|  |  |  |
| --- | --- | --- |
| Simulador del protocolo 802.11 |  | |
| El presente proyecto pretende la construcción de un simulador del protocolo 802.11 el cual define el uso de los dos niveles inferiores de la arquitectura OSI (capas física y de enlace de datos), especificando sus normas de funcionamiento en una WLAN. | | Proyecto final de carrera |

ÍNDICE:

Capitulo 1 :

1.2.-Resumen………………………………………………………………………………………………………..

1.3.-Introducción……………………………………………………………………………………………………

Capitulo 2:

2.1.-Estado del arte en cuanto a aplicaciones ……………………………………………………….

2.2.-Estado del arte en cuanto a otros simuladores…………………………………………………

Capitulo 3:

3.1.-Requisitos de proyecto

3.1.1.-Diagrama de casos de uso…………………………………………………

3.1.2- Plantillas para los casos de uso…………………………………………

3.2.-Arquitectura

3.3.1.-Decisiones de diseño……………………………………………………….

3.3.2.-Diagrama de las clases más representativas…………………….

3.3.-Implementación:

3.3.1.-Descripción de la interfaz de usuario ……………………………..

3.3.2.-Construcción del diagrama de secuencia…………………………

3.3.3.-Construcción del diagrama tridimensional ………………………

3.3.4.- Construcción de la tabla de resultados…………………………….

3.3.5.- Empleo de obstáculos…………………………..………………………….

3.3.6.- Implementación del algoritmo CSMA/CA….…………………….

Capitulo 4:

4.1.- Experimentación ……………………………………………………………………..

4.1.1.- Medición de la cobertura …………………………………………

4.1.2.- Detección del problema del nodo oculto …….…………...

Capitulo 5:

5.1.-Conclusiones y desarrollos futuros …………………………………………….

Anexo:

Manual de usuario…………………………………………………………………………………………

Bibliografía…………………………………………………………………………………………………………………

Capitulo 1 :

1.1.- RESUMEN:

Las redes inalámbricas se han convertido en la actualidad en una alternativa barata, fiable y eficaz en el marco de las transmisiones digitales. El hecho de utilizar una banda de frecuencias no regulada, de exigir la interoperabilidad entre dispositivos de diversos fabricantes, de reducir precios y de proporcionar valores de movilidad, integridad, escalabilidad y desarrollo tecnológico están permitiendo posicionar la tecnología inalámbrica como una alternativa seria a su equivalente cableada, e incluso, superándola en ciertos entornos y condiciones.

Conocer el máximo througput teórico en condiciones ideales es el primer paso para considerar el uso de dispositivos 802.11 en un medio hostil, y para ello pueden utilizarse simuladores los cuales deberían implicar una descripción física del medio donde se instalará la red así como la configuración que se pretende emplear y ciertas características físicas como son las sensibilidades al medio por parte de los dispositivos , potencias de emisión y tasa máxima

permitida de transmisión.

Partiendo de estas premisas este proyecto tiene como objetivo el diseño y construcción de un simulador de redes inalámbricas basadas en el estándar IEEE 802.11 el cual pretender poder ser utilizado para predecir y estudiar el comportamiento de una red en función de parámetros físicos como la distancia o la presencia de obstáculos y las características de los dispositivos empleados(sensibilidad al medio , potencia de emisión …etc… ).

Capítulo 2 :

**2.1.- Estado del arte en cuanto a aplicaciones:**

En la actualidad la tecnología inalámbrica abre nuevas expectativas en diferentes áreas de aplicación puesto que ha adquirido una gran importancia debido a un conjunto de claves. La *Wireless Ethernet Compatibility Alliance* garantiza la compatibilidad de funcionamiento entre equipos de distintos fabricantes. Los precios se han reducido considerablemente ofreciendo una alternativa en infraestructuras donde el cable es inviable o la movilidad es un valor añadido. Fabricantes de ordenadores y *PDAs* incluyen de serie sistemas WiFi en sus equipos. Los dispositivos de comunicación móviles comienzan a incluir sistemas WiFi de serie, lo que podría proporcionar conexiones de banda ancha.

