

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

**ESCUELA DE INGENIERÍA DE
TELECOMUNICACIÓN Y ELECTRÓNICA**



PROYECTO FIN DE CARRERA

**RED WiFi Y SERVICIOS MULTIMEDIA PARA COMPLEJO
HOTELERO EN FUERTEVENTURA**

TITULACIÓN:	Telemática
TUTORES:	Álvaro Suárez Sarmiento Elsa María Macías López
AUTOR:	Juan Francisco Ramírez Castellano
FECHA:	Julio 2014

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

**ESCUELA DE INGENIERÍA DE
TELECOMUNICACIÓN Y ELECTRÓNICA**



PROYECTO FIN DE CARRERA

**RED WiFi Y SERVICIOS MULTIMEDIA PARA COMPLEJO
HOTELERO EN FUERTEVENTURA**

Presidente:

Secretario:

Vocal:

Tutores:

Autor:

Nota:

TITULACIÓN:

Telemática

TUTORES:

Álvaro Suárez Sarmiento

Elsa María Macías López

AUTOR:

Juan Francisco Ramírez Castellano

FECHA:

Julio 2014

Título del Proyecto Fin de Carrera

***RED WiFi Y SERVICIOS MULTIMEDIA PARA
COMPLEJO HOTELERO EN FUERTEVENTURA***

Índice

1. Introducción	1
2. WiFi y servicios multimedia	9
3. Herramientas usadas	23
4. Modelado de la simulación	45
5. Diseño del Proyecto	91
6. Análisis de la infraestructura.....	105
7. Conclusiones y posibles ampliaciones	129
Pliego de condiciones y presupuesto	133
Bibliografía y referencias	141
Glosario	145
Anexo A: planos	149

Índice detallado

1. Introducción	1
1.1 Ventajas de la simulación	2
1.2 Objetivo del proyecto	5
1.3 Contexto del desarrollo del proyecto	6
1.4 Estructura de la memoria	6
2. WiFi y servicios multimedia	9
2.1 WiFi	10
2.1.2 Topologías de red	11
2.2 Estándar IEEE 802.11	12
2.3 Calidad de servicio	16
2.4 Aplicaciones multimedia	18
3. Herramientas usadas	23
3.1 Introducción.....	24
3.2 VirtualBox	25
3.2.2 Instalación.....	26
3.3 Ubuntu 12.04 TLS	31
3.3.1 Instalación.....	32
3.4 Simulador ns3	34
3.4.1 Descripción del Simulador	35
3.4.2 Ayudantes de topología.....	38
3.4.3 Organización del software	38
3.4.4 Instalación.....	39
4. Modelado de la simulación	45
4.1 Introducción.....	46
4.2 Componentes básicos.....	47

4.2.1 Modelado de etiquetas de paquetes	47
4.3 Modelado de los nodos	49
4.4 Modelado de los protocolos de comunicación	52
4.5 Modelado de los canales de comunicación.....	53
4.6 Modelado del canal inalámbrico	57
4.6.1 Modelado de la propagación.....	59
4.6.2 Modelado de la codificación de señales inalámbricas	61
4.6.3 Modelado del MAC.....	63
4.6.4 Modelado del estándar WiFi	64
4.6.5 Modelado de la movilidad	65
4.7 Modelado de edificios	66
4.8 Modelado de aplicaciones.....	68
4.8.1 Modelado de la QoS del tráfico	71
4.8.2 Modelado del tráfico de VoIP.....	72
4.8.3 Modelado del tráfico de vídeo	75
4.8.4 Modelado del tráfico HTTP.....	77
4.8.5 Modelado del tráfico FTP	78
4.9 Modelado de herramientas para captura y representación de los datos.....	79
4.9.1 FlowMonitor	80
4.9.2 Gnuplot.....	83
4.9.3 Herramientas de animación de simulaciones	85
4.10 Diagramas de casos de uso y de secuencia	87
4.10.1 Casos de uso y diagrama de secuencia del canal inalámbrico	87
4.10.3 Casos de uso y diagrama de secuencia del tráfico	89
5. Diseño del Proyecto.....	91
5.1 Introducción.....	92
5.1 Consideraciones del complejo.....	93
5.2 Situación actual	95
5.2.1 Red de telefonía y acceso a Internet.....	95

5.2.2 Red de fibra óptica	96
5.3 Dimensionado.....	98
5.3.1 Escenario exterior.....	98
5.3.2 Escenario exterior con edificios.....	99
5.3.3 Escenario exterior e interior.....	101
5.3.4 Mapa de cobertura.....	102
6. Análisis de la infraestructura.....	105
6.1 Introducción.....	106
6.2 Escenario 1.....	106
6.3 Escenario 2.....	109
6.4 Escenario 3.....	113
6.4 Escenario 4.....	117
6.4.1 Análisis del tráfico para el estándar IEEE 802.11g.....	119
6.4.12 Análisis del tráfico para el estándar IEEE 802.11n	121
6.5 Escenario final.....	124
7. Conclusiones y posibles ampliaciones	129
7.1 Conclusiones.....	130
7.1 Posibles ampliaciones.....	131
Pliego de condiciones y presupuesto	133
Pr.1 Pliego de condiciones.....	134
Pr.1.1 Descripción de la infraestructura.....	134
Pr.1.2 Requisitos hardware y software	134
Pr.2 Presupuesto	135
Pr.2.1 Descripción de las fases del proyecto.....	136
Pr.2.2 Mano de obra.....	137
Pr.2.3 Coste del material	139
Pr.2.4 Coste total del proyecto.....	139

Bibliografía y referencias	141
Glosario	145
Anexo A: planos	149

1. Introducción

En el presente capítulo se presentan las ideas básicas en las que se ha trabajado en este *Proyecto Final de Carrera (PFC)*, las herramientas usadas, los objetivos principales y la estructura de la memoria.

1.1 Ventajas de la simulación

El éxito de las redes inalámbricas actualmente es indiscutible simplificando la comunicación humana y permitiéndola estar constantemente conectada, propiciando el desarrollo de nuevos conceptos como el de *Conectividad Total*.

Concretamente la *Wireless Local Area Network (WLAN)* ha supuesto un gran avance en el terreno de las telecomunicaciones debido a que ofrecen las ventajas de una red cableada sin las limitaciones que supone estar atado a una ubicación fija, permitiendo ampliar o sustituir una infraestructura con cables cuando resulta costoso o imposible realizar un tendido.

Las instalaciones temporales son un claro ejemplo de situación en la que la red inalámbrica tiene sentido o incluso es necesaria, además algunas normativas de construcción pueden prohibir el uso de cables en determinadas instalaciones.

Sin embargo, la tecnología que hay detrás de las WLAN es sofisticada y la multitud de aplicaciones y entornos en los que se desarrolla, hacen que una WLAN sea una red de telecomunicación compleja, tanto de diseñar y dimensionar, como de implantar, y posteriormente optimizar y operar, para obtener el máximo rendimiento [6]. Por ello se suele recurrir a la simulación.

Actualmente las redes de comunicación deben de estar preparadas para servicios que realizan un consumo muy alto de recursos. Este hecho puede provocar graves defectos en el funcionamiento de las redes debido, por ejemplo, a un mal diseño que dé como resultado una red de baja capacidad no adaptada a los requerimientos de los servicios que hace uso de ella. En algunos casos la implantación de estos servicios podría influir en partes del sistema que funcionaban correctamente. Por este hecho se hace necesaria la realización de un análisis previo que permita determinar el impacto que pueden provocar dichos servicios evitando los problemas que pudieran surgir y el

consecuente descontento de los usuarios de la red. Además de evitar posibles pérdidas económicas.

El uso de simuladores durante el diseño de infraestructuras de comunicaciones tiene una serie de ventajas importantes ya que nos permite estudiar el comportamiento de diferentes diseños para la infraestructura antes de su implementación en el mundo real asegurándonos de que las configuración seleccionada para la infraestructura y los equipos que se van a emplear, para desempeñar las diferentes funciones dentro de la misma, cumplan de manera satisfactoria los objetivos y las funciones para los cuales fueron elegidos.

Además la simulación permite plantear situaciones muy parecidas a los casos reales mediante las que se podrían experimentar diversas alternativas a problemas y su posible impacto relativo a costos de implementación, mantenimiento, escalabilidad y satisfacción de sus usuarios. En la Figura 1.1 se muestra un proceso genérico para modelar una simulación [1].

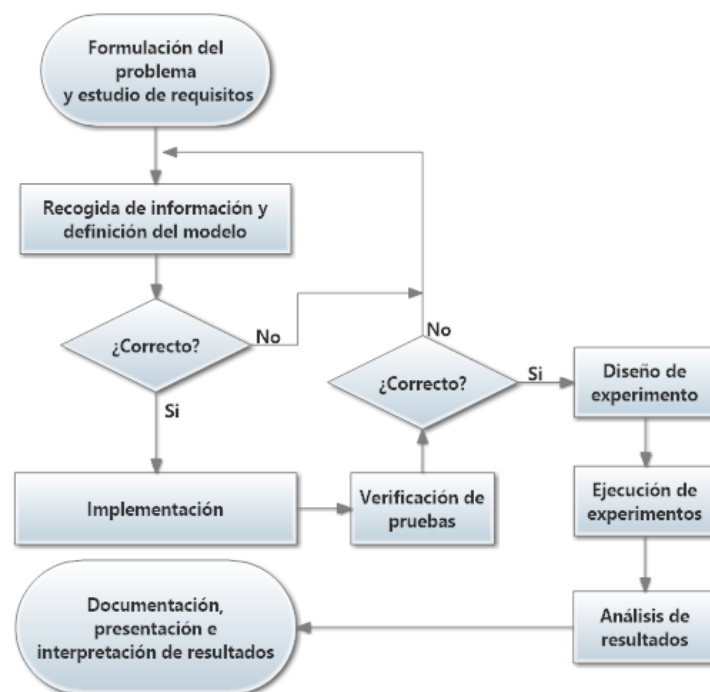


Figura 1.1. Proceso de simulación

La simulación de sistemas ofrece un método para analizar su comportamiento. Para comprender las características técnicas de la simulación, mediante la formación de modelos y técnicas estadísticas, y aplicarlas a un problema real, es necesario familiarizarse con los conceptos que describen un sistema y un modelo.

Un modelo para la simulación de redes tiene una estructura compuesta por una serie de componentes.

- **Entidades:** objetos que forman parte de un sistema.
- **Atributos:** propiedades que poseen las entidades. Pueden ser:
 - Fijados durante el diseño del sistema.
 - Fijados por el entorno.
 - Fijados por el usuario.
 - De salida.
- **Sucesos:** hechos que ocurren en un instante de tiempo y que dan lugar a cambios en el estado del sistema.
- **Colas:** estados pasivos de una entidad mientras espera el inicio de una actividad.
- **Estados:** condiciones del modelo o sus entidades, de forma que se puede saber si una acción se puede ejecutar o si se puede elegir entre varias.

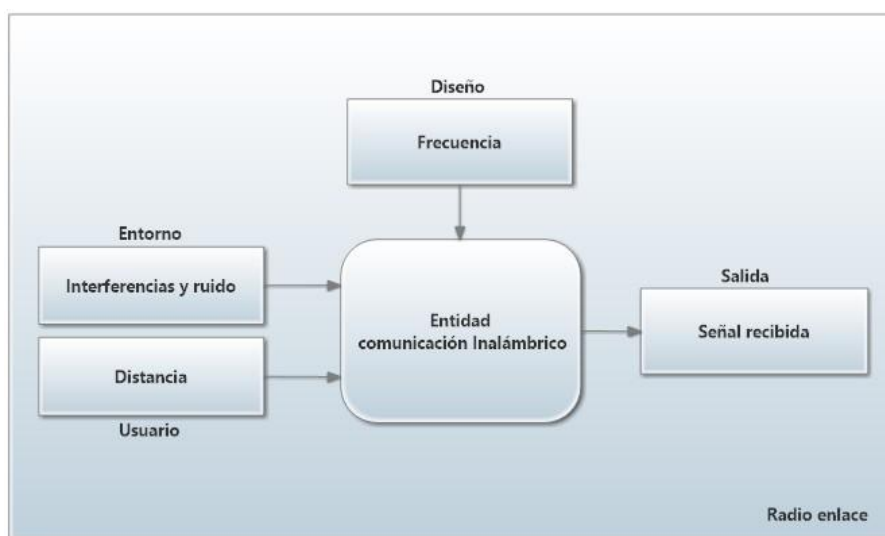


Figura 1.2. Ejemplo de sistema que simula un radioenlace

1.2 Objetivo del proyecto

El principal objetivo de este proyecto es el diseño de una WLAN que unifique las comunicaciones dentro del complejo hotelero garantizando la *Calidad de Servicio (QoS*, del inglés *Quality of Service*) para los servicios ofrecidos a los clientes en las habitaciones y los servicios de voz del complejo, a través de simulaciones teóricas que evalúen su correcto desempeño.

Para ello se realiza un análisis del estado actual del estándar *Wireless Fidelity (WiFi)* como tecnologías para construir WLAN. Es importante destacar que actualmente es el estándar más extendido ya que al complejo llegan usuarios domésticos con terminales equipados con este tipo de conexiones mayoritariamente y una de las prioridades es que tengan una elevada *Calidad de Experiencia (QoE*, del inglés *Quality of Experience*) con los servicios ofrecidos a través de la infraestructura diseñada.

Con el objetivo de garantizar la cobertura en todo el complejo hotelero se realiza un estudio teórico del alcance de las señales WiFi, mediante simulaciones en ns3, en las que se tienen en cuenta los diferentes factores que influyen en la propagación de las señales de radio dando como resultado el diseño de un mapa de coberturas que evite las sombras de cobertura en las zonas del complejo que requiera conexión inalámbrica.

Para garantizar el correcto desempeño de las aplicaciones que hacen uso de la red diseñada se realiza un estudio teórico sobre el impacto que tienen los diferentes tipos de tráfico de datos, en base a los usos para los que está destinada la infraestructura inalámbrica, mediante simulaciones en *network simulator versión 3 (ns3)* en las que se utilizan modelos de dispositivos similares a los disponibles en el Mercado, que cumplan con el estándar WiFi. Se analiza especialmente las el tráfico de *Voice over Internet Protocol (VoIP)*, ya que es uno de los más sensibles a la QoS, estableciendo las limitaciones y el desempeño de la infraestructura diseñada.

1.3 Contexto del desarrollo del proyecto

El contexto del presente proyecto es un complejo hotelero que tiene una capacidad de alojamiento de 350 habitaciones distribuidas en diferentes bloques de apartamentos por toda la zona más al este del complejo, actualmente no dispone de una infraestructura inalámbrica que de acceso a los clientes y tiene serios problemas con algunos tramos de la red de telefonía interna.

Considerando una ocupación de entorno a un 60% de media durante todo el año, se estima como número de usuarios potenciales un 70% de dicha cantidad, lo que nos da un total de 147. Todos los usuarios no hacen uso de la infraestructura simultáneamente por lo que se estima que un 30% de los usuarios podrían estar conectados a la vez.

Otra cuestión que se tiene en cuenta es que al complejo se le propone la sustitución de la red telefónica por tecnología de VoIP y parte del tráfico de las llamadas se realizaran a través de la infraestructura inalámbrica que se diseña en el presente *Proyecto Final de Carrera (PFC)*.

1.4 Estructura de la memoria

En este apartado se presenta la estructura de la memoria de este PFC.

En el capítulo 2 se realiza una breve introducción a las WLAN presentando una relación de las tecnologías WiFi disponibles actualmente, haciendo especial hincapié en los aspectos relacionados con la QoS y cómo afectan a las aplicaciones de VoIP.

En el capítulo 3 se comentan las distintas herramientas utilizadas para la realización del proyecto y se explica el proceso para la instalación de las mismas. Estas herramientas son el software de simulación ns3, el *Sistema Operativo (SO)* Ubuntu [15] y el software de virtualización *Virtualbox* [4].

En el capítulo 4 se explica detalladamente cómo se realiza el modelado de cada uno de los componentes de la red simulada objeto de estudio del presente PFC.

En el capítulo 5 se realiza un estudio teórico sobre la cobertura del complejo hotelero mediante simulaciones con *ns3* [2] para diferentes escenarios en los que se tienen en cuenta la problemática del terreno y los obstáculos que se encuentra la señal radioeléctrica al propagarse. Realizando una comparativa entre distintas casuísticas que se pueden dar en los terminales móviles que se conecten con WiFi para determinar el despliegue de los *Puntos de Acceso* (AP, del inglés *Access Point*) por el complejo hotelero, detallando un mapa de cobertura de la infraestructura.

En el capítulo 6 se estudia los diferentes tipos de tráfico que se usan en la WLAN realizando una serie de simulaciones con *ns3* en las cuales se estudian las diferentes versiones de WiFi con diferentes cargas de tráfico para estudiar el desempeño de cada uno y determinar qué tipo de puntos de acceso se usarían en la infraestructura diseñada.

En el capítulo 7 se presenta el pliego de condiciones, en el que se detallan los requisitos necesarios para poder ejecutar la instalación de la infraestructura diseñada y disponer de la conexión inalámbrica analizada, además de una descripción con los costes de la realización del presente proyecto.

En el capítulo 8 se ofrecen algunas conclusiones alcanzadas en el desarrollo del presente proyecto, así como algunas ampliaciones que pueden realizarse en futuras intervenciones en la infraestructura diseñada.

2. WiFi y servicios multimedia

En este capítulo se presentan las WLAN, WiFi, los parámetros de QoS y las aplicaciones multimedia.

2.1 WiFi

Durante bastantes años las redes de ordenadores se llevaban a cabo utilizando soluciones particulares de cada fabricante. Estas soluciones propietarias tenían en gran inconveniente de no permitir interconectar equipos de distintos fabricantes. Cada fabricante desarrollaba su propia solución y la comercializaba por su cuenta. Para el cliente esto suponía estar sometido a un sólo fabricante y a las limitaciones de las soluciones que podía ofrecer.

La única forma de resolver este problema fue desarrollar un sistema normalizado que aceptaran los fabricantes como sistema común. De esa tarea se han encargado organismos internacionales de normalización con la ayuda de los propios fabricantes. No obstante en muchas ocasiones una de las empresas o asociación de empresas ha sido la que ha logrado imponer su sistema en el mercado.

En el caso de las WLAN, el sistema que se ha impuesto es el normalizado por el *Institute of Electrical and Electronics Engineers (IEEE)* 802.11. A esta norma se la conoce más habitualmente por WiFi.

WiFi es una marca de la WiFi Alliance, anteriormente la *Wireless Ethernet Compatibility Alliance (WECA)*, organización comercial que adopta, prueba y certifica que los equipos cumplen los estándares IEEE 802.11 relacionados a las WLAN [7].

Con el sistema WiFi se pueden establecer comunicaciones a altas velocidades a través del aire con un alcance de varios cientos de metros.

Una red basada en WiFi en modo infraestructura tiene principalmente dos componentes:

- *Estación (STA, del inglés Station)*: dispositivo de procesamiento de datos digitales con interfaces de WLAN.

- *Punto de Acceso (AP, del inglés Access Point)*: ejecutan la función de puente inalámbrico al cable y permiten a las estaciones acceder a las redes cableadas a las que están conectados.

2.1.2 Topologías de red

El WiFi aporta una gran versatilidad y flexibilidad a la hora de configurar una WLAN. La gran variedad de configuraciones posibles permite que este tipo de redes se adapte a cualquier necesidad. Dentro de esa gran variedad podemos distinguir dos grandes grupos:

- *Ad hoc*: también conocidas como redes *Peer to Peer* o *independientes*; se permite exclusivamente comunicaciones directas entre los diferentes terminales que forman la red. Para estas redes no existe AP.
- *Infraestructura*: se añade un AP que realiza las funciones de coordinación centralizada de la comunicación entre los distintos STA.

Un AP tiene un rango de cobertura limitado, con lo que no puede proporcionar cobertura por sí solo a grandes áreas. Por ello para crear redes inalámbricas que tengan como objetivo cubrir zonas muy extensas se debe recurrir a la instalación de varios AP.



Figura 2.1. Topología Ad Hoc



Figura 2.2. Topología infraestructura

Esto provoca que un terminal que esté conectado a un AP en un momento determinado se cambie a otro cuando las condiciones lo requieran para poder mantener la conexión a Internet en un proceso que se llama intinerancia.

2.2 Estándar IEEE 802.11

El IEEE 802.11, cuya revisión fue aprobada en junio de 1977 define el funcionamiento e interoperabilidad de las WLAN. La especificación ha elegido la banda *Industrial, Científica y Médica ISM* (ISM, del inglés *Industrial, Scientific and Medical*) para la definición del estándar de WLAN, garantizando su validez global por ser una banda disponible a nivel mundial. La banda ISM es para uso comercial sin licencia, limitando la potencia de transmisión para las WLAN a 100mW.

IEEE 802.11 es miembro de la familia IEEE 802, una serie de especificaciones para tecnologías de las WLAN. En la Figura 2.3 [9] se representan las relaciones existentes entre los diversos componentes de la familia IEEE 802 y su lugar en el modelo de *Interconexión de Sistemas Abiertos (OSI, del inglés Open System Interconnection)*.

Las especificaciones IEEE 802 se centran en los dos niveles inferiores del modelo OSI debido a que incorporan tanto componentes físicos como enlace de datos.

El método de acceso al medio es mediante *Acceso Múltiple con Escucha de Portadora y Evasión de Colisiones (CSMA/CA, del inglés Carrier Sense Multiple Access with Collision Avoidance)*.

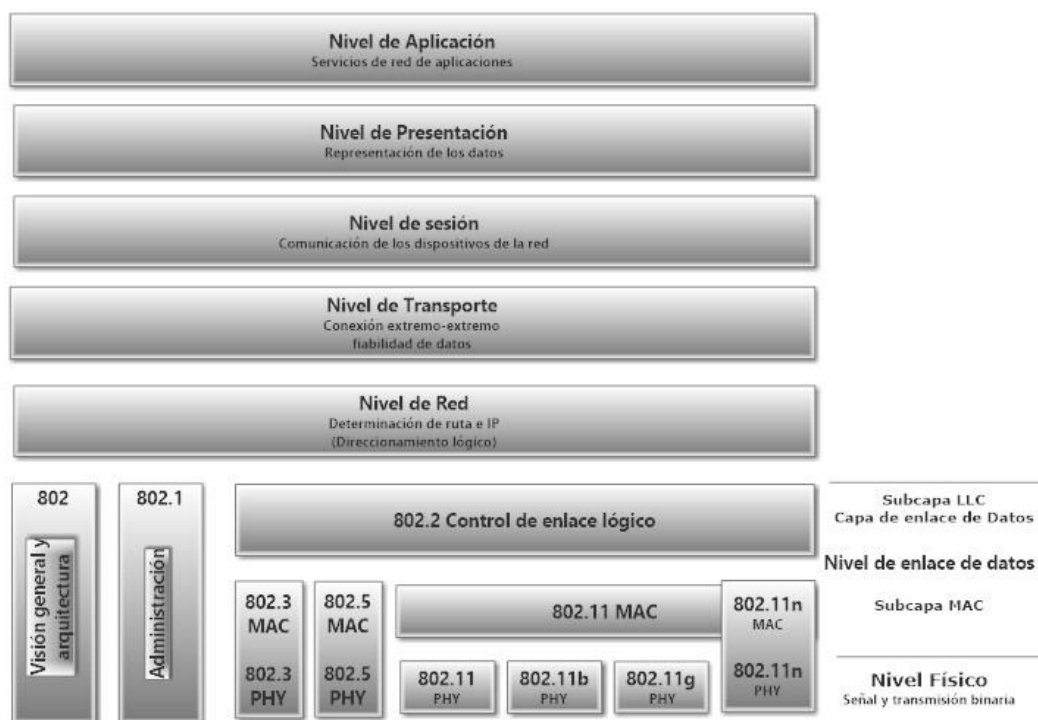


Figura 2.3. Relaciones entre los componentes de IEEE 802 y el modelo OSI

El nivel de *Control de Acceso al Medio* (MAC, del inglés *Medium Access Control*) del IEEE 802.11 contiene un conjunto de reglas para determinar cómo se accede al medio físico y se envían los datos mediante un conjunto de protocolos que controlan el acceso de las STA al espectro radioeléctrico. Los detalles de la transmisión y recepción los proporciona el nivel físico. En la Tabla 2.1 se representan los principales servicios que presta el nivel MAC.

Servicio MAC	Definición	Tipo de STA
Autenticación	Comprueba la identidad de una estación y la autoriza a asociarse	Terminales y puntos de acceso
Desautenticación	Cancela la autenticación existente	Terminales y puntos de acceso
Asociación	Asigna el terminal al punto de acceso	Puntos de acceso
Desasociación	Cancela una asociación existente	Puntos de acceso
Reasociación	Transfiere una asociación entre dos puntos de acceso	Puntos de acceso
Privacidad	Evita el acceso no autorizado a los datos	Terminales y puntos de acceso
Distribución	Asegura la transferencia de datos entre estaciones de distintos puntos de acceso	Puntos de acceso
Entrega de datos	Facilita la transferencia de datos entre estaciones	Terminales y puntos de acceso
Integración	Facilita la transferencia de datos entre redes WiFi y no WiFi	Puntos de acceso

Tabla 2.1. Servicios de la capa MAC

La especificación IEEE 802.11 incluye la MAC IEEE 802.11 y los diversos niveles físicos que dan nombre a las diferentes especificaciones IEEE 802.11b y IEEE 802.11g. Para la especificación IEEE 802.11n a parte de cambios en el nivel físico se han realizado una serie de mejoras en la MAC de la especificación.

IEEE 802.11b

La versión IEEE 802.11b del estándar original fue ratificada en 1999. Permite una velocidad máxima de transmisión de 11Mbps y utiliza el mismo método CSMA/CA definido en el estándar original. Funciona en la banda de 2,4 Ghz, usa el método de codificación y modulación *Espectro Ensanchado por Secuencia Directa (DSSSS* del inglés *Direct Sequence Spread Spectrum*). En la práctica, la velocidad de transmisión es se encuentra entre 6 y 7 Mbps.

IEEE 802.11g

La versión del estándar IEEE 802.11g se ratificó en el 2003. Al igual que el estándar IEEE 802.11b hace uso de la banda de 2,4Ghz pero opera a mayores velocidades, 54Mbps como máxima teórica y 24Mbps como velocidad real de transferencia. Puede usar dos métodos de codificación y modulación; *Multiplexación por División de Frecuencias Ortogonales (OFDM* del inglés *Orthogonal Frequency Division Multiplexing*) y DSSS. Mantiene la compatibilidad con el estándar IEEE 802.11b y utiliza las mismas frecuencias. Sin embargo, en redes configuradas con el estándar IEEE 802.11g la presencia de nodos bajo el estándar IEEE 802.11b reduce significativamente la velocidad de transmisión del conjunto.

IEEE 802.11n

El estándar IEEE 802.11n fue ratificado en septiembre del 2009 por el IEEE. Fue diseñado para reemplazar por completo la actual tecnología alámbrica Ethernet y convertirse en la tecnología dominante en LAN.

Introduce varias mejoras a nivel físico y MAC mejorando el throughput y confiabilidad. Entre estas mejoras se incluye:

- *Uso de dos bandas de frecuencias:* IEEE 802.11n permite el uso de la banda de 2,4Ghz y de la de 5Ghz. Estando esta última menos saturada en zonas urbanas.
- *Aumento de canales no solapados:* en la banda de 5Ghz se dispone de un mayor número de canales no solapados que en la de 2,4Ghz.
- *Aumento de ancho de canales:* modulación con canales de 40 MHz, aunque también puede trabajar con canales de 20Mhz manteniendo la compatibilidad con IEEE 802.11 b/g, en los que el ancho de canal es de 20 MHz.
- *Múltiple Entrada y Múltiple Salida (MIMO, en inglés Multiple Input Multiple-Output):* sistema de radio con múltiples entradas al receptor y múltiples salidas del transmisor capaz de enviar o recibir múltiples cadenas de datos simultáneamente. Los estándares IEEE 802.11b y IEEE 802.11g usan la tecnología de *Simple Entrada Simple Salida (SISO en inglés Single Input and Single Output)* con una entrada al receptor y una salida del transmisor.
- *Agregación de Tramas:* reduce la transmisión de encabezados a nivel físico ya que envía varias tramas de datos como parte de una sola transmisión.
- *QoS:* mejoras MAC con mecanismos de QoS basados en el estándar IEEE 802.11e.
- *Intervalo de guarda:* acorta el intervalo de guardia en aquellos entornos donde el efecto de multitrayecto se minimiza.

Se pueden alcanzar velocidades teóricas de 600 Mbps, lo que en la práctica se traduce en unos 100Mbps. En la Tabla 2.2 [10] se especifican las diferentes velocidades según cómo se configure los diferentes parámetros.

SISO			MIMO		
Velocidades IEEE 802.11n Requeridas (canal de 20 MHz)	Velocidades IEEE 802.11n Requeridas (canal de 40 MHz)	Intervalo de Guarda Corta Habilitado	Dos Cadenas	802.11n Velocidad Requerida (canal de 40 MHz)	Intervalo de Guarda Corta Habilitado
6.5	13.5	15	13	27	30
13	27	30	26	54	60
19.5	40.5	45	39	81	90
26	54	60	52	108	120
39	81	90	78	162	180
52	108	120	104	216	240
58.5	121.5	135	117	243	270
65	135	150	130	270	300

Tabla 2.1. Velocidades de IEEE 802.11n

2.3 Calidad de servicio

La QoS está estrechamente relacionada con la percepción que tiene el usuario final sobre el desempeño de la red ante una aplicación y se podría definir como la capacidad de proporcionar prioridad diferente para distintos flujos de datos garantizando un cierto nivel de rendimiento de las aplicaciones que los generan.

La QoS se basa, principalmente en una serie de parámetros cuantificables y que nos permiten determinar un correcto funcionamiento de las aplicaciones que dependen de ellos, dichos parámetros son:

- *Throughput* o caudal: tasa de transmisión de datos dentro de un canal de red, que hace referencia al ratio de datos y por ello se mide en bps. El recurso de WLAN que se manipula para mejorarlo suele ser el ancho de banda destinado a las aplicaciones. Las necesidades de throughput depende de las aplicaciones que generen tráfico en la red y suelen depender fundamentalmente de dos parámetros que son:
 - Tamaño del paquete de información.

- Velocidad con la que se transmite.

Puede ser variable o constante, dependiendo de la aplicación que lo genere.

- **Retraso:** medida de la duración de tiempo que se necesita para que un paquete vaya del nodo transmisor al nodo receptor. Esta cantidad de tiempo, también conocida como latencia, tiene un mínimo absoluto que es una función de los protocolos de red y los medios físicos por donde se propagan los datos. La cuantificación más importante controlar es el retraso que provoca la congestión de la WLAN. Para los paquetes de voz este retraso nunca debe de exceder los 150ms, que aún siendo un tiempo bastante corto, es muy difícil de controlar, en especial cuando un nodo debe esperar continuamente para enviar datos debido a altas cantidades de tráfico. Los retrasos grandes causan errores de transferencia y frustración durante la realización de tareas interactivas. Todos los componentes de la red añaden retraso a la comunicación.
- **Jitter o variación del retraso:** variación que sufren los retrasos de los diferentes paquetes que fluyen por la red. Tiene un alto impacto en la QoE percibida en la transmisión por el usuario. Puede ser el resultado de una red congestionada, en especial una que está saturada con tráfico intenso como el de transferencia de archivos o vídeo en alta definición. Para el tráfico de voz debe siempre mantenerse por debajo de 30ms.
- **Perdidas:** cantidad de paquetes que se pierden durante la transmisión y normalmente se expresa como un porcentaje o paquetes perdidos por segundo. La pérdida de paquetes es reconocida por el nodo encargado de la recepción, lo que fuerza a realizar un nuevo envío por parte del nodo emisor. Cuando se trata de datos de aplicaciones clásicas como la navegación o la transferencia de archivos, las pérdidas no suelen ser un gran problema, en cambio con los datos de voz existe poco margen de tiempo para poder reinsertar en paquete reenviado en la secuencia adecuada, justo en el momento correcto de la conversación. El tráfico de voz, debido a su naturaleza, tiene los requerimientos más estrictos para pérdidas de paquetes, que deben de ser en todo momento inferior al 1% para que no se degrade la señal de voz.

Las WLAN deben de garantizar siempre un cierto grado de QoS para las aplicaciones de los usuarios que hacen uso de las mismas. Por lo que la red debe manejar los recursos con el objetivo de optimizar los valores de los parámetros anteriores mediante mecanismos propios

2.4 Aplicaciones multimedia

Hoy en día se está produciendo una unificación de las redes clásicas diseñadas para diferentes objetivos, como son las redes telefónicas, diseñando una única red que se encargue de la transmisión y recepción de cualquier tipo de información buscando principalmente una simplificación en su mantenimiento y ahorro de costes.

Ante este hecho es especialmente importante tener en cuenta cómo las aplicaciones multimedia hacen uso de los recursos de esta red ya que por el tipo de información que manejan, es bastante diferente a los de los servicios más clásicos como son la navegación por Internet o la transferencia de archivos.

El crecimiento exponencial de este tipo de servicios fuerza a una correcta planificación de las redes que van a ser destinadas a su uso, en las cuales se debe garantizar mediante, mecanismos definidos, su correcto desempeño.

2.4.1 VoIP

La VoIP es sin lugar a dudas, uno de los desarrollos tecnológicos que más rápidamente están siendo adoptados hoy en día por los usuarios y por las empresas. Unas de las principales razones de ésta rápida migración es debida a la flexibilidad de IP, por la que se transporta la voz en paquetes, permitiendo estar siempre disponibles para recibir o emitir una llamada sin estar atados a un punto físico de la red incentivando el nomadismo.

Uno de sus grandes inconvenientes es la posible falta de QoS debido a la naturaleza neutral de las redes basadas en IP y a las transformaciones que sufre la señal

de audio, pero con la evolución de los sistemas ésta es cada vez mejor, similar a la del teléfono convencional, y permite la interconexión. Por este motivo se debe de garantizar, en las redes que realicen uso de aplicaciones de VoIP, unos parámetros de QoS adecuados.

Definido por primera vez en 1996 por la *Unión Internacional de Telecomunicaciones* (ITU, del inglés *International Telecommunication Union*) mediante la recomendación H323, el estándar VoIP ha proporcionado a los fabricantes una serie de normas a seguir. La recomendación H.323 por su estructura proporciona las siguientes ventajas:

- Permite el control del tráfico de la red, por lo que se disminuyen las posibilidades de que se produzcan caídas importantes en el rendimiento de las redes de datos.
- Proporciona el enlace a la red telefónica tradicional.
- Al tratarse de una tecnología soportada por las redes IP y dada la naturaleza de éstas, presenta las siguientes ventajas adicionales:
 - Es independiente del tipo de red física que lo soporta (hardware utilizado).
 - Permite la integración con las redes IP actuales.

En la actualidad la recomendación ha sido desplazada por el *Protocolo de Inicio de Sesiones* (SIP, del inglés *Session Initiation Protocol*), que comparte dichas ventajas con éste además de aportar un mayor grado de eficiencia en su desempeño.

Componentes de la VoIP

La VoIP necesita de una serie de elementos que permita realizar las diferentes transformaciones que realiza la señal de audio en su recorrido por las diversas redes telefónicas convencionales y de datos por la que transita.

Los problemas de transmitir la voz por las redes de datos surge en la necesidad de recibir los paquetes en un determinado orden, conseguir que no exista ningún tipo de pérdida y mantener una mínima tasa de transmisión.

El propio estándar define tres elementos que se encargan de los procesos necesarios para la comunicación y fundamentales en su estructura:

- *Terminales*: son los sustitutos de los actuales teléfonos y se encargan del tratamiento de la señal de voz para su transmisión a la red, realizando la captación, digitalización, y compresión de la misma. Existen dos formas de implementar los terminales de la arquitectura:
 - *Hardware*: con una estética similar a los teléfonos convencionales.
 - *Software*: es una aplicación desarrollada para la comunicación de VoIP que se puede instalar en un computador convencional.
- *Gatekeepers*: son el centro de toda la organización VoIP, y serían el sustituto para las actuales centrales de telefonía. Normalmente implementadas en software, en caso de existir, todas las comunicaciones pasarían obligatoriamente por él. La función de un Proxy VoIP o servidor, es proveer funciones de administración de llamadas centralizadas, especialmente en entornos empresariales.
- *Gateways*: se trata del medio por el cual se comunica con la red telefónica tradicional, actuando de forma transparente para el usuario.

Con estos tres elementos se puede realizar toda la comunicación de que se produce de una forma totalmente transparente en ambos sentidos, pudiendo recibir y emitir

llamadas directamente desde nuestro computador personal sin ningún problema. El la Figura 2.4 que representa la arquitectura de una red de VoIP.

