que proporcionan información sobre los niveles de tráfico presentes en la red. Entre las diferentes

alternativas para cada parámetro, se resalta el valor medio a nivel de estaciones, que proporciona información promedio en el tiempo de duración de los flujos por cada origen o destino.

Parámetro	Alternativa		C.2.1	C.2.2	C.2.3	C.2.4	Resultado
Número de paquetes / bytes transmitidos	AP	Valor inst.	1	3	1	2	1.8
		Valor medio	4	3	2	2	2.9
		Valor total	3	3	2	2	2.6
	STA	Valor inst.	1	2	1	3	1.7
		Valor medio	4	4	2	3	3.4
		Valor total	3	3	2	3	2.8
	Global	Valor inst.	1	3	1	2	1.8
		Valor medio	4	3	2	2	2.9
		Valor total	3	3	2	2	2.6
Número de paquetes / bytes recibidos	AP	Valor inst.	1	3	1	2	1.8
		Valor medio	4	3	2	2	2.9
		Valor total	3	3	2	2	2.6
	STA	Valor inst.	1	2	1	3	1.7
		Valor medio	4	4	2	3	3.4
		Valor total	3	3	2	3	2.8
	Global	Valor inst.	1	3	1	2	1.8
		Valor medio	4	3	2	2	2.9
		Valor total	3	3	2	2	2.6

Tabla 52: Evaluación de la solución del número de paquetes / bytes recibidos y transmitidos como KPIs para la clasificación de perfiles.

Con relación directa a los parámetros anteriores, la Tabla 53 muestra la evaluación de las **tasas de pérdidas y errores en paquetes** como KPIs, donde se destaca la facilidad de implementación del primero, y se selecciona como alternativa la medida de su valor medio a nivel de STA.

Parámetro	Alternativa		C.2.1	C.2.2	C.2.3	C.2.4	Resultado
Tasa de pérdida de paquetes	AP	Valor inst.	1	1	0	3	1.2
		Valor medio	4	4	1	2	3
		Valor total	4	3	1	2	2.7
	STA	Valor inst.	1	2	1	3	1.7
		Valor medio	4	4	2	3	3.4
		Valor total	3	3	2	3	2.8
	Global	Valor inst.	1	1	0	3	1.2
		Valor medio	4	3	1	2	2.7
		Valor total	4	2	1	2	2.4
Tasa de errores en paquetes	AP	Valor inst.	1	1	0	2	1
		Valor medio	4	4	1	1	2.8
		Valor total	4	3	1	1	2.5
	STA	Valor inst.	1	1	0	2	1
		Valor medio	4	4	1	1	2.8
		Valor total	4	3	1	1	2.5
	Global	Valor inst.	1	1	0	2	1
		Valor medio	4	3	1	1	2.5
		Valor total	4	2	1	1	2.2

Tabla 53: Evaluación de la solución tasa de pérdida de paquetes y errores como KPIs para la clasificación de perfiles.

En términos de medidas temporales, la Tabla 54 muestra la selección de alternativas sobre los parámetros de **retardo** y **jitter**. De forma similar a las soluciones anteriores, se decide como KPI a implementar el valor medio a de las estaciones, en este caso para ambas medidas, que resultan parámetros fundamentales en la provisión de la calidad de servicio de diferentes servicios de red.

Parámetro	Alternativa	C.2.1	C.2.2	C.2.3	C.2.4	Resultado	
Retardo	AP	Valor inst.	2	2	1	3	2
		Valor medio	4	3	2	2	2.9
		Valor total	3	4	2	2	2.9
	STA	Valor inst.	2	3	1	3	2.3
		Valor medio	4	4	2	2	3.2
		Valor total	3	4	2	2	2.9
	Global	Valor inst.	2	2	1	3	2
		Valor medio	3	3	2	2	2.6
		Valor total	3	3	2	2	2.6
Jitter	AP	Valor inst.	1	1	0	3	1.2
		Valor medio	4	3	2	2	2.9
		Valor total	3	4	3	2	3.1
	STA	Valor inst.	1	1	0	3	1.2
		Valor medio	4	4	2	2	3.2
		Valor total	3	3	3	2	2.8
	Global	Valor inst.	1	1	0	3	1.2
		Valor medio	3	3	2	2	2.6
		Valor total	3	3	3	2	2.8

Tabla 54: Evaluación de la solución del retardo y jitter percibidos como KPIs para la clasificación de perfiles.

Finalmente, se considera como parámetro la medida de la diferencia señal a ruido **SNR** percibida por las estaciones, como se muestra en la Tabla 55, ya que dicho parámetro es representativo de la disposición de los usuarios finales.

Parámetro	Alternativa	C.2.1	C.2.2	C.2.3	C.2.4	Resultado	
SNR	AP	Valor inst.	2	1	2	4	2.1
		Valor medio	3	3	3	2	2.8
		Valor total	4	2	3	3	3
	STA	Valor inst.	2	1	2	4	2.1
		Valor medio	4	3	3	2	3.1
		Valor total	4	2	3	3	3
	Global	Valor inst.	2	1	2	4	2.1
		Valor medio	3	2	3	2	2.5
		Valor total	4	2	3	3	3

Tabla 55: Evaluación de la solución del valor SNR como KPIs para la clasificación de perfiles.

En definitiva, la Tabla 57 recoge las **soluciones definitivas** seleccionadas por cada procedimiento de mejora de la QoS, basándose en valores medios de cada parámetro.

Por lo general, se asocian parámetros relativos a los APs para la diferenciación de entornos y métricas de las STAs para la clasificación de perfiles. Esto se debe a que el primero recoge una perspectiva más general –de la red– mientras que el segundo se centra en los indicadores de los usuarios finales.

Aplicación	KPI	STA	AP	Global
Diferenciación de entornos <i>(referencia temporal: tiempo de simulación)</i>	Throughput		✓	
	Número de paquetes / bytes TX		✓	
	Número de paquetes / bytes RX		✓	
	Tasa de pérdida de paquetes		✓	
	Retardo		✓	
	Jitter		✓	
	SNR		✓	
Clasificación de perfiles <i>(referencia temporal: duración del flujo de tráfico)</i>	Throughput	✓		
	Número de paquetes / bytes TX	✓		
	Número de paquetes / bytes RX	✓		
	Tasa de pérdida de paquetes	✓		
	Retardo	✓		
	Jitter	✓		
	SNR	✓		

Tabla 56: Selección de las soluciones para la implementación de KPIs a través de valores medios.

6.3.3. Selección de la Solución para los Algoritmos y Técnicas de Clustering

La selección de las técnicas de agrupamiento a utilizar se hace en función de los criterios definidos previamente, y con intención de facilitar la exposición de datos, en la Tabla 57 se muestran los identificadores definidos para cada criterio, que servirán para relacionar las valoraciones realizadas sobre los algoritmos con cada aspecto a considerar.

Criterio	Identificador
1. Compatibilidad de requisitos con el procedimiento de mejora de la QoS	C.3.1
2. Eficiencia y escalabilidad	C.3.2
3. Adaptabilidad en la forma de los clústeres	C.3.3
4. Popularidad y reconocimiento	C.3.4

Tabla 57: Mapeo de los criterios para la selección de algoritmos de clustering y sus identificadores.

Debido a que el objetivo del proyecto es valorar la aptitud de las técnicas de agrupamiento sobre los procedimientos de mejora de la calidad del servicio, se parte de una primera selección de los enfoques más utilizados en la clasificación de algoritmos extraída del apartado dedicado a los Algoritmos de Clustering, y realiza la selección de soluciones por categorías, eligiendo un total de cuatro algoritmos a implementar, como se puede observar en la Tabla 58.

Entre los **algoritmos basados en particiones**, destaca K-Means, que es una solución ampliamente utilizada y que se ajusta a los parámetros de entrada de los que se dispone –el número de grupos a formar– además de permitir el tratamiento de datos ajustado a las dimensiones del conjunto disponible de forma eficiente. Sin embargo, no dispone de la capacidad de descartar muestras lejanas, lo que puede perjudicar en términos de adaptabilidad a la forma de los clústeres.

En relación con los **algoritmos basados en densidad**, la alternativa con mayor puntuación es DBSCAN, que resulta una solución reconocida y capaz de generar clústeres de distinta forma dependiendo de la densidad de muestras, escalable en términos del tamaño del conjunto de datos pero que requiere disponer de

conocimiento previo sobre el tamaño de los clústeres a formar. En el caso del presente estudio, dicha información es conocida con anterioridad, al tratarse de simulaciones generadas sobre escenarios y casos de uso definidos previamente.

Los **algoritmos basados en la distribución** son técnicas de gran complejidad, y a pesar de que requiere de un alto número de parámetros de entrada, se selecciona como solución el uso de GMM, de forma que se pueda validar otra técnica ampliamente utilizada y conocida por su capacidad de hacer estimaciones adecuadas de densidad a través de su aplicación sobre los sets de muestras del estudio.

Por último, entre los **algoritmos jerárquicos** resalta BIRCH, que del mismo modo que en las selecciones anteriores, es el que cuenta con mayor reconocimiento como solución. En este caso, es apto para el tratamiento de grandes conjuntos de datos y descarte de muestras lejanas a los grupos, pero requiere tener nociones del tamaño de los grupos a formar y las distancias entre muestras y grupos.

Tipo de agrupamiento	Alternativa	C.2.1	C.2.2	C.2.3	C.2.4	Resultado
Particional	K-Means	4	4	2	5	3.7
	K-Medoids	4	4	3	2	3.35
Basado en densidad	DBSCAN	3	4	4	5	3.9
	OPTICS	3	4	4	4	3.7
	Mean-Shift	3	2	4	4	3.2
Basado en la distribución	DBCLASD	3	4	4	2	3.3
	GMM	3	3	4	5	3.65
Jerárquico	BIRCH	3	4	4	5	3.9
	CURE	4	4	4	2	3.6
	ROCK	3	3	4	1	2.85

Tabla 58: Selección de las soluciones para la implementación algoritmos de clustering.

7. Análisis de Riesgos

A lo largo del desarrollo del proyecto, cabe la posibilidad de que tengan lugar situaciones indeseadas que pongan en riesgo el desempeño de la planificación inicial y la finalización del propio trabajo. Por lo tanto, resulta imprescindible evaluar las posibles amenazas y eventos inoportunos que puedan afectar a la realización del estudio.

En consecuencia, el presente apartado recoge el plan de actuación frente a riesgos en tres tareas principales. Inicialmente, se realizará la identificación de los riesgos principales del proyecto. Después, se evaluará el impacto y probabilidad de ocurrencia de los riesgos definidos. Por último, se desarrollará el plan de contingencia en función de los factores anteriores de acuerdo con un orden de prioridad establecido.

7.1. Identificación de los Riesgos Principales

Los riesgos identificados en el desarrollo del Trabajo Fin de Máster son los siguientes:

- **R1: Imposibilidad técnica de implementación:** Debido al ámbito de investigación del proyecto y el manejo de herramientas de diseño en constante desarrollo, puede darse la situación de que tanto a nivel de simulación como de procesamiento de datos se encuentren obstáculos que dificulten la implementación del estudio de forma práctica.
- **R2: Superación de fechas en la planificación:** Un factor fundamental en la realización del trabajo es el tiempo definido para el desarrollo de cada tarea, cuyo incumplimiento puede desembocar en retrasos y reajustes de la planificación, y en casos extremos, en la estructura propia del proyecto.
- **R3: Fallos de comunicación:** Se trata de los posibles problemas de comunicación que puedan tener lugar entre el alumno y la directora del trabajo. Esto puede deberse a las dinámicas de comunicación habituales, que se basan en el intercambio de mensajes de correo electrónico, o falta de entendimiento a la hora de tomar decisiones conjuntas.
- **R4: Pérdida de información:** Las características del estudio requieren la disposición de múltiples fuentes de información y desarrollo, tanto documentos electrónicos y físicos como de programas informáticos de diferente naturaleza. La pérdida de información, por tanto, puede tener lugar debido a casuísticas como una mala organización, fallos de sistema o errores derivados del factor humano.
- **R5: Avería de los equipos informáticos:** En el caso de ocurrencia de pérdida del equipamiento utilizado para el manejo de las herramientas del proyecto, la planificación puede sufrir cambios súbitos y substanciales, pudiendo incluso conllevar al fracaso de esta.

7.2. Evaluación y Priorización de los Riesgos

El proceso de evaluación y priorización de riesgos toma como guía la probabilidad de ocurrencia de las situaciones planteadas en el apartado anterior y su impacto sobre el Trabajo Fin de Máster. En primer lugar, la probabilidad de ocurrencia de los riesgos se clasifica de muy baja a muy alta como se indica en la Tabla 59.

Probabilidad de ocurrencia	Valor
Muy baja	0,1
Baja	0,25
Media	0,25
Alta	0,75
Muy alta	0,9

Tabla 59: Probabilidad de ocurrencia de los riesgos.

Del mismo modo, el impacto de ocurrencia de los riesgos sobre el desarrollo del proyecto se valorará de muy leve a muy grave siguiendo la Tabla 60.

Impacto	Valor
Muy leve	0,1
Leve	0,25
Medio	0,25
Grave	0,75
Muy grave	0,9

Tabla 60: Impacto de los riesgos sobre el trabajo.

En definitiva, la estimación de los riesgos se realiza en base a los dos factores descritos en la Tabla 59 y en la Tabla 60 como se recogen de forma resumida en la Tabla 61.

Riesgo	Probabilidad de ocurrencia	Impacto
R1: Imposibilidad técnica de implementación	Baja	Muy grave
R2: Superación de fechas en la planificación	Moderada	Muy grave
R3: Fallos de comunicación	Muy baja	Grave
R4: Pérdida de información	Muy baja	Muy grave
R5: Avería de los equipos informáticos	Baja	Medio

Tabla 61: Probabilidad de ocurrencia e impacto por riesgo.

La matriz probabilidad – impacto resultante para la priorización de los riesgos en función de los dos parámetros definidos se muestra en la Tabla 62.

Impacto \ Probabilidad	Muy baja	Baja	Moderada	Alta	Muy alta
Muy leve	0,0100	0,0250	0,0500	0,0750	0,0900
Leve	0,0250	0,0625	0,12500	0,1875	0,2250
Medio	0,0500	R5 0,1250	0,2500	0,3750	0,4500
Grave	R3 0,0750	0,1875	0,3750	0,5625	0,6750
Muy grave	R4 0,0900	R1 0,2250	R2 0,4500	0,6750	0,8100

Tabla 62: Matriz probabilidad – impacto.

7.3. Plan de Contingencia

El orden de prioridad que ese obtiene como resultado de la matriz probabilidad – impacto (Tabla 62) define la secuencia de medidas a tomar en forma de plan de contingencia para los riesgos del proyecto:

- **R2: Superación de fechas en la planificación**

Para **disminuir el riesgo**, se define un plan de prevención basado en la planificación iterativa, que separa el proyecto en fases de pequeño tamaño y se contrasta cada paso con la directora solicitando consejo para realizar ajustes con cada avance en el proyecto. De este modo, se posibilita un seguimiento efectivo del estado del trabajo. Además, se asigna un margen de tiempo holgado en la realización de tareas de mayor complejidad.

En caso de que tuviera lugar la ocurrencia del riesgo, el **procedimiento a seguir** es el contacto con la dirección del proyecto y un reajuste parcial que trate de afrontar los posibles retrasos en tareas específicas teniendo el menor impacto posible.

- **R1: Imposibilidad técnica de implementación**

El **plan de prevención** para disminuir este tipo de riesgo es el conocimiento profundo del ámbito de estudio de la investigación, para lo que se establece un plan formativo previo al desarrollo de la parte práctica del proyecto.

En caso de tener que lidiar con la situación de no ser posible implementar alguna característica, el plan es **afrontar dicha situación** en dos fases. La primera, búsqueda de soluciones alternativas y valoración del impacto en la planificación. Después, conllevaría a establecer una reunión con la directora para comunicar la situación, y presentar las opciones disponibles.

- **R5: Avería de los equipos informáticos**

La **prevención de averías** en equipos informáticos se realiza mediante un uso cauteloso y correcto de los dispositivos. Al tratarse de ordenadores portátiles, no requieren un cuidado especial ni procedimientos específicos.

El **plan de contingencia** supone definir un presupuesto que permita adquirir el equipamiento necesario para realizar las tareas del trabajo.

- **R4: Pérdida de información**

El **plan de prevención** para la pérdida de información se fundamenta en la realización de copias de seguridad de los documentos y archivos de proyecto. Para ello, se dispone de dos equipos informáticos en los que almacenar dichos datos y acceso a plataformas gratuitas en la nube.

En caso de pérdida significativa de información, la única **alternativa** sería buscar la última copia realizada y retomar el proyecto en dicho punto. Si se diera la pérdida total de la información, se debería empezar el trabajo completo desde el principio.

- **R3: Fallos de comunicación**

Debido al conocimiento previo del equipo y participación en diferentes trabajos en el pasado, el riesgo de fallo de comunicación tiene una probabilidad muy baja. En consecuencia, se considera un riesgo asumible.

El plan de contingencia en caso de darse la casuística de ocurrir fallos de comunicación es establecer un encuentro con la dirección del proyecto y gestionar la situación mediante comunicación directa.

8. Metodología

En esta sección se describe el protocolo a seguir para el desarrollo del estudio en función de la selección de alternativas del apartado previo. El procedimiento se lleva a cabo en base a dos grandes tareas.

En primer lugar, se diseñan e implementan los **entornos inalámbricos** –doméstico, empresarial, interior de alta densidad y exterior de alta densidad– en **NS-3**, incluyendo tanto los aspectos de **caracterización del escenario** como el **tráfico** presente en la red. Acto seguido, se realiza la obtención de muestras a través de múltiples simulaciones.

Después, el **procesado de los datos** se lleva a cabo mediante el diseño e implementación de un programa que recoge la salida de la simulación, la preprocesa y aplica los **algoritmos de clustering** para analizar su aptitud sobre las dos propuestas de procedimientos de mejora de la QoS.

Como se ha expresado a lo largo del documento, la descripción de escenarios WLAN requiere un conocimiento previo considerable del campo de estudio. En consecuencia, se ha diseñado el documento que guía este proyecto de forma que se favorezca una **introducción incremental** al tema principal.

En un inicio, resulta necesario adquirir una **visión general** de los casos de uso de la tecnología, para lo que en los apartados de **Introducción y Contexto** se muestra una perspectiva global de la situación en la que se encuentran las redes y el protocolo IEEE 802.11.

Una vez introducido el ámbito del estudio, en el **Estado del Arte** se realiza la **profundización** en los temas centrales, posibilitando un estudio de las características con un criterio centrado en la QoS. Además, en dicha sección se recoge de forma resumida el estudio teórico realizado para caracterizar los escenarios, y posteriormente se introducen las técnicas de aprendizaje automático a considerar.

Tras disponer de toda la información a utilizar en el desarrollo del proyecto, se plantean las **alternativas principales** en base a tres problemáticas, las **decisiones de diseño** de los escenarios y el tráfico de red, la selección de KPIs sobre las aplicaciones de interés, y los algoritmos de agrupamiento a evaluar.

Por lo tanto, como **continuación natural** del procedimiento recientemente introducido, el siguiente apartado describe la **aplicación de las soluciones** en base a dos grandes bloques: implementación de los escenarios y procesado de datos.

8.1. Implementación de los Escenarios

Esta sección describe la implementación de los entornos de red en base a las decisiones de diseño.

Con intención de obtener resultados que reflejen distintas casuísticas, se hace uso de variables aleatorias uniformes en la generación de los parámetros cuya definición no se corresponde con un modelo estadístico particular. Por el contrario, cuando se dispone

de valores de referencia sobre valores que resultan variables, se establecen como variables aleatorias normales cuya media coincidirá con dicha referencia.

Por otro lado, debido a la extensión del programa y su complejidad, se reservan las muestras de código con la implementación específica al Anexo III: NS-3. No obstante, en el presente apartado se mencionan los cambios más significativos realizados para la adecuación de las herramientas del simulador al caso de uso en estudio.

El primer escenario por implementar es el **entorno residencial**, que cubre el despliegue de la tecnología IEEE 802.11 sobre entornos del hogar de distinto tipo. El modelo del escenario que se desea simular puede representarse según la Figura 29. Los valores específicos de los parámetros principales de simulación se hallan recogidos en la Tabla 63.

Como se observa en el diagrama, se describen entornos con forma variable siguiendo una estructura rectangular—en el diagrama se muestra de forma cuadrada debido a que solo se representan las medidas mínimas y máximas— y cuyas medidas utilizadas se extraen del análisis de alternativas.

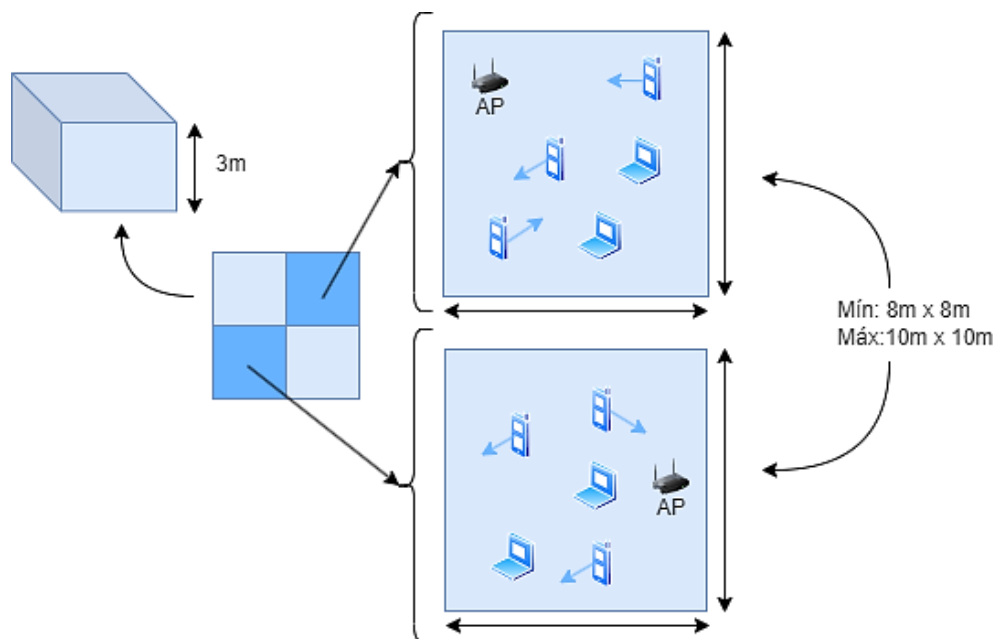


Figura 29: Diagrama representativo del entorno residencial.

Para su implementación, se hace una definición del espacio mediante el ajuste de la posición y movilidad de los nodos en el área seleccionada y según el patrón de movilidad elegido. Se utiliza generación aleatoria sobre la definición de los tamaños físicos de cada red dentro de los márgenes especificados, y se asigna un modelo RandomWalk sobre las STAs. Estas últimas se generan atendiendo también a una semilla aleatoria y ejecución concreta, mediante distribuciones normales con valor medio y varianza —el modelo estadístico de NS-3 toma como parámetro la varianza en lugar de la desviación típica— establecidos basados en las conclusiones del análisis de alternativas. En concreto, se toma una media de 10 dispositivos por red y una varianza de 15. La asignación de tráfico se realiza de forma posterior.

Con relación a la configuración de la transmisión, se hace uso de la banda frecuencial de 5 GHz, y se asignan los canales de 80 MHz a emplear mediante una variable aleatoria que establece a cada red un canal, que puede coincidir o no con el resto de las redes del entorno. La asignación de potencia de transmisión de los dispositivos se realiza mediante un modelo PHV que varía la potencia entre los márgenes especificados.

Los aspectos concretos del protocolo IEEE 802.11 se definen de forma estándar en su mayoría. Para ello, se especifica la versión en uso, IEEE 802.11ax, y entre las configuraciones aplicadas, cabe mencionar que no se hace uso de RTS/CTS al no tratarse de un entorno en el que su aplicación tenga grandes bondades y se establece un intervalo de guarda de 800 ns.

Grupo de características	Parámetro característico	Entorno residencial
Características generales del entorno	Forma y dimensión	Rectangular, representación de espacios con dimensiones que varían entre 8m x 8m y 10m x 10m
	Descripción de los APs	Un AP por red con una única interfaz de red, posicionado aleatoriamente a una altura de 1,5m y sin movilidad
	Descripción de las STAs	Una cantidad aleatoria de alrededor de 10 STAs (valor medio) por red con una única interfaz, posicionados aleatoriamente a una altura de entre 1m y 1,5m y con movilidad Random Walk
Caracterización de la transmisión	Aspectos frecuenciales generales	Banda de 5 GHz, canales de 80 MHz asignados de forma aleatoria
	Configuración de los APs	Configuración de antenas 2x2 con hasta 2 flujos espaciales, y un modelo de potencia variables entre 10 dBm y 21 dBm
	Configuración de las STAs	Configuración de antenas 1x1 con un único flujo espacial, y un modelo de potencia variables entre 10 dBm y 15 dBm
	Modelo de propagación	Modelo específico basado en LogDistance con tres distancias de referencia y pérdidas por obstáculos como suelos y paredes
Caracterización del protocolo IEEE 802.11	EDCA	Se habilita su utilización con valores estándar
	Agregación	A-MPDU con tamaño de buffer estándar (64)
	Número de repeticiones	Valor estándar
	OFDM	DL/UL OFDMA, 800ns GI
	Tiempos IFS	Estándar
	RTS/CTS	No

Tabla 63: Parámetros de implementación del entorno residencial en NS-3.

En relación con el **tráfico de red** (Tabla 64), se realiza un modelado del tráfico VoD en base a implementación planteada en [96] pero llevando a cabo una adaptación a las redes inalámbricas según la especificación de tráfico planteada por el TGax.

Asimismo, se establece un servidor HTTP, cuya implementación se lleva a cabo por métodos específicos de NS-3 y se instala en el AP. La parte de cliente se instala en las STAs. Para simular el caso de uso de forma específica –debido a que la red entre el servidor HTTP y el cliente HTTP induce casuísticas diferentes en términos de retardo y velocidad– se siguen las recomendaciones de los modelos planteados por TGax.

Los demás casos de tráfico –FTP, VoIP y Gaming– se han introducido aprovechando la implementación realizada por CTTC [97] para su estudio sobre 5G pero adaptando el modelo propuesto para FTP. La propuesta de la CTTC se realiza según la 3GPP en su especificación TR36.814 [98]. De este modo, se toma como referencia el modelo M1 de la documentación de la 3GPP, y para adaptarlo al planteamiento de TGax [72] de modelado de tráfico, se diseña el servidor FTP M2 [98] introduciendo tiempos de lectura de los ficheros como se definía en Características Principales de los Tipos de Tráfico y tamaños de fichero variables.

Los valores específicos utilizados y los modelos estadísticos aplicados en cada caso pueden consultarse en Modelado de los tipos de tráfico dentro del Anexo II: Selección de alternativas para la parametrización de los escenarios WLAN y en el Anexo III: NS-3.

Tipo de tráfico	Flujo	% AP / % STA
VoD	DL	60% / 60%
	UL	10% / 10%
HTTP	DL	40% / 40%
	UL	0% / 0%
FTP	DL	20% / 20%
	UL	0% / 0%
VoIP	DL	20% / 20%
	UL	20% / 20%
Gaming	DL	30% / 30%
	UL	30% / 30%

Tabla 64: Referencias de implementación del tráfico en el entorno residencial en NS-3.

El siguiente escenario en estudio es el **entorno empresarial**, que recoge la simulación de la tecnología IEEE 802.11 sobre entornos de oficina. El modelo por implementar se representa según la Figura 30. Del mismo modo que en el caso anterior, los valores específicos de los parámetros principales de simulación se muestran en una tabla resumen (Tabla 65).

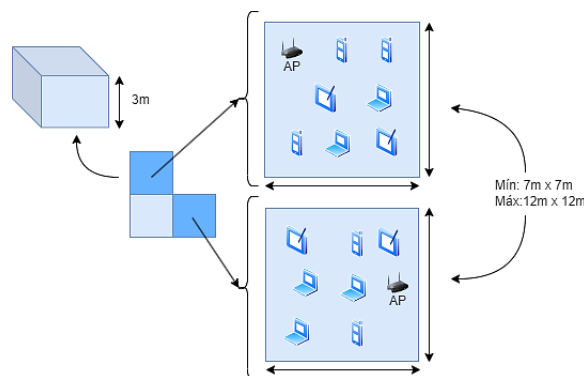


Figura 30: Diagrama representativo del entorno empresarial.

Como se puede apreciar en el diagrama, a simple vista se sigue un modelo similar al anterior, donde se describen entornos con forma variable siguiendo una estructura rectangular, pero se diferencia en la densidad de dispositivos presentes en la red y la movilidad de estos.

Se hace uso de generación aleatoria para el establecimiento de los tamaños físicos de cada red dentro de los márgenes de este tipo de entornos, y, en este caso, se asigna un modelo sin movilidad sobre las STAs. Estas últimas se generan de forma aleatoria, y la asignación de tráfico se realiza de forma posterior.

Grupo de características	Parámetro característico	Entorno residencial
Características generales del entorno	Forma y dimensión	Rectangular, representación de espacios con dimensiones que varían entre 7m x 7m y 12m x 12m
	Descripción de los APs	Un AP por red con una única interfaz de red, posicionado aleatoriamente a una altura de 1,5 m y sin movilidad.
	Descripción de las STAs	Una cantidad aleatoria de alrededor de 30 STAs por red con una única interfaz, posicionados aleatoriamente a una altura de entre 1 m y 1,5 m y sin movilidad.
Caracterización de la transmisión	Aspectos frecuenciales generales	Banda de 5 GHz, canales de 80 MHz asignados de forma determinista en canales separados.
	Configuración de los APs	Configuración de antenas 2x2 con hasta 2 flujos espaciales, y un modelo de potencia variables entre 10 dBm y 24 dBm.
	Configuración de las STAs	Configuración de antenas 1x1 con un único flujo espacial, y un modelo de potencia variables entre 10 dBm y 15 dBm.
	Modelo de propagación	Modelo definido por 3GPP para entornos de oficina (3GPP Indoor Office)
Caracterización del protocolo IEEE 802.11	EDCA	Se habilita su utilización con valores estándar.
	Agregación	A-MPDU con tamaño de buffer estándar (64)
	Número de repeticiones	Valor estándar
	OFDM	DL/UL OFDMA, 800 ns GI
	Tiempos IFS	Estándar
	RTS/CTS	No

Tabla 65: Parámetros de implementación del entorno empresarial en NS-3.

Por lo demás, la implementación del escenario sigue el mismo procedimiento que en el entorno residencial, a diferencia del número de redes a contemplar por simulación –en este caso se introducen tres redes en lugar de cuatro–.

Respecto a la configuración de la transmisión, se diferencia en que el modelo de asignación de la banda frecuencial en este caso se establece evitando solapamientos entre redes, y también se aprecian variaciones en los valores máximos de la potencia de acuerdo con la Tabla 65. En la misma tabla también se observa que la caracterización

del protocolo IEEE 802.11 se lleva a cabo del mismo modo que para el escenario residencial.

Con relación a la asignación del **tráfico de red** (Tabla 66), se realiza el modelado de igual manera que en el escenario anterior, a excepción del tráfico Gaming, ya que, al no tratarse de un escenario en el que este tipo de modelos sea característico, sus porcentajes de uso son nulos.

Tipo de tráfico	Flujo	% AP / % STA
VoD	DL	20% / 20%
	UL	10% / 10%
HTTP	DL	60% / 60%
	UL	0% / 0%
FTP	DL	20% / 20%
	UL	0% / 0%
VoIP	DL	20% / 20%
	UL	20% / 20%
Gaming	DL	0% / 0%
	UL	0% / 0%

Tabla 66: Referencias de implementación del tráfico en el entorno empresarial en NS-3.

A continuación, se introduce la implementación concreta del **entorno interior de alta densidad**. Como se puede observar en la Figura 31, las diferencias con los dos entornos anteriores son fácilmente apreciables. En este caso, se establecen regiones circulares que disponen de mayor densidad de dispositivos en movimiento y concentración de redes.

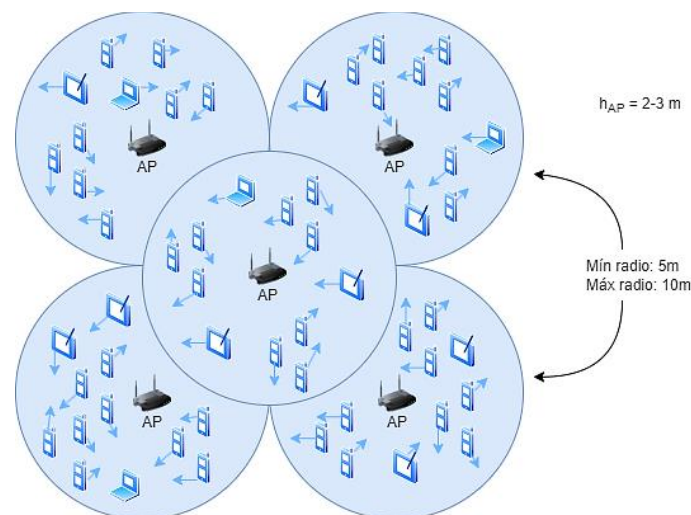


Figura 31: Diagrama representativo del entorno interior de alta densidad.

Para su implementación, se hace una definición del espacio mediante generación aleatoria de las dimensiones de las regiones entre los márgenes especificados, y del mismo modo que en el escenario residencial, se hace uso de un modelo RandomWalk sobre las STAs. Como se puede observar, en este caso se plantea el uso de cinco redes.

Además, se amplía el número máximo de nodos de los usuarios estableciendo el valor medio de los mismos como 50, y la altura de los puntos de acceso también se vuelve

variable. Respecto a los nodos terminales, al igual que en los modelos anteriores se hace uso de distribuciones normales sobre la media.

En la configuración de la transmisión se vuelve a hacer uso de la banda de 5 GHz y canales de 80 MHz, pero se define un uso de canal compartido. Asimismo, se amplía la configuración de antenas a 4x4 para APs y 2x2 para STAs, y se ajusta la potencia según la Tabla 67.

En cuanto a los aspectos concretos del protocolo IEEE 802.11, cabe mencionar que se habilita el uso de RTS/CTS de forma que su umbral sea inferior al tamaño de paquetes a utilizar.

Grupo de características	Parámetro característico	Entorno residencial
Características generales del entorno	Forma y dimensión	Circular, representación de espacios con dimensiones que varían con radios entre 5 m y 10 m
	Descripción de los APs	Un AP por red con una única interfaz de red, posicionado de forma determinista a una altura de entre 2 m y 3 m sin movilidad
	Descripción de las STAs	Una cantidad aleatoria con media de 50 STAs con una única interfaz, posicionados aleatoriamente a una altura de entre 1m y 1,5m y con movilidad Random Walk
Caracterización de la transmisión	Aspectos frecuenciales generales	Banda de 5 GHz, canales de 80 MHz asignados sobre el mismo canal principal
	Configuración de los APs	Configuración de antenas 4x4 con hasta 4 flujos espaciales, y un modelo de potencia variables entre 10 dBm y 21 dBm
	Configuración de las STAs	Configuración de antenas 2x2 con hasta 2 flujos espaciales, y un modelo de potencia variables entre 10 dBm y 15 dBm
	Modelo de propagación	Modelo específico TMB basado en LogDistance con tres distancias de referencia
Caracterización del protocolo IEEE 802.11	EDCA	Se habilita su utilización con valores estándar
	Agregación	A-MPDU con tamaño de buffer estándar (64)
	Número de repeticiones	Valor estándar
	OFDM	DL/UL OFDMA, 800ns GI
	Tiempos IFS	Estándar
	RTS/CTS	Sí

Tabla 67: Parámetros de implementación del entorno interior de alta densidad en NS-3.

La asignación del **tráfico de red** (Tabla 66) sobre los dispositivos se realiza de igual manera que en escenarios anteriores, ajustando los porcentajes a lo establecido en el análisis de alternativas.

Tipo de tráfico	Flujo	% AP / % STA
VoD	DL	60% / 60%
	UL	10% / 10%
HTTP	DL	40% / 40%
	UL	0% / 0%
FTP	DL	10% / 10%
	UL	0% / 0%
VoIP	DL	40% / 40%
	UL	40% / 40%
Gaming	DL	50% / 50%
	UL	50% / 50%

Tabla 68: Referencias de implementación del tráfico en el entorno interior de alta densidad en NS-3.

Por último, se presenta el **entorno exterior de alta densidad** y sus aspectos de implementación. Como se representa en la Figura 31, las características espaciales son similares al entorno interior de alta densidad, a diferencia de la altura de los puntos de acceso, que puede alcanzar valores superiores.

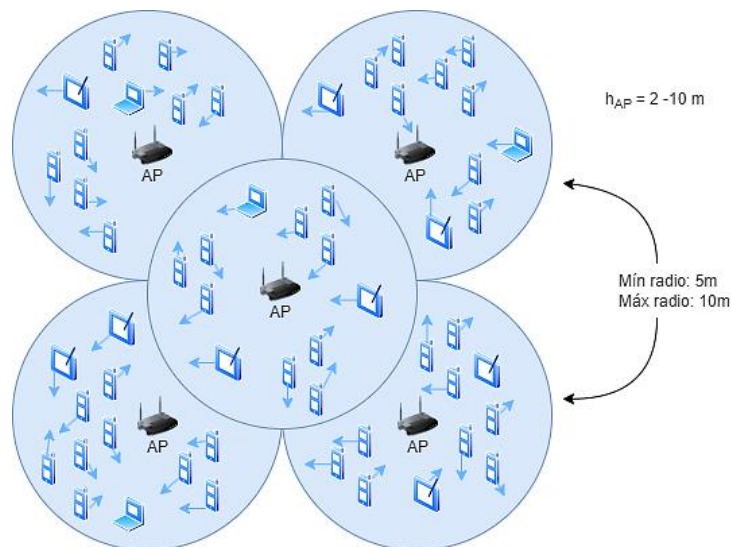


Figura 32: Diagrama representativo del entorno exterior de alta densidad.

Por lo tanto, la generación de las dimensiones de las regiones entre los márgenes especificados tiene lugar de forma aleatoria, y las STAs siguen un patrón de movimiento RandomWalk.

El número medio de nodos de los usuarios asciende a un total de 50 y como en el escenario interior de alta densidad, también se sigue una distribución normal sobre dicho valor.

En términos de configuración de la transmisión, la potencia aumenta en los APs y el modelo de propagación utilizado es el indicado en la Tabla 69. El resto de las características son comunes a lo especificado en el entorno anterior.

Grupo de características	Parámetro característico	Entorno residencial
Características generales del entorno	Forma y dimensión	Circular, representación de espacios con dimensiones que varían con radios entre 5 m y 10 m
	Descripción de los APs	Un AP por red con una única interfaz de red, posicionado de forma determinista a una altura de entre 2 m y 10 m sin movilidad
	Descripción de las STAs	Una cantidad aleatoria con valor medio 50 STAs con una única interfaz, posicionados aleatoriamente a una altura de entre 1m y 1,5m y con movilidad Random Walk
Caracterización de la transmisión	Aspectos frecuenciales generales	Banda de 5 GHz, canales de 80 MHz asignados sobre el mismo canal principal
	Configuración de los APs	Configuración de antenas 4x4 con hasta 4 flujos espaciales, y un modelo de potencia variables entre 10 dBm y 30 dBm
	Configuración de las STAs	Configuración de antenas 2x2 con hasta 2 flujos espaciales, y un modelo de potencia variables entre 10 dBm y 15 dBm
	Modelo de propagación	Modelo LogDistance
Caracterización del protocolo IEEE 802.11	EDCA	Se habilita su utilización con valores estándar
	Agregación	A-MPDU con tamaño de buffer estándar (64)
	Número de repeticiones	Valor estándar
	OFDM	DL/UL OFDMA, 800ns GI
	Tiempos IFS	Estándar
	RTS/CTS	Sí

Tabla 69: Parámetros de implementación del entorno exterior de alta densidad en NS-3.

Del mismo modo, el **tráfico de red** se simula utilizando la asignación sobre los nodos de forma equivalente a los demás escenarios, estableciendo los porcentajes de nodos afectados según la Tabla 70.

Tipo de tráfico	Flujo	% AP / % STA
VoD	DL	80% / 80%
	UL	20% / 20%
HTTP	DL	50% / 50%
	UL	0% / 0%
FTP	DL	10% / 10%
	UL	0% / 0%
VoIP	DL	60% / 60%
	UL	60% / 60%
Gaming	DL	80% / 80 %
	UL	80% / 80 %

Tabla 70: Referencias de implementación del tráfico en el entorno exterior de alta densidad en NS-3.

En definitiva, los parámetros concretos especificados para los cuatro escenarios se integran en NS-3 a través de las múltiples clases de definición de modelos y herramientas de soporte.

En la Tabla 71 se recogen los recursos principales utilizados para la definición de dichos parámetros.

Parámetro característico	Recursos NS-3
Forma y dimensión	Definición de la posición y movilidad: MobilityHelper
Descripción de los APs	Establecimiento de valores aleatorios y constantes en los modelos: UniformRandomVariable ConstantRandomVariable
Descripción de las STAs	Modelo de definición del espacio: RandomRectanglePositionAllocator UniformDiscPositionAllocator Modelo de movilidad: RandomWalk2dMobilityModel ConstantPositionMobilityModel
Aspectos frecuenciales generales	Configuración de la banda y canal sobre el nivel PHY: SpectrumWifiPhyHelper
Configuración de los APs	Configuración de antenas y potencia sobre el nivel PHY: SpectrumWifiPhyHelper
Configuración de las STAs	Configuración de antenas y potencia sobre el nivel PHY: SpectrumWifiPhyHelper
Modelo de propagación	Configuración del canal de comunicación a través del modelo de canal: SpectrumChannelHelper MultiModelSpectrumChannel ConstantSpeedPropagationDelayModel ThreeLogDistancePropagationLossModel ThreeGppIndoorOfficePropagationLossModel LogDistancePropagationLossModel
EDCA	Configuración de prioridades WifiHelper WifiMacHelper
Agregación	Definición de agregación y asignación a clases de tráfico: WifiHelper WifiMacHelper
Número de repeticiones	Configuración del número de repeticiones: WifiMacHelper
OFDM	Configuración de OFDMA: WifiHelper
Tiempos IFS	Modificación de PIFS, SIFS, etc.: SpectrumWifiPhyHelper
RTS/CTS	Configuración de RTS/CTS: WifiRemoteStationManager

Tabla 71: Recursos principales utilizados para la caracterización de escenarios en NS-3.

Del mismo modo, las herramientas utilizadas para la especificación de tráfico se resumen en la Tabla 72. Se debe tener en cuenta que algunas de ellas requieren la

instalación de módulos externos, como se indicaba anteriormente, y que, en el caso de HTTP, a pesar de que la transmisión de objetos web sea en la dirección de servidor a cliente, también se genera cierto tráfico en sentido contrario con las confirmaciones de llegada de paquetes.

Una vez implementados los escenarios, se procede a la realización de las simulaciones. Para ello se utiliza un script que lanza las simulaciones variando la semilla aleatoria y ejecución. Dicho script se incluye también en el Anexo III: NS-3. Además, la salida se genera por terminal y se vuelva a ficheros CSV (Comma Separated Values), uno con la información de los dispositivos, y otra con la de los flujos de tráfico.

Tipo de tráfico	Recursos de transmisión de tráfico en NS-3	Recursos de recepción de tráfico en NS-3
VoD	OnOffHelper	PacketSinkHelper
HTTP	ThreeGppHttpServerHelper ThreeGppHttpClientHelper	ThreeGppHttpServerHelper ThreeGppHttpClientHelper
FTP	ThreeGppFtpM2Helper	PacketSinkHelper
VoIP	TrafficGeneratorHelper TrafficGeneratorNgmnVoip UdpSocketFactory	PacketSinkHelper
Gaming	TrafficGeneratorHelper TrafficGeneratorNgmnGaming UdpSocketFactory	PacketSinkHelper

Tabla 72: Recursos principales utilizados para la simulación de tráfico en NS-3.

8.2. Procesado de Datos

El procesado de los datos obtenidos como resultados de simulación debe hacerse en al menos dos fases: el **preprocesado de los datos** y la **aplicación de los algoritmos de clustering**. Se reservan las muestras de código de ambos procedimientos al Anexo IV: Procesamiento de Resultados a través de Clustering.

En la primera fase de **preprocesado**, se realiza una selección de la información para adecuarla a la diferenciación de entornos y clasificación de perfiles y se favorece la formación de clústeres de la forma más precisa posible.

A continuación, el segundo procedimiento se basa en implementar los **algoritmos de clustering**, ajustando los parámetros de entrada de los algoritmos gracias al conocimiento del problema sobre el que se aplican y distintas técnicas para el ajuste previo de sus configuraciones.

El tratamiento de datos se ha realizado mediante diferentes librerías del lenguaje de programación Python, una herramienta conocida en la aplicación de técnicas de aprendizaje automático. En concreto, las librerías (Figura 140) utilizadas se describen a continuación.

Pandas [99] es una librería de análisis de datos que toma como unidad fundamental un DataFrame. Dicha estructura de datos es una colección de filas y columnas ordenadas con etiquetas, similar a una tabla de una base de datos, y que permite la inserción de elementos de distinta clase sobre distintas columnas. En el ámbito de aplicación de este

proyecto, Pandas facilita la estructura de datos sobre la que introducir los parámetros extraídos de la salida del simulador.

Numpy [100] es una librería ampliamente utilizada en el análisis científico-informático en Python, y provee estructuras de datos complejas que posibilitan la realización de cálculos de forma eficiente. En particular, en este trabajo permite volcar la información de interés desde los DataFrames.

Scikit-learn [101] es una librería de aprendizaje automático que influye múltiples algoritmos de agrupamiento. Además, permite interoperabilidad con Numpy.

Con propósito distinto, también se utilizan las librerías **Matplotlib** [102] y **Seaborn** [103], dos librerías dedicadas a la representación gráfica, que permiten realizar gráficos en distintos formatos y con una interfaz de alto nivel de uso sencillo.

8.2.1. Preprocesado de los Resultados de Simulación

El **preprocesado de los datos** a utilizar en los algoritmos de agrupamiento supone un paso fundamental para el análisis de las estadísticas, ya que se permite mejorar la calidad de los datos –al poder detectar valores inusuales o erróneos– y adaptar las muestras de forma que se obtengan resultados más precisos.

La salida del simulador se ha configurado de forma que se generen **ficheros CSV** de dos tipos: estadísticas por dispositivo y estadísticas por flujo. Una breve muestra de ambos se recoge en la Figura 33 y en la Figura 34, respectivamente.

device	address	avgthroughput	avgtxbytes	avgtxpackets	avgrxbytes	avgrxpackets	avgtotaldelay	avgtotaljitter	snr
STA	192.168.1.4	0.0757724	85244	212	82373	205	37.2658	20.8713	36.8826
STA	192.168.2.3	0.0730142	82141	209	71485	180	58.9238	19.1785	40.9828
STA	192.168.3.35	1.57984	1777315	1257	1767693	1234	21.3634	0.0376439	27.248
AP	192.168.3.52	2.55116	2870051	2082	497380	374	7.01905	12.2026	28.9584
STA	192.168.4.3	0.0391609	44056	133	35405	111	1.64921	0.777949	32.3483

Figura 33: Muestra de fichero (versión reducida) con resultados de simulación a nivel de dispositivo.

flow	sourceip	sourceport	destip	destport	protocol	txpackets	txbytes	rxpackets	rxbytes	avgthroughput	avgdelay	avgjitter
256	192.168.3.13	49184	192.168.3.45	80	TCP	61	9480	60	9424	0.00764889	6.50667	2.20033
118	192.168.3.5	49183	192.168.3.45	5350	UDP	277	18136	235	15530	0.0113095	66.0353	29.524
518	192.168.4.45	49185	192.168.4.51	5094	UDP	7590	995808	3530	521028	4.68298	1.34952	0.16623
1038	192.168.5.46	80	192.168.5.23	49152	TCP	181	261604	181	261604	5.69083	21.2074	0.45215

Figura 34: Muestra de fichero (versión reducida) con resultados de simulación a nivel de flujo de tráfico.

El **volcado de los ficheros al programa** de tratamiento de datos (extracto de código en la Figura 141 del Anexo III) se hace mediante métodos de la librería Pandas, que permite trasladar la información a una estructura de datos DataFrame sobre la que se pueden hacer cálculos iniciales.

En un comienzo, resulta interesante obtener histogramas básicos (extracto de código en la Figura 141 del Anexo III) sobre los datos de salida del simulador, de forma que se pueda favorecer un mejor ajuste de los algoritmos en base a las distribuciones aproximadas de los mismos.

Gracias a la representación visual –un ejemplo de ello puede observarse en la Figura 35– se pueden adquirir nociones del reparto de muestras y su posible similitud con distribuciones de probabilidad.

Sin embargo, la presencia de ruido en los datos –como los dispositivos sin flujos de tráfico asignados– y las diferencias de escala –bytes, megabytes, decibelios, milisegundos, etc.– pueden hacer de esta primera observación algo confuso. Por lo tanto, es conveniente aplicar un primer ajuste a los datos de entrada de los algoritmos, ya que estos provienen de medidas que utilizan unidades y rangos muy distintos.

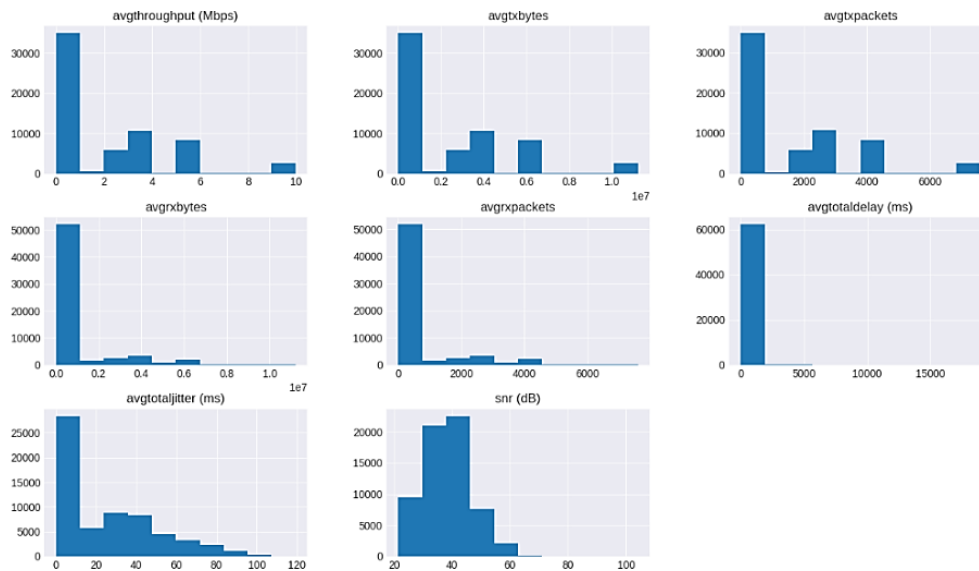


Figura 35: Histograma de los resultados totales sin normalizar de las simulaciones en función del parámetro de medida.

Para dar solución a dicha situación pueden emplearse distintos métodos de preprocesado de datos. La **normalización de datos** (extracto de código en la Figura 142 del Anexo III) posibilita igualar la escala de los atributos, fomentando un reparto equitativo de la influencia de los parámetros de medida de calidad en la formación de clústeres.

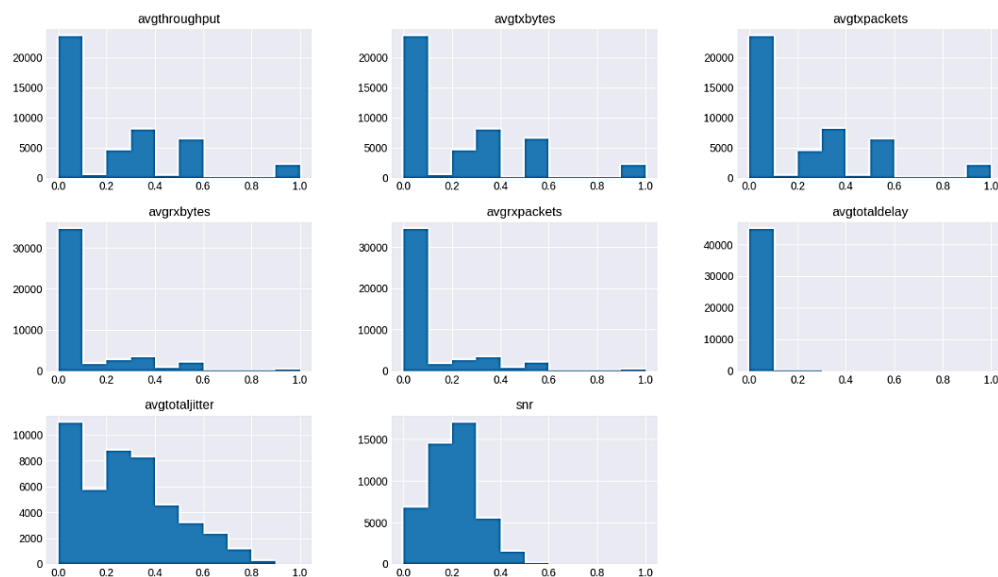


Figura 36: Histograma de los resultados totales preprocesados de las simulaciones en función del parámetro de medida.

Del mismo modo, la normalización de datos puede facilitar la convergencia de los algoritmos de agrupamiento, ya que la formación de grupos basados en distancia se ve afectada por la escala de datos en uso.

Además, dicho procedimiento puede resultar también útil junto con el **descarte de muestras ruidosas** sobre los histogramas básicos de los resultados. Ejemplo de ello es la Figura 36, donde se puede apreciar, en mayor o menor medida en función de cada métrica, variaciones en el reparto de valores entre muestras. En este caso, se consideran como ruido las muestras de datos donde no hay flujos de red de ningún tipo, ya que el estudio se define para periodos de actividad de los dispositivos.