A continuación se resumen los distintos campos de desarrollo en la actualidad.

***Demótica:*** Actualmente son varios los fabricantes que integran en un mismo equipo un punto de acceso inalámbrico más un modem y un router. Incluso los proveedores de Internet ofrecen acceso inalámbrico en sus ofertas regalando dichos dispositivos. Estos dispositivos son usados para interconectar uno o varios ordenadores y proporcionar acceso compartido a Internet, impresoras, discos duros, etc. En la actualidad diversos fabricantes están desarrollando dispositivos de ocio para acceso a contenidos digitales. Es el caso de las nuevas cadenas de música que, además de las

características estándar de reproducción de DVDs, radio, y otros, son capaces de

conectarse a librerías de audio codificadas en MP3, tanto de forma local a la red del hogar, o a servidores de Internet a través del acceso compartido en banda ancha. La televisión digital interactiva precisa de un canal de retorno yWiFi se presenta como una alternativa. La distribución de audio y video dentro del hogar puede tomar forma gracias a los nuevos estándares 802.11 a y g que ofrecen un mayor ancho de banda.

***Redes ciudadanas***: Este tipo de aplicaciones pretende compartir una conexión a Internet entre varios ciudadanos conectados. En España no está proliferando mucho esta aplicación, puesto que la calidad y estabilidad de la red precisa de inversiones. Existen diversas páginas web que ofrecen información y un cierto nivel de coordinación en ciertas regiones de España. Asímismo, los organismos de gobierno local no pueden proporcionar acceso legal según la reglamentación vigente a la hora de escribir este proyecto.

***Redes de área local inalámbricas en la empresa:*** Estas redes ofrecen ventajas como son la movilidad, la ausencia de cables, acceso a la red con independencia del puesto de trabajo, etc. Puede resultar una solución mejor que el uso del cable para tráficos bajos o entornos limpios de interferencias. Actualmente, las empresas adoptan soluciones mixtas, aprovechando su infraestructura cableada. Como se vió en 2.1.1

es posible usar estos dispositivos para proporcionar voz o también para transportar datos.

***Conexión inalámbrica en el teletrabajo:*** Un teletrabajador puede necesitar una conexión inalámbrica de banda ancha para tener acceso a los datos de una empresa en cualquier lugar. Normalmente una red privada virtual junto con una conexión inalámbrica a Internet permite tener acceso a dicho recurso. Actualmente operadoras de telefonía móvil ofrecen en sus ofertas productos que satisfagan lo anterior.

***Tecnología inalámbrica y la seguridad:*** En este sector una red inalámbrica podría permitir la interconexión de dispositivos de vigilancia como son cámaras o sensores. Hoy en día, en España, existe comercialmente el servicio de video vigilancia usando una conexión a Internet de banda ancha.

***Red inalámbrica en recintos portuarios y aeroportuarios:*** Aplicaciones actuales son la carga de combustible gestionada con la ayuda de un dispositivo inalámbrico en el camión cisterna, gestión de carga y descarga en los puertos mediante dispositivos inalámbricos en las grúas o interconexión de buques a la red inalámbrica del puerto para su uso y todo ello sin necesidad de cableado. Existen compañías de vuelo que ya ofertan acceso a Internet en sus vuelos.

***Red inalámbrica en el ámbito hospitalario:*** La gran ventaja de los equipos WiFi es que no interfieren en los equipos médicos, con lo que es posible una mejor gestión de los recursos de instrumentación e información.

***Red inalámbrica en la Universidad:*** Gracias a la cobertura existente en estas es posible acceder a información, descargar prácticas, ejercicios, etc. . . desde cualquier zona. Otra aplicación es la interconexión de edificios cercanos.

***Red inalámbrica en almacenes de distribución y grandes superficies:*** Dispositivos como *PDAs* o *Tablet PCs* permiten el uso de *software* para gestión de mercancias. Si a esto se añade una red inalámbrica la gestión se podrá llevar a cabo desde cualquier punto del almacén.

**2.1.- Estado del arte en cuanto a sistemas de simulación:**

A continuación se hace un resumen de los simuladores de red más importantes que existen en la actualidad y que pueden permitir el modelado de redes WiFi .