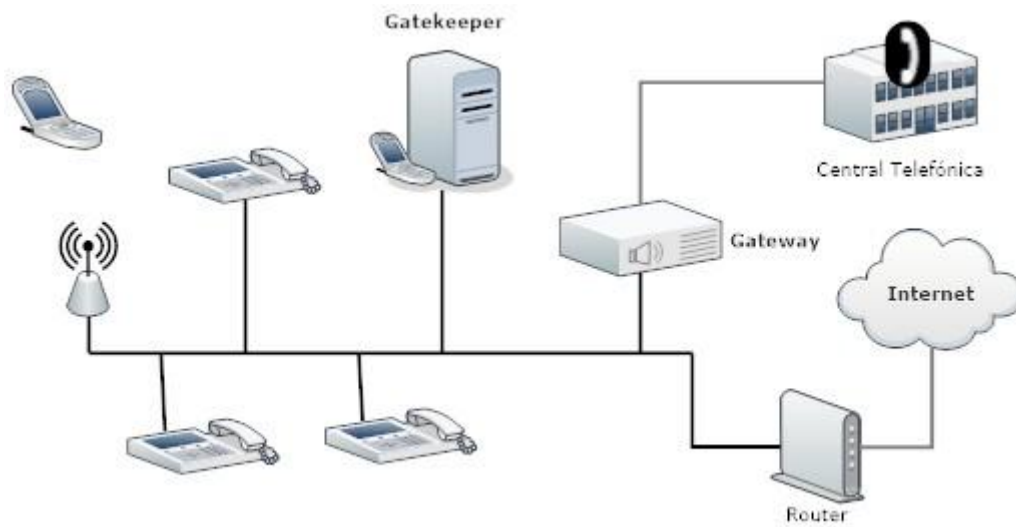


Figura 2.4. Arquitectura VoIP

3. Herramientas usadas

En este capítulo se presentan las características generales y prestaciones de las herramientas usadas, así como su preparación para afrontar la realización de las diferentes simulaciones.

3.1 Introducción

Para realizar el presente PFC se ha recurrido a una serie de herramientas que nos han permitido realizar las simulaciones de los diferentes escenarios.

- VirtualBox.
- Ubuntu.
- Simulador ns3.

VirtualBox y Ubuntu se han seleccionado debido a los requerimientos de ejecución del simulador seleccionado y los equipos disponibles para el desarrollo. Se han optimizado para una ejecución eficiente, debido a la ingente cantidad de recursos de computación que necesitan los simuladores de red.

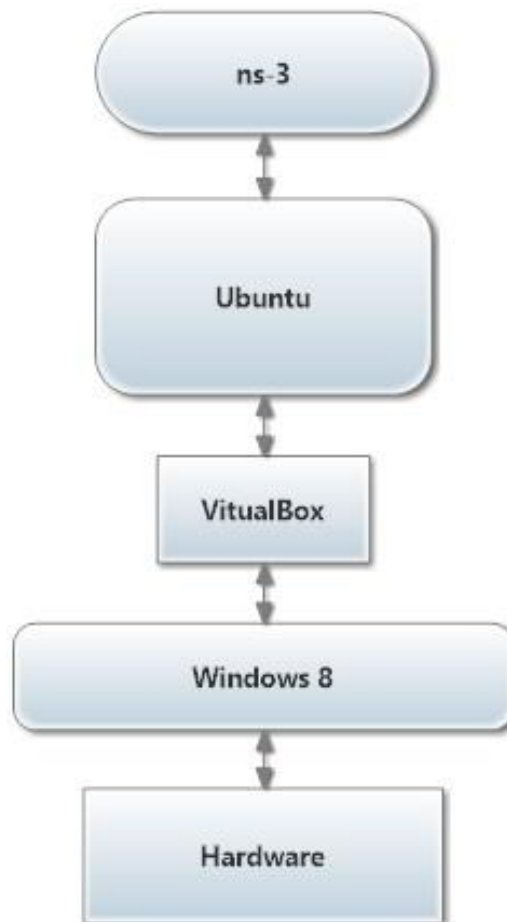


Figura 3.1. *Arquitectura de ejecución de las herramientas*

3.2 VirtualBox

Para proceder a la instalación sobre entornos Microsoft © Windows del simulador de red seleccionado se requiere de la instalación previa de una máquina virtual, debido a que Microsoft © Windows carece de soporte para dicha herramienta, para realizar la instalación de un sistema operativo basado en Linux que aporte todas las prestaciones del simulador requeridas para llevar a cabo el presente PFC. Se ha seleccionado VirtualBox debido a su versatilidad, rendimiento y a que está licenciado bajo licencia GPL versión 2. Se puede descargar directamente desde su página web la versión para Microsoft Windows requerida. En la Figura 3.2 [11] aparece una captura de la página de descargas de VirtualBox.

Descargamos la versión de 64 bits para nuestro proyecto, debido a las características del equipo usado para el presente proyecto. En la Figura 3.3 se indican las prestaciones del equipo.



Figura 3.2. Página de descargas de VirtualBox

Sistema	
Procesador:	Intel(R) Core(TM) i7-3632QM CPU @ 2.20GHz 2.20 GHz
Memoria instalada (RAM):	8,00 GB (7,86 GB utilizable)
Tipo de sistema:	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil:	La entrada táctil o manuscrita no está disponible para esta pantalla

Figura 3.3. Prestaciones del equipo usado para las simulaciones

3.2.2 Instalación

Una vez descargado el instalador se procede a ejecutar el asistente de instalación siguiendo los pasos indicados por el mismo. En la Figura 3.4 se muestra una captura del asistente.

Ya instalado el VirtualBox en el sistema se procede a preparar la máquina virtual para la instalación de Ubuntu 12.04 versión con soporte técnico extendido (*TLS*, de sus siglas en inglés *Long Term Support*) sobre el que, finalmente, se instalará el software del simulador.

Se arranca el VirtualBox y se selecciona *nueva* en la pestaña superior de la Figura 3.5.



Figura 3.4. Asistente de instalación de VirtualBox



Figura 3.5. Creamos una nueva máquina virtual

Se selecciona el tipo de sistema operativo que va a hospedar en la pantalla de la Figura 3.6, para el presente proyecto Ubuntu de 64 bits.

Se selecciona *next* en la pantalla de la Figura 3.6 y en la siguiente ventana, representada en la Figura 3.7, se selecciona la cantidad de memoria *Random Access Memory (RAM)* para la máquina virtual, se recomienda la máxima cantidad de memoria RAM posible.

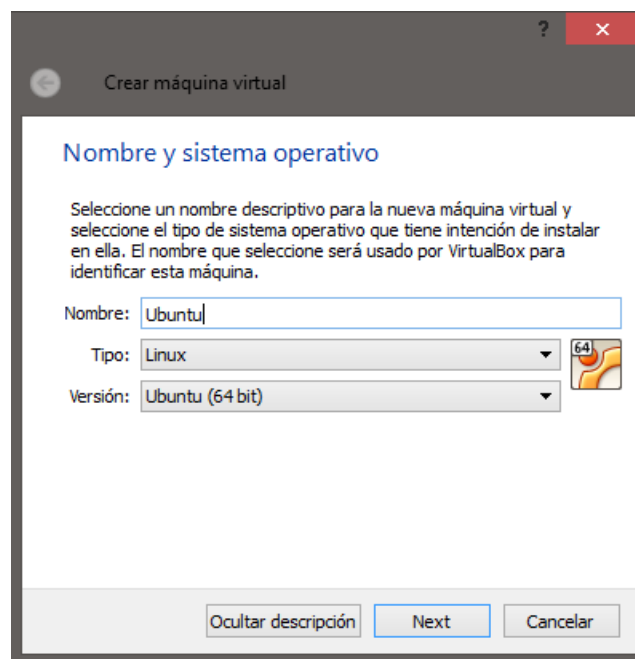


Figura 3.6. Configuración del tipo máquina virtual que se desea crear

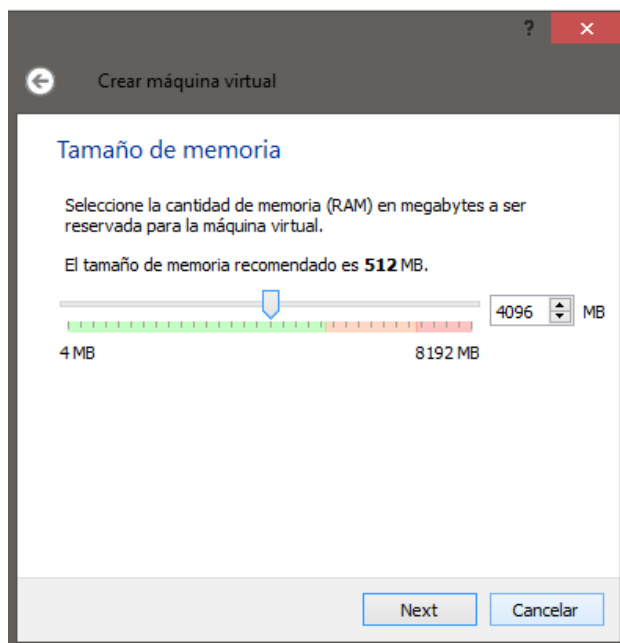


Figura 3.7. Configuración de cantidad de RAM

Definimos la unidad de disco duro para la posterior instalación en el menú de la Figura 3.8.

Se selecciona una unidad de disco duro con tamaño fijo, para optimizar el rendimiento de la máquina virtual, en la pantalla representada en la Figura 3.9.

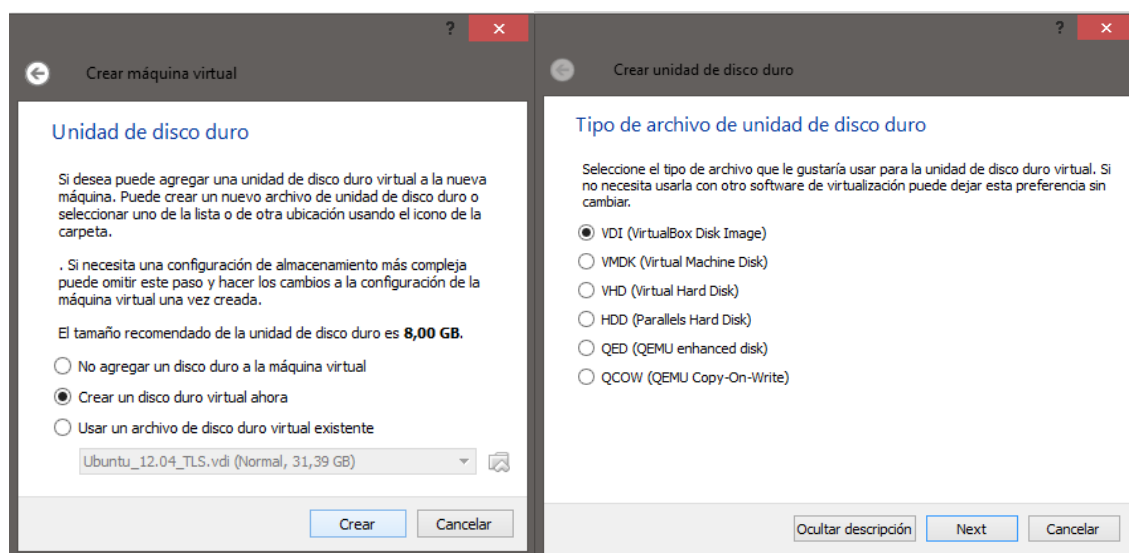


Figura 3.8. Configuración del tipo de disco duro que se desea simular

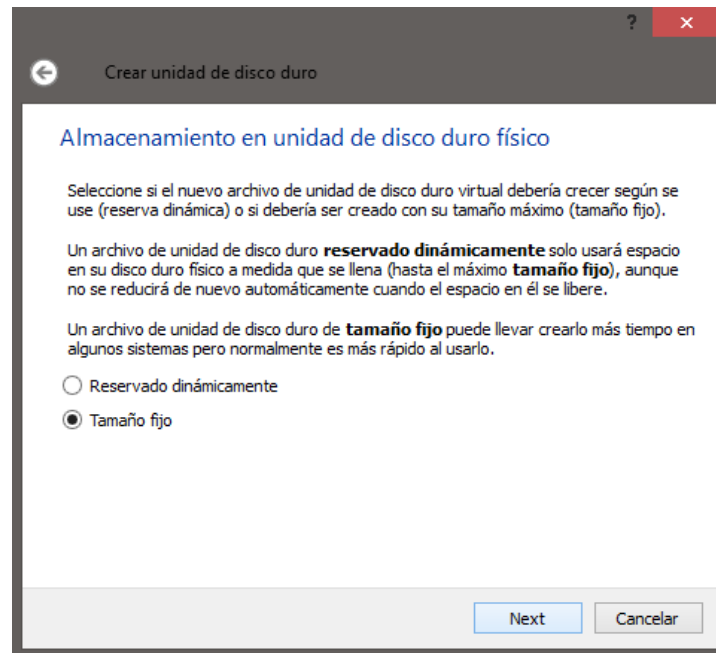


Figura 3.9. Tipo de reserva en el disco duro local para el sistema virtual

Se selecciona el tamaño del disco duro, Figura 3.10, para el sistema huésped, un mínimo de 30GB para tener espacio suficiente para el sistema Linux, el simulador ns3, sus dependencias, herramientas de análisis y los resultados generados en las simulaciones.

Una vez esté todo lo anterior realizado se procederá a la creación de la máquina virtual, apareciendo la ventana de la Figura 3.11 que nos indicará el porcentaje del proceso.

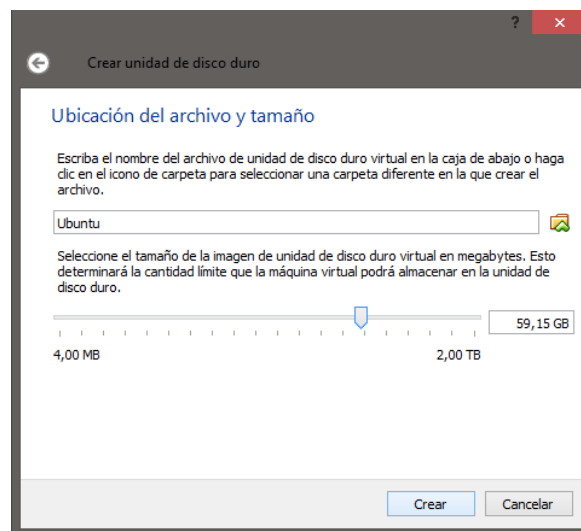


Figura 3.10. Configuración de tamaño de disco duro virtual

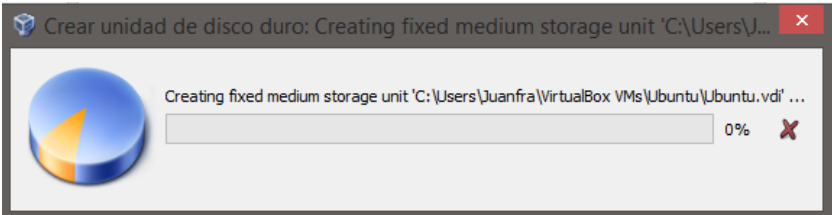


Figura 3.11. Iniciando la creación de la máquina virtual

Configuraciones en la BIOS

Para poder virtualizar sistemas operativos de 64 bits, en la mayoría de los computadores actuales que lo soporten, se debe activar en la *Basic Input/Output System (BIOS)* dicho soporte que suele venir deshabilitado, para ello se arranca el computador y en durante el arranque se ejecuta la secuencia de entrada en la BIOS (normalmente la tecla *Supr* o algún pulsador especificado por el fabricante del equipo).

Una vez dentro la BIOS del sistema se activa el soporte de virtualización que se indica en la Figura 3.11 seleccionando *enabled*.

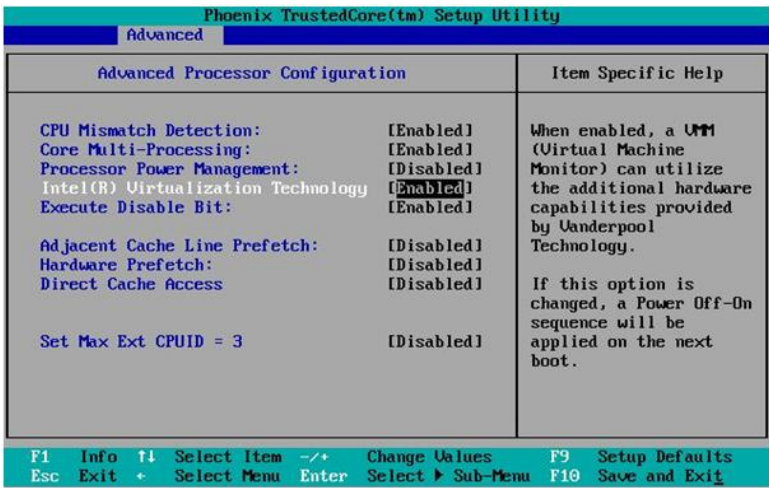


Figura 3.12. Panel de configuración de la BIOS

Optimizar la máquina virtual

Una vez creada la máquina virtual se procede a optimizarla para un mejor rendimiento, debido a que se usará un procesador i7 de 4 núcleos se selecciona la opción que permite usar todos los núcleos a la vez, y no solo uno como viene predeterminado. Para acceder a dichas opciones se hace clic con el botón derecho sobre el icono de la máquina virtual que creada y se selecciona la opción configuración, una vez en configuración se selecciona la opción de sistema y dentro de sistema la solapa procesador, aparecerá el siguiente en recuadro de la Figura 3.13 y se selecciona la máxima cantidad de procesadores que permita la configuración.

3.3 Ubuntu 12.04 TLS

El simulador ns3 puede correr en diversas plataformas, teniendo un fuerte soporte en las basadas en UNIX y un soporte muy básico (o prácticamente nulo en las últimas versiones) para las plataformas basadas en sistemas Microsoft Windows mediante el software Cygwin. Por dicha razón para la ejecución de las simulaciones realizadas en este estudio se ha escogido un sistema operativo basado en GNU/Linux como es Ubuntu. Se ha seleccionado la versión 12.04 LTS para procesadores con arquitectura de 64 bits. Los sistemas operativos Ubuntu operan bajo *Licencia Pública General* (GPL, del inglés General Public License).

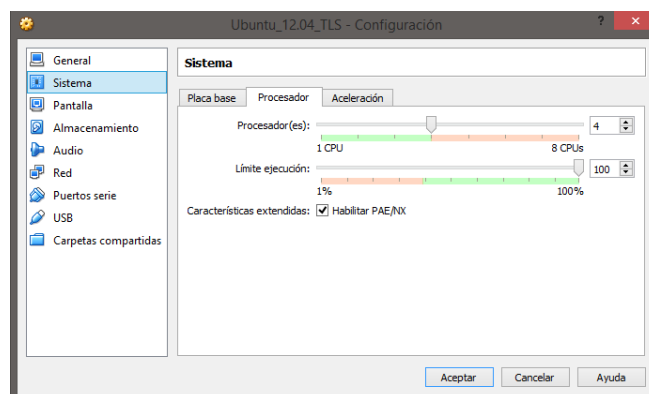


Figura 3.13. Configuración de los procesadores de la máquina virtual

3.3.1 Instalación

Para la instalación de Ubuntu sobre la máquina virtual ya instalada se procede a la descarga de la distribución seleccionada, versión Desktop, de su página web [13], siguiendo las especificaciones del equipo donde se va a realizar el desarrollo, comentada en el apartado anterior.

Una vez descargada la imagen *ubuntu-12.04-desktop-amd64.iso* desde la página web de Ubuntu se procede a su instalación en la máquina virtual, preparada con anterioridad. Para ello se selecciona la máquina virtual preparada en VirtualBox y nos aparecerá una ventana como la de la Figura 3.14 indicando que carguemos una imagen para la instalación del sistema, se escoge la imagen descargada desde la página web de Ubuntu y se selecciona la opción iniciar.

Arranca el proceso de instalación del sistema operativo con la pantalla de bienvenida del asistente representada en la Figura 3.15.

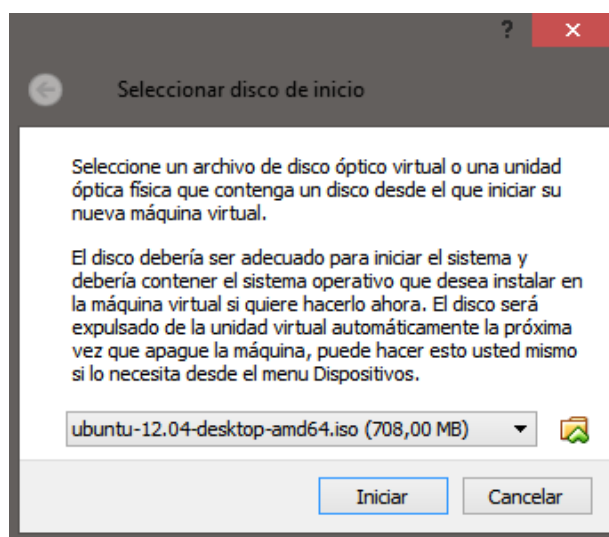


Figura 3.14. Selección de la imagen.iso para la instalación de Ubuntu



Figura 3.15. Iniciamos el proceso de instalación con el asistente

Se selecciona la opción instalar Ubuntu y se inicia el proceso de configuración de la instalación, seleccionando en ventana de la Figura 3.16 todo el disco duro virtual dimensionado en el apartado anterior de VirtualBox.

Finalmente se introduce los datos que se solicitan cuando aparezca la ventana representada en la Figura 3.17.

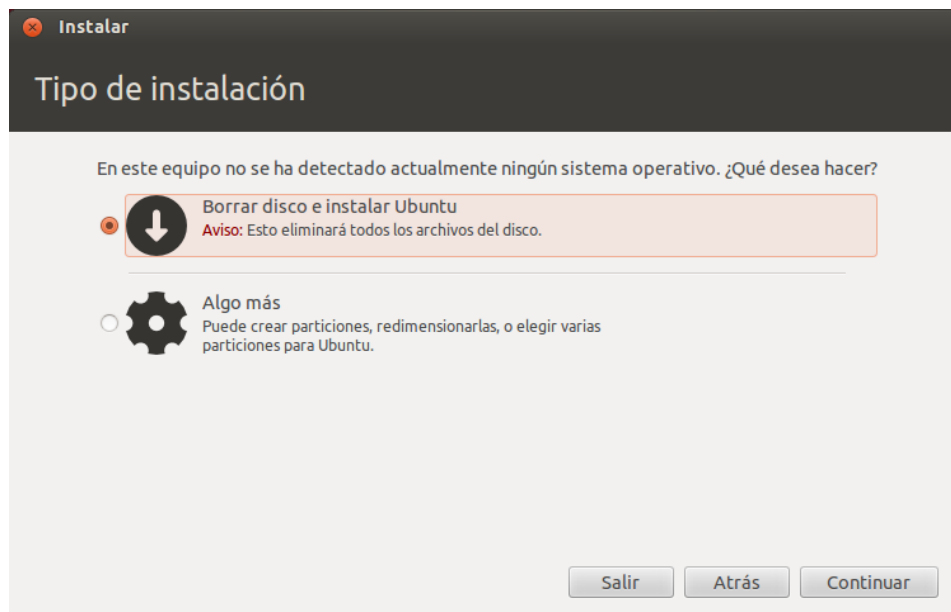


Figura 3.16. Se selecciona Borrar disco e instalar Ubuntu

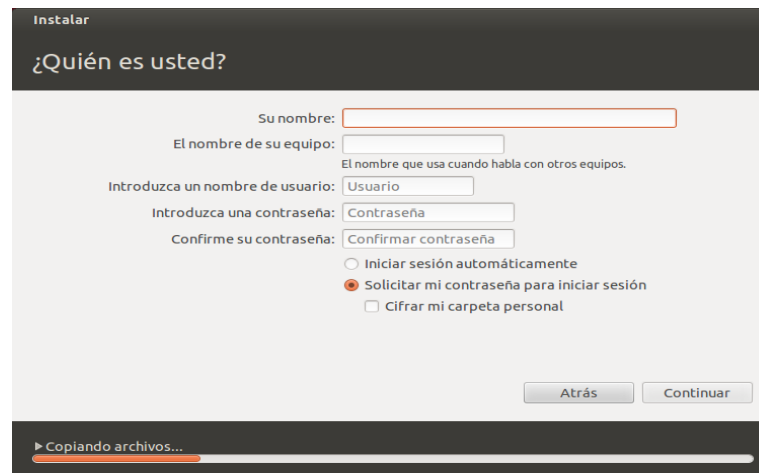


Figura 3.17. Configuración de usuario

Una vez finalizado el proceso anterior ya se tiene el sistema operativo Ubuntu preparado para comenzar la instalación de ns3.

3.4 Simulador ns3

Network Simulator es el nombre de una serie de simuladores de red basados en tiempo discreto (ns1, ns2 y ns3). La primera versión fue desarrollada entre 1995 y 1997 por el *Lawrence Berkeley National Laboratory* y estaba basado en el simulador *REAL*. Es Software Libre bajo licencia GNU GPLv2 y actualmente el proyecto está mantenido por una serie de investigadores, instituciones y voluntariado.

Actualmente ns2 es la versión del simulador más usada en ámbitos académicos y contiene una gran cantidad de módulos para todo tipo de redes. La versión ns3 es relativamente nueva desarrollada con el objetivo de solventar las carencias de ns2, se espera que en un futuro lo reemplace, como pasó anteriormente con ns1.

Está programado en su totalidad en C++, mediante la *Programación Orientada a Objetos* (OOP, del inglés *Object Oriented Programming*). Dirigido, principalmente al ámbito académico y de investigación su primera versión fue lanzada en julio de 2008 y la

última hasta la fecha 3.19 en diciembre de 2013. Siendo el sucesor del simulador ns2, ns3 fue creado desde cero buscando aumentar la fidelidad de las simulaciones con la realidad y un mayor orden de las bibliotecas que permitan poder modificarlo con una mayor facilidad.

El núcleo de ns3, al igual que en las versiones anteriores de Network Simulator, está organizado en bibliotecas C++. Cada una de éstas contiene las clases necesarias para simular un programa, protocolo u objeto en particular, como podría ser una aplicación que genera tráfico de paquetes de datos, una red inalámbrica, con sus diferentes componentes, o un enlace punto a punto ofreciendo un entorno virtual capaz de modelar el comportamiento de todo tipo de redes y de aplicaciones, dando como resultado simulaciones de tráfico de red bastante próximo al real.

La programación de ns3 permite modelar, gestionar y someter a pruebas de prestaciones la infraestructura de red diseñada.

Para simular una red, es necesario crear un script en C++ (o en Python ya que exporta casi la totalidad de su API a dicho lenguaje), en el cual se definen todos los objetos a utilizar, sus parámetros, el vínculo entre ellos y las características generales de la simulación.

El código de las simulaciones programadas con ns3 sigue un orden lógico que se representa en el diagrama de la Figura 3.18.

3.4.1 Descripción del Simulador

En esta sección se explica los modelos abstractos del simulador ns3 para una mejor comprensión de su funcionamiento y de las simulaciones en los capítulos posteriores y se repasan algunos términos que se usan comúnmente en redes pero en el ámbito más concreto del simulador.

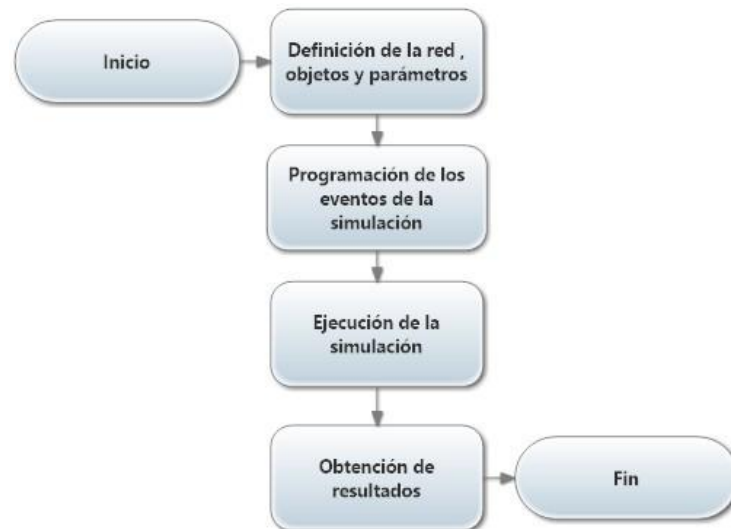


Figura 3.18. Creación de una simulación en ns3

Clases de abstracción

Conceptos básicos para modelar las abstracciones de un sistema de comunicación que tienen como objetivo la comprensión del funcionamiento de ns3.

- *Nodo*: se usa el término de nodo debido a que se habla de un simulador de redes y no un simulador de Internet en el que se suelen usar términos como anfitrión o sistema final. La abstracción nodo en ns3 es una abstracción que representa un sistema informático completo y proporciona métodos para administrar las representaciones de dichos sistemas.
- *Aplicación*: abstracción básica en ns3 para la simulación de programas y proporciona métodos para administrar las representación de diversidad de aplicaciones a nivel usuario. Para las aplicaciones desarrolladas en el presente PFC se usa una serie de clases de aplicaciones de ns3 para generar tráfico de paquetes de diferentes tipo como pueden ser vídeo o VoIP. Dichas abstracciones están compuestas por un conjunto cliente/servidor, origen y destino de los paquetes simulados.
- *Canal*: representa los flujos de datos sobre las redes simuladas. Dicha abstracción proporciona métodos para gestionar los objetos de comunicación de la red

simulada y conectar los nodos a ellos. Un canal especializado puede simular algo tan simple como un cable de red y algo tan complicado como un espacio tridimensional lleno de obstáculos en el caso de las redes inalámbricas.

- *Dispositivo de red:* para conectar los diferentes nodos a la red se requiere un determinado tipo medio de comunicación y una tarjeta de red, que pueda interactuar con dicho medio, la cual tiene que ser instalada en el nodo. Abarca el software y el hardware simulado de una tarjeta de red es el dispositivo de red. Un dispositivo de red está instalado en un nodo con el fin de permitir que éste se comunique con otros nodos en la simulación a través de canales. Un nodo puede tener varios dispositivos de red para comunicarse con varios canales como pasaría con un ordenador en el mundo real.

En la Figura 3.19 se describe el diagrama de abstracciones y como se simula el flujo de datos, el cual es modificado por cada uno de los componentes que forman parte de las abstracciones de ns3.

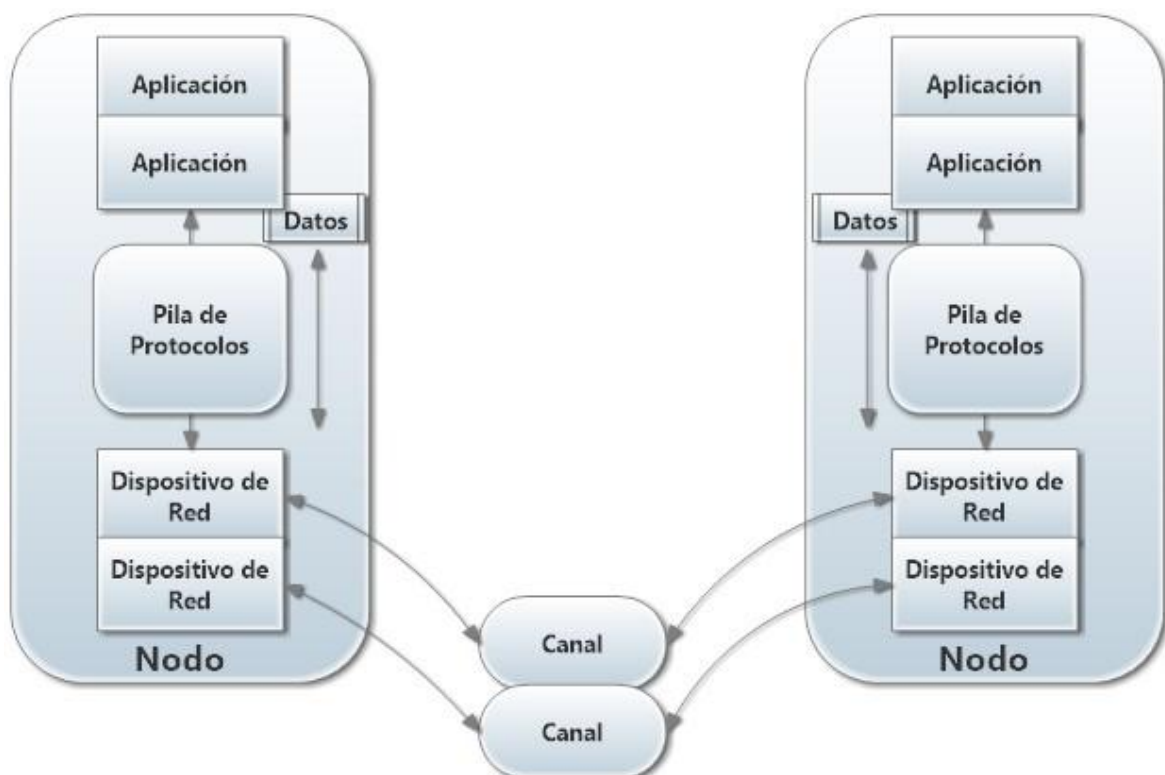


Figura 3.19. Abstracciones de ns3

3.4.2 Ayudantes de topología

Los scripts de simulación en ns3 pueden ser escritos enteramente con la API de C++ en bajo nivel implementada en el simulador, esto permite un control total de todos los objetos y parámetros de la simulación. Sin embargo, la complejidad de un script escrito de esta forma hace que la programación del mismo se vuelva larga, tediosa de escribir y complicada de entender.

Para facilitar dicho proceso viene implementada en el simulador una API ayudante llamada *Helpers*, encapsulando la API de bajo nivel principal, que facilitan la realización de la simulación. A modo de ejemplo, se pueden usar para operaciones básicas a la hora de crear un dispositivo de red, instalarlo en una red de nodos, configurar la pila de protocolos de dicho nodo y conectarlo a un Canal, facilitando la creación de redes con multitud de nodos y canales.

Los ayudantes de topología son muy eficientes a la hora de manejar y configurar muchos objetos a la vez, ofreciendo una interfaz muy amigable a la hora de realizar la programación de la simulación.

3.4.3 Organización del software

El código del simulador se organiza, principalmente en el directorio `src/` dentro de la carpeta principal del simulación, y su organización se describe en la Figura 3.20.

La parte principal y más importante es el núcleo que se implementa en el directorio `src/core` y contiene aquellos componentes que son comunes en todos los modelos de protocolo, hardware y entornos simulados. La gestión de los paquetes son fundamentales en cualquier simulador de red, siendo el objeto de estudio más importante en la inmensa mayoría de las simulaciones, dicha gestión se implementa en el directorio `src/network`.

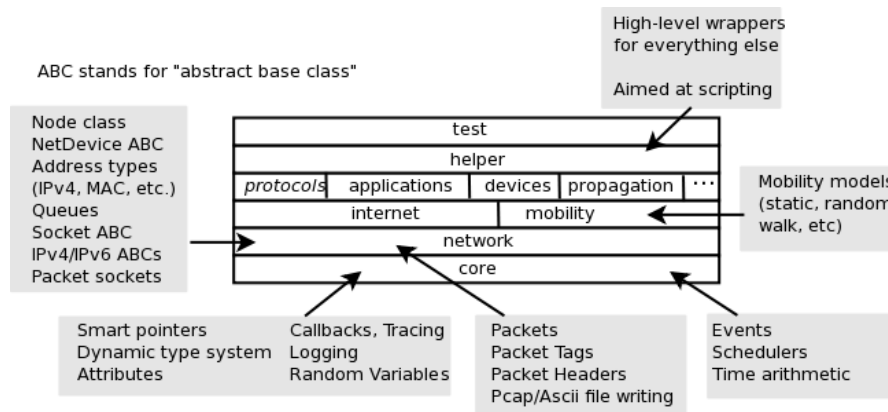


Figura 3.20. Organización de módulos en ns3

Estos dos módulos son la base de cualquier simulación genérica y son independientes de los modelos de red y de los dispositivos específicos de cada simulación.

El resto de módulos como *mobility*, *devices*, *propagation*... son más específicos de cada simulación y sus requerimientos particulares, pudiendo hacerse uso o no dependiendo de las necesidades, topologías, objetos de estudio...

3.4.4 Instalación

Estando la máquina virtual de VirtualBox funcionando correctamente con la distribución de Ubuntu 12.04 LTS se procede a la instalación del simulador ns3, dicha instalación puede llegar a ser bastante tediosa debido a la cantidad de dependencias que requiere y a la complejidad del software, por lo que se procede a hacer 3 claras distinciones en el proceso de instalación:

- Instalación de dependencias.
- Instalación del simulador.
- Test de validación de módulos instalados.

Instalación de dependencias

Para la instalación de dependencias del simulador se recurre a una herramienta de Ubuntu llamada *apt-get*, que es un repositorio que descarga las dependencias indicadas mediante la línea de comando de Linux.

Los requisitos mínimos para el funcionamiento del simulador son los paquetes de *gcc*, *g++* y *python* para los módulos más básicos. Pero para el presente proyecto se hace uso de una serie de módulos necesarios para la simulación los medios de propagación, creación de flujo de datos, funcionalidades en los nodos, recolección de datos de salida, representación y depuración de la simulación. Los paquetes requeridos se instalan mediante el terminal de Ubuntu con la orden de comandos especificada a continuación:

```
juanfra@juanfra-VirtualBox:~$ sudo apt-get install gcc g++ python gcc g++ python python-dev
mercurial bzip2 gdb valgrind gsl-bin libgsl0-dev libgsl0ldbl flex bison libfl-dev tcpdump sqlite sqlite3
libsqlite3-dev libxml2 libxml2-dev libgtk2.0-0 libgtk2.0-dev vtun lxc uncrustify doxygen graphviz
imagemagick texlive texlive-extra-utils texlive-latex-extra python-sphinx dia python-pygraphviz
python-kiwi python-pygoocanvas libgoocanvas-dev libboost-signals-dev libboost-filesystem-dev
openmpi-bin openmpi-common openmpi-doc libopenmpi-dev gcc-multilib unrar p7zip-full cvs git
dh-autoreconf libqt4-core libqt4-dev libqt4-gui qt4-dev-tools cmake python-pygccxml
```

Con el comando anterior se solucionan todas las dependencias del software de ns3 de los módulos requeridos para el presente PFC procediendo al proceso de instalación con todas las garantías de éxito.