Asimismo, la **distinción entre dispositivos** gracias a las etiquetas de los datos de salida del simulador permite diferenciar las gráficas anteriores en función de si se trata de un AP o una STA. De este modo, la Figura 37 muestra el desglose de valores de las métricas para ambos tipos de equipos. En ella se pueden vislumbrar formas diferentes para los indicadores de rendimiento en función del dispositivo.

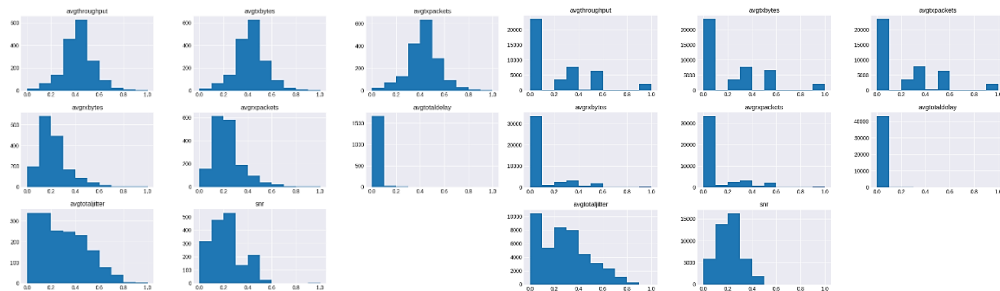


Figura 37: Histograma de los resultados de los AP (izq.) y STA (der.) con valores normalizados en función del parámetro de medida.

Más aún, en relación con el estudio de escenarios, se pueden obtener gráficos indicativos del reparto de muestras por entorno en función del **cruce de distintas métricas** (extracto de código en la Figura 141 del Anexo III).

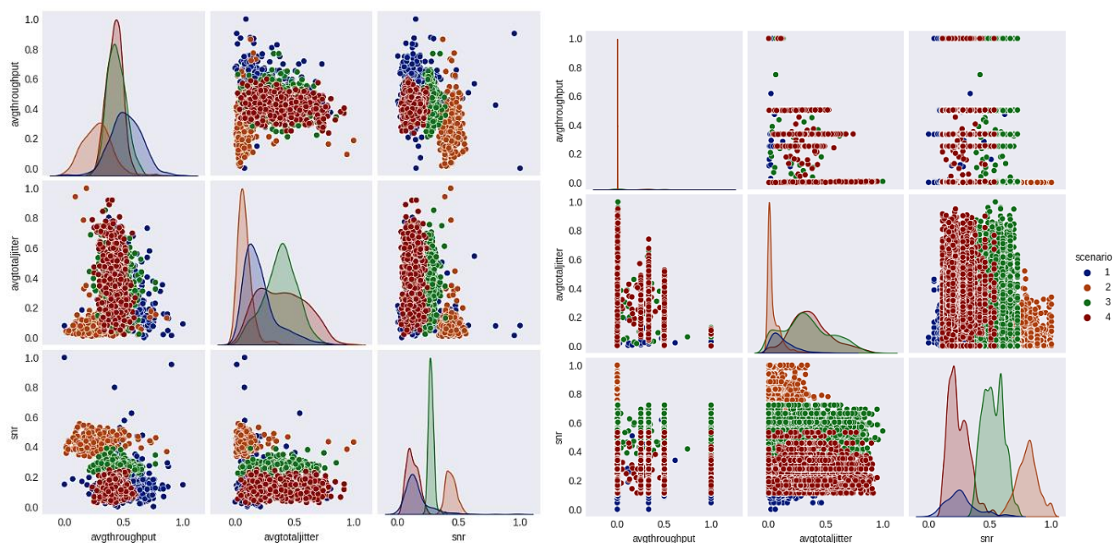


Figura 38: Cruce de variables de los resultados de los APs (izq.) y STAs (der.) con valores normalizados.

Muestra de ello es la Figura 38, donde se pueden observar los puntos en planos bidimensionales donde cada eje recoge la información de un parámetro y sobre los cuales se diferencian los escenarios según lo indicado en la leyenda lateral.

Por último, además de los recursos anteriores, la Figura 39 muestra en un **gráfico tridimensional** la disposición de puntos sin etiquetar (extracto de código en la Figura 143 del Anexo III) para las métricas obtenidas de APs y STAs en base a distintas dimensiones.

Como se puede apreciar, sin disponer de la información de los grupos a formar por los algoritmos de forma previa, resulta complicado entrever qué muestras pueden pertenecer a cada grupo, y la validez de cada algoritmo en base a su ajuste a la característica del entorno de red a extraer.

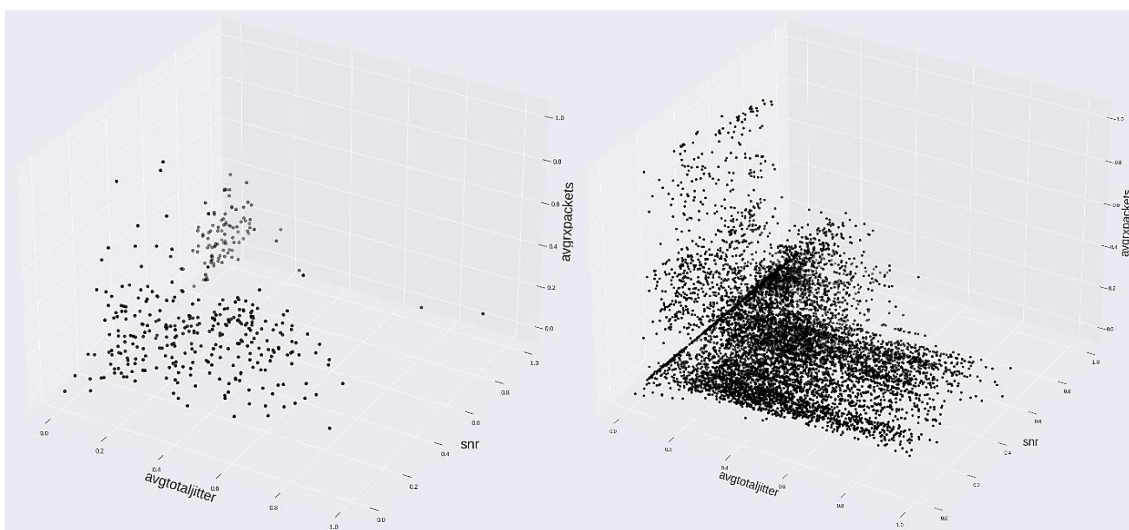


Figura 39: Representación tridimensional de métricas de APs (izq.) y STAs (der.) en base a distintos ejes después del preprocesado.

8.2.2. Implementación de los Algoritmos de Clustering

Una vez llevada a cabo la fase del procesado inicial de datos, estos se hallan preparados para ser introducidos en los algoritmos de agrupamiento y realizar así las tareas de diferenciación de entornos y clasificación de perfiles. Con dicho fin, se utilizan las columnas correspondientes a los parámetros de interés seleccionados para cada tarea de los resultados de simulación preprocesados.

En concreto, en ambos casos se hace uso de un total de 6 variables características del set de datos, pero para APs en diferenciación de escenarios y STAs en clasificación de perfiles, respectivamente. Dichos parámetros son valores medios del throughput, paquetes transmitidos y recibidos, retardo, jitter y relación señal a ruido.

Bajo esta premisa, a continuación, se recoge la implementación de los algoritmos de clustering resultantes del análisis de alternativas.

8.2.2.1. Implementación de K-Means

Como se describe en el apartado dedicado al **clustering particional**, la aplicación del algoritmo K-Means requiere una correcta selección del **parámetro K**, correspondiente al número de clústeres.

Aunque en este estudio el número de clústeres es conocido de forma previa a la agrupación en el estudio de escenarios, es interesante realizar una aproximación a los valores de K óptimos para dicho procedimiento, además de resultar necesario para la clasificación de perfiles según la definición del experimento.

Por consiguiente, se aplica el método de estimación de número de grupos óptimo a través del análisis de la variación de la distancia entre los centroides [104] y un rango de posibles grupos previamente definidos. Debido a su representación gráfica, es denominado **Elbow Curve** o **curva de quiebre** (extracto de código en la Figura 145 del Anexo III), en la que se trata de localizar un punto de inflexión. En esta aplicación concreta, se trata de hallar el punto de quiebre que favorezca la estabilización del número de clústeres de los escenarios o perfiles de usuario.

En la Figura 40 se muestra la comparativa entre las curvas resultantes para el caso de APs y STAs. A simple vista, se puede apreciar que en el caso de las métricas de puntos de acceso el punto de codo está entre los valores K=2 y K=4. Del mismo modo, en el caso de las estaciones se observa un cambio de tendencia entre los valores K=2 y K=4.

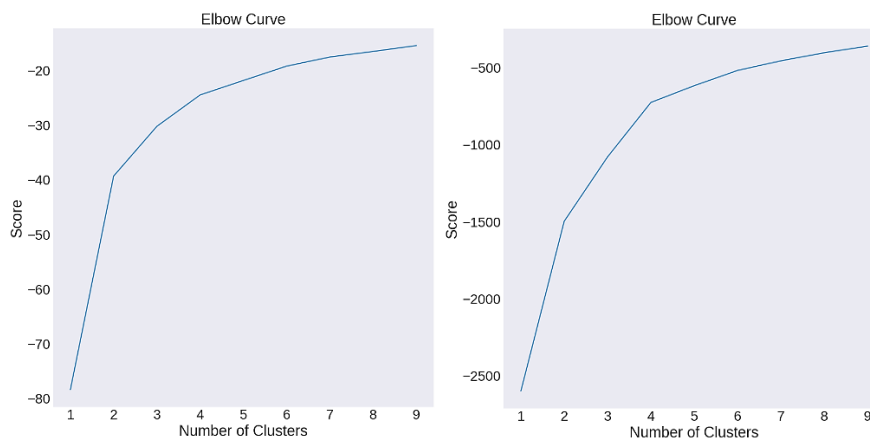


Figura 40: Curvas de quiebre para las métricas de APs (izq.) y STAs (der.).

Tras analizar los valores óptimos de K, se toma la decisión de cuál utilizar en cada caso. En el caso de la **diferenciación de entornos**, se utiliza el parámetro K=4 ya que el número de escenarios es conocido, y se pretende evaluar la adecuación de los diferentes algoritmos a los diferentes entornos.

En la **clasificación de perfiles**, sin embargo, el experimento se basa en una aproximación distinta, ya que se persigue hacer una comparativa basada en el ajuste óptimo de los algoritmos y los perfiles de tráfico presentes en los distintos escenarios. Por lo tanto, e igual que en el caso de la diferenciación de entornos, se hace uso de la estimación K=4.

Una vez seleccionado el valor de K en cada caso, la aplicación del algoritmo resulta sencilla, ya que tan solo ha de definirse el número de inicializaciones del algoritmo en la búsqueda de un ajuste óptimo –por defecto se establece a 10– y el número máximo de iteraciones de este. En la Figura 146 del Anexo III se muestra el fragmento de código utilizado en Python mediante la librería Scikit-Learn.

En el caso de la diferenciación de entornos, se introducen al algoritmo las muestras de estadísticas obtenidas a través de los APs tomando una de las redes de cada escenario como referencia. La entrada del algoritmo compone, por tanto, de los datos normalizados obtenidos del CSV de salida del simulador, junto con el número de clústeres a generar.

De forma sencilla, el algoritmo calcula los centroides para los datos proporcionados a través del valor de K, y a través de dichos elementos, se predicen los grupos a los que pertenece cada muestra del conjunto de datos.

El **resultado de la agrupación** puede entonces representarse de forma gráfica –en dos o tres dimensiones– a través de las librerías anteriormente mencionadas, lo que puede resultar una propuesta más intuitiva que la asociación de valores a cada muestra según los datos de salida del algoritmo. Además, puede hacerse uso de los mismos recursos de representación para mostrar la ubicación de los centroides definidos por K-Means.

En definitiva, la Figura 41 muestra los grupos obtenidos en el caso de agrupación de K=4 para APs y K=4 para STAs. Estos resultados se contrastarán y estudiarán en profundidad en el apartado Análisis de Resultados.

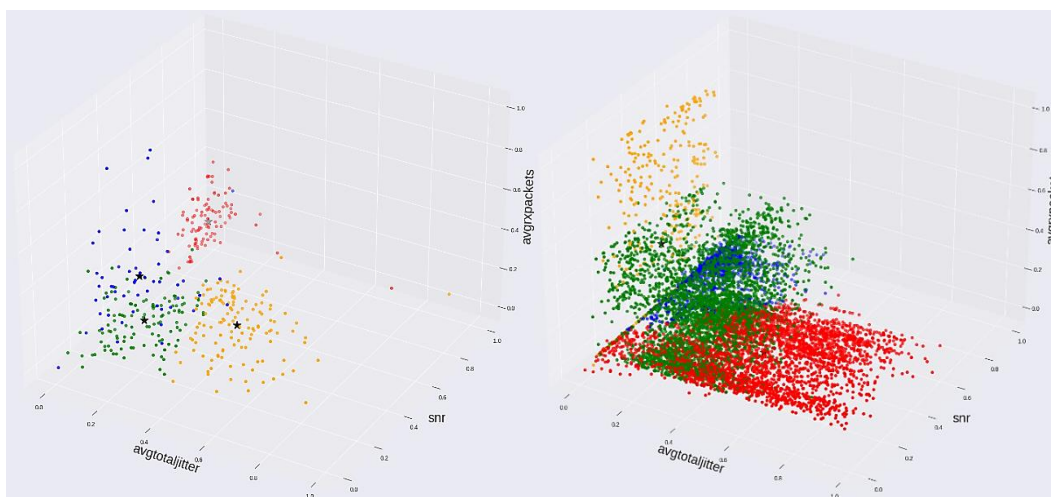


Figura 41: Resultados de agrupación K-Means sobre las métricas preprocesadas de APs (izq.) y STAs (der.).

8.2.2.2. Implementación de DBSCAN

Entre los algoritmos del **clustering basado en densidad**, **DBSCAN** [77] trata de la realización de agrupaciones sobre las muestras tomando como eje dos parámetros principales: épsilon (EPS) y el número mínimo de muestras por clúster.

En este caso no se dispone de un conocimiento previo suficiente sobre los parámetros para ajustar el algoritmo a los dos procedimientos bajo estudio. Por lo tanto, resulta

fundamental una correcta **estimación** de valores que permitan un correcto calibrado de DBSCAN. [105]

La selección de un valor adecuado para el **parámetro ϵ** resulta indispensable para definir el alcance de cada punto, de forma que se establezca un valor capaz de representar la distancia óptima entre puntos para la formación de grupos. Por otro lado, el **número mínimo de muestras** permite caracterizar la conectividad entre puntos, ya que establece un límite inferior en la generación de cada clúster.

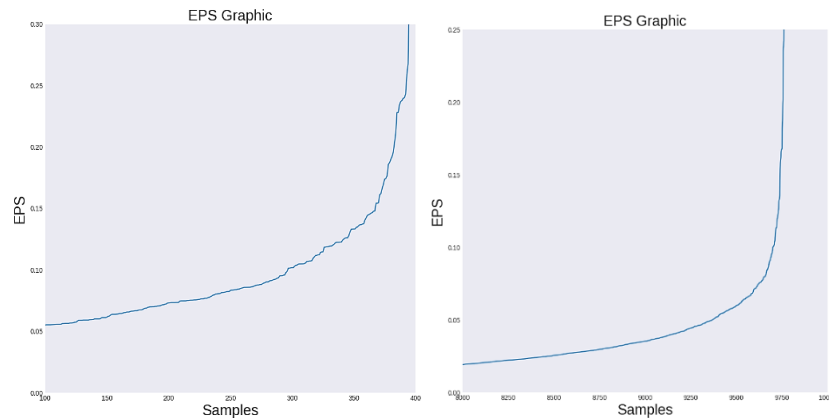


Figura 42: Curvas de estimación de EPS para las métricas de APs (izq.) y STAs (der.).

El primero se calcula siguiendo un método de **cálculo de la distancia entre puntos**, ilustrado en la Figura 42 para los casos de métricas de APs y STAs y cuyo código se muestra en la Figura 148 del Anexo III. El procedimiento realizado calcula la distancia de cada muestra al vecino más cercano, permitiendo obtener una gráfica que muestra una curva de evolución similar a la curva de quiebre de K-Medias. Sin embargo, en este caso se busca el valor para el cual la curva aumenta el número de muestras recogidas para dicho valor de ϵ .

En el caso del número de muestras, se selecciona un **criterio generalista** que se basa en la utilización del número de variables empleadas para definir cada muestra incrementado en una unidad.



Figura 43: Resultados de agrupación DBSCAN sobre las métricas preprocesadas de APs (izq.) y STAs (der.).

A través de los resultados de la Figura 42, se seleccionan valores de EPS similares, que sean próximos al punto de quiebre. Para el estudio de las métricas sobre APs se hace uso de $EPS=0,115$, y en el caso de STAs un valor $EPS=0,121$.

Por otra parte, en la Figura 43 se puede observar la formación de grupos para ambos casos según los valores de ϵ y el número de muestras establecidos anteriormente. Del mismo modo que en K-Means, los resultados se analizarán con mayor detalle en el apartado Análisis de Resultados.

8.2.2.3. Implementación de GMM

Con relación al **clustering basado en la distribución, GMM** [91] requiere disponer del conocimiento de múltiples parámetros para la correcta ejecución del agrupamiento.

Uno de los principales es el **número de componentes** gaussianas –o clústeres– subyacentes en el conjunto de datos. Existen varios métodos para tratar de deducir el número de componentes. En la Figura 151 del Anexo III se muestran los criterios AIC (Akaike Information Criterion) y BIC (Bayesian Information Criterion) [91], que en conjunto tratan de equilibrar la bondad del ajuste del modelo y su complejidad.

En la Ecuación 14 se muestra el criterio AIC, donde k es el número de parámetros presentes en el modelo y L la verosimilitud máxima del modelo.

$$AIC = 2k - 2 \ln(L)$$

Ecuación 14: Criterio AIC [106]

Por otro lado, en la Ecuación 15 se puede observar el criterio BIC, donde n es el número de muestras, k el número de parámetros en el modelo y L la verosimilitud máxima del modelo.

$$BIC = \ln(n)k - 2 \ln(L)$$

Ecuación 15: Criterio BIC [106]

De esta forma, por su propia definición se puede intuir que BIC penaliza en mayor medida la complejidad del modelo que AIC. El procedimiento de estimación, por tanto, se resume en el entrenamiento de múltiples modelos GMM y el cálculo de AIC y BIC para cada uno de ellos. Después, se deberá seleccionar el modelo de menor valor AIC o BIC.

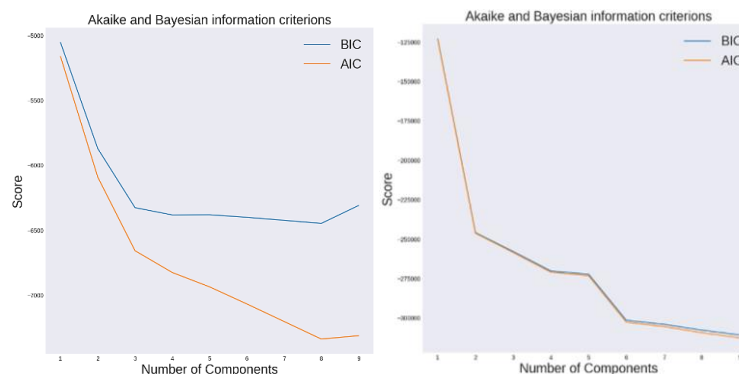


Figura 44: Curvas AIC y BIC para las métricas de APs (izq.) y STAs (der.).

Además del número de componentes, resulta necesario conocer múltiples parámetros, como la **matriz de covarianza**, la **media** y los **pesos** o probabilidades de cada componente gaussiana. Para ello, como se mencionaba en el apartado Clustering basado en la distribución, es común emplear el algoritmo EM (Expectation-Maximization) [91] para estimar los parámetros de GMM durante el proceso de entrenamiento.

En este caso, en la ejecución del algoritmo (Figura 152 del Anexo III) se dispone de un proceso de entrenamiento que incluye el algoritmo EM para estimar los parámetros anteriormente mencionados. El resultado del clustering basado en un número de componentes de 4 para el caso de los APs y 3 para STAs se puede observar en la Figura 45.

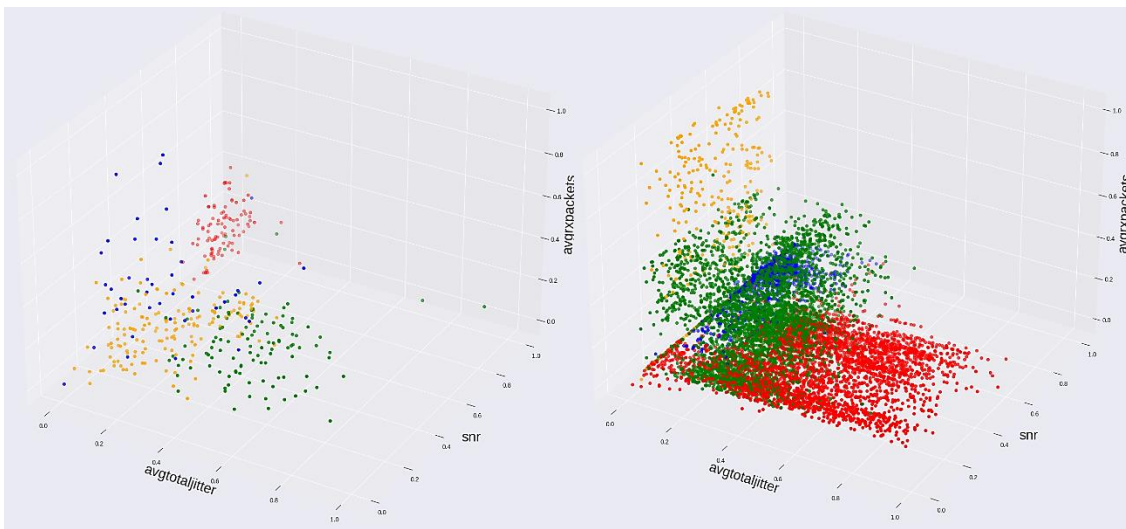


Figura 45: Resultados de agrupación GMM sobre las métricas preprocesadas de APs (izq.) y STAs (der.).

8.2.2.4. Implementación de BIRCH

En último lugar, entre los **algoritmos de agrupamiento jerárquico**, BIRCH [91] basa su proceso de clustering en una estructura de árbol de características que, como se adelantaba en el apartado sobre Clustering jerárquico, requiere dos parámetros principales: el umbral y el factor de ramificación.

En el caso del **umbral**, [91] depende de la naturaleza de los datos y la granularidad a obtener en los clústeres. Un valor menor resultará en grupos menores y más numerosos, y un valor mayor en un menor número de grupos de mayor tamaño. Por lo tanto, se puede proceder utilizando visualización de los datos, o a través de un valor arbitrario sobre el que iterar en base a la calidad del clustering resultante.

Por otro lado, el **factor de ramificación** [91] puede asociarse a la disponibilidad de recursos computacionales, ya que define la profundidad del árbol. De este modo, a mayor factor de ramificación, mayor anchura y menor profundidad del árbol, lo que hace más exigente cada operación debido a la multitud de ramas. Una técnica general para la estimación de este parámetro es empezar con un valor menor e ir aumentándolo de forma gradual manteniendo un equilibrio entre rendimiento y consumo de recursos.

Tras realizar múltiples pruebas sobre diferentes valores de ambos parámetros, se han establecido para el caso de los APs un umbral de 0,085 y un factor de ramificación de 2. Por otro lado, para las STAs se define un umbral de 0.5 y un factor de ramificación con valor 4. El resultado del clustering sobre ambos casos se halla representado en la Figura 68.

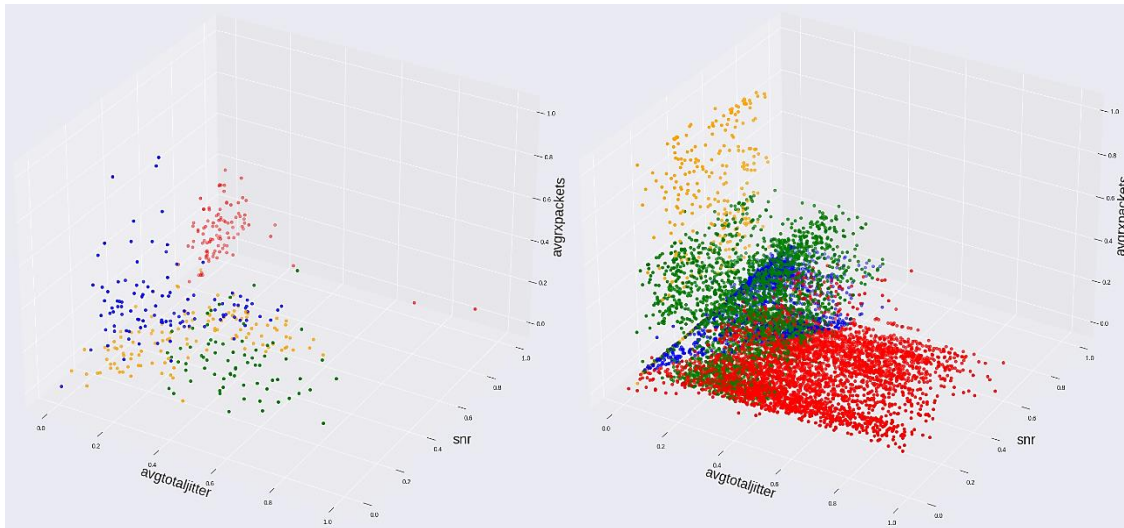


Figura 46: Resultados de agrupación BIRCH sobre las métricas preprocesadas de APs (izq.) y STAs (der.).

9. Análisis de Resultados

Tras haber definido el procedimiento para la puesta en práctica de la metodología, se procede a realizar el análisis de los resultados obtenidos.

El objetivo principal del presente estudio es demostrar la validez de los algoritmos de agrupamiento para la clasificación de entornos y usuarios de la red simulados. Para facilitar el seguimiento del análisis, la Tabla 73 presenta la correlación entre el identificador de cada escenario, el nombre de este, su color de representación en las etiquetas, y el valor asociado a dicho color y escenario en los programas elaborados.

ID Escenario	Escenario	Color	Valor de etiqueta
1	Doméstico	Azul	0
2	Oficina	Rojo	1
3	Interior alta densidad	Verde	2
4	Exterior alta densidad	Naranja	3

Tabla 73: Mapeo de identificadores para el seguimiento del análisis de resultados.

La estructura del análisis de resultados se divide en dos grandes secciones. En primer lugar, se abordará la **diferenciación de entornos**, y posteriormente, la **clasificación de perfiles**. Cada apartado consta de los resultados obtenidos para cada algoritmo y ofrece una comparativa entre ellos para determinar su adecuación a los dos procedimientos.

9.1. Resultados de Agrupación para la Diferenciación de Entornos

El procedimiento llevado a cabo para posibilitar la diferenciación de entornos toma como base las muestras de datos obtenidas de las métricas de los APs en las simulaciones.

Debido a la extensa caracterización de escenarios y la disponibilidad de etiquetas sobre las muestras derivada de esta, el análisis de los algoritmos se llevará a cabo mediante la **comparativa entre los clústeres resultantes** tras la ejecución del **agrupamiento y los grupos formados sobre** las mismas gráficas haciendo uso de **las etiquetas** de escenarios definidas en origen.



Figura 47: Gráficas tridimensionales de las muestras de APs sin etiquetar (izq.) y etiquetadas (der.).

Así, en la Figura 47, la gráfica del lado izquierdo presenta las muestras sin etiquetar conforme a la definición de escenarios establecida, y representa el total de las muestras estudiadas sobre APs sin conocimiento del entorno al que pertenecen. En contraste, y de acuerdo con la Tabla 73, el gráfico derecho representa el mismo conjunto de datos, pero con su respectiva etiqueta del escenario al que pertenecen.

Por lo tanto, a continuación se presentan los resultados de cada algoritmo en estudio para el proceso de realizar dicho etiquetado sin conocimiento previo del escenario.

9.1.1. Resultados de Agrupación de K-Means

Conforme se adelantó en el apartado Implementación de K-Means de la Metodología, las **especificaciones del procedimiento** adoptado para la ejecución de K-Means se encuentran resumidos en la Tabla 74.

Característica		Valor
Parámetro K		4
Número de inicializaciones		10
Número máximo de iteraciones		100
Número de muestras	Total	400
	Escenario 1	100
	Escenario 2	100
	Escenario 3	100
	Escenario 4	100

Tabla 74: Resumen del experimento de K-Means sobre las métricas de APs.

El resultado de la agrupación mediante el algoritmo K-Means para la diferenciación de entornos está representado en la Figura 71, en la cual se compara con los escenarios previamente etiquetados. Además, se muestran los **centroides de los clústeres** resultantes.

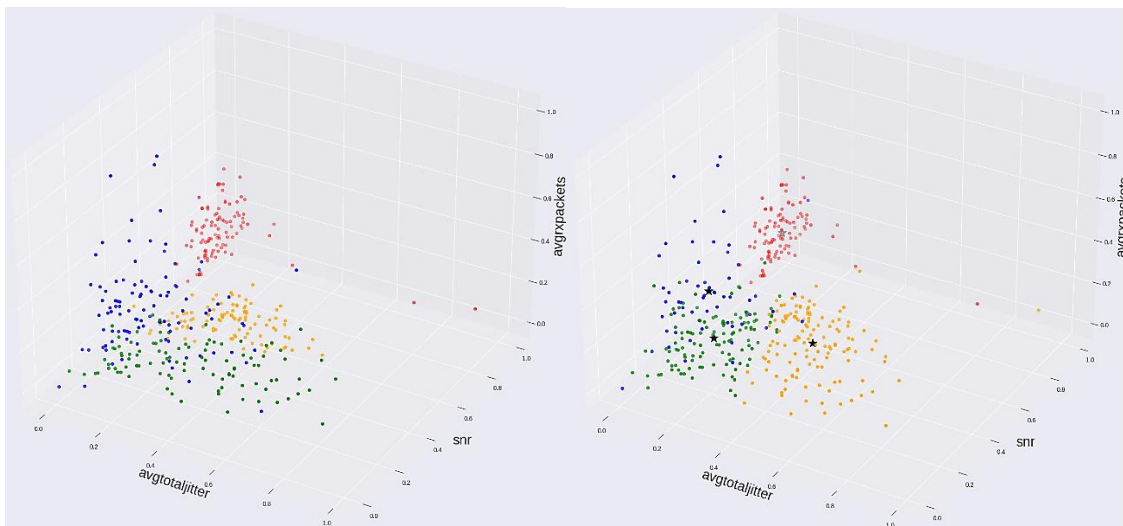


Figura 48: Gráfica tridimensional con las muestras de las métricas de los APs etiquetadas (izq.) y el resultado del agrupamiento de K-Means (der.).

Visualmente resulta evidente que **el algoritmo logra identificar los escenarios de manera parcialmente análoga al resultado esperado**. Específicamente, la

utilización de la distancia como criterio para la conformación de grupos facilita la separación de muestras en cuatro grupos que concuerdan en gran medida con los escenarios establecidos.

La distribución de las muestras obtenidas se detalla en la Tabla 75. En ella, inicialmente se presenta el número total de muestras asignadas por grupo y, posteriormente, la distribución de etiquetas correspondientes a las muestras de cada entorno.

Clasificación	Color	Valor	Cantidad asignada
Total	Azul	0	55
	Rojo	1	99
	Verde	2	131
	Naranja	3	115
Escenario 1	Azul	0	49
	Rojo	1	1
	Verde	2	14
	Naranja	3	36
Escenario 2	Azul	0	1
	Rojo	1	98
	Verde	2	1
	Naranja	3	0
Escenario 3	Azul	0	5
	Rojo	1	0
	Verde	2	67
	Naranja	3	28
Escenario 4	Azul	0	0
	Rojo	1	0
	Verde	2	49
	Naranja	3	51

Tabla 75: Resultados de K-Means sobre las métricas de APs.

Dentro de los resultados presentados en la Tabla 75, cabe destacar que el algoritmo **clasifica con alta precisión el segundo escenario**, mientras que se evidencia una mezcla de muestras entre los escenarios pertenecientes a la alta densidad.

Adicionalmente, uno de los grupos conformados es de menor tamaño en comparación con los demás, lo que resulta en un incremento de desplazamientos de muestras entre grupos, dado que el experimento cuenta con una **cantidad equitativa de muestras** por entorno.

En contraste con lo anterior, la Figura 49 muestra los resultados de K-Means sobre los escenarios de manera bidimensional, permitiendo una **visualización de interacciones entre variables** más diversa. Dentro de las gráficas, es notorio cómo el grupo con etiqueta 1 (color rojo) se configura claramente en función de la distancia, mientras que, en los demás casos, las muestras aparecen más intercaladas.

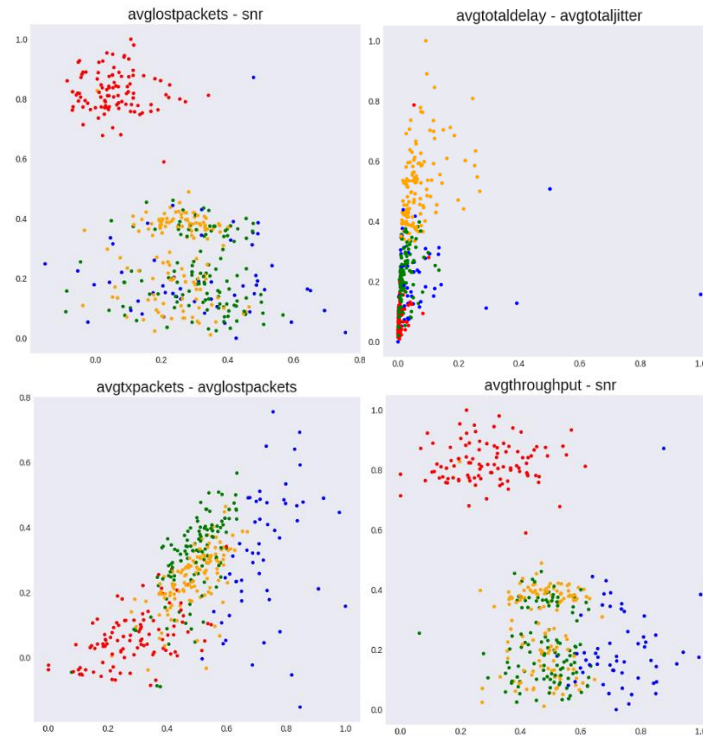


Figura 49: Gráficas bidimensionales del cruce de parámetros de los APs con el resultado de agrupamiento de K-Means.

9.1.2. Resultados de Agrupación de DBSCAN

De acuerdo con lo definido en el apartado Implementación de DBSCAN de la Metodología, los **detalles del procedimiento** seguido para la ejecución de DBSCAN se recogen en la Tabla 76.

Característica		Valor
Parámetro EPS		0,115
Parámetro Mínimo de Muestras		7
Tipo de Distancia		Euclídea
Número de muestras	Total	400
	Escenario 1	100
	Escenario 2	100
	Escenario 3	100
	Escenario 4	100

Tabla 76: Resumen del experimento de DBSCAN sobre las métricas de APs.

El resultado de la agrupación mediante el algoritmo DBSCAN para la diferenciación de entornos se puede observar en la Figura 50, en la que se compara con los escenarios etiquetados de forma óptima.

Asimismo, al disponer DBSCAN de la **capacidad de descartar muestras** en base a los umbrales establecidos en términos de distancia y muestras, se representan dichos resultados a través de puntos de color negro.

En la Figura 50, por tanto, se puede apreciar que el algoritmo es capaz de **extraer parte de los identificadores** de escenarios en base a lo establecido de forma teórica. Sin embargo, a pesar de que visualmente la asociación se **ajuste a los escenarios**, se

realiza un **considerable descarte de muestras** debido a la lejanía de estas respecto a cada grupo.



Figura 50: Gráfica tridimensional con las muestras de las métricas de los APs etiquetadas (izq.) y el resultado del agrupamiento de DBSCAN (der.).

En concreto, la utilización de la densidad de puntos como eje para la clasificación en grupos resulta en la **formación de un total de cuatro clústeres**. Dicha separación coincide con lo esperado.

Un mayor detalle de la distribución de las muestras obtenidas se puede observar en la Tabla 77. Del mismo modo que para el algoritmo anterior inicialmente se recoge el número total de muestras por grupo y después, el reparto de etiquetas sobre las muestras de cada entorno.

Clasificación	Color	Valor	Cantidad asignada
Total	Azul	0	8
	Rojo	1	84
	Verde	2	17
	Naranja	3	145
Escenario 1	Azul	0	7
	Rojo	1	0
	Verde	2	1
	Naranja	3	14
Escenario 2	Azul	0	0
	Rojo	1	84
	Verde	2	0
	Naranja	3	0
Escenario 3	Azul	0	1
	Rojo	1	0
	Verde	2	16
	Naranja	3	40
Escenario 4	Azul	0	0
	Rojo	1	0
	Verde	2	0
	Naranja	3	91

Tabla 77: Resultados de DBSCAN sobre las métricas de APs en formato tabla.

Dentro de los resultados presentados en la Tabla 77, cabe destacar que el algoritmo **reconoce adecuadamente el segundo y cuarto escenario**. Por el contrario, se puede intuir que la alta dispersión de muestras apreciable en la Figura 50 provoca un alto descarte de estas sobre el primer escenario. Además, se aprecian errores de desplazamiento de muestras entre grupos para el caso de escenarios de alta densidad.

A modo de soporte de lo anterior, la Figura 51 representa los resultados de DBSCAN de forma bidimensional. Así, se favorece una **observación en mayor detalle de interacciones entre variables**. Dentro de las diferentes gráficas, se puede apreciar un alto descarte de muestras, en mayor medida para el grupo con etiqueta 0 (color azul). Sin embargo, para el caso de ajuste de grupos cuyas muestras no son descartadas en gran medida, el resultado se asemeja a lo establecido en un inicio.

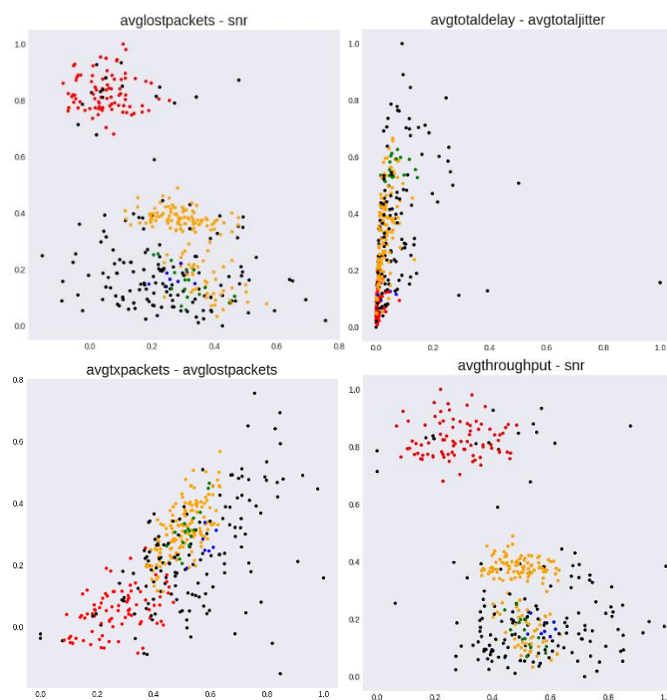


Figura 51: Gráficas bidimensionales del cruce de parámetros de los APs con el resultado de agrupamiento de DBSCAN.

9.1.3. Resultados de Agrupación de GMM

Los **aspectos principales del procedimiento** llevado a cabo para la ejecución de GMM se recogen en la Tabla 78, que sigue las especificaciones definidas en el apartado Implementación de GMM.

Característica		Valor
Número de Componentes		4
Número de inicializaciones		10
Número máximo de iteraciones		100
Número de muestras	Total	400
	Escenario 1	100
	Escenario 2	100
	Escenario 3	100
	Escenario 4	100

Tabla 78: Resumen del experimento de GMM sobre las métricas de APs.

El resultado del clustering mediante el algoritmo GMM para la diferenciación de entornos se muestra en la Figura 52, donde se puede contrastar el resultado con las etiquetas establecidas en base a la definición de las simulaciones.

En contraposición a DBSCAN, el algoritmo no hace descarte de muestras, sino que asigna todo punto a un grupo representativo de los escenarios. Visualmente, se puede apreciar una **identificación favorable de los entornos** definidos, pero existen múltiples desplazamientos entre grupos frente a lo definido de forma teórica.

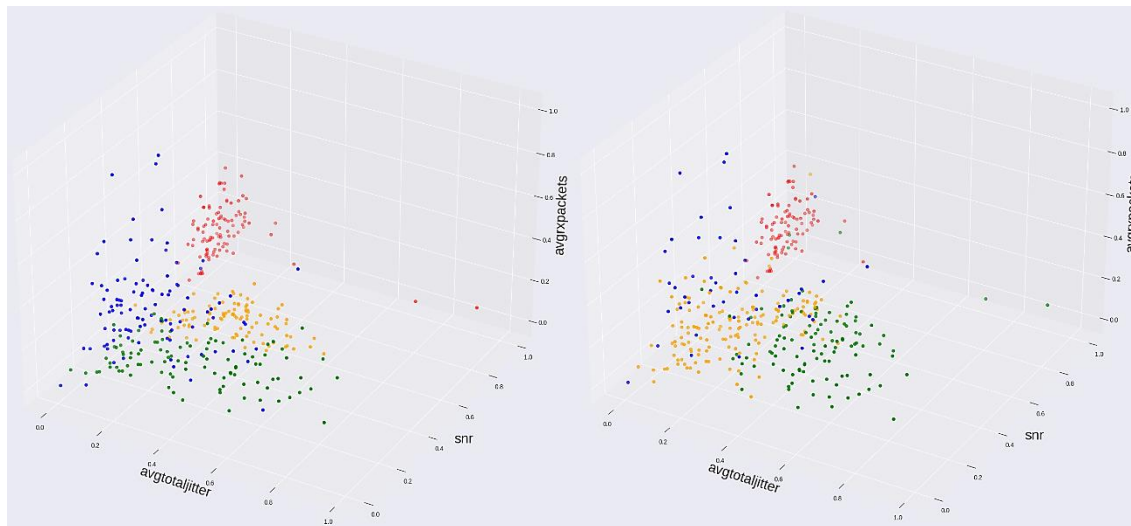


Figura 52: Gráfica tridimensional con las muestras de las métricas de los APs etiquetadas (izq.) y el resultado del agrupamiento de GMM (der.).

En concreto, según lo establecido como parámetros de entrada se realiza la **formación de un total de cuatro clústeres**. Entre estos, a simple vista se observa una distribución similar a la original en la que muestras de los **entornos de alta densidad** se hallan repartidas en **mismo número aproximado** que en la definición teórica pero **distinta posición espacial**.

La predicción llevada a cabo por GMM se basa en la formación de clústeres en base a distribuciones gaussianas, lo que provoca que los grupos dispongan de **formas flexibles**, como también se puede observar en la Figura 52.

Para un análisis en mayor detalle, la Tabla 79 presenta, al igual que en los casos de estudio de algoritmos anteriores, el número total de muestras asignadas por grupo y, posteriormente, la distribución de etiquetas correspondientes a las muestras de cada entorno.

Entre los resultados presentados en la tabla, se aprecia una **clara definición del segundo escenario**. Por el contrario, gran parte de las **muestras del primer escenario se hallan asignadas a otros grupos**, lo que hace que menos de la mitad de las muestras del entorno en cuestión sean identificadas como tal. Por otro lado, a pesar de que la mayoría de las muestras de los entornos de alta densidad se reconocen adecuadamente dentro de sus grupos correspondientes, se aprecia un considerable número de **muestras asociadas de cada entorno de alta densidad en su contrario**.

Clasificación	Color	Valor	Cantidad asignada
Total	Azul	0	44
	Rojo	1	93
	Verde	2	153
	Naranja	3	110
Escenario 1	Azul	0	39
	Rojo	1	0
	Verde	2	53
	Naranja	3	8
Escenario 2	Azul	0	1
	Rojo	1	93
	Verde	2	1
	Naranja	3	5
Escenario 3	Azul	0	4
	Rojo	1	0
	Verde	2	53
	Naranja	3	43
Escenario 4	Azul	0	0
	Rojo	1	0
	Verde	2	46
	Naranja	3	54

Tabla 79: Resultados de GMM sobre las métricas de APs en formato tabla.

De forma adicional, la Figura 53 provee **perspectivas adicionales a la representación** de muestras a través de **gráficas bidimensionales**. En este caso, la definición de las distribuciones de puntos para los escenarios se asemeja al gráfico tridimensional de la Figura 52. Sin embargo, en contraste con distribuciones basadas en distancia o densidad, se aprecian puntos que espacialmente están distantes a la mayoría de las muestras de su propio grupo.

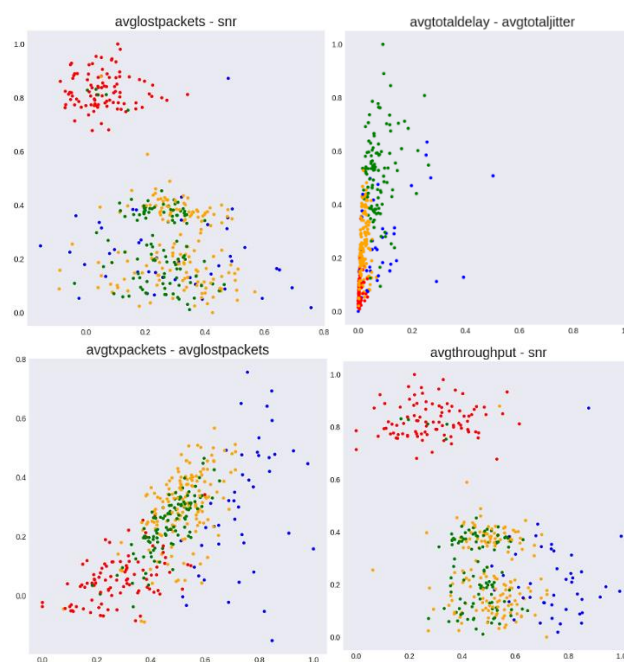


Figura 53: Gráficas bidimensionales del cruce de parámetros de los APs con el resultado de agrupamiento de GMM.

9.1.4. Resultados de Agrupación de BIRCH

En último lugar, tal y como se adelantó en el apartado Implementación de BIRCH de la Metodología, las **especificaciones del procedimiento** adoptado para la ejecución de BIRCH se hallan resumidos en la Tabla 74.

A diferencia de las aproximaciones anteriores, no se dispone de métodos de estimación sobre los parámetros característicos del algoritmo. Sin embargo, un número mínimo de iteraciones ha resultado suficiente para que BIRCH realice una clasificación aproximada de los entornos de gran afinidad al resto de resultados de los algoritmos de clustering.

Característica		Valor
Número de Componentes		4
Factor de ramificación		2
Umbral		0.085
Número de muestras	Total	400
	Escenario 1	100
	Escenario 2	100
	Escenario 3	100
Escenario 4		100

Tabla 80: Resumen del experimento de GMM sobre las métricas de APs.

El ajuste del agrupamiento mediante BIRCH para la diferenciación de entornos se muestra en la Figura 54, donde se puede comparar el resultado con los escenarios previamente etiquetados. Cabe mencionar que en los datos de entrada mediante la implementación realizada **se permite la introducción del número de componentes**, que, a pesar de no ser obligatorio para el correcto funcionamiento del algoritmo, permite realizar ajustes de mayor flexibilidad.

Del mismo modo que en K-Means o GMM, no se realiza ningún descarte de muestras, por lo que se relaciona todo punto a uno de los clústeres. A simple vista, se puede apreciar una **identificación próxima a los entornos definidos**, aunque existen **ligeras variaciones** entre los escenarios de alta densidad.

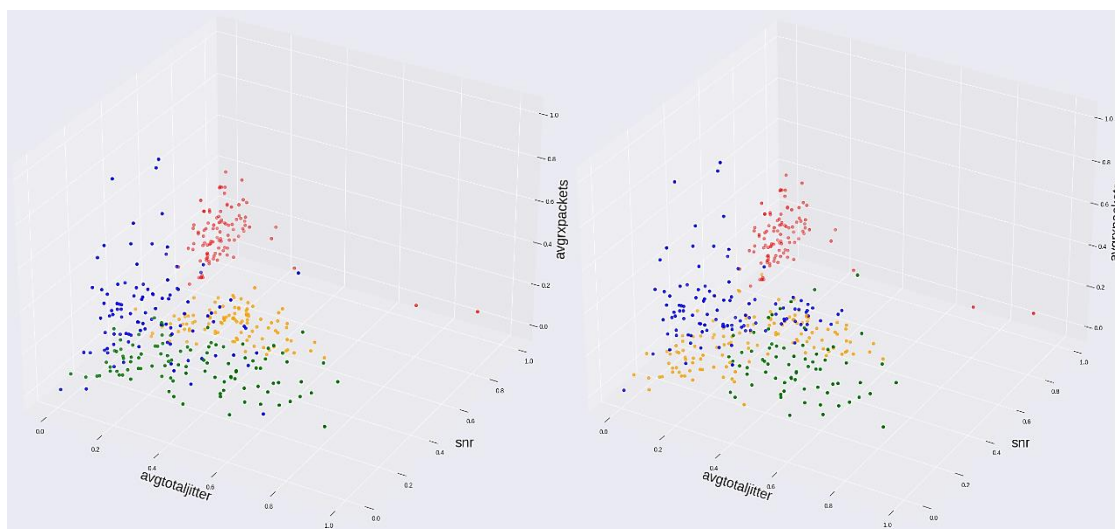


Figura 54: Gráfica tridimensional con las muestras de las métricas de los APs etiquetadas (izq.) y el resultado del agrupamiento de BIRCH (der.).

En concreto, de acuerdo con lo establecido en el número de componentes se generan un total de **cuatro clústeres**. Entre estos, a simple vista se observa una **distribución bastante cercana a la original**, aunque se pueden apreciar **ligeros desplazamientos** de muestras entre grupos del **primer entorno y los de alta densidad**.

De forma análoga a los anteriores casos de estudio, la Tabla 81 recoge el número total de muestras asignadas por grupo y, después, la distribución de etiquetas correspondientes a las muestras de cada entorno.

Clasificación	Color	Valor	Cantidad asignada
Total	Azul	0	107
	Rojo	1	100
	Verde	2	123
	Naranja	3	70
Escenario 1	Azul	0	68
	Rojo	1	0
	Verde	2	17
	Naranja	3	15
Escenario 2	Azul	0	0
	Rojo	1	100
	Verde	2	0
	Naranja	3	0
Escenario 3	Azul	0	34
	Rojo	1	0
	Verde	2	65
	Naranja	3	1
Escenario 4	Azul	0	5
	Rojo	1	0
	Verde	2	41
	Naranja	3	54

Tabla 81: Resultados de BIRCH sobre las métricas de APs en formato tabla.

Entre los resultados presentados en la tabla, la asignación relativa al **segundo escenario** es idéntica a la original, lo que se traduce en un **agrupamiento óptimo**. Asimismo, en el caso del **primer y tercer escenario** el **reconocimiento de muestras** a los entornos correspondientes se realiza **de forma satisfactoria**. En otras palabras, para ambos casos predominan las muestras con identificadores al grupo correspondiente definido de forma teórica en un origen.

Por último, en el caso del cuarto escenario, se interpretan como parte del mismo muestras propias de dicho entorno y también del otro entorno de alta densidad. Por lo tanto, vuelve a tener lugar un **ajuste limitado** en la diferenciación de entornos dentro de los **escenarios de alta densidad**.

De forma complementaria, la Figura 55 representa los resultados de la agrupación de forma bidimensional, lo que permite una **observación en mayor detalle de interacciones entre variables**.

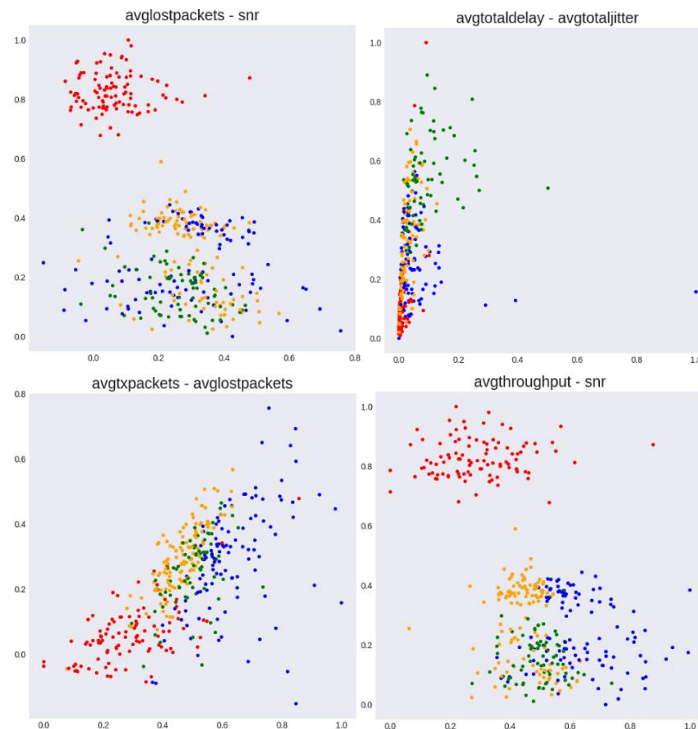


Figura 55: Gráficas bidimensionales del cruce de parámetros de los APs con el resultado de agrupamiento de BIRCH.

9.2. Resultados de Agrupación para la Clasificación de Perfiles

De forma análoga a la diferenciación de entornos, la clasificación de perfiles trata de **extraer el contexto de la red a través de las métricas de los usuarios**. Para ello, se utilizan las estadísticas obtenidas desde las estaciones en las simulaciones de red.