# Network Simulator NS2:

Se trata de un simulador de eventos discretos dirigidos a la investigación de redes.

Provee soporte para simulación de TCP, multicast y distintos algoritmos de enrutamiento sobre redes cableadas y redes inalámbricas de área local y redes usando enlaces satélite.



(Imagen del simulador NS2)

Soporta modo *ad-hoc* y distintos modelos de propagación. Está escrito en lenguaje C++, por lo que se necesita un compilador para la plataforma que se tenga instalada en la máquina. Las simulaciones son definidas mediante un lenguaje *script* llamado TCL, aunque existe un editor gráfico que evita tener que manejar este lenguaje. El simulador también dispone de una interfaz gráfica como herramienta adicional que permite visualizar las simulaciones.

Este simulador permite definir para cada simulación distintos algoritmos de enrutamiento, distintos generadores de tráfico, distintas disciplinas de encolado y QoS, fallos como pérdidas probabilísticas o deterministas, fallos en la conexión, distintos modelos de propagación de la señal, energía de los dispositivos, contadores, etc...

Nos ofrece datos sobre el estado del protocolo, flujos de paquete y probabilidad de encolamiento y de pérdida, movimiento de nodos en redes inalámbricas y notas de los sucesos más importantes.

Actualmente ofrece soporte para redes inalámbricas 802.11 en modo *ad-hoc* e infraestructura, un modelo espacial para un entorno en movimiento con patrones de movimiento, modo de funcionamiento PCF, mobile IP, redes *ad-hoc* multisalto, TCP sobre enlace inalámbrico, redes de área local con CSMA/CA, CSMA/CD y el protocolo *snoop*. Como se ha mencionado antes es posible.

definir distintos modelos de error, con lo que permite simular comunicaciones inalámbricas 802.11b en entornos industriales

En resumen, este simulador es una herramienta gratuita y muy potente dentro del

campo de la simulación de redes. Es a la vez muy flexible dada la posibilidad de trabajar con scripts tcl que permite agregar toda la potencia de un lenguaje de programación a los elementos propios de la simulación.

# OMNeT++:

En este caso también se tiene un entorno de simulación de eventos discretos. Su área de aplicación originaria era la simulación de redes de comunicación, pero debido a su arquitectura genérica y su flexibilidad está siendo usado en otros campos dentro de las tecnologías de la información como son el estudio de colas o arquitecturas *hardware*.

 OMNeT++ construye sus modelos a partir de una arquitectura modular. Cada componente o módulo está escrito en C++ y se integran en componentes más complejos o modelos mediante NED, un lenguaje de alto nivel. Este entorno posee interfaces gráficas debido a su arquitectura modular las simulaciones y los modelos pueden ser fácilmente integrados en las aplicaciones de usuario. Auque no se trate de un simulador de redes en realidad, está ganando popularidad en el ambiente científico como tal y detrás de él está creciendo una gran comunidad de desarrolladores. Puede ser usado para modelar protocolos de comunicación, redes de ordenadores y modelos de tráfico, sistemas multiprocesadores y sistemas distribuidos, etc. Se distribuye libremente bajo una licencia pública académica

y actualmente están escritos modelos para IPv6 bajo redes 802.11, Mobility Framework donde incluye un modelo para 802.11 y soporte para redes *ad-hoc* con soporte para usuarios móviles. Tiene soporte para prácticamente todos los sistemas operativos Linux, Unix y plataformas Win32.

## Cnet:

Es un simulador de redes de ordenadores. Este simulador no está enfocado al uso

comercial. Es un simulador de eventos discretos que permite experimentar con distintas

capas de enlace, red y protocolos de enrutamiento. Ha sido desarrollado y es usado por

cientos de estudiantes desde 1991. Únicamente funciona en plataformas Linux y su uso

se enfoca a la enseñanza.