Instalación del simulador

Para poder proceder a la instalación del simulador previamente se debe descargar el software desde la página oficial del proyecto [2] en la sección *releases*. Para el presente proyecto se han realizado las simulaciones para la versión 3.19 de ns3.

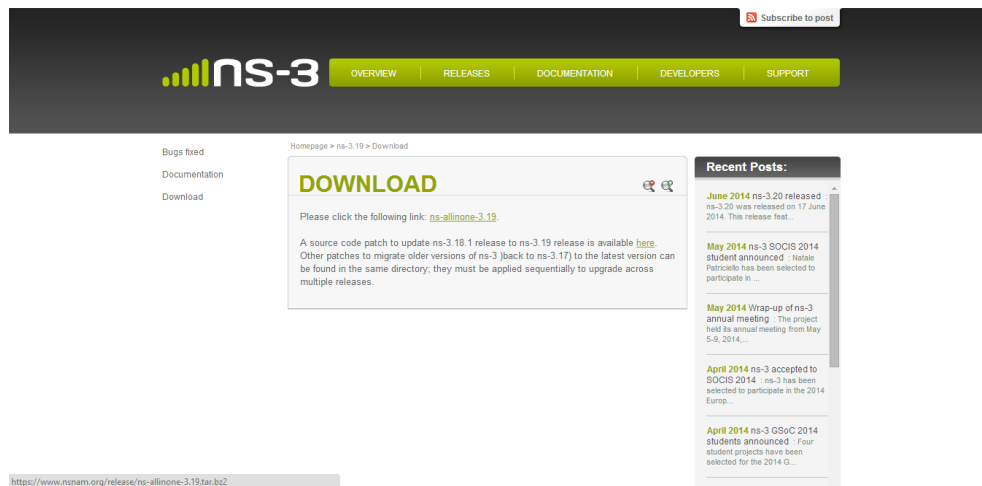


Figura 3.21. Página de descargas de ns3

Se procede a la descarga del archivo *ns-allinone-3.19.tar.bz2*, una vez descargado se descomprime en una carpeta del sistema. Una vez descomprimido se accede al directorio descomprimido *ns-allinone-3.19* mediante el terminal de Ubuntu y se procede a la instalación del simulador mediante la ejecución del archivo *build.py*. Su ejecución se realiza mediante el siguiente comando en la consola de Ubuntu:

```
juanfra@juanfra-VirtualBox:~/ns-allinone-3.19$ ./build.py
```

La herramienta *build.py* procede a realizar una comprobación de las dependencias y, si se cumplen correctamente, se comienza la compilación e instalación del simulador. En la Figura 3.22 se expone una captura de pantalla del proceso.

```
[ 936/1673] cxx: src/mpl/model/mpl-interface.cc -> build/src/mpl/model/mpl-interface.cc.1.o
[ 937/1673] cxx: src/mpl/model/remote-channel-bundle-manager.cc -> build/src/mpl/model/remote-channel-bundle-manager.cc.1.o
[ 938/1673] cxx: src/mpl/model/null-message-mpl-interface.cc -> build/src/mpl/model/null-message-mpl-interface.cc.1.o
[ 939/1673] cxx: src/mpl/model/null-message-simulator-impl.cc -> build/src/mpl/model/null-message-simulator-impl.cc.1.o
[ 940/1673] cxx: src/mpl/model/mpl-receiver.cc -> build/src/mpl/model/mpl-receiver.cc.1.o
[ 941/1673] cxx: src/mpl/model/granted-time-window-mpl-interface.cc -> build/src/mpl/model/granted-time-window-mpl-interface.cc.1.o
[ 942/1673] cxx: build/src/mobility/bindings/ns3module.cc -> build/src/mobility/bindings/ns3module.cc.7.o
[ 943/1673] cxx: src/mobility/helper/ns2-mobility-helper.cc -> build/src/mobility/helper/ns2-mobility-helper.cc.1.o
[ 944/1673] cxx: src/mobility/model/random-waypoint-mobility-model.cc -> build/src/mobility/model/random-waypoint-mobility-model.cc.1.o
[ 945/1673] cxx: src/mobility/model/random-walk-2d-mobility-model.cc -> build/src/mobility/model/random-walk-2d-mobility-model.cc.1.o
[ 946/1673] cxx: src/mobility/model/random-direction-2d-mobility-model.cc -> build/src/mobility/model/random-direction-2d-mobility-model.cc.1.o
[ 947/1673] cxx: src/mobility/model/position-allocator.cc -> build/src/mobility/model/position-allocator.cc.1.o
[ 948/1673] cxx: src/mobility/model/gauss-markov-mobility-model.cc -> build/src/mobility/model/gauss-markov-mobility-model.cc.1.o
[ 949/1673] cxx: src/mobility/model/constant-velocity-helper.cc -> build/src/mobility/model/constant-velocity-helper.cc.1.o
```

Figuras 3.22. Instalación de ns3

Una vez finalizada la instalación aparece la pantalla de la Figura 3.23 indicando los módulos construidos durante el proceso de instalación.

```
'build' finished successfully (9.629s)

Modules built:
antenna          aodv          applications
bridge           buildings    config-store
core             csma         csma-layout
dsdv             dsr          emu
energy           fd-net-device flow-monitor
internet         lte          mesh
mobility         mpi          netanim (no Python)
network          nix-vector-routing olsr
point-to-point  point-to-point-layout propagation
sixlowpan        spectrum     stats
tap-bridge       test (no Python) topology-read
uan              virtual-net-device visualizer
wave             wifi         wimax

Modules not built (see ns-3 tutorial for explanation):
brite           click         openflow
```

Figura 3.23. Módulos de ns3 instalados

Test de validación de módulos instalados

Finalizada la instalación, el simulador posee una herramienta para realizar un test con la finalidad de comprobar que todos los módulos son funcionales y validar la instalación llevando a cabo pruebas básicas, mediante pequeñas simulaciones, contrastando sus resultados.

La ejecución se realiza mediante la ejecución del archivo *test.py* a través de siguiente comando desde la consola de Ubuntu:

```
juanfra@juanfra-VirtualBox:~/ns-allinone-3.19$ ./test.py
```

El test realiza las comprobaciones de cada una de las clases del simulador y una vez finalizadas las comprobaciones indica si las ha pasado correctamente como se representa en la Figura 3.24.

```

PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/netanim/examples/ns3.19-grid-animation-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/network/examples/ns3.19-main-packet-header-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/network/examples/ns3.19-main-packet-tag-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/netanim/examples/ns3.19-dumbbell-animation-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/nix-vector-routing/examples/ns3.19-nix-simple-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/olsr/examples/ns3.19-simple-point-to-point-olsr-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/core/examples/ns3.19-main-random-variable-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/spectrum/examples/ns3.19-adhoc-aloha-ideal-phy-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/spectrum/examples/ns3.19-adhoc-aloha-ideal-phy-with-microwave-oven-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/spectrum/examples/ns3.19-adhoc-aloha-ideal-phy-matrix-propagation-loss-model-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/stats/examples/ns3.19-double-probe-example-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/stats/examples/ns3.19-file-aggregator-example-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/stats/examples/ns3.19-file-helper-example-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/stats/examples/ns3.19-gnuplot-aggregator-example-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/stats/examples/ns3.19-gnuplot-helper-example-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/network/examples/ns3.19-red-tests-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/uan/examples/ns3.19-uan-rc-example-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/netanim/examples/ns3.19-wireless-animation-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/wave/examples/ns3.19-wave-simple-80211p-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/wimax/examples/ns3.19-wimax-simple-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/virtual-net-device/examples/ns3.19-virtual-net-device-debug
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/wimax/examples/ns3.19-wimax-ipv4-debug
PASS: Example examples/tutorial/first.py
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/wimax/examples/ns3.19-wimax-multicast-debug
PASS: Example examples/routing/simple-routing-ping6.py
PASS: Example examples/wireless/mixed-wireless.py
PASS: Example src/bridge/examples/csma-bridge.py
PASS: Example src/core/examples/sample-simulator.py
PASS: Example examples/wireless/wifi-ap.py
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/propagation/examples/ns3.19-main-propagation-loss-debug
PASS: Example src/flow-monitor/examples/wifi-olsr-flowmon.py
PASS: Example /home/juanfra/workspace/bake/source/ns-3.19/build/src/uan/examples/ns3.19-uan-cw-example-debug
291 of 294 tests passed (291 passed, 3 skipped, 0 failed, 0 crashed, 0 valgrind errors)

```

Figura 3.24. Ejecución del test de validación

4. Modelado de la simulación

En este apartado se realiza una explicación de cómo se modelan los diferentes componentes de la red diseñada y los aspectos más importantes a tener en cuenta para las simulaciones realizadas.

4.1 Introducción

En el presente PFC se ha fijado como objetivo crear un modelo de red para simular el comportamiento básico de los diferentes nodos de la red, los protocolos a utilizar en la comunicación y el comportamiento del canal de comunicación centrándonos en el apartado inalámbrico y en la generación de comportamientos de flujo de datos para estudiar la evolución de los diferentes parámetros de QoS.

Para realizar el modelado de los distintos escenarios se realiza una programación de diferentes objetos que simulan cada una de las partes de la que se compone la red y los diferentes medios, alámbricos e inalámbricos, por los que se fluyen los paquetes de datos de los distintos tipos de tráfico.

En las siguientes líneas explicamos las diferentes clases, junto sus atributos y métodos más relevantes para el presente PFC, mediante las que crearemos los objetos que representan cada una de las partes de la y la forma en la que se han modelado para poder realizar las diferentes simulaciones.

También se explica las clases que nos permiten realizar las diferentes animaciones las cuales han facilitando el estudio de los resultados y la depuración del código haciendo las simulaciones más interactivas.

Todos los objetos de la simulación se crean con constructores y se accede a ellos por referencia. El código para la declaración del un objeto cualquiera del simulador en el siguiente:

```
Ptr<TipoObjeto> NombreDelObjeto= CreateObject<TipoObjeto> ();
```

La declaración crea un objeto que se maneja durante toda la ejecución de la simulación pasándolo por referencia, ya que *Ptr* es una herramienta del propio simulador desarrollada para trabajar de ésta forma.

4.2 Componentes básicos

El simulador tiene una serie de componentes básicos que interactúan con los diferentes objetos de la simulación:

- *Paquetes*: mediante la clase *Packet* se crea el objeto que simula un paquete de datos real, el cual va sufriendo una serie de transformaciones, para añadirle las cabeceras de los diferentes protocolos a medida de que va interactuando con los diferentes objetos de la simulación.
- *Colas*: mediante la clase *Queue* se controla la los diferentes sistemas de colas que gestionan el tráfico de la simulación.
- *Schedule*: método que nos permite programar eventos específicos en instantes determinados de la simulación.
- *Etiquetas*: métodos usados para hacer un seguimiento de los diferentes parámetros que pueden afectar a un paquete de datos durante su recorrido por los diferentes puntos del red en la simulación.

Para el presente PFC se ha desarrollado una clase de tipo etiqueta con el objetivo de evaluar la potencia de recepción de los paquetes de datos en los terminales de la infraestructura simulada.

4.2.1 Modelado de etiquetas de paquetes

El simulador ns3 posee una serie de herramientas que facilitan la creación de utilidades personalizadas para el seguimiento y posterior estudio del tráfico de la simulación. Para el presente proyecto se ha hecho uso de la herramienta de etiquetado. Se ha creado la etiqueta *RssiTag* para etiquetar los paquetes recibidos con su indicador de fuerza de la señal recibida (*RSSI*, en inglés *Received Signal Strength Indicator*) para que una vez el paquete haya subido por las diferentes capas de la pila de protocolos del receptor, llegando a la capa de aplicación, extraer la información de RSSI y hacer la representación

de la potencia de la señal lo mas fidedigna posible con el tráfico estudiado. El funcionamiento de la herramienta se representa en la figura 4.1.

Parte de los atributos y métodos que contiene la clase creada están representados en la Figura 4.2.

Atributos:

- *m_rssi*: variable que contiene el valor del RSSI de la etiqueta.

Métodos:

- *Set*: método que inserta el valor del RSSI pasado por parámetro en la etiqueta.

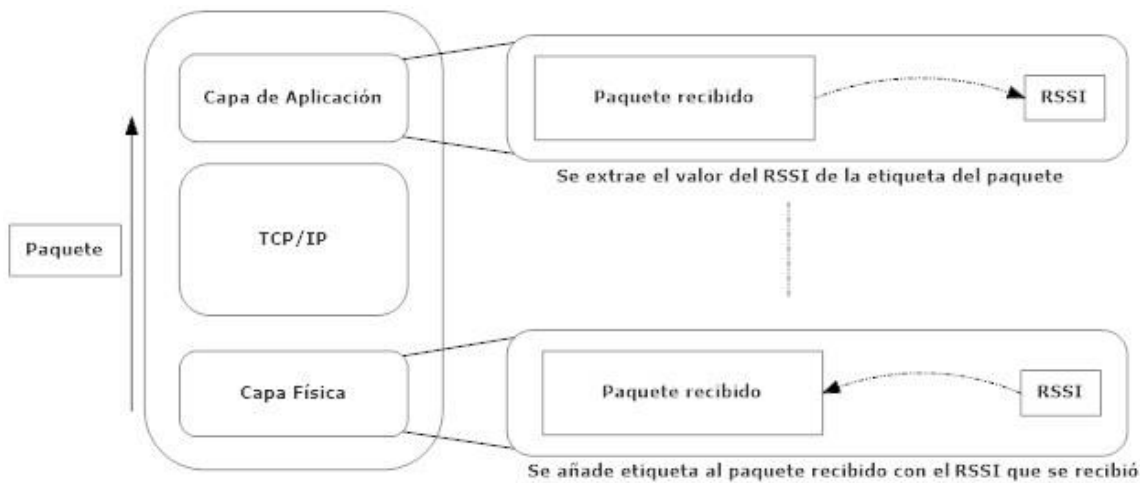


Figura 4.1. Tratamiento del paquete y su RSSI en el receptor



Figura 4.2. Diagrama de la clase RssiTag

- *Get*: método que devuelve el valor del RSSI almacenado en la etiqueta cuando es llamado.

4.3 Modelado de los nodos

Los nodos son la parte más básica de la simulación y es una abstracción que representa a un sistema informático dentro de la simulación. Se manejan mediante las clases:

- **Node.**
- **NodeContainer.**

Clase Node

Maneja todos los aspectos básicos internos del nodo, las aplicaciones que generan el tráfico y las tarjetas de red que gestionan las conexiones del nodo por los diferentes medios por los que se realice la comunicación. Se crea mediante un constructor que genera un objeto al cual se accede por referencia.

La clase contiene los atributos y los métodos representados en la Figura 4.3.

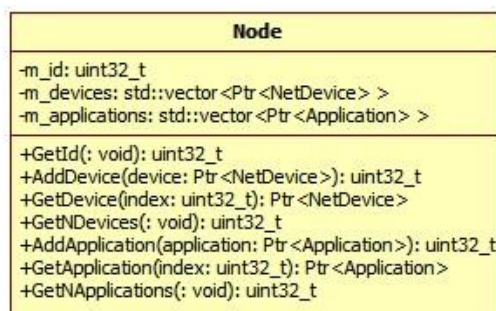


Figura 4.3. Diagrama de la clase Node

Atributos:

- *m_devices*: vector con los objetos *Netdevice* que representan las interfaces de red de este nodo y que están conectados a otros nodos a través de instancias *channel*.
- *m_applications*: vector con los objetos *Application* que representan las aplicaciones que son las encargadas de generar tráfico desde el nodo.
- *m_id*: identificador único del nodo en la *NodeList*.

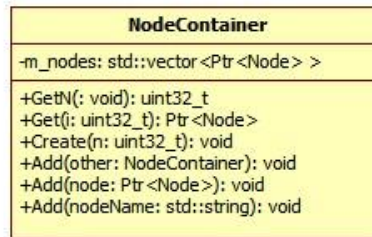
Métodos:

- *GetId*: devuelve el identificador del nodo.
- *AddDevice*: añade un objeto *NetDevice* al nodo y nos devuelve su posición dentro del vector de objetos *NetDevice* en la que se añade.
- *GetDevice*: se le pasa como parámetro el número dentro del index que ocupa el objeto *NetDevice* a recuperar dentro del vector de *NetDevice* del nodo y nos devuelve.
- *GetNDevices*: nos devuelve la cantidad de objetos *NetDevice* que contiene el nodo.
- *AddApplication*: añade un objeto *Application* al nodo y nos devuelve la posición dentro del vector de objetos *Application* en la que se añade.
- *GetApplication*: se le pasa como parámetro la posición, dentro del vector de objetos *Application*, que ocupa la aplicación que queramos obtener y nos la devuelve.

Clase **NodeContainer**

Es un *Helper* que facilita la creación de los nodos y su manejo y conceptualmente es un contenedor de objetos *Node* que se manejan conjuntamente.

La clase contiene los atributos y los métodos representados en la Figura 4.4.

Figura 4.4. Diagrama de la clase *NodeContainer*

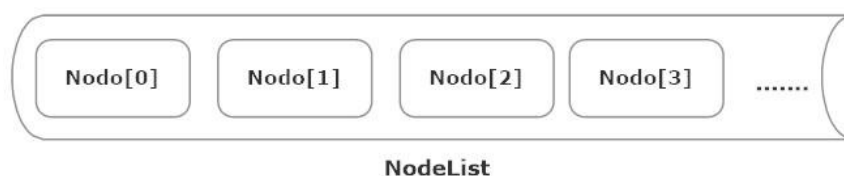
Atributos:

- *m_nodes*: vector que contiene todos los objetos *Node* que hay dentro del contenedor.

Métodos:

- *GetNn*: devuelve la cantidad de nodos que contiene el *NodeContainer*
- *Get*: le pasamos como parámetro el índice del nodo a extraer del contenedor y nos devuelve el objeto *Node* almacenado en dicho índice.
- *Create*: le pasamos como parámetro la cantidad de objetos *Node* con los que se desea crear el *NodeContainer*.
- *Add*: es un método sobrecargado al cual le podemos pasar el nombre de un nodo, el objeto *Node* o un *NodeContainer* y lo añade al *NodeContainer* por el final del vector.

Cada nodo creado se añade automáticamente a una *NodeList*, que es una lista que contiene todos los nodos de la simulación y se añaden en orden de creación. En la figura 4.5 se representa la cola de nodos *NodeList*.

Figura 4.5 *NodeList* de *ns3*

Para las simulaciones que modelan en los diferentes escenarios se agrupan los nodos en *NodeContainer* según una serie de criterios como puede ser:

- Segmento de red a la que pertenece el nodo.
- Tecnología usada como canal de comunicación.
- Algoritmo de movilidad del nodo.

Realizando esta labor se aporta un mayor entendimiento al código de la simulación y se facilita la programación de los diferentes escenarios objeto de estudio en el presente PFC.

4.4 Modelado de los protocolos de comunicación

El modelado de los protocolos se realiza siguiendo el modelo de Internet y se crea mediante el *Helper InternetStackHelper* que crea una pila de protocolos de dicho modelo con sus diferentes capas por donde fluyen los paquetes que irán modificándose en las diferentes capas añadiéndose las respectivas cabeceras de los distintos protocolos.

La pila interactúa con *Ipv4 AddressHelper* e *Ipv4GlobalRoutingHelper*, encargándose éstos últimos de manejar todo lo relacionado con el direccionamiento lógico y las tablas de enrutamiento de los nodos mediante el protocolo Ipv4.

Parte de los métodos que contiene la clase *Ipv4AddressHelper* están representados en la Figura 4.6.

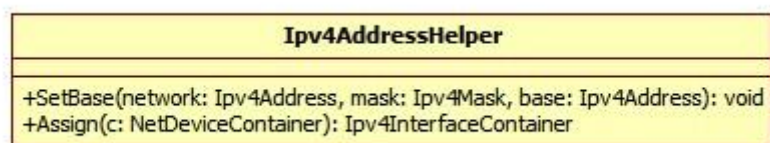


Figura 4.6. Diagrama de la clase *Ipv4AddressHelper*

Métodos:

- *SetBase*: método mediante el cual definimos el direccionamiento de la red.
- *Assign*: método que asigna una dirección IP a la NetDevice que se le pase por parámetro.

4.5 Modelado de los canales de comunicación

El modelado de las partes que representen los diferentes canales de comunicación en la simulaciones, cables y señales radioeléctricas, se realiza haciendo uso de una serie de objetos definidos dentro del simulador para tal fin en los que siempre se tiene que tener en cuenta dos partes bien diferenciadas:

- *Tarjetas de Red*: objetos que permiten al nodo la comunicación con el medio permitiendo la emisión o recepción de flujos de datos hacia o desde el canal.
 - *NetDevices*.
- *Canales*: objetos que definen las características del canal de comunicación y que pueden dividirse, según el medio utilizado, en dos grupos.
 - Alámbricos o cableados:
 - *PointToPointHelper*.
 - *CsmaHelper*.
 - Inalámbricos
 - *WiFiHelper*.
 - *YansWiFiChannel*.
 - *YansWiFiPhyHelper*.
 - *QosWiFiMacHelper*.
 - *HTWiFiMacHelper*.

Las dos partes están fuertemente relacionadas debido a que, tal y como pasa en un entorno real, no podremos usar una tarjeta de red que esté diseñada para un canal inalámbrico con un canal alámbrico.

4.5.1 Tarjetas de Red

Dispone de las clases que se presentan a continuación.

NetDevices

Son las clases de ns3 desarrolladas para la simulación de la estructura de una tarjeta de red en el mundo real siendo una interfaz entre el canal y el nodo, conteniendo todos los métodos necesarios para la permitir la comunicación y el flujo de datos entre estos dos objetos de la simulación.

Parte de los métodos que contiene la clase están representados en la Figura 4.7.

Métodos:

- *GetIfIndex*: devuelve el índice que ocupa la *NetDevice* dentro del *NetDeviceList* del nodo.
- *GetChannel*: devuelve el canal asociado al *NetDevice*.
- *SetAddress*: le pasamos como parámetro la dirección MAC que queremos tenga la *NetDevice*.
- *GetAddress*: devuelve la dirección MAC que contiene la *NetDevice*.

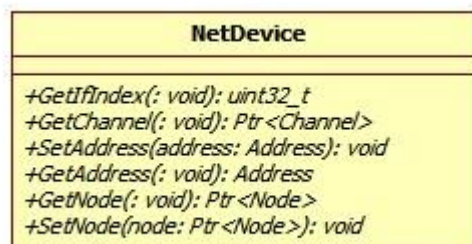


Figura 4.7. Diagrama de la clase *NetDevice*

- *GetNode*: devuelve el nodo que contiene a la *NetDevice*.
- *SetNode*: añade la *NetDevice* al nodo que se pasa como parámetro.

Un nodo puede contener varios *NetDevices* para comunicarse mediante varios canales de comunicación con diferentes características. Debido a ello dentro de cada nodo existe un *DeviceList* con todos los *NetDevices* de dicho nodo que se ordenan por orden de creación. En la figura 4.8 se representa la *DeviceList* que se crea dentro del nodo.

NetDeviceContainer

Similar al *NodeContainer*, crea un contenedor de *NetDevice* para facilitar el manejo de varias *NetDevices*. En la simulación del presente PFC se usa, principalmente, para gestionar todo lo relacionado con las direcciones IPv4 de los diferentes *NetDevice*. Tiene exactamente los mismos atributos que *NodeContainer* a excepción del método *create* que *NetDevice* no lo incluye.

4.5.2 Canales

Dispone de las clases que se presentan a continuación.

Canales alámbricos

Los canales cableados o alámbricos son los más sencillos de modelar. Para la simulaciones se han modelado dos tipo de canales alámbricos mediante los *Helper* de ns3.

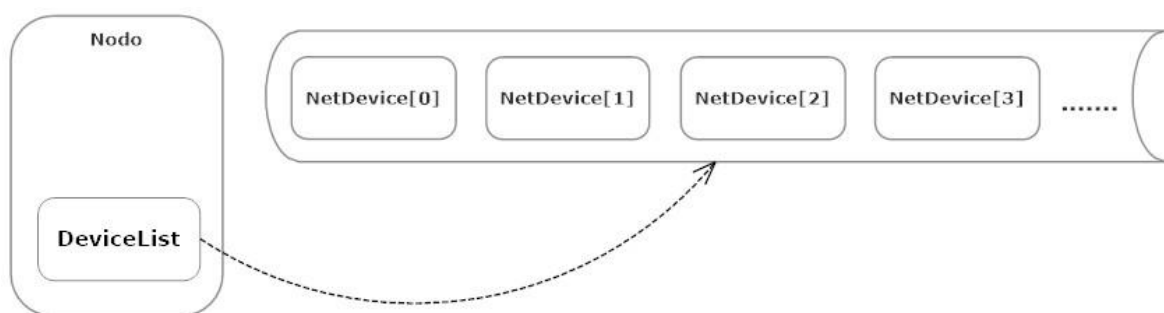


Figura 4.8. *DeviceList* de un nodo

- **PointToPointHelper**
- **CsmaHelper**

El *Helper PointToPointHelper* nos permite definir una conexión punto a punto entre dos nodos de nuestra red simulada y el *Helper CsmaHelper* permite definir una conexión LAN entre varios nodos. Mediante estas clases se crearan objetos *PointToPointNetDevices* y *CsmaNetDevices* con unas características determinadas asociadas a un canal, Punto-a-Punto o LAN, con unas propiedades definidas, dependiendo de los requerimientos de la simulación, que se instalan en los nodos de la red que lo requieran.

Parte de los métodos que contienen están representados en la Figura 4.9.

Métodos:

- *SetDeviceAttribute*: se pasa como parámetro el nombre del valor que queremos definir para la conexión de la *NetDevice* con el canal y el valor de dicho parámetro. Por ejemplo la velocidad que soporta la tarjeta de red.
- *SetChannelAttribute*: se pasa como parámetro el nombre del valor que queremos definir para el canal de comunicación y el valor de dicho parámetro. Por ejemplo el retraso que provoca la propagación del cable

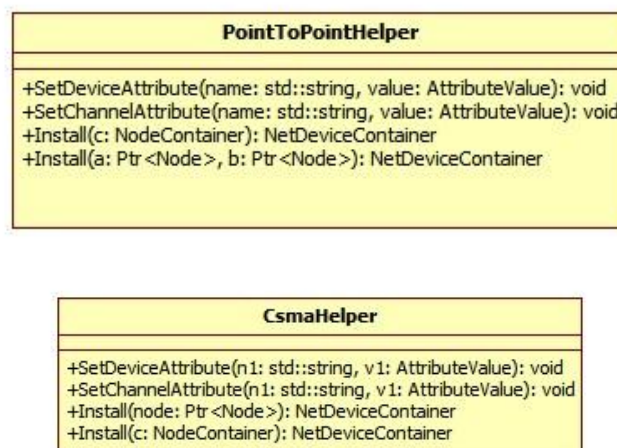


Figura 4.9. Diagrama de las clases *PointToPointHelper* y *CsmaHelper*

- *Install*: método sobrecargado al cual le podemos pasar los dos objetos Node o un NodeContainer, que contenga dos objetos node (debido a que es una comunicación punto a punto entre únicamente dos nodos), y se instala la NetDevice en los nodos, devolviendo el objeto NetDevice con la configuración programada para dicho canal.

Canales inalámbricos

Los canales inalámbricos son los más complicados de modelar debido a la alta complejidad que representa el manejo y procesamiento de señales radioeléctricas, ya que estas sufren una serie de fenómenos que afectan a su propagación y por consiguiente a su recepción por parte de los nodos receptores. Por todo ello se explican en el próximo apartado de una forma bastante amplia, ya que la simulación es una de las partes más importantes del presente PFC.

4.6 Modelado del canal inalámbrico

Existen varias partes diferenciadas en la simulación de un entorno inalámbrico y para cada una de ellas se modelan objetos del simulador que den como resultado una WifiNetDevice que gestione dicho canal modelado.

Para crear un *WifiNetDevice*, se debe modelar principalmente cinco aspectos de la comunicación inalámbrica:

- *Modelar la propagación*: se modela un tipo de propagación determinado teniendo en cuenta las características del entorno que queremos simular y el estándar que se encarga de gestionar dichas señales reflejando, de la forma más realista posible, características del escenario y de la propia señal radioeléctrica, como pueden ser las interferencias o el ruido del entorno, y como le afectan. Las principales características que se modelan en el canal es el modelo de pérdidas y el modelo de retraso de propagación.

- *Modelar parámetros de codificación*: se modela el envío y recepción de la señal inalámbrica de que se propaga por el canal. En este apartado se decide si cada trama se decodifica con éxito o no en función de la intensidad de señal recibida y el ruido.
- *Modelar la MAC*: está principalmente relacionado con la arquitectura de la red y el nivel del nodo dentro de la misma. Los usuarios configuran la arquitectura WiFi (ad-hoc o infraestructura) y si dicha arquitectura requiere QoS o no.
- *Modelar estándar IEEE 802.11*: se modela la norma WiFi que se desea implementar (IEEE 802.11b, IEEE 802.11g o IEEE 802.11n) y el algoritmo de control de la velocidad.
- *Modelar movilidad*: se modela la movilidad de los nodos por el escenario ya que la *WifiNetDevice* tiene que tener en cuenta en todo momento la posición de los nodos implicados en la comunicación inalámbrica para calcular sus respectivos valores radioeléctricos.

El modelado del canal se lleva a cabo mediante la clase *YansWifiChannel* que está basada en un modelo detallado en [3]. Parte de sus atributos y métodos están representados en la Figura 4.7.

Atributos:

- *m_loss*: modelo de propagación predeterminado para la simulación.
- *m_delay*: modelo de retraso predeterminado para la simulación.

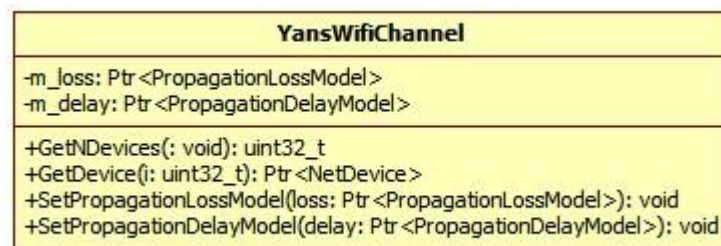


Figura 4.10. Diagrama de la clase *YansWifiChannel*

Métodos:

- *GetNDevices*: devuelve la cantidad de NetDevices asociadas al canal.
- *GetDevice*: se pasa como parámetro el índice del objeto NetDevice a extraer del canal y nos devuelve el objeto de la lista Channellist.
- *SetPropagationLossModel*: método por el cual le pasamos un modelo matemático de pérdidas de propagación para configurar el canal.
- *SetPropagationDelayModel*: método por el cual configuramos el canal con un modelo matemático de retraso de propagación.

4.6.1 Modelado de la propagación

Para el modelado de la propagación se trabaja con las dos partes que definen sus características por un lado se modela sus retraso a la hora de propagarse con la clase predeterminada por el simulador el cual toma como valor de la velocidad de propagación la velocidad de la luz y por otro las pérdidas con la clase *HybridBuildingsPropagationLossModel*, que hace uso de diferentes algoritmos de propagación según las posiciones relativas de los nodos emisores y receptores con los edificios modelados en el escenario.

Parte de los atributos y de los métodos que contiene están representados en la Figura 4.11.

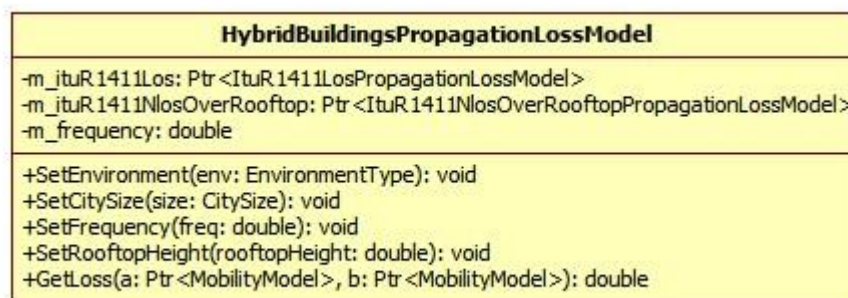


Figura 4.11. Diagrama de la clase *HybridBuildingsPropagationLossModel*

Atributos:

- *m_ituR1411Lo*: variable tipo *ItuR1411LosPropagationLossModel* la cual es un objeto con el que se implementa la recomendación ITU 1411 para comunicaciones con *Línea de Visión (LoS, del inglés Line of Sight)*
- *m_ituR1411NlosOverRooftop*: variable tipo *ItuR1411NlosOverRooftopPropagationLossModel* la cual es un objeto con el que se implementa la recomendación ITU 1411 para comunicaciones *Sin Línea de Visión (NLoS, del inglés Non Line of Sight)* a partir de un cierto umbral.
- *m_frequency*: variable que define la frecuencia de trabajo a la que se realiza el modelado de la aplicación.

Métodos:

- *SetEnvironment*: método usado para configurar el entorno de propagación de la señal radioeléctrica que puede ser de tres clases.
 - *UrbanEnvironment*: entornos urbanos.
 - *SubUrbanEnvironment*: zonas periférica de una ciudad.
 - *OpenAreasEnvironment*: zona de áreas abiertas.

Mediante este método se calcula como afecta la presencia de características intrínsecas existentes en este tipo de entorno, como puede ser viandantes, vehículos... a la propagación para el cálculo de las pérdidas.

- *SetCitySize*: método usado para configurar el tamaño de la distribución de los edificios y tener en cuenta datos estadísticos para calcular como afecta a la propagación de la señal. Existen tres tipos definidos para los cálculos.
 - *SmallCity*: para pequeños núcleos.
 - *MediumCity*: para ciudades de tamaño medio.
 - *LargeCity*: para grandes ciudades.

- *SetFrequency*: método que permite configurar la frecuencia a la que se hace el estudio de pérdidas.
- *GetLoss*: método que calcula las pérdidas en base a la recomendación ITU 1411 adecuada a la posición relativa entre nodos y edificios.

4.6.2 Modelado de la codificación de señales inalámbricas

El modelado de la codificación de la señal para su posterior propagación a través del canal se realiza mediante el Helper *YansWifiPhyHelper* y, al igual que *YansWifiChannel*, su diseño está basado en [5]. Es una de las clases más importante del módulo WiFi debido al desempeño que tiene con éste.

Parte de los atributos y de los métodos que contiene están representados en la Figura 4.12.

Atributos:

- *m_channel*: variable que contiene el canal al que está asociado el *YansWifiPhy*.

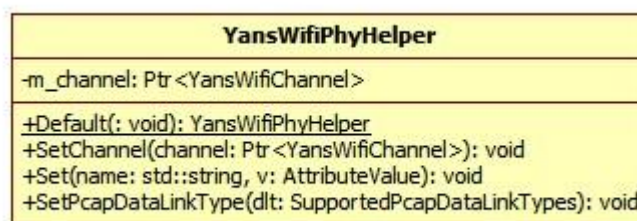


Figura 4.12. Diagrama de la clase *YansWifiPhyHelper*

Métodos:

- *Default*: su llamada realiza la configuración estándar de todos los parámetros de *YansWifiPhy* como pueden ser umbrales de potencia, de transmisión o ganancia de recepción entre muchos otros.
- *SetChannel*: se pasa como parámetro el *YansWifiChannel* por el cual se va a comunicar el *YansWifiPhy*.
- *SetPcapDataLinkType*: función que programa las cabeceras de los paquetes para la generación de Pcap para IEEE 802.11. Dichos .pcap se analizan posteriormente con la aplicación Wireshark.

Para poder llevar a cabo los estudios de potencia de señal se han implementado dos métodos a la clase *YansWifiPhy*, usada por el *Helper YansWifiHelper*, para la creación de la *WifiNetDevice* y se ha modificado un tercero que ya estaba implementado en dicha clase para poder analizar cómo afecta la potencia de recepción al tráfico sin necesidad de transferencia de flujo de paquetes de datos.

Métodos creados en *YansWifiPhy*:

- *ObtenerRssi*: devuelve la potencia de recepción de la señal del nodo en dBm para poder analizar la cobertura en la zona por la que se está moviendo.
- *ObtenerSnr*: devuelve la relación señal a ruido de recepción de la señal del nodo para analizar la calidad de la cobertura en la zona por la que se está moviendo.

Métodos modificados en *YansWifiPhy*:

- *EndReceive*: método encargado de procesar el paquete recibido y enviarlo a las capas superiores. Se ha introducido una etiqueta a cada paquete con su correspondiente RSSI de recepción para un análisis posterior de la relación entre la potencia de la señal y los parámetros de QoS del flujo de

paquetes. Con dicho propósito se ha creado un *rss-tag* que permite la etiquetación del RSSI en el paquete recibido.