Sin embargo, a diferencia del procedimiento anterior, no se dispone de una caracterización detallada de los perfiles a obtener, ya que **la definición de perfiles de usuarios dispone de infinidad de posibilidades** en términos de caracterización del **contexto del usuario** –edad, contexto social y cultural, aspectos temporales como la hora del día o día de la semana, etc.– y parámetros a seleccionar, lo que queda fuera del alcance del presente estudio. Por lo tanto, la aproximación utilizada para la validación de los algoritmos sobre este procedimiento se define a continuación.

Las **métricas disponibles** son extractos de las mismas simulaciones realizadas para la diferenciación de entornos, pero en este caso, con **parámetros a nivel de STA** y no de AP. Los escenarios de simulación definidos establecen **cinco tipos de tráfico** repartidos de forma distintiva entre las STAs que conforman cada red.

No obstante, la asignación se hace de forma aleatoria sobre el total de dispositivos, por lo que puede haber grandes cantidades de terminales que no generen flujos de tráfico específicos, y dispositivos que generen tráfico de múltiples tipos. De forma complementaria, se puede emplear el fichero de flujos de tráfico generado para realizar comprobaciones.

Como se ha definido en el apartado de Implementación de los Algoritmos de Clustering, los algoritmos en estudio disponen de métodos para la estimación de sus parámetros

principales. Por lo tanto, el procedimiento a seguir se basa en la utilización de los resultados de dichas **estimaciones sobre las muestras sin etiquetar**, y realizar un **análisis comparativo de los grupos generados** en términos de número y valores de las métricas.

Dichos grupos seguirán una identificación similar a la empleada en la diferenciación de entornos, de forma que el mapeo resultante de las muestras seguirá las especificaciones de la Tabla 82.

ID Escenario	Clasificación	Color	Valor de etiqueta
1	Clúster 1	Azul	0
2	Clúster 2	Rojo	1
3	Clúster 3	Verde	2
4	Clúster 4	Naranja	3

Tabla 82: Mapeo de identificadores para el seguimiento del análisis de resultados de la clasificación de perfiles.

En definitiva, a continuación se presentan los resultados de cada algoritmo en estudio para el proceso de realizar dicho etiquetado sin conocimiento previo del escenario.

9.2.1. Resultados de Agrupación de K-Means

Conforme se adelantó en el apartado Implementación de K-Means de la Metodología, los **detalles del procedimiento** adoptado para la ejecución de K-Means se encuentran resumidos en la Tabla 83.

En este caso, las diferencias principales residen en el número de muestras y los resultados de las métricas, en los que se observa un reparto de valores muy distinto al caso de los APs.

Característica		Valor
Número de Componentes		4
Número de inicializaciones		10
Número máximo de iteraciones		100
Número de muestras	Total	9762
	Escenario 1	809
	Escenario 2	1753
	Escenario 3	3844
	Escenario 4	3356

Tabla 83: Resumen del experimento de K-Means sobre las métricas de STAs.

El resultado de la agrupación mediante el algoritmo K-Means sobre las métricas de STAs se halla representado en la Figura 56, en la cual se **compara con las muestras sin etiquetar**.

A simple vista, se hace una separación de **cuatro grupos de características** que espacialmente se encuentran distanciados de forma notoria, y que constan de densidades, **tamaños y formas distintas**.

Las apreciaciones anteriores pueden expandirse gracias a las perspectivas adicionales provistas por la representación bidimensional de la Figura 57.

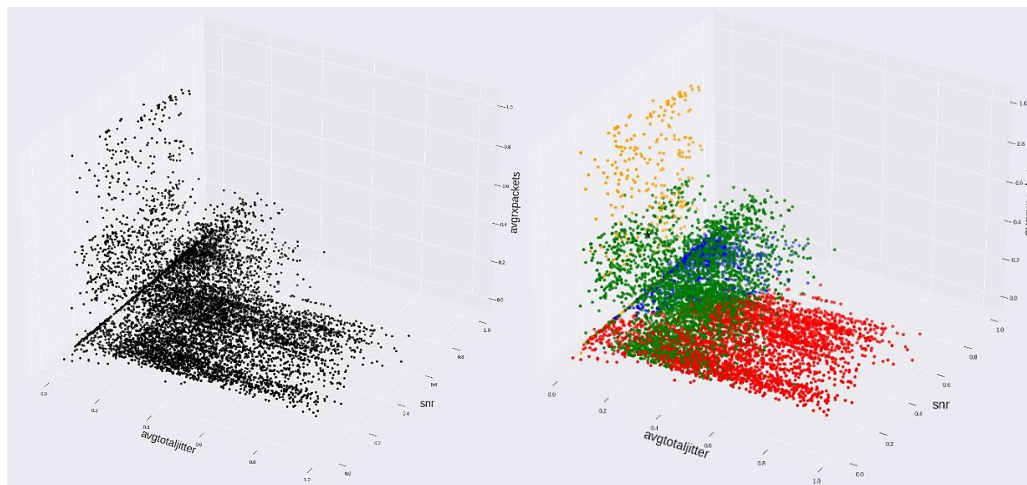


Figura 56: Gráfica tridimensional con las muestras de las métricas de las STAs sin etiquetar (izq.) y el resultado del agrupamiento de K-Means (der.).

En términos de **SNR**, todos los escenarios muestran una variación considerable entre las muestras, siendo similar en todos los clústeres a excepción del primero. En cuanto a la dispersión y valores de los parámetros de **throughput, paquetes transmitidos y recibidos**, se hace una diferenciación clara en la Figura 57 entre los primeros dos clústeres y el resto, siendo los primeros mínimos y el clúster 4 máximo.

Por último, en la gráfica relativa al **retardo** y el **jitter** también se aprecian diferencias notables entre el reparto de muestras de los dos primeros clústeres respecto a los segundos.

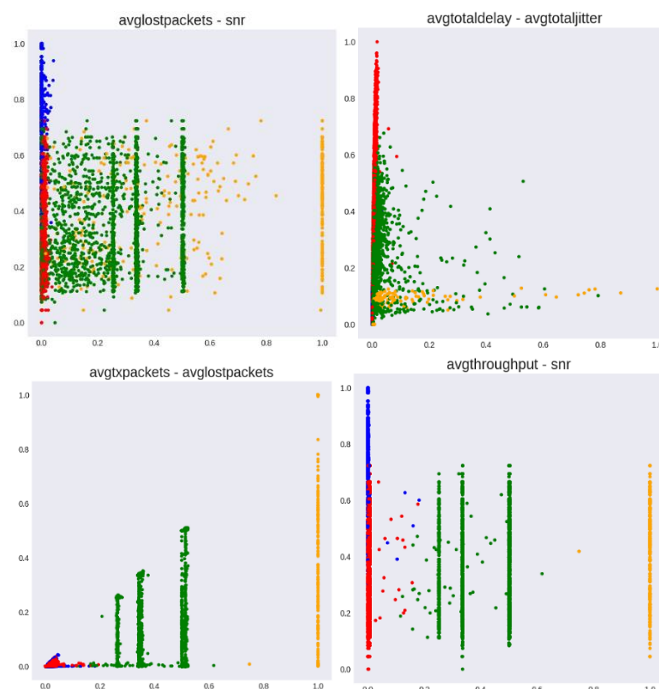


Figura 57: Gráficas bidimensionales del cruce de parámetros de las STAs con el resultado de agrupamiento de K-Means.

9.2.2. Resultados de Agrupación de DBSCAN

De forma análoga a K-Means, en el apartado Implementación de DBSCAN de la Metodología se establecen los **detalles del procedimiento** (Tabla 84) seguido para la ejecución de DBSCAN.

Característica		Valor
Parámetro EPS		0,121
Parámetro Mínimo de Muestras		7
Tipo de Distancia		Euclídea
Número de muestras	Total	9762
	Escenario 1	809
	Escenario 2	1753
	Escenario 3	3844
	Escenario 4	3356

Tabla 84: Resumen del experimento de DBSCAN sobre las métricas de STAs.

La representación de la agrupación resultante de la aplicación del algoritmo DBSCAN basada en las métricas de las STAs se encuentra en la Figura 58, donde se compara con las muestras sin etiquetado.

Visualmente, se distingue una **división de cuatro conjuntos de características** que difieren en gran medida en tamaño y forma. En general, gran parte de los grupos muestran una **notable correspondencia con** el resultado obtenido para **K-Means**.

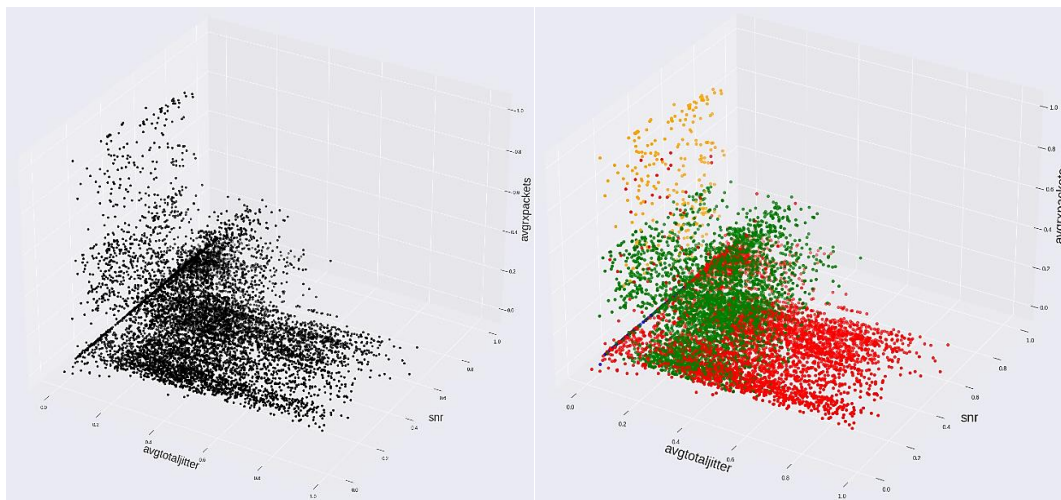


Figura 58: Gráfica tridimensional con las muestras de las métricas de las STAs sin etiquetar (izq.) y el resultado del agrupamiento de DBSCAN (der.).

Específicamente, el clúster 1 tan solo se encuentra en el límite inferior de la intersección entre el eje de las ordenadas y las cotas, y es el grupo con mayor diferenciación respecto al algoritmo anterior. Esta observación puede enriquecerse con las visiones adicionales proporcionadas por la representación bidimensional en la Figura 59.

En términos de **SNR, retardo y jitter**, presenta unas características similares a lo establecido para el mismo clúster en K-Means. Sin embargo, en cuanto a la dispersión y valores de los parámetros de **throughput, paquetes transmitidos y recibidos**, se distingue claramente de dichos resultados. En concreto, se trata de valores cuyas

transmisiones no han sido efectivas, ya que tienen valores máximos de throughput y paquetes transmitidos, y valores mínimos en cuanto a paquetes recibidos.

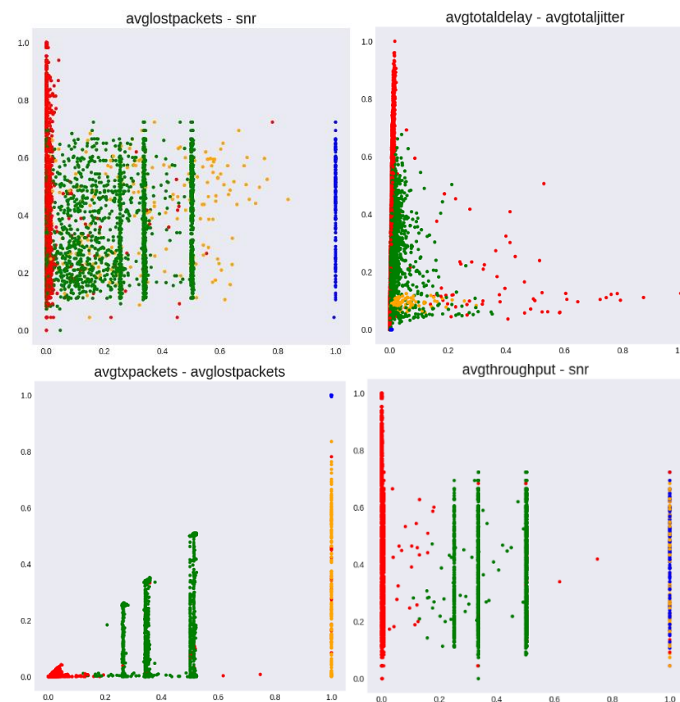


Figura 59: Gráficas bidimensionales del cruce de parámetros de las STAs con el resultado de agrupamiento de DBSCAN.

9.2.3. Resultados de Agrupación de GMM

Los **aspectos principales del procedimiento** llevado a cabo para la ejecución de GMM se recogen en la Tabla 85, que sigue las especificaciones definidas en el apartado Implementación de GMM.

Característica		Valor
Número de Componentes		4
Número de inicializaciones		10
Número máximo de iteraciones		100
Número de muestras	Total	9762
	Escenario 1	809
	Escenario 2	1753
	Escenario 3	3844
	Escenario 4	3356

Tabla 85: Resumen del experimento de GMM sobre las métricas de STAs.

El resultado de la agrupación mediante el algoritmo GMM sobre las métricas de STAs se encuentra representado en la Figura 60, en la cual se **compara con las muestras sin etiquetar**.

A simple vista, se hace una separación similar a los algoritmos anteriores, en la que se distinguen **cuatro grupos de características** que difieren en densidad, **tamaño y forma**. Sin embargo, la tendencia es más cercana al clustering según K-Means que en comparación con DBSCAN.

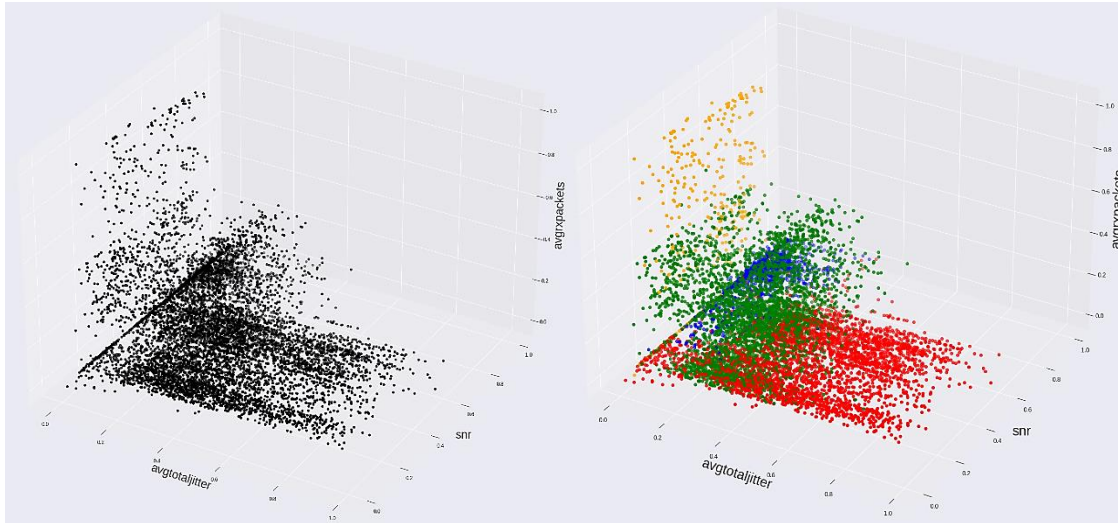


Figura 60: Gráfica tridimensional con las muestras de las métricas de las STAs sin etiquetar (izq.) y el resultado del agrupamiento de GMM (der.).

Las deducciones anteriores se amplían mediante las perspectivas añadidas de la **representación bidimensional** de la Figura 61.

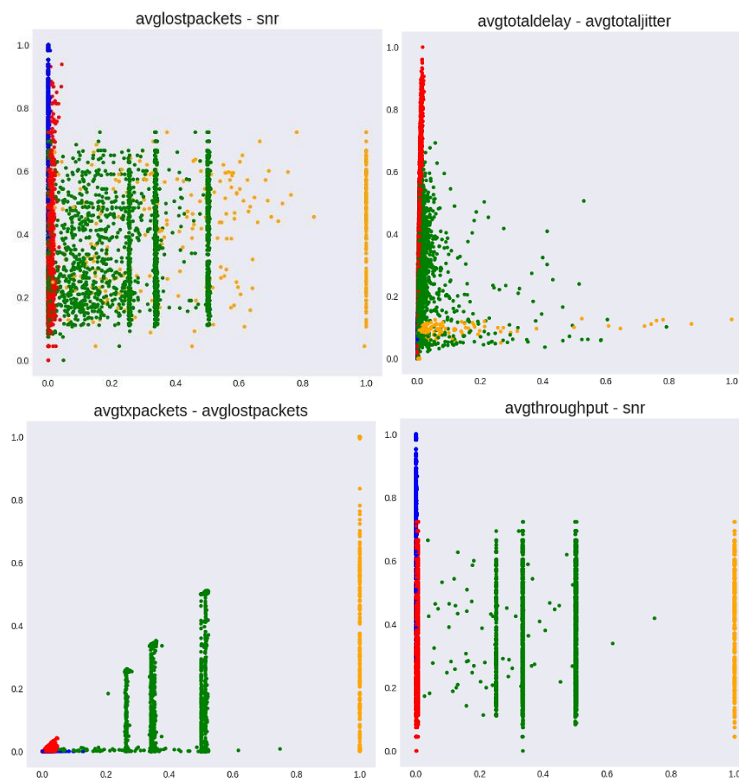


Figura 61: Gráficas bidimensionales del cruce de parámetros de las STAs con el resultado de agrupamiento de GMM.

Con relación a los valores de **SNR**, todos los clústeres presentan una alta dispersión de valores entre sus muestras, siendo similar en todos los grupos a excepción del primero. En términos de **throughput, paquetes transmitidos y recibidos**, sucede igual que en

la agrupación mediante K-Means, de forma que se hace una diferenciación clara en la entre los primeros dos clústeres y los demás, adoptando los primeros valores mínimos.

Para terminar, en la gráfica del cruce de variables de **retardo** y el **jitter** también se aprecian diferencias entre el reparto de muestras de los dos primeros clústeres y el resto. De forma comparativa, la variación del retardo es mayor en el tercer y cuarto clúster, mientras que la dispersión de los valores de jitter es mayor en los dos primeros.

9.2.4. Resultados de Agrupación de BIRCH

En último lugar, las **especificaciones del procedimiento** adoptado para la ejecución de BIRCH se hallan resumidos en la Tabla 86. En esta se siguen las indicaciones definidas en el apartado Implementación de BIRCH de la Metodología.

Característica		Valor
Número de Componentes		4
Factor de ramificación		4
Umbral		0.5
Número de muestras	Total	9762
	Escenario 1	809
	Escenario 2	1753
	Escenario 3	3844
Escenario 4		3356

Tabla 86: Resumen del experimento de BIRCH sobre las métricas de STAs.

El agrupamiento a través de BIRCH para la clasificación de perfiles se muestra en la Figura 62, donde se puede observar el resultado frente a las muestras sin etiquetar.

Cabe mencionar que, al igual que en la implementación de diferenciación de entornos, en los datos de entrada **se permite la introducción del número de componentes**, que permite realizar ajustes con mayor flexibilidad.

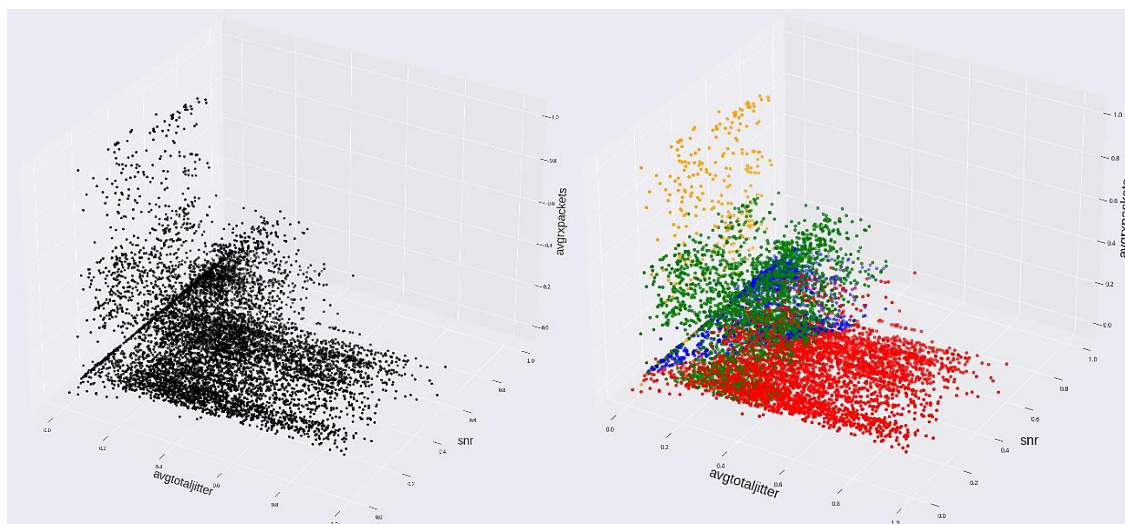


Figura 62: Gráfica tridimensional con las muestras de las métricas de las STAs sin etiquetar (izq.) y el resultado del agrupamiento de BIRCH (der.).

En este caso, se ha establecido un número de **cuatro componentes** como parámetro, y tan solo ha resultado necesaria una **iteración mínima** sobre los valores del factor de

ramificación y umbral para obtener una formación de grupos consistente con el número de componentes establecido. En concreto, los resultados adquieren cierta **semejanza con los clústeres generados a través de los otros algoritmos**.

La dispersión de las muestras y los valores adquiridos por ellas se pueden ver representados bidimensionalmente en la Figura 63. Los **resultados son muy similares a los de K-Means y GMM**, pero se dispone de mayor variabilidad en términos de throughput, paquetes transmitidos y recibidos, y retardo para el primer clúster.

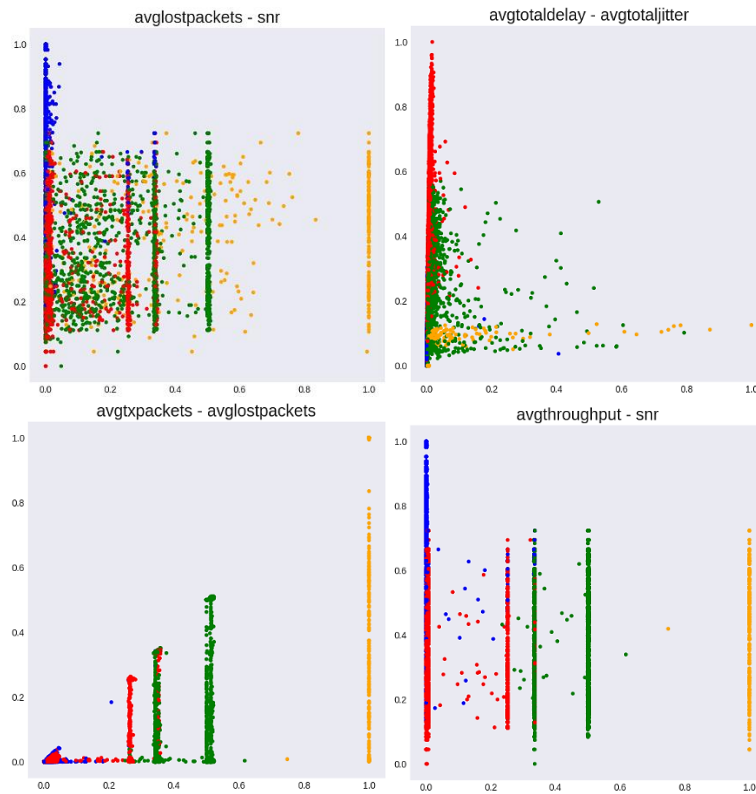


Figura 63: Gráficas bidimensionales del cruce de parámetros de las STAs con el resultado de agrupamiento de BIRCH.

9.3. Análisis de Resultados sobre la Diferenciación de Entornos y la Clasificación de Perfiles

Los resultados de simulación presentan **comportamientos variados** sobre el conjunto de datos en términos de rendimiento sobre los procedimientos de mejora de QoS. En general, todos los algoritmos son **capaces de extraer información del contexto de la red y el usuario** en base a las métricas establecidas.

Sin embargo, aunque el desempeño de los algoritmos resulte aceptable en gran medida, existen **múltiples diferencias que subrayan la importancia de seleccionar el algoritmo adecuado**. Entre ellas, destaca la dependencia sobre las características específicas del conjunto de datos.

En el presente estudio se ha trabajado con dos conjuntos de datos de distinto tamaño y forma. En primer lugar, en la **diferenciación de entornos** (Figura 64) se ha hecho uso

de una estructura reducida y uniforme, de forma que cada escenario dispusiera de una cantidad igual y acotada de muestras.

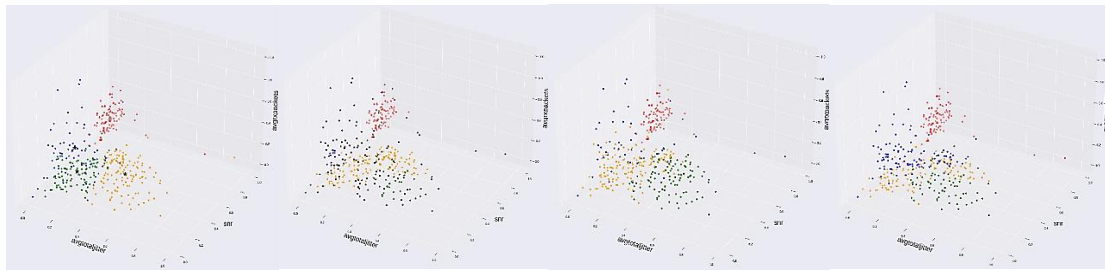


Figura 64: Gráficas tridimensionales resultantes de K-Means, DBSCAN, GMM y BIRCH para la diferenciación de entornos.

Por otro lado, sobre las mismas ejecuciones del simulador, el número de estaciones es múltiple y resulta variable, lo que hace que el set de datos utilizado como base para la **clasificación de perfiles** (Figura 65) sea más numeroso y con un número irregular de muestras por escenario.



Figura 65: Gráficas tridimensionales resultantes de K-Means, DBSCAN, GMM y BIRCH para la clasificación de perfiles.

Además, en este segundo procedimiento no se dispone de una serie de etiquetas conocidas sobre las muestras, sino que se obtienen los clústeres a través de la aplicación directa de los algoritmos.

Para facilitar la comparativa de resultados, se han extraído **características generales de los grupos generados** en base a las métricas de STAs de acuerdo con la Tabla 87. En ella se indica mediante el uso de símbolos positivos (+), negativos (-) y neutros (=) la diversidad y valor medio de los parámetros en cada clúster. Además, a través del fichero con la información de flujos, es posible mapear los servicios utilizados en las muestras de tráfico de las STAs. En la misma tabla se comenta la disposición general de servicios por clúster.

Como se ha expuesto anteriormente, **K-Means** es un algoritmo cuyo rendimiento tiene una fuerte **dependencia sobre la posición inicial** de los centroides para la formación de grupos, y tendencia a la generación de **grupos de tamaño equilibrado y forma regular**. En este caso concreto, la dependencia respecto a la posición inicial de los centroides no supone un problema, ya que para subsanar dicho inconveniente la implementación realizada permite ejecución iterativa de inicializaciones desde un principio, de modo que se haga uso de la mejor aproximación inicial.

Resultado	Parámetro	Variación	Valor medio	Tipos de tráfico
Clúster 1	avgthroughput	--	---	Asignación variable de servicios, en general ausencia de VoD (todos los tipos de tráfico implementados)
	avgtxpaquets	--	---	
	avgrxpaquets	--	---	
	avgtotaldelay	--	---	
	avgtotaljitter	+	=	
	snr	+++	=	
Clúster 2	avgthroughput	-	--	Asignación variable de servicios (todos los tipos de tráfico implementados)
	avgtxpaquets	--	--	
	avgrxpaquets	--	--	
	avgtotaldelay	--	---	
	avgtotaljitter	+++	=	
	snr	++	=	
Clúster 3	avgthroughput	++	++	Asignación variable con ligera diversidad de servicios, principalmente VoD
	avgtxpaquets	++	++	
	avgrxpaquets	++	++	
	avgtotaldelay	++	=	
	avgtotaljitter	++	=	
	snr	++	=	
Clúster 4	avgthroughput	---	+++	Alta concentración de servicios por dispositivo (todos los tipos de tráfico implementados)
	avgtxpaquets	---	+++	
	avgrxpaquets	---	+++	
	avgtotaldelay	+++	=	
	avgtotaljitter	---	--	
	snr	++	=	

Tabla 87: Caracterización genérica de los clústeres obtenidos a través de los algoritmos para la tarea de clasificación de perfiles.

Además de tratarse de un **algoritmo simple y eficiente**, ha demostrado ser **efectivo para el set de datos utilizado en ambos casos**.

En la **diferenciación de entornos**, los grupos obtenidos comparten una cantidad considerable de muestras con la definición en base a etiquetas de los escenarios. Sin embargo, en el caso de estudio actual muestra alguna **dificultad para distinguir los entornos de alta densidad entre sí**, lo que indica que, en base a la complejidad de los patrones establecidos para definir los escenarios, puede presentar algún desafío para una correcta clasificación en entornos cuyas métricas sigan patrones irregulares y bajo escenarios con mayor número de grupos.

Para el caso de la **clasificación de perfiles**, los **clústeres generados** se forman a través de muestras cuyas **métricas varían en términos de dispersión y valor medio**. En comparación con la definición de grupos de la Tabla 87 establecida a través de la definición de grupos del total de los algoritmos, ofrece **resultados satisfactorios**. Asimismo, el contraste de las muestras de tráfico por dispositivo con el **fichero de flujos** de tráfico corrobora la división de servicios establecida en la tabla.

En segundo lugar, **DBSCAN** tiene bondades parcialmente contrapuestas a las de K-Means. En este caso, el algoritmo permite identificar **clústeres de formas arbitrarias**, y permite **detectar ruido** entre las muestras de datos. Sin embargo, al contrario que en K-Means, el **establecimiento de parámetros** resulta una **tarea crítica** para el

desempeño del agrupamiento. Esto puede provocar la necesidad de iterar sobre diferentes valores hasta la obtención de un **calibrado adecuado** al conjunto de datos.

En relación con la **diferenciación de entornos** se realiza un descarte de muestras considerable. En concreto, en el primer escenario la dispersión de muestras sobre los ejes hace que se realice un descarte de la mayoría de las muestras de dicho entorno. En consecuencia, se puede intuir la **dificultad de generación de clústeres sobre entornos muy variables**. Por el contrario, en el caso del segundo y cuarto escenario se obtienen resultados con algún descarte de muestras, pero sin **ningún error en la asignación sobre los entornos**. De forma similar a K-Means, el algoritmo puede mostrar dificultades en la distinción de entornos con algunas características en común, como es el caso de alta densidad y las muestras del tercer entorno asignadas erróneamente.

Por otro lado, DBSCAN muestra un ajuste distinto al resto de algoritmos en la formación de clústeres relativa a la **clasificación de perfiles**. En específico, el **primer clúster** se forma en base a las muestras cuya diferencia entre valores de transmisión y recepción es máxima, a diferencia del clúster definido en la Tabla 87. A pesar de esto, es capaz de distinguir el resto de los clústeres con un **rendimiento similar a K-Means, GMM y BIRCH**, por lo que ofrece **resultados parcialmente satisfactorios**.

En tercer lugar, **GMM** ofrece **flexibilidad** en la generación de clústeres en términos de **forma**, ya que asume que los datos provienen de la mezcla de distribuciones gaussianas. Esta característica permite modelar **entornos con características complejas** y variadas. Sin embargo, su naturaleza probabilística y la necesidad de definir múltiples parámetros lo vuelve un **algoritmo exigente en términos de conocimiento del conjunto de datos**. Asimismo, puede mostrar **sensibilidad a sets grandes y ruidosos**.

Su desempeño en la **diferenciación de entornos** viene marcado por una **correcta identificación del segundo escenario**, y una distribución de muestras equilibrada sobre los entornos de alta densidad. Sin embargo, entre estos últimos prácticamente la mitad de las muestras son asignadas de forma errónea. Por lo tanto, el algoritmo **puede distinguir que se trata de un entorno de alta densidad**, pero su aplicación **no diferencia entre las dos definiciones separadas** sobre escenarios de interior y exterior. Asimismo, se observan **dificultades para la definición del clúster** correspondiente al **primer escenario**, ya que se recogen en él muestras de múltiples entornos.

Con relación a la **clasificación de perfiles**, es capaz de extraer los **mismos grupos de características** para la formación de clústeres que **K-Means**, con quien comparte en gran medida su resultado final de agrupamiento. **Las muestras** que forman cada clúster **varían en términos de dispersión y valor medio** en las medidas del dispositivo. En comparación con la definición de grupos de la Tabla 87, resulta un **ajuste próximo al promedio**. Además, al igual que en K-Means, se puede comprobar la adecuación del set de datos a los servicios definidos en la tabla. Para ello, se utiliza la correlación entre las muestras pertenecientes a los diferentes clústeres y los servicios

presentes en las estaciones –extraídos del **fichero de flujos** de tráfico– a las que se corresponden dichas muestras. Además, ofrece un resultado **coherente con la definición de las simulaciones**, ya que en la propia definición de la asignación de servicios a STAs y el reparto de tráfico subyacen múltiples **distribuciones gaussianas**.

Por último, **BIRCH** permite trabajar con **grandes cantidades de datos** mediante la estructuración del conjunto de muestras y es adecuado para **estructuras jerárquicas**. Por lo tanto, el desempeño del algoritmo depende de la estructura interna del set de datos, lo que puede provocar **dificultades sobre clasificaciones de gran precisión**.

Respecto a su aplicabilidad sobre la **diferenciación de entornos**, la **iteración inicial** necesaria para el establecimiento de los parámetros de umbral y factor de ramificación ha resultado **breve**, ya que con un número mínimo de ramas es capaz de intuir la estructura del conjunto de datos. En el caso del **primer entorno**, ha mostrado el **mejor ajuste** entre los cuatro algoritmos en estudio, además de un **óptimo reconocimiento del segundo escenario**. En los casos de **alta densidad**, el **ajuste** sobre las muestras relativas al tercer entorno es muy **cercano al obtenido con K-Means**.

Con relación a la **clasificación de perfiles**, los **clústeres generados** resultan más próximos a los resultados de K-Means y GMM. No obstante, se dispone de mayor dispersión de valores en términos de throughput, paquetes transmitidos y recibidos, y retardo para el primer clúster. Además, los **servicios disponibles** en las muestras de dicho clúster disponen de **mayor variabilidad**, ya que se introduce un mayor número de muestras con servicios de vídeo.

Para finalizar con el análisis de alternativas, a continuación se recogen las **conclusiones globales** en base a los resultados obtenidos.

En el caso de la **diferenciación de entornos**, la Tabla 88 muestra de forma resumida la valoración realizada sobre el ajuste de los algoritmos al estudio.

La columna de valoración de aciertos puntúa el número de muestras detectadas por algoritmo para cada escenario de forma positiva, y en el caso de la valoración de errores, se puntúa siguiendo una lógica negativa. En consecuencia, la ausencia de puntuación indica un mal ajuste en el caso de los aciertos, y un buen ajuste en el caso de los errores.

Como se puede observar en la tabla, **K-Means** y **BIRCH** obtienen mejores puntuaciones de forma general, lo que se traduce en un mayor ajuste en la formación de grupos respecto a la definición teórica de los escenarios. En el caso de DBSCAN y GMM, se obtienen buenos ajustes para casos concretos de los escenarios, pero muestran dificultades en la detección de parte de los escenarios, por lo que su aptitud frente al experimento establecido resulta parcial.

Por otro lado, en función de los resultados obtenidos para el estudio de la **clasificación de perfiles**, **todos los algoritmos** han demostrado disponer de **capacidad de elaborar grupos de características similares** partiendo de un conjunto de datos no etiquetado, y con cierto grado de correlación respecto a los servicios disponibles en las

estaciones. En concreto, **K-Means** y **GMM** obtienen resultados muy similares y relacionables a perfiles de uso servicios genéricos.

Clasificación	Color	Valoración de Aciertos	Valoración de Errores
Escenario 1	K-Means	✓✓	XX
	DBSCAN	-	X
	GMM	✓	XX
	BIRCH	✓✓	X
Escenario 2	K-Means	✓✓✓	-
	DBSCAN	✓✓✓	-
	GMM	✓✓✓	-
	BIRCH	✓✓✓	-
Escenario 3	K-Means	✓✓	X
	DBSCAN	✓	XX
	GMM	✓✓	XX
	BIRCH	✓✓	X
Escenario 4	K-Means	✓✓	XX
	DBSCAN	✓✓✓	-
	GMM	✓✓	XX
	BIRCH	✓✓	XX

Tabla 88: Valoración resumida de los algoritmos sobre la diferenciación de entornos.

En definitiva, tanto la **diferenciación de entornos** como la **clasificación de perfiles** resultan tareas de alta complejidad que pueden beneficiarse de **enfoques híbridos** sobre los algoritmos de clustering. De este modo, además del ajuste a ambos procedimientos revisado en el presente análisis, se debe considerar la posibilidad de implementar varios algoritmos de forma combinada para favorecer mayor robustez y precisión de resultados.

10. Descripción del Equipo de Trabajo y Planificación

A continuación, se describe el equipo para la realización del Trabajo Fin de Máster y la planificación utilizada para el desarrollo del estudio.

10.1. Equipo de Trabajo

Los miembros del equipo de trabajo son el alumno o proyectista y la directora. El presente apartado, por tanto, introduce a ambos participantes.

En primer lugar, la **directora del Trabajo Fin de Máster** es Eva Ibarrola Armendáriz, profesora del Máster en Ingeniería de Telecomunicación que forma parte del departamento de Ingeniería de Comunicaciones (del Área de Ingeniería Telemática). Además, es investigadora del grupo de investigación NQAS (Networking, Quality and Security).

Por otro lado, el **proyectista** es Mikel Acha Alonso, y su función principal es la del diseño de la solución del trabajo, la realización de pruebas y la extracción de conclusiones en el estudio.

10.2. Planificación de Trabajo

El alcance del Trabajo Fin de Máster consta de tres grandes grupos de tareas: una primera fase de formación sobre el ámbito del estudio y las herramientas a emplear, el conjunto de tareas prácticas de desarrollo de la metodología y una última fase de análisis y extracción de conclusiones.

Debido a la extensión del proyecto, se ha realizado una planificación minuciosa que abarque todas las tareas realizadas a lo largo del estudio mediante su agrupación en paquetes de trabajo que organizan las labores realizadas, y estableciendo una serie de hitos de control que posibiliten un seguimiento y registro de la finalización de las tareas. La parte documental se ha hecho de forma transversal a todo el desarrollo del proyecto.

Concepto	ID
Tarea	T.
Paquete de trabajo	P.T.
Hito	H.

Tabla 89: Relación entre los identificadores de la planificación del proyecto y el concepto que representan.

Por consiguiente, la Tabla 90 recoge la estructura completa de la planificación – identificador, nombre, descripción y duración en días– en base a los conceptos anteriormente introducidos y en la Figura 66 se muestra el diagrama de Gantt correspondiente al desarrollo del proyecto.

Asimismo, en la Tabla 89 se recogen los identificadores utilizados tanto en la tabla con la estructura completa del proyecto como el diagrama de Gantt.

ID	Nombre	Descripción	Duración
P.T.1	Formación básica y de profundización	Formación básica y de profundización en WLAN, QoS, ML y simulación de red. Recopilación de programas iniciales.	88
T.1.1	Formación en el estándar IEEE 802.11 (profundización)	Formación en los niveles físico y de enlace de la tecnología WLAN. Estudio de las últimas características disponibles.	24
T.1.2	Formación en QoS (profundización)	Formación avanzada sobre técnicas de caracterización, análisis y gestión de tráfico en redes inalámbricas.	20
T.1.3	Formación en métodos de agrupamiento (profundización)	Formación avanzada en técnicas de agrupamiento. Estudio de implementación de algoritmos en herramientas software.	12
T.1.4	Formación en NS-3 (básica y profundización)	Realización de tutoriales y lectura de documentación de NS-3 para redes WLAN.	64
H.1	Fin de la fase de formación	Hito de fin de la formación y familiarización con las herramientas a utilizar en la investigación.	0
P.T.2	Especificaciones del trabajo	Definición de especificaciones del proyecto, objetivos y alcance.	20
T.2.1	Definición del proyecto	Selección de los aspectos principales del proyecto y las problemáticas generales a resolver.	5
T.2.2	Definición de los objetivos y el alcance	Definición de los objetivos del proyecto y el alcance de este.	5
T.2.3	Definición de los procedimientos a implementar	Definición preliminar de los procedimientos a diseñar en el proyecto.	10
H.2	Proyecto definido	Hito de fin de la definición del proyecto.	0
P.T.3	Estudio teórico sobre casos de uso WLAN	Estudio teórico de referencia para la parametrización y diseño de las simulaciones.	25
T.3.1	Estudio de la actualidad de la tecnología	Revisión teórica y casos de uso de IEEE 802.11.	10
T.3.2	Estudio de escenarios	Revisión documental de los escenarios de aplicación del protocolo IEEE 802.11.	15
T.3.3	Estudio de tráfico	Revisión documental de los modelos de tráfico habituales en redes WLAN.	15
H.3	Estudio teórico finalizado	Hito de fin del estudio teórico que guía la parametrización de escenarios y usuarios.	0
P.T.4	Análisis de las problemáticas de ingeniería	Análisis sobre las técnicas de clustering a utilizar y comparativa de parámetros y KPIs para a implementar en el simulador.	35
T.4.1	Análisis del estudio teórico sobre casos de uso WLAN	Análisis de las conclusiones del estudio teórico de cara a definir los parámetros y la implementación de estos sobre NS-3.	16
T.4.2	Análisis y estudio de los KPIs afines a la tecnología y tráfico de interés	Breve estudio sobre los KPIs más significativos de acuerdo con la tecnología empleada y los casos de uso en estudio.	15
T.4.3	Análisis de las técnicas de clustering	Comparativa de las técnicas de agrupamiento más extendidas para su selección y posterior implementación.	17

H.4	Decisiones técnicas finalizadas	Hito de fin de la selección de soluciones a implementar en el simulador y el procesado de datos con ML.	0
P.T.5	Diseño de la solución	Diseño de la solución en base al estudio de alternativas.	75
T.5.1	Diseño de los escenarios WLAN	Diseño de los scripts de configuración de escenarios WLAN en NS-3.	45
T.5.2	Diseño de los modelos de tráfico	Diseño de los modelos de tráfico de red en NS-3.	30
T.5.3	Diseño de la aplicación de procesado mediante clustering	Diseño de los programas de procesamiento de datos mediante clustering para la realización de los procedimientos de mejora de la QoS.	20
H.5	Diseño finalizado	Hito de fin de diseño de la solución para las simulaciones y el procesado de los datos.	0
P.T.6	Desarrollo de la solución	Realización de las simulaciones, implementación de los algoritmos y realización de pruebas.	35
T.6.1	Implementación de los algoritmos de clustering en Python	Programación de los algoritmos a través de Python. Realización de pruebas con distintos sets de datos.	15
T.6.2	Realización de las simulaciones	Realización de múltiples simulaciones por entorno. Pruebas de funcionamiento para los casos de uso del estudio.	30
T.6.3	Aplicación de los algoritmos sobre los resultados de simulación	Utilización de los programas con las técnicas de agrupamiento sobre los resultados de simulación.	5
H.6	Resultados obtenidos	Hito de fin de la implementación de la solución y la realización de pruebas.	0
P.T.7	Análisis de resultados, pruebas, comparativa y conclusiones	Análisis de los resultados de simulación y del procesado mediante clustering. Comparativa crítica de la aptitud de los algoritmos para los procedimientos de mejora de la QoS.	60
T.7.1	Análisis de aptitud de la solución de simulación	Análisis de los resultados obtenidos en las simulaciones.	30
T.7.2	Pruebas de rendimiento de clustering	Pruebas de rendimiento de los algoritmos mediante el procesado de múltiples muestras de los escenarios	15
T.7.3	Comparativa de algoritmos	Comparativa de formación de clústeres para los procedimientos de mejora de la QoS.	12
T.7.4	Extracción de conclusiones	Obtención de conclusiones a partir de los resultados y comparativas anteriores.	60
H.7	Estudio concluido	Hito de fin del estudio y análisis de técnicas de clustering aplicado a la mejora de la QoS	0
P.T.8	Documentación	Documentación de seguimiento y memoria del trabajo.	369
T.8.1	Documentación de seguimiento	Redacción del documento de seguimiento con los avances en el proyecto.	339
T.8.2	Documentación final	Elaboración de la documentación final a entregar.	30
H.8	Fin del proyecto	Hito de fin de la documentación y proyecto.	0

Tabla 90: Planificación estructurada del proyecto.

Trabajo Fin de Máster

Mikel Acha Alonso

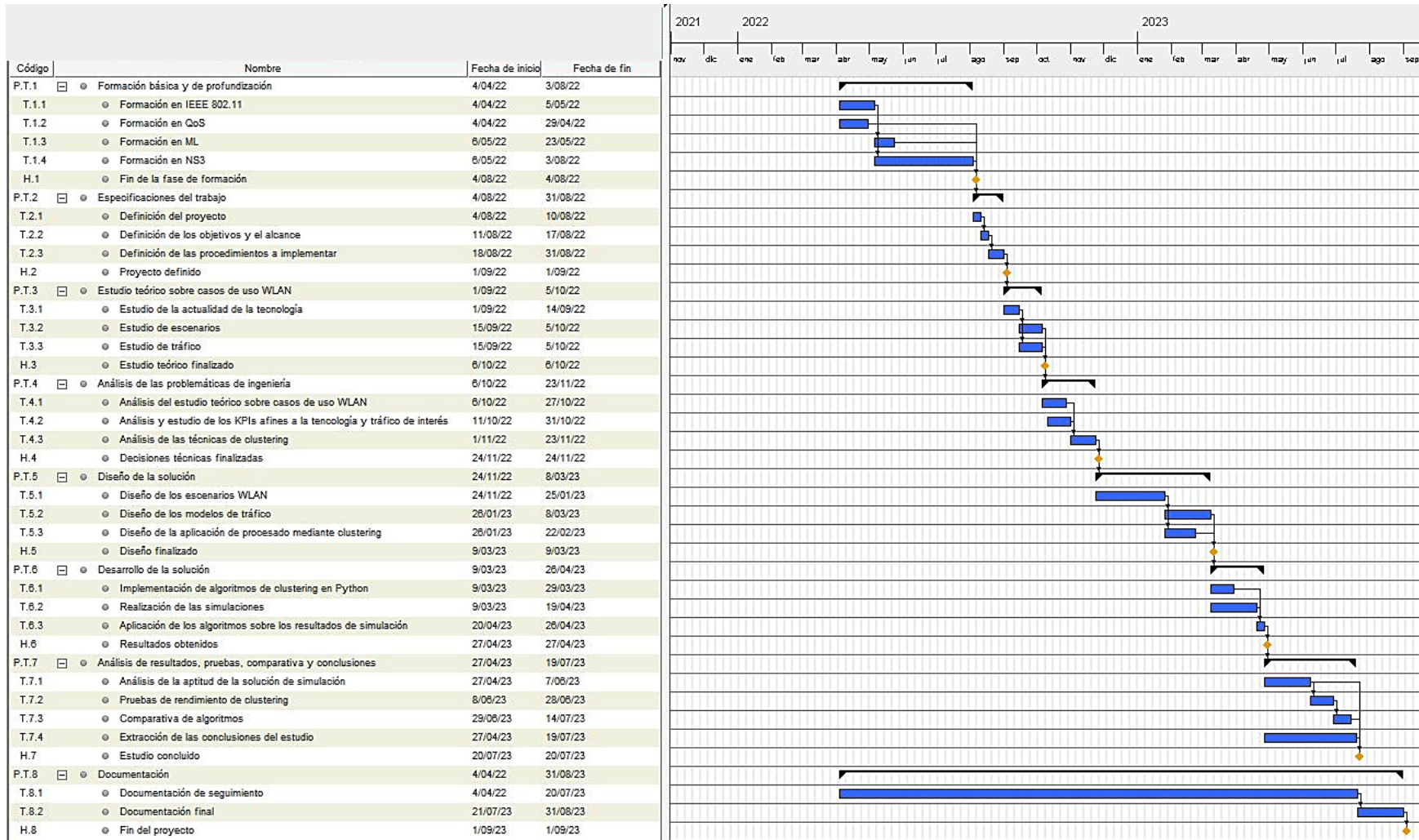


Figura 66: Diagrama de Gantt del proyecto.

11. Presupuesto

El presente apartado describe los aspectos económicos del proyecto. Para ello, se realiza una división de costes y amortizaciones derivados de la realización del trabajo mediante la separación en cuatro secciones. Las primeras tres se dedican a la definición de las partidas de horas internas, amortizaciones y gastos, seguidas por una cuarta sección dedicada a los costes totales.

11.1. Horas Internas

Las **horas internas** engloban el **tiempo dedicado** por parte del personal participe en el desarrollo del trabajo, entre los que se incluye un ingeniero junior, el alumno, bajo la supervisión de una ingeniera con experiencia, la directora. En cuanto a la dedicación, se ajusta a la realización de las tareas principales de la investigación y desarrollo en el primer caso, y de la dirección del proyecto y guía en la toma de decisiones en el segundo.

La Tabla 91 recoge los **gastos en recursos humanos** en el desarrollo del Trabajo Fin de Máster.

Horas Internas			
Concepto	Número de horas (h)	Coste horario (€/h)	Coste total (€)
Ingeniero junior	600	25	15.000
Directora	200	50	10.000
Subtotal			25.000

Tabla 91: Recursos humanos.

11.2. Amortizaciones

En la partida de amortizaciones (Tabla 92) se incluyen los recursos informáticos utilizados para el desarrollo del estudio, en el que se ha primado el uso de software libre, evitando utilizar programas que requirieran pagos de suscripciones o licencias.

De este modo, los **activos contemplados** se reducen a la utilización de dos equipos portátiles, uno con sistema operativo Windows 11 y otro con Ubuntu 22.04.

Amortizaciones				
Concepto	Coste de adquisición (€)	Vida útil (h)	Utilización (h)	Coste total (€)
HP Omen 16	850	26.280	450	14,56
ASUS TUF 15	900	26.280	300	10,27
Subtotal				24,83

Tabla 92: Amortizaciones.

11.3. Gastos

Respecto a la partida de gastos, se tienen en consideración los aspectos derivados de los materiales utilizados exclusivamente para la elaboración del trabajo.

Por consiguiente, se ponen en valor los **materiales de oficina** de uso común utilizados, y los costes provenientes de la **documentación** accedida y utilizada en el estudio, como artículos de divulgación científica y capítulos de documentación técnica.

Las entradas de la partida de gastos derivadas de la introducción anterior se hallan expuestas en la Tabla 93.

Gastos	
Concepto	Coste total (€)
Material de oficina	30
Documentación	370
Subtotal	400

Tabla 93: Gastos

11.4. Costes Totales

En definitiva, los costes que conforman el **presupuesto** ascienden a 27.977,31€, cuyo desglose en las diferentes partidas se muestra en la Tabla 94.

Costes totales	
Concepto	Coste (€)
Horas internas	25.000
Amortizaciones	24,83
Gastos	400
Subtotal	25.424,83
Imprevistos (10%)	2.552,48
Total	27.977,31

Tabla 94: Costes totales.

12. Conclusiones

El estudio realizado **cumple satisfactoriamente los objetivos** definidos en un principio, y ofrece infinidad de **aportaciones en los múltiples ámbitos** trabajados a lo largo del Trabajo Fin de Máster: simulación de redes, modelado estadístico del tráfico de red, aplicación de técnicas de aprendizaje automático y mejora de la QoS.

En primer lugar, se desarrolla un amplio Estado del Arte, cuyo cometido principal es posibilitar una **exhaustiva caracterización de la red** que permita la definición posterior de entornos realistas.

Para ello, se parte de la documentación definida por el grupo de desarrollo del estándar IEEE 802.11ax y se enriquece la definición de diferentes entornos de acuerdo con una extensa revisión documental, en la que se recogen **múltiples casos de estudio** de diferentes escenarios WLAN. Asimismo, se incluye dentro de dicha definición el **modelado de diferentes tipos de tráfico**.

A continuación, tras realizar la **selección de los parámetros y herramientas a implementar** a través del Análisis de Alternativas, se realiza una fase de aprendizaje del uso del simulador de red NS-3. Además, se lleva a cabo una definición de **KPIs** para la realización de procedimientos de **mejora de QoS** a nivel de redes y usuarios sobre la tecnología y casos de uso en estudio.

En consecuencia, se desarrollan **múltiples programas** sobre **lenguaje C++** para la **definición de escenarios** de red y **modelado de tráfico**. En dicho ámbito, se presentan como aportaciones principales la **guía de referencia** para la definición de **escenarios WLAN de última generación** en NS-3, y la contribución sobre el modelo de simulación de tráfico FTP diseñado por la CTTC [97] relativo al modelo M1 de la 3GPP, al que se le incluye la **definición del modelo FTP M2** que se corresponde con la definición empleada en entornos WLAN [72].

Después, partiendo de la selección de **algoritmos de clustering** del Análisis de Alternativas, se realizan una serie de programas en **lenguaje Python** que se adecúan al manejo de los **ficheros de salida del simulador** de red, y ofrecen múltiples funcionalidades de **representación y manejo de los datos** bajo estudio.

Una vez implementados los algoritmos, se estudia su aptitud para la realización de las tareas de **extracción del contexto del usuario y la red**. Con dicho fin, se definen dos procedimientos de mejora de la QoS: la **diferenciación de entornos** y la **clasificación de perfiles**.

El primer caso se nutre de la **definición de escenarios** para comprobar la capacidad de los algoritmos de reconocer las características implícitas a los mismos por medio de las muestras de los puntos de acceso a cada red.

El segundo, en cambio, se enfoca en las estaciones presentes en las redes y realiza una escueta aproximación para la **extracción del contexto del usuario** mediante métricas obtenidas directamente desde los dispositivos finales.