****

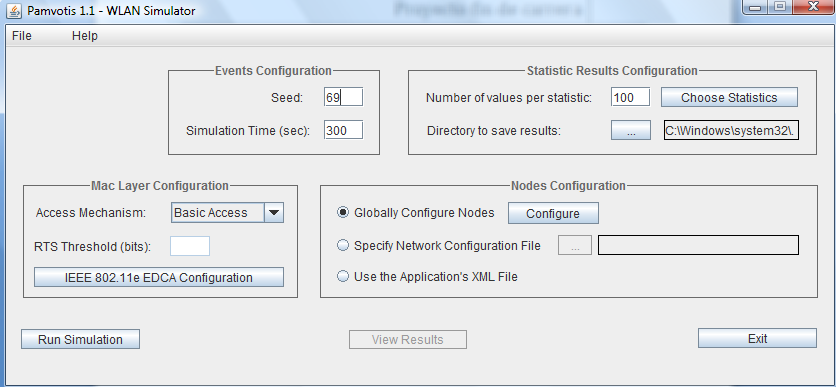
(Captura de pantalla del simulador Cnet)

# PYTHAGOR Y PANVOTIS:

Ambos pertenecen al mismo creador y al igual que Network Simulator (NS2) están creados bajo licencia Open Source para la creación por parte de los usuarios de sus propopios protocolos. Actualmente es compatible con todas las extensiones del estándar 802.11 a excepcción del nuevo estándar 802.11n.

Soportan todos los tipos de capa física y sus velocidades permitidas por el estándar y permite configurar el empleo de RTS/CTS así como su umbral de activación.También soporta distintos tipos de distribuciones de carga de tráfico y de carga útil de paquete y operaciones de red en modo mixto combinando nodos operando bajo el estándar 802.11b y 802.11g.

Por último señalar que Pythagor fue escrito en java mientras que Panvotis (mas reciente) está escrito en csharp y su configuración se hace a través de archivos xls.



(Captura de pantalla del simulador Panvotis)

Capítulo 3:

**3.1.- Requisitos del sistema:**

El requisito principal de este proyecto es la construcción de un software capaz de imitar el comportamiento de una red wireless construida bajo el estándar 802.11 con el fin de obtener una representación gráfica de la interacción de los dispositivos implicados según la configuración previa por parte del usuario y la posterior recogida de datos que den una idea aproximada del rendimiento llevado a cabo durante el tiempo de simulación.

El sistema deberá:

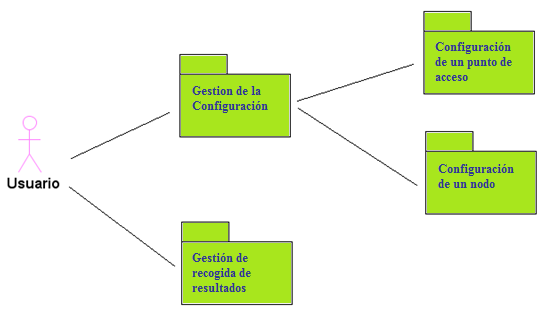
* Dar soporte a todos los tipos de capa física que se comercializan en la actualidad (802.11, 802.11a, 802.11b, 802.11g, 802.11n).
* Dar soporte a todos los rangos de datos permitidos en cada especificación de cada capa física.
* Permitir la configuración independiente de cada dispositivo.
* Dar soporte al mecanismo RTS/CTS, y permitir la configuración de su umbral de en cada nodo.
* Soporte de diferentes distribuciones para cargas de tráfico.
* Soporte del problema del nodo oculto, donde los nodos pueden ser configurados para investigar el problema.

Los datos a recoger serán los siguientes:

* Troughput.
* Utilización.
* Retardo de acceso al medio.
* Retardo de encolamiento.
* Retardo total de las tramas.
* Retardo del jitter.
* Longitud de la cola de tramas generadas.