La modificación realizada en el método es la siguiente:

```
RssiTag tag; //Declaramos la etiqueta que hemos creado
packet->RemovePacketTag (tag); //removemos por si se ha etiquetado en otro punto de la simulación
tag.Set(signalDbm); //Insertamos el valor de potencia de la señal en dBm en la etiqueta
if (! packet->PeekPacketTag (tag))
{
    packet->AddPacketTag (tag); //Etiquetamos el RSSI con la etiqueta
}
```

4.6.3 Modelado del MAC

El modelado de la MAC se realiza mediante dos *Helper*, según el estándar WiFi que se desee simular:

- *NqosWifiMacHelper*: no tiene implementado QoS en las colas de transmisión de la MAC, por lo que todo el tráfico siempre tiene la misma preferencia. Usa políticas *Best Effort*. Es el modelo de los estándar IEEE802.11b y IEEE802.11g.
- *HtWifiMacHelper*: implementa modo de alto rendimiento y calidad de servicio en la MAC de los nodos de la simulación, pudiendo crear diferentes prioridades en las colas de los nodos y mejorar los parámetros de QoS para el tráfico que lo requiera. Es el modelo del estándar IEEE 802.11n.

El modelado del MAC se realiza con los valores *Defaul* del simulador, ya que cumple con los estándares del IEEE 802.11, y el simulador se encarga de su creación y gestión internamente, teniendo que indicar la arquitectura de la red WiFi y el *Service Set Identifier (SSID)*.

Declaración la capa MAC, para los estándares IEEE 802.11b y IEEE 802.11g, para un AP con un SSID *Ssid*:

```
QosWifiMacHelper mac = QosWifiMacHelper :: default ();
mac.SetType ("ns3::ApWifiMac", "Ssid", SsidValue (ssid));
```

Declaración la capa MAC, para el estándar IEEE 802.11n, para un STA asociado con el SSID *Ssid*:

```
HtWifiMacHelper mac = HtWifiMacHelper :: default ();
mac.SetType ("ns3::StaWifiMac", "Ssid", SsidValue (ssid));
```

4.6.4 Modelado del estándar WiFi

El modelado del estándar se lleva a cabo con el *Helper WifiHelper*, con el que una vez modelado el MAC y el canal, tiene un método por el cual realizamos la instalación de la *WifiNetDevice* en el nodo que lo requiera.

Parte de los atributos y métodos que contiene la clase están representados en la Figura 4.13.

Métodos:

- *Default*: crea un objeto *WifiHelper* con todos los parámetros por defecto del simulador.

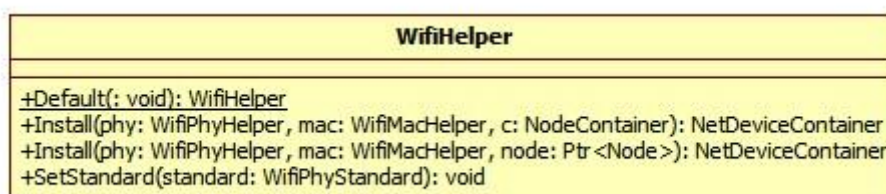


Figura 4.13. Diagrama de la clase *WifiHelper*

- *Install*: método sobrecargado al cual se le pasa la YansWifiPhy, el MAC modelado con o sin QoS y un objeto Node o NodeContainer a los instalan la configuración canal-MAC modelada.
- *SetStandard*: método por el que se pasa por parámetro el tipo de estándar que se va a implementar en la red WiFi modelada, IEEE 802.11b, IEEE 802.11g o IEEE 802.11n

4.6.5 Modelado de la movilidad

Para el modelado de la movilidad se hace uso del *Helper MobilityHelper* que crea modelos de movilidad para simular los movimientos de los terminales móviles, mediante algoritmos de aleatoriedad, por el escenario modelado a partir de una posición inicial programada.

Parte de los atributos y métodos que contiene la clase están representados en la Figura 4.14.

Atributos:

- *m_mobilityStack*: vector donde se almacena los modelos de movilidad de los nodos de la simulación.

MobilityHelper
-m_mobilityStack: std::vector<Ptr<MobilityModel>>
+SetPositionAllocator(allocator: Ptr<PositionAllocator>): void
+SetPositionAllocator(type: std::string, n1: std::string, v1: AttributeValue, n2: std::string, v2: AttributeValue): void
+SetMobilityModel(type: std::string, n1: std::string, v1: AttributeValue, n2: std::string, v2: AttributeValue): void
+Install(node: Ptr<Node>): void
+Install(container: NodeContainer): void
-CourseChanged(stream: Ptr<OutputStreamWrapper>, mobility: Ptr<const MobilityModel>): void

Figura 4.14. Diagrama de la clase *MobilityHelper*

Métodos:

- *SetPositionAllocator*: método sobrecargado al que se le puede pasar un objeto *PositionAllocator* con las posiciones iniciares de todos los nodos o un algoritmo de aleatoriedad y sus atributos para establece diferentes posiciones iniciales aleatorias.
- *SetMobilityModel*: método por el cual se pasa un modelo de movilidad con sus atributos para definir el movimiento del nodo por el espacio del escenario.
- *Install*: método que programa la movilidad configurada en el nodo o nodos que le pase por parámetro.

4.7 Modelado de edificios

Para el modelado de los edificios que se representan en las simulaciones usamos dos tipos de clases:

- *Building*: con ella se define el volumen que ocupa el edificio en el espacio y sus principales características. Parte de los atributos y métodos que contiene la clase están representados en la Figura 4.15.

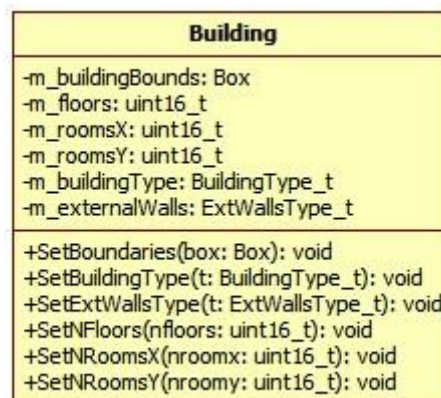


Figura 4.15. Diagrama de la clase *Building*

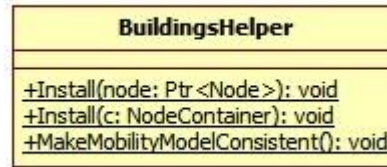
Atributos:

- *m_buildingBounds*: variable de tipo *box* que define el volumen que ocupa el edificio en el espacio del escenario de la simulación.
- *m_floors*: número de plantas que tiene el edificio (el eje Z de las tres dimensiones).
- *m_roomsX*: número de subdivisiones en el eje X que tiene el edificio.
- *m_roomsY*: número de subdivisiones en el eje Y que tiene el edificio.
- *m_buildingType*: define el tipo de edificio que se está modelando (Oficinas, comercial o residencial).
- *m_externalWalls*: tipo de paredes externas que tiene el edificio (madera, cemento con ventanas, cemento sin ventanas o de piedra).

Métodos:

- *Métodos Set*: métodos con los que se modifican cada uno de los atributos del edificio.
- *BuildingsHelper*: crea los objetos *MobilityBuildingInfo*, los cuales se encargan de calcular las posiciones relativas entre los nodos y los edificios (si están dentro o fuera del edificio, en la primera planta o en superiores...) posibilitando calcular como afecta la presencia de los edificios a la propagación de las señales radioeléctricas.

Parte de los atributos y métodos que contiene la clase están representados en la Figura 4.16.

Figura 4.16. Diagrama de la clase *BuildingsHelper*

Métodos:

- *Install*: método sobrecargado que permite indicar, en los modelos de movilidad de los nodos, la posición en el espacio de los nodos para que interactúen con ellos.
- *MakeMobilityModelConsistent*: una vez se haya definido todos los modelos de movilidad, se haya definido los edificios y indicado a los modelos de movilidad de los nodos la posición de los edificios este método inicializa los estados del *MobilityBuildingInfo* de todos los nodos.

4.8 Modelado de aplicaciones

Buscando estudiar el impacto de los distintos flujos de datos sobre la infraestructura diseñada, se ha hecho una selección de los tráficos de red, tanto multimedia como de datos, más comunes. Los tipos de tráfico son:

- VoIP.
- Vídeo Streaming.
- *Hypertext Transfer Protocol (HTTP)*.
- *File Transfer Protocol (FTP)*.

Para el modelado del envío y recepción del tráfico comentado hacemos uso de cuatro clases *Application* del simulador.

OnOffHelper

Simula un flujo constante de datos en el estado ON y desactiva dicho flujo en el estado OFF. Es la clase que se usa para el modelado de las aplicaciones de VoIP y HTTP. A la hora de crear el objeto con la clase se deberá definir la ip, el puerto y el protocolo que se desea usar.

Parte de los atributos y métodos que contiene la clase están representados en la Figura 4.17.

Métodos:

- *SetAttribute*: se pasa el nombre del atributo de la clase principal del Helper *OnOffApplication* que se desea modificar y el valor que se le desea dar por parámetro. En la simulación se usarán los atributos *DataRate*, para determinar la velocidad de transmisión y *PacketSize* para determinar el tamaño de los paquetes.
- *Install*: método que programa la aplicación configurada en el nodo o nodos que le pase por parámetro.

UdpTraceClientHelper

Clase usada para el modelado del tráfico de vídeo y está específicamente diseñada para ello en el simulador. Los parámetros que hay que pasarle son la dirección IP destino, el puerto y un archivo trace.txt que contiene los datos para la generación del tráfico del vídeo.

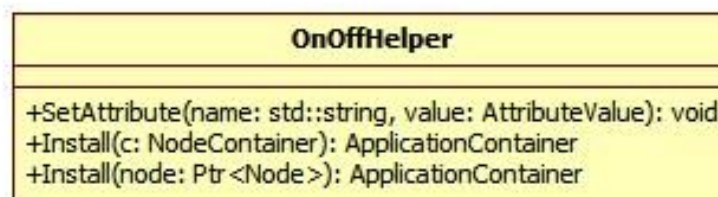


Figura 4.17. Diagrama de la clase *OnOffHelper*

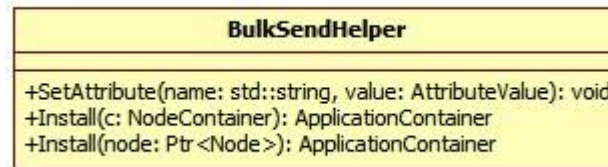


Figura 4.18. Diagrama de la clase *BulkSendHelper*

BulkSendHelper

Clase que genera un flujo de datos continuado que se envía lo más rápido posible ocupando todo el ancho de banda que haya disponible. A la hora de crear el objeto con la clase se debe definir la dirección IP, el puerto y el protocolo que se desea usar.

Parte de los atributos y métodos que contiene la clase están representados en la Figura 4.18.

Métodos:

- *SetAttribute*: se pasa el nombre del atributo de la clase principal del Helper *BulkSendApplication* que se desea modificar y el valor que se le desea dar por parámetro, en la simulación se usa el atributo *MaxBytes*, para determinar la cantidad de bytes que se transmiten.
- *Install*: método que programa la aplicación configurada en el nodo o nodos que se le pase por parámetro.

PacketSinkHelper

Consume el tráfico creado por la aplicaciones anteriores actuando de receptor de los paquetes en la capa de aplicación. Cuando se crea el objeto con el constructor se le pasan los parámetros de protocolo de comunicación y el puerto por el que realiza la escucha.

La cantidad de aplicaciones que se deseen en las simulaciones se puede pasar desde consola mediante las variables *nAppVoip*, *nAppVideo*, *nAppHttp* y *nAppFtp*. Y su

instalación se realiza de forma secuencial, instalando primero las de VoIP en los primeros *nAppVoip* nodos y así consecutivamente con las de vídeo, HTTP y FTP. Por lo que se instala una aplicación por nodo, si dichas variables sumaran más que la cantidad total de nodos se lanza un mensaje indicándolo y sale de la simulación, la codificación que implementa dicha casuística es la siguiente:

```
if (nSta < nAppTotales) //nSta es la cantidad de nodos cliente
{
std::cout <<"El número de Aplicaciones totales"<<nAppTotales<<" es mayor que la cantidad
de terminales"<< nSta <<std::endl;
exit (1);
}
```

4.8.1 Modelado de la QoS del tráfico

En la clases principales, a la que recurren los *Helpers OnOffHelper* y *UdpTraceClientHelper*, *OnOffApplicationHelper* y *UdpTraceClientHelper* se realizó una serie de cambios para aportar QoS y poder analizar el desempeño de los mecanismos del estándar IEEE 802.11n en éste sentido sobre la red diseñada. Para los cambios realizados se hizo uso de la clase *QosTag* de ns3, similar a la *RssiTag* diseñada y explicada anteriormente, la cual nos permite dar preferencia al tráfico generado marcando los paquetes con la QoS deseada según los valores especificados en la tabla de la Figura 4.1. Para ello se ha añadido un nuevo atributo a la clase *QosTag* que podemos configurar desde el *Helper* y se ha marcado el campo del paquete correspondiente para que sea procesado por la MAC, modelada mediante el *Helper HtWifiMacHelper*, y le otorgue la preferencia correspondiente. En la Tabla 4.1 se observa el grado de calidad de servicio que tiene ns3 [2].

Valor	Prioridad
1	background
0	best effort (default)
3	excellent effort
4	Control de carga
5	vídeo, 100ms > retraso y jitter
6	voz, 10ms > retraso y jitter
7	Control de red

Tabla 4.1. QoS programable en ns3

4.8.2 Modelado del tráfico de VoIP

El modelado del flujo de datos de VoIP para la simulación se realiza representando uno de los interlocutores de una llamada de voz bidireccional (único flujo de datos de voz en sentido servidor-cliente). El escenario se representa en la Figura 4.19.



Figura 4.19. Flujo de datos de VoIP en sentido servidor-cliente

Se realiza la simulación de un flujo de paquetes basando en una de las principales características de la comunicación del lenguaje, existencia de dos estados diferenciados.

- *Habla*: es el estado denominado ON en el que el usuario está hablando durante el cual los paquetes de voz de tamaño constante se transmiten a intervalos regulares.
- *Silencio o escucha*: el estado denominado OFF en que el usuario permanece en silencio o a la escucha y no está transmitiendo paquetes de voz.

La duración de los estados ON y OFF se estiman basándonos en un modelo de Markov con dos estados (ON y OFF) y tres parámetros:

- La duración media del estado de ON: $t_{ON} = 1 / \alpha$. Dada por la siguiente distribución exponencial:

$$F(T) = \alpha e^{-\alpha T}$$

- La duración media del estado OFF: $t_{OFF} = 1 / \beta$. Dada por la siguiente función exponencial:

$$F(T) = \beta e^{-\beta T}$$

- La tasa de transmisión de paquetes es constante durante el estado ON: $\lambda = 1/T$.

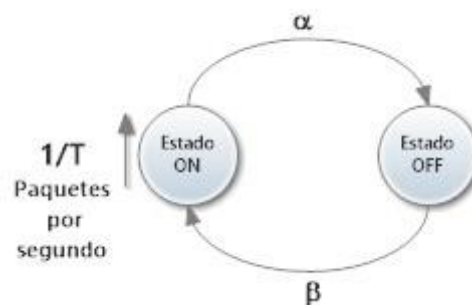


Figura 4.20. Estados de la comunicación de voz

Los paquetes son transportados por el puerto 4569 y el protocolo es *User Datagram Protocol (UDP)*, debido que se desea simular un flujo de datos multimedia. El

tamaño de los paquetes y el tiempo empaquetado de los mismos dependen del codec usado para la encapsulación de la voz.

En la tabla 4.2 se muestra los valores utilizados en para la generación del flujo de tráfico de VoIP [11].

La velocidad real en bits depende del códec utilizado y el tiempo de empaquetado. El códec seleccionado para este proyecto es el G.711 debido a que se lleva a cabo sin compresión asegurando así la mejor calidad de voz debido a lo cual, tal y como se puede observar en la tabla 4.1, posee una clasificación alta en la escala *Mean Opinion Score* (MOS). Sin embargo, como resultado, sus requisitos de ancho de banda son más altos. La velocidad de datos producidos por este codec es de 64 kbps y el tiempo de empaquetado (*Voice Play Load Size*) es de 20 ms. El tiempo de empaquetado es un parámetro crítico para el cálculo del ancho de banda debido a que cuanto mayor sea este parámetro, más muestras de voz se insertan en un paquete, por lo que se requiere un menor número de paquetes a transmitir y menor ancho de banda, sin embargo, el retraso de extremo a extremo aumenta. Para tiempos de empaquetado más cortos los paquetes se transmiten con más frecuencia, reduciendo el retraso pero, debido a este hecho, se requiere mayor ancho de banda ya que cada paquete se transmite con su propia cabecera (RTP, UDP, cabeceras del protocolo IP...).

Codec Information				Bandwidth Calculations					
Codec & Bit Rate (Kbps)	Codec Sample Size (Bytes)	Codec Sample Interval (ms)	Mean Opinion Score (MOS)	Voice Payload Size (Bytes)	Voice Payload Size (ms)	Packets Per Second (PPS)	Bandwidth MP or FRF.12 (Kbps)	Bandwidth w/cRTP MP or FRF.12 (Kbps)	Bandwidth Ethernet (Kbps)
G.711 (64 Kbps)	80 Bytes	10 ms	4.1	160 Bytes	20 ms	50	82.8 Kbps	67.6 Kbps	87.2 Kbps
G.729 (8 Kbps)	10 Bytes	10 ms	3.92	20 Bytes	20 ms	50	26.8 Kbps	11.6 Kbps	31.2 Kbps
G.723.1 (6.3 Kbps)	24 Bytes	30 ms	3.9	24 Bytes	30 ms	33.3	18.9 Kbps	8.8 Kbps	21.9 Kbps
G.723.1 (5.3 Kbps)	20 Bytes	30 ms	3.8	20 Bytes	30 ms	33.3	17.9 Kbps	7.7 Kbps	20.8 Kbps
G.726 (32 Kbps)	20 Bytes	5 ms	3.85	80 Bytes	20 ms	50	50.8 Kbps	35.6 Kbps	55.2 Kbps
G.726 (24 Kbps)	15 Bytes	5 ms		60 Bytes	20 ms	50	42.8 Kbps	27.6 Kbps	47.2 Kbps
G.728 (16 Kbps)	10 Bytes	5 ms	3.61	60 Bytes	30 ms	33.3	28.5 Kbps	18.4 Kbps	31.5 Kbps
G722_64k(64 Kbps)	80 Bytes	10 ms	4.13	160 Bytes	20 ms	50	82.8 Kbps	67.6Kbps	87.2 Kbps
ilbc_mode_20(15.2Kbps)	38 Bytes	20 ms	NA	38 Bytes	20 ms	50	34.0Kbps	18.8 Kbps	38.4Kbps
ilbc_mode_30(13.33Kbps)	50 Bytes	30 ms	NA	50 Bytes	30 ms	33.3	25.867 Kbps	15.73Kbps	28.8 Kbps

Tabla 4.2. Tabla con parámetros de los diferentes codec

La implementación del tráfico de VoIP en la simulación se realiza mediante el Helper OnoffApplicationHelper. Los tiempos de estado ON y estado OFF se produjeron utilizando una clase de ns3 diseñada para generar números aleatorios siguiendo una función exponencial a la que se le pasa la duración media de cada uno de los estados de 0,352 para el estado ON y 0,65 para el estado OFF. Para estos valores de duración media tenemos en cuenta que la estimación se realiza sobre llamadas realizadas con una duración media de 3 minutos distribuidos exponencialmente siguiendo los estudios estadísticos en [22]. Para hacer el estudio más realista, debido a que ns3 no lo tiene en cuenta, se añadió el tamaño de la cabecera RTP a la carga útil de voz por lo que ésta finalmente tiene un valor de 172 bytes.

$$\text{Carga útil total} = 160 \text{ bytes} + 12 \text{ bytes cabecera RTP} = 172 \text{ bytes}$$

Debido al aumento de la carga útil la velocidad de datos también se amplía pasando a valer 68.800 bps.

$$\text{Velocidad de transmisión} = 172 \text{ bytes} \times 8 \times 50 \text{ paquetes por segundo} = 68800 \text{ Kbps}$$

La codificación que implementa la aplicación de VoIP para la generación del tráfico con las características comentadas es la siguiente:

```
OnOffHelper serVoip = OnOffHelper("ns3::UdpSocketFactory", InetSocketAddress(ip, puertoVoip));
serVoip.SetAttribute("OnTime",StringValue("ns3::ExponentialRandomVariable (0.352)"));
serVoip.SetAttribute("OffTime",StringValue("ns3::ExponentialRandomVariable (0.650)"));
serVoip.SetAttribute("PacketSize", UintegerValue(172));
serVoip.SetAttribute("DataRate",DataRateValue(68800));
serVoip.SetAttribute("ClaseDeAcceso", UintegerValue (UintegerValue (6))); //QoS
```

4.8.3 Modelado del tráfico de vídeo

El modelado del flujo de datos de vídeo de la simulación se realiza partiendo de la premisa en la cual los paquetes con la información audiovisual son enviados en streaming

desde un servidor de vídeo hacia el cliente, siendo la comunicación unidireccional. El escenario se representa en la Figura 4.21.



Figura 4.21 Flujo de datos del streaming de vídeo

El tráfico es modelado haciendo uso del Helper de `UdpTraceClientHelper` que envía un flujo de paquetes UDP por el puerto 9000 basado en un archivo con las trazas de un streaming codificado en MPEG4. El archivo de trazas, *trace.txt*, se especifica por el usuario, pasándose como parámetro a `UdpTraceClientHelper`, y la herramienta simula un tráfico de paquetes con el tipo, los tamaños y los tiempos de llegada especificados en el dicho archivo. Los archivos de traza deben tener cuatro columnas que especifica el índice del marco, el tipo de trama, el tiempo de la trama que se ha generado por el codificador y el tamaño del fotograma en bytes, en la Figura 4.22 hay un ejemplo de archivo de traza.

El archivo de trazas del streaming usado en este PFC simula la cámara de un parking y fue obtenido de [13].

Frame No.	Frametype	Time[ms]	Length [byte]
1	I	0	402
2	P	120	490
3	B	40	142
4	B	80	567
5	P	240	539
6	B	160	142
7	B	200	159
8	P	360	76
9	B	280	38
10	B	320	33

Figura 4.22. Ejemplo del formato del archivo de trazas

La codificación que implementa la aplicación de vídeo para la generación del tráfico con las características comentadas es la siguiente:

```
UdpTraceClientHelper serVideo (ip, puertoVideo, "trace.txt");
serVoip.SetAttribute("ClaseDeAcceso", UIntegerValue (UIntegerValue (5))); //QoS
```

4.8.4 Modelado del tráfico HTTP

En el modelado del flujo de datos HTTP de la simulación se estima un sólo nivel de actividad y el tráfico es de descarga siempre en dirección servidor-cliente. El escenario se representa en la Figura 4.22.

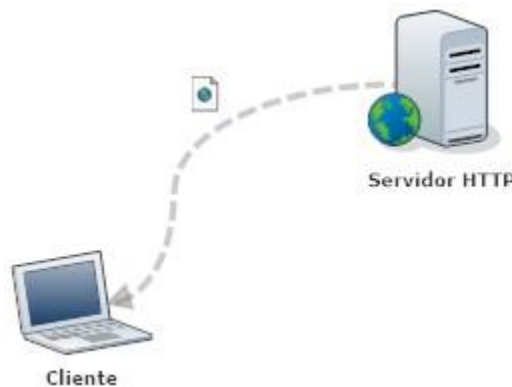


Figura 4.22. Servicio HTTP con un sólo nivel de actividad por cliente

Se modela el flujo de datos a partir de las características del tráfico HTTP que permiten definir dos estados.

- Descarga: estado ON se está descargando la página WEB del servidor
- Lectura: estado OFF se está realizando la lectura del contenido de la página WEB.

Los estados ON y OFF varían aleatoriamente siguiendo una distribución de Pareto con los valores especificados en la tabla 4.1 [12].

Estado	Distribución	Media	Varianza
ON	Pareto	200	400
OFF	Pareto	30	60

Tabla 4.3. *Parámetros del modelo WEB*

Los paquetes son transportados por TCP, por el puerto de comunicación número 80 y implementado de extremo a extremo en el simulador. El *Data Rate* para la conexión TCP es de 1024 kbps y los paquetes generados tienen un tamaño de 1024 bytes.

La codificación que implementa la aplicación de HTTP para la generación del tráfico con las características comentadas es la siguiente:

```
OnOffHelper serHttp = OnOffHelper("ns3::TcpSocketFactory", InetSocketAddress(ip, 80));
serHttp.SetAttribute("OnTime", StringValue("ns3::ParetoRandomVariable (200,400)"));
serHttp.SetAttribute("OffTime", StringValue("ns3::ParetoRandomVariable (30,60)"));
serHttp.SetAttribute("PacketSize", UIntegerValue(1024));
serHttp.SetAttribute("DataRate", DataRateValue(1024));
```

4.8.5 Modelado del tráfico FTP

En el modelado del flujo de datos FTP se estima el envío de un archivo 30 MB de tamaño en momentos aleatorios de la simulación en sentido servidor-cliente. El escenario se representa en la Figura 4.7.



Figura 4.23. Se envía un archivo en sentido servidor-cliente

Para crear el tráfico se hace uso del Helper *BulkSendHelper* limitando el parámetro, *tamArchFtp*, mediante el que se le indica la cantidad de bytes que se desea enviar, para nuestro caso definiremos dicho parámetro como 30000000, ya que se desea enviar un archivo de 30 MB. La aplicación se activa en diferentes momentos de la simulación para comprobar el impacto de un uso tan intensivo que hace de la red este tipo de tráfico. El tráfico es transportado mediante el protocolo TCP y por el puerto 20.

La codificación que implementa la aplicación de FTP para la generación del tráfico con las características comentadas es la siguiente:

```
BulkSendHelper serFtp ("ns3::TcpSocketFactory", InetAddress (ip, 20));
serFtp.SetAttribute ("MaxBytes", UIntegerValue (tamArchFtp));
```

4.9 Modelado de herramientas para captura y representación de los datos

El simulador tiene una serie de herramientas para capturar y representar los datos obtenidos de la simulación. Para el presente PFC se ha hecho uso, principalmente, de cuatro herramientas del simulador para dicho fin.

- *FlowMonitor*.
- *GnuPlot*.
- *NetAdmin*.
- *Pyviz*.

4.9.1 FlowMonitor

Es una clase diseñada para el estudio del tráfico a nivel del nivel 3 de Internet, usado para el estudio de los parámetros de QoS de la red diseñada. Se hace uso de la clase mediante el *Helper FlowMonitorHelper*. Con *FlowMonitor* se hace un estudio de los parámetros de:

- *Throughput*.
- *Retraso*.
- *Jitter*.
- *Pérdidas*.

Debido a que el *FlowMonitor* hace, predeterminadamente, una media de los parámetros durante el tiempo total de la simulación y para el análisis de la infraestructura del presente PFC se requiere hacer un estudio en tiempo real de simulación, se ha creado un método llamado *limpiarEstadisticasDelFlujo* que cada vez que es llamado se le pasa el id del flujo al que se le desea inicializar los parámetros estudiados. Esto permite hacer una media cada segundo de los parámetros de la herramienta para posteriormente inicializarlos para un nuevo ciclo.

La codificación del método comentado es la siguiente:

```
void FlowMonitor::limpiarEstadisticasDelFlujo(FlowId IdFlujo)
{
    FlowStats &estadisticas = GetStatsForFlow (IdFlujo);

    estadisticas.rxPackets = 0;
    estadisticas.rxBytes = 0;
```



```

estadisticas.txPackets = 0;
estadisticas.txBytes = 0;
estadisticas.lostPackets = 0;
estadisticas.timesForwarded = 0;
estadisticas.delaySum = Seconds (0);
estadisticas.jitterSum = Seconds (0);
estadisticas.lastDelay = Seconds (0);
estadisticas.timeFirstRxPacket = Seconds (0);
estadisticas.timeLastRxPacket = Seconds (0);
estadisticas.timeFirstTxPacket= Seconds (0);
estadisticas.timeFirstRxPacket= Seconds (0);
estadisticas.timeLastTxPacket= Seconds (0);
}

```

Cada segundo se hace una captura de los parámetros y se llama al método *limpiarEstadisticasDelFlujo* para inicializarlos haciendo una media de los parámetros en dicho intervalo de la siguiente manera:

$$\text{Throughput} = \text{rxBytes} * 8 / (\text{timeLastRxPacket} - \text{timeFirstRxPacket})$$

$$\text{Retraso} = \text{delaySum} / \text{rxPackets}$$

$$\text{Jitter} = \text{jitterSum} / \text{rxPackets} - 1$$

Para el análisis de las pérdidas con más nivel de detalle se hace uso de otro mecanismo los *callbacks*, debido a que en para analizar las simulaciones queremos hacer un estudio del impacto de la señal inalámbrica en las perdidas de paquetes a nivel de la capa física y su impacto en los parámetros de QoS. El *FlowMonitor*, tal y como está diseñado, no vale para dicho fin debido a que trabaja en la capa tres del modelo ISO. La tarea que realiza dicho mecanismo es la llamada a dos funciones según el paquete se reciba con éxito o se pierda.

La declaración de dicha herramienta es la siguiente:

- Config::ConnectWithoutContext("/NodeList/1/DeviceList/*/\${ns3::WifiNetDevice/Phy/PhyRxEnd", MakeCallback (&paqueteRecibido));
- Config::ConnectWithoutContext("/NodeList/1/DeviceList/*/\${ns3::WifiNetDevice/Phy/PhyRxDrop", MakeCallback(&paquetePerdido));

Cada vez que un paquete es recibido correctamente en la capa física llama a la función *paqueteRecibido* pasándole como parámetro el paquete recibido y cada vez que un paquete se pierde se llama a *paquetePerdido* pasándole como parámetros el paquete perdido.

Parte de los atributos y métodos que contiene la clase están representados en la Figura 4.24.

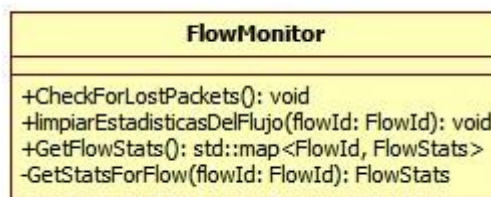


Figura 4.24. Diagrama de la clase *FlowMonitor*

Métodos:

- *CheckForLostPackets*: método que realiza una comprobación de paquetes perdidos. El creador de la aplicación recomienda llamar a este método antes de proceder a la lectura de los valores que se requieran.
- *limpiarEstadisticasDelFlujo*: método creado para este PFC con el objetivo de limpiar los valores de seguimiento cada vez que es llamado.
- *GetFlowStats*: método que recupera las estadísticas de todos los flujos monitorizados.
- *GetStatsForFlow*: método al que se le pasa por parámetro el id de un flujo que queramos recuperar y nos lo devuelve.

Para instalar los monitores de flujo en los nodos que se desea estudiar se hace uso del Helper *FlowMonitorHelper*, el cual tiene un método *Install* para ello.

4.9.2 Gnuplot

Para la realización de las gráficas se ha creado un método llamado *imprimirGrafica* que genera un *archivo.plt* y un *archivo.png* con la imagen de la gráfica del parámetro que se desea estudiar. La clase *Gnuplot2dDataset* actúa como un colector de los datos que se desean estudiar y la clases *Gnuplot* crea la estructura del *archivo .plt* para posteriormente ser manejado con el programa Gnuplot.

Parte de los atributos y métodos que contiene la clase están representados en la Figura 4.25.

Métodos:

- *SetStyle*: permite definir estilo de representación de la gráfica que se desea representar. Los estilos disponibles son *LINES*, *POINTS*, *LINES_POINTS*, *DOTS*, *IMPULSES*, *STEPS*, *FSTEPS* y *HISTEPS* para las distintas representaciones realizadas en el presente PFC se ha seleccionado *LINES*.
- *Add*: método que va añadiendo los valores que se desean estudiar al *dataSet* para posteriormente representarlos mediante la clase *gluplot*.

Mediante la clase *Gnuplot* se especifican los datos que se van a representar, el tipo y nombre del archivo de la imagen que queremos que nos genere, rango de valores a representar y el título de la gráfica. Parte de los atributos y métodos que contiene la clase están representados en la Figura 4.26.

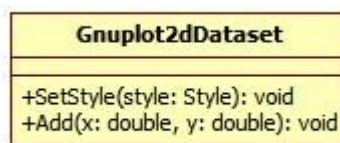


Figura 4.25. Diagrama de la clase *Gnuplot2dDataset*

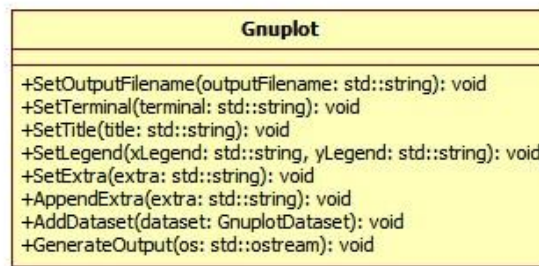


Figura 4.26. Diagrama de la clase Gnuplot

Métodos:

- *SetOutputFilename*: asigna nombre, pasado por parámetro, al archivo se salida que se genera.
- *SetTerminal*: selecciona un formato de salida para la imagen creada mediante el programa gnuplot.
- *SetTitle*: configura el título de la gráfica.
- *SetLegend*: configura las leyendas de los datos que se representan.
- *AppendExtra*: define los rangos de los datos que se representan.
- *AddDataset*: agrega un dataset a la gráfica, el cual contiene el conjunto de datos que se desean representar.
- *GenerateOutput*: genera el archivo.plt, preparado para ser tratado con el programa gnuplot.

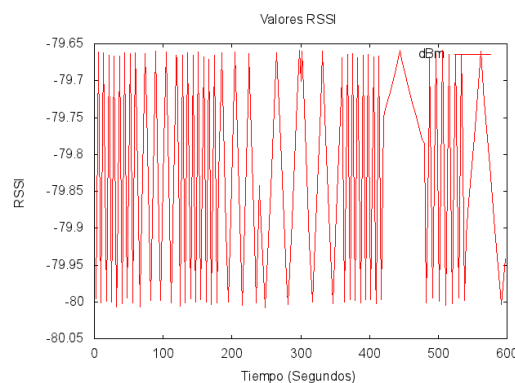


Figura 4.27. Gráfica de una simulación generada por gnuplot

4.9.3 Herramientas de animación de simulaciones

NetAdmin es un animador offline basado en el conjunto de herramientas *Qt4*. Con los datos recogidos, utilizando un archivo de rastreo *eXtensible Markup Language (XML)*, crea una animación la simulación. Para crear el archivo de rastreo XML ns3 usa la clase *AnimationInterface*. En el presente PFC es de gran utilidad para animar las simulaciones realizadas y analizar la evolución del RSSI del nodo en base a su posición.

Parte de los atributos y métodos que contiene la clase están representados en la Figura 4.28.

Métodos:

- *SetNodeDescription*: permite escribir una descripción sobre el nodo, o un contenedor de nodos, representado en la animación.
- *UpdateNodeDescription*: permite actualizar la descripción de un nodo durante la simulación.
- *SetNodeColor*: permite seleccionar un color para un nodo, o un contenedor de nodos, representado en la animación.
- *UpdateLinkDescription*: permite cambiar la descripción del canal de comunicación entre dos nodos unidos mediante conexiones punto a punto.
- *EnablePacketMetadata*: activa el almacenamiento de los metadatos de un paquete en el archivo XML.

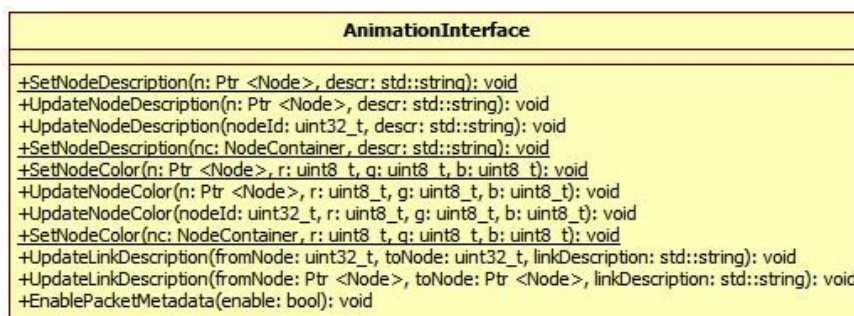


Figura 4.28. Diagrama de la clase *AnimationInterface*



Figura 4.29. Animación con NetAdmin

Pyviz es una herramienta que permite realizar una animación en tiempo real de la simulación y visualizarla en tiempo de ejecución. Ha sido de gran utilidad principalmente para depurar el código de la simulación. Representa los nodos implicados en la simulación, estáticos o dinámicos, su movilidad y el tráfico que soportan.