En ambos experimentos se concluye que los algoritmos de clustering tienen un **gran potencial para la obtención de las características** implícitas de la red, lo que sienta las bases para la automatización de la definición de indicadores clave sobre entornos y usuarios, que resulta un ámbito de estudio clave para la mejora en la gestión de la calidad de servicio global. Entre los algoritmos analizados, se destaca el empleo de **K-Means** debido a su buen ajuste a ambos procedimientos, su simplicidad y su eficiencia.

Además, el presente proyecto **complementa** el estudio comenzado en el **Trabajo Fin de Grado**, enriqueciéndolo con un desarrollo de mucha mayor precisión y complejidad, un ámbito de estudio más ambicioso y favoreciendo el aprendizaje de nuevas técnicas de gestión de la calidad de servicio.

Por último, es digno de mención que los resultados derivados del trabajo también contribuyen y son consolidados en un **artículo de investigación**. Dicho artículo, titulado **QoX Management for Modern Networks**, ha sido enviado al número especial Advancements in QoS/QoE for Future Networks and Their Applications de la revista Applied Science (ISSN 2076-3417).

En definitiva, el **Trabajo Fin de Máster** realizado sienta las **bases para la definición** detallada de **casos de uso realistas** de la tecnología Wi-Fi, y **revalida la aptitud** del aprendizaje no supervisado, y en concreto, **de las técnicas de clustering**, para las tareas de **mejora de la QoS**.

13. Bibliografía

- [1] Cisco. (2020). A new perspective on the modern workplace: Forward-looking lessons from real-life pandemic experiences. [Online]. Disponible: <https://www.cisco.com/c/dam/en/us/solutions/collateral/collaboration/fow-flash-survey.pdf>
- [2] Cisco. (2019). 2020 Global Networking Trends Report. [Online]. Disponible: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>
- [3] Oracle. (2020). What is IoT? [Online]. Disponible: <https://www.oracle.com/internet-of-things/what-is-iot/>
- [4] Cisco. (2020). Cisco Annual Internet Report 2018-2023 White Paper. [Online]. Disponible: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>
- [5] Gurvey, S. (2019). 5G and Wi-Fi 6: Evolution and revolution. Cisco Newsroom.
- [6] IEEE Std 802.11ax-2021 Amendment 1: Enhancements for High-Efficiency WLAN. IEEE, 2021.
- [7] Cisco. (2020). IEEE 802.11ax: The Sixth Generation of Wi-Fi White Paper. [Online]. Disponible: <https://www.cisco.com/c/en/us/solutions/collateral/wireless/white-paper-c11-741490.html>
- [8] Cisco. (2021). 2022 Global Networking Trends Report. [Online]. Disponible: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>
- [9] Cheshire, J. (2019). Understand cloud concepts. Pearson.
- [10] ITU-T. (2008). E.800: Definiciones de términos relativos a la calidad de servicio. ITU-T.
- [11] ITU-T. (2017). Quality of Service Regulation Manual. ITU-T.
- [12] ITU-T. (2017). P.10/G.100: Vocabulary for performance, quality of service and quality of experience. ITU-T.
- [13] ITU-T. (2021). Recommendation E.800SerSup9 (10/21): Supplement to E.800: Definitions of terms related to quality of service. ITU-T.
- [14] Selinis, I., Katsaros, K., Allayioti, M., Vahid, S., & Tafazolli, R. (2018). The race to 5G era: LTE and Wi-Fi. IEEE Access, 6, 62343-62362.
- [15] Kulin, M. (2021). A survey on machine learning-based performance improvement of wireless networks: PHY, MAC and network layer. Electronics, 10(3), 318.

- [16] Usama, M. (2019). Unsupervised machine learning for networking: Techniques, applications and research challenges. *IEEE Transactions on Network and Service Management*, 16(2), 498-514.
- [17] Acha, M. (2020) Análisis De Técnicas De Aprendizaje Automático Para La Mejora De La Calidad De Servicio En Redes Inalámbricas.
- [18] Barrachina-Muñoz, S., Wilhelmi, F., Selinis, I., & Bellalta, B. (2019). Komondor: a Wireless Network Simulator for Next-Generation High-Density WLANs. In *2019 Wireless Days (WD)* (pp. 1-8). IEEE.
- [19] Kurose, J. F., & Ross, K. W. (2010). *Redes de Computadoras: un enfoque descendente basado en Internet*. Addison-Wesley.
- [20] Bianchi, G. (2000). Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3), 535-547.
- [21] IEEE Std 802.11e-2005 Amendment 8: Medium Access Control. IEEE, 2005.
- [22] IEEE 802.11, The Working Group Setting the Standards for Wireless LANs. IEEE Standards, 2022.
- [23] Links, C. (2022). The Evolution of Wi-Fi networks: from IEEE 802.11 to Wi-Fi 6E. Wevolver.
- [24] Cisco Meraki. (2020). Wi-Fi 6: The Next Generation of Wireless. [Online]. Disponible: https://meraki.cisco.com/lib/pdf/meraki_whitepaper_wifi6.pdf
- [25] Cisco Meraki. (2022). Wi-Fi 6 (802.11ax) Technical Guide. Cisco Meraki.
- [26] MathWorks. (2023). Beamforming. [Online]. Disponible: <https://www.mathworks.com/discovery/beamforming.html>
- [27] Cisco. (2021). Wi-Fi 6E: The Next Great Chapter in Wi-Fi White Paper. [Online]. Disponible: <https://www.cisco.com/c/en/us/solutions/collateral/enterprise-networks/802-11ax-solution/nb-06-wi-fi-6e-wp-cte-en.html>
- [28] Cisco. (2021). What is Wi-Fi 6E. [Online]. Disponible: <https://www.cisco.com/c/en/us/products/wireless/what-is-wi-fi-6e.html>
- [29] nan, I., Keceli, F., & Ayanoglu, E. (2009). Analysis of the 802.11e enhanced distributed channel access function. *IEEE Transactions on Communications*, 57(9), 2644-2655.
- [30] Gao, D., Cai, J., Jianfei, Y., & Ngan, K. (2005). Admission control in IEEE 802.11e wireless LANs. *IEEE Network*, 19(1), 40-46.
- [31] A Survey on Machine Learning-Based Performance Improvement of Wireless Networks: PHY, MAC and Network Layer. (2021). *Electronics*, 10(6), 867.
- [32] ITU-D. (2017). *Quality of Service Regulation Manual*. Geneva, Switzerland: International Telecommunication Union.

- [33] ITU-T. (2001). G.1000 (11/2001) - Quality of service in communications: Framework and definitions. Geneva, Switzerland: International Telecommunication Union.
- [34] O. Corporation. (2010). Quality of service requirements.
- [35] ITU-T. (2019). Y.1540 - IP data communication service - Performance parameters related to the availability and transfer of IP packets. Geneva, Switzerland: International Telecommunication Union.
- [36] Kim, M. S., Hwang, J. W., Gwak, G.-H., Lee, L. H., & Lee, J. (2003). QoS parameters to network performance metrics mapping for SLA monitoring. In Proceedings of the 2003 ACM/IEEE conference on Communications architectures and protocols (pp. 163-173). ACM.
- [37] Blogs, M. (2020). Mean time to restore service (MTRS) IT service management.
- [38] ITU-T. (2011). Y.1541 (12/2011) - Network performance objectives for IP-based services. Geneva, Switzerland: International Telecommunication Union.
- [39] ITU-T. (2016). G.1011 (07/2016) - Reference guide to quality of experience assessment methodologies. Geneva, Switzerland: International Telecommunication Union.
- [40] BEREC. (2014). Monitoring quality of internet access services in the context of net neutrality.
- [41] ITU-T. (2014). E.804 (02/2014) - QoS aspects for popular services in mobile networks. Geneva, Switzerland: International Telecommunication Union.
- [42] ITU-T. (2022). E.803 (07/2022) - Quality of service parameters for supporting service aspects. Geneva, Switzerland: International Telecommunication Union.
- [43] Cintra, M. D., Carballido, V., Carlos, C., Costa, C., & Ibarrola, L. C. E. (2014). A machine learning management model for QoE enhancement in next-generation wireless ecosystems. *IEEE Transactions on Vehicular Technology*, 63(9), 4151-4165.
- [44] Qin, D., Zhang, Z., Yu, Y., Shu, K., Dou, W., & Qin, D. (2008). Combining user profiles and situation contexts for spontaneous service. In Proceedings of the 17th international conference on World Wide Web (pp. 1157-1158). ACM.
- [45] Kihl, M., Aurelius, A., Lagerstedt, C., Ödling, P. (2010). Traffic analysis and characterization of Internet user behavior.
- [46] Task Group AX. Common Parameters for All Simulation Scenarios. IEEE Mentor.
- [47] Sandoval, J., Cespedes, S. (2021). Performance Evaluation of IEEE 802.11ax for Residential Networks. *LATINCOM 2021*.
- [48] Nguyen, K. A. (2017). A performance guaranteed indoor positioning system using conformal prediction and the WiFi signal strength. *Journal of Information and Telecommunication*.

- [49] Rath, H. K., Timmadasari, S., Panigrahi, B., & Simha, A. (2017). Realistic indoor path loss modeling for regular WiFi operations in India. 2017 Twenty-third National Conference on Communications (NCC), 1-6.
- [50] 3GPP. (2020). Study on Channel Model for Frequencies from 0.5 to 100 GHz (Release16).
- [51] Oliveira, L., Obraczka, K., & Rodríguez, A. (2016). Characterizing user activity in WiFi networks: university campus and urban area case studies. 2016 IEEE International Conference on Communications (ICC), 1-6.
- [52] O. Systems. (2019). Deploy PoE Anywhere and Everywhere
- [53] D. Tynan, «Colleges tackle the great outdoors in campus Wi-Fi upgrades,» *EdTech*, 2018.
- [54] P. P. (2013). Deploying external WiFi on a campus based network.
- [55] Oliveira, L., Obraczka, K., & Rodríguez, A. (2016). Characterizing user activity in WiFi networks: university campus and urban area case studies. 2016 IEEE International Conference on Communications (ICC), 1-6.
- [56] Sundaresan, S., Feamster, N., & Teixeira, R. (2015). Measuring the performance of user traffic in home wireless networks. Passive and Active Network Measurement Conference, 1-6.
- [57] Krishnan, E. T. E. R. I. S. N. M., & Krishnan, K. D. N. N. (2018). D-MIMOO – Distributed MIMO for Office Wi-Fi Networks. 2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), 1-6
- [58] Skendi, B. K., & Skendi, L. L. A. (2019). Cost-effective transition to Wi-Fi 5 standard in Small Office / Home Office (SOHO) environment. 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 1-4.
- [59] Giordano, A. G.-R. L. H., & Giordano, D. L.-P. L. (2020). Next Generation Wi-Fi: Deployment Guidelines and Benefits of Massive MIMO for the Enterprise. 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), 1-6.
- [60] Adame, M. C., & T., B. B. (2019). The TMB path loss model for 5 GHz indoor WiFi scenarios: On the empirical relationship between RSSI, MCS, and spatial streams. *IEEE Access*, 7, 133225-133239.
- [61] Bhat, B. R. A. C. A. K., & Mishra, P. K. N. (2015). Usage of 802.11n in practice: a measurement study. *IEEE Transactions on Mobile Computing*, 14(12), 2712-2724.
- [62] Rathore, R. G. R., & Joshi, R. D. (2021). Performance Analysis of IEEE802.11ax (Wi-Fi 6) Technology using Multi-user MIMO and Up-Link OFDMA for Dense Environment. 2021 IEEE 2nd International Conference on Applied Electromagnetics, Signal Processing, & Communication (AESPC), 1-6.

- [63] Kaya, A., Calin, D., & Viswanathan, H. (2015). On the performance of stadium high density carrier Wi-Fi enabled LTE small cell deployments. 2015 IEEE International Conference on Communications (ICC), 5382-5387.
- [64] Fuxjager, P. R. P., & H. R. F. I. G. (2011). Measurement based small-cell coverage analysis for urban macro-offload scenarios. 2011 International Conference on Wireless Communications and Mobile Computing (IWCMC), 285-290.
- [65] Mir, M. R., & K. J. (2016). Experimental Results For the Propagation of Outdoor IEEE 802.11 Links. 2016 International Conference on Wireless Communications and Signal Processing (WCSP), 1-6.
- [66] Campo Muñoz, W. Y., Astaiza Hoyos, E., & Muñoz Sanabria, L. (2017). Traffic modelling of the VoD service through NS-3. 2017 International Conference on Information Networking (ICOIN), 521-526.
- [67] C. M., Implementing Mechanisms to Improve the Performance of LTE-WLAN Aggregation, 2017.
- [68] Pokhrel, S. R. V. H. L. C. A. L., & Adame, B. B. B. (2020). Adaptive admission control for IoT applications in home WiFi networks. IEEE Transactions on Mobile Computing, 19(1), 244-258.
- [69] da Hora, D., van Doorselaer, K., van Oost, K., & Teixeira, R. (2018). Predicting the effect of home WiFi quality on QoE. IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, 216-224.
- [70] Scalable Ground-Truth Annotation for Video QoE. (2018).
- [71] Wu, W., Liu, Y., Yao, J., Fang, X., Shan, F., Yang, M., Ling, Z., & Luo, J. (2022). Learning-aided client association control for high-density WLANs.
- [72] TGax. (2016). 11ax Evaluation Methodology.
- [73] IEEE Mentor. (2013). IEEE 802.11-13/0722r1 - Traffic Model Updates to Evaluation Methodology.
- [74] Ammar, T. G. I. (2011). A new tool for generating realistic Internet traffic in NS-3.
- [75] Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding Machine Learning. From Theory to Algorithms. Cambridge University Press.
- [76] Hastie, J., & Kirsch, D. (2020). Machine Learning for Dummies. John Wiley & Sons.
- [77] Burkov, A. (2019). The Hundred-Page Machine Learning Book. Andriy Burkov.
- [78] Barber, D. (2010). Bayesian Reasoning and Machine Learning. Cambridge University Press.
- [79] Sammut, G. I. W. C. (2017). Encyclopedia of Machine Learning and Data Mining. Springer.

- [80] Google Developers. (2018). What is clustering?
- [81] Xu, D. X., & Tian, Y. (2015). A Comprehensive Survey of Clustering Algorithms. Springer.
- [82] Google Developers. (2018). Algoritmos de agrupamiento.
- [83] Elsadek, A. E., & Elsadek, M. (2017). Developing a Neural Networks Model for Evaluating Financial Performance of Residential Companies based on FCM. *International Journal of Advanced Computer Science and Applications*, 8(1).
- [84] M. Documentation. (2023). Graph Clustering.
- [85] Uk, I. (2017). A Review of Various Clustering Techniques. *International Journal of Computer Applications*, 165(11).
- [86] Wang, C. W. D. S., & Liu, G. S. F. (2013). Investigation of Absorption and Scattering Properties of Soot Aggregates of Different Fractal Dimension at 532 nm Using RDG and GMM. *Applied Optics*, 52(15), 3602-3608.
- [87] Ran, D. G. R., & Chen, H. (2013). A Scalable Self-organizing Map Algorithm for Textual Classification: A Neural Network Approach to Thesaurus Generation.
- [88] nsnam. (2023). NS3 Documentation.
- [89] Ngai, S. N., y Yau, Y. C. (2010). Investigation on QoS of Campus-wide WiFi Networks. *IEEE Transactions on Consumer Electronics*, 56(3), 1138-1147.
- [90] TGax. (2014). Proposed 802.11ax Functional Requirements.
- [91] S.-I. developers. (2023). Scikit Learn: Clustering.
- [92] VanderPlas, J. (2016). Python Data Science Handbook.
- [93] Verma, Y. (2021). Guide to BIRCH clustering algorithm with Python codes.
- [94] Guha, R. R., & S. S. Guha. (2000). CURE: An Efficient Clustering Algorithm for Large Databases.
- [95] Guha, R. R., & S. S. Guha. (2000). ROCK: A Robust Clustering Algorithm for Categorical Attributes.
- [96] Muñoz, W. Y. C. (2017). Traffic modelling of the video-on-demand service through NS-3.
- [97] CTTC. (2023). 3GPP NR ns-3 module.
- [98] 3GPP. (2015). TR36.814 Release 9.
- [99] Castellano, P. e. (2020). Introducción a Pandas.
- [100] Sáez, A. (2020). Breve Introducción A La Librería Numpy, Infinitos Contrastes.

- [101] Scikit-learn. (2020). Scikit-Learn: Machine Learning In Python — Scikit-Learn 0.23.1 Documentation.
- [102] Ali, A. R. (2023). Introducción a la librería Matplotlib de Python.
- [103] Rodríguez, D. (2018). Visualización de datos en Python con Seaborn.
- [104] Machine Learning, A. M. (2023). K-Means en Python paso a paso.
- [105] Maklin, C. (2019). DBSCAN Python Example: The Optimal Value For Epsilon (EPS).
- [106] Chakrabarti, A. (2011). Akaike Information Criterion.
- [107] Patro, A., Govindan, S., & Banerjee, S. (2013). Observing home wireless experience through WiFi APs.
- [108] Pefkianakis, I., et al. (2015). Characterizing home wireless performance: the gateway view. IEEE INFOCOM 2015.
- [109] Oh, S. H., & Kim, J. (2021). WiFi Positioning in 3GPP Indoor Office with Modified Particle Swarm Optimization.
- [110] Incelligent. (2015). Airport Wi-Fi challenges.
- [111] Company, A. (2022). Dubai Mall builds service difference on limitless Wi-Fi.
- [112] Afanasyev, M., Chen, T., Voelker, G. M., & Snoeren, A. C. (2010). Usage patterns in an urban WiFi network. IEEE/ACM Transactions on Networking, 18(4), 1170-1183.
- [113] Sridhar, G. R., Bhide, C. B., & Zhao, R. J. M. (2014). Alleviating airport WiFi congestion: a comparison of 2.4GHz and 5GHz Wifi. Texas Symposium on Wireless and Microwave Circuits and Systems, Waco, TX, USA, 2014.
- [114] IEEE 802.11 group. (2017). IEEE 802.11 HEW SG proposed CSD.
- [115] ns-3 project. (2023). NS-3 Tutorial - Release ns-3.38.1.
- [116] ns-3 project. (2023). NS-3 Model Library - Release ns-3.38.1.
- [117] ns-3 project. (2023). NS-3 Manual - Release ns-3.38.1.
- [118] Qu, Q., Li, B., Yang, M., et al. (2019). Survey and Performance Evaluation of the Upcoming Next Generation WLAN Standard - IEEE 802.11ax.
- [119] Amewuda, A. B., Katsriku, F. A., & Abdulai, J. D. (2018). Implementation and Evaluation of 802.11ac for residential networks in NS3.
- [120] Lyu, S., & Münz, G. C. G. (2006). Traffic Anomaly Detection Using K-Means Clustering.
- [121] Thing, V. L. L. (2017). IEEE 802.11 Network Anomaly Detection and Attack Classification: A Deep Learning Approach.

- [122] Kolas, G. K. A. S., & Stavrou, A. G. C. (2016). Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset.
- [123] Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network Anomaly Detection: Methods, Systems and Tools.

14. Anexo I: Estudio Detallado de los Escenarios Wi-Fi

Este anexo cubre la información completa extraída de la revisión documental sobre los casos de aplicación de la tecnología WLAN, con el ánimo de complementar al apartado Resumen del Estudio de Escenarios de Aplicación del Estándar IEEE 802.11.

14.1. Aspectos Topológicos de los Cuatro Escenarios Bajo Estudio

Como se resume en el apartado Definición Espacial y Topología, con relación a los aspectos topológicos en los casos de uso de la tecnología IEEE 802.11, se estudian las **dimensiones** de los entornos, las **características específicas** a tener en cuenta en cada caso, el **posicionamiento** tanto de los AP como las STA, y la **movilidad** de los mismos.

En el caso de los **entornos residenciales**, se recogen artículos donde se plantean despliegues de red con APs estáticos y STAs de distinto tipo sobre hogares –en su mayoría apartamentos con varias habitaciones– donde se toma como referencia el análisis de muestras de tráfico obtenidas desde los APs instalados.

En la investigación de entornos de aplicación de Wi-Fi residenciales centrada en aplicaciones de última generación –específicamente la introducción de dispositivos IoT– se propone un escenario simple [68] en el que se establece una red compuesta por un único AP y varias STAs. Aunque en los ejemplos incluidos muestran hasta 12 dispositivos terminales, la propuesta de entorno de simulación tan solo incluye dos, donde estos difieren en características al tratarse de un dispositivo IoT y un teléfono móvil.

Otro estudio que presenta un entorno de aplicación sencillo [69] trata de predecir el efecto de la calidad en redes Wi-Fi residenciales sobre la QoE. En el mismo, se toman como base las muestras de tráfico obtenidas de 3.470 AP de clientes de un ISP residencial. Para definir entornos de simulación, se establece un entorno con un AP y dos STA de diferentes características. Esta vez, sin embargo, se añade un tercer STA para tratar de simular la existencia de interferencia en la red, forzando para ello una ocupación gradual del medio compartido.

Un análisis de tráfico sobre un despliegue más reducido [107] presenta los resultados del análisis de tráfico en redes inalámbricas de 30 pisos durante seis meses a través de APs propietarios. Las STA se corresponden a dispositivos típicos como portátiles, impresoras, smartphones, tablets, consolas de videojuegos, Smart TV o cámaras de seguridad. El despliegue de red se compone de varios pisos en edificios residenciales con apartamentos de tamaño medio –con una o dos habitaciones por piso– con APs compartidos, estableciendo por lo general entre dos y cuatro AP por planta.

De forma similar, un estudio sobre 167 hogares donde se dispone de redes inalámbricas durante un periodo de cuatro meses [108] resalta la concentración de una gran parte del tráfico en un número bajo de dispositivos, habitualmente entre dos o tres STA en las horas activas donde se consumen más servicios de red. El total de dispositivos distintos en cada hogar varía entre 1 y 25, siendo la mediana 7 dispositivos –incluidos los no

residentes o visitantes– y 13 el valor máximo de dispositivos pertenecientes a los habitantes de estos.

Un caso de aplicación actual [47] donde se hace una comparativa de los últimos estándares IEEE 802.11 sobre entornos residenciales mediante herramientas de simulación, propone un escenario doméstico de alrededor de 80 m² con forma cuadrada donde se ubica un único AP en un lateral del plano, dando servicio a un número de STAs variable entre 4 y 16. Se contemplan tanto usuarios móviles como fijos.

Para finalizar con la información extraída con relación a la definición espacial y topología del ámbito residencial, TGax [46] presenta una propuesta de escenario para la simulación de entornos residenciales. En ella, se define un edificio de cinco plantas con apartamentos de 100 m² y 3 m de altura dispuestos como se observa en la Figura 67.

La configuración propuesta se compone de un AP por apartamento y un número de STAs que varía entre 1 y 10. Se toma como referencia una altura de 1.5 m tanto para las STA como los AP. También se contempla la existencia de paredes y pisos a atravesar por la señal, aunque no se propone ningún valor por defecto.

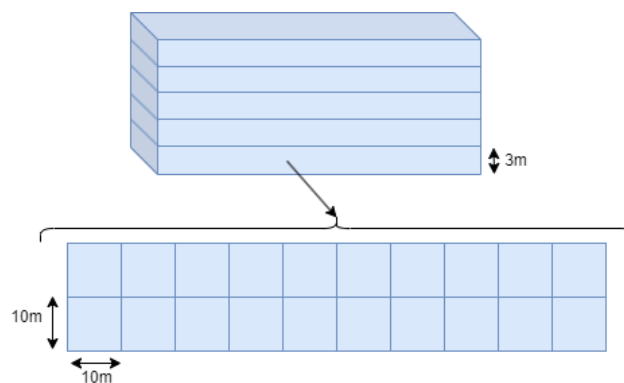


Figura 67: Disposición de escenario residencial propuesta en [46].

En el **ámbito empresarial**, se recogen artículos donde se plantean despliegues de red con AP y STA estáticos sobre entornos de oficina, en su mayoría centrados en la toma de muestras de señal para el estudio de sistemas de posicionamiento Wi-Fi.

Un estudio donde se realizan medidas sobre escenarios reales WLAN de oficinas conforma la propuesta de un modelo de confianza [48] que trata de reflejar la incertidumbre de la predicción de posicionamiento indoor, y en cuya validación se realiza la toma de medidas aplicable al presente estudio. Gracias a ella, se permite caracterizar parcialmente la topología real de entornos de oficina.

Uno de los escenarios analizados se trata de un edificio de oficinas en forma de L con unas dimensiones de 45.4 m × 32.6 m, en el que se dispone de un total de 27 oficinas interconectadas mediante tres pasillos. Dentro de la planta bajo estudio se hallan un total de 9 APs, y las medidas tomadas para la detección de puntos de acceso reflejan la existencia de al menos el triple de APs alcanzables desde el mismo entorno.

Otro escenario destacable es una planta dedicada a laboratorios de computación de una universidad, formada por salas –la toma de medidas del artículo se realiza en una

habitación con dimensiones de 6.1 m × 4.8 m– a ambos lados de un corredor de 45 m de largo, que se corresponde también con la estructura de edificios de oficina. En la misma se dispone de un total de cinco APs.

Por otro lado, en un estudio sobre la propagación de la señal Wi-Fi en entornos realistas de oficina [49], se toma como referencia un edificio de tres plantas con unas dimensiones de 600 m² cada una. Cada piso se conforma a partir de cubículos con puestos de oficina fijos, pasillos que conectan los cubículos, y algún espacio abierto compartido entre toda la oficina.

Además de los escenarios descritos anteriormente, en el estudio del modelo de canal para frecuencias entre 0.5 y 100 GHz de la 3GPP [50] se definen también entornos de oficina como espacios de 120 m × 50 m (con una altura de 3 m) donde se establecen un total de 12 estaciones base –contempladas como APs en [109] para el caso de IEEE 802.11– en forma de cuadrícula, es decir, separadas entre sí por una distancia de 20 m tanto en horizontal como en vertical (Figura 68).

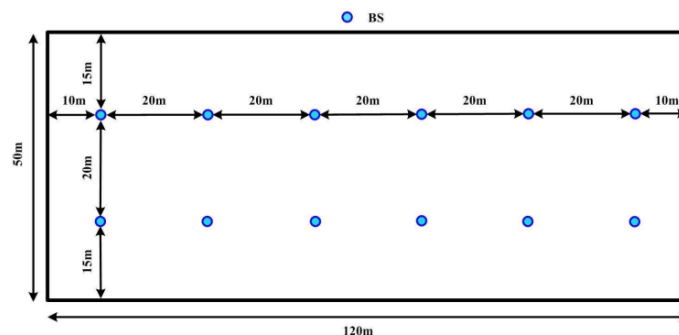


Figura 68: Disposición de escenario empresarial propuesta en [50].

Para terminar con el análisis topológico del ámbito empresarial, el TGax [46] describe en detalle su propuesta de características para la simulación de dicho entorno. En ella, se lleva a cabo un planteamiento mixto, tratando de cubrir tanto empresas pequeñas, consistentes en pequeñas oficinas o habitaciones, como en medianas empresas, compuestas por un rango comprendido entre dos y cuatro oficinas.

Se define entonces un escenario compuesto por hasta 8 oficinas con un reparto de 4 AP por oficina. Las dimensiones de cada una son 20 m × 20 m como se puede apreciar en la Figura 69. La disposición de los puestos donde se ubican de forma aleatoria y estática –sin movilidad– las STA se definen mediante cubículos de 2 m × 2 m.

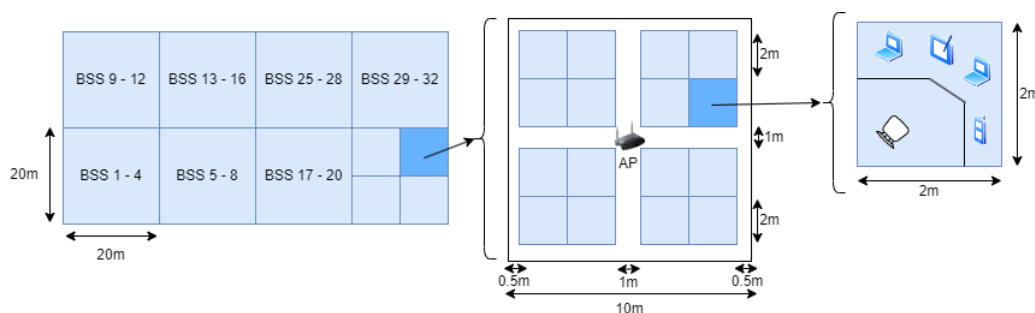


Figura 69: Disposición de escenario empresarial propuesta en [46].

Con relación a **los entornos de interior de alta densidad**, se realiza una búsqueda de información relativa a escenarios donde se establecen múltiples BSSs de pequeño tamaño distribuidos de forma uniforme, como pueden ser, entre otros, aeropuertos, estaciones de tren, espacios comerciales de interior o despliegues para dar acogida a eventos de diferentes características. Se asume, por tanto, alta densidad tanto de APs como de STAs.

En algunos estudios con múltiples tecnologías se incluyen despliegues de redes inalámbricas. Un ejemplo de ello son los diferentes casos de estudio de aplicación de tecnología PoE [52], donde también se propone la coexistencia con la tecnología Wi-Fi. Entre los despliegues recogidos, destaca el establecimiento de redes inalámbricas sobre un aeropuerto de aproximadamente 631.740 m² en el que se colocan un total de 300 APs para acoger hasta a 15.000 usuarios simultáneos. Además, también se plantea un caso de estudio sobre un centro comercial, en el que se establecen APs tanto en el interior como en el exterior de forma próxima a cámaras de seguridad IP y cuyo despliegue se establece por zonas.

Otro caso de aplicación de entorno de alta densidad que toma como referencia un aeropuerto [110] plantea el despliegue de infraestructura en función de las características cambiantes del tráfico –alta movilidad, picos de densidad de usuarios en las redes, cambios súbitos en el número de usuarios general– instalando un total de 200 Aps en el escenario compuesto por varios edificios.

Respecto a las aplicaciones reales de última generación Wi-Fi, un caso de uso práctico lo conforma el establecimiento de 2.200 APs con tecnología Wi-Fi 6 para dar cobertura a 40.000 usuarios móviles diarios sobre un centro comercial de 502.000 m² [111].

Además de estudios sobre despliegues en entornos como terminales de aeropuertos o espacios indoor de centros comerciales, un modelo distinto de entorno de alta densidad puede ser el caso de uso de un estadio donde tienen lugar eventos en un espacio cerrado. En una propuesta de simulaciones sobre entornos de este tipo [63], se proponen dos despliegues de diferente densidad de APs. El primero cubre el área bajo estudio con 62 APs colocados a una distancia de entre 30 y 40 m, dando cobertura a 1.240 usuarios. El segundo, en cambio, duplica el número de APs estableciendo 124 APs a una distancia de entre 15 y 20 m para acoger hasta 2.480 STAs. Con ello, el planteamiento es que cada AP provea de conexión hasta 320 asientos del estadio, contemplando hasta 20 usuarios simultáneos en el mismo.

Del mismo modo, otro escenario interior afín al entorno bajo estudio es el análisis de un auditorio [51] donde se despliegan cuatro APs sobre un espacio de 33.5 m × 18 m × 8 m para dar cobertura a 195 usuarios, con un máximo de 32 usuarios por AP.

Por último, el TGax [46] también propone un escenario de interior de alta densidad, que en este caso está formado por una infraestructura de red formada por múltiples BSS desplegados en forma hexagonal (Figura 70).

Se dispone de una arquitectura en forma de celdas hexagonales –regulares y simétricas– con 8.5 m de apotema y separados a una distancia de 17 m, en las cuales se

coloca un AP en el centro a una altura de 3 m. Las STAs se ubican de forma aleatoria en todo el plano, asignando una cantidad de entre 1 y 30 terminales que pueden disponer de movilidad por AP.

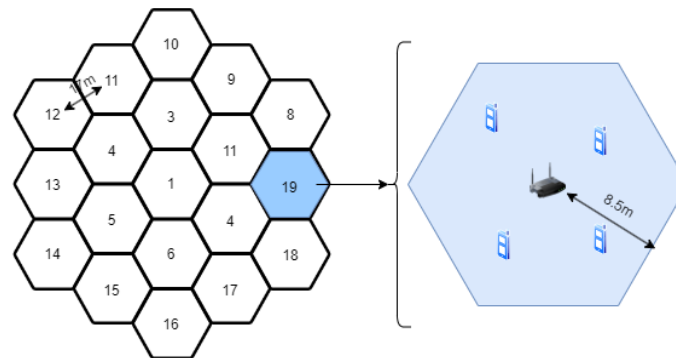


Figura 70: Disposición de escenario indoor de alta densidad propuesta en [46].

Por último, respecto a los **entornos de exterior de alta densidad**, se estudian entornos de grandes dimensiones de exterior donde se aprecia gran densidad de STAs y una disposición de APs de tal forma que haya una distancia considerable entre ellos. Entre los ejemplos de casos reales de estos escenarios destacan los despliegues de red en ciudad y los campus universitarios abiertos.

Una investigación sobre esta clase de escenarios [55] tiene lugar en la caracterización de entornos Wi-Fi reales en espacios abiertos a nivel de usuario en base a muestras de tráfico y conocimiento del medio.

En primer lugar, se estudia un área urbana donde se establecen 340 APs para dar servicio a 45.000 usuarios de una ciudad. El despliegue se lleva a cabo colocando los puntos de acceso en espacios accesibles públicamente, cubriendo desde parques hasta calles comerciales populares. Cabe destacar que el número de APs activos varía a lo largo de la muestra, disponiendo de 185 en el último mes del estudio. Asimismo, la mayoría de los usuarios se conectan a menos de 40 APs.

Por otro lado, el mismo estudio contempla un campus universitario de grandes dimensiones donde se dispone de 3.300 APs para dar servicio a 24.000 usuarios. Al tratarse de una universidad con disposición compacta, los APs dan cobertura outdoor suficiente realizando la instalación en los propios edificios. Se destaca que la mayoría de los usuarios se conecta a menos de 400 APs.

Otra investigación sobre patrones de uso en redes Wi-Fi urbanas [112] destaca la presencia dominante de dispositivos móviles. La red bajo análisis está compuesta por 500 APs repartidos en forma mallada a lo largo de la ciudad y se observan máximos de 2.500 clientes activos en la misma en intervalos de 15 minutos.

En el estudio de la demanda de acceso Wi-Fi outdoor en crecimiento sobre entornos universitarios [54] también se realiza un estudio de implantación de APs en exteriores. En este caso, se ubica en una gran universidad de 1.768.476 m², y que cuenta con 1.300 APs instalados en el interior de sus edificios. En las pruebas prácticas realizadas en dicho

estudio para la extracción de conclusiones, se realiza la instalación de 10 APs en puntos críticos –áreas comunes, lugares de encuentro de estudiantes, avenidas deportivas– para inferir la necesidad de una mayor planificación, investigación y despliegue de la red.

Otros ejemplos de escenarios universitarios outdoor [53] recogen la información de tres universidades que ayuda a la deducción de características de dichos entornos. En la primera de ellas se destaca que la mitad de los 33.000 dispositivos que se conectan a la red inalámbrica universitaria son smartphones. En la segunda, se describe un escenario de 1.093.000 m² compuesto por 500 APs y que permite el soporte de hasta 450 usuarios por AP. Por último, la tercera universidad consta de 3.642.000 m² con una cantidad de 1.100 APs –tanto interiores como exteriores– para dar acceso a la red a 11.000 estudiantes.

Para finalizar con la información extraída en relación con la definición espacial y topología de los distintos entornos, el TGax [46] propone un escenario de exterior de alta densidad formado por una infraestructura de red con múltiples BSS desplegados en forma hexagonal (Figura 71).

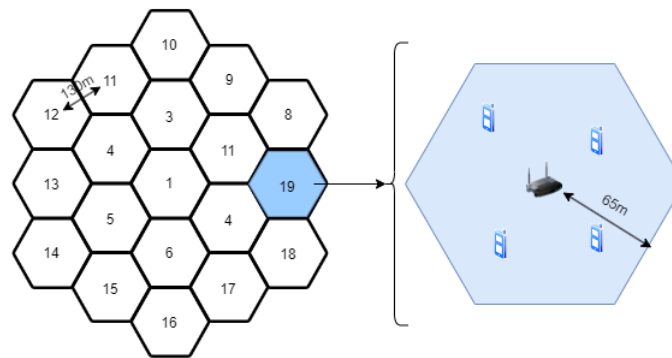


Figura 71: Disposición de escenario outdoor de alta densidad propuesta en [46].

De forma similar a la disposición propuesta en el mismo documento para el escenario anterior [46], se propone una arquitectura en forma de celdas hexagonales –regulares y simétricas– de gran tamaño, con 65 m de apotema, y separados los APs a una distancia de 130m, que se colocan en el centro de cada celda a una altura de 10 m. En este caso destaca una mayor concentración de STAs y una separación considerable entre los APs. Las STAs se ubican de forma aleatoria en todo el plano, asignando una cantidad de entre 1 y 50 terminales que pueden disponer de movilidad por AP.

14.2. Características Físicas, a Nivel de Enlace y Configuración de la Transmisión de los Cuatro Escenarios Bajo Estudio

Como se resume en el apartado Características Físicas, a Nivel de Enlace y Configuración de la, a continuación, se desarrolla la información analizada al completo para los cuatro casos bajo estudio en términos de la **banda frecuencial** a utilizar –2,4 GHz, 5 GHz o 6 GHz–, la **configuración de canales** en base a la selección y uso de los canales sobre las bandas estudiadas, el **ancho de canal** –20 MHz, 40 MHz, 80 MHz o 160 MHz– aplicable a cada caso, el uso de **características MIMO** y el **número de antenas** a utilizar tanto en Aps como en STAs, la **potencia de transmisión** de estos, las **configuraciones OFDM**

que se ajustan a la tecnología bajo estudio, y los **modelos de propagación** afines a los entornos analizados.

Asimismo, se presenta la información específica de **nivel de enlace** como el uso de **EDCA**, la aplicación de métodos de **agregación**, la configuración de **retransmisiones**, y el empleo de **RTS/CTS**.

En primer lugar, se extraen las características físicas y configuraciones del **ámbito residencial**.

En relación con los factores a tener en cuenta que se exponían al principio del apartado, ha de mencionarse una amplia utilización de algunas versiones del estándar IEEE 802.11 a lo largo de la revisión de documentos sobre entornos residenciales, de forma que, aun habiéndose presentado protocolos más actualizados de acuerdo a las fechas de desarrollo de los artículos, predominan como referencia a la hora de realizar simulaciones y análisis de casos reales.

En concreto, la versión con mayor recurrencia entre los documentos donde se estudian hogares y apartamentos se trata del estándar 802.11n –que aparece en [68] [69] [107] [108] [56]– seguido por las versiones 802.11g [107] [56], 802.11ac [69] [47] y 802.11ax [46] [47].

El uso de las versiones mencionadas afecta directamente a la disponibilidad de gran parte de las características, como la disponibilidad de bandas de frecuencia, tamaños de canal, o posibilidad de utilizar MIMO y OFDM.

En cuanto a las bandas utilizadas, en la gran mayoría de los artículos donde se menciona explícitamente cuál se utiliza o propone para cada estudio [46] [68] [107] [108] [47] [56] se incluye la banda de 2,4 GHz. En un estudio sobre 60 hogares pertenecientes a distintos países [56] se aprecia una utilización casi cuatro veces mayor para la banda de 2,4GHz frente a la de 5 GHz. Asimismo, en los documentos donde se proponen escenarios actuales se consideran también la banda de 5 GHz [46] [47] y la de 6 GHz [46].

El reparto de bandas de frecuencia y la versión del estándar utilizada también afecta directamente al uso de unos tamaños de canal u otros. En general, en la documentación analizada predomina el uso de canales de 20 MHz, el tamaño por defecto para la banda de 2,4 GHz. La caracterización residencial sobre la banda de 5GHz [46] [108] [47] considera por lo general canales de 40 MHz. También se utilizan 80 MHz tan sólo para versiones 802.11ac y 802.11ax [46] [47] y 160 MHz en el caso de 802.11ax [47].

Con relación a las características MIMO y el número de antenas utilizadas, un estudio sobre la QoE en entornos domésticos [69] incluye configuraciones 2×2 para los APs, y tanto 1×1 como 2×2 para las STAs, haciendo uso de un único flujo espacial. Otra investigación sobre calidad en entornos residenciales [108] también establece la configuración de antenas de APs 2×2 con modos MIMO para diversidad espacial y multiplexación, y en el caso concreto de la evaluación del rendimiento de las últimas versiones Wi-Fi [47] se propone el uso de MIMO con configuraciones de 3 flujos para 802.11ac y 8 para 802.11ax. Además, en la propuesta de parámetros para la definición

de escenarios de TGax [46] se definen configuraciones de 2×2 o 4×4 para los APs y 1×1 o 2×2 para las STAs.

Respecto a la potencia de transmisión, se hacen uso de APs con 17 dBm [107], 19 dBm [108] o 21 dBm [47], y STAs con 14 dBm [108], 16 dBm [108] o 18 dBm [47].

Para finalizar con las características físicas del entorno residencial, el uso de OFDM tan sólo se considera en los estudios de las últimas versiones Wi-Fi [47] [46], especificando el establecimiento de intervalos de guarda bajos (800 ns). Del mismo modo, en los artículos se proponen distintos modelos de propagación. Uno de ellos [52] considera un modelo de pérdidas por espacio libre donde incluye atenuación causada por elementos obstruyentes como paredes. El TGax [47], sin embargo, considera pérdidas logarítmicas en función de la distancia en base a tres (ThreeLogDistance) y añade atenuaciones debidas a que la señal puede atravesar paredes o pisos –también se contempla fading y shadowing–.

Con relación a las especificaciones MAC, el TGax [46] definen como parámetros de simulación para IEEE 802.11ax en entornos domésticos la utilización de EDCA con los valores específicos por defecto para cada clase de tráfico. En cuanto a la gestión de canales, se utiliza asignación aleatoria de canales no solapados. También se define agregación únicamente a nivel de A-MPDUs –con un tamaño de agregación de 64 MPDUs y un tamaño de ventana correspondiente a BA– por lo que no se utilizan A-MSDUs, y se establece un límite máximo de reintentos de envío establecido a 10. Por último, no se hace uso de RTS/CTS, y se presupone gestión independiente de los APs.

Por otra parte, en estudios sobre la introducción de aplicaciones IoT en entornos Wi-Fi [68] también se incluyen múltiples parámetros a nivel de enlace, para IEEE 802.11g/n en este caso. Entre ellos, se establece como 7 el número máximo de reintentos de envío y tampoco se hace uso de RTS/CTS. Para los tiempos de trama se utilizan valores estándar.

A continuación, se recogen las características físicas y configuraciones del **ámbito empresarial**.

De forma paralela al entorno anterior, la documentación analizada dispone de una mayor presencia de ciertas versiones del protocolo. Del mismo modo que en el ámbito residencial, entre los artículos estudiados las características observadas con más recurrencia son las del estándar IEEE 802.11n –cuyo uso es contemplado sobre los entornos de oficina [49] [57] [58]– seguido por IEEE 802.11ac [57] [59] y por IEEE 802.11ax [46] [59].

En relación a las bandas frecuenciales, predomina la utilización de la banda de 5 GHz [46] [57] [59] [58] sobre la de 2,4 GHz [46] [49] [58]. Asimismo, ha de considerarse el hecho de que el ancho de canal a seleccionar esté relacionado a la banda utilizada, y en este caso se destaca el uso de canales de 80 MHz sobre la banda de 5 GHz [46] [57] [58] [59], y se consideran 20 MHz en el caso de 2,4 GHz [46] [49] [58].

Respecto a las características MIMO y las configuraciones de antenas utilizadas en los dispositivos, el TGax [46] propone el uso de APs con 4×4 antenas y STAs de 1×1 o 2×2

antenas. En otra propuesta específica de MIMO para entornos de oficina [57], sin embargo, reducen el número de antenas incluyendo tan solo 2 x 2 antenas para los APs. Por otro lado, en estudios sobre Wi-Fi 5 para oficinas [58] se amplía el uso de MIMO con múltiples flujos espaciales, pero en casos prácticos –debido a limitaciones de equipamiento– se establecen configuraciones 1 x 1. Por último, en estudios sobre los beneficios de MIMO en las últimas generaciones de Wi-Fi [59] se proponen múltiples configuraciones para los APs: 2 x 2, 4 x 4 y 10 x 8.

En términos de potencia de transmisión, por lo general se hacen uso de niveles similares a los demás entornos. En el caso de los APs se proponen 24dBm [46] [59], y para las STAs 15 dBm [59] o 21 dBm [46]. Por otro lado, en una investigación específica sobre tecnología MIMO en oficinas se propone como potencia de transmisión de grupo –para 4 APs– 10 dBm [57].

Para terminar con las características físicas del entorno empresarial, el uso de OFDM no se menciona de forma explícita en limitadas ocasiones [46] y [58], recomendándose intervalos de guarda bajos (800 ns). Además, en cuanto a modelos de propagación, el TGax [46] establece pérdidas logarítmicas en función de la distancia en base a tres valores (ThreeLogDistance) junto con atenuaciones prevenientes de la posibilidad de que la señal atravesase paredes –también se contempla fading y shadowing. En estudios específicos sobre pérdidas de propagación [49] se mencionan diferentes modelos utilizados para este tipo de entornos, como la propuesta de la ITU para entornos de interior o modelos para escenarios con múltiples paredes. En primer lugar, se definen las pérdidas en la recepción de la señal desde otro piso en 21 dBm, y 36 dBm en caso de atravesar dos pisos. Por otro lado, se definen 2.67 dBm de pérdidas para atravesar obstáculos de madera, 2.73 dBm al atravesar paredes de cemento, 4.5 dBm para obstáculos de cristal, y 6 dBm para la existencia de pilares de 0.6m x 0.6m. Asimismo, otras propuestas para oficinas utilizan un modelo de pérdidas de interior tanto con LOS como NLOS, y teniendo en cuenta condiciones de fading [57], o el modelo InH 3GPP para interiores de oficina [59].

En cuanto a las características a nivel de enlace, el TGax [46] define de forma similar al escenario residencial los parámetros de simulación para IEEE 802.11ax en entornos empresariales. Se utiliza EDCA con valores estándar, y en este caso se establece una gestión de canales con asignación predefinida tanto para 5 GHz como para 2,4 GHz. En el primero se establecen 4 canales de 80 MHz a definir con una configuración específica en cada BSS. En el segundo caso, se establecen 3 canales de 20 MHz de forma similar. También se define agregación a nivel de A-MPDUs –con el tamaño de agregación máximo posible y un tamaño de ventana correspondiente a BA– pero no A-MSDUs, y se establece, como en el escenario anterior, un límite máximo de reintentos de envío establecido en 10. Por último, no se hace uso de RTS/CTS, y se permite asumir gestión de Aps mediante una única entidad.

Del mismo modo, en la guía de despliegue de Wi-Fi de última generación [59] también se destaca la inclusión de agregación de MPDUs.

En el **ámbito interior de alta densidad** sucede lo mismo que para los dos casos anteriores, pero se intuye un mayor interés en su estudio para los protocolos más

actuales. Entre la documentación analizada, la versión del estándar predominante es IEEE 802.11ax [111] [60] [62] [46], aunque también se han encontrado estudios de esta clase de entornos para versiones anteriores del protocolo, como IEEE 802.11ac [60] [62], IEEE 802.11n [61], IEEE 802.11a/b/g [63] [61] o IEEE 802.11b [51].

En términos de bandas frecuenciales, se propone principalmente para su análisis la banda de 5 GHz [62] [60] [46] en las últimas versiones, aunque en otros casos también se hace mención al poco aprovechamiento de dicha configuración en caso de disponer tanto de la banda de 5 GHz como de la de 2,4 GHz [113]. Para 5 GHz se proponen anchos de canal desde 20 MHz [63] [60] hasta 80 MHz [46] [60], mientras que en la de 2,4 GHz se hace uso de 20 MHz mayoritariamente [46] [51] [61].

Con relación a las características MIMO y configuraciones de antenas, se plantea la utilización de Wi-Fi 6 –IEEE 802.11ax– con APs de 2 x 2 o 4 x 4 antenas, y STAs con 1 x 1 o 2 x 2 antenas [46]. También se proponen configuraciones tanto para Wi-Fi 5 –IEEE 802.11ac– como Wi-Fi 6 con Aps de 3 x 3 y STAs de 2 x 2 antenas [60], y Wi-Fi 4 con Aps de 2 x 2 [61].

Por otra parte, la potencia de transmisión configurada tanto en APs como en STAs es similar a la de los escenarios anteriormente analizados. En cuanto a los APs, se definen potencias de entre 4 y 23 dBm [60], dando valores concretos en algunos casos, como 10 dBm [63], 17 dBm [46] o 20 dBm [51]. Por otro lado, la potencia de transmisión es especificada por el TGax como 15 dBm [46].

Respecto al uso de OFDM, su utilización se menciona específicamente en el caso de IEEE 802.11ax [46] [62], donde se especifica su utilización con intervalos de guarda bajos, en concreto, de 800ns.

Por último, en cuanto a modelos de propagación, el TGax establece [46] pérdidas logarítmicas en función de la distancia partiendo de tres valores de referencia de distancia (ThreeLogDistance) –también se definen modelos para fading y shadowing– como modelo. En un estudio donde se emplea un modelo realista mediante ray tracing [63] se especifican pérdidas por obstáculos de cemento –se contempla la existencia de paredes y techos, entre otros– de 26dB. Por otra parte, también se valida el modelo propuesto por el grupo IEEE 802.11ax para este tipo de escenarios [60], denominado TMB path loss model.

Con relación a las especificaciones MAC, el TGax [46] define como parámetros de simulación para IEEE 802.11ax sobre entornos de interior de alta densidad la utilización de EDCA con los valores específicos por defecto para cada clase de tráfico. La gestión de canales se realiza mediante la asignación de todos ellos sobre una banda u otra. En caso de establecerse sobre la banda de 2,4 GHz, se emplean canales de 20 MHz con reutilización de frecuencias 3. Por otra parte, si se hace uso de la banda de 5 GHz, se asignan canales de 80 MHz con reutilización de frecuencias 1. En cuanto a agregación, se define a nivel de A-MPDUs –con el tamaño máximo de agregación y un tamaño de ventana correspondiente a BA– pero no se utilizan A-MSDUs. Además, se establece un límite máximo de reintentos de envío de 10, y no se hace uso de RTS/CTS. Se permite asumir que la gestión de los APs sea centralizada.

Entre la documentación analizada, cabe destacar la consideración del uso de RTS/CTS en simulaciones de estadios con despliegues IEEE 802.11a/g de alta densidad [63], y el uso de agregación sobre MPDUs en los casos de uso prácticos de IEEE 802.11n [61].

Para terminar con el análisis de las características físicas y configuración de la transmisión, se estudia el **ámbito exterior de alta densidad**.

Al igual que en el entorno interior de alta densidad, se percibe un mayor interés en su análisis con las versiones más actuales del estándar. En concreto, se vuelve objeto de estudio para el protocolo IEEE 802.11ax [46] [62] promoviendo alta eficiencia en entornos con gran concentración de dispositivos y usuarios. Además, se ha recogido la información sobre despliegues urbanos para IEEE 802.11 b/g/n [64] y IEEE 802.11 b/g [112], y campus universitarios con IEEE 802.11ac [53] y IEEE 802.11a/g/n [54].

En cuanto a las bandas utilizadas, se contempla el uso tanto de la banda de 2,4 GHz como la de 5 GHz en la documentación, pero se da preferencia a la de 5 GHz en las últimas versiones del estándar [62] [46]. Asimismo, se menciona que la mayoría del tráfico recae en la banda de 2,4 GHz [64].

Como sucedía en los casos anteriores, para 2,4 GHz se propone de forma general la utilización de canales de 20 MHz [46] [54] [53] [64] y en la banda de 5 GHz se contemplan desde los 20 MHz [46] [53] [62] [54] hasta los 80 MHz [53] [62] o 160 MHz [46].

En relación con el uso de MIMO y las configuraciones de antenas utilizadas, en el caso de IEEE 802.11ax se proponen configuraciones de 2×2 o 4×4 para los APs y 1×1 o 2×2 para las STAs [46]. También se contempla el uso de MU-MIMO con hasta 4 u 8 streams espaciales [62], pero se menciona la limitación de que las STA habitualmente desplegadas tan solo admiten 1 o 2 flujos. En el caso de IEEE 802.11ac [53] se hace uso de APs con configuraciones 4×4 y 3×3 en entornos universitarios, y en IEEE 802.11n se proponen configuraciones de 3×2 [54] y 2×2 [64] para los APs.

Respecto a la potencia de transmisión, los valores utilizados son similares a los demás entornos.

Los APs, en general, se definen con rangos entre -1 dBm [54] y 30 dBm [46], dándose casos específicos acotados entre 2 dBm y 23 dBm [53], y valores típicos de 20 dBm [64]. En algunos casos se menciona la utilización de diferentes potencias para bandas distintas, pero con valores muy próximos [53] dentro de los límites generales. Para las STAs se definen con 15 dBm [46] de potencia de transmisión.

Para finalizar con el entorno exterior de alta densidad, el uso de OFDM, al igual que en los demás entornos, tan sólo se menciona de forma explícita en contadas ocasiones [51] [46]. En general se recomiendan intervalos de guarda bajos (800ns) [54] [53], pero también se hace alusión [46] a intervalos mayores (3200 ns). Por último, el TGax propone el uso de los modelos de la 3GPP para Umi –escenarios urbanos micro-celulares– o Uma –escenarios urbanos macro-celulares–.

Con relación a las especificaciones MAC, se definen como parámetros de simulación para IEEE 802.11ax en entornos exteriores de alta densidad [46] la utilización de EDCA

con los valores específicos por defecto para cada clase de tráfico. En cuanto a la gestión de canales, se utiliza asignación predefinida con reutilización de frecuencias 1, haciendo uso de canales de 80 MHz para la banda de 5 GHz con el mismo canal primario asignado, y canales de 20 MHz sobre la banda de 2,4 GHz. También se define agregación a nivel de A-MPDUs –con un tamaño de agregación de 64 MPDUs y un tamaño de ventana correspondiente a BA– por lo que no se utilizan A-MSDUs, y se establece un límite máximo de reintentos de envío establecido en 10. Por último, no se hace uso de RTS/CTS, y se presupone gestión independiente de los APs.