**3.1.1.- Diagramas de subsistemas y casos de uso:**

# Diagrama de subsistemas:

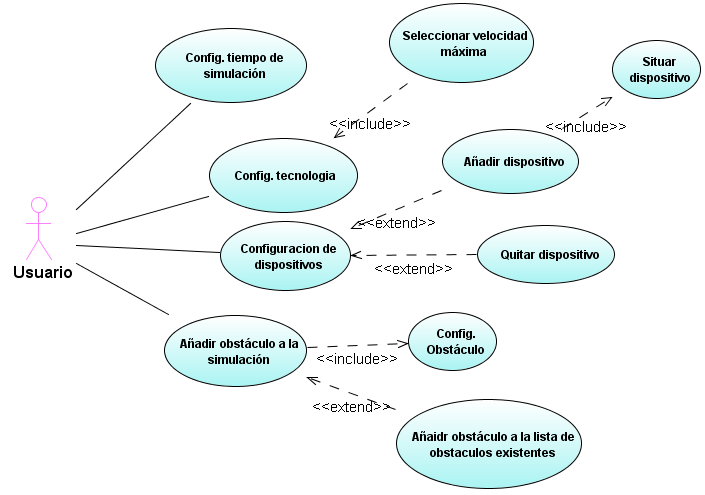


En la figura anterior podemos observar el diagrama de subsistemas de aplicación, como vemos existen 4 subsistemas claramente diferenciados por sus funciones.

El sub sistema “Gestión de la configuración” llevara acabó todos los procesos dedicados a recoger los parámetros necesarios para lanzar una simulación, mientras que los subsistemas de “Configuración de un punto de acceso” y “Configuración de un nodo” permitirán la recogida de parámetros que modificarán el funcionamiento de los dispositivos en la simulación.

El subsistema “Gestión de recogida de resultados” englobará todos los procesos encargados de recoger los resultados de la simulación y mostrarlos al usuario.

# Diagrama de subsistema “gestión de la configuración”:



|  |  |
| --- | --- |
| CU: | Config. tiempo de simulación |
| Objetivo: | Establecer el número de iteraciones que se ejecutarán en la simulación |
| Prerrequisito: | (Ninguno) |
| Actores: | Usuario |
| Acciones: | Establecer duración de la simulación. |
| CU: | **Config. tecnología** |
| Objetivo: | Configurar la simulación para una tecnología 802.11 determinada |
| Prerrequisito: | (Ninguno) |
| Actores: | Usuario |
| Acciones: | Configurar la simulación con los parámetros asociados a dicha tecnología (tiempos SIFS, DIFS, frecuencias de señal…etc...) |

|  |  |
| --- | --- |
| CU: | Config. dispositivos |
| Objetivo: | Permitir al usuario añadir o quitar de la simulación, tanto puntos de acceso como nodos inalámbricos. |
| Prerrequisito: | (Ninguno) |
| Actores: | Usuario |
| Acciones: |  |

|  |  |
| --- | --- |
| CU: | Seleccionar velocidad máxima |
| Objetivo: | Acotar el rendimiento de la red. |
| Prerrequisito: | Haber seleccionado una tecnología. |
| Actores: | Usuario |
| Acciones: | Establecer la velocidad máxima de transferencia a la que trabajaran todos los dispositivos |

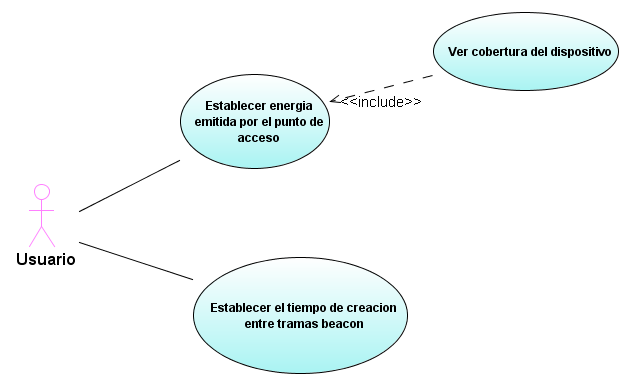
|  |  |
| --- | --- |
| CU: | Añadir dispositivo |
| Objetivo: | Añadir a la simulación un nuevo dispositivo |
| Prerrequisito: | (Ninguno) |
| Actores: | Usuario |
| Acciones: | Se solicita al sistema la generación de un nuevo dispositivo (punto acceso/ nodo) |