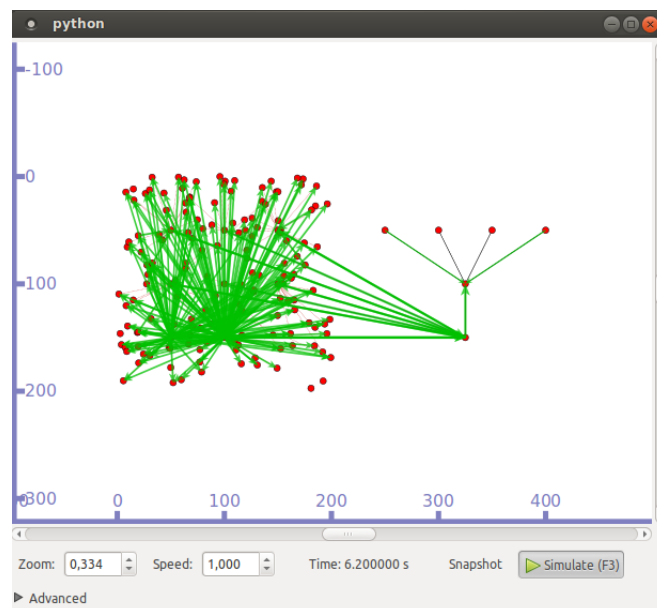


Figura 4.30. Ejecución de Pyviz

4.10 Diagramas de casos de uso y de secuencia

Con los diagramas de casos de uso y los diagramas de secuencia se trata de explicar cómo interactúan las diferentes clases que están implicadas en las simulaciones de un escenario inalámbrico en la que existe flujo de datos. Con ese objetivo se realizan diferentes diagramas de uso *Unified Modelling Language (UML)* donde quede claro el rol de cada una de las clases, el actor que las diseña y el que las usa, así como un diagrama de secuencia para tratar de aclarar como fluye la información.

4.10.1 Casos de uso y diagrama de secuencia del canal inalámbrico

Para el canal inalámbrico el diseñador configura los *Helpers* relacionados con el canal inalámbrico y las dimensiones de los edificios mediante la clase *Building*. A los parámetros configurados en mediante la clase *Building* y la movilidad con el Helper *MobilityHelper* recurre la clase *BuildingsHelper* para crear una *MobilityBuildingInfo* de la cual hace uno la *HybridBuildingPropagationLossModel* para el cálculo de las pérdidas en la señal radioeléctrica.

En la Figura 4.31 se representa un diagrama de secuencia donde se aclara cómo interactúan las diferentes clases implicadas en la simulación del canal inalámbrico. En el diagrama se pueden diferenciar los procesos en los que se configuran los distintos objetos que se pretenden simular y los procesos de flujo de información entre los mismos.

En la Figura 4.32 se representa un diagrama de casos de uso del canal inalámbrico con el objetivo de diferenciar las clases que se configuran por parte del diseñador de la simulación y los objetos sobre los que se actúa.

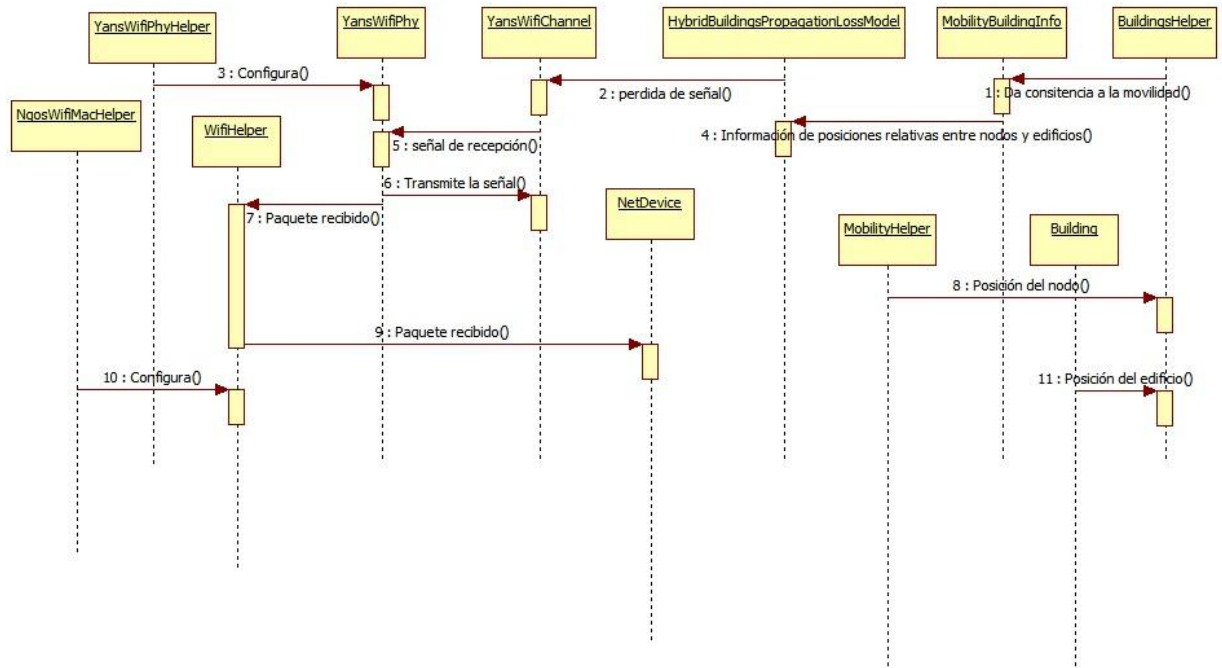


Figura 4.31. Diagrama de secuencia de canales inalámbricos

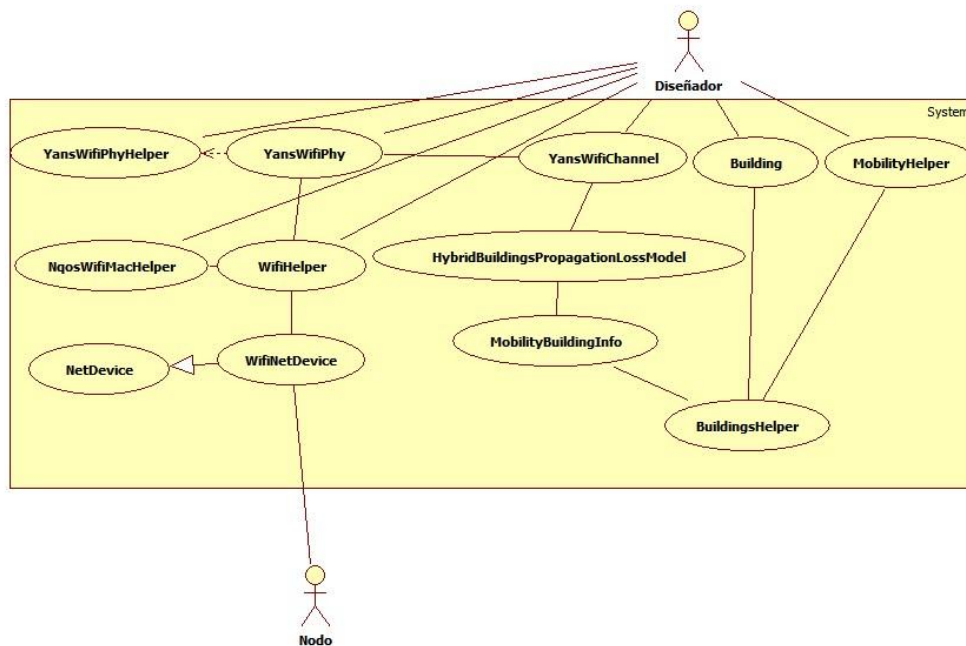


Figura 4.32. Casos de uso de los canales inalámbricos

4.10.3 Casos de uso y diagrama de secuencia del tráfico

El diseñador configura los *Helpers OnOffHelper*, *UdpTraceClient* y *BulkSendHelper* para generar tráfico desde los nodos transmisores y configura el *PacketSinkHelper* para que los nodos receptores puedan recibir el tráfico. Cada paquete de datos es modelado mediante la clase *Packet* que se encarga de crear un objeto con las características de un paquete de datos que interactúa con todos los objetos señalados en verde. El diseñador configura la clase *Ipv4AddressHelper* con el direccionamiento que desee para la red señalada e interactúa con la clase *Ipv4* que es la encargada de gestionar el protocolo de direccionamiento. Todos los objetos de tráfico interactúan con *InternetStackHelper* que es el encargado de tratar el paquete a través de sus diferentes capas que modelan los protocolos de Internet.

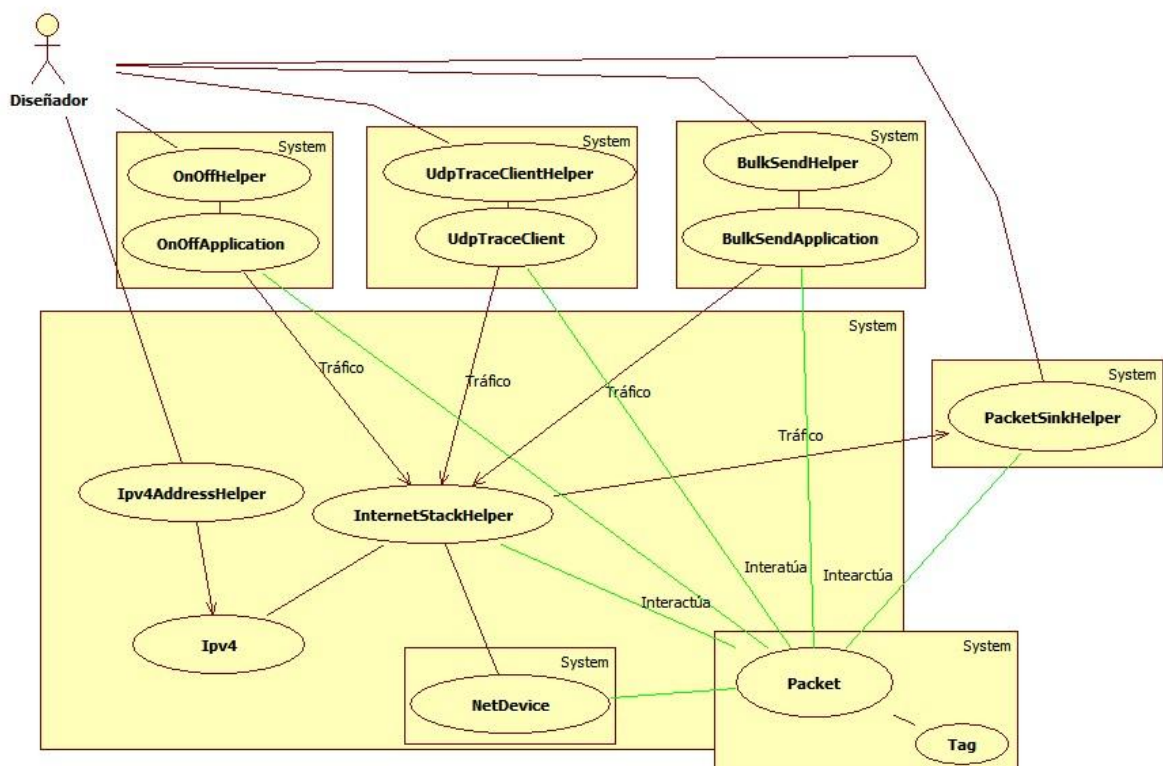


Figura 4.33. Casos de uso de tráfico

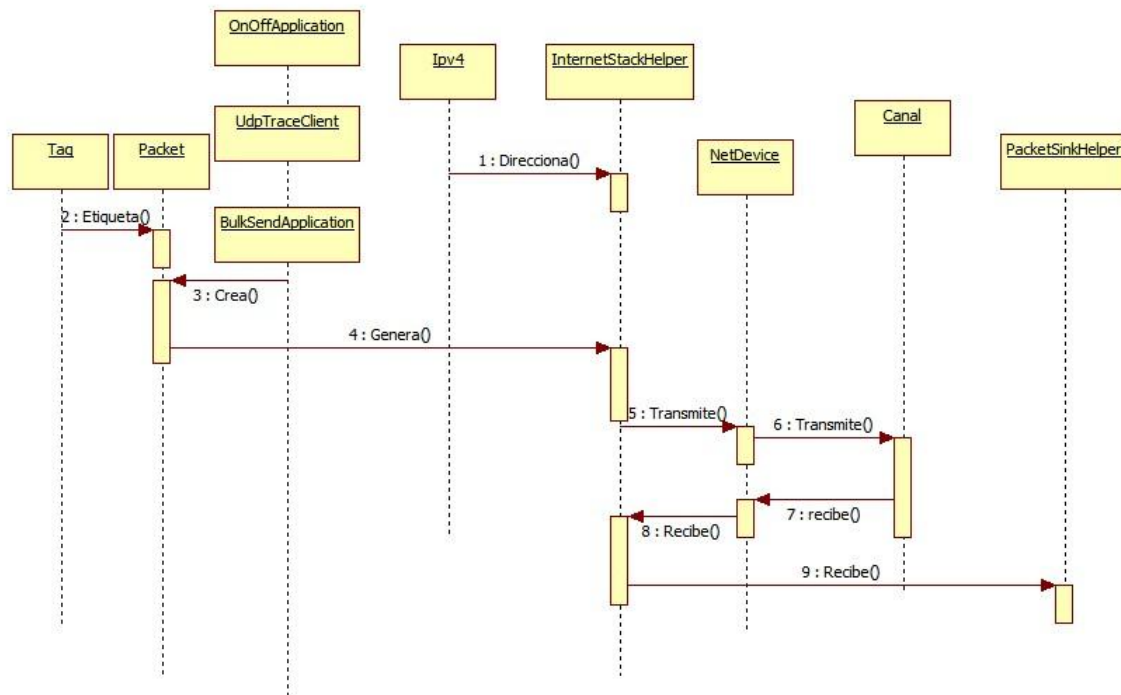


Figura 4.34. Diagrama de secuencia de tráfico

5. Diseño del Proyecto

En este capítulo se trata la problemática de la situación actual de la infraestructura del complejo, se realiza el diseño de la infraestructura objeto del presente PFC y se detalla un mapa de cobertura.

5.1 Introducción

El alcance de un sistema inalámbrico se puede definir como la distancia física y lineal entre dos puntos que permiten una conexión inalámbrica posible y es el principal objeto de estudio en este PFC, para poder realizar de forma óptima el dimensionado en la cantidad de puntos de acceso a repartir por el complejo y seleccionar las mejores localizaciones para dichos puntos. Pero también sabemos que la forma de la onda del espectro radioeléctrico de las señales radioeléctricas no es lineal sino que presenta diferentes tipos en función de las antenas usadas.

En una comunicación entre antenas direccionales, quizás su alcance sea de varios Km, pero a lo mejor con un tercer punto y manteniendo los mismos equipos esta comunicación no es posible. Por lo tanto, aunque el alcance de una antena depende también de factores como los obstáculos o las interferencias, lo que se suele hacer es realizar el cálculo suponiendo unas condiciones ideales y, posteriormente, estimar unas pérdidas adicionales por falta de condiciones ideales. Para realizar los cálculos teóricos del alcance de una señal, considerando todos los posibles obstáculos, recurriremos a ns3, obteniendo datos teóricos en la simulación muy cercanos a la realidad.

Hay que tener en cuenta, además, que las señales electromagnéticas se propagan por el medio a una velocidad cercana a la de la luz. Incluso tienen la capacidad y habilidad de poder traspasar paredes, techos, puertas o cualquier obstáculo (teóricamente). Además, gracias al fenómeno conocido como difracción, pueden colarse por los pequeños agujeros. En cualquier caso, unos obstáculos los pasa más fácilmente que otros.

Además de las pérdidas de propagación, en una instalación inalámbrica existen distintos dispositivos que producen pérdidas o aportan ganancia a la señal. El cálculo teórico del alcance de una transmisión se basa en sumar los factores de la instalación que aportan ganancias y restar los que producen pérdidas. Al final, obtenemos un nivel de señal. El que este nivel de señal sea suficiente para una buena recepción depende del equipo receptor. Pero hay que calcular el proceso inverso, es decir las comunicaciones

inalámbricas son siempre bidireccionales y los datos técnicos para cada equipo son diferentes si están emitiendo o recibiendo. Esto es, un cliente puede transmitir datos a un AP y este no recibirlos, y al contrario, ya que las ganancias de emisión y recepción pueden no ser las mismas.

Las antenas y amplificadores inalámbricos añaden ganancia. Al igual que las tarjetas y los APs pero los conectores y los cables añaden pérdidas.

Con todo lo anterior, es importante saber que la respuesta sobre cuál puede ser el alcance de un sistema inalámbrico se consigue mediante de la experimentación y una correcta simulación en la que se tenga en cuenta todos los factores importantes de la infraestructura a dimensionar. Es por eso que, a priori, se realiza una estimación en base a los datos arrojados por la simulación de cuál va a ser la distribución de los APs de la red diseñada.

5.1 Consideraciones del complejo

Se trata de un complejo hotelero que, desde el punto de vista de este proyecto, cuenta con cuatro zonas diferenciadas:

- Zonas de instalaciones de ocio con bar-cafetería, teatro y que incluye el comedor del personal.
- Zona de oficinas y recepción.
- Zona de habitaciones de clientes.
- Zona de apartamentos del personal que incluye las viviendas de diversos directivos del hotel.

La zona de habitaciones de clientes está dispuesta en diversos bloques de dos o tres plantas que se encuentran unidos por estructuras de madera a través de las cubiertas. Los bloques son los denominados: fuerteventura (F), Tenerife (T), La Palma (P), Gran Canaria (GC), Lanzarote (L) y La Gomera (LG).

La zona de apartamentos de personal se encuentra al norte del complejo y está constituido por bloques de dos y cuatro plantas. Estos bloques están denominados como El Hierro (H). A estos bloques se unen dos viviendas unifamiliares destinadas a directivos del hotel también denominadas con el distintivo de El Hierro (H).

En el extremo nordeste del complejo se sitúa el cuarto de instalaciones destinado a la cabecera principal del hotel que da servicio a las zonas de instalaciones de ocio con bar-cafetería, teatro y que incluye el comedor del personal. La distribución del complejo se representa en la Figura 5.1.

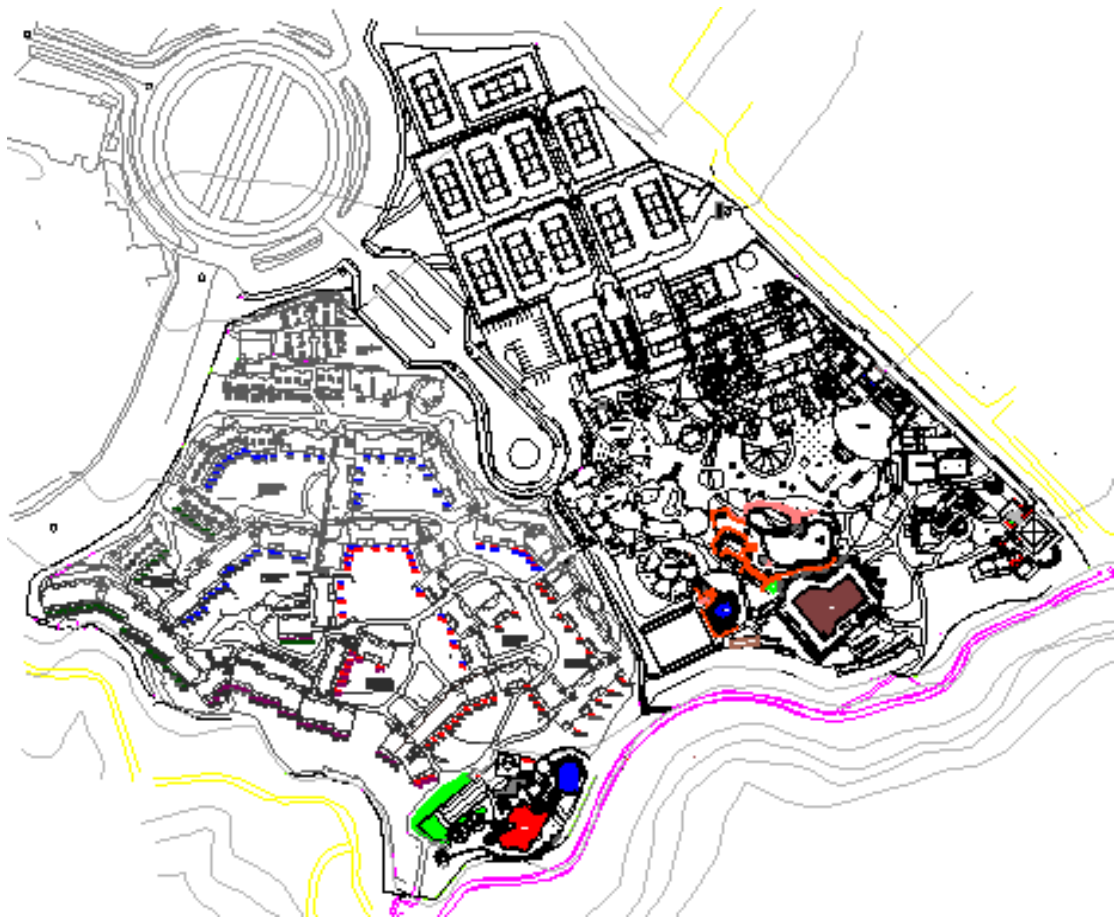


Figura 5.1. Plano del complejo hotelero

5.2 Situación actual

Se realiza un análisis de la situación actual de la red de telefonía interna del complejo hotelero y las características de la red de fibra óptica que es usada como base para la instalación de los APs que proveen la cobertura para la infraestructura inalámbrica diseñada en el presente PFC.

5.2.1 Red de telefonía y acceso a Internet

A continuación se detalla la situación en la que se encuentra las infraestructuras de telecomunicaciones que proveen actualmente los servicios de telefonía y acceso a Internet en el complejo.

Infraestructura de Telefonía

La instalación de la infraestructura telefónica actual del complejo consta de dos redes diferenciadas denominadas Red de Distribución y Red de Dispersión, así como unos puntos de distribución donde finaliza la Red de Distribución y se inicia la Red de Dispersión.

- *Red de Distribución:* se inicia en el Punto de Interconexión, situado en el cuarto de comunicaciones, detrás de la recepción. Este punto de interconexión tiene como misión hacer de puente entre la instalación de telefonía básica del complejo y la centralita telefónica que gestiona el trasiego de llamadas de entrada y salida.
- *Red de Dispersión:* partiendo de los puntos de distribución la Red de Dispersión formada por cables telefónicos de un par se dirige a las bases de acceso telefónico situados en el interior de cada una de las habitaciones.

La Red de Dispersión de las plantas bajas de los bloques situados al Sur del complejo en particular, a una cota más baja, se han visto afectados por alguna inundación que han afectado a su funcionamiento y que, por otra parte, siguen estando expuestos en caso de futuras lluvias.

Por diversas cuestiones, entre las que se encuentra el mal estado de las canalizaciones, es bastante dificultosa la sustitución física tanto de la Red de Distribución como la de Dispersión, por lo que se propone sustitución de dichas redes integrándolas dentro de la red inalámbrica objeto de estudio del presente PFC, mediante tecnología para proveer VoIP.

Acceso a Internet

El complejo hotelero cuenta con acceso a Internet a través de ADSL disponible para los clientes. Sin embargo, no pueden acceder a este servicio desde sus habitaciones viéndose obligados a desplazarse a la recepción para disfrutar de dicho acceso. El servicio se ofrece a través de un AP doméstico y para disfrutar del mismo se requiere disponer de la clave WEP de la red inalámbrica.

5.2.2 Red de fibra óptica

Se parte de una idea inicial unificar todos los servicios de telecomunicaciones del hotel. Siguiendo dicha idea la red de fibra óptica se convierte en la base para la instalación de la red inalámbrica diseñada desarrollando un papel fundamental ya que se encarga de distribuir y gestionar el tráfico de información que recogen las distintas antenas inalámbricas.

La red inalámbrica aporta un valor añadido y supone una solución eficiente, ya que combina la velocidad de la transmisión que proporciona la fibra óptica con el alcance y velocidad del sistema WiFi, lo que la convierte en una fórmula perfecta para el acceso a Internet sin cables.

La red de fibra óptica se ha dimensionado para permitir el traslado y distribución a muy alta velocidad los paquetes de servicios multimedia y de datos, contemplándose además la posibilidad de inclusión de nuevos servicios utilizando el soporte de transmisión que ofrece, para lo cual se han dejado fibras libres que permitirían la implementación de dichos servicios.

Se ha determinado realizar el tendido de cable multimodo de 48 fibras ópticas para la transferencia de datos en el cuál se dejaría un excedente de fibras para futuras aplicaciones que puedan surgir en el complejo hotelero. El cable de fibra óptica multimodo recorrería un tramo canalizado que parte desde la recepción del hotel hasta alcanzar la zona Fuerteventura, a partir de la cual se distribuiría al resto del complejo.

En los distintos puntos del complejo, donde se situarían los diferentes puntos de acceso de la red inalámbrica, se realizaría el preceptivo empalme de fibras ópticas que permita llevar la señal hasta cada terminación con su punto de acceso inalámbrico.

Para el estudio objeto de este PFC partimos de la red de fibra óptica comentada, como ya establecida, haciendo el estudio y dimensionado sólo de la parte inalámbrica. En la Figura 5.2 se detalla un mapa con su dotación.

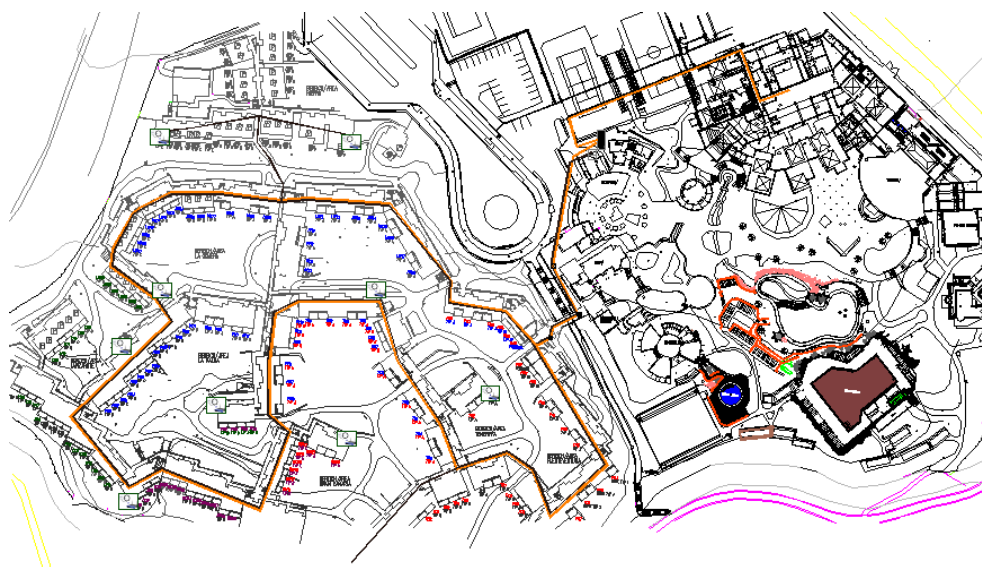


Figura 5.2. Despliegue de fibra óptica

5.3 Dimensionado

Para el dimensionado se han estudiado tres escenarios simulados, analizando los diferentes resultados y en base a ellos se ha propuesto una distribución que se analiza con más profundidad en el capítulo de 6. Para las simulaciones se han modelando señales con una frecuencia de 2,4 Ghz debido a que se busca la total compatibilidad entre los estándares IEEE 802.11b, IEEE802.11g y IEEE802.11n para la infraestructura objeto de estudio.

5.3.1 Escenario exterior

El primer escenario a estudiar consta de dos nodos, un nodo AP y un nodo cliente, en un espacio abierto sin obstáculos. El nodo AP se posiciona a una altura de tres metros del suelo. El nodo cliente se posiciona inicialmente justo debajo del nodo AP y a 1 metro de altura comenzando, en el evento 1 segundo de la simulación, un movimiento de alejamiento lineal respecto al nodo AP a una velocidad de 0,5 mps. El escenario se representa en la Figura 5.3.

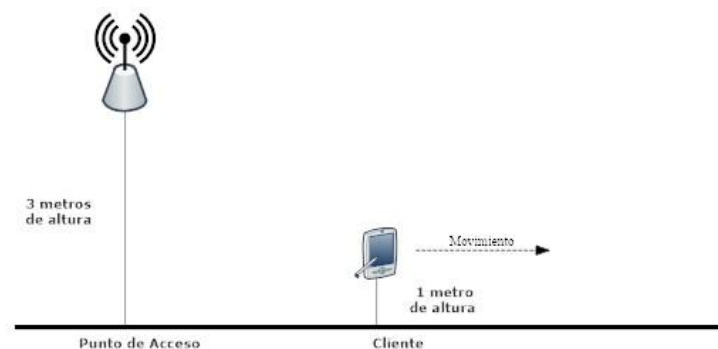


Figura 5.3. Escenario exteriores

La gráfica resultante de dicha simulación se representa en la Figura 5.4 en la cual se observa cómo se mantienen los valores del RSSI dentro de un rango aceptable, establecemos el rango óptimo de señal por debajo de los -80 dBm, hasta los 200 m de distancia.

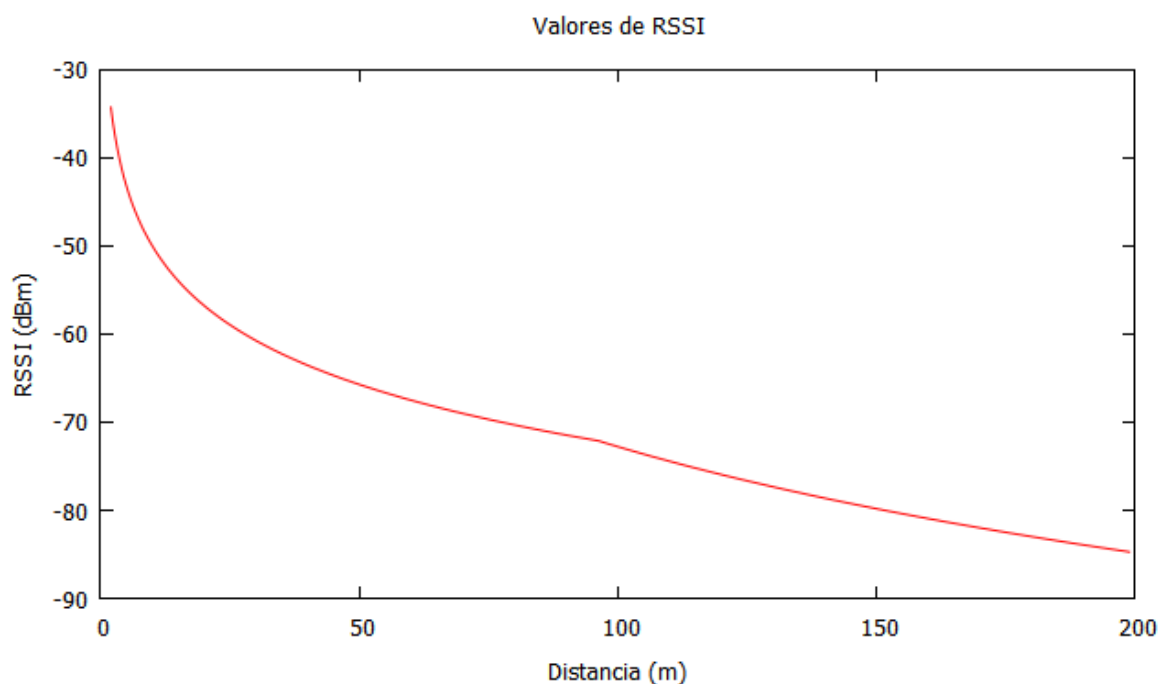


Figura 5.4. Degradación del RSSI

5.3.2 Escenario exterior con edificios

El segundo escenario a estudiar, consta de dos nodos en un espacio abierto con un edificio en medio de la comunicación para estudiar cómo afecta su presencia. El nodo AP está situado a una altura de tres metros del suelo y a 10 metros de distancia del edificio que está situado en medio. El edificio ocupa un volumen de 2mx2mx4m. El nodo cliente está a 1m de altura del suelo, e inicialmente, junto a la pared del edificio desde donde comienza un movimiento aleatorio dentro de un recuadro en la parte posterior del edificio. Teniendo en cuenta los resultados de la simulación anterior definimos la zona por donde se desplazará el nodo cliente dentro de los 200m donde el RSSI de la señal se demostró aceptable en áreas abiertas. El movimiento que se define para el nodo cliente es un cambio de posición cada 5 segundos y un desplazamiento en cada cambio con velocidad de 0,5mps. El escenario se representa en la Figura 5.5.

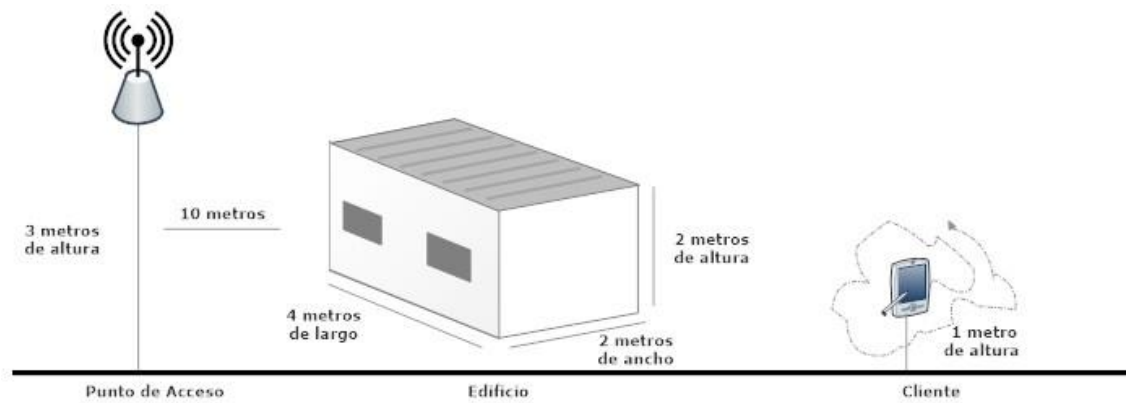


Figura 5.5. Escenario exterior con edificios

En éste escenario al ser un movimiento aleatorio y no lineal en la parte posterior del edificio representamos los datos de la distancia y del RSSI respecto al tiempo en la Figura 5.6.

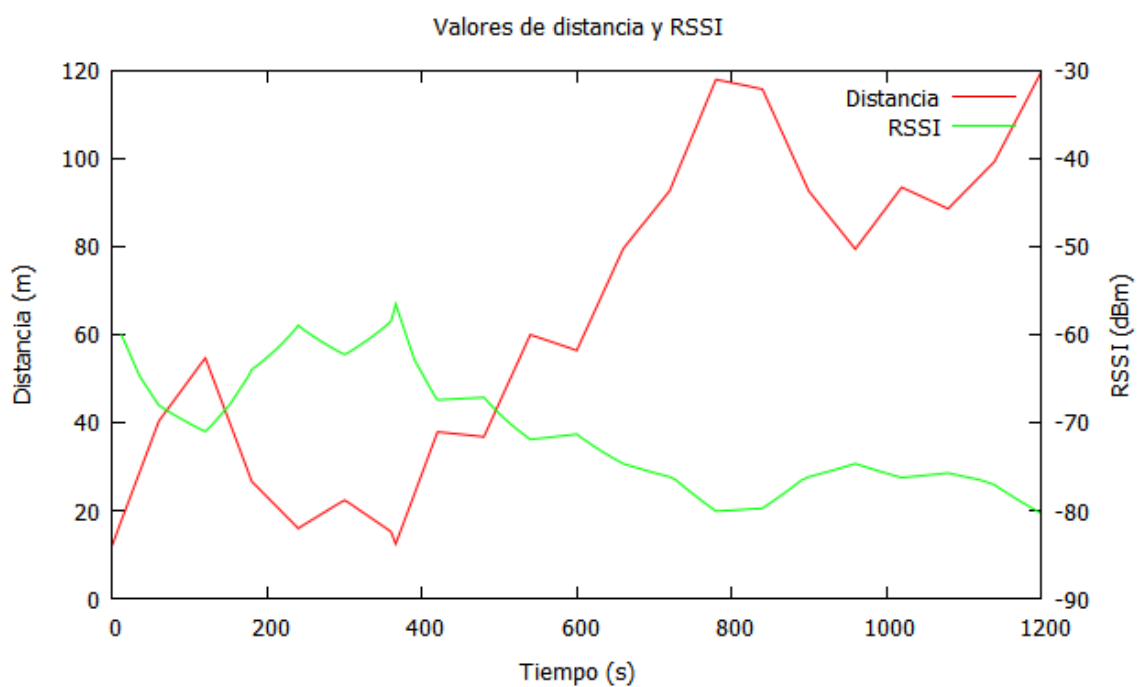


Figura 5.6. Variación de la distancia y RSSI

En el nuevo escenario empeoran los valores de RSSI, comparando los resultados con los datos representados en la gráfica anterior sin edificios que afecten negativamente a la señal, no llegando estos a valores a sobrepasar los -55 dBm.

La señal sufre la mayor degradación y sobrepasa el umbral de -80 dBm a partir de los 100 metros de distancia, poco antes del segundo 800 de la simulación, dando una media de de 10 dBm más de degradación en comparación con la gráfica del primer escenario para las mismas distancias.