14.2.1.1. Modelado de tráfico

Como se introduce en el apartado Caracterización del Tráfico de Red, para llevar a cabo la definición de los modelos de tráfico a implementar en el simulador de red se deben estudiar tanto las aplicaciones más habituales y sus parámetros específicos como las características relativas al tipo de tráfico.

De acuerdo con los criterios para desarrollo de estándares de la IEEE 802.11 [114], cada vez más gente depende de Wi-Fi para suplir la necesidad de conectividad en tareas como el teletrabajo, entretenimiento, búsqueda de información en la web, etc. Existe una gran variedad de aproximaciones para la caracterización de estos tipos de tráfico de forma realista, ya sea en base a las características estadísticas generales del tráfico IP o en base a la caracterización de servicios de red concretos.

A continuación, se recogen las aplicaciones a implementar en cada escenario y las características estadísticas que permiten el modelado de cada tipo de aplicación. De este modo, la siguiente información conforma la ampliación en detalle de los tipos de tráfico recogidos en el apartado Tipos de Tráfico por Entorno y define el modelado estadístico de los tipos de tráfico especificados en Características Principales de los Tipos de Tráfico.

14.2.1.1.1. Caracterización de los Tipos de Tráfico por Entorno

En el apartado Tipos de Tráfico por Entorno se adelanta la existencia de múltiples estudios que hacen distinción de tipos de tráfico en base a aplicaciones toman como base para la construcción diferenciada en entornos Wi-Fi la propuesta del grupo de estudio de IEEE 802.11ax para diferentes escenarios [46].

Por lo tanto, se presenta a continuación la información relativa al **tráfico de red en base a aplicaciones comunes para cada entorno de interés**.

En el **entorno residencial**, el grupo TGax [46] propone las transmisiones caracterizadas en la Tabla 11. Como se puede observar, se diferencian flujos de tráfico en cada sentido de la transmisión para cada tipo de aplicación. Además, los servicios considerados desde o hacia cada STA deben ser asignados de forma aleatoria sobre el número total de dispositivos de dicho rol. Por ejemplo, en este entorno TGax propone considerar un total de 10 STAs, por lo que para la aplicación D1 relativa a Buffered Video Streaming se asignarán dos STA origen y dos STA destino.

ID	Origen / Destino	Modelo de tráfico	ID de tráfico	Dirección	STAs origen / destino	AC
D1	AP/STA	Buffered Video Streaming	BV6	Asimétrico Bidireccional	2 / 2	VI
D2	AP/STA	Buffered Video Streaming	BV3	Asimétrico Bidireccional	4 / 4	VI
D3	AP/STA	FTP	FTP	Asimétrico Bidireccional	2 / 2	BE
D4	AP/STA	HTTP	HTTP	Asimétrico Bidireccional	4 / 4	BE
D5	AP/STA	Gaming	GMG	Asimétrico Bidireccional	3 / 3	VI
D6	AP/STA	VoIP	VOIP	Simétrico Bidireccional	2 / 2	VO
D7	AP/STA	MGMT: Beacon	Beacon de 220 octetos a 1 Mbps (2,4 GHz) o 6 Mbps (5 GHz) cada 100 TUs	Unidireccional	1 / 0	VI
U1	STA/AP	Buffered Video Streaming	BV3	Asimétrico Bidireccional	1 / 1	VI
U2	STA/AP	FTP	FTP	Asimétrico Bidireccional	2 / 2	BE
U3	STA/AP	Gaming	GMG	Asimétrico Bidireccional	3 / 3	VI
U4	STA/AP	MGMT: Probe Req	TBD	Unidireccional	Todos las STAs sin asociar / 0	VI

Tabla 95: Modelo de tráfico para entorno residencial definido por el TGax. [46]

Otro ejemplo de tráfico del entorno residencial es el tráfico IoT [68]. En los estudios sobre este tipo de tráfico se modelan las comunicaciones entre dispositivos del hogar con acceso a Internet mediante la caracterización de tráfico ascendente desde los dispositivos al AP y tratándose de tráfico asimétrico, con un límite superior de datos a transmitir previamente establecido, y donde se supone un retardo constante en transmisión. Se define el tráfico IoT como tráfico en tiempo real sin pérdidas.

En estudios sobre casos reales de entornos domésticos [69] también se hace referencia a diferentes modelos de tráfico. En las muestras obtenidas de un ISP residencial sobre 3470 Aps se destacan las aplicaciones web –cuya posterior simulación se propone como el acceso secuencial a 10 páginas distintas en 10 ocasiones–, VoD –cuya simulación se realiza mediante reproducción de vídeo haciendo uso de DASH durante 2 minutos– y

audio y vídeo RTC –para lo que se usa WebRTC con 20 muestras de 8 segundos recomendadas por la ITU-T y vídeos 640x480 a 30fps durante un periodo de 10s–.

Por último, en otro estudio en el que se recogen datos de una gran red de miles de hogares se extraen muestras de hasta 20 protocolos distintos con sus aplicaciones asociadas. En concreto, se destacan sesiones de tráfico reproducibles según una distribución lognormal tanto en términos de duración como de volumen.

En el **entorno de oficinas**, el grupo TGax [46] realiza una propuesta similar al residencial, pero acotando los tipos de tráfico y aplicando un ajuste proporcional al número de nodos–STAs y Aps– contemplados. También se hace alusión a los diferentes tipos de tráfico sobre este tipo de escenarios en múltiples estudios de casos reales. Entre ellos, se contemplan casos de uso como transmisión de vídeo para teleconferencias, servicios cloud y streaming de audio y vídeo [27].

Además, cabe destacar el crecimiento drástico de las aplicaciones de tráfico de vídeo –incluyendo el vídeo bajo demanda, las teleconferencias y VR/AR– en los últimos años [70] sobre servicios como Skype, Hangouts o FaceTime en entornos empresariales.

ID	Origen / Destino	Modelo de tráfico	ID de tráfico	Dirección	STAs origen / destino	AC
D1	AP/STA	Buffered Video Streaming	BV6	Asimétrico Bidireccional	2 / 2	VI
D2	AP/STA	Buffered Video Streaming	BV3	Asimétrico Bidireccional	6/6	VI
D3	AP/STA	VDI	VDI	Asimétrico Bidireccional	48 / 48	VI
D4	AP/STA	VoIP	VOIP	Simétrico Bidireccional	10 / 10	VO
D5	AP/STA	MGMT: Beacon	Beacon de 280 octetos a 1 Mbps (2,4 GHz) o 6 Mbps (5 GHz) cada 100 Tus	Unidireccional	1 / 0	VI
U1	STA/AP	MGMT: Probe Req	TBD	Unidireccional	Todos las STAs sin asociar / 0	VI

Tabla 96: Modelo de tráfico para entorno de oficinas definido por el TGax. [46]

Por otro lado, en el **entorno de interior de alta densidad**, el grupo TGax [46] propone un set de configuraciones de tráfico similar al entorno residencial, pero de mayor dimensión en términos de dispositivos.

De acuerdo con múltiples casos de uso [27] sobre entornos inalámbricos, serían aplicables a este tipo de escenarios flujos derivados de tráfico de streaming y AR/VR en

caso de cubrir eventos, o acceso web o transmisión de vídeo para entretenimiento en el caso de espacios de interior que se puedan asemejar a lugares turísticos.

Del mismo modo, también se propone para la simulación de escenarios de alta densidad tanto de interior como de exterior [71] la utilización de tráfico de tres tipos de uso común. Los servicios simulados son la transferencia de ficheros, para lo que se utiliza Iperf3, streaming de vídeo sobre HTTP haciendo uso de DASH, y acceso a páginas web mediante la selección de 37 sitios web recurrentes, diferenciando entre corta y larga duración en función de la velocidad de carga de estas.

Otros estudios del rendimiento de red sobre casos de uso de alta densidad [66], por otra parte, contemplan el modelado de VoD (Video over Demand) sobre topologías sencillas debido a la relevancia de dicho tipo de tráfico para estos entornos.

ID	Origen / Destino	Modelo de tráfico	ID de tráfico	Dirección	STAs origen / destino	AC
D1	AP/STA	Buffered Video Streaming	BV6	Asimétrico Bidireccional	12 / 12	VI
D2	AP/STA	Buffered Video Streaming	BV3	Asimétrico Bidireccional	8 / 8	VI
D3	AP/STA	FTP	FTP	Asimétrico Bidireccional	4 / 4	BE
D4	AP/STA	HTTP	HTTP	Asimétrico Bidireccional	12 / 12	BE
D5	AP/STA	Gaming	GMG	Asimétrico Bidireccional	16 / 16	VI
D6	AP/STA	VoIP	VOIP	Simétrico Bidireccional	12 / 12	VO
D7	AP/STA	MGMT: Beacon	Beacon de 280 octetos a 1 Mbps (2,4 GHz) o 6 Mbps (5 GHz) cada 100 Tus	Unidireccional	1 / 0	VI
U1	STA/AP	Buffered Video Streaming	BV3	Asimétrico Bidireccional	4 / 4	VI
U2	STA/AP	FTP	FTP	Asimétrico Bidireccional	4 / 4	BE
U3	STA/AP	Gaming	GMG	Asimétrico Bidireccional	16 / 16	VI
U4	STA/AP	MGMT: Probe Req	-	Unidireccional	Todos las STAs sin asociar / 0	VI

Tabla 97: Modelo de tráfico para entorno interior de alta densidad definido por el TGax. [46]

Para finalizar la caracterización de las aplicaciones comunes a cada escenario, en relación con el **entorno exterior de alta densidad**, el grupo TGax [46] propone una serie de aplicaciones igual al entorno anterior, pero con un ligero aumento en los dispositivos. Además de los anteriores, se mencionan aplicaciones como AR/VR en el ámbito de la educación [27], como también ocurría en los entornos de interior de alta densidad. En este caso, se aplica a los espacios universitarios abiertos. Asimismo, al ser un entorno susceptible de cubrir eventos, también se correspondería con la aplicación de transmisión de vídeo [27].

ID	Origen / Destino	Modelo de tráfico	ID de tráfico	Dirección	STAs origen / destino	AC
D1	AP/STA	Buffered Video Streaming	BV6	Asimétrico Bidireccional	5 / 5	VI
D2	AP/STA	Buffered Video Streaming	BV3	Asimétrico Bidireccional	20 / 20	VI
D3	AP/STA	Multicast Video Streaming	MC2	Unidireccional	1 / 12	VI
D4	AP/STA	FTP	FTP	Asimétrico Bidireccional	4 / 4	BE
D4	AP/STA	HTTP	HTTP	Asimétrico Bidireccional	15 / 15	BE
D5	AP/STA	Gaming	GMG	Asimétrico Bidireccional	25 / 25	VI
D6	AP/STA	VoIP	VOIP	Simétrico Bidireccional	20 / 20	VO
D7	AP/STA	MGMT: Beacon	Beacon de 280 octetos a 1 Mbps (2,4 GHz) o 6 Mbps (5 GHz) cada 100 Tus	Unidireccional	1 / 0	VI
U1	STA/AP	Buffered Video Streaming	BV3	Asimétrico Bidireccional	10 / 10	VI
U2	STA/AP	FTP	FTP	Asimétrico Bidireccional	4 / 4	BE
U3	STA/AP	Gaming	GMG	Asimétrico Bidireccional	25 / 25	VI
U4	STA/AP	MGMT: Probe Req	-	Unidireccional	Todas las STAs sin asociar / 0	VI

Tabla 98: Modelo de tráfico para entorno exterior de alta densidad definido por el TGax. [46]

Al igual que para los entornos de interior, en la caracterización de tráfico sobre entornos de alta densidad tanto de exterior como interior [71] se propone la utilización de transferencia de ficheros, streaming de vídeo sobre HTTP y acceso a páginas web recurrentes.

14.2.1.1.2. Modelado de los tipos de tráfico

El presente apartado se centra en los **parámetros específicos** y los **modelos estadísticos** representativos de los tipos de tráfico del apartado Características Principales de los Tipos de Tráfico, recogidos en la Tabla 15.

Para comenzar, el **tráfico de gestión** se modela de forma distinta para clientes asociados a algún AP y para clientes no asociados a ningún AP.

Para clientes que no están asociados a ningún AP, se definen periodos de sondeo para el envío de probe request de 12 segundos para un 50% de las STAs, y 12,5 para el resto. En caso de no hallarse asociado un cliente tras 5 intentos sobre todos los canales, se repite el mismo procedimiento duplicando dicho periodo, hasta alcanzar un máximo de 400 segundos.

Se entiende como posible canal cualquiera de los soportados sobre la banda de trabajo. Los SSID de los probe request se obtienen mediante mensajes de tipo broadcast probe request a SSIDs comodín (wildcard SSID) –utilizado para el caso en el que la STA se puede conectar a cualquier BSS– más un rango entre 0 y 3 SSIDs establecidos. El tamaño de frame para probe request es de 80 o 160 bytes.

Para clientes asociados, se define un periodo de sondeo de 60 segundos, canales iguales a los que cada cliente está asociado siempre que el RSSI de los beacons del AP asociado sean superiores a una potencia mínima, en cuyo caso se probaría cada canal soportado en la banda, y tamaños de frame para las probe request de 80 o 160 B.

El SSID de las probe request se establece sobre el SSID del AP asociado siguiendo el mismo criterio de potencia que para la selección de canal, y solicitudes broadcast a SSIDs comodines en caso contrario.

Dentro de las múltiples casuísticas del **tráfico de vídeo**, se define el modelo de tráfico **Wireless Display** como una aplicación de vídeo en la que los fragmentos o slices de vídeo se generan en un intervalo fijo, como puede ser, por ejemplo, el envío de vídeo a un formato de 1080p en intervalos de 1/4080 segundos.

ID	Formato de vídeo	Tiempo entre llegadas (IAT)	MaxSliceSize	p	b
WD1	1080p60	1/4080 s	92.160 KB	300 Mbps	300 Mbps
WD2	1080p60 3D	1/4080 s	92.160 KB	450 Mbps	300 Mbps
WD3	4K UHD 60fps	1/8100 s	184.320 KB	600 Mbps	300 Mbps
WD4	8K UHD 60fps	1/16200 s	368.640 KB	1200 Mbps	300 Mbps

Tabla 99: Parámetros para el modelado de tráfico Wireless Display. [72]

Los parámetros para la generación de paquetes de vídeo se recogen en la Tabla 99, asumiendo una distribución normal con $\mu = 9.19$ Kbytes y $\sigma = 1.350$ Kbytes.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Ecuación 16: Función de densidad de probabilidad de una distribución Normal.

Con relación al algoritmo para la generación de cada paquete de vídeo, se define como entrada la tasa de bits (p) en Mbps que se introduce dividiéndolo por la tasa de bits fija (b) de 300 Mbps, y como salida el tamaño de la porción de vídeo (L) en Kbytes, de forma que a cada tiempo entre llegadas (Inter Arrival Time, IAT) se genera un tamaño de porción L con distribución Normal($(\mu^*(p/b), \sigma^*(p/b))$) siguiendo la Ecuación 16. Asimismo, se establece que para los casos en los que se cumpla $L > \text{MaxSliceSize}$, se defina su valor al tamaño máximo establecido.

Otra clase de tráfico de vídeo habitual es el tráfico **streaming con buffer**, cuya implementación se realiza mediante la generación de tamaños de vídeo en bytes de acuerdo con una distribución Weibull con la función de distribución de probabilidad de la Ecuación 17.

$$f(x) = \begin{cases} \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-\left(\frac{x}{\lambda}\right)^k}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Ecuación 17: Función de densidad de probabilidad de una distribución Weibull. [72]

Dependiendo de la subclase de tasa de bits de vídeo se definen diferentes parámetros λ y k , de acuerdo con la Tabla 100.

ID	Tasa de bits del vídeo	λ	k
BV6	15,6 Mbps	54210	0.8099
BV5	10 Mbps	34750	0.8099
BV4	8 Mbps	27800	0.8099
BV3	6 Mbps	20850	0.8099
BV2	4 Mbps	13900	0.8099
BV1	2 Mbps	6950	0.8099

Tabla 100: Parámetros λ y k para tasas de bits de vídeo. [72]

A nivel de transporte se definen segmentos TCP de 1500 bytes y se fragmentan los paquetes de vídeo en segmentos TCP, a los que se añade latencia—cuando llegan al AP—de acuerdo con una distribución Gamma cuya función de distribución de probabilidad se corresponde con la Ecuación 18.

$$f(x; k, \theta) = \frac{x^{k-1} e^{-\frac{x}{\theta}}}{\theta^k \Gamma(k)} \text{ para } x > 0 \text{ y } k, \theta > 0$$

Ecuación 18: Función de densidad de probabilidad de una distribución Gamma. [72]

Se toma como parámetros $k = 0,2463$ y $\theta = 60,227$. De esta forma, la latencia media resultante es de 14,834 ms. Para simular latencias superiores o inferiores, se debe escalar theta linealmente de forma que la media de la distribución Gamma se mantenga como el producto de K y θ .

Los mensajes ACK de TCP recíprocos desde el receptor se definen con tamaño de 40 bytes, y tiempo entre llegadas de 1 ms respecto a la recepción de vídeo en destino.

Una alternativa al modelado de este tipo de tráfico se recoge del análisis sobre tres topologías en uno de los estudios sobre tráfico real de la definición de escenarios [66], en el que se consumen tres vídeos distintos de acuerdo con la Tabla 101.

Característica	Vídeo 1	Vídeo 2	Vídeo 3
Duración	94 s	109 s	28 s
Tamaño	62,4 MB	58,9 MB	9,7 MB
Tasa de bits	5,2 Mbps	4,3 Mbps	2,8 Mbps
Complejidad	Movimiento apreciable durante toda la duración del vídeo	Movimiento medio	Baja movilidad
Codec	MPEG2	MPEG2	MPEG2
Formato de container	TS	TS	TS
Resolución	1920x1080	1920x1080	1920x1080
FPS	29,97	29,97	29,97
Relación de aspecto	16:9	16:9	16:9

Tabla 101: Características de los vídeos utilizados para el modelado de tráfico VoD. [66]

El análisis realizado sobre el tráfico se traduce en el comportamiento matemático descrito en la Tabla 102, donde se establecen modelos estadísticos diferentes en base a los tipos de trama de vídeo y con parámetros específicos para cada vídeo distinto.

Otro modelo de tráfico de vídeo de interés es el de **videoconferencias**, para el que se presupone la existencia de jitter en la transmisión desde el AP, lo que hace que se pueda modelar del mismo modo que el tráfico de vídeo anterior siguiendo la especificación de la Tabla 100, haciendo uso del modelo de tráfico específico de la clase BV1. En sentido contrario, en cambio, se define el tráfico de forma periódica a nivel de aplicación desde la STA.

Con relación a los patrones de eventos a nivel de aplicación –cuándo comienza la ejecución, cuánto dura el uso de una aplicación, etc.– se considera que para diferentes escenarios se pueden emplear modelos como Poisson e Hiper-Exponencial.

El primero puede utilizarse para casos donde hay muchos usuarios, generando cada uno poco tráfico y solicitando acceso a la red de forma aleatoria. El segundo, por el contrario, se puede usar para casos pico en los que los usuarios hacen solicitudes de acceso a red que sobresalen notoriamente de la media.

		Vídeo 1		Vídeo 2		Vídeo 3	
Tipo de trama	Función	T(s)	S(bytes)	T(s)	S(bytes)	T(s)	S(bytes)
I	PDF	Lognormal	Weibull	Logistic	Weibull	-	-
	μ	0,4026	57177,5	0,3993	61379,2	-	-
	Σ	0,0352	6592,3	0,0289	2910,3	-	-
	KS	0,03	0,04	0,06	0,06	-	-
P	PDF	Weibull	Lognormal	Lognormal	Weibull	Exponencial	Weibull
	μ	0,1358	38446,8	0,1396	3447,4	0,1375	26923,8
	Σ	0,0688	9491,8	0,0972	12343	0,1261	14113,4
	KS	0,09	0,09	0,1	0,07	0,1	0,09
B	PDF	Gamma	Weibull	Lognormal	Lognormal	Lognormal	Gamma
	μ	0,0508	12270,8	0,0514	10706,5	0,05	59314
	Σ	0,0338	7051,2	0,0559	8272,7	0,0911	3895,4
	KS	0,01	0,1	0,07	0,09	0,05	0,1
Audio	PDF	Gamma	Logistic	Gamma	Logistic	Weibull	Normal
	μ	0,1484	2743,1	0,1468	2738,3	0,1457	2740
	Σ	0,0466	139,1	0,0614	137,6	0,0933	118,9
	KS	0,04	0,07	0,02	0,07	0,05	0,08

Tabla 102: Comportamiento estadístico del tráfico de vídeo en [66].

Además de los anteriores, el tráfico de vídeo **streaming multicast** entre el AP y las estaciones también sufre jitter a lo largo de la red, y se puede modelar del mismo modo que el streaming de vídeo definido anteriormente, pero con los parámetros especificados en la Tabla 103.

ID	Tasa de bits del vídeo	λ	K
MC1	3 Mbps	10425	0.8099
MC2	6 Mbps	20850	0.8099

Tabla 103: Parámetros para tráfico de Streaming Multicast

Por otro lado, la Tabla 104 resume los parámetros específicos a los procedimientos principales para la caracterización del tráfico **VDI (Virtual Desktop Infrastructure)**.

Componente	Distribución		Parámetros	
	DL	UL	DL	DL
Llegada inicial de paquete (ms)	Uniforme	Uniforme	a = 0, b = 20	a = 0, b = 20
Tiempo de llegada de paquete (ms)	Exponencial $\lambda = 1/medida$	Exponencial $\lambda = 1/medida$	media = 60,2269	media = 48,2870
Tamaño de paquete (B)	Normal Bimodal	Normal	$\mu_1 = 41,0$ $\sigma_1 = 3,2$ $\mu_2 = 1478,3$ $\sigma_2 = 11,6$	$\mu = 50,598$ $\sigma = 5,0753$

Tabla 104: Parámetros del tráfico VDI.

La llegada de paquetes de este tipo a nivel MAC se modela según el diagrama de la Figura 72.



Figura 72: Diagrama de la llegada de paquetes VDI. [72]

Dicho proceso puede ser descrito según una distribución exponencial, cuya función de densidad de probabilidad se recoge en la Ecuación 19.

$$f(x; \lambda) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Ecuación 19: Función de densidad de probabilidad para una distribución Exponencial. [72]

El tamaño de los frames MSDU en bytes para transmisiones en ambos sentidos es distinta. En el caso de UL se corresponde con una distribución normal, y para DL con una distribución bimodal normal (Ecuación 20).

$$f(x) = \frac{53,7}{76,1 \cdot \sigma_1 \sqrt{2\pi}} \exp\left(-\frac{(x - \mu_1)^2}{2\sigma_1^2}\right) + \frac{22,4}{76,1 \cdot \sigma_2 \sqrt{2\pi}} \exp\left(-\frac{(x - \mu_2)^2}{2\sigma_2^2}\right)$$

Ecuación 20: Función de densidad de probabilidad para la distribución Normal Bimodal del tamaño de paquete de tráfico VDI. [72]

El tráfico relativo a la **navegación web HTTP** puede modelarse según los componentes de la Tabla 105. El número de objetos embebidos en las páginas web sigue la distribución de la Ecuación 21.

$$f(x) = \frac{\alpha k^\alpha}{x^{\alpha+1}}, k \leq x \leq m$$

$$f(x) = \left(\frac{k}{m}\right)^\alpha, x = m$$

Ecuación 21: Función de densidad de probabilidad para la distribución Pareto Truncada. [72]

Componente	Distribución	Parámetros
Tamaño de fichero (S)	Lognormal Truncado	media = 10710 B SD = 25032 B mín = 100 B máx = 2 B (antes del truncado) $\sigma = 1.37$ $\mu = 8.37$ Si $x > \text{máx}$ o $x < \text{mín}$, se descarta el resultado y se genera un nuevo valor para x
Tamaño del objeto embebido (SE)	Lognormal Truncado	media = 7758 B SD = 126168 B mín = 50 bytes máx = 2 MB (antes del truncado) $\sigma = 2.36$ $\mu = 6.17$ Si $x > \text{máx}$ o $x < \text{mín}$, se descarta el resultado y se genera un nuevo valor para x
Número de objetos embebidos por página (Nd)	Pareto Truncado	media = 5.64 máx = 53 (antes del truncado) $\alpha = 1.1$ $k = 2$ $m = 55$ Restar k del valor aleatorio generado para obtener Nd Si $x > \text{máx}$, se descarta el resultado y se genera un nuevo valor para x
Tiempo de lectura (D_{pc})	Exponencial	media = 30 s $\lambda = 0.033$
Tiempo de análisis (parsing) (T_p)	Exponencial	media = 0,13 s $\lambda = 7.69$

Tabla 105: Parámetros de tráfico HTTP.

El procedimiento parte de que el cliente HTTP envía una solicitud HTTP con tamaño 350 bytes para iniciar el evento en el servidor.

Los tamaños de MTU siguen la misma distribución que en el tipo de tráfico anterior, y los ACK recíprocos a los mensajes se generan con tamaño 40 bytes y con un tiempo entre llegadas equivalente a 1 ms de delay desde la recepción del frame HTTP en el cliente.

La generación de **tráfico FTP (File Transfer Protocol)** consiste, en primer lugar, en crear un fichero utilizando las estadísticas recogidas en la Tabla 106.

Esta tabla provee el modelado de los parámetros del tráfico FTP e incluye el tráfico tanto en DL como en UL, con tamaños de MTU de 1500 bytes o 576 bytes respectivamente.

Componente	Distribución	Parámetros
Tamaño de fichero (S)	Lognormal Truncado	media = 2 MB SD = 0,722 MB máx = 5 MB $\sigma = 0.35$ $\mu = 14.45$ Si $x > \text{máx}$ o $x < \text{mín}$, se descarta el resultado y se genera un nuevo valor para x
Tiempo de lectura (D)	Exponencial	media = 180 s $\lambda = 0.006$

Tabla 106: Parámetros de tráfico FTP

Después de completar la transferencia, se usa una nueva conexión TCP con $W=1$, esperando eventualmente un tiempo de lectura hasta la siguiente transferencia.

En base a los resultados de la distribución (Ecuación 22) de tamaños de paquete, el 76% de los ficheros se transmiten utilizando una MTU de 1500 bytes, mientras que el 24% con una MTU de 576 bytes. Los tamaños incluyen 40 bytes de cabecera IP.

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma x} \exp\left[-\frac{(\ln x - \mu)^2}{2\sigma^2}\right], x \geq 0$$

Ecuación 22: Función de densidad de probabilidad para la distribución Lognormal Truncada. [72]

De forma equivalente a HTTP, el tráfico relativo a los ACKs recíprocos a las transferencias se genera como transmisiones de 40 bytes con tiempo entre llegadas igual a un delay de 1 ms entre la recepción del frame FTP en el cliente.

Otra aplicación de amplia utilización es el servicio **VoIP**, cuyos parámetros a considerar se recogen en la Tabla 107.

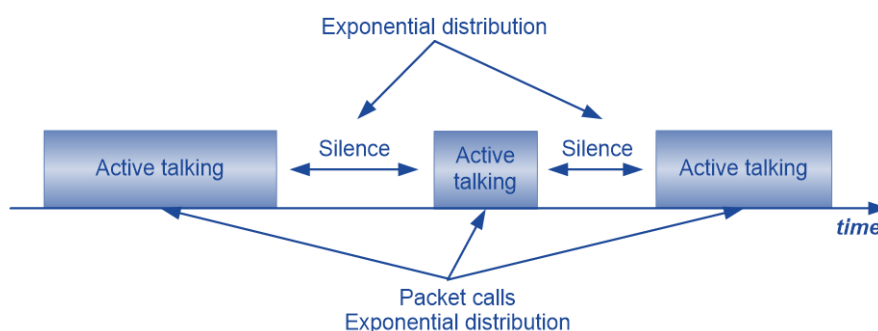


Figura 73: Diagrama de la llegada de paquetes VoIP. [72]

La evaluación del estado de actividad o silencio (Figura 15) es realizada a una tasa de frames $R=1/T$, donde T es la duración de frame del codificador cuyo valor típico es 20 ms para estado de habla, y 160 ms para estado de silencio, respectivamente.

Componente	Parámetros
Tasa de fuente	12.2 Kbps
Tamaño de carga (payload) de paquetes en estado activo	33 byte
Tamaño de carga (payload) de paquetes en estado silencio	7 byte
Cabeceras de protocolo	Ipv4: 3 byte Ipv6: 5 byte
Intervalo del codificador de voz	20+τ ms
Intervalo del codificador de ruido	160+τ ms
Duración de los estados Activo / Silencio	Exponencial Media = 1.25 s
Jitter DL	Laplaciano β=5.11 ms
Jitter UL	0
Factor de actividad de voz	$\frac{b}{a+b}$ a = b = 0.016

Tabla 107: Parámetros del tráfico VoIP. [72]

Los paquetes VoIP son generados en intervalos de tiempo $iT + \tau$, donde τ es el jitter (Ecuación 23) entre llegadas de paquetes (**¡Error! No se encuentra el origen de la referencia.**), e i el índice de frames del codificador. Durante el estado activo, se generan paquetes de tamaño fijo, y el modelo se actualiza en intervalos regulares.

$$f(x) = \frac{1}{2\beta} e^{-\frac{|\tau|}{\beta}}, -80ms \leq \tau \leq 80ms$$

Ecuación 23: Función de densidad de probabilidad para la distribución Laplaciana del jitter DL en tráfico VoIP. [72]

La tasa de la fuente de voz asume 12,2 kbps con un factor de actividad de voz del 50%. El tamaño del payload de estado de habla y silencio son 33 bytes y 7 bytes respectivamente. Las cabeceras comprimidas incluyendo el checksum UDP ocupan 3 bytes para Ipv4 y 5 bytes para Ipv6. El tamaño total de MSDU para periodos de habla es de 36 bytes para Ipv4 y 38 bytes para Ipv6, y en estado de silencio de 10 bytes para Ipv4 y de 12 bytes para Ipv6.

Por último, el modelado del **tráfico gaming** se puede realizar mediante una distribución Largest Extreme Value (Ecuación 24), definiendo el tiempo inicial de comienzo como una distribución uniforme (Ecuación 25) entre 0 y 40 ms para simular la relación temporal aleatoria entre la llegada de paquetes al cliente y el límite de trama del enlace inverso.

$$f(x) = \frac{1}{b} e^{-\frac{x-a}{b}} e^{-e^{-\frac{x-a}{b}}}, b > 0$$

Ecuación 24: Función de densidad de probabilidad de una distribución Largest Extreme Value.

$$f(x) = \frac{1}{b-a} \quad a \leq x \leq b$$

Ecuación 25: Función de densidad de probabilidad de una distribución Uniforme.

Los parámetros por considerar se recogen de forma resumida en la Tabla 108, en los que se ha considerado una cabecera comprimida UDP de 2 bytes y de IPv4 de 20 bytes.

Componente	Distribución		Parámetros	
	DL	UL	DL	UL
Llegada inicial de paquete (ms)	Uniforme	Uniforme	a = 0, b = 40	a = 0, b = 40
Tiempo de llegada de paquete (ms)	Largest Extreme Value [X = [a - b ln(-ln Y)]] Y ∈ U(0,1)	Largest Extreme Value [X = [a - b ln(-ln Y)]] Y ∈ U(0,1)	a = 50, b = 4,5	a = 40, b = 6
Tamaño de paquete (B)	Largest Extreme Value [X = [a - b ln(-ln Y)]] + 22* Y ∈ U(0,1)	Largest Extreme Value [X = [a - b ln(-ln Y)]] + 22* Y ∈ U(0,1)	a = 330, b = 82	a = 45, b = 5,7

Tabla 108: Parámetros del tráfico gaming.

15. Anexo II: Selección de alternativas para la parametrización de los escenarios WLAN

Con el objetivo de hacer más sencillo el seguimiento de las siguientes tablas y cálculos para la selección de alternativas de parametrización WLAN, en la Tabla 109 se recoge el mapeo entre los criterios y sus identificadores, y en la Tabla 110 se hace lo propio para los escenarios en estudio.

Criterio	Identificador
Referencia del Estudio teórico	C.1.1
Ajuste a la actualidad de la tecnología	C.1.2
Innovación	C.1.3
Complejidad de implementación	C.1.4
Consumo de recursos	C.1.5

Tabla 109: Mapeo de los criterios de selección de la parametrización WLAN y cada identificador asociado.

Escenario	Identificador
Entorno doméstico	E1
Entorno de oficinas	E.2
Entorno interior de alta densidad	E.3
Entorno exterior de alta densidad	E.4

Tabla 110: Mapeo de los entornos en estudio y sus identificadores asociados.

De este modo, el cruce de criterios, parámetros, dispositivos y escenarios puede recogerse sobre cada tabla, lo que facilita la interpretación de las decisiones tomadas.

Asimismo, cabe mencionar que, debido a las dimensiones de las tablas y la multitud de aspectos a contemplar en cada una, los nombres de las alternativas se ven abreviados lo máximo posible, haciendo enfoque sobre el concepto más que en las especificaciones derivadas del mismo. En la Selección de la Solución para la Parametrización de los entornos WLAN se puede encontrar mayor detalle sobre estos.

Para comenzar, la Tabla 111 recoge la aplicación de los criterios anteriormente mencionados sobre cada posible implementación de los **aspectos espaciales** para cada escenario.

Parámetro	Alternativa	C.1.1	C.1.2	C.1.3	C.1.4	C.1.5	Resultado	
Forma	E.1	Rectangular	5	3	3	4	4	3.9
		Circular	0	3	3	3	4	2.25
		Cuadrículas	0	3	3	2	4	2.1
	E.2	Rectangular	5	3	3	4	4	3.9
		Circular	0	3	3	3	4	2.25
		Cuadrículas	0	3	3	2	4	2.1
	E.3	Rectangular	0	2	3	4	4	2.15
		Circular	5	4	3	3	4	4
		Cuadrículas	0	3	3	2	4	2.1
	E.4	Rectangular	0	2	3	4	4	2.15
		Circular	5	4	3	3	4	4
		Cuadrículas	0	3	3	2	4	2.1
Variación de la dimensión del escenario	E.1	Baja	5	4	3	3	5	4.15
		Media	2	3	3	3	3	2.7
		Alta	0	2	3	3	1	1.55

	E.2	Baja Media Alta	2 5 0	3 5 2	3 3 3	3 3 3	5 3 1	3 4.1 1.55
	E.3	Baja Media Alta	0 0 5	2 4 5	3 3 3	3 3 3	5 3 1	1.9 2.35 3.8
	E.4	Baja Media Alta	0 0 5	2 4 5	3 3 3	3 3 3	5 3 1	1.9 2.35 3.8
	Área	E.1	Baja Media Alta	3 4 0	3 4 3	3 3 3	3 3 3	3 3 3
E.2		Baja Media Alta	3 4 0	4 4 3	3 3 3	3 3 3	3 3 3	3.25 3.55 2.1
E.3		Baja Media Alta	3 4 3	2 4 4	3 3 3	3 3 3	3 3 3	2.75 3.55 3.25
E.4		Baja Media Alta	3 4 3	2 4 4	3 3 3	3 3 3	3 3 3	2.75 3.55 3.25

Tabla 111: Selección de alternativas espaciales por escenario.

A continuación, la Tabla 112 describe las diferentes selecciones realizadas en base a los criterios definidos para la **caracterización topológica, de posicionamiento y movilidad tanto para APs como STAs**. Por cada parámetro se recoge a la izquierda el valor específico de los APs, y el de las STAs a la derecha.

Parámetro	Alternativa	C.1.1	C.1.2	C.1.3	C.1.4	C.1.5	Resultado	
Modelo de definición del número de APs / STAs por red	E.1	Determinista Aleatorio	5 / 1 1 / 5	3 / 3 3 / 3	3 / 3 3 / 4	4 / 4 3 / 3	4 / 4 3 / 3	3.9 / 2.7 2.4 / 3.75
	E.2	Determinista Aleatorio	5 / 1 1 / 5	3 / 3 3 / 3	3 / 3 3 / 4	4 / 4 3 / 3	4 / 4 3 / 3	3.9 / 2.7 2.4 / 3.75
	E.3	Determinista Aleatorio	5 / 1 1 / 5	3 / 3 3 / 3	3 / 3 3 / 4	4 / 4 3 / 3	4 / 4 3 / 3	3.9 / 2.7 2.4 / 3.75
	E.4	Determinista Aleatorio	5 / 1 1 / 5	3 / 3 3 / 3	3 / 3 3 / 4	4 / 4 3 / 3	4 / 4 3 / 3	3.9 / 2.7 2.4 / 3.75
Densidad de APs / STAs por red	E.1	Baja Media Alta	5 / 5 3 / 3 2 / 2	5 / 5 3 / 3 2 / 2	3 / 3 3 / 3 3 / 3	3 / 3 3 / 3 3 / 3	5 / 5 3 / 3 1 / 1	4.4 / 4.4 3 / 3 2.15 / 2.15
	E.2	Baja Media Alta	5 / 2 3 / 5 2 / 3	5 / 2 3 / 5 2 / 3	3 / 3 3 / 3 3 / 3	3 / 3 3 / 3 3 / 3	5 / 5 3 / 3 1 / 1	4.4 / 2.75 3 / 4.1 2.15 / 2.7
	E.3	Baja Media Alta	5 / 0 3 / 3 2 / 5	5 / 0 3 / 3 2 / 5	3 / 3 3 / 3 3 / 3	3 / 3 3 / 3 3 / 3	5 / 5 3 / 3 1 / 1	4.4 / 1.65 3 / 3 2.15 / 3.8
	E.4	Baja Media Alta	5 / 0 3 / 3 2 / 5	5 / 0 3 / 3 2 / 5	3 / 3 3 / 3 3 / 3	3 / 3 3 / 3 3 / 3	5 / 5 3 / 3 1 / 1	4.4 / 1.65 3 / 3 2.15 / 3.8
Interfaces de Red por AP / STA	E.1	Única Múltiples	5 / 5 0 / 0	5 / 5 1 / 0	3 / 3 3 / 3	5 / 5 0 / 0	5 / 5 0 / 0	4.7 / 4.7 1.15 / 0.9
	E.2	Única Múltiples	5 / 5 0 / 0	5 / 5 1 / 0	3 / 3 3 / 3	5 / 5 0 / 0	5 / 5 0 / 0	4.7 / 4.7 0.7 / 0.45
	E.3	Única Múltiples	5 / 5 0 / 0	5 / 5 1 / 0	3 / 3 3 / 3	5 / 5 0 / 0	5 / 5 0 / 0	4.7 / 4.7 0.7 / 0.45
	E.4	Única	5 / 5	5 / 5	3 / 3	5 / 5	5 / 5	4.7 / 4.7

		Múltiples	0/0	1/0	3/3	0/0	0/0	0.7/0.45	
Tipo de posicionamiento de APs / STAs	E.1	Determinista	0/0	0/0	3/3	3/3	4/4	1.5/1.5	
		Aleatorio	5/5	5/5	3/4	3/3	3/3	4.1/3.15	
	E.2	Determinista	0/3	0/2	3/3	3/3	4/4	1.5/2.9	
		Aleatorio	5/3	5/3	3/4	3/3	3/3	4.1/3.15	
Modelo de altura de los APs / STAs	E.3	Determinista	5/3	5/3	3/3	3/3	4/4	4.25/3.15	
		Aleatorio	0/5	0/5	3/4	3/3	3/3	1.35/ 4.25	
	E.4	Determinista	5/0	5/0	3/3	3/3	4/4	4.25/1.5	
		Aleatorio	0/5	0/5	3/4	3/3	3/3	1.35/ 4.25	
Modelo de movilidad de los APs / STAs	E.1	Determinista	5/0	5/0	3/3	3/3	4/4	4.25/1.5	
		Aleatorio	0/5	0/5	3/3	3/3	3/3	1.35/ 4.1	
	E.2	Determinista	5/0	5/0	3/3	3/3	4/4	4.25/1.5	
		Aleatorio	0/5	0/5	3/3	3/3	3/3	1.35/ 4.1	
Modelo de movilidad de los APs / STAs	E.3	Determinista	0/0	0/0	3/3	3/3	4/4	1.5/1.5	
		Aleatorio	5/5	5/5	3/3	3/3	3/3	4.1/4.1	
		E.4	Determinista	0/0	0/0	3/3	3/3	4/4	1.5/1.5
		Aleatorio	5/5	5/5	3/3	3/3	3/3	4.1/4.1	
	E.1	Estático	5/4	5/4	3/3	3/3	4/4	4.25/3.7	
		V. cte.	0/0	0/0	3/3	3/3	3/3	1.35/1.35	
		A. cte.	0/0	0/0	3/3	3/3	3/3	1.35/1.35	
		R. Walk	0/5	0/5	3/3	3/3	3/3	1.35/ 4.1	
	E.2	Estático	5/5	5/5	3/3	3/3	4/4	4.25/4.25	
		V. cte.	0/0	0/0	3/3	3/3	3/3	1.35/1.35	
		A. cte.	0/0	0/0	3/3	3/3	3/3	1.35/1.35	
		R. Walk	0/4	0/4	3/3	3/3	3/3	1.35/1.9	
E.3	Estático	5/3	5/3	3/3	3/3	4/4	4.25/3.15		
	V. cte.	0/0	0/0	3/3	3/3	3/3	1.35/1.35		
	A. cte.	0/0	0/0	3/3	3/3	3/3	1.35/1.35		
	R. Walk	0/5	0/5	3/3	3/3	3/3	1.35/ 4.1		
E.4	Estático	5/0	5/0	3/3	3/3	4/4	4.25/1.5		
	V. cte.	0/0	0/0	3/3	3/3	3/3	1.35/1.35		
	A. cte.	0/0	0/0	3/3	3/3	3/3	1.35/1.35		
	R. Walk	0/5	0/5	3/3	3/3	3/3	1.35/ 2.45		

Tabla 112: Selección de alternativas topológicas, de posicionamiento y movilidad por escenario y dispositivo.

Después, se recoge la selección de **alternativas en términos de frecuencia** (Tabla 113), pudiéndose apreciar repeticiones entre los cálculos debido a que la selección de los tamaños de canal y banda a utilizar deben ser comunes para asegurar cierta homogeneidad en las simulaciones. Esto se debe a que el empleo de diferentes bandas de frecuencia está asociado a la utilización, por ejemplo, de distintos anchos de canal, lo que afecta a gran parte de las condiciones tecnológicas del planteamiento de este estudio. En consecuencia, resulta necesario relacionar dichos parámetros de forma común a todos los escenarios para la realización de las simulaciones.

Parámetro	Alternativa	C.1.1	C.1.2	C.1.3	C.1.4	C.1.5	Resultado	
Banda frecuencial	E.1	2,4 GHz	3	3	3	3	3	3
		5 GHz	5	5	3	3	3	4.1
		6 GHz	3	4	4	3	3	3.4
	E.2	2,4 GHz	3	3	3	3	3	3
		5 GHz	5	5	3	3	3	4.1
		6 GHz	3	4	4	3	3	3.4
	E.3	2,4 GHz	3	3	3	3	3	3
		5 GHz	5	5	3	3	3	4.1
		6 GHz	3	4	4	3	3	3.4

Modelo de configuración de canales	E.4	2,4 GHz	3	3	3	3	3	3
		5 GHz	5	5	3	3	3	4.1
		6 GHz	3	4	4	3	3	3.4
	E.1	Aleatorio	5	5	3	3	3	4.1
		Compartido	4	4	3	3	3	3.55
	Determinista	2	2	3	2	3	2.3	
	E.2	Aleatorio	3	3	3	3	3	3
	Compartido	4	4	3	3	3	3.55	
	Determinista	5	5	3	2	3	3.95	
	E.3	Aleatorio	3	3	3	3	3	3
	Compartido	5	5	3	3	3	4.1	
	Determinista	4	4	3	2	3	3.4	
	E.4	Aleatorio	3	3	3	3	3	3
	Compartido	5	5	3	3	3	4.1	
	Determinista	4	4	3	2	3	3.4	

Tabla 113: Selección de alternativas en términos de frecuencia.

Asimismo, se separa la **selección del ancho de canal** (Tabla 114) de las dos anteriores ya que es consecuencia de la elección previa de la banda de frecuencias. Por ejemplo, en 2,4 GHz no se pueden utilizar canales de 160 MHz debido a limitaciones técnicas y regulatorias del empleo del espectro.

Parámetro	Alternativa	C.1.1	C.1.2	C.1.3	C.1.4	C.1.5	Resultado	
Ancho de canal (en base a la banda elegida)	E.1	20 MHz	1	1	1	3	5	1.9
		40 MHz	1	1	3	3	4	2.05
		80 MHz	5	5	3	3	3	4.1
		160 MHz	3	3	4	3	1	2.85
	E.2	20 MHz	1	1	1	3	5	1.9
		40 MHz	1	1	3	3	4	2.05
		80 MHz	5	5	3	3	3	4.1
		160 MHz	3	3	4	3	1	2.85
	E.3	20 MHz	1	1	1	3	5	1.9
		40 MHz	1	1	3	3	4	2.05
		80 MHz	5	5	3	3	3	4.1
		160 MHz	3	3	4	3	1	2.85
	E.4	20 MHz	1	1	1	3	5	1.9
		40 MHz	1	1	3	3	4	2.05
		80 MHz	5	5	3	3	3	4.1
		160 MHz	3	3	4	3	1	2.85

Tabla 114: Selección del ancho de canal.

Una vez realizada la especificación frecuencia, se procede a seleccionar las **alternativas de transmisión para cada dispositivo y escenario**, como se recoge en la Tabla 115.

Parámetro	Alternativa	C.1.1	C.1.2	C.1.3	C.1.4	C.1.5	Resultado	
Configuración de antenas de los APs	E.1	1x1	3	3	2	3	4	3
		2x2	5	5	3	3	4	4.25
		4x4	2	3	3	3	3	2.7
		8x8	1	3	4	3	3	2.55
	E.2	1x1	3	3	2	3	4	3
		2x2	5	5	3	3	4	4.25
		4x4	2	3	3	3	3	2.7
		8x8	1	3	4	3	3	2.55
	E.3	1x1	2	3	2	3	4	2.7
		2x2	3	3	3	3	4	3.15
		4x4	5	5	3	3	3	4.1

	E.4	8x8	3	3	4	3	3	3.15
		1x1	2	3	2	3	4	2.7
		2x2	3	3	3	3	4	3.15
		4x4	5	5	3	3	3	4.1
		8x8	3	3	4	3	3	3.15
Configuración de antenas de los APs	E.1	1x1	5	3	2	3	4	3.6
		2x2	3	3	3	3	3	3
		3x3	1	4	4	3	3	2.8
	E.2	1x1	5	3	2	3	4	3.6
		2x2	3	3	3	3	3	3
		3x3	1	4	4	3	3	2.8
	E.3	1x1	3	3	2	3	4	3
		2x2	5	3	3	3	3	3.6
		3x3	2	4	4	3	3	3.1
	E.4	1x1	3	3	2	3	4	3
		2x2	5	3	3	3	3	3.6
		3x3	2	4	4	3	3	3.1
Número de flujos espaciales de los APs / STAs	E.1	Único	3 / 5	3 / 4	3 / 3	3 / 3	3 / 3	3 / 3.85
		Bajo	5 / 3	4 / 4	3 / 3	3 / 3	3 / 3	3.85 / 3.25
		Medio	3 / 3	4 / 4	3 / 3	3 / 3	4 / 4	3.6 / 3.6
		Alto	1 / 1	4 / 4	3 / 3	3 / 3	4 / 4	2.8 / 2.8
	E.2	Único	3 / 5	3 / 4	3 / 3	3 / 3	3 / 3	3 / 3.85
		Bajo	5 / 3	4 / 4	3 / 3	3 / 3	3 / 3	3.85 / 3.25
		Medio	3 / 3	4 / 4	3 / 3	3 / 3	4 / 4	3.6 / 3.6
		Alto	1 / 1	4 / 4	3 / 3	3 / 3	4 / 4	2.8 / 2.8
	E.3	Único	1 / 3	3 / 4	3 / 3	3 / 3	3 / 3	2.4 / 3.25
		Bajo	3 / 5	4 / 4	3 / 3	3 / 3	3 / 3	3.25 / 3.85
		Medio	5 / 3	4 / 4	3 / 3	3 / 3	4 / 4	4 / 3.6
		Alto	3 / 1	4 / 4	3 / 3	3 / 3	4 / 4	3.6 / 2.8
E.4	Único	1 / 3	3 / 4	3 / 3	3 / 3	3 / 3	2.4 / 3.25	
	Bajo	3 / 5	4 / 4	3 / 3	3 / 3	3 / 3	3.25 / 3.85	
	Medio	5 / 3	4 / 4	3 / 3	3 / 3	4 / 4	4 / 3.6	
	Alto	3 / 1	4 / 4	3 / 3	3 / 3	4 / 4	3.6 / 2.8	
Modelo de potencia TX de los APs / STAs	E.1	Estático	3 / 3	3 / 3	3 / 3	3 / 3	3 / 3	3 / 3
		Variable	4 / 4	4 / 4	3 / 3	3 / 3	3 / 3	3.55 / 3.55
	E.2	Estático	3 / 3	3 / 3	3 / 3	3 / 3	3 / 3	3 / 3
		Variable	4 / 4	4 / 4	3 / 3	3 / 3	3 / 3	3.55 / 3.55
E.3	Estático	3 / 3	3 / 3	3 / 3	3 / 3	3 / 3	3 / 3	
	Variable	4 / 4	4 / 4	3 / 3	3 / 3	3 / 3	3.55 / 3.55	
E.4	Estático	3 / 3	3 / 3	3 / 3	3 / 3	3 / 3	3 / 3	
	Variable	4 / 4	4 / 4	3 / 3	3 / 3	3 / 3	3.55 / 3.55	
Potencia TX de los APs / STAs	E.1	Baja	3 / 2	3 / 3	3 / 3	3 / 3	3 / 3	3 / 2.7
		Media	5 / 5	4 / 4	3 / 3	3 / 3	3 / 3	3.85 / 3.85
		Alta	3 / 1	3 / 3	3 / 3	3 / 3	3 / 3	3 / 2.4
	E.2	Baja	1 / 3	3 / 3	3 / 3	3 / 3	3 / 3	2.4 / 3
		Media	3 / 5	4 / 4	3 / 3	3 / 3	3 / 3	3.25 / 3.85
		Alta	5 / 3	4 / 3	3 / 3	3 / 3	3 / 3	3.85 / 3
	E.3	Baja	1 / 3	3 / 3	3 / 3	3 / 3	3 / 3	2.4 / 3
		Media	3 / 5	4 / 4	3 / 3	3 / 3	3 / 3	3.25 / 3.85
		Alta	5 / 3	4 / 3	3 / 3	3 / 3	3 / 3	3.85 / 3
	E.4	Baja	1 / 3	3 / 3	3 / 3	3 / 3	3 / 3	2.4 / 3
		Media	3 / 5	4 / 4	3 / 3	3 / 3	3 / 3	3.25 / 3.85
		Alta	5 / 3	4 / 3	3 / 3	3 / 3	3 / 3	3.85 / 3

Tabla 115: Selección de las alternativas de transmisión por escenario y dispositivo.

En los parámetros que aglutinan tanto los puntos de acceso como las estaciones, los parámetros a la izquierda hacen alusión a la selección para los APs, y los de la derecha para las STAs.

Debido a la extensión de la tabla para la configuración de la transmisión, se ha separado la selección del **modelo de propagación** a la Tabla 116, donde se especifica los modelos elegidos para ser simulados en cada escenario bajo estudio, siguiendo los criterios especificados en el apartado dedicado a los Criterios de Selección para la Parametrización de los entornos WLAN.

Parámetro	Alternativa	C.1.1	C.1.2	C.1.3	C.1.4	C.1.5	Resultado	
Modelo de propagación	E.1	Pérdidas fijas	0	3	3	3	4	2.25
		Pérdidas aleatorias	0	3	3	3	4	2.25
		LogDistance	2	3	3	3	4	2.85
		3 distance LogDistance	2	3	3	3	3	2.7
		Friis	0	3	3	3	3	2.1
		V. cte.	0	3	3	3	3	2.1
		ITU-R 1411 LOS	0	3	3	3	3	2.1
		Nakagami	0	3	3	3	3	2.1
		3GPP Office Mixed	0	3	3	3	3	2.1
		3GPP Office	0	3	3	3	3	2.1
		UMa	0	3	3	3	3	2.1
		UMi	0	3	3	3	3	2.1
		RMa	0	3	3	3	3	2.1
		Modelado específico	4	4	4	2	3	3.55
	E.2	Pérdidas fijas	0	3	3	3	4	2.25
		Pérdidas aleatorias	0	3	3	3	4	2.25
		LogDistance	3	3	3	3	4	3.15
		3 distance LogDistance	3	3	3	3	3	3
		Friis	0	3	3	3	3	2.1
		V. cte.	0	3	3	3	3	2.1
		ITU-R 1411 LOS	0	3	3	3	3	2.1
		Nakagami	0	3	3	3	3	2.1
		3GPP Office Mixed	4	4	3	3	3	3.55
		3GPP Office	5	4	3	3	3	3.85
		UMa	0	3	3	3	3	2.1
		UMi	0	3	3	3	3	2.1
		RMa	0	3	3	3	3	2.1
		Modelado específico	4	4	4	1	3	3.4
	E.3	Pérdidas fijas	0	3	3	3	4	2.25
		Pérdidas aleatorias	0	3	3	3	4	2.25
		LogDistance	4	3	3	3	4	3.45

		3 distance LogDistance	4	3	3	3	3	3.3
		Friis	0	3	3	3	3	2.1
		V. cte.	0	3	3	3	3	2.1
		ITU-R 1411 LOS	0	3	3	3	3	2.1
		Nakagami	0	3	3	3	3	2.1
		3GPP Office Mixed	0	3	3	3	3	2.1
		3GPP Office	0	3	3	3	3	2.1
		UMa	0	3	3	3	3	2.1
		UMi	0	3	3	3	3	2.1
		RMa	0	3	3	3	3	2.1
		Modelado específico	4	4	4	2	3	3.55
	E.4	Pérdidas fijas	0	3	3	3	4	2.25
		Pérdidas aleatorias	0	3	3	3	4	2.25
		LogDistance	4	3	3	5	4	3.75
		3 distance LogDistance	4	3	3	3	3	3.3
		Friis	0	3	3	3	3	2.1
		V. cte.	0	3	3	3	3	2.1
		ITU-R 1411 LOS	3	3	3	3	3	3
		Nakagami	0	3	3	3	3	2.1
		3GPP Office Mixed	0	3	3	3	3	2.1
		3GPP Office	0	3	3	3	3	2.1
		UMa	4	4	3	3	3	3.55
		UMi	4	4	3	3	3	3.55
		RMa	3	4	3	3	3	3.25
		Modelado específico	4	4	3	2	3	3.4

Tabla 116: Selección de los modelos de propagación a implementar en cada escenario.