|  |  |
| --- | --- |
| CU: | Quitar dispositivo |
| Objetivo: | Quitar de la simulación un dispositivo (punto de acceso/ nodo) |
| Prerrequisito: | (Ninguno) |
| Actores: | Usuario |
| Acciones: | Se solicita al sistema la eliminación de un dispositivo (punto acceso/ nodo) |
| CU: | **Situar dispositivo** |
| Objetivo: | Situar en el espacio virtual el dispositivo seleccionado |
| Prerrequisito: | Haber generado previamente el dispositivo y tenerlo seleccionado |
| Actores: | Usuario |
| Acciones: | Solicitar al sistema la configuración en el objeto que representa el dispositivo de sus variables (x,y,z) referentes a la situación de dicho dispositivo |

|  |  |
| --- | --- |
| CU: | Añadir obstáculo a la simulación |
| Objetivo: | Situar entre dos dispositivos un obstáculo |
| Prerrequisito: | Tener creados previamente los dispositivos entre los cuales ira situado el obstáculo |
| Actores: | Usuario |
| Acciones: | Solicitar al sistema la incursión de un obstáculo para la simulación |

|  |  |
| --- | --- |
| CU: | Añadir obstáculo a la lista existente |
| Objetivo: | Ampliar la lista disponible de obstáculos. |
| Prerrequisito: | Ninguno |
| Actores: | Usuario |
| Acciones: | Solicitar al sistema la inclusión de una nueva línea en el fichero dedicado para la lista de obstáculos. |

# Diagrama del subsistema “configuración de un punto de acceso”:

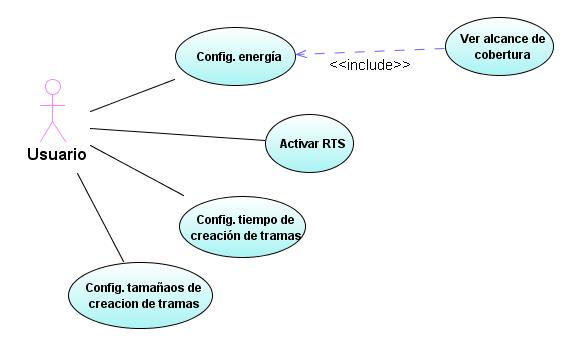
****

|  |  |
| --- | --- |
| CU: | Establecer energía emitida por el punto de acceso |
| Objetivo: | Configurar la energía emitida por un punto de acceso. |
| Prerrequisito: | Haber creado un objeto punto de acceso. |
| Actores: | Usuario. |
| Acciones: | Solicitar al sistema la modificación del parámetro relacionado con la energía de dicho punto de acceso. |

|  |  |
| --- | --- |
| CU: | Establecer tiempo de creación entre tramas beacon. |
| Objetivo: | Configurar el tiempo transcurrido entre la creación de las tramas de baliza. |
| Prerrequisito: | Haber creado un objeto punto de acceso. |
| Actores: | Usuario |
| Acciones: | Solicitar al sistema la modificación del contador relacionado con el tiempo de creación de tramas beacon. |

|  |  |
| --- | --- |
| CU: | Ver cobertura del dispositivo. |
| Objetivo: | Ver la disminución de la señal en función de la distancia |
| Prerrequisito: | Haber creado un objeto punto de acceso |
| Actores: | Usuario |
| Acciones: | Solicitar al sistema ver una gráfica representativa de la disminución de la señal en decibelios con respecto a la distancia. |

# Diagrama del subsistema “configuración de un nodo ”:



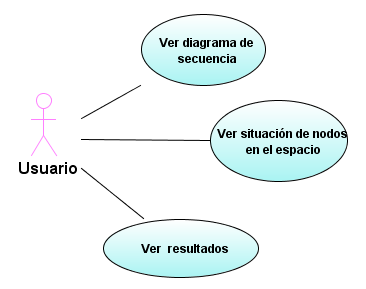
|  |  |
| --- | --- |
| CU: | Config. energía |
| Objetivo: | Establecer la energía en mW. que emitirá la antena de un nodo al trasmitir |
| Prerrequisito: | Haber creado un objeto nodo. |
| Actores: | Usuario |
| Acciones: | Solicitar al sistema la modificación del parámetro relacionado con la energía de dicho nodo. |
| CU: | **Ver alcance de cobertura** |
| Objetivo: | Ver la disminución de la señal en función de la distancia |
| Prerrequisito: | Haber creado un objeto nodo |
| Actores: | Usuario |
| Acciones: | Solicitar al sistema ver una gráfica representativa de la disminución de la señal en decibelios con respecto a la distancia. |