5.3.3 Escenario exterior e interior

El tercer escenario a estudiar, similar a los anteriores, consta de dos nodos en un espacio abierto en el que el nodo cliente se sitúa dentro del edificio. El nodo AP se sitúa a una altura de tres metros del suelo, el edificio del escenario planteado tendría unas dimensiones de 2mx2mx4m y estaría situado a 50m del nodo AP. El nodo cliente está a 1m de altura del suelo y se posiciona inicialmente en el centro del edificio comenzando un movimiento aleatorio por todo su interior similar al del escenario anterior cambiado de posición cada 5s y desplazándose a una velocidad de 0,5 mps. El escenario se representa en la Figura 5.7.

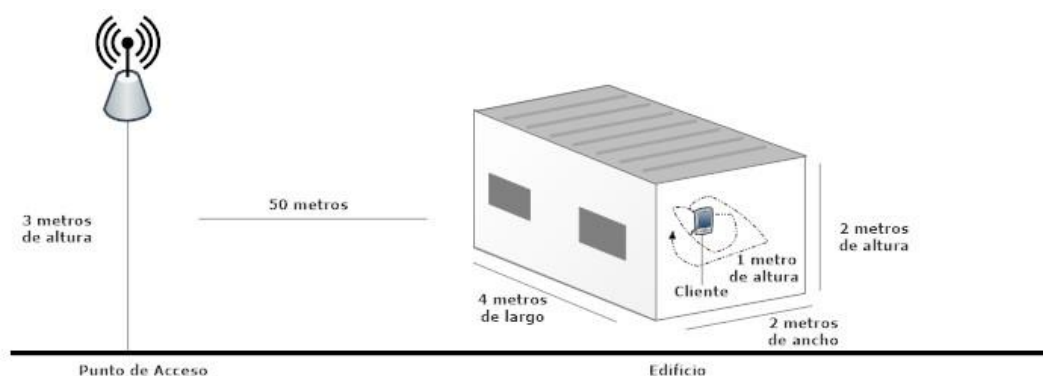


Figura 5.7. *Escenario exterior e interior*

Representamos los valores de la distancia y del RSSI respecto al tiempo en la gráfica de la Figura 5.8.

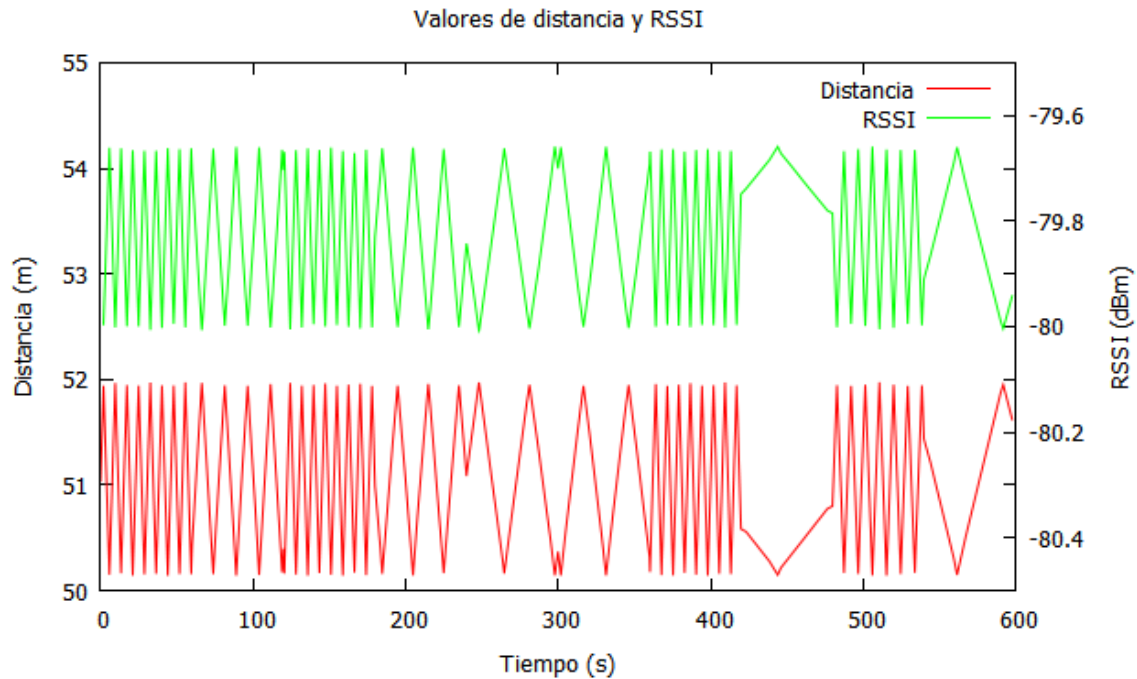


Figura 5.8. Variación de la distancia y RSSI

Para un rango de distancia entre 50 y 52 m, el desplazamiento del nodo cliente en el interior del edificio, la señal sufre una degradación de casi 20 dBm respecto al escenario sin obstáculos y de 10 dBm respecto al escenario en el que el nodo se mueve por fuera del edificio.

5.3.4 Mapa de cobertura

Analizados los resultados de los tres escenarios y considerando el solape de canales, que no es tomada en cuenta por el simulador ns3, llegamos a la conclusión de que los nodos deberían estar repartidos por el complejo a una distancia de separación entre AP que debe de estar comprendida entre valores de 75 a 95 m y a una altura aproximada entre 3 y 4m dependiendo de cada una de las zonas donde se posicionen.

Siguiendo dichas premisas llegamos a la conclusión de que para ofrecer una cobertura de WiFi óptima con dichos datos teóricos debemos seguir la distribución de APs, con sus respectivas antenas como se representa en los mapas de cobertura de la

Figura 5.9 para la zona de clientes y de la Figura 5.10 (en los planos del anexo A se puede estudiar el despliegue con más detalle).

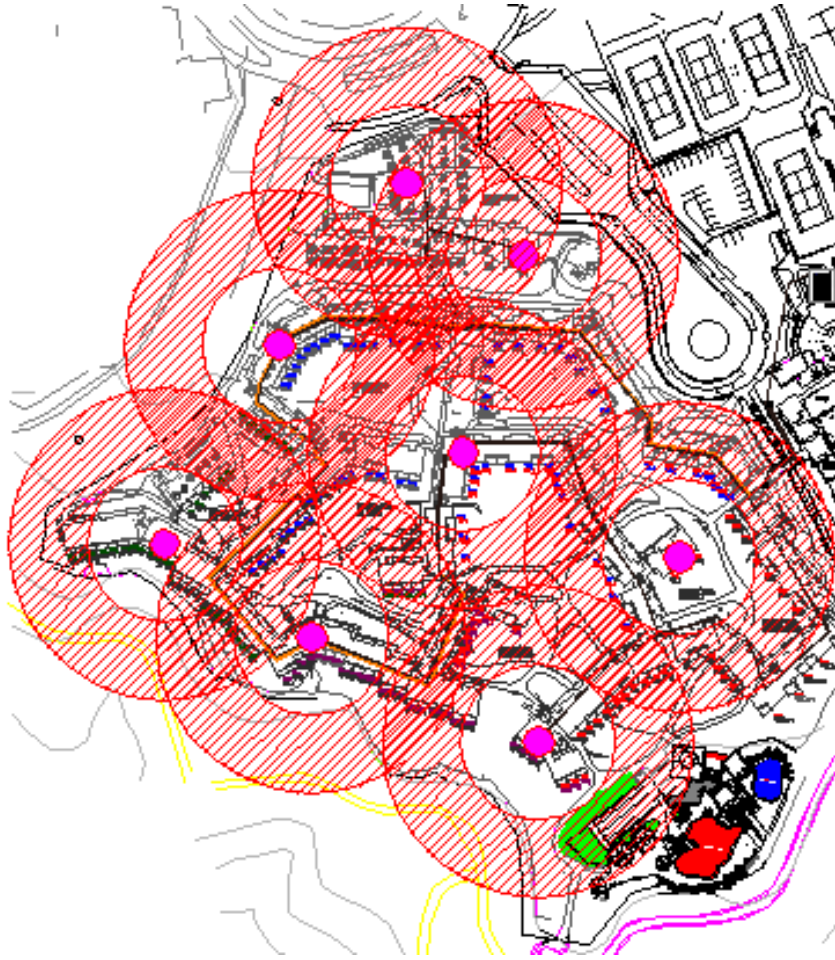


Figura 5.9. Mapa de cobertura para la zona de clientes

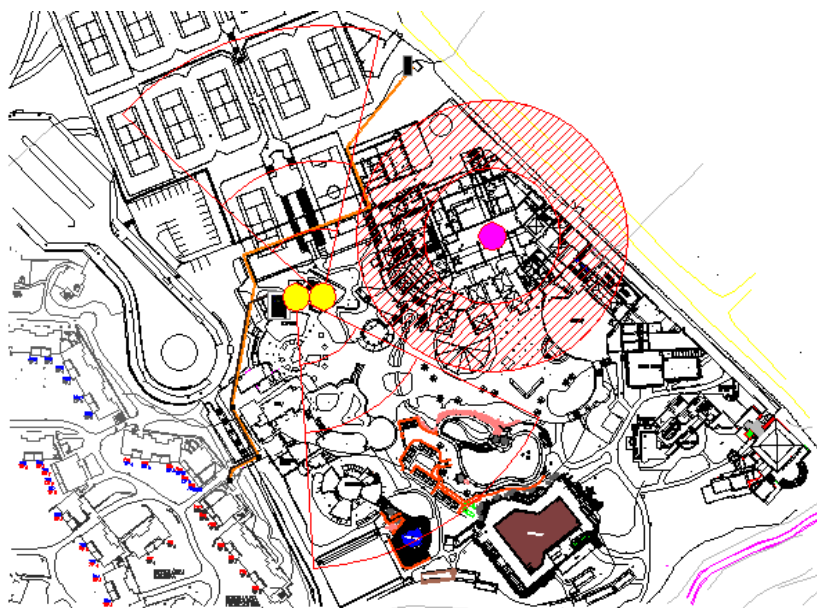


Figura 5.10. Mapa de cobertura para la zona de ocio.

6. Análisis de la infraestructura

En este capítulo se realiza una simulación de la futura infraestructura inalámbrica analizando su comportamiento con una carga de diferentes tipos de tráfico similar a la que pudiera producirse en la infraestructura real.

6.1 Introducción

Cada vez es más importante tener en cuenta el comportamiento de las redes de datos ante el tráfico de las diferentes aplicaciones que hacen uso de las mismas siendo las más críticas las que requieren alto grado de QoS, como suelen ser las aplicaciones multimedia. En el presente capítulo se analiza el impacto de los tipos de tráfico más comunes:

- Llamadas telefónicas.
- Transmisión de vídeo.
- Navegación Web.
- Transferencia de archivos.

El estudio se centra especialmente en cómo se ve afectado el tráfico de las llamadas de voz, debido a que la infraestructura diseñada va a soportar parte de la carga de las llamadas del complejo mediante la sustitución de la actual red de telefonía por VoIP.

Los parámetros que se van a analizar son los relacionados con la QoS sobre las aplicaciones modeladas, principalmente el cómo se ven afectados por la distancia y la carga de la red en dicho momento tratando de definir unos valores umbrales para los cuales se garanticen el correcto funcionamiento de la infraestructura.

El estudio se realiza sobre varios tipos de escenarios y casuísticas, comenzando con los escenarios más simples para finalmente centrarnos en los escenarios más cercanos al real.

6.2 Escenario 1

Se representa una comunicación punto a punto entre un nodo servidor y un nodo cliente, en movilidad, sobre una WLAN configurada en modo ad hoc en la cual se realiza una transferencia de información continua de datos en sentido servidor-cliente mientras el

terminal móvil se va alejado linealmente del servidor que se encuentra estático en una posición.. El escenario se representa en le figura 6.2.

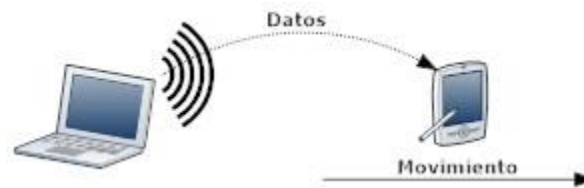


Figura 6.1. Escenario 1

El escenario se estudia cómo evoluciona el retraso, el jitter y las pérdidas de los paquetes en relación a la distancia entre los nodos para los diferentes estándares WiFi.

El tráfico enviado es *Constant Bit Rate (CBR)* de paquetes UDP con un tamaño de 1024 bytes cada 10 ms de tiempo y el tráfico comienza a transmitirse desde el segundo 3 de simulación. Se realizan tres simulaciones, una por cada estándar WiFi IEEE 802.11b, IEEE 802.11g y IEEE 802.11n. Las velocidades de los estándares IEEE 802.11b y IEEE 802.11g son automáticas reguladas por el algoritmo de velocidad predefinido en el simulador y la del IEEE 802.11n es constante a 13,5 Mbps. El nodo se desplaza hasta una distancia de 200m teniendo en cuenta las dimensiones físicas de la infraestructura diseñada en el capítulo 5.

Al aumentar la distancia se provoca un aumento de las pérdidas de paquetes, presentadas en la Figura 6.5, las cuales repercuten directamente en el resto de parámetros estudiados debido a la retransmisión de los paquetes perdidos.

En la gráfica de la Figura 6.2 se observa que los estándares IEEE 802.11b y IEEE 802.11g son menos estables que el estándar IEEE 802.11n durante toda la simulación, estando todos ellos por debajo de 1 ms de retraso, prácticamente, en toda la simulación a excepción del estándar IEEE 802.11b para el cual se dispara el retraso a partir de los 180 m. Un retraso tan bajo indica que el medio está todo el tiempo ocioso y las tramas recibidas son transmitidas inmediatamente sin llegar a saturar el canal de transmisión.

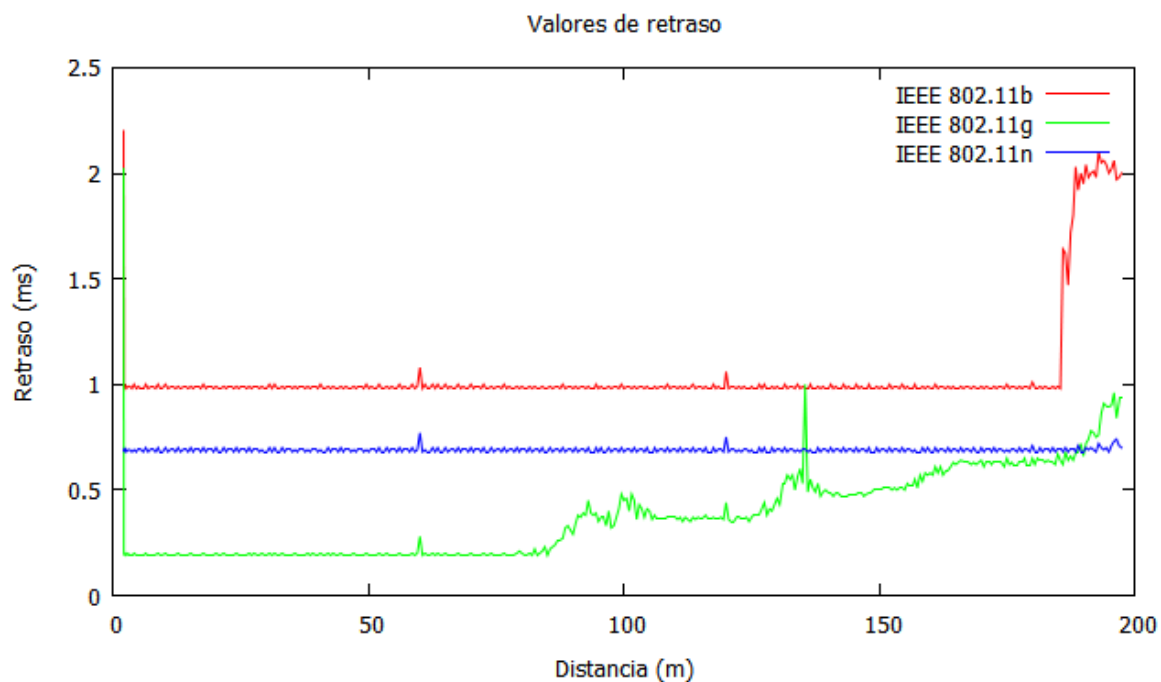


Figura 6.2. Gráfica de retraso de los diferentes estándares IEEE 802.11

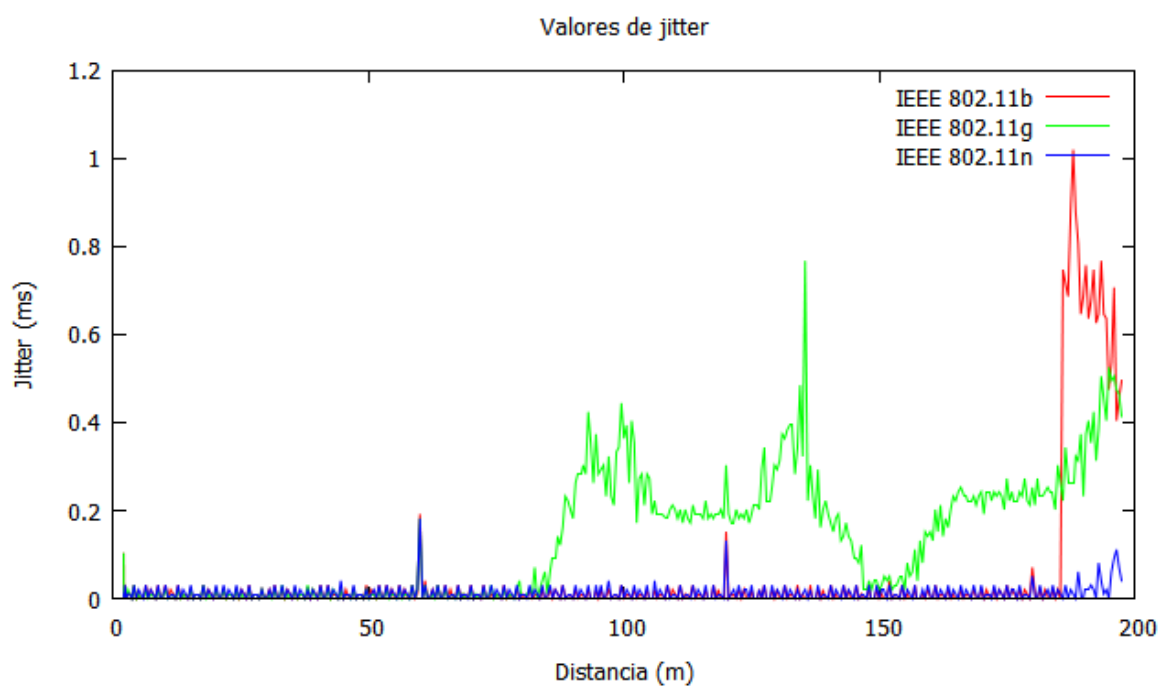


Figura 6.3. Gráfica del jitter de los diferentes estándares IEEE 802.11

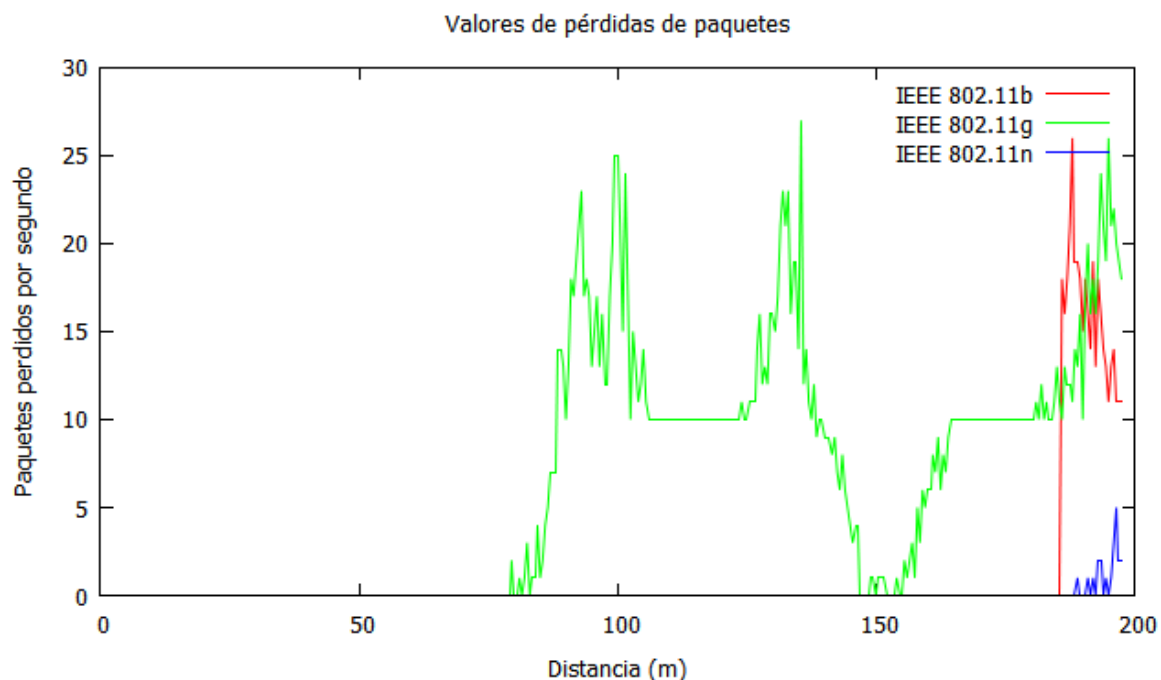


Figura 6.4. Gráfica de pérdidas de los diferentes estándares IEEE 802.11

El jitter, representado en la Figura 6.3, se mantiene prácticamente nulo para el estándar IEEE 802.11n y se observan pequeñas variaciones del retraso en los estándares IEEE 802.11b y IEEE 802.11g. Esto refleja las mejoras aportadas al estándar IEEE 802.11n siendo el más estable de los tres en los mismos rango de potencia de recepción y, por lo tanto, el más idóneo para la transmisión de datos sensibles a la variación de los parámetros QoS.

Queda reflejada la superioridad en estabilidad del estándar IEEE 802.11n teniendo una pérdidas que en ningún momento de la simulación superan los 5 paquetes por segundo para los mismos rangos de distancia que los estándares IEEE 802.11b y IEEE 802.11g.

6.3 Escenario 2

Se representa una comunicación en modo infraestructura entre un servidor y un cliente donde ambos están estáticos a una distancia de 50 m entre sí. La transferencia de datos

entre los nodos fluye en sentido cliente-servidor. El escenario se representa en la figura 6.6.

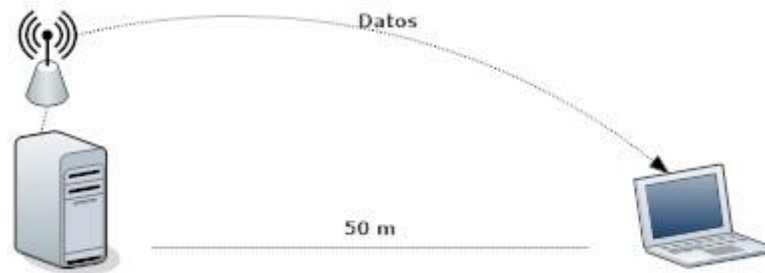


Figura 6.6. Escenario 2

El tráfico enviado es CBR y se envían 500000 paquetes UDP con un tamaño de 1024 bytes, el tiempo entre envío es de 0,4 ms, el tráfico comienza a transmitirse desde el segundo 3 de simulación.

El envío total es de 512MB de información. Se realizan tres simulaciones, una por cada estándar WiFi IEEE 802.11b, IEEE 802.11g y IEEE 802.11n. Las velocidades de cada uno de los estándares es la máxima posible, 11Mbps para IEEE 802.11b y 54Mbps para IEEE 802.11g y 150Mbps para IEEE 802.11n, éste último limitado por la tecnología SISO ya que el simulador no soporta MIMO para el módulo WiFi.

En la gráfica de la Figura 6.7 se observa el resultado del mejor desempeño del protocolo IEEE 802.11n ya que a la misma distancia de 50 m nos da mejores valores de throughput que los estándares IEEE 802.11b y IEEE 802.11g, lo que conlleva a un mejor ancho de banda para la transferencia de paquetes de información.

En las gráficas de las Figuras 6.8, 6.9 y 6.10 se observa el mejor desempeño de la especificación de IEEE 802.11n, en comparación con los estándares IEEE 802.11b y IEEE 802.11g, principalmente en los tiempos de retraso prácticamente nulo en comparación.

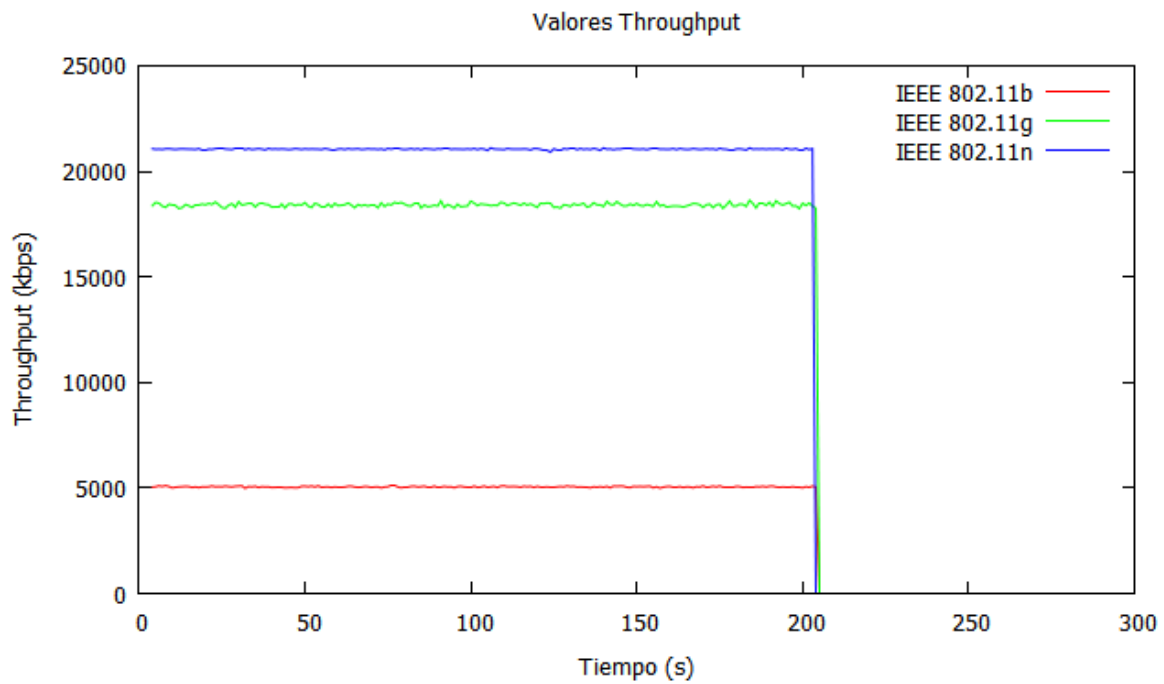


Figura 6.7. Throughput de los diferentes estándares IEEE 802.11

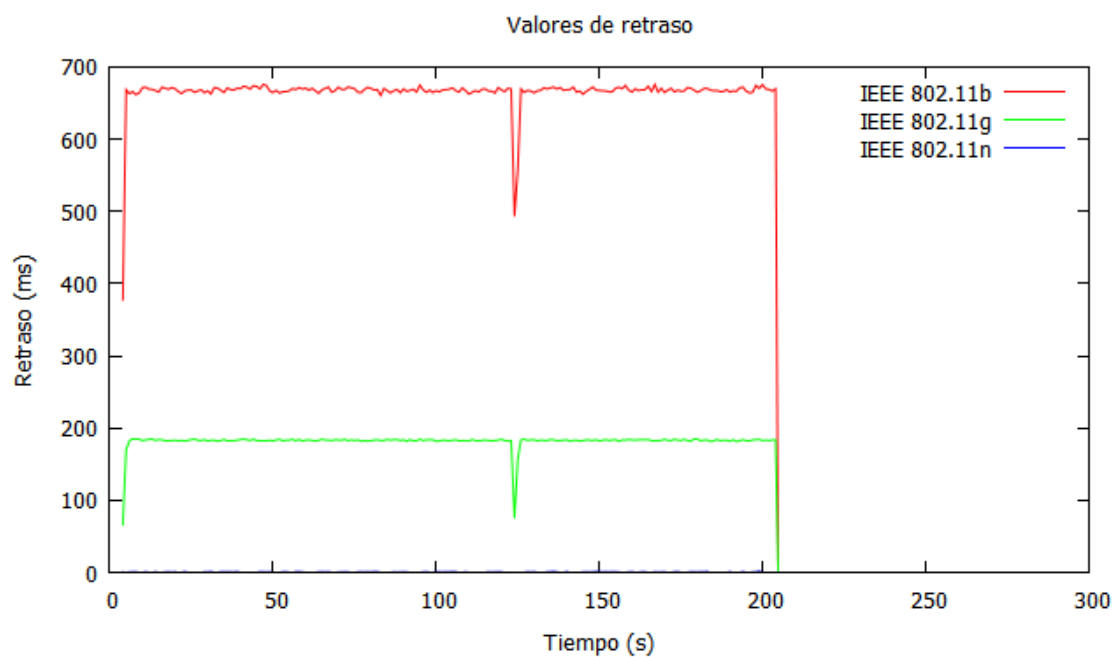


Figura 6.8. Retraso de los diferentes estándares IEEE 802.11

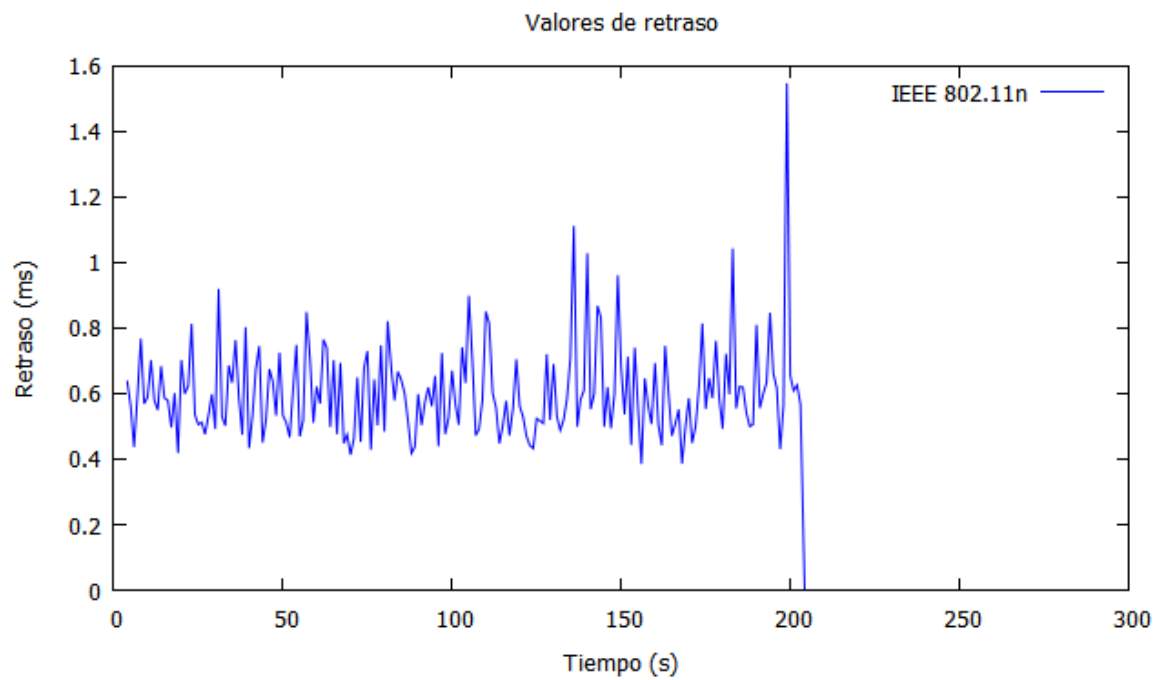


Figura 6.9. Retraso para el estándar IEEE802.11n

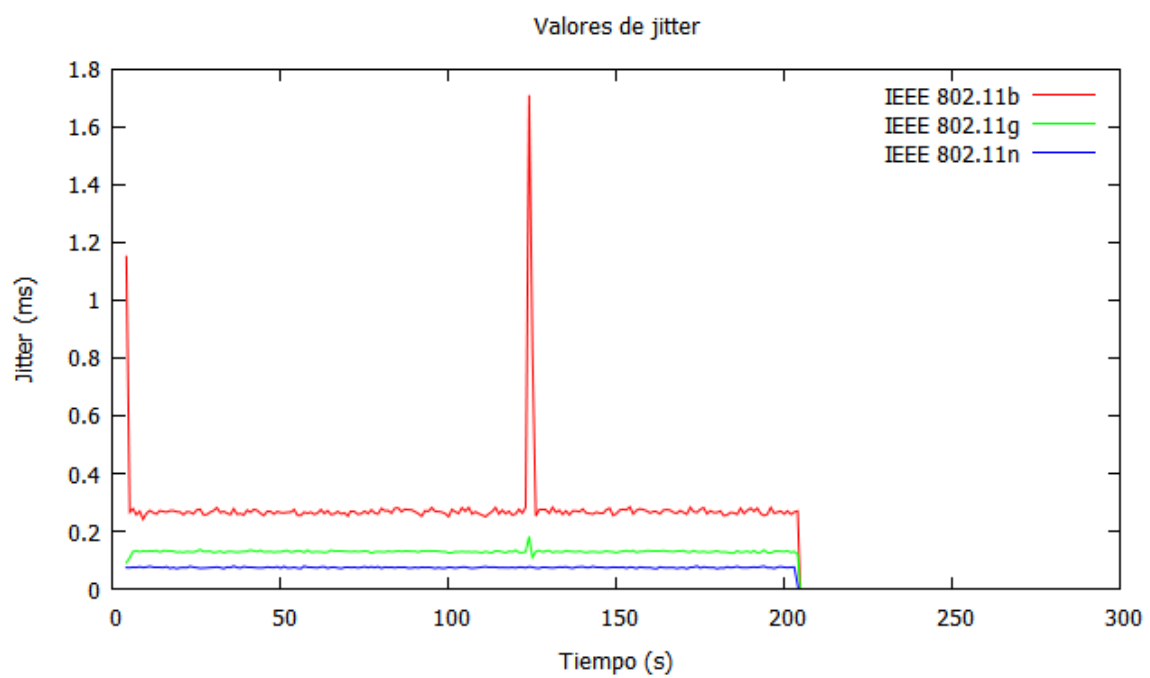


Figura 6.10. Jitter del escenario 2

6.4 Escenario 3

Se representa la comunicación en modo infraestructura entre un servidor y una cantidad variable de nodos clientes. Los nodos se encuentran repartidos en un área de 80mx80m y el servidor de VoIP se encuentra en el centro del área. El tráfico de datos es unidireccional en sentido servidor-cliente. El tráfico modelado para el escenario trata de simular, de una forma próxima a la real, llamadas de VoIP, codificado con el codec G711, entre los clientes y el servidor. El estudio se hace sobre uno de los nodos de la simulación analizando como sus parámetros de QoS se ven afectados por las diferentes casuísticas representadas. El escenario se representa en la figura 6.11.

Se incrementa el número de usuarios para estudiar cómo afecta dicha variación a los parámetros de QoS de los estándares IEEE 802.11g y IEEE 802.11n, ya que el IEEE 802.11b está en desuso y es claramente inferior en las condiciones del escenario.