Tras la selección de las configuraciones principales a nivel de escenario, dispositivos y transmisión, en la Tabla 117 se recogen las **características a nivel de protocolo**, incluyendo parámetros tanto de nivel físico como de enlace.

Parámetro	Alternativa	C.1.1	C.1.2	C.1.3	C.1.4	C.1.5	Resultado	
EDCA	E.1	Sí	5	5	3	3	3	4.1
		No	0	0	3	3	3	1.35
	E.2	Sí	5	5	3	3	3	4.1
		No	0	0	3	3	3	1.35
	E.3	Sí	5	5	3	3	3	4.1
		No	0	0	3	3	3	1.35
	E.4	Sí	5	5	3	3	3	4.1
		No	0	0	3	3	3	1.35
Modificación de parámetros EDCA	E.1	Sí	0	3	3	3	3	2.1
		No	5	3	3	3	3	3.6
	E.2	Sí	0	3	3	3	3	2.1
		No	5	3	3	3	3	3.6
	E.3	Sí	0	3	3	3	3	2.1

	E.4	No	5	3	3	3	3	3.6
		Sí	0	3	3	3	3	2.1
Agregación	E.1	No	3	3	3	3	3	3
		A-MPDU	5	3	3	3	3	3.6
		A-MSDU	0	3	3	2	2	1.8
		Ambas	0	3	3	2	2	1.8
	E.2	No	3	3	3	3	3	3
		A-MPDU	5	3	3	3	3	3.6
		A-MSDU	0	3	3	2	2	1.8
		Ambas	0	3	3	2	2	1.8
	E.3	No	3	3	3	3	3	3
		A-MPDU	5	3	3	3	3	3.6
		A-MSDU	0	3	3	2	2	1.8
		Ambas	0	3	3	2	2	1.8
E.4	No	3	3	3	3	3	3	
	A-MPDU	5	3	3	3	3	3.6	
	A-MSDU	0	3	3	2	2	1.8	
	Ambas	0	3	3	2	2	1.8	
Repeticiones	E.1	Estándar	5	5	3	3	3	4.1
		Específicas	0	0	3	3	3	1.35
	E.2	Estándar	5	5	3	3	3	4.1
		Específicas	0	0	3	3	3	1.35
E.3	Estándar	5	5	3	3	3	4.1	
	Específicas	0	0	3	3	3	1.35	
E.4	Estándar	5	5	3	3	3	4.1	
	Específicas	0	0	3	3	3	1.35	
GI	E.1	800 ns	5	4	3	3	3	3.85
		1600 ns	3	4	3	3	4	3.4
		3200 ns	3	4	3	3	4	3.4
	E.2	800 ns	5	4	3	3	3	3.85
		1600 ns	3	4	3	3	4	3.4
		3200 ns	3	4	3	3	4	3.4
	E.3	800 ns	5	4	3	3	3	3.85
		1600 ns	3	4	3	3	4	3.4
		3200 ns	3	4	3	3	4	3.4
	E.4	800 ns	5	4	3	3	3	3.85
		1600 ns	3	4	3	3	4	3.4
		3200 ns	3	4	3	3	4	3.4
Tiempos IFS	E.1	Estándar	5	5	3	3	3	4.1
		Específicos	0	0	3	3	3	1.35
	E.2	Estándar	5	5	3	3	3	4.1
		Específicos	0	0	3	3	3	1.35
E.3	Estándar	5	5	3	3	3	4.1	
	Específicos	0	0	3	3	3	1.35	
E.4	Estándar	5	5	3	3	3	4.1	
	Específicos	0	0	3	3	3	1.35	
RTS / CTS	E.1	Sí	3	4	3	3	2	3.1
		No	5	3	3	3	4	3.75
	E.2	Sí	3	4	3	3	2	3.1
		No	5	3	3	3	4	3.75
	E.3	Sí	5	4	3	3	2	3.7
		No	3	3	3	3	4	3.15
	E.4	Sí	5	4	3	3	2	3.7
		No	3	3	3	3	4	3.15
Umbral RTS / CTS	E.1	Estándar	5	5	3	3	2	3.95
		Específico	0	0	3	3	4	1.5

	E.2	Estándar Específico	5 0	5 0	3 3	3 3	2 4	3.95 1.5
	E.3	Estándar Específico	5 0	5 0	3 3	3 3	2 4	3.95 1.5
	E.4	Estándar Específico	5 0	5 0	3 3	3 3	2 4	3.95 1.5

Tabla 117: Selección de las características tecnológicas principales a nivel de protocolo.

Una vez cerrada la caracterización tecnológica, se procede a seleccionar los **tipos de tráfico a implementar** en el simulador. Debido a cuestiones logísticas, y con la intención de no extender el trabajo en exceso, se ha decidido seleccionar los 5 tipos de tráfico con mayor puntuación para su implementación en todos los casos.

Parámetro	Alternativa		C.1.1	C.1.2	C.1.3	C.1.4	C.1.5	Resultado
Tipo de tráfico	E.1	Gestión	3	3	3	4	4	3.3
		VoD	5	5	5	4	3	4.55
		Videoconf.	0	3	5	3	3	2.4
		HTTP	5	5	1	5	4	4.25
		FTP	4	4	5	3	3	3.85
		VoIP	4	5	5	3	3	4.1
		Gaming	5	5	5	3	3	4.4
		IoT	2	5	5	3	4	3.65
	E.2	AR/VR	2	5	5	3	3	3.5
		Gestión	3	3	3	4	4	3.3
		VoD	5	5	5	4	3	4.55
		Videoconf.	3	3	5	3	3	2.7
		HTTP	5	5	1	5	4	4.25
		FTP	4	4	5	3	3	3.85
		VoIP	1	5	5	3	3	2.9
		Gaming	2	5	5	3	3	3.8
	E.3	IoT	2	5	5	3	4	3.65
		AR/VR	2	5	5	3	3	3.5
		Gestión	3	3	3	4	4	3.3
		VoD	5	5	5	4	3	4.55
		Videoconf.	0	3	5	3	3	2.4
		HTTP	5	5	1	5	4	4.25
		FTP	4	4	5	3	3	3.85
		VoIP	4	5	5	3	3	4.1
	E.4	Gaming	5	5	5	3	3	4.4
		IoT	2	5	5	3	4	3.65
		AR/VR	2	5	5	3	3	3.5
		Gestión	3	3	3	4	4	3.3
VoD		5	5	5	4	3	4.55	
Videoconf.		0	3	5	3	3	2.4	
HTTP		5	5	1	5	4	4.25	
FTP		4	4	5	3	3	3.85	

Tabla 118: Selección de los tipos de tráfico a implementar en cada escenario.

De este modo, los modelos a introducir son el tráfico VoD, HTTP, FTP, VoIP y Gaming. Una vez seleccionados cuáles utilizar, se debe definir la **asignación de los flujos** a los dispositivos. Como en ocasiones anteriores, en las posiciones a la izquierda de la Tabla

119 se indica el flujo en sentido de transmisión desde los APs, y a la derecha la transmisión en sentido contrario, es decir, desde las STAs.

Parámetro	Alternativa	C.1.1	C.1.2	C.1.3	C.1.4	C.1.5	Resultado	
Tráfico VoD por porcentaje de APs / STAs	E.1	Nulo	0/0	0/0	0/0	5/5	5/5	1.5 / 1.5
		Bajo	3/3	4/4	4/4	3/3	2/2	3.25 / 3.25
		Medio	5/0	4/4	4/4	3/3	2/2	3.85 / 2.35
		Alto	3/0	4/4	4/4	3/3	2/2	3.25 / 2.35
	E.2	Nulo	0/0	0/0	0/0	5/5	5/5	1.5 / 1.5
		Bajo	5/3	4/4	4/4	3/3	2/2	3.85 / 3.25
		Medio	3/0	4/4	4/4	3/3	2/2	3.25 / 2.35
		Alto	3/0	4/4	4/4	3/3	2/2	3.25 / 2.35
	E.3	Nulo	0/0	0/0	0/0	5/5	5/5	1.5 / 1.5
		Bajo	3/3	4/4	4/4	3/3	2/2	3.25 / 3.25
		Medio	5/0	4/4	4/4	3/3	2/2	3.85 / 2.35
		Alto	3/0	4/4	4/4	3/3	2/2	3.25 / 2.35
	E.4	Nulo	0/0	0/0	0/0	5/5	5/5	1.5 / 1.5
		Bajo	3/3	4/4	4/4	3/3	2/2	3.25 / 3.25
		Medio	3/0	4/4	4/4	3/3	2/2	3.25 / 2.35
		Alto	5/0	4/4	4/4	3/3	2/2	3.85 / 2.35
Tráfico HTTP por porcentaje de APs / STAs	E.1	Nulo	0/5	0/5	0/0	5/5	5/5	1.5 / 5
		Bajo	3/0	3/0	3/3	4/4	3/3	3.15 / 1.5
		Medio	5/0	5/0	3/3	4/4	3/3	4.25 / 1.5
		Alto	3/0	3/0	3/3	4/4	3/3	3.15 / 1.5
	E.2	Nulo	0/5	0/5	0/0	5/5	5/5	1.5 / 5
		Bajo	3/0	3/0	3/3	4/4	3/3	3.15 / 1.5
		Medio	5/0	5/0	3/3	4/4	3/3	4.25 / 1.5
		Alto	3/0	3/0	3/3	4/4	3/3	3.15 / 1.5
	E.3	Nulo	0/5	0/5	0/0	5/5	5/5	1.5 / 5
		Bajo	3/0	3/0	3/3	4/4	3/3	3.15 / 1.5
		Medio	5/0	5/0	3/3	4/4	3/3	4.25 / 1.5
		Alto	3/0	3/0	3/3	4/4	3/3	3.15 / 1.5
	E.4	Nulo	0/5	0/5	0/0	5/5	5/5	1.5 / 5
		Bajo	3/0	3/0	3/3	4/4	3/3	3.15 / 1.5
		Medio	5/0	5/0	3/3	4/4	3/3	4.25 / 1.5
		Alto	3/0	3/0	3/3	4/4	3/3	3.15 / 1.5
Tráfico FTP por porcentaje de APs / STAs	E.1	Nulo	0/3	0/3	2/2	5/5	5/5	1.8 / 3.45
		Bajo	5/3	5/3	4/4	4/4	3/3	4.4 / 3.3
		Medio	3/3	3/3	4/4	4/4	3/2	3.3 / 3.15
		Alto	0/3	0/3	4/4	4/4	3/2	1.65 / 3.15
	E.2	Nulo	0/3	0/3	2/2	5/5	5/5	1.8 / 3.45
		Bajo	5/3	5/3	4/4	4/4	3/3	4.4 / 3.3
		Medio	3/3	3/3	4/4	4/4	3/2	3.3 / 3.15
		Alto	0/3	0/3	4/4	4/4	3/2	1.65 / 3.15
	E.3	Nulo	0/3	0/3	2/2	5/5	5/5	1.8 / 3.45
		Bajo	5/3	5/3	4/4	4/4	3/3	4.4 / 3.3
		Medio	3/3	3/3	4/4	4/4	3/2	3.3 / 3.15
		Alto	0/3	0/3	4/4	4/4	3/2	1.65 / 3.15
	E.4	Nulo	0/3	0/3	0/0	5/5	5/5	1.8 / 3.45
		Bajo	5/3	5/3	4/4	4/4	3/3	4.4 / 3.3
		Medio	3/3	3/3	4/4	4/4	3/2	3.3 / 3.15
		Alto	0/3	0/3	4/4	4/4	3/2	1.65 / 3.15
Tráfico VoIP por porcentaje de APs / STAs	E.1	Nulo	0/0	0/0	0/0	5/5	5/5	1.5 / 1.5
		Bajo	5/5	4/4	4/4	4/4	3/3	4.15 / 4.15
		Medio	3/3	4/4	4/4	4/4	3/3	3.55 / 3.55
		Alto	0/0	3/3	4/4	4/4	3/3	2.4 / 2.4
	E.2	Nulo	0/0	0/0	0/0	5/5	5/5	1.5 / 1.5

Tráfico Gaming por porcentaje de APs / STAs		Bajo	5/5	4/4	4/4	4/4	3/3	4.15 / 4.15
		Medio	3/3	4/4	4/4	4/4	3/3	3.55 / 3.55
		Alto	0/0	3/3	4/4	4/4	3/3	2.4 / 2.4
	E.3	Nulo	0/0	0/0	0/0	5/5	5/5	1.5 / 1.5
		Bajo	3/3	4/4	4/4	4/4	3/3	3.55 / 3.55
		Medio	5/5	4/4	4/4	4/4	3/3	4.15 / 4.15
	E.4	Alto	3/3	3/3	4/4	4/4	3/3	3.3 / 3.3
		Nulo	0/0	0/0	0/0	5/5	5/5	1.5 / 1.5
		Bajo	3/3	4/4	4/4	4/4	3/3	3.55 / 3.55
	E.1	Medio	5/5	4/4	4/4	4/4	3/3	4.15 / 4.15
		Alto	3/3	3/3	4/4	4/4	3/3	3.3 / 3.3
		Nulo	0/0	0/0	4/4	4/4	3/3	1.65 / 1.65
E.2	Bajo	5/5	3/3	4/4	4/4	3/3	3.9 / 3.9	
	Medio	3/3	3/3	4/4	4/4	3/3	3.3 / 3.3	
	Alto	0/0	0/0	4/4	4/4	3/3	1.65 / 1.65	
E.3	Nulo	5/5	5/5	0/0	5/5	5/5	4.25 / 4.25	
	Bajo	3/3	3/3	4/4	4/4	3/3	3.3 / 3.3	
	Medio	3/3	3/3	4/4	4/4	3/3	3.3 / 3.3	
E.4	Alto	0/0	0/0	4/4	4/4	3/3	1.65 / 1.65	
	Nulo	0/0	0/0	0/0	5/5	5/5	1.5 / 1.5	
	Bajo	3/3	3/3	4/4	4/4	3/3	3.3 / 3.3	
E.1	Medio	3/3	3/3	4/4	4/4	3/3	3.3 / 3.3	
	Alto	5/5	5/5	4/4	4/4	3/3	4.4 / 4.4	
	Nulo	3/3	3/3	4/4	4/4	3/3	3.3 / 3.3	
E.2	Bajo	0/0	0/0	0/0	5/5	5/5	1.5 / 1.5	
	Medio	3/3	3/3	4/4	4/4	3/3	3.3 / 3.3	
	Alto	3/3	3/3	4/4	4/4	3/3	3.3 / 3.3	
E.3	Nulo	5/5	5/5	4/4	4/4	3/3	4.4 / 4.4	
	Bajo	3/3	3/3	4/4	4/4	3/3	3.3 / 3.3	
	Medio	3/3	3/3	4/4	4/4	3/3	3.3 / 3.3	
E.4	Alto	5/5	5/5	4/4	4/4	3/3	4.4 / 4.4	

Tabla 119: Selección de la densidad de tráfico por tipos y dispositivos a implementar en cada escenario.

16. Anexo III: NS-3

NS-3 juega un papel fundamental en el desarrollo del presente Trabajo Fin de Máster. Sin embargo, su complejidad y la ambición del proyecto por caracterizar las redes inalámbricas de forma rigurosa requieren de una introducción en detalle. En consecuencia, se ha decidido reservar dicho contenido para este anexo, y profundizar en su conocimiento partiendo de una introducción al simulador previa a la explicación de la implementación final de entornos.

El código empleado también se encuentra disponible en GitHub con acceso público: <https://github.com/Atxa14/TFM-Files>

16.1. Introducción

NS-3 [115] es un simulador de eventos discretos de redes, de software libre y código abierto, con licencia GNU GPLv2, que es mantenido por una comunidad global.

Ejecutado de forma habitual a través de la línea de comandos y escrito en C++, se basa en el procesamiento de eventos de simulación a través de llamadas a funciones C++, organizadas por un organizador (scheduler). De forma conceptual, el simulador lleva el registro de una serie de eventos programados para ejecutarse en un tiempo de simulación específico. [115]

En base a ello, NS-3 provee modelos sobre el funcionamiento y rendimiento de las redes de paquetes de datos y provee un motor de simulación para el desarrollo de experimentos.

Los usuarios de NS-3 obtienen su código fuente, lo compilan en librerías compartidas o estáticas, y enlazan las librerías a programas de su autoría. El programa principal `main()` es donde se configura el escenario de simulación específico y donde se ejecuta y detiene la simulación.

Además, NS-3 proporciona la posibilidad de configurar los programas de usuario en Python, aunque la disponibilidad de tutoriales y módulos en dicho lenguaje es sustancialmente menor a la oferta de recursos en C++.

16.1.1. Modelo de abstracción de NS-3

La simulación de redes se lleva a cabo mediante diferentes niveles de abstracción [115] que representan los componentes de red y su comportamiento y características. A continuación, se introducen los elementos principales con un enfoque descendente en términos de abstracción.

Se utiliza el término **nodo** [115] para describir los hosts o equipos terminales. Se hace uso del término genérico proveniente de la teoría de Grafos debido a que NS-3 también permite la simulación de protocolos ajenos a Internet. Dicha abstracción se representa bajo la clase `Node` en C++.

Habitualmente, el software de los ordenadores se clasifica en dos amplias clases. [115] Por un lado, el software de sistema (System Software) organiza varios recursos como la memoria, los ciclos del procesador, discos, red, etcétera de acuerdo con un modelo de

computación. Sin embargo, este no suele utilizar directamente los recursos para beneficiar al usuario. El usuario típicamente ejecuta una aplicación que adquiere y utiliza los recursos controlados por el software de sistema para alcanzar algún objetivo.

En NS-3 no se dispone como tal del concepto de sistema operativo, pero sí de aplicación. Una **aplicación** (clase Application) [115] se ejecuta en un Node para dirigir las simulaciones, y es la abstracción básica mediante la cual un programa de usuario genera actividad a simular.

En NS-3 los nodos se conectan a objetos [115] que representan **canales** de comunicación. La clase Channel provee métodos para el manejo de objetos de comunicación y para conectar nodos a ellos. Los canales pueden especializarse en el sentido de programación orientada a objetos, pudiendo así modelar cosas simples como cables, o más complicadas como switch Ethernet u obstrucciones en el espacio tridimensional para comunicaciones inalámbricas.

Los **dispositivos de red** (Net Device) [115] son la abstracción de los drivers de dispositivos de red utilizados para el control de estos, e incluyen tanto el controlador software como el hardware simulado que componen las interfaces de red. Los Net Device se instalan en los nodos para permitirles comunicarse con otros nodos a través de canales. Un nodo puede tener varios dispositivos de red.

Además de los recursos anteriores, los **Topology Helpers** [115] ayudan al establecimiento de escenarios facilitando las configuraciones típicas de los dispositivos: asignación de direcciones IP, direcciones MAC, instalación de dispositivos de red en los nodos, etc.

En el caso concreto de la tecnología Wi-Fi, NS-3 provee de las siguientes **características** [116] para la implementación **del protocolo 802.11**:

- DCF 802.11 básico con modos infraestructura y ad-hoc.
- Capas físicas 802.11a, 802.11b, 802.11g, 802.11n (bandas de 2,4 GHz y 5 GHz), 802.11ac, 802.11ax (bandas de 2,4 GHz, 5 GHz y 6 GHz) y 802.11be.
- Extensiones de agregación de MSDU y MPDU de 802.11n, permitiendo su combinación.
- OFDMA en bajada y subida de 802.11ax incluyendo MU-EDCA.
- Descubrimiento y configuración multi-link de 802.11be.
- Extensiones de EDCA basadas en QoS y colas de 802.11e.
- Múltiples modelos de propagación y delay.
- Modelos de errores de paquete y detección de trama validados.
- Algoritmos de control de tasa de datos, como Aarf, Arf, Cara, Onoe, Rraa, ConstantRate, Minstrel y Minstrel-HT.
- 802.11s (en malla).
- 802.11p y WAVE (vehicular).

El conjunto de modelos 802.11 proporcionados en NS-3 persigue ofrecer una implementación precisa a nivel de MAC de la especificación 802.11, así como una

abstracción a nivel de paquete del nivel físico (PHY) de acuerdo con las especificaciones 802.11a/b/e/g/n/ac/ax/be.

En NS-3, como se ha introducido anteriormente, los nodos pueden tener múltiples dispositivos WifiNetDevice en canales separados, y un WifiNetDevice puede coexistir con otros tipos de dispositivos.

A través del uso de SpectrumWifiPhy, [116] es posible construir escenarios que involucren interferencia entre canales o múltiples tecnologías inalámbricas en un solo canal.

En definitiva, se basa en la utilización de **tres subcapas** [116]:

- **PHY:** Modelos dedicados a la representación de operaciones y funciones comunes de la capa PHY y características específicas de versiones.
- **MAC baja (low MAC):** Estos modelos representan funciones como el acceso al medio (DCF y EDCA), protección de tramas (RTS/CTS) y reconocimiento (ACK/BlockAck). En ns-3, la capa MAC de nivel inferior está compuesta por una jerarquía de Frame Exchange Manager, un Channel Access Manager y una entidad MAC intermedia.
- **MAC alta (high MAC):** Modelos que implementan procesos no críticos en medida de tiempos de Wi-Fi, como la generación de balizas a nivel MAC, sondas y máquinas de estado de asociación, así como un conjunto de algoritmos de control de tasa.

A continuación, la Figura 74 muestra un resumen del diseño basado en capas de la arquitectura Wi-Fi.

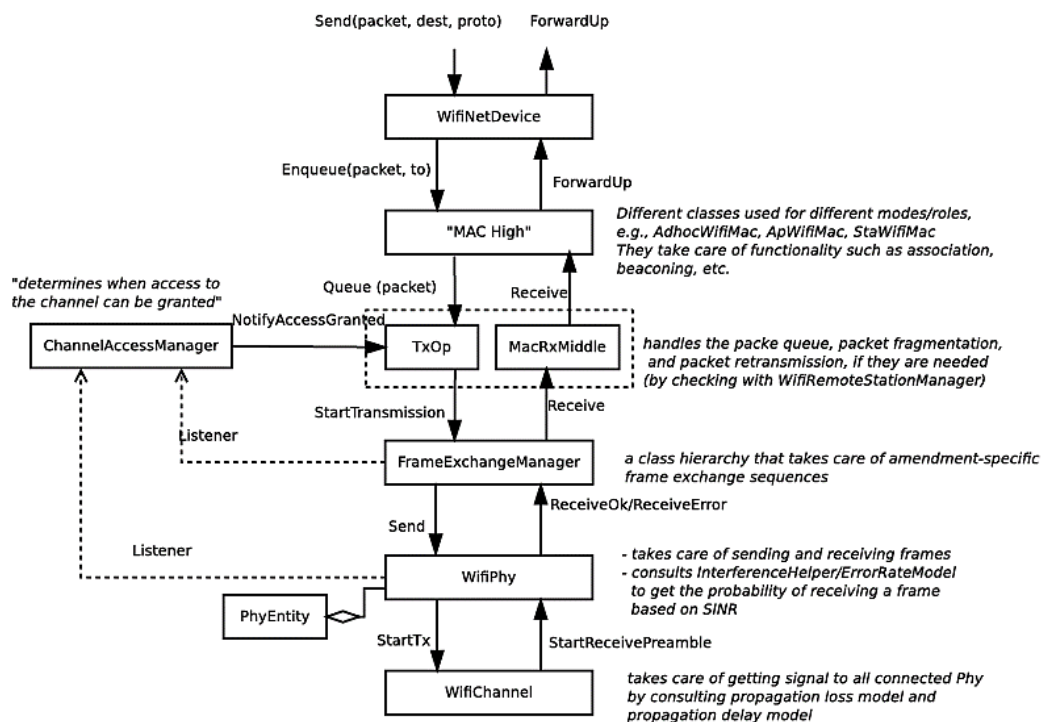


Figura 74: Diseño en capas del modelo Wi-Fi en NS-3. [116]

16.1.2. Limitaciones de las simulaciones Wi-Fi en NS-3

La especificación del estándar IEEE 802.11 es amplia y compleja, [116] lo que vuelve complicado cubrir todos los aspectos en las simulaciones de NS-3.

Los modelos de capa física y canal operan en una base de paquetes, sin considerar efectos de propagación de frecuencia selectiva ni de interferencia en su modelo por defecto (VansWifiPhy). [116] En la actualidad tampoco se admite el uso de antenas direccionales.

Para escenarios de ruido aditivo blanco gaussiano (AWGN) o escenarios de interferencia de banda ancha (wideband), [116] el rendimiento se rige por la aplicación de modelos analíticos (basados en la modulación y factores como el ancho de canal) a la relación señal-ruido recibida, donde el ruido combina el efecto del ruido térmico y la interferencia de otros paquetes Wi-Fi. La interferencia de otras tecnologías inalámbricas solo se modela cuando se utiliza SpectrumWifiPhy.

La siguiente lista recoge gran parte de las limitaciones [116] de los modelos físicos y de canal. Para ello, se especifica la característica incompatible o no admitida, y si procede, la versión del protocolo a la que aplica:

- MU-RTS/CTS (802.11ax/be)
- MIMO: no se pueden configurar más de 4 antenas por dispositivo.
- MU-MIMO (802.11ac/ax/be)
- Beamforming (802.11n/ac/ax/be)
- RIFS (802.11n)
- PCF/HCF/HCCA (802.11)
- Anuncio de Cambio de Canal (Channel Switch Announcement)
- Autenticación y cifrado
- Retrasos de procesamiento
- RTS/CTS y ACK en formatos HT/VHT/HE/EHT
- El modelo de consumo de energía no considera MIMO.
- El uso de preámbulo de 802.11ax es compatible con PHY pero actualmente no es aprovechada por la capa MAC.
- Implementación reducida de 802.11be: aproximación mínima de PHY (sin capa MAC).

En la capa MAC, se han implementado la mayoría de las funciones principales que se encuentran en 802.11a/b/e/g/n/ac/ax. Sin embargo, existen casuísticas concretas donde pueden tener lugar algunas limitaciones. [116] En concreto, el soporte para 802.11n, ac y ax está en constante evolución.

Por último, NS-3 incluye algunas implementaciones no impuestas por el estándar:

- Se asume que el conjunto de tasas básicas (BSSBasicRateSet) para 802.11b es de 1-2 Mbit/s, y para 802.11a/g de 6-12-24 Mbit/s.
- Se asume que el conjunto de tasas operativas (OperationalRateSet) contiene todas las tasas establecidas para el estándar.

- El administrador Wi-Fi siempre selecciona la tasa básica más baja para las tramas de gestión.

16.2. Modelado de Escenarios Wi-Fi con NS-3

El presente apartado recoge la explicación en detalle de la implementación del código en NS-3 para la realización de simulaciones Wi-Fi. En general, se procede del mismo modo que en los scripts de ejemplo –se incluyen bajo el directorio examples en la propia descarga del simulador– de las múltiples tecnologías que permite implementar NS-3.

16.2.1. Configuraciones Iniciales

Como resulta habitual, inicialmente se realizan una serie de **declaraciones include** (Figura 75) que permiten que el código tenga acceso a diferentes módulos que complementan el programa. En NS-3 se favorece la agrupación de include-s [115] sobre módulos relativamente grandes, lo que antepone la facilidad a la eficiencia en el diseño de scripts. Los ficheros adheridos al programa [115] se ubican en el directorio ns3 en el proceso de construcción (build) para evitar colisión de nombres de ficheros.

```

#include "ns3/applications-module.h"
#include "ns3/boolean.h"
#include "ns3/command-line.h"
#include "ns3/config.h"
#include "ns3/core-module.h"
#include "ns3/delay-jitter-estimation.h"
#include "ns3/double.h"
#include "ns3/enum.h"
#include "ns3/flow-monitor-module.h"
#include "ns3/internet-module.h"
#include "ns3/internet-stack-helper.h"
#include "ns3/ipv4-address-helper.h"
#include "ns3/ipv4-global-routing-helper.h"
#include "ns3/log.h"
#include "ns3/mobility-helper.h"
#include "ns3/mobility-module.h"
#include "ns3/multi-model-spectrum-channel.h"
#include "ns3/network-module.h"
#include "ns3/on-off-helper.h"
#include "ns3/packet-sink-helper.h"
#include "ns3/packet-sink.h"
#include "ns3/rng-seed-manager.h"
#include "ns3/simulator.h"
#include "ns3/spectrum-wifi-helper.h"
#include "ns3/ssid.h"
#include "ns3/stats-module.h"
#include "ns3/string.h"
#include "ns3/table-based-error-rate-model.h"
#include "ns3/udp-client-server-helper.h"
#include "ns3/uinteger.h"
#include "ns3/wifi-acknowledgment.h"
#include "ns3/wifi-tx-vector.h"
#include "ns3/spectrum-helper.h"
#include "ns3/three-gpp-ftp-m2-helper.h"
#include "ns3/traffic-generator-ngmn-gaming.h"
#include "ns3/traffic-generator-ngmn-video.h"
#include "ns3/traffic-generator-ngmn-voip.h"
#include <iostream>
#include <regex>

```

Figura 75: Código en NS-3 – listado de includes

A continuación, se declara el **espacio de nombres** [115] del programa, como se observa en la Figura 76. Se agrupan todas las declaraciones relativas a NS-3 en un ámbito fuera del espacio de nombres global y se incluye para introducir el espacio de nombres en la región declarativa actual, es decir, en la región global.

```

using namespace ns3;
////////////////////////////////////
// LOG COMPONENT DEFINITION
NS_LOG_COMPONENT_DEFINE("Wifi-Scenario-X"); // Logging Keyword definition

```

Figura 76: Código en NS-3 – espacio de nombres y componente de log

En la misma figura se incluye la declaración del **componente de logging**, que permite activar o desactivar mensajes de log en consola en base a la referencia a dicho nombre.

Después, se incluyen **múltiples funciones** que se emplean para encapsular gran parte de la lógica del programa. Entre ellas, se dispone tanto de funciones genéricas para el tratamiento de datos y variables como de procedimientos específicos de NS-3 y la implementación de escenarios Wi-Fi.

Sin embargo, debido al carácter didáctico del presente anexo, se procede a explicar directamente la **función main()** y desglosarla progresivamente, haciendo mención a las particularidades de los entornos simulados.

En consecuencia, a pesar de existir cuatro scripts dedicados cada uno a un escenario en concreto, y con ánimo de fomentar la reproducción del estudio a futuro sobre nuevos escenarios, se describe un único programa sobre el que se establecen valores genéricos –valores por defecto en su gran mayoría– sobre las variables mostradas en las siguientes figuras.

En cada escenario se asignarán los valores correspondientes obtenidos de la Selección de la Solución para la Parametrización de los entornos WLAN, y el desarrollo de la explicación de las funciones incluirá los comentarios que resulten necesarios para que la reproducción de todos los scripts sea sencilla e intuitiva. Por tanto, en las ocasiones en las que varíe la programación entre escenarios, se asegurará la inclusión de todos los modelos empleados para cada entorno.

16.2.2. Función Principal: Definición Inicial y Ajuste de Parámetros

Para comenzar, la Figura 77 muestra la definición de las **primeras variables** que se utilizan de forma repetida a lo largo del programa. En este caso se tratan de los casos más generales, como el tiempo de simulación, la semilla aleatoria y ejecución para la aleatorización de datos, los activadores de logs y ficheros pcap, o el número de STAs y APs por red.

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// MAIN
int
main(int argc, char* argv[])
{
    //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
    // ENVIRONMENT SETTINGS - DEFAULT PARAMETERS
    //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
    // Simulation basic parameters
    uint32_t simulationTime = 10;          // Simulation duration in seconds
    uint32_t seedNumber = 0;              // RNG seed
    uint32_t runNumber = 0;               // Run number

    // Logging and tracing
    bool verbose = false;                 // Turn on all Wifi logging
    bool tracing = false;                 // Turn on PCAP tracing on APs

    // Network Settings
    uint32_t nNetwork = 1;                 // Number of Networks
    uint32_t nStaA = 1;                    // Number of STAs of network A
    uint32_t nStaB = 1;                    // Number of STAs of network B
    | | | | |                               // ...
    uint32_t nAp = 1;                       // Number of APs
}

```

Figura 77: Código en NS-3 – definición de variables (I)

La propuesta de escenarios realizada tan solo contempla un único AP por red, y un número variables de STAs en cada una de las redes.

Después, la Figura 78 recoge los aspectos físicos del escenario, incluyendo variables a emplear en el cálculo de pérdidas de propagación y parámetros específicos de modelos de interior, así como las dimensiones de los espacios que cubre cada red Wi-Fi.

A continuación, en la misma figura se definen las características de las antenas en términos de potencia, número de antenas y configuraciones MIMO. Cabe destacar que debido a la posibilidad de introducir potencias variables [116] en el código de NS-3 para un modelado más realista –control de potencia, manejo de energía, interferencia, etc.– se introducen una serie de niveles en base a una potencia mínima y máxima que se extraerá también de la Selección de la Solución para la Parametrización de los entornos WLAN.

```

// Physical aspects of the environment
double lossScenario = 1;           // Loss scenario selection
double distance = 1;               // (Scenario specific) Distance in meters between AP and STA
double nFloors = 1;                // (Scenario specific) Propagation loss object
double nWalls = 1;                 // (Scenario specific) Propagation loss object
double lossFrequencyReference = 0; // Frequency reference for loss calculation
double minXSize = 5.0;             // Lower bound for X size of place definition
double maxXSize = 10.0;            // Upper bound for X size of place definition
double minYSize = 5.0;             // Lower bound for Y size of place definition
double maxYSize = 10.0;            // Upper bound for Y size of place definition
double minZSize = 1.0;             // Lower bound for Z size of place definition
double maxZSize = 2.0;             // Upper bound for Z size of place definition

// Transmission Configuration
double txPowerAp = 20;              // Max TX Power for APs
double txPowerSta = 20;             // Max TX Power for STAs
double txPowerMinAp = 10;           // Min TX Power reference for APs and STAs
double txPowerMinSta = 10;          // Min TX Power reference for APs and STAs

uint32_t nTxPowerLevelsAp =
  (uint32_t)(txPowerAp - txPowerMinAp); // Iteration levels for TX power
uint32_t nTxPowerLevelsSta =
  (uint32_t)(txPowerSta - txPowerMinSta); // Iteration levels for TX power

uint32_t nAntennasAp = 2;           // Number of antennas for APs
uint32_t nAntennasSta = 1;         // Number of antennas for STAs

uint32_t nTxSpatialStreamsAp = 2;   // Number of TX Spatial Streams for APs
uint32_t nTxSpatialStreamsSta = 1;  // Number of TX Spatial Streams for STAs
uint32_t nRxSpatialStreamsAp = 2;   // Number of RX Spatial Streams for APs
uint32_t nRxSpatialStreamsSta = 1;  // Number of RX Spatial Streams for STAs
  
```

Figura 78: Código en NS-3 – definición de variables (II)

Una vez definido lo anterior, la Figura 79 recoge la definición de las últimas variables generales utilizadas a lo largo del programa. El resto de las variables se definirán de forma específica a cada escenario –como pueden ser el direccionamiento, las configuraciones específicas de los servicios, la definición de puertos a emplear por servicio o la selección de STAs para cada tipo de tráfico–.

Cabe mencionar que una parte de las variables, como ya ocurría con la generación de logs y ficheros pcap, tan solo contienen valores booleanos que activan y desactivan funcionalidades. Al igual que con el resto de los parámetros, esto puede incluir su activación en función del escenario concreto.

Después de finalizar la inicialización de variables, en la Figura 80 se incluye la asignación de valores a las variables a través de **línea de comandos** para los parámetros generales y de definición general del entorno.

```

// Phy parameters
double frequency = 5; // 2.4 GHz / 5 GHz / 6 GHz
int32_t mcs = -1; // -1 indicates unset value
uint32_t gi = 800; // Guard interval (ns)
uint32_t channelWidth = 80; // Channel Width (MHz)
std::string phyModel{"Spectrum"}; // Phy level model (Yans/Spectrum)
bool enableObsPd = true; // Enable OBSS/PD
double obsPdThreshold = -82.0; // OBSS/PD threshold in dBm
double ccaEdTrSta = -82; // CCA-ED Threshold of STAs in dBm
double ccaEdTrAp = -82; // CCA-ED Threshold of APs in dBm

// Mac parameters
std::string dlAckSeqType{"NO-OFDMA"}; // Type of acknowledgment sequence for DL MU PPDUs of DL
// OFDMA (NO-OFDMA, ACK-SU-FORMAT, MU-BAR, AGGR-MU-BAR)
bool enableUlofdma = false; // In case of OFDMA: enable UL
bool enableBsrp = false; // In case of OFDMA: enable BSRP
Time accessReqInterval = MilliSeconds(100); // ARI value for OFDMA

// Traffic configuration
bool useRts = false; // RTS / CTS usage
bool useExtendedBlockAck = false; // MPDU Long Buffer Size (256 for extended / 64 for normal size)
uint32_t payloadSize = 1448; // Payload Size (bytes)
uint32_t dataRate = 100000000; // Data Rate (bps)
  
```

Figura 79: Código en NS-3 – definición de variables (III)

De esta forma, a pesar de que los valores de inicialización se definan en el propio código, se provee de flexibilidad al permitir ejecutar el mismo con parámetros directamente pasados mediante la línea de comandos que modifiquen el comportamiento del programa.

```

/////////////////////////////////////////////////////////////////
// COMMAND LINE PARAMETERS
/////////////////////////////////////////////////////////////////
CommandLine cmd(_FILE_);

// Simulation basic parameters
cmd.AddValue("simulationTime", "Simulation time (seconds)", simulationTime);
cmd.AddValue("seedNumber", "RNG seed number", seedNumber);
cmd.AddValue("runNumber", "Simulation run number", runNumber);

// Logging and tracing
cmd.AddValue("verbose", "Enable log components", verbose);
cmd.AddValue("tracing", "Enable pcap tracing", tracing);

// Network Settings
cmd.AddValue("nNetwork", "Number of wifi Networks", nNetwork);
cmd.AddValue("nStaA", "Max number of wifi STA devices for AP A", nStaA);
cmd.AddValue("nStaB", "Max number of wifi STA devices for AP B", nStaB);
| | | | //...

// Physical aspects of the environment
cmd.AddValue("lossScenario", "Loss scenario selection", lossScenario);
cmd.AddValue("distance", "Distance (meters) between STA and AP for propagation loss calculation", distance);
cmd.AddValue("nFloors", "Number of floors to be considered in propagation loss calculation", nFloors);
cmd.AddValue("nWalls", "Number of walls to be considered in propagation loss calculation", nWalls);
cmd.AddValue("lossFrequencyReference", "Loss frequency reference for propagation loss calculation", lossFrequencyReference);
cmd.AddValue("minXSize", "Lower bound for X size of place definition", minXSize);
cmd.AddValue("maxXSize", "Upper bound for X size of place definition", maxXSize);
cmd.AddValue("minYSize", "Lower bound for Y size of place definition", minYSize);
cmd.AddValue("maxYSize", "Upper bound for Y size of place definition", maxYSize);
cmd.AddValue("minZSize", "Lower bound for Z size of place definition", minZSize);
cmd.AddValue("maxZSize", "Upper bound for Z size of place definition", maxZSize);
  
```

Figura 80: Código en NS-3 – introducción de variables por la línea de comandos (I)

Del mismo modo, se añade la posibilidad de introducir como entrada por consola los parámetros de configuración de la transmisión y variables específicas de la tecnología, como se recoge en la Figura 81.

Finalmente, se procesan los valores recibidos a través de los argumentos que recibe la función principal en la ejecución del programa desde la línea de comandos.

```

// Transmission Configuration
cmd.AddValue("txPowerAp", "AP transmission power in dBm", txPowerAp);
cmd.AddValue("txPowerSta", "STA transmission power in dBm", txPowerSta);
cmd.AddValue("txPowerMinAp", "AP minimum transmission power in dBm", txPowerMinAp);
cmd.AddValue("txPowerMinSta", "STA minimum transmission power in dBm", txPowerMinSta);
cmd.AddValue("nAntennasAp", "Number of antennas per AP", nAntennasAp);
cmd.AddValue("nAntennasSta", "Number of antennas per STA", nAntennasSta);
cmd.AddValue("ntxSpatialStreamsAp", "Number of TX Spatial Streams per AP", ntxSpatialStreamsAp);
cmd.AddValue("ntxSpatialStreamsSta", "Number of TX Spatial Streams per STA", ntxSpatialStreamsSta);
cmd.AddValue("nrxSpatialStreamsAp", "Number of RX Spatial Streams per AP", nrxSpatialStreamsAp);
cmd.AddValue("nrxSpatialStreamsSta", "Number of RX Spatial Streams per STA", nrxSpatialStreamsSta);

// Phy parameters
cmd.AddValue("frequency", "Whether working in the 2.4, 5 or 6 GHz band (other values gets rejected)", frequency);
cmd.AddValue("mcs", "If set, limit testing to a specific MCS (0-11)", mcs);
cmd.AddValue("gi", "Guard Interval (3200/1600/800)", gi);
cmd.AddValue("channelWidth", "Channel Width (20/40/80/160MHz)", channelWidth);
cmd.AddValue("phyModel", "PHY Model (Yans / Spectrum)", phyModel);
cmd.AddValue("enableObssPd", "Enable/disable OBSS_PD", enableObssPd);
cmd.AddValue("obssPdThreshold", "OBSS PD Threshold (dBm)", obssPdThreshold);
cmd.AddValue("ccaEdTrSta", "CCA ED Threshold for STAs (dBm)", ccaEdTrSta);
cmd.AddValue("ccaEdTrAp", "OBSS PD Threshold for APs (dBm)", ccaEdTrAp);

// Mac parameters
cmd.AddValue("dLackType", "Ack sequence type for DL OFDMA (NO-OFDMA, ACK-SU-FORMAT, MU-BAR, AGGR-MU-BAR)", dLackSeqType);
cmd.AddValue("enableUlofdma", "Enable UL OFDMA (useful if DL OFDMA is enabled and TCP is used)", enableUlofdma);
cmd.AddValue("enableBsrp", "Enable BSRP (useful if DL and UL OFDMA are enabled and TCP is used)", enableBsrp);
cmd.AddValue("accessReqInterval", "ARI value (milliseconds) for OFDMA", accessReqInterval);

// Traffic configuration
cmd.AddValue("useRts", "Enable RTS/CTS", useRts);
cmd.AddValue("useExtendedBlockAck", "MPDU Long Buffer Size (256 for extended / 64 for normal size)", useExtendedBlockAck);
cmd.AddValue("payloadSize", "The application payload size in bytes", payloadSize);
cmd.AddValue("dataRate", "Data rate (bps)", dataRate);

cmd.Parse(argc, argv);
  
```

Figura 81: Código en NS-3 – introducción de variables por la línea de comandos (II)

Después de definir las variables generales a emplear en la función principal, se realizan algunos **ajustes iniciales** [116] derivados de la tecnología a implementar en el simulador.

```

////////////////////////////////////
// ENVIRONMENT SETTINGS - FIXES AND ARRANGEMENTS
////////////////////////////////////
// Enable RTS/CTS
if (useRts)
{
    Config::SetDefault("ns3::WifiRemoteStationManager::RtsCtsThreshold", StringValue("0"));
}

// Set + check type of acknowledgment sequence
if (dLackSeqType == "ACK-SU-FORMAT")
{
    Config::SetDefault("ns3::WifiDefaultAckManager::DLAckSeqType",
        EnumValue(WifiAcknowledgment::DL_MU_BAR_BA_SEQUENCE));
}
else if (dLackSeqType == "MU-BAR")
{
    Config::SetDefault("ns3::WifiDefaultAckManager::DLAckSeqType",
        EnumValue(WifiAcknowledgment::DL_MU_TF_MU_BAR));
}
else if (dLackSeqType == "AGGR-MU-BAR")
{
    Config::SetDefault("ns3::WifiDefaultAckManager::DLAckSeqType",
        EnumValue(WifiAcknowledgment::DL_MU_AGGREGATE_TF));
}
else if (dLackSeqType != "NO-OFDMA")
{
    NS_ABORT_MSG("Invalid DL ack sequence type (must be NO-OFDMA, ACK-SU-FORMAT, MU-BAR or "
        "AGGR-MU-BAR)");
}
  
```

Figura 82: Código en NS-3 – ajustes iniciales (I)

Se establecen así condicionantes sobre el uso de procedimientos (Figura 82) como RTS/CTS o OFDMA para una correcta simulación, y se asegura el cumplimiento de los requisitos pertinentes.

En concreto, un ejemplo de ello es el uso del modelo PHY Spectrum de NS-3, ya que el modelo básico Yans es insuficiente para simular OFDMA [116] o interacciones a nivel de señal entre redes colindantes.

```
// Check Phy Level model
if (phyModel != "Spectrum")
{
    NS_ABORT_MSG("Invalid PHY model (must be Spectrum)");
}

// Set compatible Phy in case of OFDMA
if (dLackSeqType != "NO-OFDMA")
{
    phyModel = "Spectrum"; // SpectrumWifiPhy is required for OFDMA
}

// Set TCP Segment Size
Config::SetDefault("ns3::TcpSocket::SegmentSize", UIntegerValue(payloadSize));

// Enable RNG Custom seed and run
RngSeedManager::SetSeed(seedNumber);
RngSeedManager::SetRun(runNumber);
```

Figura 83: Código en NS-3 – ajustes iniciales (II)

16.2.3. Función Principal: Configuración de Red

Una vez finalizados los ajustes iniciales, comienza la implementación del escenario. En primer lugar, como en la mayoría de los scripts de NS-3 [115] se definen los nodos y los dispositivos de red (Figura 84).

```
////////////////////////////////////
// Node Creation
////////////////////////////////////
NodeContainer wifiStaNodesA;
NodeContainer wifiApNodesA;
NodeContainer wifiStaNodesB;
NodeContainer wifiApNodesB;
NodeContainer wifiStaNodesC;
NodeContainer wifiApNodesC;
NodeContainer wifiStaNodesD;
NodeContainer wifiApNodesD;

CreateNodesNormal(wifiStaNodesA, wifiApNodesA, nStaA, nAp, "A", nStaA, 15);
if (nNetwork >= 2)
|   CreateNodesNormal(wifiStaNodesB, wifiApNodesB, nStaB, nAp, "B", nStaB, 15);
//...

////////////////////////////////////
// NetDevice Creation
////////////////////////////////////
NetDeviceContainer apDevicesA, staDevicesA;
NetDeviceContainer apDevicesB, staDevicesB;
//...
```

Figura 84: Código en NS-3 – definición de nodos y dispositivos de red

Para mantener un control separado de los dispositivos y facilitar el manejo en cada red, se almacenan de forma separada en objetos NodeContainer y NetDeviceContainer.

Los nodos se generan según una función específica (Figura 85) que emplea una distribución normal con los parámetros especificados de forma que varíe para cada escenario en torno a las STAs a definir en cada caso.

Se utiliza una generación de nodos según dicha distribución para proveer de aleatoriedad al escenario, de forma que ronde los valores de referencia –usando dicho valor como la media y estableciendo una varianza que no sea decisiva para la diferenciación de la densidad de dispositivos– establecidos en la Selección de la Solución para la Parametrización de los entornos WLAN.

```

void CreateNodesNormal(NodeContainer& wifiStaNodes,
                      NodeContainer& wifiApNodes,
                      uint32_t nSta,
                      const uint32_t nAp,
                      const std::string& label,
                      const double mean,
                      const double variance)
{
    Ptr<NormalRandomVariable> n = CreateObject<NormalRandomVariable>();
    n->SetAttribute("Mean", DoubleValue(mean));
    n->SetAttribute("Variance", DoubleValue(variance));

    double value = n->GetValue();
    nSta = static_cast<uint32_t>(std::max(1.0, value)); // Ensure at least 1 STA

    wifiStaNodes.Create(nSta);
    wifiApNodes.Create(nAp);
}
  
```

Figura 85: Código en NS-3 – función de generación de nodos

Seguido, se declaran e instancian los objetos de soporte para Wi-Fi, y se habilitan, en caso de establecerse como valores booleanos verdaderos, la generación de logs y el uso de OBSS-PD (Figura 86). Asimismo, se define la versión del estándar como IEEE 802.11ax, de forma que se habilitan las funcionalidades [116] correspondientes a dicho protocolo: modulaciones y esquemas de codificación, OFDMA y MU-MIMO, eficiencia espectral, configuración de canal, etc.

```

////////////////////////////////////
// WIFI (and WIFI MAC) HELPERS
////////////////////////////////////
// Wifi Helper
WifiHelper wifi;
Ssid ssid;

// Wifi Logging
if (verbose)
{
    wifi.EnableLogComponents();
}

// Wifi Standard
wifi.SetStandard(WIFI_STANDARD_80211ax);

if (enableObssPd)
{
    wifi.SetObssPdAlgorithm("ns3::ConstantObssPdAlgorithm", "ObssPdLevel",
                          DoubleValue(obssPdThreshold));
}

// Mac Helper
WifiMacHelper mac;
  
```

Figura 86: Código en NS-3 – instanciación de objetos de soporte de Wi-Fi y primeras configuraciones

Después, se realiza la llamada a la función para la configuración completa de la red Wi-Fi (Figura 87). Como se persigue que se puedan generar escenarios más complejos a futuro o reproducción de entornos específicos, la definición se hace de forma separada y progresiva para cada red, llamando a una función que realiza la gran mayoría de ajustes a nivel PHY y MAC.

La función de configuración de red Wi-Fi tiene diferentes variantes para los escenarios en estudio. A continuación, se presenta una de ellas, de forma que se vaya desglosando la lógica empleada para la configuración de red.


```

////////////////////////////////////
// NETWORK CONFIGURATION: PHY + MAC
////////////////////////////////////

// Network configuration: ssid, frequency (band, channel size and number)...
if (nNetwork >= 1)
{
    ssid = Ssid("ns3-802.11ax-A");
    ConfigureWifiNetwork(seedNumber, runNumber, tracing, wifi, mac, frequency, mcs, channelWidth, gi, nFloors, nWalls,
        nAntennasAp, nAntennasSta, ntxSpatialStreamsAp, ntxSpatialStreamsSta, nrxSpatialStreamsAp,
        nrxSpatialStreamsSta, txPowerMinSta, txPowerSta, nTxPowerLevelsSta, ccaEdTrSta, txPowerMinAp,
        txPowerAp, nTxPowerLevelsAp, ccaEdTrAp, dLackSeqType, enableUlofdma, useExtendedBlockAck, enableBsrp,
        accessReqInterval, ssid, 0, wifiStaNodesA, wifiApNodesA, staDevicesA, apDevicesA);
}

if (nNetwork >= 2)
{
    ssid = Ssid("ns3-802.11ax-B");
    ConfigureWifiNetwork(seedNumber, runNumber, tracing, wifi, mac, frequency, mcs, channelWidth, gi, nFloors, nWalls,
        nAntennasAp, nAntennasSta, ntxSpatialStreamsAp, ntxSpatialStreamsSta, nrxSpatialStreamsAp,
        nrxSpatialStreamsSta, txPowerMinSta, txPowerSta, nTxPowerLevelsSta, ccaEdTrSta, txPowerMinAp,
        txPowerAp, nTxPowerLevelsAp, ccaEdTrAp, dLackSeqType, enableUlofdma, useExtendedBlockAck, enableBsrp,
        accessReqInterval, ssid, 1, wifiStaNodesB, wifiApNodesB, staDevicesB, apDevicesB);
}

//...

```

Figura 87: Código en NS-3 – llamadas a la función de configuración de red

La Figura 88 muestra el comienzo de la función de configuración de red. Como se puede apreciar, inicialmente se instancia un objeto SpectrumChannelHelper en el que apoyarse para realizar la configuración.

```

void ConfigureWifiNetwork(uint32_t seedNumber, uint32_t runNumber, bool tracing,
    WifiHelper wifi, WifiMacHelper mac, double frequency,
    int mcs, uint32_t channelWidth, uint32_t gi,
    uint32_t nFloors, uint32_t nWalls, uint32_t nAntennasAp,
    uint32_t nAntennasSta, uint32_t ntxSpatialStreamsAp,
    uint32_t ntxSpatialStreamsSta, uint32_t nrxSpatialStreamsAp,
    uint32_t nrxSpatialStreamsSta, double txPowerMinSta,
    double txPowerSta, uint32_t nTxPowerLevelsSta,
    double ccaEdTrSta, double txPowerMinAp, double txPowerAp,
    uint32_t nTxPowerLevelsAp, double ccaEdTrAp,
    std::string dLackSeqType, bool enableUlofdma,
    bool useExtendedBlockAck, bool enableBsrp,
    Time accessReqInterval, Ssid ssid, uint32_t staGroupIndex,
    NodeContainer wifiStaNodes, NodeContainer wifiApNodes,
    NetDeviceContainer &staDevices, NetDeviceContainer &apDevices)
{
    // Network configuration: ssid, frequency (band, channel size and number)...
    SpectrumChannelHelper channelHelper;
    std::string channelStr("");
    setChannelInfo(staGroupIndex, frequency, channelWidth, channelStr);
}

```

Figura 88: Código en NS-3 – función de configuración de red (I)

Seguido, se llama a la función setChannelInfo, definida según la Figura 89. Para ello, se le pasan como parámetros la información fundamental: índice de la red, banda de frecuencia, ancho de canal y la cadena de caracteres a utilizar para la generación del código que establece la configuración del canal.

Dentro de la función, se comprueba que la información de frecuencia sea correcta, y se selecciona un número de canal según otra función de definición propia. En este caso, se basa en la selección aleatoria del canal en base a los canales disponibles para las distintas frecuencias.

Acto seguido, se forma la cadena de caracteres con la información del número de canal, el ancho de canal, la banda de frecuencia y el índice del canal primario.