|  |  |
| --- | --- |
| CU: | Activar RTS |
| Objetivo: | Activar en dicho nodo la función RTS |
| Prerrequisito: | Haber creado un objeto nodo. |
| Actores: | Usuario |
| Acciones: | Solicitar al sistema la activación de la función RTS de dicho nodo así como el umbral de activación de dicha función. |

|  |  |
| --- | --- |
| CU: | Config. tamaños de creación de tramas |
| Objetivo: | Establecer el tamaño que tendrán las tramas que se crearan. |
| Prerrequisito: | Haber creado un objeto nodo. |
| Actores: | Usuario |
| Acciones: | Solicitar al sistema la configuración del tamaño en bytes de las tramas con tres funciones: constante, continua y exponencial. |

|  |  |
| --- | --- |
| CU: | Config. el tiempo entre creación de tramas |
| Objetivo: | Establecer el tiempo que tardarán las tramas en crearse una detrás de otra. |
| Prerrequisito: | Haber creado un objeto nodo. |
| Actores: | Usuario |
| Acciones: | Solicitar al sistema la configuración del tiempo de creación de tramas con tres funciones para la creación de estos tiempos: constante, continua y exponencial. |

# Diagrama del subsistema “configuración de un nodo”:



|  |  |
| --- | --- |
| CU: | Ver diagrama de secuencia |
| Objetivo: | Poder ver de manera grafica una represetación de la ejecución del algoritmo CSMA/CA. |
| Prerrequisito: | Haber ejecutado una simulación. |
| Actores: | Usuario. |
| Acciones: | Solicitar al sistema la representación gráfica de la interacción de los dispositivos implicados en la simulación en función del tiempo. |

|  |  |
| --- | --- |
| CU: | Ver situación de nodos |
| Objetivo: |  |
| Prerrequisito: | Haber ejecutado una simulación. |
| Actores: | Usuario. |
| Acciones: | Solicitar al sistema una vista tridimensional de la situación de los nodos en un espacio tridimensional. |

|  |  |
| --- | --- |
| CU: | Ver resultados |
| Objetivo: |  |
| Prerrequisito: | Haber ejecutado una simulación. |
| Actores: | Usuario. |
| Acciones: |  |

**3.1.2.-Plantillas de los casos de uso:**

# Subsistema “Gestión de la configuración”:

Configuración tiempo de simulación:

Descripción general:

Este caso de uso se encarga de establecer el tiempo en microsegundos que dura la simulación.

Actores que participan:

El usuario de la aplicación.

Precondición:

No hay una simulación en curso.

Flujo de eventos:

1.-El caso de uso se inicia junto con la ejecución de la aplicación.

2.-El usuario introduce el tiempo de simulación deseado (en microsegundos) en la casilla correspondiente (componente JText de la librería swing).

3.-El caso de uso finaliza cuando el usuario inicia la ejecución.

Configuración de la tecnología:

Descripción general:

Este caso de uso se encarga de establecer la tecnología que se utilizará durante la simulación. Según la tecnología escogida los tiempos para SIFS, DIFS ,tamaño del slot , frecuencia de señal y tamaño inicial de la ventana de backoff cambiarán.

Actores que participan:

El usuario de la aplicación.

Precondición:

No hay una simulación en curso.

Flujo de eventos:

1.-El usuario pulsa sobre el componente comboBox para seleccionar ua tecnolgía disponible .

2.-El Sistema muestra al usuario.

3.-El usuario selecciona la tecnología deseada (Ej:802.11,802.11a…)

4.-Se ejecuta el caso de uso Seleccionar velocidad máxima.

Seleccionar velocidad máxima :

Descripción general:

Permite al usuario establecer una velocidad máxima

Actores que participan:

El usuario de la aplicación.