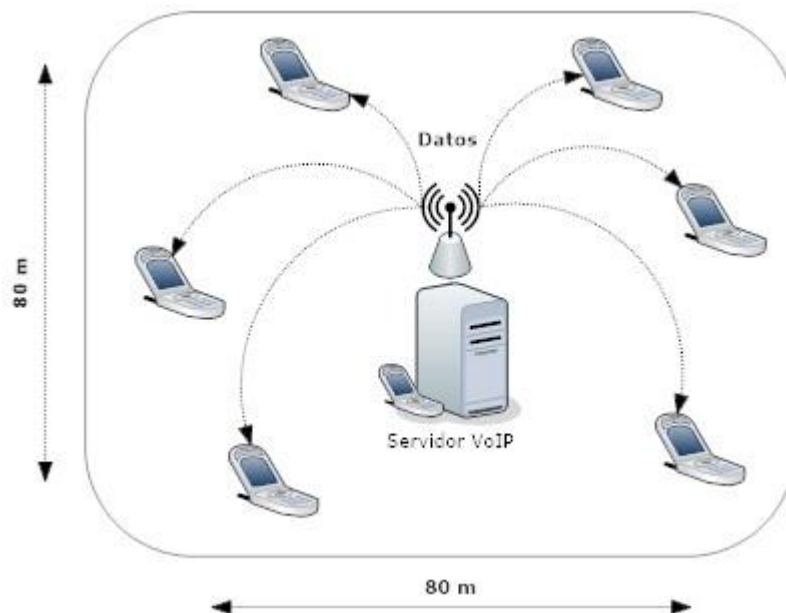


Figura 6.11. *Escenario 3*

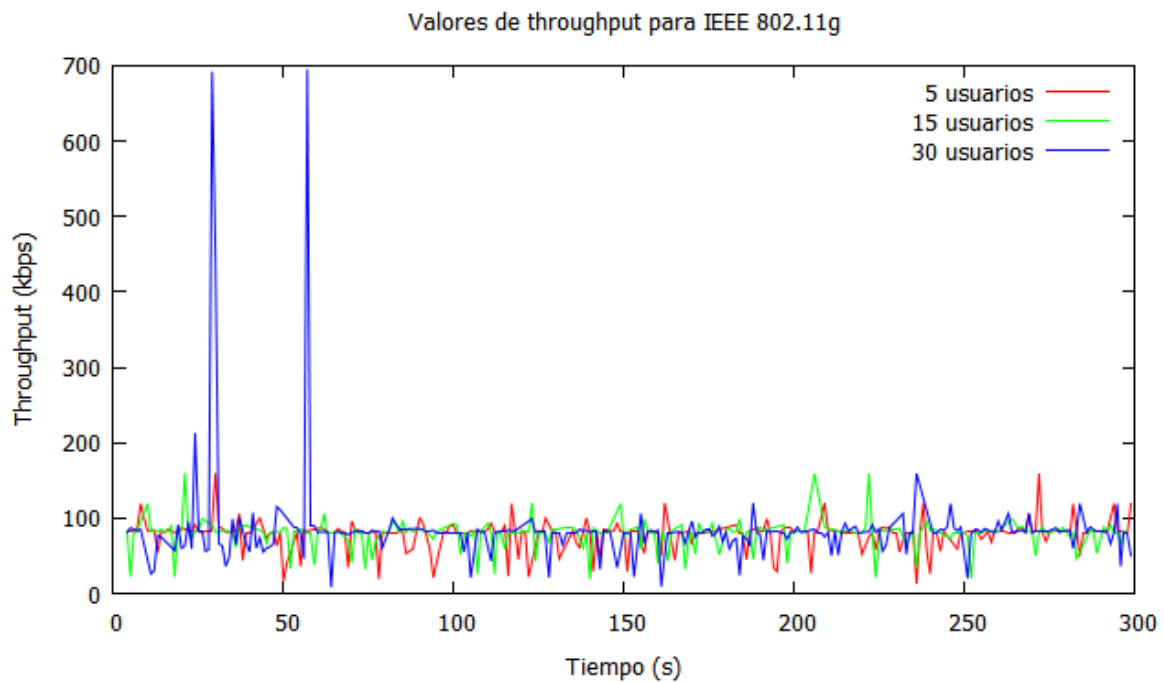


Figura 6.12. Throughput con variación de clientes para el estándar IEEE 802.11g

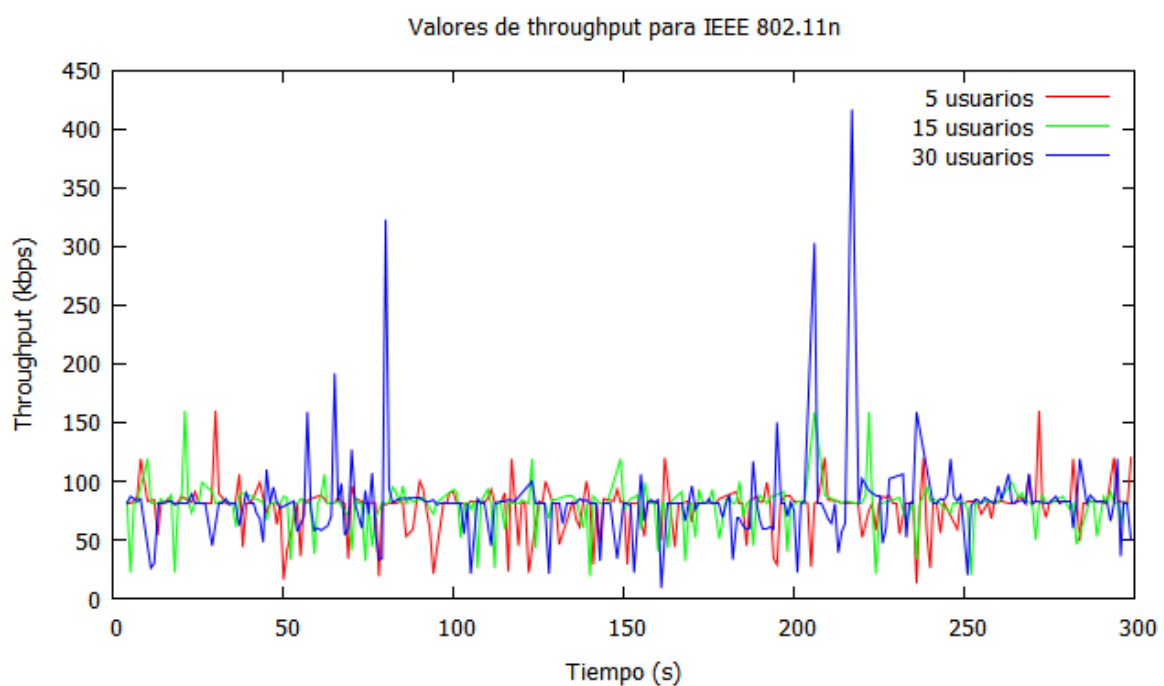


Figura 6.13. Throughput con variación de clientes para el estándar 802.11n

Analizando las gráficas de la Figura 6.14 se observa que para una cantidad de 5 y 15 usuarios el throughput se encuentra dentro de un rango aceptable, provocando unos picos pronunciados para 30 usuarios lo que indica que la cola del AP se ve saturada debido a la gran cantidad de peticiones de envío en esos instantes de tiempo.

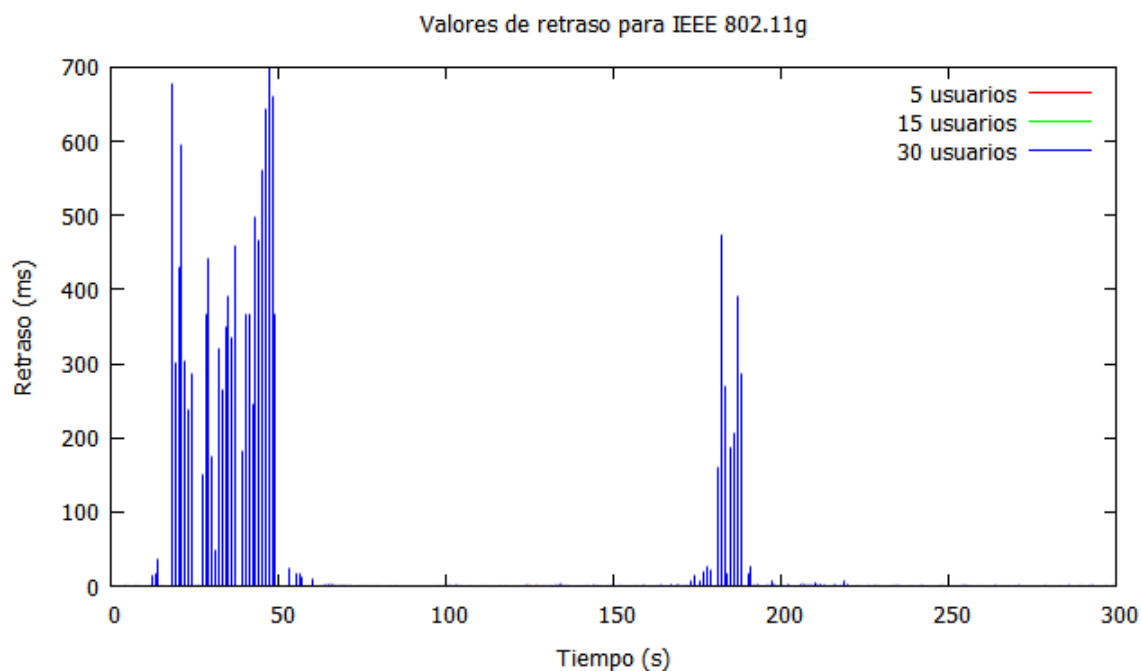


Figura 6.14. Retraso con variación de clientes para el estándar IEEE 802.11g

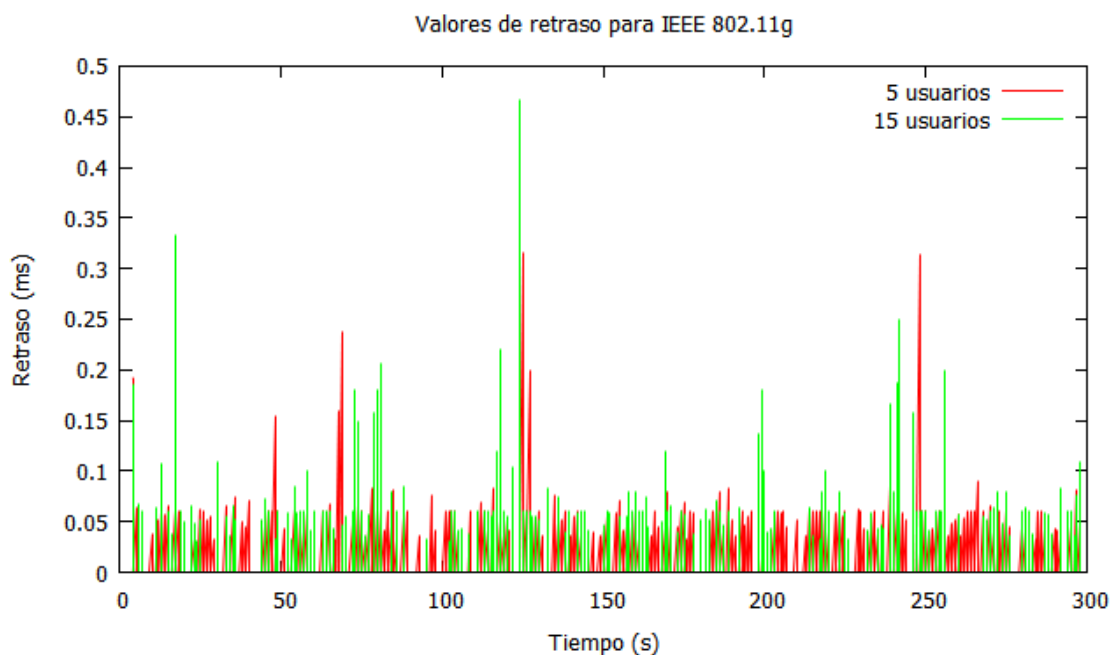


Figura 6.15. Ampliación de la gráfica de la Figura 6.10

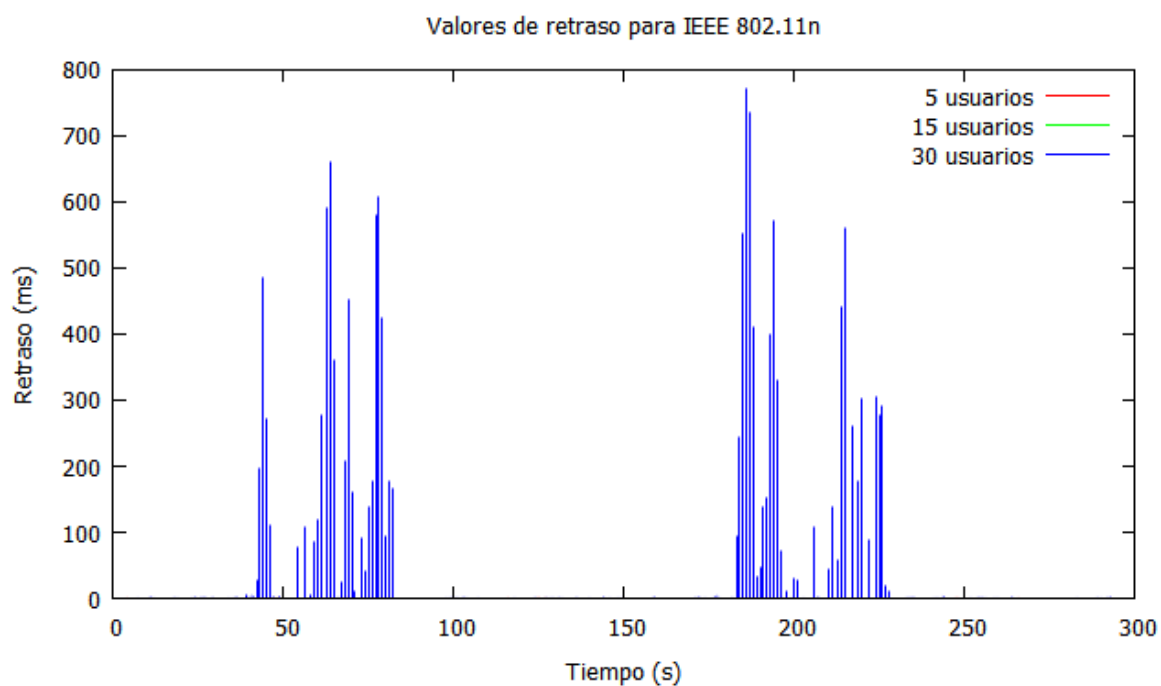


Figura 6.16. Retraso con variación de clientes para el estándar IEEE 802.11n

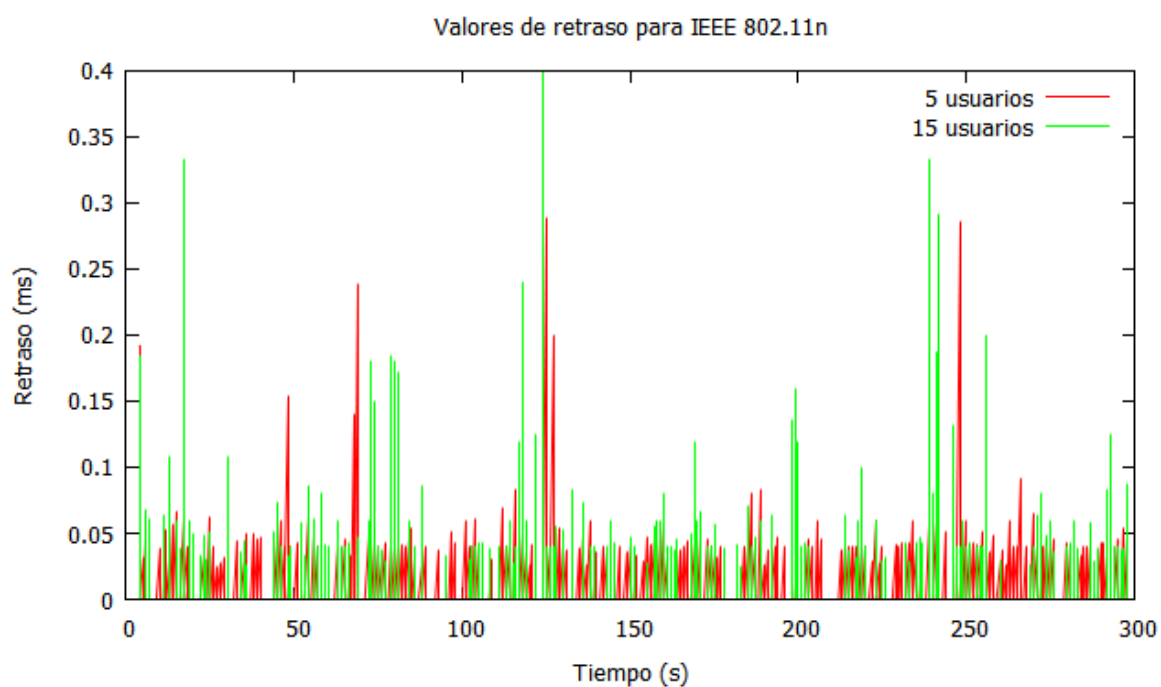


Figura 6.17. Ampliación de la gráfica de la Figura 6.16

En la gráfica de la Figura 6.16 se observa el retraso provocado por la saturación del canal de comunicación debido a la alta cantidad de clientes que requieren de servicio en momentos puntuales que coinciden con los picos de throughput.

En ambos estándares WiFi los datos de retraso para 30 usuarios son inaceptables para servicios críticos con parámetros de QoS, como es la VoIP.

En la grafica de la Figura 6.17 queda evidente que ambos estándares realizan una gestión adecuada para 5 y 15 usuarios. Basado en lo anterior y teniendo en cuenta que este tipo de tráfico no puede tener un retraso, en ningún caso, superior a 150ms se estima una cantidad máxima de 15 usuarios por punto de acceso para afrontar el uso de servicios de voz con garantías.

6.4 Escenario 4

Se representa la comunicación en modo infraestructura entre cuatro servidores emitiendo flujos de datos de diferentes tipos de tráfico y una cantidad fija de nodos clientes receptores. Los nodos se encuentran repartidos en un área de 80mx80m, en movimiento constante a velocidad de 0,5 mps, y los servidores emiten los datos a través de un AP situado en el centro del área. El tráfico de datos es unidireccional en sentido servidor-cliente. Se presentan los cuatro tipos de tráfico más comunes en el tipo de infraestructura objeto de estudio del presente PFC. La configuración de los nodos clientes, en movilidad, es de un nodo por tipo de tráfico: 1 terminal recibiendo tráfico VoIP, 1 terminal recibiendo tráfico de vídeo streaming, 1 terminales recibiendo tráfico HTTP y 1 terminal recibiendo tráfico FTP.

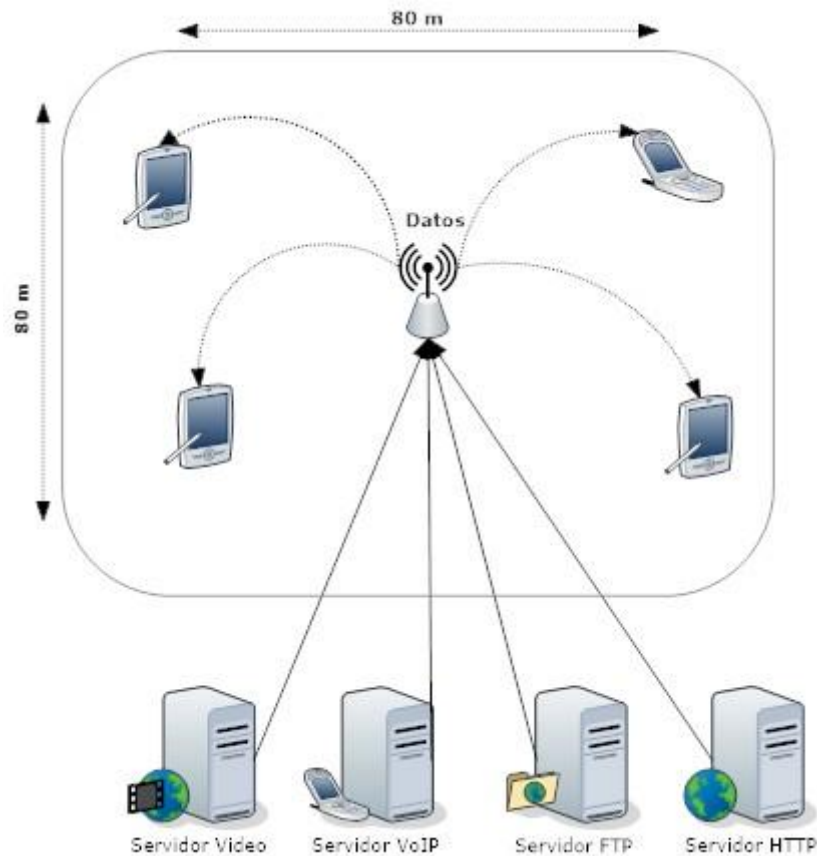


Figura 6.18. Escenario 4

Las aplicaciones de VoIP, HTTP, y vídeo Streaming comienzan a generar tráfico en el segundo 3 de la simulación manteniendo el flujo de datos, según el modelado de cada tipo de tráfico, hasta la finalización a los 300s. La aplicación de FTP envía un archivo de 30 MB en el segundo 200 de la simulación.

Se realiza una simulación del escenario con el estándar IEEE 802.11g y posteriormente con el IEEE 802.11n. El propósito es realizar un estudio comparativo entre los dos estándares y valorar las diferencias entre ambos prestando especial atención al tráfico multimedia.

6.4.1 Análisis del tráfico para el estándar IEEE 802.11g

En las gráficas de las Figura 6.19 y 6.20 se observa que las aplicaciones multimedia hacen un uso más intensivo del canal de comunicación. Las aplicaciones Web y de transferencia de archivos son menos intensas pero requieren de más recursos en los momentos que hacen uso del canal, debido al tamaño de los objetos que transfieren y a que usan protocolos orientados a conexión.

En la gráfica del retraso en la Figura 6.21 y en la gráfica del jitter en la Figura 6.22 se observa la problemática que provoca el uso intensivo del canal por la transferencia del archivo a los 200s de simulación, provocando un aumento considerable de los sus valores en todos los flujos de datos analizados. En el tráfico de los protocolos orientados a conexión HTTP y FTP no es problemático el aumento de dichos valores, sin embargo para servicios multimedia en aumento de los parámetros de retraso y jitter puede llegar a provocar problemas graves en la comunicación.

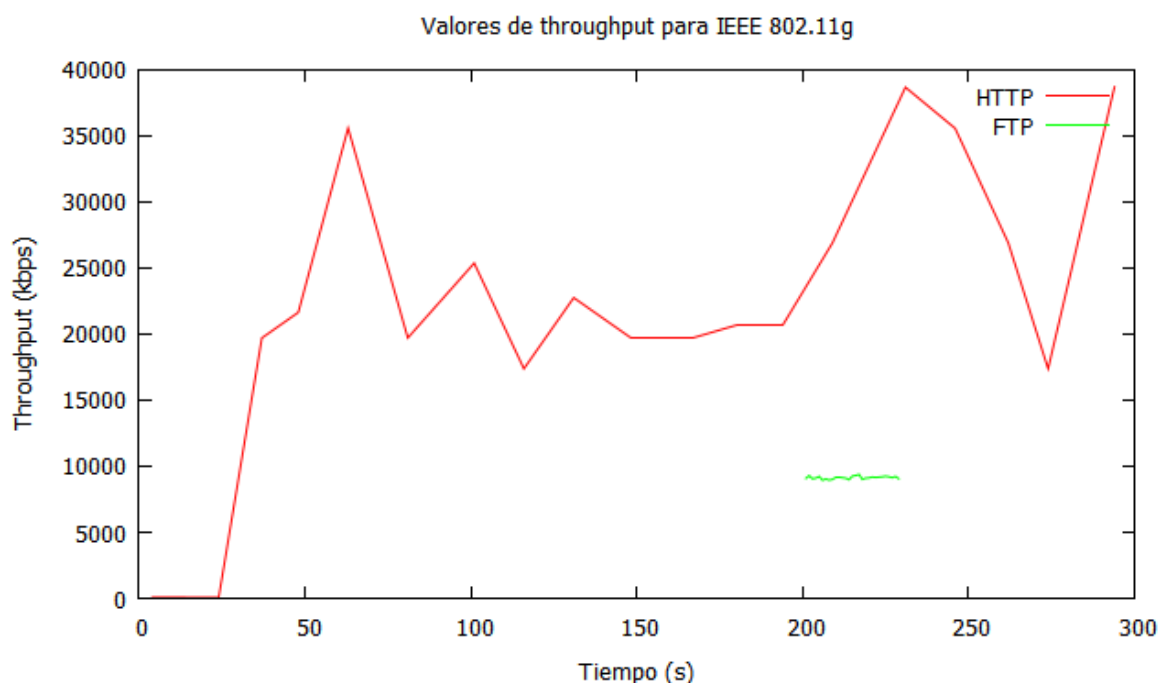


Figura 6.19. Gráficas del throughput del tráfico HTTP y FTP

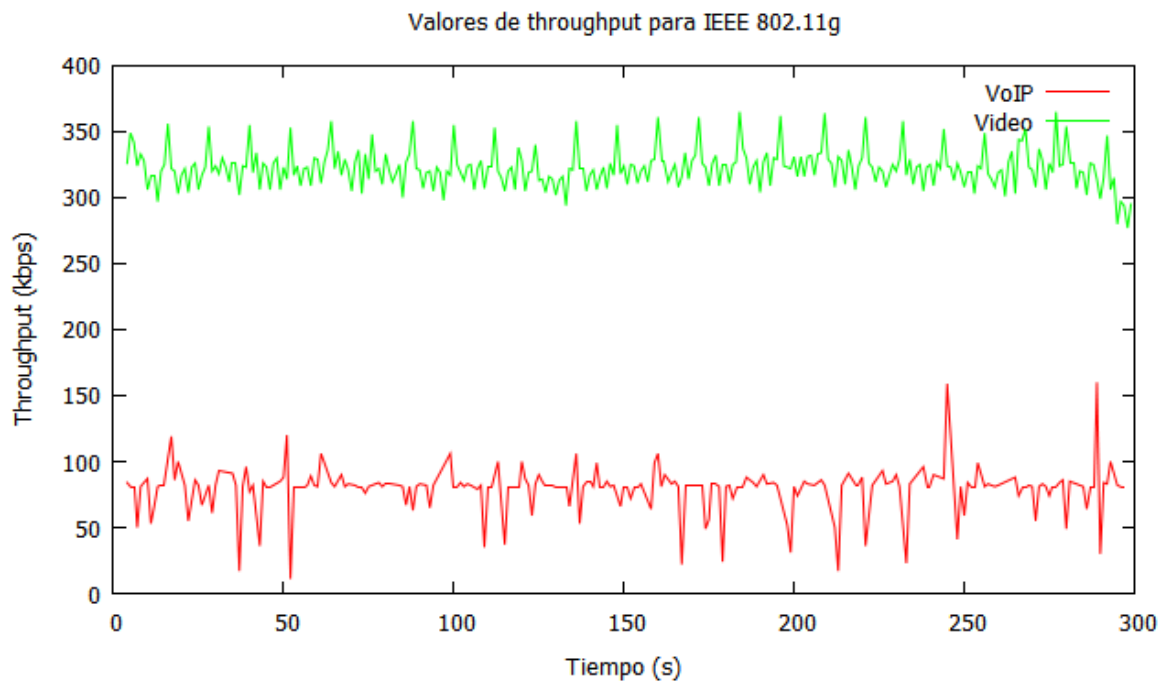


Figura 6.20. Gráficas throughput del tráfico de vídeo y la VoIP

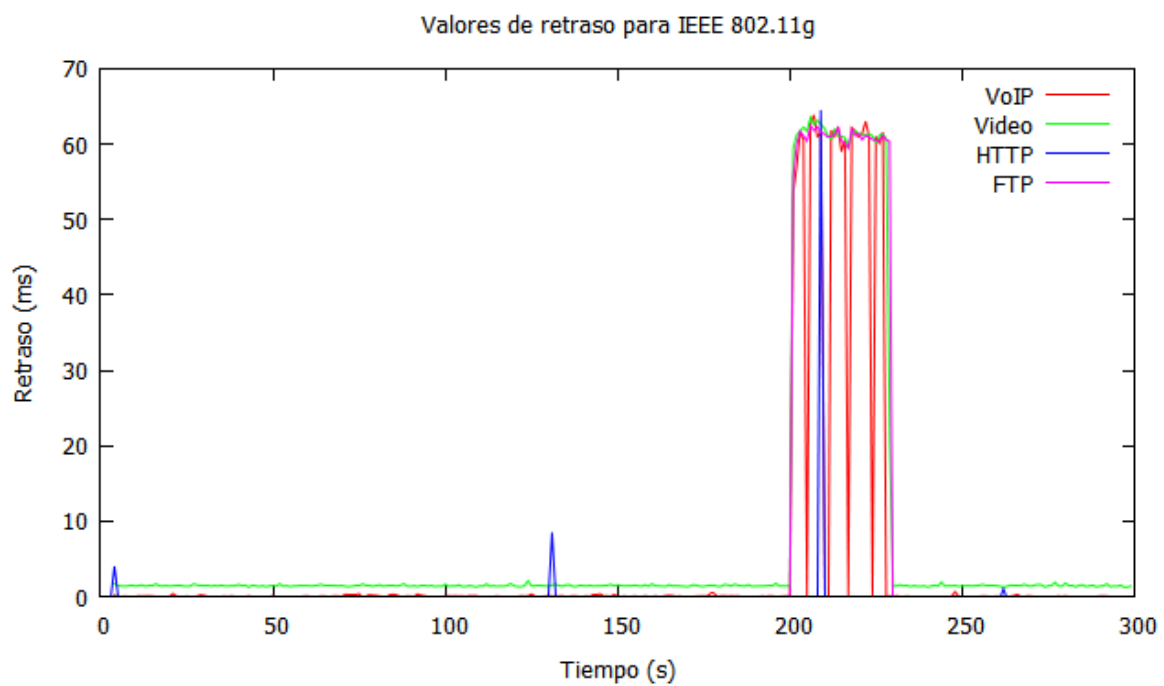


Figura 6.21. Gráfica del Retraso

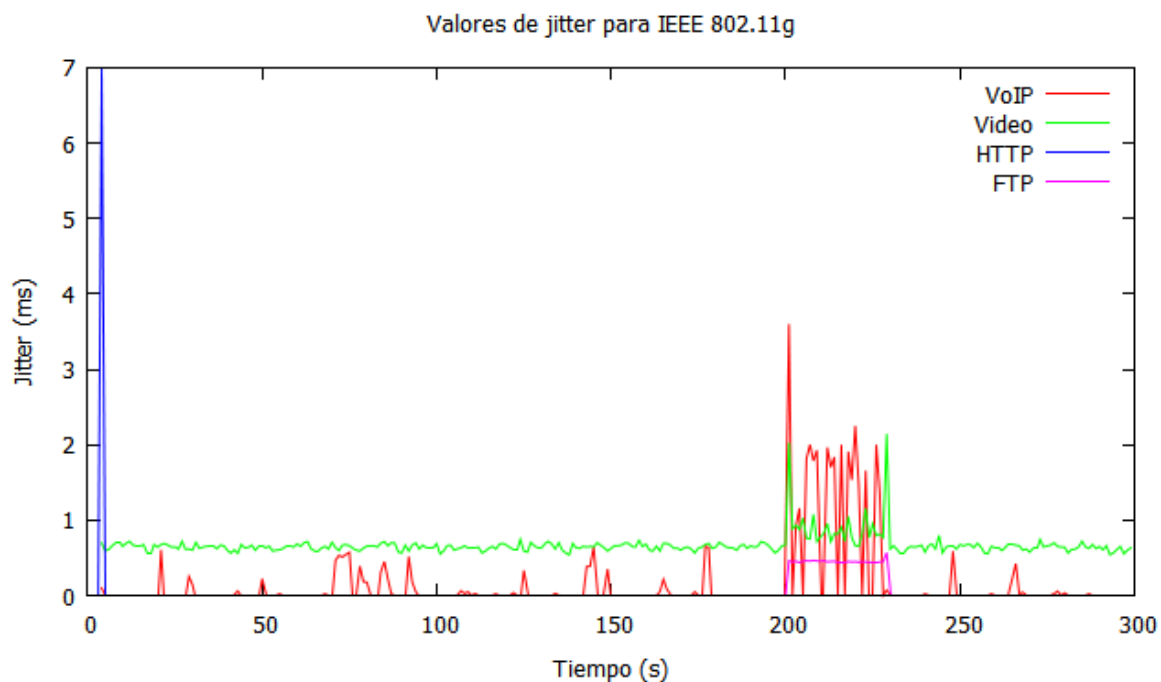


Figura 6.22. Gráficas del jitter

6.4.12 Análisis del tráfico para el estándar IEEE 802.11n

El throughput para el estándar IEEE 802.11n en la gráficas de las Figuras 6.23 y 6.24 es prácticamente el mismo que para el IEEE 802.11g, debido a que el canal no se llega a saturar, en cambio para los parámetros de retraso y de jitter, representados en las gráficas de las figuras Figura 6.25 y 6.26 se ve una clara diferencia del estándar IEEE 802.11n con respecto al estándar IEEE 802.11g, viéndose el tráfico de vídeo y VoIP mínimamente afectado, con el único retraso intrínseco del propio canal de comunicación. Por lo que para el estándar IEEE 802.11n se garantiza el correcto funcionamiento de las aplicaciones multimedia aunque haya otras aplicaciones haciendo un uso abusivo del canal que provocan las aplicaciones de transferencia de archivos, como es el caso del FTP.