```
void setChannelInfo(int nNetwork, double frequency, int channelWidth, std::string& channelStr)
{
    int32_t channelNumber = 0;

    if (frequency == 6 || frequency == 5 || frequency == 2.4)
    {
        channelNumber = getRandomChannelNumber(channelWidth, frequency);
        if (channelNumber == -1)
        {
            std::cout << "Wrong frequency value or channel width!" << std::endl;
            return;
        }

        std::ostringstream oss;
        oss << "{" << channelNumber << ", " << channelWidth;

        if (frequency == 6)
        {
            oss << ", BAND_6GHZ, 0";
        }
        else if (frequency == 5)
        {
            oss << ", BAND_5GHZ, 0";
        }
        else if (frequency == 2.4)
        {
            oss << ", BAND_2_4GHZ, 0";
        }

        oss << "}";
        channelStr += oss.str();
    }
    else
    {
        std::cout << "Wrong frequency value!" << std::endl;
        return;
    }
}
```

Figura 89: Código en NS-3 – función de configuración de canal

En este caso, el cálculo del canal a emplear viene definido por la Figura 90 para el caso de la banda de frecuencias de 6 GHz, y por la Figura 91 para las bandas de 5 GHz y 2,4 GHz.

```
int32_t getRandomChannelNumber(int channelWidth, double frequency)
{
    ns3::Ptr<ns3::UniformRandomVariable> randomSelector =
    ns3::CreateObject<ns3::UniformRandomVariable>();
    randomSelector->SetAttribute("Min", ns3::DoubleValue(0));
    randomSelector->SetAttribute("Max", ns3::DoubleValue(2));

    if (frequency == 6)
    {
        if (channelWidth == 20)
        {
            return 1 + 4 * randomSelector->GetInteger();
        }
        else if (channelWidth == 40)
        {
            return 3 + 8 * randomSelector->GetInteger();
        }
        else if (channelWidth == 80)
        {
            return 7 + 16 * randomSelector->GetInteger();
        }
        else if (channelWidth == 160)
        {
            return 15 + 32 * randomSelector->GetInteger();
        }
    }
}
```

Figura 90: Código en NS-3 – selección aleatoria de canal (!)

Al comienzo de la Figura 90 se puede observar que se emplea una variable aleatoria uniforme para la selección del canal. Esto se define de dicha forma porque no existe ninguna preferencia, a priori, para la selección de un canal u otro. Se establecen los márgenes de forma que se asegure que en ningún caso se exceda los números de canal propios de cada banda de frecuencias. Se delega a futuras mejoras un mayor ajuste de

los canales de forma que se permita emplear la totalidad de los posibles canales en cada banda.

```

else if (frequency == 5)
{
    if (channelWidth == 20)
    {
        return 36 + 4 * randomSelector->GetInteger();
    }
    else if (channelWidth == 40)
    {
        return 38 + 8 * randomSelector->GetInteger();
    }
    else if (channelWidth == 80)
    {
        return 106 + 16 * randomSelector->GetInteger();
    }
    else if (channelWidth == 160)
    {
        return 50 + 64 * randomSelector->GetInteger();
    }
}
else if (frequency == 2.4 && channelWidth == 20)
{
    return 1 + 5 * randomSelector->GetInteger();
}
return -1; // Invalid channelWidth or frequency
}

```

Figura 91: Código en NS-3 – selección aleatoria de canal (II)

Por último, si los datos de entrada no se ajustan a lo establecido, se devuelve un -1 para indicar dicho error.

Si se varía ligeramente el código anterior se pueden obtener modelos alternativos para la selección de canales en las distintas redes, como puede ser el uso de canales distintos predefinidos por red o la compartición de canal para todas las redes.

```

int32_t getChannelNumber(int channelWidth, double frequency, int nNetwork)
{
    if (frequency == 6)
    {
        if (channelWidth == 20)
        {
            return 1 + 4 * (nNetwork - 1);
        }
        else if (channelWidth == 40)
        {
            return 3 + 8 * (nNetwork - 1);
        }
        else if (channelWidth == 80)
        {
            return 7 + 16 * (nNetwork - 1);
        }
        else if (channelWidth == 160)
        {
            return 15 + 32 * (nNetwork - 1);
        }
    }
}

```

Figura 92: Código en NS-3 – selección determinista de canal (I)

Esto se puede apreciar claramente en la Figura 92 y Figura 93, que sirven a dichos propósitos. Si en la función que realiza la llamada al código de las figuras en lugar de pasar un valor que varía entre las distintas redes se utiliza un valor fijo, la selección de canal será la misma para todas las redes. Sin embargo, si se usa un índice en función de la red, será determinista según la función de selección de canal.

De este modo, con tan solo una ligera variación en distintos lugares de la función, se pueden ejecutar distintos modelos de selección que permitan caracterizar entornos con casos controlados de un modo u otro.

```

else if (frequency == 5)
{
    if (channelWidth == 20)
    {
        return 36 + 4 * (nNetwork - 1);
    }
    else if (channelWidth == 40)
    {
        return 38 + 8 * (nNetwork - 1);
    }
    else if (channelWidth == 80)
    {
        if (nNetwork <= 1)
        {
            return 42;
        }
        else if (nNetwork == 2)
        {
            return 106;
        }
        else if (nNetwork == 3)
        {
            return 122;
        }
    }
    else if (channelWidth == 160)
    {
        return 50 + 64 * (nNetwork - 1);
    }
}
else if (frequency == 2.4 && channelWidth == 20)
{
    return 1 + 5 * (nNetwork - 1);
}

return -1; // Invalid channelWidth or frequency
}

```

Figura 93: Código en NS-3 – selección determinista de canal (II)

Para continuar con la función de configuración de red introducida en la Figura 88, en la Figura 94 se muestra la configuración de los aspectos de propagación, esquemas de gestión del estado de las estaciones, el control de tráfico, y la configuración de GI y buffer de MPDUs.

```

// Propagation
channelHelper.SetChannel("ns3::MultiModelSpectrumChannel");
channelHelper.SetPropagationDelay("ns3::ConstantSpeedPropagationDelayModel");
double lossScenario = 40.05 + +20 * log10(frequency / 2.4) + 5 * nWalls +
    pow(18.3 * nFloors, (nFloors + 2) / (nFloors + 1) - 0.46);
channelHelper.AddPropagationLoss("ns3::ThreeLogDistancePropagationLossModel",
    "ReferenceLoss", DoubleValue(lossScenario),
    "Distance0", DoubleValue(1),
    "Exponent0", DoubleValue(2),
    "Distance1", DoubleValue(5),
    "Exponent1", DoubleValue(3.5));

// Station state and data/control traffic generation management
std::ostringstream oss;
oss << "HeMcs" << mcs;
wifi.SetRemoteStationManager("ns3::ConstantRateWifiManager",
    "DataMode", StringValue(oss.str()),
    "ControlMode", StringValue(oss.str()));

// Set guard interval and MPDU buffer size
wifi.ConfigHeOptions("GuardInterval",
    TimeValue(NanoSeconds(gi)),
    "MpdubufferSize", UIntegerValue(useExtendedBlockAck ? 256 : 64),
    "BssColor", UIntegerValue(1));

```

Figura 94: Código en NS-3 – función de configuración de red (II)

Cabe mencionar que por simplicidad y falta de modelos de cálculo de retardos de propagación [116] dentro de los módulos específicos para dicho fin en NS-3 (tan solo se dispone de cálculo a velocidad constante o modelo de retardo aleatorio), se ha definido el retardo de propagación a velocidad constante.

El cálculo de las pérdidas de propagación puede variar según distintos modelos, como en las propuestas recogidas en la Figura 95. En dicho extracto de código se recogen las pérdidas según los modelos propuestos en la Selección de la Solución para la Parametrización de los entornos WLAN.

Los modelos LogDistance y ThreeGppOffice se realizan mediante una implementación con sus valores por defecto. Sin embargo, para emular las pérdidas de interior relativas a entornos domésticos y de alta densidad, se realizan ajustes adicionales.

En el primero se aplica la fórmula de la Ecuación 26, donde el valor f_c corresponde a la banda de frecuencias utilizada en GHz, d al máximo entre la distancia tridimensional y un metro, y F y W al número de pisos y paredes atravesadas, respectivamente.

$$PL(d) = 40,5 + 20 * \log_{10} \frac{f_c}{2,4} + 20 * \log_{10} \min(d, 5) + (d > 5) * 35 * \log_{10} \frac{d}{5} + 18,3 * F^{\frac{F+2}{F+1}-0,46} + 5 * W$$

Ecuación 26: Función de pérdidas de propagación para el entorno doméstico.

En el segundo, en cambio, se adapta la Ecuación 27 a los modelos disponibles en NS-3. El valor f_c corresponde a la banda de frecuencias utilizada en GHz y d al máximo entre la distancia tridimensional y un metro.

$$PL(d) = 40.5 + 20 * \log_{10} \frac{f_c}{2.4} + 20 * \log_{10} \min(d, 10) + (d > 10) * 35 * \log_{10} \frac{d}{10}$$

Ecuación 27: Función de pérdidas de propagación para el entorno interior de alta densidad.

Como se puede observar en la Figura 95, se hace uso del modelo de pérdidas en base a tres distancias para el establecimiento de los diferentes márgenes de distancia sobre los que establecer el valor de las pérdidas de propagación.

```
// Propagation
channelHelper.SetChannel("ns3::MultiModelSpectrumChannel");
channelHelper.SetPropagationDelay("ns3::ConstantSpeedPropagationDelayModel");
channelHelper.AddPropagationLoss("ns3::LogDistancePropagationLossModel");

// Propagation
channelHelper.SetChannel("ns3::MultiModelSpectrumChannel");
channelHelper.SetPropagationDelay("ns3::ConstantSpeedPropagationDelayModel");
channelHelper.AddPropagationLoss("ns3::ThreeGppIndoorOfficePropagationLossModel");

// Propagation
channelHelper.SetChannel("ns3::MultiModelSpectrumChannel");
channelHelper.SetPropagationDelay("ns3::ConstantSpeedPropagationDelayModel");
double lossScenario = 40.05 + 20 * log10(frequency / 2.4);
channelHelper.AddPropagationLoss("ns3::ThreeLogDistancePropagationLossModel",
    "ReferenceLoss", DoubleValue(lossScenario),
    "Distance0", DoubleValue(1),
    "Exponent0", DoubleValue(2),
    "Distance1", DoubleValue(10),
    "Exponent1", DoubleValue(3.5));

// Propagation
channelHelper.SetChannel("ns3::MultiModelSpectrumChannel");
channelHelper.SetPropagationDelay("ns3::ConstantSpeedPropagationDelayModel");
double lossScenario = 40.05 + 20 * log10(frequency / 2.4) + 5 * nWalls +
    pow(18.3 * nFloors, (nFloors + 2) / (nFloors + 1) - 0.46);
channelHelper.AddPropagationLoss("ns3::ThreeLogDistancePropagationLossModel",
    "ReferenceLoss", DoubleValue(lossScenario),
    "Distance0", DoubleValue(1),
    "Exponent0", DoubleValue(2),
    "Distance1", DoubleValue(5),
    "Exponent1", DoubleValue(3.5));
```

Figura 95: Código en NS-3 – alternativas de cálculo de pérdidas de propagación.

Después, se definen las configuraciones a nivel PHY y MAC para STAs y la configuración PHY de los APs, como se puede observar en la Figura 96. Los valores utilizados se corresponden con los definidos para cada escenario, y en caso de incluirse en el código sin haber sido previamente especificados por entorno, adoptan valores por defecto según la documentación de NS-3.

```
// GENERAL - PHY
SpectrumWifiPhyHelper phy;
phy.SetPcapDataLinkType(WifiPhyHelper::DLT_IEEE802_11_RADIO);
phy.SetChannel(channelHelper.Create());
phy.Set("ChannelSettings", StringValue(channelStr));

// STA CONFIGURATION - PHY
phy.Set("TxPowerStart", DoubleValue(txPowerMinSta));
phy.Set("TxPowerEnd", DoubleValue(txPowerSta));
phy.Set("TxPowerLevels", UIntegerValue(nTxPowerLevelsSta));
phy.Set("CcaEdThreshold", DoubleValue(ccaEdTrSta));
phy.Set("RxSensitivity", DoubleValue(-92.0));

phy.Set("Antennas", UIntegerValue(nAntennasSta));
phy.Set("MaxSupportedTxSpatialStreams", UIntegerValue(nTxSpatialStreamsSta));
phy.Set("MaxSupportedRxSpatialStreams", UIntegerValue(nRxSpatialStreamsSta));

// STA CONFIGURATION - MAC
mac.SetType("ns3::StaWifiMac",
            "Ssid", SsidValue(ssid),
            "V0_MaxAmpduSize", UIntegerValue(8388607),
            "BE_MaxAmpduSize", UIntegerValue(8388607),
            "BK_MaxAmpduSize", UIntegerValue(8388607),
            "VI_MaxAmpduSize", UIntegerValue(8388607)
);

staDevices = wifi.Install(phy, mac, wifiStaNodes);

// AP CONFIGURATION - PHY
phy.Set("TxPowerStart", DoubleValue(txPowerMinAp));
phy.Set("TxPowerEnd", DoubleValue(txPowerAp));
phy.Set("TxPowerLevels", UIntegerValue(nTxPowerLevelsAp));
phy.Set("CcaEdThreshold", DoubleValue(ccaEdTrAp));
phy.Set("RxSensitivity", DoubleValue(-92.0));
phy.Set("Antennas", UIntegerValue(nAntennasAp));
phy.Set("MaxSupportedTxSpatialStreams", UIntegerValue(nTxSpatialStreamsAp));
phy.Set("MaxSupportedRxSpatialStreams", UIntegerValue(nRxSpatialStreamsAp));
```

Figura 96: Código en NS-3 – función de configuración de red (III)

A continuación, en la Figura 97 se muestra la configuración MAC de los APs y una vez establecidos los parámetros necesarios, se permite la captura de tráfico en ellos. Asimismo, se asignan flujos de números aleatorios a los dispositivos en la red Wi-Fi. El número de secuencia se incrementa después de dicha asignación para asegurar que los flujos de números aleatorios sean independientes [117] entre diferentes partes de la simulación. De este modo, se asegura que las decisiones aleatorias en la simulación sean independientes y coherentes entre diferentes partes de la red.

```

// AP CONFIGURATION - MAC
if (dAckSeqType != "NO-OFDMA")
{
    mac.SetMultiUserScheduler("ns3::RrMultiUserScheduler",
                              "EnableUlofdma", BooleanValue(enableUlofdma),
                              "EnableBsrp", BooleanValue(enableBsrp),
                              "AccessReqInterval", TimeValue(accessReqInterval));
}

mac.SetType("ns3::ApWifiMac",
            "Ssid", SsidValue(ssid),
            "EnableBeaconJitter", BooleanValue(false),
            "VO_MaxAmpduSize", UIntegerValue(8388607),
            "BE_MaxAmpduSize", UIntegerValue(8388607),
            "BK_MaxAmpduSize", UIntegerValue(8388607),
            "VI_MaxAmpduSize", UIntegerValue(8388607));

apDevices = wifi.Install(phy, mac, wifiApNodes);

// Basic tracing
if (tracing)
{
    phy.SetPcapDataLinkType(WifiPhyHelper::DLT_IEEE802_11_RADIO);
    phy.EnablePcap("Wifi-4-A", apDevices.Get(0)); // traces AP 0
}

// RNG: For independent replications, the most rigorous set up is to define both seed
// (fixed) and run number (incremental)
RngSeedManager::SetSeed(seedNumber); // set randomness seed
RngSeedManager::SetRun(runNumber); // set run number
int64_t streamNumber = 100;
streamNumber += wifi.AssignStreams(apDevices, streamNumber);
streamNumber += wifi.AssignStreams(staDevices, streamNumber);
}
  
```

Figura 97: Código en NS-3 – función de configuración de red (IV)

16.2.4. Función Principal: Movilidad

En términos de movilidad, se hace uso de MobilityHelper para definir el posicionamiento y los modelos de movilidad de APs y STAs. Para ello, primero se han de definir los límites espaciales [117] de los entornos, que variarán según la forma de estos: cuadrada, rectangular, circular, etc.

La Figura 98 recoge la instanciación del objeto de movilidad y la definición de los tamaños máximos en cada eje. Se definen dos máximos en función de X e Y debido a que la totalidad del espacio se reparte en cuatro entornos y los máximos son de cada uno de ellos. Por lo tanto, si se parte de un origen nulo ($X=0, Y=0$), el máximo alcanzable será ($X=2*X_{máx}, Y=2*Y_{máx}$).

```

////////////////////////////////////
// MOBILITY
////////////////////////////////////
// Mobility Helper
MobilityHelper mobility;

std::string xSize = getUniformRandomValueString(minXSize, maxXSize);
std::string xSize2 = std::to_string(2 * std::stod(xSize));
std::string ySize = getUniformRandomValueString(minYSize, maxYSize);
std::string ySize2 = std::to_string(2 * std::stod(ySize));
std::string zSize = getUniformRandomValueString(minZSize, maxZSize);

// STA - Dynamic configuration: Random Walk
mobility.SetPositionAllocator("ns3::RandomRectanglePositionAllocator",
                              "X", StringValue("ns3::UniformRandomVariable[Min=0|Max=" + xSize + "]"),
                              "Y", StringValue("ns3::UniformRandomVariable[Min=0|Max=" + ySize + "]"),
                              "Z", StringValue(zSize));
mobility.SetMobilityModel("ns3::RandomWalk2dMobilityModel",
                           "Mode", StringValue("Time"), "Time", StringValue("3s"),
                           "Speed", StringValue("ns3::ConstantRandomVariable[Constant=1.0]"),
                           "Bounds", StringValue("0|" + xSize + "|0|" + ySize));
mobility.Install(wifiStaNodesA);
  
```

Figura 98: Código en NS-3 – configuración de movilidad (I)

Además, en la misma figura se muestra la configuración de un modelo RandomWalk a una velocidad de 1 m/s (se ha ajustado el valor por defecto ya que se correspondería con velocidades más similares a correr que a andar), y que varía de dirección cada varios segundos.

Por otro lado, en la Figura 99 se muestra la definición de modelos sin movilidad – posicionamiento estático o constante– para STAs y APs.

```

// STA - Constant Position
mobility.SetPositionAllocator("ns3::RandomRectanglePositionAllocator",
    "X", StringValue("ns3::UniformRandomVariable[Min=" + xSize + "|Max=" + xSize2 + "]"),
    "Y", StringValue("ns3::UniformRandomVariable[Min=0|Max=" + ySize + "]"),
    "Z", StringValue(zSize));
mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");
mobility.Install(wifiStaNodesA);

// AP - Constant position
mobility.SetPositionAllocator(
    "ns3::RandomRectanglePositionAllocator",
    "X", StringValue("ns3::UniformRandomVariable[Min=0|Max=" + xSize + "]"),
    "Y", StringValue("ns3::UniformRandomVariable[Min=0|Max=" + ySize + "]"),
    "Z", StringValue("zSize"));
mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");
mobility.Install(wifiApNodesA);
  
```

Figura 99: Código en NS-3 – configuración de movilidad (II)

Para terminar, además de casos donde el posicionamiento viene dado por figuras rectangulares, en la Figura 100 se puede observar la definición de un modelo constante en base a un vector de posicionamiento para el AP, con intención de proporcionar directamente las coordenadas de su ubicación, y el posicionamiento aleatorio de STAs en base a un círculo cuyo centro se define con las coordenadas y se define su área en base a su radio. En la figura se sigue la fórmula basada en el teorema de Pitágoras para calcular el radio máximo de un círculo circunscrito en un rectángulo de lados X e Y.

```

// AP - Constant position
Ptr<ListPositionAllocator> positionAllocA = CreateObject<ListPositionAllocator>();
positionAllocA->Add(Vector(xSize, ySize, zSizeAp));
mobility.SetPositionAllocator(positionAllocA);
mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");
mobility.Install(wifiApNodesA);

// STA - Dynamic configuration: Random Walk
mobility.SetPositionAllocator("ns3::UniformDiscPositionAllocator",
    "rho", DoubleValue(sqrt(pow(xSize, 2) + pow(ySize, 2)) / 2),
    "X", DoubleValue(xSize),
    "Y", DoubleValue(ySize),
    "Z", DoubleValue(zSize));
mobility.SetMobilityModel("ns3::RandomWalk2dMobilityModel",
    "Mode", StringValue("Time"),
    "Time", StringValue("3s"),
    "Speed", StringValue("ns3::ConstantRandomVariable[Constant=1.0]"),
    "Bounds", StringValue("0|" + std::to_string(xSize2) + "|0|" + std::to_string(ySize2)));
mobility.Install(wifiStaNodesA);
  
```

Figura 100: Código en NS-3 – configuración de movilidad (III)

16.2.5. Función Principal: Protocolos de Internet

Antes de definir el tráfico en la red, se debe instalar la pila de protocolos de internet en los dispositivos y asignar el direccionamiento oportuno. Esto se puede observar en la Figura 101.

El direccionamiento se ha realizado usando IPv4 y con direcciones privadas de clase C con CIDR 24.

El experimento actual se basa en flujos que van directamente entre STAs y APs de la misma red, de forma que las estaciones se comunican directamente con el punto de acceso y viceversa. Por lo tanto, no requiere de configuraciones de mayor complejidad.


```

////////////////////////////////////
// INTERNET PROTOCOL STACK
////////////////////////////////////
InternetStackHelper stack;

stack.Install(wifiApNodesA);
stack.Install(wifiStaNodesA);

stack.Install(wifiApNodesB);
stack.Install(wifiStaNodesB);

//...

////////////////////////////////////
// ADDRESSING
////////////////////////////////////
Ipv4AddressHelper address;

address.SetBase("192.168.1.0", "255.255.255.0");
Ipv4InterfaceContainer staNodeInterfacesA;
Ipv4InterfaceContainer apNodeInterfacesA;
staNodeInterfacesA = address.Assign(staDevicesA);
apNodeInterfacesA = address.Assign(apDevicesA);

address.SetBase("192.168.2.0", "255.255.255.0");
Ipv4InterfaceContainer staNodeInterfacesB;
Ipv4InterfaceContainer apNodeInterfacesB;
staNodeInterfacesB = address.Assign(staDevicesB);
apNodeInterfacesB = address.Assign(apDevicesB);

//...

```

Figura 101: Código en NS-3 – configuración de los protocolos de internet y direccionamiento

16.2.6. Función Principal: Aplicaciones y Generación de Tráfico

La mayor complejidad de la definición de escenarios reside en el presente apartado, que trata sobre la generación de tráfico entre nodos. A continuación, se introduce la abstracción empleada para la implementación del tráfico.

Inicialmente, se dispone de un número de STAs y APs conocido por cada red. Sin embargo, para proveer de aleatoriedad según las estadísticas propuestas en la Selección de la Solución para la Parametrización de los entornos WLAN, se procede de la forma siguiente.

Primero **se calcula el número de STAs seleccionados** para participar en el flujo de tráfico (Figura 102) según una variable normal en base a las estadísticas anteriormente mencionadas. De este modo, se sigue la referencia teórica del estudio realizado.

```

uint32_t ComputeNumSelectedSta(uint32_t nSta, double mean, double variance)
{
    double percentage = getNormalRandomValue(mean, variance);

    uint32_t numSelectedSta = round(nSta * percentage);
    if (numSelectedSta == 0)
    {
        numSelectedSta = 1;
    }
    else if (numSelectedSta > nSta)
    {
        numSelectedSta = nSta;
    }
    return numSelectedSta;
}

```

Figura 102: Código en NS-3 – función para el cálculo del número de STAs por tipo de tráfico

Después, se genera un array del tamaño calculado –se delega a futuras ampliaciones y mejoras del proyecto la migración a vectores o asignación dinámica, ya que no es una práctica recomendable definir arrays cuyo tamaño no es conocido en tiempo de compilación– y se procede a inicializar el array (Figura 103) con valores superiores al máximo de STAs de la red como guarda para detectar errores en caso de hacer un recorrido incorrecto.

```
void
FillSelectedSta(uint32_t* array, uint32_t size, uint32_t nStaMax)
{
    for (uint32_t i = 0; i < size; i++)
    {
        array[i] = nStaMax * 2;
    }
}
```

Figura 103: Código en NS-3 – función de inicialización del array de selección de STAs

El array inicializado con anterioridad se modifica a través de la selección de las STAs (Figura 104) de forma aleatoria según una distribución uniforme. De esta forma, se toman todas las STAs disponibles de la red y se seleccionan sin criterio de preferencia, pero asegurando que no se repitan, de forma que por tipo de tráfico no pueda haber solapamiento en un mismo terminal.

```
void selectSta(uint32_t* selectedSta, uint32_t numSelectedSta, uint32_t nStaMin, uint32_t nStaMax) {
    if(numSelectedSta == 1 && nStaMin == 0 && nStaMax == 0) {
        selectedSta[0] = 0;
        return;
    }
    uint32_t staSelectionIndex = 0;
    while (staSelectionIndex < numSelectedSta) {
        uint32_t currentSta = 1;
        Ptr<UniformRandomVariable> newSta = CreateObject<UniformRandomVariable>();
        newSta->SetAttribute("Min", DoubleValue(nStaMin));
        newSta->SetAttribute("Max", DoubleValue(nStaMax + 0.01));
        currentSta = newSta->GetValue();
        uint32_t* lastSelectedSta = selectedSta + numSelectedSta;

        if (lastSelectedSta == std::find(selectedSta, lastSelectedSta, currentSta)) {
            selectedSta[staSelectionIndex] = currentSta;
            staSelectionIndex++;
        }
    }
}
```

Figura 104: Código en NS-3 – función de selección de STAs

A continuación, se desarrolla la implementación de cada tipo de tráfico a través de aplicaciones de NS-3.

Cabe mencionar que la **activación y desactivación de las aplicaciones** (Figura 105) se gestiona de forma similar en la mayoría de los casos, para lo que se ha definido una función específica.

```
void StartStopApplication(ApplicationContainer &clientApplications,
                          ApplicationContainer &serverApplications,
                          const double simulationTime)
{
    clientApplications.Start(Seconds(0.0));
    clientApplications.Stop(Seconds(simulationTime));
    serverApplications.Start(Seconds(0.0));
    serverApplications.Stop(Seconds(simulationTime));
}
```

Figura 105: Código en NS-3 – función de activación y desactivación de aplicaciones

En este experimento se toma como inicio de todo tráfico los 0 segundos, y se finaliza en el tiempo de simulación definido.

En primer lugar, en la Figura 106 se recoge el procedimiento para la simulación del **tráfico VoD**. Como se mencionaba anteriormente, se eligen las STAs participantes en el flujo de tráfico antes de definir la aplicación.

```

////////////////////////////////////
// APPLICATIONS
////////////////////////////////////
// NETWORK A
////////////////////////////////////
// 1. Traffic type: VoD
// 1.1. Random Node Selection
// Minimum Value Selection: STA definition depends on random values
uint32_t nStaAmin = 0; // index from TOTAL NODES
uint32_t nStaAmax = nStaAmin + nStaA - 1; // index from TOTAL NODES

uint32_t numSelectedStaA = ComputeNumSelectedSta(nStaA, percentVoDDL, percentVariance);
uint32_t selectedStaA[numSelectedStaA];
FillSelectedSta(selectedStaA, sizeof(selectedStaA) / sizeof(selectedStaA[0]), nStaAmax);
selectSta(selectedStaA, numSelectedStaA, nStaAmin, nStaAmax);

uint32_t numSelectedStaA1 = ComputeNumSelectedSta(nStaA, percentVoDUL, percentVariance);
uint32_t selectedStaA1[numSelectedStaA1];
FillSelectedSta(selectedStaA1, sizeof(selectedStaA1) / sizeof(selectedStaA1[0]), nStaAmax);
selectSta(selectedStaA1, numSelectedStaA1, nStaAmin, nStaAmax);

// 1.2. VoD Flow: AP --> STA
ApplicationContainer vodServerApplicationsA, vodClientApplicationsA;
ApplicationContainer vodServerApplicationsA1, vodClientApplicationsA1;
uint16_t portUdp = 5050;
uint16_t tosValue = 0xb8;
size_t numStasA = sizeof(selectedStaA) / sizeof(selectedStaA[0]);

VoDServer(vodServerApplicationsA, vodClientApplicationsA, wifiStaNodesA, wifiApNodesA,
          selectedStaA, numStasA, dataRate, payloadSize, portUdp, tosValue);
StartStopApplication(vodClientApplicationsA, vodServerApplicationsA, simulationTime);

VoDClient(vodServerApplicationsA1, vodClientApplicationsA1, wifiStaNodesA, wifiApNodesA,
          selectedStaA, numStasA, dataRate, payloadSize, portUdp, tosValue);
StartStopApplication(vodClientApplicationsA1, vodServerApplicationsA1, simulationTime);

```

Figura 106: Código en NS-3 – implementación del tráfico VoD

En el caso de VoD se hace una implementación directa a través de sockets UDP [66] para simular el tráfico de vídeo. Para ello, se establece un puerto personalizado para poder distinguirlo en las capturas de tráfico de prueba y se establece un valor del tipo de servicio de vídeo.

```

void VoDServer(ApplicationContainer& vodServerApplications,
              ApplicationContainer& vodClientApplications,
              NodeContainer& wifiStaNodes,
              NodeContainer& wifiApNodes,
              uint32_t* selectedStas,
              size_t numStas,
              const uint32_t& dataRate,
              const uint32_t payloadSize,
              uint16_t portUdp,
              const uint16_t tosValue)
{
    for (size_t i = 0; i < numStas; ++i)
    {
        uint32_t staIndex = selectedStas[i];
        auto ipv4 = wifiStaNodes.Get(staIndex)->GetObject<Ipv4>();
        const auto address = ipv4->GetAddress(1, 0).GetLocal();

        InetSocketAddress sinkSocket(address, portUdp + staIndex);
        sinkSocket.SetTos(tosValue);

        OnOffHelper onOffHelper("ns3::UdpSocketFactory", sinkSocket);
        onOffHelper.SetAttribute(
            "OnTime",
            StringValue("ns3::WeibullRandomVariable[Shape=0.8099|Scale=20850]"));
        onOffHelper.SetAttribute("OffTime",
            StringValue("ns3::GammaRandomVariable[Alpha=0.2463|Beta=60.227]"));
        onOffHelper.SetAttribute("DataRate", DataRateValue(dataRate));
        onOffHelper.SetAttribute("PacketSize", UintegerValue(payloadSize));
        onOffHelper.SetAttribute("MaxBytes", UintegerValue(10989173));
        vodServerApplications.Add(onOffHelper.Install(wifiApNodes.Get(0)));

        PacketSinkHelper packetSinkHelper("ns3::UdpSocketFactory", sinkSocket);
        vodClientApplications.Add(packetSinkHelper.Install(wifiStaNodes.Get(staIndex)));
    }
}

```

Figura 107: Código en NS-3 – función de generación de tráfico desde el servidor VoD

Después, se llama a la función definida para la generación de tráfico VoD desde el servidor VoD –descendente, en sentido desde el AP a las STAs–, que se puede observar en la Figura 107. En ella se aprecia la implementación de sockets para el envío de

paquetes según la definición y parámetros de la Selección de la Solución para la Parametrización de los entornos WLAN.

En sentido contrario también se definen flujos de tráfico de vídeo de la misma categoría (BV3), por lo que los modelos estadísticos se repiten, y tan solo varía que, en este caso, el origen son múltiples dispositivos a uno único –el AP– de forma que se simula tráfico en sentido contrario. La función correspondiente se puede observar en la Figura 108.

```
void VoDClient(ApplicationContainer& vodServerApplications,
              ApplicationContainer& vodClientApplications,
              NodeContainer& wifiStaNodes,
              NodeContainer& wifiApNodes,
              uint32_t* selectedStas,
              size_t numStas,
              const uint32_t& dataRate,
              const uint32_t payloadSize,
              uint16_t& portUdp,
              const uint16_t tosValue)
{
    for (size_t i = 0; i < numStas; ++i)
    {
        uint32_t staIndex = selectedStas[i];

        auto ipv4 = wifiApNodes.Get(0)->GetObject<Ipv4>();
        const auto address = ipv4->GetAddress(1, 0).GetLocal();

        InetSocketAddress sinkSocket(address, portUdp + staIndex);
        sinkSocket.SetTos(tosValue);

        OnOffHelper onOffHelper("ns3::UdpSocketFactory", sinkSocket);
        onOffHelper.SetAttribute(
            "OnTime",
            StringValue("ns3::WeibullRandomVariable[Shape=0.8099|Scale=20850]"));
        onOffHelper.SetAttribute("OffTime",
            StringValue("ns3::GammaRandomVariable[Alpha=0.2463|Beta=60.227]"));
        onOffHelper.SetAttribute("DataRate", DataRateValue(dataRate));
        onOffHelper.SetAttribute("PacketSize", UintegerValue(payloadSize));
        onOffHelper.SetAttribute("MaxBytes", UintegerValue(10989173));
        vodServerApplications.Add(onOffHelper.Install(wifiStaNodes.Get(staIndex)));

        PacketSinkHelper packetSinkHelper("ns3::UdpSocketFactory", sinkSocket);
        vodClientApplications.Add(packetSinkHelper.Install(wifiApNodes.Get(0)));
    }
}
```

Figura 108: Código en NS-3 – función de generación de tráfico hacia el servidor VoD

En segundo lugar, el **tráfico HTTP** (Figura 109) se simula mediante la aplicación propia de HTTP [116] de NS-3. Del mismo modo que con el tráfico VoD, se realiza una selección inicial de STAs a participar en los flujos, y se diseña una función que implemente la generación de tráfico deseada.

```
// 2. Traffic type: HTTP
// 2.1. Random Node Selection
uint32_t numSelectedStaA2 = ComputeNumSelectedSta(nStaA, 0.5, percentVariance);
uint32_t selectedStaA2[numSelectedStaA2];
FillSelectedSta(selectedStaA2, sizeof(selectedStaA2) / sizeof(selectedStaA2[0]), nStaAmax);
selectSta(selectedStaA2, numSelectedStaA2, nStaAmin, nStaAmax);

// 2.2. HTTP Flow
ApplicationContainer httpServerAppsA, httpClientAppsA;
HttpServer(httpAppIndexA, httpServerAppsA, httpClientAppsA, wifiStaNodesA,
           wifiApNodesA, selectedStaA2, sizeof(selectedStaA2) / sizeof(uint32_t));
StartStopApplication(httpClientAppsA, httpServerAppsA, simulationTime);
```

Figura 109: Código en NS-3 – implementación del tráfico HTTP

En este caso se hace uso de los recursos de NS-3 ThreeGppHttpServer y ThreeGppHttpClient que permiten simular tráfico HTTP. Se delegan los ajustes a nivel de paquete y gestión de QoS del tráfico a la propia implementación de la aplicación.

En la Figura 110 se puede observar el empleo de los objetos relacionados a la generación de tráfico HTTP y los ajustes según lo definido en la Selección de la Solución para la Parametrización de los entornos WLAN.

```
void HttpServer(ApplicationContainer& httpServerApps,
               ApplicationContainer& httpClientApps,
               NodeContainer& wifiStaNodes,
               NodeContainer& wifiApNodes,
               uint32_t* selectedStas,
               size_t numStas)
{
    uint32_t httpAppIndex = 0;
    for (size_t i = 0; i < numStas; ++i)
    {
        uint32_t staIndex = selectedStas[i];

        auto ipv4 = wifiApNodes.Get(0)->GetObject<Ipv4>();
        const auto serverAddress = ipv4->GetAddress(1, 0).GetLocal();

        ThreeGppHttpServerHelper serverHelper(serverAddress);
        httpServerApps.Add(serverHelper.Install(wifiApNodes.Get(0)));

        Ptr<ThreeGppHttpServer> httpServer =
            httpServerApps.Get(httpAppIndex)->GetObject<ThreeGppHttpServer>();
        PointerValue varPtr;
        httpServer->GetAttribute("Variables", varPtr);
        Ptr<ThreeGppHttpVariables> httpVariables = varPtr.Get<ThreeGppHttpVariables>();
        httpVariables->SetMainObjectSizeMean(10710);
        httpVariables->SetMainObjectSizeStdDev(25032);
        httpVariables->SetEmbeddedObjectSizeMean(7758);
        httpVariables->SetEmbeddedObjectSizeStdDev(126168);
        httpVariables->SetNumOfEmbeddedObjectsScale(2);
        httpVariables->SetNumOfEmbeddedObjectsShape(1.1);
        httpVariables->SetNumOfEmbeddedObjectsMax(53);

        ThreeGppHttpClientHelper clientHelper(serverAddress);
        httpClientApps.Add(clientHelper.Install(wifiStaNodes.Get(staIndex)));

        httpAppIndex++;
    }
}
```

Figura 110: Código en NS-3 – función de generación de tráfico entre servidor y cliente HTTP

Los tipos de tráfico restantes parten de aproximaciones distintas a las dos anteriores. Se ha aprovechado la implementación realizada por CTTC [97] adaptando sus modelos para cumplir con lo establecido en el presente experimento.

```
#ifndef THREE_GPP_FTP_M2_HELPER_H
#define THREE_GPP_FTP_M2_HELPER_H

#include <ns3/internet-apps-module.h>
#include <ns3/internet-module.h>
#include <ns3/network-module.h>
#include <ns3/node.h>
#include <ns3/ping-helper.h>
#include <ns3/traffic-generator-helper.h>

namespace ns3
{
class ThreeGppFtpM2Helper : public Object
{
public:
    ThreeGppFtpM2Helper();
    ThreeGppFtpM2Helper(ApplicationContainer* serverApps,
                        ApplicationContainer* clientApps,
                        NodeContainer* serverNodes,
                        NodeContainer* clientNodes,
                        Ipv4InterfaceContainer* serversIps);
    ~ThreeGppFtpM2Helper() override;
    static TypeId GetTypeId();
    void Configure(uint16_t port,
                  Time serverStartTime,
                  Time clientStartTime,
                  Time clientStopTime,
                  double ftpLambda,
                  double ftpMu,
                  double ftpSigma,
                  double dataRate);
    void Start();

private:
    void DoConfigureFtpServers();
    void DoConfigureFtpClients();
    void DoStartFileTransfer();
    Time DoGetNextTime() const;
    uint32_t m_lastClient{0};
    Ptr<ExponentialRandomVariable> m_ftpArrivals;
    Ptr<UniformRandomVariable> m_startJitter;
    uint16_t m_port{0};
    Time m_clientStartTime{Seconds(0)};
    Time m_clientStopTime{Seconds(0)};
    double m_ftpLambda{0.0};
    double m_ftpMu{0.0};
    double m_ftpSigma{0.0};
    double m_dataRate{0.0};
    double m_transferTime{0.0};
    Ptr<LogNormalRandomVariable> m_ftpFileSize;
    Time m_serverStartTime{Seconds(0)};
    bool m_boolConfigured{false};
    ApplicationContainer* m_serverApps;
    ApplicationContainer* m_clientApps;
    NodeContainer* m_serverNodes;
    NodeContainer* m_clientNodes;
    Ipv4InterfaceContainer* m_serversIps;
    ApplicationContainer m_pingApps;
};
};
#endif
```

Figura 111: Código en NS-3 – fichero de encabezado del generador de tráfico FTP M2

En el caso del **tráfico FTP**, se toma la referencia del modelo M1 de la documentación de la 3GPP e implementado por la CTTC, y se diseña a partir de él un servidor FTP M2 [98] introduciendo tiempos de lectura de los ficheros como se definía en Características Principales de los Tipos de Tráfico y tamaños de fichero variables.

En la Figura 111 se muestra el fichero de encabezado para el generador de tráfico FTP según el modelo M2. En este, se indican los métodos y atributos públicos y privados de la clase ThreeGppFtpM2Helper. El código está en el espacio de nombres ns3.

Sin embargo, resulta más intuitivo analizar directamente el código de la clase, como se hará referenciando los extractos de este en las siguientes figuras. En primer lugar, la generación de objetos ThreeGppFtpM2Helper se gestiona según los constructores y el destructor presentes en la Figura 112. Del mismo modo que con el fichero de encabezado, el código se incluye en el espacio de nombres de ns3.

```

#include "three-gpp-ftp-m2-helper.h"
#include <ns3/packet-sink-helper.h>
#include <ns3/traffic-generator-ftp-single.h>

namespace ns3
{

NS_LOG_COMPONENT_DEFINE("ThreeGppFtpM2Helper");

ThreeGppFtpM2Helper::ThreeGppFtpM2Helper(ApplicationContainer* serverApps,
                                         ApplicationContainer* clientApps,
                                         NodeContainer* serverNodes,
                                         NodeContainer* clientNodes,
                                         Ipv4InterfaceContainer* serversIps)
{
  NS_LOG_FUNCTION(this);
  m_serverApps = serverApps;
  m_clientApps = clientApps;
  m_serverNodes = serverNodes;
  m_clientNodes = clientNodes;
  m_serversIps = serversIps;
}

ThreeGppFtpM2Helper::ThreeGppFtpM2Helper()
{
  NS_LOG_FUNCTION(this);
}

ThreeGppFtpM2Helper::~ThreeGppFtpM2Helper()
{
  NS_LOG_FUNCTION(this);
}

```

Figura 112: Código en NS-3 – fichero de implementación del generador de tráfico FTP M2 (I)

A continuación, en la Figura 113 se pueden observar los métodos de configuración de clientes y servidores. De forma previa, se añade el método específico de NS-3 para la implementación de un tipo de ID para la clase en cuestión, que habilita ciertas características para la gestión de los objetos.

En el caso de los servidores, el procedimiento seguido es similar a la definición de los receptores de paquetes en otros casos, ya que se hace uso de PacketSinkHelper que permite la recepción de paquetes. En los clientes, sin embargo, se hace de forma similar a los nodos que transfieren tráfico, y en este caso se emplea un ayudante definido por la CTTC [97] para la generación de tráfico FTP.

En el presente experimento, se entiende por servidores, según el código definido para FTP, las STAs. Por otro lado, los clientes serán los APs, que son quienes envían –en sentido descendente– los ficheros.

```

TypeId ThreeGppFtpM2Helper::GetTypeId()
{
    static TypeId tid = TypeId("ns3::ThreeGppFtpM2Helper")
        .SetParent<Object>()
        .AddConstructor<ThreeGppFtpM2Helper>();
    return tid;
}

void ThreeGppFtpM2Helper::DoConfigureFtpServers()
{
    NS_LOG_FUNCTION(this);
    Address apLocalAddress(InetSocketAddress(Ipv4Address::GetAny(), m_port));
    PacketSinkHelper packetSinkHelper("ns3::UdpSocketFactory", apLocalAddress);
    *m_serverApps = packetSinkHelper.Install(*m_serverNodes);
    m_serverApps->Start(m_serverStartTime);
}

void ThreeGppFtpM2Helper::DoConfigureFtpClients()
{
    NS_LOG_FUNCTION(this);
    uint32_t ftpSegSize = 1448; // bytes
    TrafficGeneratorHelper ftpHelper("ns3::UdpSocketFactory", Address(),
        TrafficGeneratorFtpSingle::GetTypeId());
    ftpHelper.SetAttribute("PacketSize", UintegerValue(ftpSegSize));
    ftpHelper.SetAttribute("FileSize", UintegerValue(m_ftpFileSize->GetValue()));

    for (uint32_t i = 0; i < m_serversIps->GetN(); i++)
    {
        Ipv4Address ipAddress = m_serversIps->GetAddress(i, 0);
        AddressValue remoteAddress(InetSocketAddress(ipAddress, m_port));
        ftpHelper.SetAttribute("Remote", remoteAddress);
        m_clientApps->Add(ftpHelper.Install(*m_clientNodes));
    }

    m_clientApps->Start(m_clientStartTime + Seconds(m_startJitter->GetValue()));
}

```

Figura 113: Código en NS-3 – fichero de implementación del generador de tráfico FTP M2 (II)

Después, en la Figura 114 se define la función de inicio de transferencia de archivos. Para ello, inicia el envío de tráfico mediante un método basado en ráfagas de paquetes, y programa la siguiente invocación de la función –el siguiente envío– para un momento definido mediante DoGetNextTime.

```

void ThreeGppFtpM2Helper::DoStartFileTransfer()
{
    NS_LOG_FUNCTION(this);
    NS_ASSERT(m_lastClient >= 0 && m_lastClient < m_clientApps->GetN());
    Ptr<Application> app = m_clientApps->Get(m_lastClient);
    NS_ASSERT(app);
    Ptr<TrafficGenerator> fileTransfer = DynamicCast<TrafficGenerator>(app);
    NS_ASSERT(fileTransfer);
    fileTransfer->SendPacketBurst();

    m_lastClient += 1;
    if (m_lastClient == m_clientApps->GetN())
    {
        m_lastClient = 0;
    }
    Simulator::Schedule(DoGetNextTime(), &ThreeGppFtpM2Helper::DoStartFileTransfer, this);
}

```

Figura 114: Código en NS-3 – fichero de implementación del generador de tráfico FTP M2 (III)

Seguido, la Figura 115 define la función de configuración de la transferencia FTP. Se emplean, para ello, el puerto sobre el que establecer las conexiones, los inicios y finales de las aplicaciones en los nodos, las variables estadísticas para la estimación de los tamaños de fichero y el modelado de llegadas según el modelo M2, y la tasa de datos.

Las llegadas se modelan de forma equivalente a M1, pero se introduce retardo –tiempo de lectura definido en la Figura 116– entre la sucesión de envíos de ficheros. Asimismo, el tamaño del fichero se define según lo definido en Tabla 106.

```

void ThreeGppFtpM2Helper::Configure(uint16_t port,
                                     Time serverStartTime,
                                     Time clientStartTime,
                                     Time clientStopTime,
                                     double ftpLambda,
                                     double ftpMu,
                                     double ftpSigma,
                                     double dataRate)
{
    NS_LOG_FUNCTION(this);
    NS_ABORT_MSG_IF(m_boolConfigured, "Already configured FTP M2 helper.");
    NS_ABORT_MSG_IF(m_serverNodes->GetN() == 0 || m_clientNodes->GetN() == 0 ||
                   m_serversIps->GetN() == 0,
                   "Server and/or client nodes or IP server interfaces not set.");
    m_port = port;
    m_clientStartTime = clientStartTime;
    m_clientStopTime = clientStopTime;
    m_ftpLambda = ftpLambda;
    m_ftpMu = ftpMu;
    m_ftpSigma = ftpSigma;
    m_dataRate = dataRate;
    m_serverStartTime = serverStartTime;
    m_boolConfigured = true;

    m_ftpFileSize = CreateObject<LogNormalRandomVariable>();
    m_ftpFileSize->SetAttribute("Mu", DoubleValue(m_ftpMu));
    m_ftpFileSize->SetAttribute("Sigma", DoubleValue(m_ftpSigma));
    m_transferTime = pow(m_dataRate, -1)*m_ftpFileSize->GetValue();

    m_ftpArrivals = CreateObject<ExponentialRandomVariable>();
    m_ftpArrivals->SetAttribute("Mean", DoubleValue(1 / m_ftpLambda));
    // Randomly distribute the start times across 100ms interval
    m_startJitter = CreateObject<UniformRandomVariable>();
    m_startJitter->SetAttribute("Max", DoubleValue(0.100));
}

```

Figura 115: Código en NS-3 – fichero de implementación del generador de tráfico FTP M2 (IV)

Por último, la Figura 116 recoge los métodos de comienzo de las aplicaciones, y la generación de tiempos para nuevos eventos, que incluye el retardo para el siguiente envío de fichero.

```

void ThreeGppFtpM2Helper::Start()
{
    NS_LOG_FUNCTION(this);
    NS_ASSERT(m_boolConfigured);

    DoConfigureFtpServers();
    DoConfigureFtpClients();

    Simulator::Schedule(m_clientStartTime + DoGetNextTime(),
                       &ThreeGppFtpM2Helper::DoStartFileTransfer,
                       this);
}

Time ThreeGppFtpM2Helper::DoGetNextTime() const
{
    return Seconds(m_ftpArrivals->GetValue())+ Seconds(m_transferTime);
}

```

Figura 116: Código en NS-3 – fichero de implementación del generador de tráfico FTP M2 (V)

Una vez hecho esto, se procede de forma similar a los modelos de tráfico anteriores. Tras seleccionar los nodos de forma aleatoria, se genera un contenedor de nodos auxiliar para facilitar el recorrido de índices interno de las funciones.

Este contenedor es utilizado en la instanciación del objeto de soporte ThreeGppFtpM1Helper como se muestra en la Figura 117. Después, se configura según lo definido en las Características Principales de los Tipos de Tráfico, y se inicializa la aplicación en los nodos participantes de los flujos. La selección de puertos se ha realizado de forma no estándar para llevar un seguimiento sencillo de las numeraciones utilizadas, pero puede ser modificado de forma sencilla ajustando a los usados en casos reales.


```

// 3. Traffic type: FTP
// 3.1. Random Node Selection
uint32_t numSelectedStaA3 = ComputeNumSelectedSta(nStaA, 0.2, 0.01);
uint32_t selectedStaA3[numSelectedStaA3];
FillSelectedSta(selectedStaA3, sizeof(selectedStaA3) / sizeof(selectedStaA3[0]), nStaAmax);
selectSta(selectedStaA3, numSelectedStaA3, nStaAmin, nStaAmax);

// 3.2. FTP Flow
uint16_t portFtp = 5150;
double lambdaFtp = 0.2;
double muFtp = 14.45;
double sigmaFtp = 0.35;
ApplicationContainer ftpServersA, ftpClientsA;

NodeContainer ftpStaNodesA;
Ipv4InterfaceContainer ftpStaInterfacesA;

for (uint32_t staIndex : selectedStaA3)
{
  ftpStaNodesA.Add(wifiStaNodesA.Get(staIndex));
  ftpStaInterfacesA.Add(staNodeInterfacesA.Get(staIndex));
}

Ptr<ThreeGppFtpM2Helper> ftpHelperA;
ftpHelperA = CreateObject<ThreeGppFtpM2Helper>(&ftpServersA, &ftpClientsA, &ftpStaNodesA,
                                              &wifiApNodesA, &ftpStaInterfacesA);
ftpHelperA->Configure(portFtp, Seconds(0.0), Seconds(0.0), Seconds(simulationTime),
                    lambdaFtp, muFtp, sigmaFtp, dataRate);

ftpHelperA->Start();
  
```

Figura 117: Código en NS-3 – implementación del tráfico FTP

Como se mencionaba anteriormente, el **tráfico Gaming** también hace uso de recursos provistos por la CTTC, para cuyos objetos de generación de tráfico se han definido dos funciones de encapsulación, una para el cliente (Figura 119) y otra para el servidor (Figura 120), respectivamente.

La Figura 118 muestra la selección de STAs participantes y los parámetros establecidos para este tipo de tráfico junto con las llamadas a las funciones que establecen como fuente y receptores del tráfico tanto STAs como APs. Además, se incluye la llamada de ajuste de los tiempos de inicio y final de las aplicaciones.

```

// 4.2. Gaming Flow:
uint32_t portGaming = 5250;
uint32_t aPacketSizeDL = 330;
uint32_t aPacketSizeUL = 45;
double bPacketSizeDL = 82;
double bPacketSizeUL = 5.7;
double aPacketArrivalDL = 50;
double aPacketArrivalUL = 40;
double bPacketArrivalDL = 4.5;
double bPacketArrivalUL = 6;
uint32_t initArrivalMin = 0;
uint32_t initArrivalMax = 40;
ApplicationContainer gamingServersApA, gamingClientsStaA, gamingServersStaA, gamingClientsApA;
size_t staSizeA = sizeof(selectedStaA4) / sizeof(selectedStaA4[0]);

GamingDL(portGaming, aPacketSizeDL, aPacketSizeUL, bPacketSizeDL, bPacketSizeUL, aPacketArrivalDL,
         aPacketArrivalUL, bPacketArrivalDL, bPacketArrivalUL, initArrivalMin, initArrivalMax,
         gamingServersApA, gamingClientsStaA, gamingServersStaA, gamingClientsApA, wifiApNodesA,
         wifiStaNodesA, selectedStaA4, staSizeA);
GamingUL(portGaming, aPacketSizeDL, aPacketSizeUL, bPacketSizeDL, bPacketSizeUL, aPacketArrivalDL,
         aPacketArrivalUL, bPacketArrivalDL, bPacketArrivalUL, initArrivalMin, initArrivalMax,
         gamingServersApA, gamingClientsStaA, gamingServersStaA, gamingClientsApA, wifiApNodesA,
         wifiStaNodesA, selectedStaA4, staSizeA);

StartStopApplication(gamingClientsStaA, gamingServersApA, simulationTime);
StartStopApplication(gamingClientsApA, gamingServersStaA, simulationTime);
  
```

Figura 118: Código en NS-3 – implementación del tráfico Gaming

De forma más similar a la propuesta para VoD que al tráfico FTP, la función establece un cliente de aplicación que genera tráfico desde cada STA seleccionada al AP, asignando los valores de los parámetros de configuración de tráfico Gaming: puerto específico,

tamaños y tiempos de referencia para la llegada de paquetes, aplicaciones y nodos a utilizar.

En el caso del tráfico en sentido ascendente (Figura 119), la fuente obtiene la dirección IP a la que enviar tráfico, y utiliza un objeto de ayuda de TrafficGeneratorHelper que permite ajustar los parámetros anteriormente mencionados. Después, se instala dicha aplicación en las STAs. Por el contrario, en el AP se instala un PacketSinkHelper para recibir los paquetes.