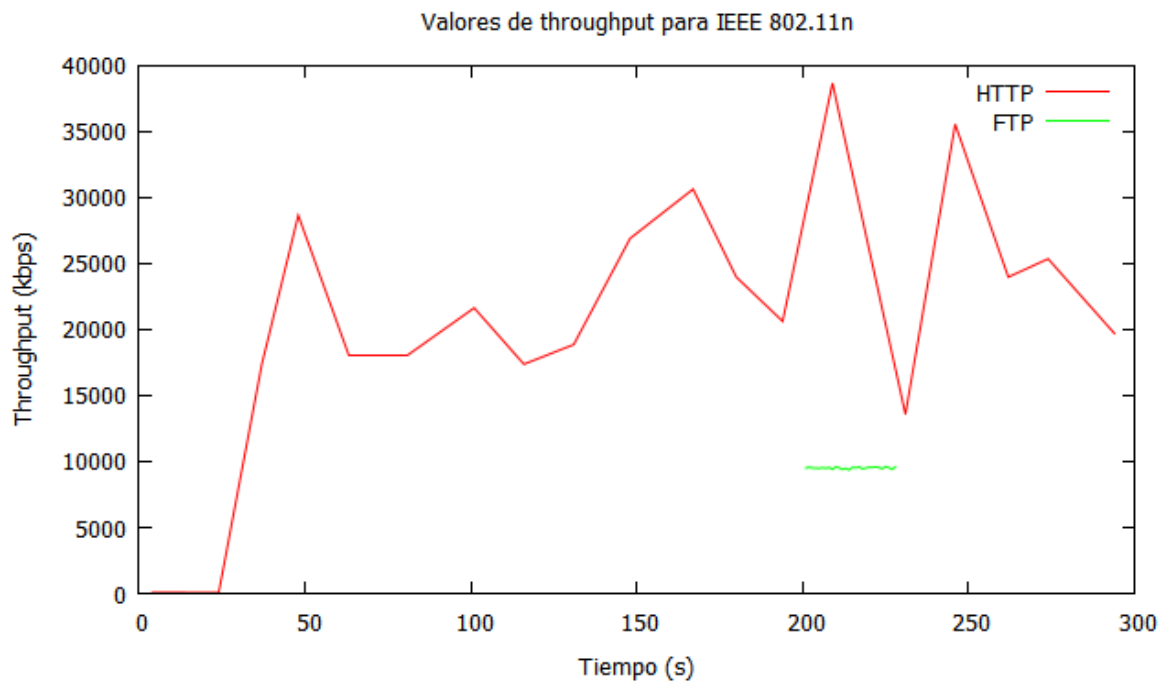


Figura 6.23. Gráficas del throughput del tráfico HTTP y FTP

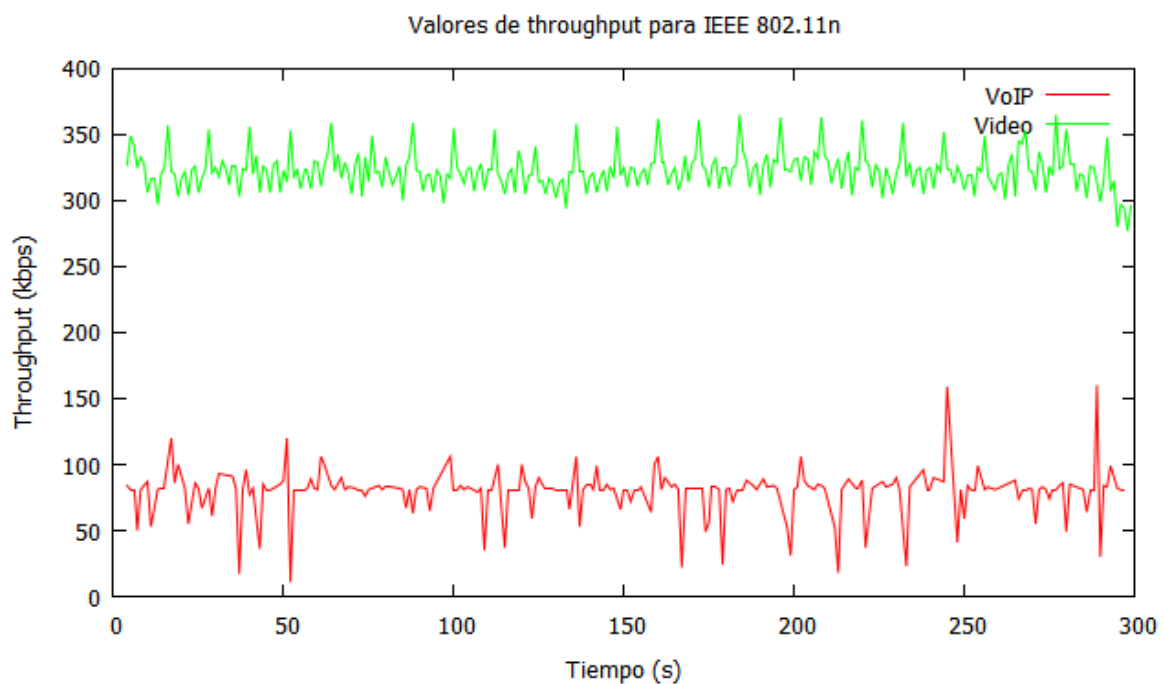


Figura 6.24. Gráficas throughput del tráfico de vídeo y la VoIP

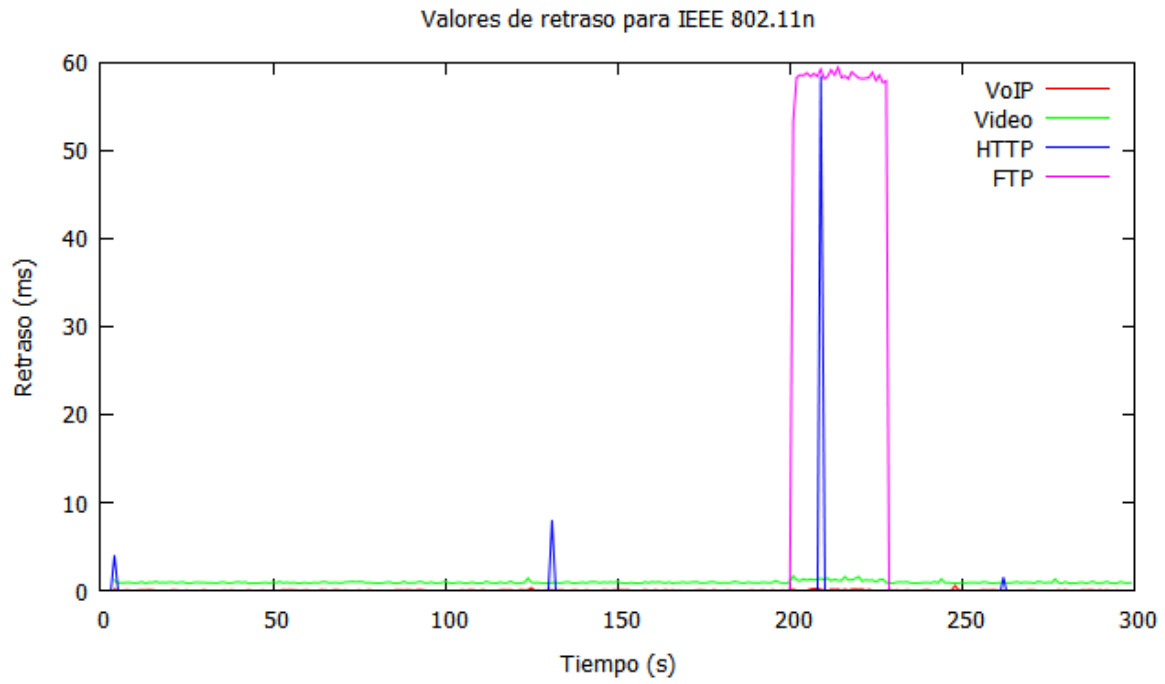


Figura 6.25 Gráfica del retraso

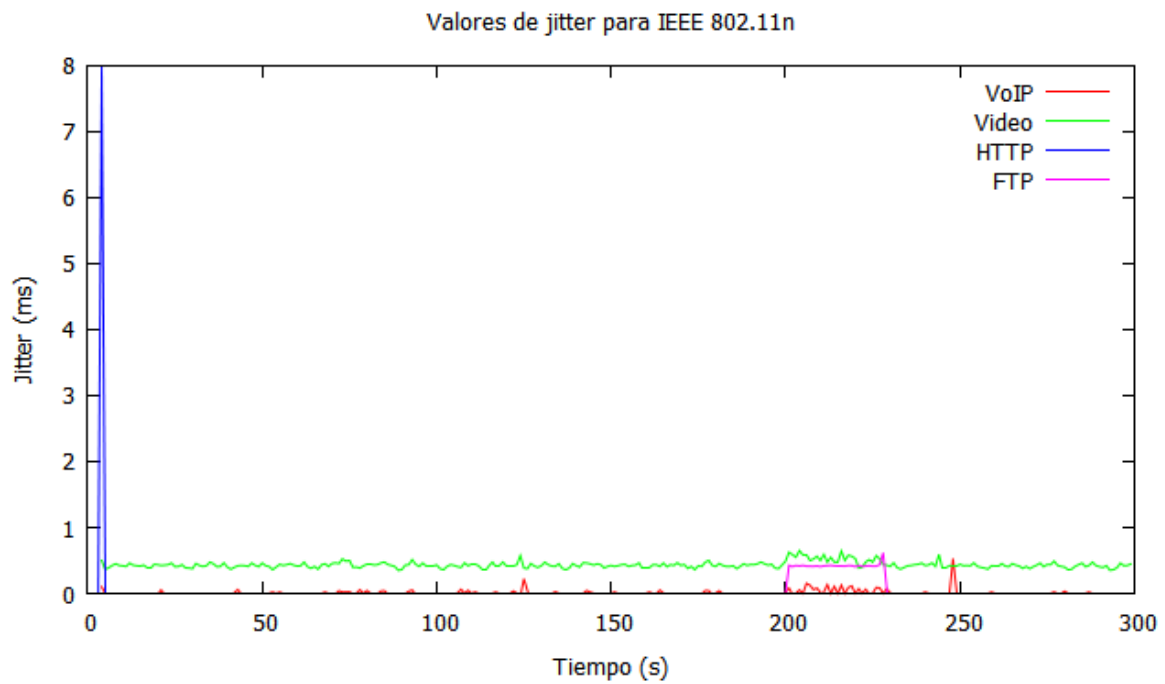


Figura 6.26. Gráfica del jitter

6.5 Escenario final

Se representa un escenario similar al escenario 4 del presente capítulo con la diferencia de que la cantidad de usuarios que harán uso del AP será aumentado considerablemente y el estándar que se estudia es el IEEE 802.11n debido a su mejor desempeño en las simulaciones anteriores. La configuración de los nodos clientes es la siguiente:

- 5 Realizando llamadas de voz.
- 1 Realizando vídeo Streaming.
- 10 Realizando navegación web.
- 1 Realizando transferencia de archivos.
- 3 Conectados sin provocar carga de tráfico de aplicaciones.

Por lo que hay un total de 20 nodos conectados al AP, la cantidad es sensiblemente superior a la media esperada para la infraestructura objeto de estudio del presente PFC para garantizar su correcto desempeño.

En la Figura 6.27 se visualiza la simulación en tiempo real con *Pyviz*.

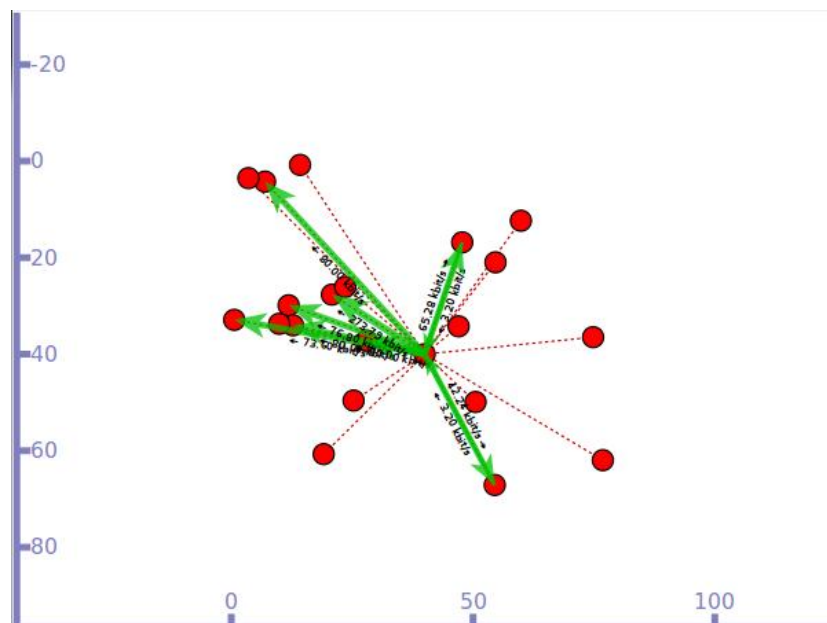


Figura 6.27. Animación del escenario final con *Pyviz*

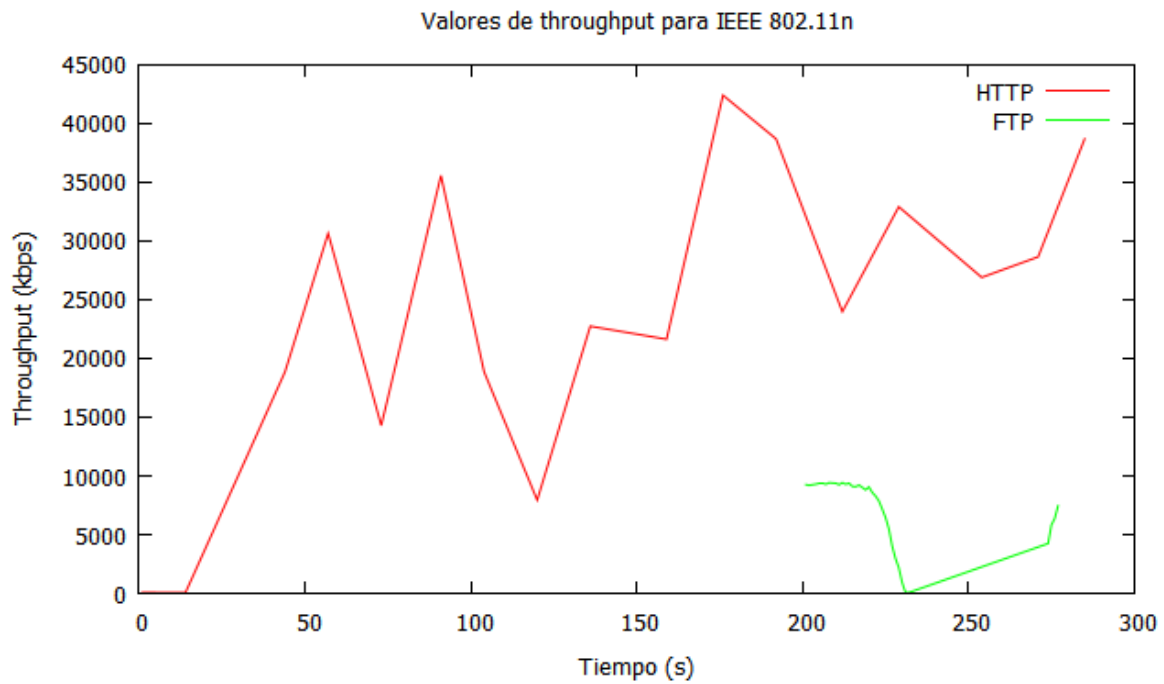


Figura 6.28 Throughput del escenario final

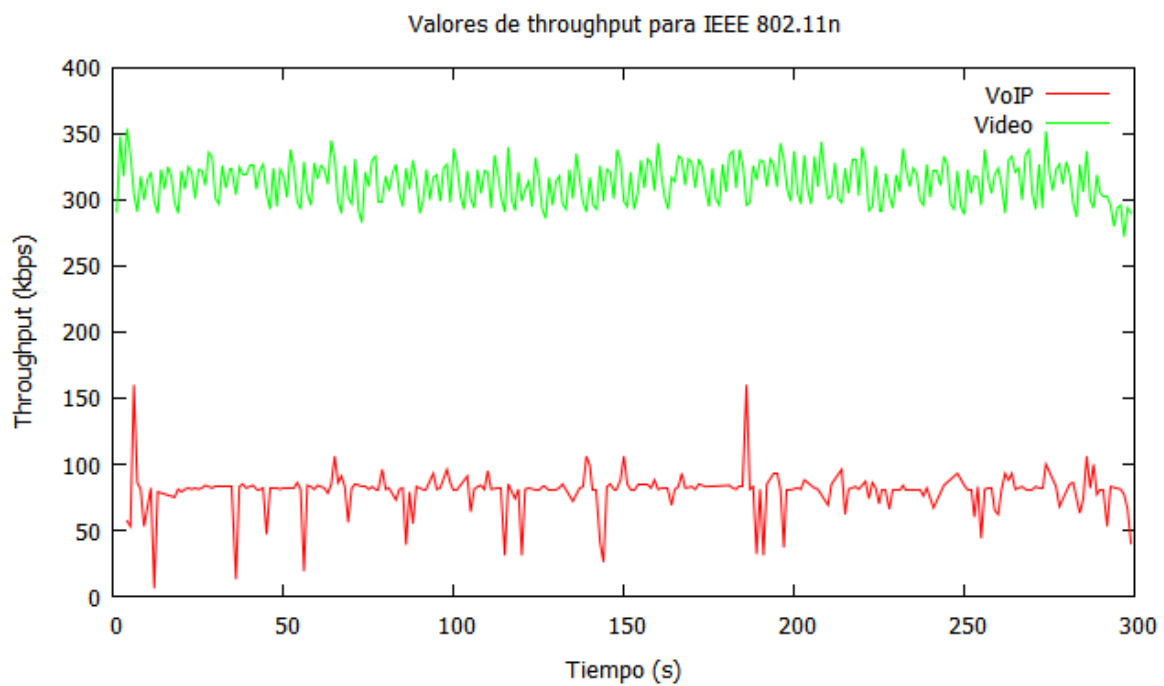


Figura 6.29. Ampliación de la gráfica 6.28

En las gráficas de las Figuras 6.28 y 6.29 se muestra cómo el tráfico de voz y el de vídeo no se ven afectados de forma importante en el escenario simulado, no se puede

decir lo mismo del tráfico FTP, representado en la Figura 6.28, ya que se observa una caída del throughput en el entorno a los 230s de simulación.

La caída del tráfico FTP es debida a la saturación del canal que se ve desbordado provocando pérdidas de paquetes, representadas en la gráfica de la Figura 6.18 del tráfico. El tráfico de vídeo también experimenta pérdidas mínimas de paquetes debidas a tal situación, dichas pérdidas se consideran aceptables ya que no se ve reflejada en su throughput representado en la gráfica de la Figura 6.29.

En las gráficas del retraso de la Figura 6.31 y del jitter en la Figura 6.32 se evidencia las mejoras implementadas en la capa MAC del IEEE 802.11n orientadas a un mejor desempeño de los servicios multimedia, obteniéndose unos valores de dichos parámetros muy aceptables teniendo en cuenta la cantidad de servicios que hacen uso del canal.

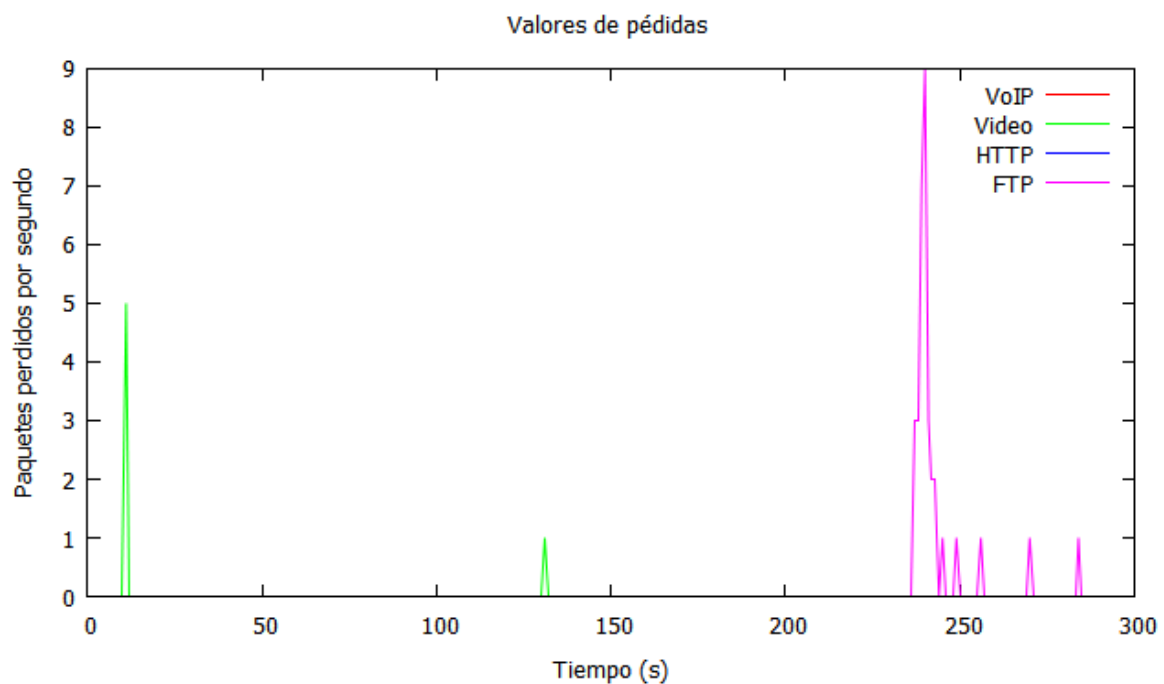


Figura 6.30. Pérdidas del escenario final

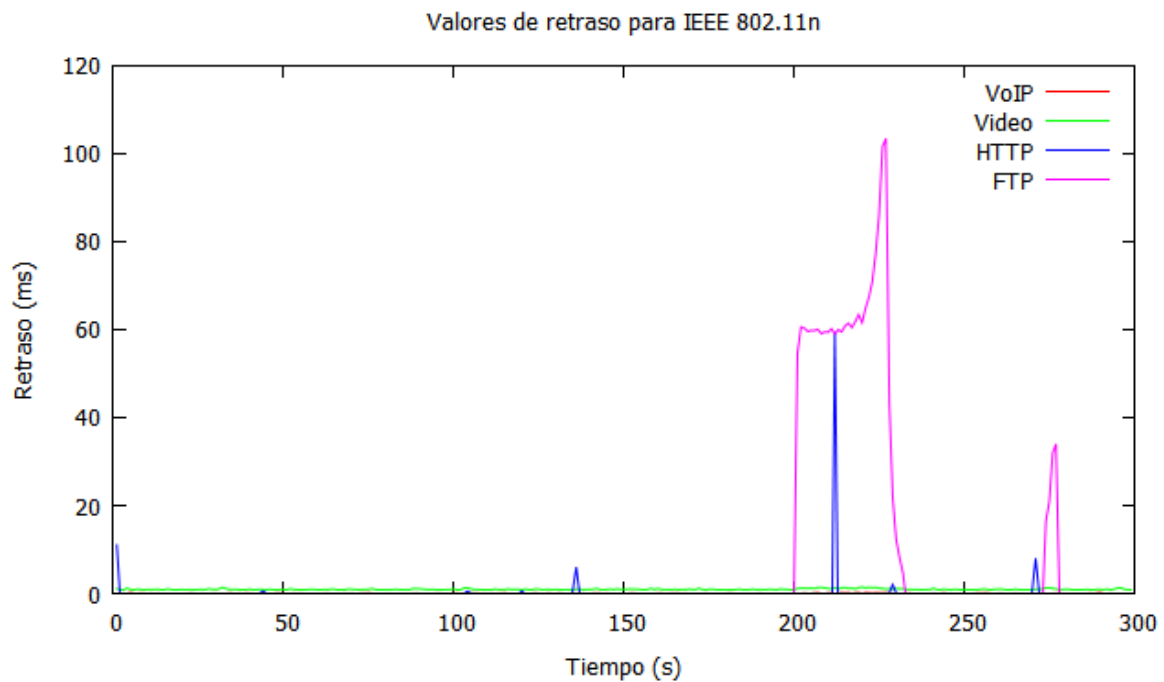


Figura 6.31. Retraso del escenario final

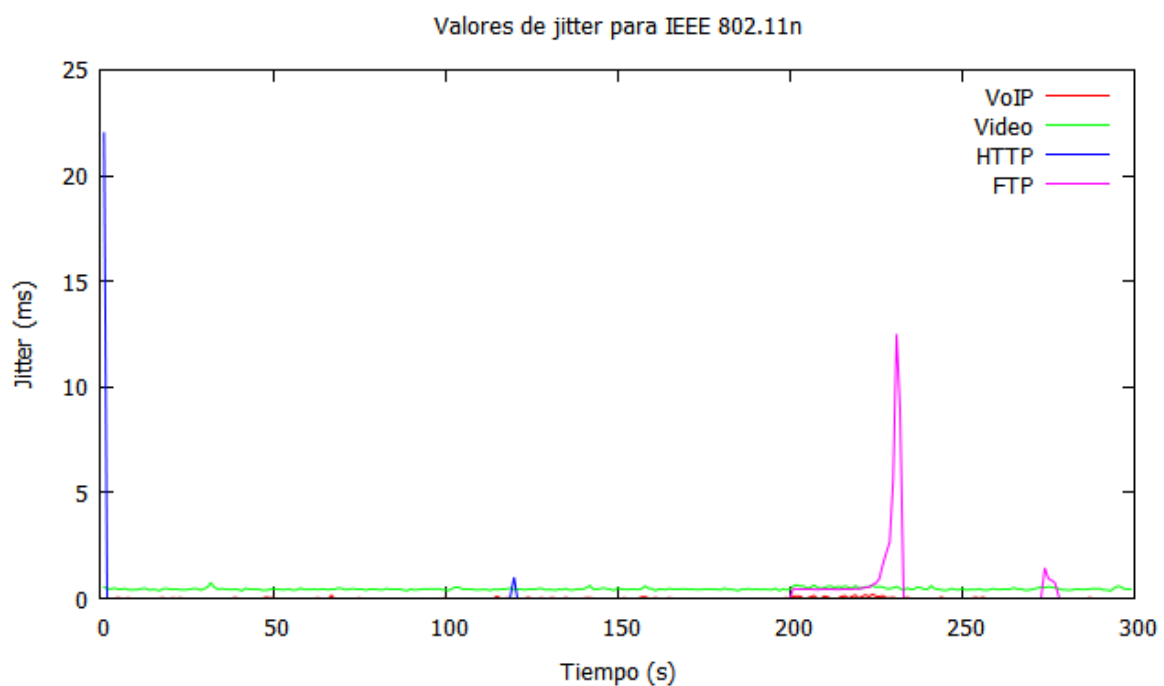


Figura 6.32. Jitter del escenario final

A modo de conclusión, queda totalmente demostrada la superioridad del estándar IEEE 802.11n respecto los anteriores, principalmente en lo referente a su desempeño con

las aplicaciones multimedia garantizando el correcto funcionamiento de las mismas siempre que no haya más de 20 usuarios conectados por punto de acceso.

En el presente capítulo también ha quedado claro el efecto que las aplicaciones de transferencia de archivos tiene sobre las redes de comunicaciones, saturando el canal y empeorando las el buen funcionamiento de las mismas, por lo que se recomienda implementar algún mecanismo que evite el uso abusivo de dichas aplicaciones para garantizar la correcta QoE de los usuarios de la infraestructura diseñada en el presente PFC.

7. Conclusiones y posibles ampliaciones

En este capítulo se presentan las conclusiones obtenidas mediante la realización de este proyecto y se exponen algunas ideas sobre futuras actualizaciones de la red diseñada.

7.1 Conclusiones

El desarrollo de una correcta planificación de una infraestructura inalámbrica no es trivial debido a que se tiene que tener en cuenta una serie de peculiaridades, principalmente por el medio de propagación que usan dichas redes, para garantizar una correcta QoE. Este factor es determinante en infraestructuras destinadas a comercializar servicios y unificarlos como es el caso del presente PFC.

Durante el desarrollo de este PFC se han realizado una serie de simulaciones con el objetivo de realizar un diseño de la infraestructura inalámbrica con garantías, tanto de coberturas como de QoS de cara a las aplicaciones que van a hacer uso de la misma.

Se ha profundizado en el estudio las simulaciones aplicadas a ámbito telemático mediante ns3, del que no se tenían nociones previas, utilizado para el diseño de los distintos nodos y canales que conforman una infraestructura de comunicaciones.

Se han realizado una simulación de diferentes escenarios en los que se ha tratado de modelar situaciones, previsibles en el complejo hotelero, para los terminales móviles y así poder calcular como varía su recepción de señales radioeléctricas en la banda de 2,4 Ghz, haciendo especial hincapié en realizar el diseño de la infraestructura lo más ajustado como sea posible a las peculiaridades del complejo hotelero sin la necesidad de desplazarse hasta Fuerteventura.

Se han estudiado también las características de las transmisiones de vídeo y voz a través de las redes de comunicaciones analizando su funcionamiento y los diferentes aspectos que les afectan.

Se han desarrollado una serie de aplicaciones para generar los diferentes cargas de tráfico sobre los AP, los cuales ha sido modelados mediante modelos matemáticos basados en el estándar WiFi, para analizar el comportamiento de éstos ante diferentes tipos de tráfico generado por aplicaciones, basadas en las necesidades actuales del

complejo hotelero, y cantidad de terminales conectados requiriendo conexión inalámbrica. Concluyendo, por AP, las limitaciones y el desempeño marcados por los diferentes estándares WiFi analizados.

Gracias a la profundidad que se ha alcanzado en este proyecto, se ha podido diseñar una red inalámbrica que asegure un nivel óptimo de rendimiento en cuanto a aplicaciones interactivas se refiere. Asegurando una cobertura para los clientes de la infraestructura así como una QoS para sus aplicaciones, más que aceptable mediante simulación.

7.1 Posibles ampliaciones

Existen algunas mejoras que podrían realizarse en futuras intervenciones en la infraestructura diseñada.

En futuras ampliaciones se podría realizar el diseño paralelo de una red que cumpla con el estándar IEEE 802.11s y que tenga dos propósitos principales; el primero sería el de ofrecer un servicio con QoS diferenciado para los clientes fidelizados del complejo, los cuales tendrían preferencia de tráfico en sus aplicaciones sobre el resto de clientes y la segunda como una red de respaldo mediante conexión de datos por redes móviles o a través de un enlace punto a punto con algún complejo cercano garantizando siempre el servicio de datos.

Teniendo en cuenta que la seguridad de las WLAN es un aspecto importante y que, debido a la naturaleza del medio que se usa para la comunicación, suelen tener importantes carencias en dicho aspecto una posible línea de trabajo futura sería la de estudiar el impacto que tendrían diferentes mecanismos que aseguren la confidencialidad de los datos, así como su integridad y autenticidad.

Pliego de condiciones y presupuesto

En este apartado se presentan el pliego de condiciones en el que se detallan los requisitos necesarios para poder ejecutar la infraestructura diseñada destinada a disponer de conexión inalámbrica las zonas indicadas del complejo hotelero, así como una breve descripción del coste de realización del proyecto.

Pr.1 Pliego de condiciones

En este apartado se indican los requisitos necesarios para poder ejecutar la infraestructura de la red diseñada en este proyecto.

Pr.1.1 Descripción de la infraestructura

La infraestructura está compuesta por 13 puntos de acceso con sus respectivas antenas, distribuidos a lo largo del complejo hotelero como se indica en los planos adjuntos, a su vez estos puntos de acceso, están conectados a diferentes dispositivos como switches, patch pannels, encaminadores, alojados en un armario rack dispuesto para tal fin en el *Recinto de Instalaciones de Telecomunicaciones (RITU)* localizado la zona de administración.

La cobertura inalámbrica teórica presentada en los mapas de cobertura, en la práctica se ve inevitablemente modificada debido a la naturaleza de las señales radioeléctricas, no pudiéndose tener un cálculo exacto de todos los factores que afectan a las mimas, dichos factores solo son identificables mediante una medición in situ del complejo hotelero una vez desplegada la infraestructura.

Pr.1.2 Requisitos hardware y software

Los PA para llevar a cabo la infraestructura deben ser compatibles con el estándar IEEE 802.11 en todas las versiones que trabajan en la banda de 2,4Ghz para garantizar la compatibilidad con la mayoría de dispositivos disponibles en el mercado y estarán dotados de antenas con ganancia de 9dBi.

Para la interconexión con la red de fibra óptica se requieren convertidores de medios Ethernet-fibra óptica, y viceversa, los cuales conectan con los puntos de acceso de la red WiFi mediante cable UTP categoría 5e. La alimentación de los puntos de acceso se hará mediante PoE que nos proporcionarán los propios conversores de medios.

La infraestructura se configura en modo heterogéneo para que sea accesible con tarjetas de red de los estándares IEEE 802.11b, IEEE 802.11g y IEEE 802.11n, aunque es evidente de que la diferencia de velocidades y de mecanismos de QoS en los diferentes estándares influyen en la QoS de las aplicaciones multimedia.

El servicio propuesto para dar soporte a la WLAN es totalmente compatible con cualquier sistema de gestión de acceso a Internet que la propiedad decida implementar.

La propiedad debe considerar el ancho de banda contratado dependiendo de la QoS adecuada a los estándares que se quiera ofrecer a los clientes del complejo, proponiéndose en este proyecto unos mínimos deseables.

Pr.2 Presupuesto

El propósito de este apartado es presentar el presupuesto de este PFC, formado por el presupuesto de la mano de obra y el presupuesto del material utilizado.

Para calcular el coste de la mano de obra, hemos subdividido la elaboración del proyecto en distintas fases de desarrollo:

- Información.
- Definición.
- Análisis.
- Desarrollo de las simulaciones teóricas.
- Diseño.

- Análisis de la red.
- Escritura de la memoria.

El coste de cada fase se calcula con el número de horas empleadas en el mismo, utilizando los baremos del COITT.

Pr.2.1 Descripción de las fases del proyecto

En este apartado queda reflejada cada una de las etapas en que se divide la realización de este proyecto, definiendo en cada caso el tipo de trabajo realizado.

- *Información*: esta primera fase está orientada a la obtención de la información necesaria con el objetivo de alcanzar los conocimientos necesarios para el desarrollo del proyecto. Destacando en estudio de los diferentes programas de análisis y simulación de redes inalámbricas, el estudio de aplicaciones en este tipo de redes, así como funcionamiento y las posibilidades del mercado para este tipo de tecnologías.
- *Definición*: en esta fase se fijan los objetivos del proyecto partiendo de la situación que da origen al mismo, así como determinando que tipos de tecnologías requiere y el vínculo que existe entre ellas.
- *Análisis*: esta fase está dedicada a definir los módulos necesarios para llevar a cabo las simulaciones, analizando funcionalidades y definiendo la estructura de la codificación de la misma.
- *Desarrollo de las simulaciones teóricas*: esta fase está dirigida hacia el desarrollo y codificación de los diferentes algoritmos, implicados en las diversas simulaciones del presente PFC con ns3, definidos en la fase anterior. Una vez definidos, se

inician las simulaciones y comienza la labor de comprobación y depuración de todo este código.

- *Diseño*: en esta fase se realizan una serie de simulaciones teóricas con el objetivo de diseñar una infraestructura garantizando la cobertura inalámbrica, mediante puntos de acceso que cumplan los estándares WiFi, a todo el complejo hotelero que proporcione conexión a Internet en todo el recinto. De los datos obtenidos se deduce el plano de coberturas presentado en este proyecto
- *Análisis de la red*: esta fase se realizan una serie de simulaciones teóricas con la finalidad de analizar los niveles de rendimiento de las aplicaciones multimedia, así como las gráficas de carga de tráfico, sobre la red WiFi definiendo los límites de infraestructura o su desempeño.
- *Escritura de la memoria*: en esta fase se desarrolla la presente memoria, analizando todos los pasos llevados a cabo en cada una de las fases anteriores, así como el desarrollo de los conceptos para entenderla. Además se desarrollará el presupuesto del coste que supone desarrollar el mencionado proyecto.

Pr.2.2 Mano de obra

Seguidamente exponemos el presupuesto de la mano de obra empleada en cada fase. El número de horas trabajadas en cada fase se indica en la columna de la izquierda. El tiempo equivalente en meses, con una media de 5 horas al día, 5 días a la semana, 4 semanas al mes, es de 6 meses.

La tabla Pr.1 muestra las horas invertidas en cada una de las fases de realización del presente proyecto.

Fase	Horas
Información	110
Definición	90
Diseño de la infraestructura	110
Simulación de la infraestructura	160
Análisis	120
Escritura de la memoria	190
Total	780

Tabla Pr.1 Horas invertidas en cada etapa del proyecto.

Los honorarios debidos a esta mano de obra se calculan siguiendo la fórmula del COITT establecida para calcular los honorarios en trabajos tarifados por horas empleadas para el año 2003, cuyo significado es el siguiente:

$$H = 78 \cdot H_e + 65 \cdot H_n.$$

$$H_e = \text{Horas especiales} \cdot C.$$

$$H_n = \text{Horas normales} \cdot C.$$

$$C = \text{Coeficiente reductor}.$$

Utilizando un coeficiente reductor de 0,4, los honorarios por mano de obra son los siguientes:

$$H = 65 \cdot 600 \cdot 0,4 = 15.600 \text{ euros}.$$

Pr.2.3 Coste del material

En este apartado se explican los costes del material utilizado para la realización de este PFC.

Concepto	Importe (euros)	Tiempo de uso (meses)	Amortización
Un ordenador portátil	900	7,5	1.200
Impresora <i>chorro de tinta</i>	180	7,5	180
SO Windows 8	120	7,5	120
Fungibles	90	7,5	90
Total coste material			1.590

Tabla Pr.2 Coste del material empleado

La tabla Pr.2 muestra un resumen del coste del material empleado para la realización de este proyecto.

Todo el coste del material tiene como plazo de amortización los 6 meses de elaboración del proyecto.

Pr.2.4 Coste total del proyecto

En la tabla Pr.3 se muestra el coste total del proyecto.

Concepto	Coste (euros)
Mano de obra	19.500
Coste de material	1.590
Total	21.090

Tabla Pr.3 Coste total del proyecto.

El presupuesto total de este proyecto es de VEINTIUNMIL NOVENTA EUROS.

El proyectando:

Fdo: Juan Francisco Ramírez Castellano

Bibliografía y referencias

En este apartado se presenta la relación de documentos bibliográficos y referencias que se utilizaron en la fase de documentación de este PFC.

Bibliografía y referencias

- [1] Simulación de eventos discretos
<http://italica.us.es/asignaturas/Simulacion>

- [2] Documentación ns3
<http://www.nsnam.org>

- [3] Yet Another Network Simulator
<http://cutebugs.net/files/wns2-yans.pdf>.

- [4] VirtualBox
<https://www.virtualbox.org/wiki/Downloads>.

- [5] Unión Internacional de Telecomunicaciones
<http://www.itu.int/>.

- [6] La situación de las Tecnologías WLAN basadas en el estándar IEEE 802.11 y sus variantes ("WiFi"). Colegio Oficial de Ingenieros de Telecomunicación. ISBN: 84-931582-8-3.

- [7] 802.11 (WiFi) Neil Reid y Ron Seide. ISBN:0-07-222623-4.

- [8] WiFi Cómo construir una red inalámbrica. José A. Carballar. ISBN: 84-7897-630-2.

- [9] Redes Wireless 802.11. Matthew S Gast. ISBN: 84-415-1953-6.

- [10] IEEE 802.11n
http://en.wikipedia.org/wiki/IEEE_802.11n-2009.

- [11] <http://www.cisco.com/c/en/us/support/docs/voice/voice-quality/7934-bwidth-consume.html>.
- [12] GENERIC MODELING OF MULTIMEDIA TRAFFIC SOURCES. Hassan HASSAN and Jean-Marie GARCIA and Olivier BRUN. Laboratory for Analysis and Architecture of Systems
- [13] Trace Files
<http://www2.tkn.tu-berlin.de/research/trace/ltvt.html>
- [14] Marco Aurelio Alzate Monroy. Modelos de tráfico y calidad de servicio en redes IP, Seminario Actividades de Investigación. Maestría en Ciencias de la Información y las Comunicaciones.
- [15] Ubuntu
<http://www.ubuntu.com/>

Glosario

En este apartado se presenta un glosario con los términos y abreviaturas utilizadas a lo largo de este PFC.

Glosario

A

AP – *Access Point*.

B

C

CBR – *Constant Bit Rate*. Modo de generación de tráfico a una velocidad constante.

CSMA/CA – *Carrier Sense Multiple Access with Collision Avoidance*.

CSMA/CD – *Carrier Sense Multiple Access with Collision Detection*.

CTS – *Clear To Send*.

D

DCF – *Distributed Coordination Function*.

DSSS– *Direct Sequence spread Spectrum*.

E

F

FIFO – *First In First Out*.

FTP – *File Transfer Protocol*.

G

GB – *Gigabyte*.

H

HTTP – *Hipertext Transfer Protocol*.

I

IEEE – *Institute of Electrical and Electronics Engineers*.

IP – *Internet Protocol*.

ITU – *International Telecommunication Union*.

J

K

L

M

MAC – *Medium Access Control*.

MB – *Megabyte*.

MBR – *Multiple Bit Rate*.

MOS – *Mean Opinion Score.*

MIMO – *Multiple-input Multiple-output*

MPEG – *Moving Picture Experts Group.*

N

O

OSI – *Open System Interconnection.*

OFDM– *Orthogonal Frequency Division Multiplexing.*

P

PoE – *Power Over Ethernet.*

Q

QoS – *Quality of Service.*

QoE – *Quality of Experience.*

R

RTP – *Real-time Transport Protocol.*

RSSI – *Received Signal Strength Indicator.* Indicador de fuerza de la señal recibida.

S

STA. *Station.* Estación.

SIP – *Session Initiation Protocol.*

SISO – *Multiple-input Multiple-output*

SO – *Sistema Operativo.*

MOS – *Mean opinion score.*

T

TCP – *Transmission Control Protocol.*

U

UDP – *User Datagram Protocol.*

V

VBR – *Variable Bit Rate.*

W

WAN – *Wide Area Networks.*

WLAN – *Wireless Local Area Networks.*

X

Y

Anexo A: planos

En el siguiente apéndice se muestran los planos del complejo hotelero objeto de estudio de este PFC.