```

void GamingUl(uint16_t portGaming,
uint32_t aPacketSizeDl, double aPacketSizeUl,
double bPacketSizeDl, double bPacketSizeUl,
double aPacketArrivalDl, double aPacketArrivalUl,
double bPacketArrivalDl, double bPacketArrivalUl,
uint32_t initArrivalMin, uint32_t initArrivalMax,
ApplicationContainer& gamingServersAp, ApplicationContainer& gamingClientsSta,
ApplicationContainer& gamingServersSta, ApplicationContainer& gamingClientsAp,
const NodeContainer& wifiApNodes, const NodeContainer& wifiStaNodes,
uint32_t* selectedSta, size_t staSize)
{
for (size_t i = 0; i < staSize; ++i)
{
uint32_t staIndex = selectedSta[i];

// Source
auto ipv4Server = wifiApNodes.Get(0)->GetObject<Ipv4>();
Ipv4Address ipAddressServer = ipv4Server->GetAddress(1, 0).GetLocal();
AddressValue serverAddress(InetSocketAddress(ipAddressServer, portGaming));

TrafficGeneratorHelper gamingHelper("ns3::UdpSocketFactory", Address(),
TrafficGeneratorNgmnGaming::GetTypeId());

gamingHelper.SetAttribute("IsDownLink", BooleanValue(true));
gamingHelper.SetAttribute("aParamPacketSizeDl", UintegerValue(aPacketSizeDl));
gamingHelper.SetAttribute("bParamPacketSizeDl", DoubleValue(bPacketSizeDl));
gamingHelper.SetAttribute("aParamPacketArrivalDl", DoubleValue(aPacketArrivalDl));
gamingHelper.SetAttribute("bParamPacketArrivalDl", DoubleValue(bPacketArrivalDl));
gamingHelper.SetAttribute("InitialPacketArrivalMin", UintegerValue(initArrivalMin));
gamingHelper.SetAttribute("InitialPacketArrivalMax", UintegerValue(initArrivalMax));
gamingHelper.SetAttribute("Remote", serverAddress);
gamingClientsSta.Add(gamingHelper.Install(wifiStaNodes.Get(staIndex)));
}

// Sink
InetSocketAddress clientAddress(Ipv4Address::GetAny(), portGaming);
PacketSinkHelper packetSinkHelper("ns3::UdpSocketFactory", clientAddress);
gamingServersAp.Add(packetSinkHelper.Install(wifiApNodes.Get(0)));
}
  
```

Figura 119: Código en NS-3 – función de generación de tráfico Gaming en sentido ascendente

En la Selección de la Solución para la Parametrización de los entornos WLAN se define este tipo de tráfico como bidireccional, y siguiendo la propuesta del TGax [72] para el flujo en sentido contrario, se define de la misma manera el tráfico en sentido descendente. Esto se puede observar en la Figura 120.

```

void GamingDl(uint32_t portGaming,
uint32_t aPacketSizeDl, uint32_t aPacketSizeUl,
double bPacketSizeDl, double bPacketSizeUl,
double aPacketArrivalDl, double aPacketArrivalUl,
double bPacketArrivalDl, double bPacketArrivalUl,
uint32_t initArrivalMin, uint32_t initArrivalMax,
ApplicationContainer& gamingServersAp, ApplicationContainer& gamingClientsSta,
ApplicationContainer& gamingServersSta, ApplicationContainer& gamingClientsAp,
NodeContainer& wifiApNodes, NodeContainer& wifiStaNodes,
uint32_t* selectedSta, size_t staSize)
{
for (size_t i = 0; i < staSize; ++i)
{
uint32_t staIndex = selectedSta[i];

// Source
auto ipv4Server = wifiStaNodes.Get(staIndex)->GetObject<Ipv4>();
Ipv4Address ipAddressServer = ipv4Server->GetAddress(1, 0).GetLocal();
AddressValue serverAddress(InetSocketAddress(ipAddressServer, portGaming + 50));

TrafficGeneratorHelper gamingHelper("ns3::UdpSocketFactory",
InetSocketAddress(ipAddressServer, portGaming + 50),
TrafficGeneratorNgmnGaming::GetTypeId());
gamingHelper.SetAttribute("IsDownLink", BooleanValue(true));
gamingHelper.SetAttribute("aParamPacketSizeDl", UintegerValue(aPacketSizeUl));
gamingHelper.SetAttribute("bParamPacketSizeDl", DoubleValue(bPacketSizeUl));
gamingHelper.SetAttribute("aParamPacketArrivalDl", DoubleValue(aPacketArrivalUl));
gamingHelper.SetAttribute("bParamPacketArrivalDl", DoubleValue(bPacketArrivalUl));
gamingHelper.SetAttribute("InitialPacketArrivalMin", UintegerValue(initArrivalMin));
gamingHelper.SetAttribute("InitialPacketArrivalMax", UintegerValue(initArrivalMax));
gamingHelper.SetAttribute("Remote", serverAddress);
gamingClientsAp.Add(gamingHelper.Install(wifiApNodes.Get(0)));

// Sink
InetSocketAddress clientAddress(Ipv4Address::GetAny(), portGaming + 50);
PacketSinkHelper packetSinkHelper("ns3::UdpSocketFactory", clientAddress);
gamingServersSta.Add(packetSinkHelper.Install(wifiStaNodes.Get(staIndex)));
}
}
  
```

Figura 120: Código en NS-3 – función de generación de tráfico Gaming en sentido descendente

En quinto y último lugar, el **tráfico VoIP** se establece siguiendo la misma propuesta que con el tráfico Gaming, cambiando los modelos estadísticos subyacentes al hacer uso de objetos de distinta clase para la generación de paquetes.

En la Figura 121 se recoge la selección de STAs participantes y los parámetros específicos del modelo VoIP junto con las llamadas a las funciones de cliente y servidor. Además, de la misma forma que en casos anteriores, se incluye la llamada de ajuste de los tiempos de inicio y final de las aplicaciones.

```
// 5. Traffic type: VoIP
// 5.1. Random Node Selection
uint32_t numSelectedStaA5 = ComputeNumSelectedSta(nStaA, 0.2, 0.01);
uint32_t selectedStaA5[numSelectedStaA5];
FillSelectedSta(selectedStaA5, sizeof(selectedStaA5) / sizeof(selectedStaA5[0]), nStaAmax);
selectSta(selectedStaA5, numSelectedStaA5, nStaAmin, nStaAmax);

// 5.2. VoIP Flow:
uint32_t portVoIP = 5350;
uint32_t encoderFrameLength = 20;
uint32_t meanTalkSpurtDuration = 1250;
double voiceActivityFactor = 0.5;
uint32_t voicePayload = 40;
uint32_t sidPeriodicity = 160;
uint32_t sidPayload = 15;
ApplicationContainer voIPServersApA, voIPClientsStaA, voIPServersStaA, voIPClientsApA;

VoIPDL(portVoIP, encoderFrameLength, meanTalkSpurtDuration, voiceActivityFactor, voicePayload,
        sidPeriodicity, sidPayload, voIPServersApA, voIPClientsStaA, voIPServersStaA, voIPClientsApA,
        wifiApNodesA, wifiStaNodesA, selectedStaA5, sizeof(selectedStaA5) / sizeof(selectedStaA5[0]));
VoIPUL(portVoIP, encoderFrameLength, meanTalkSpurtDuration, voiceActivityFactor, voicePayload,
        sidPeriodicity, sidPayload, voIPServersApA, voIPClientsStaA, voIPServersStaA, voIPClientsApA,
        wifiApNodesA, wifiStaNodesA, selectedStaA5, sizeof(selectedStaA5) / sizeof(selectedStaA5[0]));

StartStopApplication(voIPClientsStaA, voIPServersApA, simulationTime);
StartStopApplication(voIPClientsApA, voIPServersStaA, simulationTime);
```

Figura 121: Código en NS-3 – implementación del tráfico VoIP

Se definen dos funciones complementarias que gestionan el tráfico en los dos sentidos de la transmisión. La primera establece el tráfico transmitido en la dirección de las STAs al AP. Esto se puede observar en la Figura 122.

```
void VoIPUL(uint32_t portVoIP,
            uint32_t encoderFrameLength, uint32_t meanTalkSpurtDuration,
            double voiceActivityFactor, uint32_t voicePayload,
            uint32_t sidPeriodicity, uint32_t sidPayload,
            ApplicationContainer& voIPServersAp, ApplicationContainer& voIPClientsSta,
            ApplicationContainer& voIPServersSta, ApplicationContainer& voIPClientsAp,
            const NodeContainer& wifiApNodes, const NodeContainer& wifiStaNodes,
            uint32_t* selectedSta, size_t staSize)
{
    for (size_t i = 0; i < staSize; ++i)
    {
        uint32_t staIndex = selectedSta[i];

        // Source
        auto ipv4Server = wifiApNodes.Get(0)->GetObject<Ipv4>();
        Ipv4Address ipAddressServer = ipv4Server->GetAddress(1, 0).GetLocal();
        AddressValue serverAddress(InetSocketAddress(ipAddressServer, portVoIP));

        TrafficGeneratorHelper voIPHelper("ns3::UdpSocketFactory", Address(),
                                         TrafficGeneratorNgmnVoip::GetTypeId());

        voIPHelper.SetAttribute("EncoderFrameLength", UintegerValue(encoderFrameLength));
        voIPHelper.SetAttribute("MeanTalkSpurtDuration", UintegerValue(meanTalkSpurtDuration));
        voIPHelper.SetAttribute("VoiceActivityFactor", DoubleValue(voiceActivityFactor));
        voIPHelper.SetAttribute("VoicePayload", UintegerValue(voicePayload));
        voIPHelper.SetAttribute("SIDPeriodicity", UintegerValue(sidPeriodicity));
        voIPHelper.SetAttribute("SIDPayload", UintegerValue(sidPayload));
        voIPHelper.SetAttribute("Remote", serverAddress);
        voIPClientsSta.Add(voIPHelper.Install(wifiStaNodes.Get(staIndex)));
    }

    // Sink
    InetSocketAddress clientAddress(Ipv4Address::GetAny(), portVoIP);
    PacketSinkHelper packetSinkHelper("ns3::UdpSocketFactory", clientAddress);
    voIPServersAp.Add(packetSinkHelper.Install(wifiApNodes.Get(0)));
}
```

Figura 122: Código en NS-3 – función de generación de tráfico VoIP en sentido ascendente

La segunda (Figura 123), en cambio, establece los flujos de tráfico en sentido contrario, desde el AP a las STA.

```
void VoIPDL(uint32_t portVoIP,
uint32_t encoderFrameLength, uint32_t meanTalkSpurtDuration,
double voiceActivityFactor, uint32_t voicePayload,
uint32_t sidPeriodicity, uint32_t sidPayload,
ApplicationContainer& voIPServersAp, ApplicationContainer& voIPClientsSta,
ApplicationContainer& voIPServersSta, ApplicationContainer& voIPClientsAp,
const NodeContainer& wifiApNodes, const NodeContainer& wifiStaNodes,
uint32_t* selectedSta, size_t staSize)
{
for (size_t i = 0; i < staSize; ++i)
{
uint32_t staIndex = selectedSta[i];

// Source
auto ipv4Server = wifiStaNodes.Get(staIndex)->GetObject<Ipv4>();
Ipv4Address ipAddressServer = ipv4Server->GetAddress(1, 0).GetLocal();
AddressValue serverAddress(InetSocketAddress(ipAddressServer, portVoIP + 50));

TrafficGeneratorHelper voIPHelper("ns3::UdpSocketFactory",
InetSocketAddress(ipAddressServer, portVoIP + 50),
TrafficGeneratorNgmnVoip::GetTypeId());
voIPHelper.SetAttribute("EncoderFrameLength", UintegerValue(encoderFrameLength));
voIPHelper.SetAttribute("MeanTalkSpurtDuration", UintegerValue(meanTalkSpurtDuration));
voIPHelper.SetAttribute("VoiceActivityFactor", DoubleValue(voiceActivityFactor));
voIPHelper.SetAttribute("VoicePayload", UintegerValue(voicePayload));
voIPHelper.SetAttribute("SIDPeriodicity", UintegerValue(sidPeriodicity));
voIPHelper.SetAttribute("SIDPayload", UintegerValue(sidPayload));
voIPHelper.SetAttribute("Remote", serverAddress);
voIPClientsAp.Add(voIPHelper.Install(wifiApNodes.Get(0)));

// Sink
InetSocketAddress clientAddress(Ipv4Address::GetAny(), portVoIP + 50);
PacketSinkHelper packetSinkHelper("ns3::UdpSocketFactory", clientAddress);
voIPServersSta.Add(packetSinkHelper.Install(wifiStaNodes.Get(staIndex)));
}
}
```

Figura 123: Código en NS-3 – función de generación de tráfico VoIP en sentido descendente

Cabe mencionar que tanto en el tráfico Gaming como VoIP también se han especificado puertos custom para facilitar el seguimiento. Aun así, en dichos tipos de tráfico los puertos pueden variar dependiendo de la implementación específica, el software y los protocolos empleados.

16.2.7. Función Principal: Monitorización, Ejecución de las Simulaciones y Cálculos Finales

Hasta ahora se ha definido la configuración completa de la red, pero además de los modelos empleados, se requiere monitorización y extracción de datos durante las simulaciones para poder obtener resultados que sirvan al propósito analítico del proyecto.

Por un lado, se utiliza la conexión de eventos a callbacks (funciones de devolución de llamada) para extraer **datos de señal** [117] y muestras según se indica en la Figura 124.

```
//////////////////////////////////////
// MONITORING
//////////////////////////////////////
// Signal and noise analysis from AP point of view
Config::ConnectWithoutContext("/NodeList/" + std::to_string(nStaA) + "/DeviceList*/Phy/MonitorSnifferRx",
| MakeCallback(&MonitorSniffRxA));
g_signalDbmAvg[0] = 0;
g_noiseDbmAvg[0] = 0;
g_samples[0] = 0;
Config::ConnectWithoutContext("/NodeList/" + std::to_string(0) + "/DeviceList*/Phy/MonitorSnifferRx",
| MakeCallback(&MonitorSniffRxStA));
g_signalDbmAvgSta[0] = 0;
g_noiseDbmAvgSta[0] = 0;
g_samplesSta[0] = 0;

if (nNetwork >= 2)
{
Config::ConnectWithoutContext("/NodeList/" + std::to_string(nStaA + nStaB + nAp) + "/DeviceList*/Phy/MonitorSnifferRx",
| MakeCallback(&MonitorSniffRxB));
g_signalDbmAvg[1] = 0;
g_noiseDbmAvg[1] = 0;
g_samples[1] = 0;
Config::ConnectWithoutContext("/NodeList/" + std::to_string(nStaA + nAp) + "/DeviceList*/Phy/MonitorSnifferRx",
| MakeCallback(&MonitorSniffRxBSta));
g_signalDbmAvgSta[1] = 0;
g_noiseDbmAvgSta[1] = 0;
g_samplesSta[1] = 0;
}
}
//...
```

Figura 124: Código en NS-3 – extracción de métricas de señal de los AP

Para ello, se ha realizado encapsulación en varias funciones (Figura 125) para facilitar la realización de callbacks, de forma que sirva tanto para APs como STAs y siga una estructura similar a la de la documentación de NS-3. [117] De este modo, se facilita su actualización frente a posibles futuros cambios en versiones del simulador.

```

////////////////////////////////////
// Monitor for APs
void MonitorSniffRxCommon(Ptr<const Packet> packet, uint16_t channelFreqMhz,
                          WifiTxVector txVector, MpduInfo aMpdu,
                          SignalNoiseDbm signalNoise, uint16_t staId,
                          uint8_t groupIndex, bool isStaVersion)
{
    if (isStaVersion)
    {
        g_samplesSta[groupIndex]++;
        g_signalDbmAvgSta[groupIndex] +=
            ((signalNoise.signal - g_signalDbmAvgSta[groupIndex]) / g_samplesSta[groupIndex]);
        g_noiseDbmAvgSta[groupIndex] +=
            ((signalNoise.noise - g_noiseDbmAvgSta[groupIndex]) / g_samplesSta[groupIndex]);
    }
    else
    {
        g_samples[groupIndex]++;
        g_signalDbmAvg[groupIndex] +=
            ((signalNoise.signal - g_signalDbmAvg[groupIndex]) / g_samples[groupIndex]);
        g_noiseDbmAvg[groupIndex] +=
            ((signalNoise.noise - g_noiseDbmAvg[groupIndex]) / g_samples[groupIndex]);
    }
}

void
MonitorSniffRx(Ptr<const Packet> packet, uint16_t channelFreqMhz,
               WifiTxVector txVector, MpduInfo aMpdu,
               SignalNoiseDbm signalNoise, uint16_t staId)
{
    MonitorSniffRxCommon(packet, channelFreqMhz, txVector, aMpdu, signalNoise, staId, 0, false);
}

void
MonitorSniffRxAP(Ptr<const Packet> packet, uint16_t channelFreqMhz,
                  WifiTxVector txVector, MpduInfo aMpdu,
                  SignalNoiseDbm signalNoise, uint16_t staId)
{
    MonitorSniffRxCommon(packet, channelFreqMhz, txVector, aMpdu, signalNoise, staId, 0, true);
}

```

Figura 125: Código en NS-3 – función de cálculo de métricas de señal

Por otro lado, se utiliza FlowMonitor (Figura 126) para obtener **datos de los flujos de tráfico** generados. El monitor se instala en todos los nodos del despliegue de red.

```

////////////////////////////////////
// FLOW MONITOR
FlowMonitorHelper flowmonHelper;
Ptr<FlowMonitor> monitor = flowmonHelper.InstallAll();

```

Figura 126: Código en NS-3 – declaración e instalación de FlowMonitor

En este caso el cálculo no se realiza mediante callbacks –lanzando la función por la ocurrencia de eventos– de forma directa como en los datos de señal, sino que FlowMonitor permite acceder a la información de los flujos de forma posterior a la ejecución de la simulación.

Por lo tanto, en la Figura 127 se muestran las instrucciones de ejecución de la simulación. Se deja un tiempo de guarda de un segundo para asegurar que las aplicaciones se finalicen de forma correcta y sin cierres abruptos que pueden afectar a las métricas.

```

////////////////////////////////////
// RUNNING
////////////////////////////////////
NS_LOG_INFO("Running simulation...");
Simulator::Stop(Seconds(simulationTime + 1));
Simulator::Run();

```

Figura 127: Código en NS-3 – ejecución de la simulación

Después, se puede realizar el cálculo en base a los flujos que se han generado a lo largo de las simulaciones. En la Figura 128 se recoge la inicialización de variables y objetos y preparación de estas para el análisis de estadísticas.

```

////////////////////////////////////
// RESULT CALCULATION
////////////////////////////////////
// Flow statistics
////////////////////////////////////
std::string oss("");
oss += "HeMcs" + std::to_string(mcs);
monitor->CheckForLostPackets();
Ptr<Ipv4FlowClassifier> classifier = DynamicCast<Ipv4FlowClassifier>(flowmonHelper.GetClassifier());
FlowMonitor::FlowStatsContainer stats = monitor->GetFlowStats();

double delayValues[stats.size()];
uint64_t j = 0;

uint16_t jitter[nStaA + nStaB + nStaC + nStaD + nAp * nNetwork] = {0};
Time avgJitter[nStaA + nStaB + nStaC + nStaD + nAp * nNetwork] = {Seconds(0.0)};
Time avgDelay[nStaA + nStaB + nStaC + nStaD + nAp * nNetwork] = {Seconds(0.0)};
double avgThroughput[nStaA + nStaB + nStaC + nStaD + nAp * nNetwork] = {0.0};
uint32_t avgTxPackets[nStaA + nStaB + nStaC + nStaD + nAp * nNetwork] = {0};
uint32_t avgTxBytes[nStaA + nStaB + nStaC + nStaD + nAp * nNetwork] = {0};
uint32_t avgRxPackets[nStaA + nStaB + nStaC + nStaD + nAp * nNetwork] = {0};
uint32_t avgRxBytes[nStaA + nStaB + nStaC + nStaD + nAp * nNetwork] = {0};

uint32_t flowID[stats.size() + 1];
std::string flowSourceAddress[stats.size() + 1];
uint32_t flowSourcePort[stats.size() + 1];
std::string flowDestinationAddress[stats.size() + 1];
uint32_t flowDestinationPort[stats.size() + 1];
std::string flowProtocol[stats.size() + 1];
uint32_t flowTxPackets[stats.size() + 1];
uint32_t flowTxBytes[stats.size() + 1];
double flowTxOffered[stats.size() + 1];
uint32_t flowRxPackets[stats.size() + 1];
uint32_t flowRxBytes[stats.size() + 1];
double flowThroughput[stats.size() + 1];
double flowMeanDelay[stats.size() + 1];
double flowLastDelay[stats.size() + 1];
double flowMeanJitter[stats.size() + 1];

```

Figura 128: Código en NS-3 – cálculo de estadísticas sobre los flujos de tráfico (I)

Acto seguido, se define el bucle sobre el que se itera a través de las estadísticas de flujos (Figura 129) haciendo uso de un mapa que identifica a los flujos por un ID y las estadísticas correspondientes a dicho flujo.

Se utiliza un clasificador para encontrar detalles del flujo de la iteración concreta, y se extraen datos del tráfico como el protocolo, las direcciones y puertos que intervienen.

```

uint32_t flowIndex = 0;
for (std::map<FlowId, FlowMonitor::FlowStats>::const_iterator i = stats.begin();
     i != stats.end();
     ++i)
{
    Ipv4FlowClassifier::FiveTuple t = classifier->FindFlow(i->first);
    flowID[flowIndex] = i->first;

    std::stringstream protoStream;
    protoStream << (uint16_t)t.protocol;
    if (t.protocol == 6)
    {
        protoStream.str("TCP");
        flowProtocol[flowIndex] = "TCP";
    }
    if (t.protocol == 17)
    {
        protoStream.str("UDP");
        flowProtocol[flowIndex] = "UDP";
    }
}

std::ostringstream oss;
t.sourceAddress.Print(oss);
flowSourceAddress[flowIndex] = oss.str();
flowSourcePort[flowIndex] = t.sourcePort;
std::ostringstream oss2;
t.destinationAddress.Print(oss2);
flowDestinationAddress[flowIndex] = oss2.str();
flowDestinationPort[flowIndex] = t.destinationPort;

```

Figura 129: Código en NS-3 – cálculo de estadísticas sobre los flujos de tráfico (II)

A continuación, se procede a extraer métricas específicas (Figura 130), como el número de paquetes y bytes transmitidos y recibidos, o la tasa de transferencia ofrecida –en megabits por segundo– para un flujo particular. En caso de que el flujo haya sido efectivo –de forma que al menos haya un paquete recibido– se calculan las estadísticas temporales como la duración de la transmisión, el delay o el jitter. Esto ocurre debido a que se toma como referencia la recepción para el cálculo de tiempos, por lo que al

menos debe haber un paquete para realizar estimaciones. Si no, se establecen los tiempos como nulos.

```

flowRxPackets[flowIndex] = i->second.rxPackets;
flowRxBytes[flowIndex] = i->second.rxBytes;

flowTxPackets[flowIndex] = i->second.txPackets;
flowTxBytes[flowIndex] = i->second.txBytes;
flowTxOffered[flowIndex] =
    i->second.txBytes * 8.0 / simulationTime / 1000.0 / 1000.0;

if (i->second.rxPackets > 0)
{
    double rxDuration =
        i->second.timeLastRxPacket.GetSeconds() - i->second.timeFirstTxPacket.GetSeconds();

    delayValues[j] = 1000 * i->second.delaySum.GetSeconds() / i->second.rxPackets;
    j++;

    flowThroughput[flowIndex] = i->second.rxBytes * 8.0 / rxDuration / 1000 / 1000;
    flowMeanDelay[flowIndex] =
        double(1000 * i->second.delaySum.GetSeconds()) / (i->second.rxPackets);
    flowLastDelay[flowIndex] = i->second.lastDelay.GetDouble() / 1000 / 1000;
    flowMeanJitter[flowIndex] =
        1000 * i->second.jitterSum.GetSeconds() / i->second.rxPackets;
}
else
{
    flowThroughput[flowIndex] = 0;
    flowMeanDelay[flowIndex] = 0;
    flowLastDelay[flowIndex] = 0;
    flowMeanJitter[flowIndex] = 0;
}
}

```

Figura 130: Código en NS-3 – cálculo de estadísticas sobre los flujos de tráfico (III)

Seguido, en la Figura 131 se muestra un ejemplo –para el caso de cuatro redes– de cómo se gestionaría el descubrimiento de redes y el índice de posición de los nodos en base al patrón de direccionamiento establecido en el apartado Función Principal: Protocolos de Internet por las distintas redes.

```

std::ostringstream oss3;
t.sourceAddress.Print(oss3);
std::string sourceAddress = oss3.str();

std::regex rgxNetwork("[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}");
std::smatch matchNetwork;
std::regex_search(sourceAddress, matchNetwork, rgxNetwork);

std::regex rgxNode("[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}");
std::smatch matchNode;
std::regex_search(sourceAddress, matchNode, rgxNode);

uint32_t pos = 0;

if (matchNetwork[1].str() == "1")
{
    pos = std::stoi(matchNode[1].str()) - 1;
}
else if (matchNetwork[1].str() == "2")
{
    pos = std::stoi(matchNode[1].str()) + nStaA;
}
else if (matchNetwork[1].str() == "3")
{
    pos = std::stoi(matchNode[1].str()) + nStaA + nStaB + nAp;
}
else
{
    pos = std::stoi(matchNode[1].str()) + nStaA + nStaB + nStaC + 2 * nAp;
}
}

```

Figura 131: Código en NS-3 – cálculo de estadísticas sobre los flujos de tráfico (IV)

Por último, se almacenan las estadísticas del flujo en las posiciones de los arrays correspondientes, como se puede observar en la Figura 132.

```

avgThroughput[pos] += i->second.txBytes * 8.0 / (simulationTime-1) / 1000.0 / 1000.0;
avgDelay[pos] += (i->second.delaySum);
avgJitter[pos] += (i->second.jitterSum);
avgTxPackets[pos] += (i->second.txPackets);
avgTxBytes[pos] += (i->second.txBytes);
avgRxPackets[pos] += (i->second.rxPackets);
avgRxBytes[pos] += (i->second.rxBytes);
jNode[pos]++;
flowIndex++;
}

```

Figura 132: Código en NS-3 – cálculo de estadísticas sobre los flujos de tráfico (V)

16.2.8. Función Principal: Volcado de Resultados a Fichero

Para finalizar con la función principal, se vuelcan los datos obtenidos a dos ficheros separados. El primero, recoge las estadísticas de los dispositivos, y el segundo, las métricas obtenidas de los flujos de tráfico.

Primero, en la Figura 133 se define el nombre de fichero, se realiza su apertura y se juntan los nodos en un único container.

```

////////////////////////////////////
// Device statistics (FILE)
////////////////////////////////////
std::string filenameDev("");
filenameDev += "Scenario-X-DeviceStats.csv";
const char* c = filenameDev.c_str();
(void)!freopen(c, "a", stdout);

NodeContainer nodes;
nodes.Add(wifiStaNodesA);
nodes.Add(wifiApNodesA);
nodes.Add(wifiStaNodesB);
nodes.Add(wifiApNodesB);
//...

```

Figura 133: Código en NS-3 – volcado de datos al fichero de estadísticas de dispositivos (I)

Gracias a lo anterior, se itera del siguiente modo: primero, se genera la información de a qué red pertenece el nodo en cuestión y de qué tipo es. En la Figura 134 se muestra el recorrido de los nodos y la forma de calcular los datos especificados para el caso de cuatro redes.

```

for (uint32_t i = 0; i < nStaA + nStaB + nStaC + nStaD + nNetwork; i++)
{
    std::cout << seedNumber << ", " << runNumber << ", ";

    if (i <= nStaA)
    {
        std::cout << "A" << ", ";
    }
    else if (i > nStaA && i <= nStaA + nStaB + 1)
    {
        std::cout << "B" << ", ";
    }
    else if (i > nStaA + nStaB + 1 && i <= nStaA + nStaB + nStaC + 2)
    {
        std::cout << "C" << ", ";
    }
    else
    {
        std::cout << "D" << ", ";
    }

    if (i == nStaA || i == nStaA + nStaB + 1 || i == nStaA + nStaB + nStaC + 2 ||
        i == nStaA + nStaB + nStaC + nStaD + 3)
    {
        std::cout << "AP"
        | | | << ", ";
    }
    else
    {
        std::cout << "STA"
        | | | << ", ";
    }
}

```

Figura 134: Código en NS-3 – volcado de datos al fichero de estadísticas de dispositivos (II)

Después, en la Figura 135 se recoge la escritura de los valores de las configuraciones del nodo – banda utilizada, esquema MCS, ancho de banda, GI – y las estadísticas calculadas. Dependiendo de la red en cuestión y el tipo de dispositivo, se cogen los valores relativos a las posiciones de los arrays que correspondan según lo definido en los apartados anteriores. De este modo, se alterna la escritura de STAs y AP de cada red de forma estructurada.


```

std::ostringstream oss;
const auto addressIP = nodes.Get(i)->GetObject<Ipv4>()->GetAddress(1, 0).GetLocal();
addressIP.Print(oss);
std::cout << oss.str() << ", ";

std::cout << mcs << ", " << channelWidth << ", " << gi << ", " << avgThroughput[i] / jNode[i]
<< ", " << avgTxBytes[i] / jNode[i] << ", " << avgTxPackets[i] / jNode[i] << ", "
<< avgRxBytes[i] / jNode[i] << ", " << avgRxPackets[i] / jNode[i] << ", "
<< avgDelay[i] / jNode[i] << ", " << avgJitter[i] / jNode[i] << ", ";

if (i < nStaA)
{
    std::cout << g_signalDbmAvgSta[0] << ", " << g_noiseDbmAvgSta[0] << ", "
    << (g_signalDbmAvgSta[0] - g_noiseDbmAvgSta[0]) << std::endl;
}
else if (i == nStaA)
{
    std::cout << g_signalDbmAvg[0] << ", " << g_noiseDbmAvg[0] << ", "
    << (g_signalDbmAvg[0] - g_noiseDbmAvg[0]) << std::endl;
}
else if (i > nStaA && i < nStaA + nStaB + 1)
{
    std::cout << g_signalDbmAvgSta[1] << ", " << g_noiseDbmAvgSta[1] << ", "
    << (g_signalDbmAvgSta[1] - g_noiseDbmAvgSta[1]) << std::endl;
}
else if (i == nStaA + nStaB + 1)
{
    std::cout << g_signalDbmAvg[1] << ", " << g_noiseDbmAvg[1] << ", "
    << (g_signalDbmAvg[1] - g_noiseDbmAvg[1]) << std::endl;
}
else if (i > nStaA + nStaB + 1 && i < nStaA + nStaB + nStaC + 2)
{
    std::cout << g_signalDbmAvgSta[2] << ", " << g_noiseDbmAvgSta[2] << ", "
    << (g_signalDbmAvgSta[2] - g_noiseDbmAvgSta[2]) << std::endl;
}
else if (i == nStaA + nStaB + nStaC + 2)
{
    std::cout << g_signalDbmAvg[2] << ", " << g_noiseDbmAvg[2] << ", "
    << (g_signalDbmAvg[2] - g_noiseDbmAvg[2]) << std::endl;
}
else if (i > nStaA + nStaB + nStaC + 2 && i < nStaA + nStaB + nStaC + nStaD + 3)
{
    std::cout << g_signalDbmAvgSta[3] << ", " << g_noiseDbmAvgSta[3] << ", "
    << (g_signalDbmAvgSta[3] - g_noiseDbmAvgSta[3]) << std::endl;
}
else
{
    std::cout << g_signalDbmAvg[3] << ", " << g_noiseDbmAvg[3] << ", "
    << (g_signalDbmAvg[3] - g_noiseDbmAvg[3]) << std::endl;
}
}
fclose(stdout);
  
```

Figura 135: Código en NS-3 – volcado de datos al fichero de estadísticas de dispositivos (III)

Para finalizar la escritura, se cierra el flujo estándar de salida, para poder dirigirlo al segundo fichero.

Dicho fichero es el que debe contener la información de los flujos de tráfico. Se realiza un bucle (Figura 136) en función del índice de flujo para recorrerlos todos, y se vuelven a utilizar expresiones regulares para distinguir la red y el tipo de dispositivo.

```

////////////////////////////////////
// Flow statistics (FILE)
////////////////////////////////////
std::string filenameFlow("");
filenameFlow += "Scenario1-FlowStats.csv";
const char* c2 = filenameFlow.c_str();
(void)ffreopen(c2, "a", stdout);

for (flowIndex = 0; flowIndex < stats.size(); flowIndex++)
{
    std::cout << seedNumber << ", " << runNumber << ", ";

    std::regex rgxNetwork("[0-9]{1,3}\\.[0-9]{1,3}\\.(.*)\\.[0-9]{1,3}");
    std::smatch matchNetwork;
    std::regex_search(flowSourceAddress[flowIndex], matchNetwork, rgxNetwork);

    std::regex rgxNode("[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.(.*)");
    std::smatch matchNode;
    std::regex_search(flowSourceAddress[flowIndex], matchNode, rgxNode);
  
```

Figura 136: Código en NS-3 – volcado de datos al fichero de estadísticas de flujos de tráfico (I)

Como se toma como referencia el emisor, en lugar de definirse como AP o STA, al flujo se le asocia un identificador de UL o DL para distinguir el tráfico ascendente y descendente (Figura 137).

```

if (matchNetwork[1].str() == "1")
{
    std::cout << "A" << ", ";
    if ((uint32_t)std::stoi(matchNode[1].str()) == nStaA + nAp)
    {
        std::cout << "DL" << ", ";
    }
    else
    {
        std::cout << "UL" << ", ";
    }
}
else if (matchNetwork[1].str() == "2")
{
    std::cout << "B" << ", ";
    if ((uint32_t)std::stoi(matchNode[1].str()) == nStaB + nAp)
    {
        std::cout << "DL" << ", ";
    }
    else
    {
        std::cout << "UL" << ", ";
    }
}
else if (matchNetwork[1].str() == "3")
{
    std::cout << "C" << ", ";
    if ((uint32_t)std::stoi(matchNode[1].str()) == nStaC + nAp)
    {
        std::cout << "DL" << ", ";
    }
    else
    {
        std::cout << "UL" << ", ";
    }
}
}

```

Figura 137: Código en NS-3 – volcado de datos al fichero de estadísticas de flujos de tráfico (II)

Por último, como se puede observar en la Figura 138 se recoge la escritura de los valores de las configuraciones del flujo –identificador, dirección y puertos origen y destino– y las estadísticas calculadas.

```

else
{
    std::cout << "D" << ", ";
    if ((uint32_t)std::stoi(matchNode[1].str()) == nStaD + nAp)
    {
        std::cout << "DL" << ", ";
    }
    else
    {
        std::cout << "UL" << ", ";
    }
}

std::cout << flowID[flowIndex] << ", " << flowSourceAddress[flowIndex] << ", "
<< flowSourcePort[flowIndex] << ", " << flowDestinationAddress[flowIndex] << ", "
<< flowDestinationPort[flowIndex] << ", " << flowProtocol[flowIndex] << ", "
<< flowTxPackets[flowIndex] << ", " << flowTxBytes[flowIndex] << ", "
<< flowTxOffered[flowIndex] << ", " << flowRxPackets[flowIndex] << ", "
<< flowRxBytes[flowIndex] << ", " << flowThroughput[flowIndex] << ", "
<< flowMeanDelay[flowIndex] << ", " << flowLastDelay[flowIndex] << ", "
<< flowMeanJitter[flowIndex] << "\n";
}

```

Figura 138: Código en NS-3 – volcado de datos al fichero de estadísticas de flujos de tráfico (III)

16.2.9. Función Principal: Final de Función

Si el programa se ejecuta con éxito, simplemente termina devolviendo un valor 0, como indica la Figura 139.

```

} return 0; //End of main()

```

Figura 139: Código en NS-3 – final de función main

17. Anexo IV: Procesamiento de Resultados a través de Clustering

Un adecuado tratamiento de los resultados de las simulaciones es crucial para el desarrollo del presente estudio. Sin embargo, con ánimo de favorecer posteriores reproducciones del presente trabajo y con carácter didáctico, el presente anexo recoge los aspectos principales de los programas utilizados para el tratamiento de los resultados de las simulaciones con por medio de técnicas de clustering.

El código empleado se encuentra disponible en GitHub con acceso público: <https://github.com/Atxa14/TFM-Files>

17.1. Preprocesado de los Resultados de Simulación

En primer lugar, el preprocesado hace uso de una serie de librerías de manejo, representación y preprocesado de datos numéricos recogidas en la Figura 140. Estas serán utilizadas de forma recursiva a lo largo de todo el código.

```

#####
# LIBRERÍAS #####
#####
# Librerías de manejo de datos numéricos
import pandas as pd
import numpy as np
import seaborn as sb

# Representación
import matplotlib.pyplot as plt

# Recursos de SK-Learn generales
from sklearn.preprocessing import MinMaxScaler
  
```

Figura 140: Código en Python – librerías de uso genérico para el manejo, representación y preprocesado de los resultados de simulación.

A continuación, en la Figura 141 se realiza la definición del entorno de trabajo, a través del volcado del fichero de resultados a la estructura DataFrame, y mostrando mediante diferentes gráficos las estadísticas de los datos a analizar.

```

#####
# WORKING ENVIRONMENT DEFINITION #####
#####
# Graphic representation characteristics: sizes and style
np.set_printoptions(threshold=np.inf)

plt.rcParams['figure.figsize'] = (10, 9)
plt.style.use('seaborn-dark')

# File dump to DataFrame
filename = "SET-101/C/Scenarios-APStats.csv"
dataframe = pd.read_csv(filename)
print(dataframe.head()) # Show first 5 lines

# Get data statistics
print(dataframe.describe()) # Generate descriptive statistics
print(dataframe.groupby('scenario').size()) # Get sample number per environment

# Sample distribution histogram
dataframe.drop(['scenario'],axis=1).hist() # Scenario axis histogram
plt.show()

# Variable crossing
sb.pairplot(dataframe.dropna(), hue='scenario', palette="dark",
            vars=["snr", "avgrxpackets", "avgthroughput"])
plt.show()
  
```

Figura 141: Código en Python – definición del entorno de trabajo.

Después, se realiza el preprocesado de los datos. Para ello, se realizan ajustes sobre las columnas cuyos valores no son numéricos para facilitar los cálculos, como se puede apreciar en la Figura 142. Asimismo, también se lleva a cabo el procedimiento de normalización de datos, y se hace una copia del resultado mediante el volcando de la información sobre un nuevo fichero.

```

#####
# PREPROCESSING #####
#####
# Mapping for non numeric parameters
mapping_strings = {'A': 1, 'B': 2, 'C': 3, 'D': 4, 'E': 5, 'STA': 1, 'AP': 2}
mapped_dataframe = dataframe.replace(mapping_strings)
mapped_dataframe['address_'] = mapped_dataframe['address'].str.replace(r'192\.168\.\\d+\.\\d+', '', regex=True)

# Data scaling
min_max_scaler = MinMaxScaler()
array_scaled = min_max_scaler.fit_transform(mapped_dataframe)
dataframe_scaled = pd.DataFrame(array_scaled, columns=mapped_dataframe.columns)

# Scaled file generation
df = pd.DataFrame(dataframe_scaled)
df.to_csv(r"scaledData.csv")

# Data structure definition
X = np.array(dataframe_scaled[["avgthroughput", "avgrxpackets", "snr",
                              "avgtotaljitter", "avgtotaldelay", "avgtxpackets"]])
Y = np.array(mapped_dataframe['scenario'])
print(X.shape) # X axis check
print(Y.shape) # Y axis check
  
```

Figura 142: Código en Python – preprocesado de los datos (I)

Acto seguido, y para terminar con esta primera parte común del código, se realiza una primera representación de las muestras de datos de forma tridimensional y sin mostrar ninguna etiqueta relativa a los escenarios.

```

# 3D plotting of samples
plt.rcParams['figure.figsize'] = (20, 20)
figure1= plt.figure()
axes1 = figure1.add_subplot(111, projection='3d')
axes1.set_xlabel('avgthroughput', fontsize=22)
axes1.set_ylabel('avgrxpackets', fontsize=22)
axes1.set_zlabel('snr', fontsize=22)
colors1 =['black', 'black', 'black']
assignment1 =[]

# Assing colors to each sample
for row in Y:
    assignment1.append(colors1[row])
axes1.scatter(X[:, 0], X[:, 1], X[:, 4], c=assignment1)

# Plot display
plt.show()
  
```

Figura 143: Código en Python – preprocesado de los datos (II)

17.2. Implementación de los Algoritmos de Clustering

Los extractos de código anteriores son comunes a todos los algoritmos. Sin embargo, a continuación, se presentan las partes relativas a cada uno de los programas distintos desarrollados, es decir, la implementación de cada algoritmo por separado.

17.2.1. Implementación de K-Means

En primer lugar, se ha de importar la clase K-Means (Figura 144) de la librería Scikit Learn para posibilitar su implementación en el código.

```

from sklearn.cluster import KMeans
  
```

Figura 144: Código en Python – importación de K-Means de la librería SKLearn.

En este caso, el **parámetro fundamental K** del algoritmo se puede analizar mediante el estudio de la variación de la distancia a los centroides de un rango de grupos posibles. Este método, también conocido como **curva de quiebre**, se desarrolla en Python según el fragmento de código de la Figura 145.

```

#####
# K PARAMETER ANALYSIS #####
#####
}# K range definition
clusterQuantity = range(1, 10)

# K parameter calculation
kmeans = [KMeans(n_clusters=i) for i in clusterQuantity]
score = [kmeans[i].fit(X).score(X) for i in range(len(kmeans))]

# Display Elbow Curve
plt.plot(clusterQuantity, score)
plt.tick_params(axis='both', which='major', labelsize=20)
plt.xlabel('Number of Clusters', fontsize=22)
plt.ylabel('Score', fontsize=22)
plt.title('Elbow Curve', fontsize=22)
plt.show()
  
```

Figura 145: Código en Python – estimación del parámetro K de K-Means.

Por último, K-Means se implementa mediante los métodos proporcionados por Scikit Learn como se indica en la Figura 146. Una vez realizada la agrupación, se pueden mostrar los valores adquiridos por los centroides de los clústeres y la predicción de pertenencia a grupos a través de etiquetas. En este caso se hace uso, a modo de ejemplo, de 10 inicializaciones del algoritmo. Además, se establece un total de 100 iteraciones sobre el algoritmo para la realización del clustering.

```

#####
# K-MEANS #####
#####
}# K selection de K and centroid calculation
kmeans = KMeans(n_init='auto', n_clusters=4, max_iter=100).fit(X)
centroids = kmeans.cluster_centers_
print(centroids)

# Cluster prediction
labels = kmeans.predict(X)
print(labels)
  
```

Figura 146: Código en Python – ejecución de K-Means.

17.2.2. Implementación de DBSCAN

Al igual que en K-Means, en la implementación de DBSCAN inicialmente se importa la clase de Scikit Learn correspondiente a este algoritmo de agrupamiento (Figura 147). Adicionalmente, se requiere la clase del algoritmo NearestNeighbors para el estudio de los parámetros de DBSCAN.

```

# DBSCAN: required specific classes
from sklearn.cluster import DBSCAN
from sklearn.neighbors import NearestNeighbors
  
```

Figura 147: Código en Python – importación de clases para la implementación de DBSCAN.

La realización de dicho análisis se lleva a cabo mediante el procedimiento mostrado en la Figura 148. En el caso del **parámetro EPS** se hace uso del algoritmo de cálculo de vecinos más cercanos, de modo que se pueda estudiar de forma gráfica la evolución de EPS y las muestras incluidas en clústeres para cada posible valor de este parámetro.

Asimismo, el **número mínimo de muestras** por grupo se establece como el número de variables de cada muestra incrementado en una unidad.

```

#####
# EPSILON PARAMETER ANALYSIS #####
#####
# Nearest neighbors usage
neighbor=NearestNeighbors(n_neighbors=30)
neighbors=neighbor.fit(X)
distances, indexes = neighbors.kneighbors(X)
distances = np.sort(distances, axis=0)
distances = distances[:, 1]
plt.title('EPS Graphic')
plt.ylim(0,0.7)
plt.xlim(0,400)
plt.plot(distances)
plt.show()

#MIN_SAMPLES Selection
#MIN_SAMPLES >= data_dimensions + 1
  
```

Figura 148: Código en Python – estimación de los parámetros de DBSCAN.

Se ha de tener en cuenta que el límite en el eje X y en el eje Y para los diferentes estudios de mejora de la calidad del servicio dependen directamente del número de muestras empleadas. En este caso, la Figura 148 muestra un ejemplo para el caso de utilizar 400 muestras en total, pero dicho valor deberá ajustarse si el conjunto de datos así lo hace.

Por último, se aplica el algoritmo (Figura 149) y se realiza la asignación de etiquetas a las muestras para la formación de los clústeres.

```

# DBSCAN
dbscan = DBSCAN(eps=0.135,min_samples=9).fit(X)

# Cluster prediction
labels = dbscan.labels_
print(labels)
  
```

Figura 149: Código en Python – ejecución de DBSCAN.

17.2.3. Implementación de GMM

De forma similar a los dos algoritmos anteriores, en primer lugar, se importa la clase correspondiente a GMM de Scikit Learn (Figura 122).

```

# GMM
from sklearn.mixture import GaussianMixture
  
```

Figura 150: Código en Python – importación de clases para la implementación de GMM.

GMM requiere disponer del conocimiento del **número de componentes** gaussianas – o clústeres– subyacentes en el conjunto de datos. Existen varios métodos para tratar de deducir el número de componentes. En la Figura 151 se muestran los criterios AIC y BIC, que en conjunto tratan de equilibrar la bondad del ajuste del modelo y su complejidad.

El procedimiento se resume en el entrenamiento de múltiples modelos GMM y el cálculo de AIC y BIC para cada uno de ellos. Después, se deberá seleccionar el modelo de menor valor AIC o BIC.

```
#####
# COMPONENT NUMBER ANALYSIS #####
#####
# Component range
component_quantity = range(1, 10)

# Centroid distance fixing method
models = [GaussianMixture(n).fit(X) for n in component_quantity]

# Curve display (AIC, BIC)
plt.plot(component_quantity, [m.bic(X) for m in models], label='BIC')
plt.plot(component_quantity, [m.aic(X) for m in models], label='AIC')
plt.xlabel('Number of Components')
plt.ylabel('Score')
plt.legend(loc='best')
plt.title('Akaike and Bayesian information criterions')
plt.show()
```

Figura 151: Código en Python – estimación del número de componentes de GMM.

Sin embargo, en GMM también resulta necesario proveer de la **matriz de covarianza, la media de las distribuciones, y los pesos**. Para ello, el mismo objeto GaussianMixture emplea el algoritmo EM (Expectation-Maximization) para estimar los parámetros de GMM durante el proceso de entrenamiento mediante el método fit. Dicho entrenamiento se puede apreciar en el código de la Figura 152, junto con la ejecución final del algoritmo GMM.

```
#####
# GMM #####
#####
# Clustering settings
gmm = GaussianMixture(n_init='auto', n_components=4, max_iter=100).fit(X)

# Cluster prediction
labels = gmm.predict(X)
print(labels)
```

Figura 152: Código en Python – ejecución de GMM.

17.2.4. Implementación de BIRCH

Por último, en el caso de BIRCH también ha de importarse la clase específica del algoritmo mediante la librería Scikit Learn (Figura 153).

```
# Birch
from sklearn.cluster import Birch
```

Figura 153: Código en Python – importación de clases para la implementación de BIRCH.

A diferencia de los algoritmos anteriores, en el caso de BIRCH no se dispone de un método de estimación específico para la definición de sus parámetros: el **umbral** y el **factor de ramificación**.

Por lo tanto, el procedimiento utilizado se basa en la iteración sobre distintos valores de ambos parámetros. Un ejemplo de ejecución se muestra en la Figura 154.

```
#####
# BIRCH #####
#####
# Birch settings
birch = Birch(n_clusters=4, branching_factor=10, threshold=0.5).fit(X)

# Cluster prediction
labels = birch.predict(X)
print(labels)
```

Figura 154: Código en Python – ejecución de BIRCH.

17.3. Representación de los Resultados

Para hacer más intuitivo el análisis de resultados, se han diseñado una serie de funciones y procedimientos para el **ajuste gráfico de los clústeres**. A continuación, se presentan los recursos principales basados en representación gráfica y conteo de muestras en tablas para facilitar en **análisis de resultados**.

En primer lugar, la Figura 155 recoge el procedimiento de representación en base a tablas de la asignación de etiquetas por clúster y escenario, lo que posibilita evaluar el número de errores en cada escenario y sobre qué otro clúster se realiza la desviación de cada muestra. De forma breve, toma como eje en cada tabla una de las etiquetas, y evalúa sobre cuál de los clústeres ha sido asignada.

```

#####
# RESULT ANALYSIS #####
#####
# Check scenario based cluster hypothesis
scenario_ids = [0, 1, 2, 3]
colores = ['blue', 'red', 'green', 'orange']

for i in scenario_ids:
    group_referrer_index = copy['label'] == i
    group_referrals = copy[group_referrer_index]
    group_diversity = pd.DataFrame()
    group_diversity['scenario'] = scenario_ids
    group_diversity['quantity'] = group_referrals.groupby('scenario').size().reset_index(drop=True)
    group_diversity.fillna(0, inplace=True)
    group_diversity['quantity'] = group_diversity['quantity'].astype(int)
    print(f"Assignment of label {i} per scenario")
    print(group_diversity)
    print("\n")
  
```

Figura 155: Código en Python – representación en tablas de la diversidad de grupos para el caso del etiquetado de escenarios.

En segundo lugar, las tablas anteriores pueden contrastarse con la tabla que se obtiene a través del fragmento de código de la Figura 156, que realiza el conteo de la asignación de muestras, devolviendo el tamaño de cada clúster con la etiqueta correspondiente.

```

# Show how much samples are in each cluster
copy = pd.DataFrame()
dataframe_scaled.reset_index(inplace=True)
print(dataframe_scaled)
copy['index'] = dataframe_scaled['index'].values
copy['scenario'] = dataframe_scaled['scenario'].values
copy['label'] = labels

colors1 = ['blue', 'red', 'green', 'orange']

tam_clusters = pd.DataFrame()
tam_clusters['color'] = colors1
tam_clusters['quantity'] = copy.groupby('label').size()
print(tam_clusters)
  
```

Figura 156: Código en Python – conteo de muestras de cada escenario por clúster.

Para finalizar, la Figura 157 muestra cómo obtener gráficos bidimensionales de forma estructurada, facilitando así la generación de nuevas combinaciones de variables. De este modo, se generan figuras cuyos ejes son los parámetros extraídos del simulador y se permite observar nuevas perspectivas que influyen en la agrupación.

Este último extracto de código resulta de especial interés para la extracción de conclusiones, ya que permite relacionar variables, observar densidades de muestras desde distintas perspectivas, y hacer un seguimiento en detalle para la formación de

perfiles apoyándose y contrastando estas gráficas con la información generada sobre flujos de tráfico.

```
# Bidimensional plots
combinations = [
    {
        "x": dataframe["avgtxpackets"].values,
        "y": dataframe["avgtxpackets"].values - dataframe["avgrxpackets"].values,
        "title": 'avgtxpackets - avgrxpackets'
    },
    {
        "x": dataframe["avgtxpackets"].values - dataframe["avgrxpackets"].values,
        "y": dataframe["snr"].values,
        "title": 'avglostpackets - snr'
    },
    {
        "x": dataframe["avgthroughput"].values,
        "y": dataframe["snr"].values,
        "title": 'avgthroughput - snr'
    },
    {
        "x": dataframe["avgtotaldelay"].values,
        "y": dataframe["avgtotaljitter"].values,
        "title": 'avgtotaldelay - avgtotaljitter'
    }
]

plt.rcParams['figure.figsize'] = (7, 7)
for combo in combinations:
    plt.scatter(combo["x"], combo["y"], c=assignment1, s=10)
    plt.title(combo["title"], fontsize=18)
    plt.show()
```

Figura 157: Código en Python – representación bidimensional de muestras.

17.3.1. Funciones Complementarias

En relación con la representación de los clústeres, se ha de mencionar que la asignación de la numeración de las etiquetas es aleatoria, por lo que estas variarán en cada ejecución. En otras palabras, si se asignan colores fijos a las etiquetas como se hace en el caso de los escenarios, la imagen basada en las etiquetas predefinidas se coloreará como se desee, pero el etiquetado posterior al clustering podrá variar en el orden de colores.

```
import re
from collections import Counter
```

Figura 158: Código en Python – importación de clases para la reordenación de etiquetas.

Para dar solución a dicha problemática sobre casos de uso en cuyas formaciones de grupos predomina una etiqueta distinta, se han diseñado las funciones de la Figura 159, para lo que se requiere la importación de los recursos indicados en la Figura 158.

```
def detect_predominant_number(segment):
    positive_segment = [num for num in segment if num >= 0]
    count = Counter(positive_segment)
    return count.most_common(1)[0][0] if positive_segment else None

def reorder_labels(labels, partition):
    n = len(labels)
    part = n // partition
    predominant_numbers = []
    for i in range(partition):
        start, end = i * part, (i + 1) * part
        predominant_numbers.append(detect_predominant_number(labels[start:end]))
    mapping = {predominant_numbers[i]: i for i in range(partition) if predominant_numbers[i] is not None}
    reordered_labels = [mapping.get(num, num) if num >= 0 else num for num in labels]
    return np.array(reordered_labels)
```

Figura 159: Código en Python – funciones de reordenación de etiquetas.

Cabe mencionar que el código anterior también contempla el caso de descarte de muestras, de forma que las muestras asignadas con valores negativos sean ignoradas en el reajuste del etiquetado.

De este modo, por ejemplo, una vez obtenidas las etiquetas resultantes en el agrupamiento, basta con usar la instrucción de la Figura 160 para reordenarlas en base al número de grupos formados.

```
Labels = reorder_labels(Labels,4)
```

Figura 160: Código en Python – ejemplo de ejecución de la reordenación de etiquetas.