Precondición:

No hay una simulación en curso.

Flujo de eventos:

1.-Este caso de uso es iniciado por el caso de uso Configuración de la tecnología.

2.- El usuario selecciona una de las velocidades permitidas por la tecnología previamente seleccionada.

3.-El sistema recoge la velocidad seleccionada y finaliza el caso de uso.

Configuración de dispositivos:

Descripción general:

Este caso de uso se utiliza para permitir al usuario añadir o quitar puntos de acceso o nodos a la simulación.

Actores que participan:

El usuario de la aplicación

Precondición:

No hay una simulación en curso

Flujo de eventos:

1.-El usuario solicita al sistema la creación de nodos o puntos de acceso (activa la pestaña dedicada a esta función)

2.-El sistema permite al usuario quitar o colocar puntos de acceso o nodos.

3.-Si el usuario desea añadir un dispositivo se activa el caso de uso “Añadir dispositivo”.

4.-Si el usuario desea quitar un dispositivo se activa el caso de uso “Quitar dispositivo”.

Añadir dispositivos:

Descripción general:

Este caso de uso se utiliza para permitir al usuario añadir un dispositivo putno de acceso o nodo a la simulación.

Actores que participan:

El usuario de la aplicación

Precondición:

No hay una simulación en curso

Flujo de eventos:

1.-El usuario solicita al sistema la creación de nodos o puntos de acceso (activa la pestaña dedicada a esta función)

Quitar dispositivos:

Descripción general:

Este caso de uso se utiliza para permitir al usuario añadir un dispositivo putno de acceso o nodo a la simulación.

Actores que participan:

El usuario de la aplicación

Precondición:

No hay simulación en curso y previamente el usuario a creado un dispositivo.

Flujo de eventos:

1.-El usuario solicita al sistema la creación de nodos o puntos de acceso (activa la pestaña dedicada a esta función).

2.-El sistema elimina de la lista de dispositivos el dispositivo seleccionado.

Situar dispositivos:

Descripción general:

Este caso de uso permite al usuario situar en el espacio el dispositivo configurando sus coordenadas x, y, z.

Actores que participan:

El usuario de la aplicación

Precondición:

No hay una simulación en curso y el usuario ha creado previamente un dispositivo.

Flujo de eventos:

1.-El usuario selecciona el dispositivo.

2.-El sistema permite al usuario configurar las coordenadas de situación del dicho dispositivo.

# Subsistema “Gestión de un punto de acceso”:

Establecer energía emitida por el punto de acceso:

Descripción general:

Este caso de uso permite al usuario configurar la energía emtida por un AP en mW.

Actores que participan:

El usuario de la aplicación

Precondición:

No hay una simulación en curso y el usuario ha seleccionado un AP.

Flujo de eventos:

1.-El usuario selecciona el dispositivo.

2.-El sistema permite al usuario configurar a través de un componte de la librería swing dicho parámetro.

3.-Se ejecuta el caso de uso Ver alcance de cobertura.

Ver alcance de cobertura:

Descripción general:

Permite al usuario ver una representación grafica de la disminución de la potencia de la señal del AP con respecto a la distancia.

Actores que participan:

El usuario de la aplicación

Precondición:

No hay una simulación en curso y el usuario acaba de modificar el parámetro que configura la potencia de emisión de un AP.

Flujo de eventos:

1.-El sistema genera una grafica (gracias a la api java chart) con la disminución de la potencia del dispositivo con la distancia.

# Subsistema “Gestión de un nodo”:

Configurar energía:

Descripción general:

Este caso de uso permite al usuario configurar la energía emitida por un nodo en mW.

Actores que participan:

El usuario de la aplicación

Precondición:

No hay una simulación en curso y el usuario ha seleccionado un nodo.

Flujo de eventos:

1.-El usuario selecciona el dispositivo.

2.-El sistema permite al usuario configurar a través de un componte de la librería swing dicho parámetro.

3.-Se ejecuta el caso de uso Ver alcance de cobertura.

Activar RTS:

Descripción general:

Este caso de uso permite al usuario configurar la activación de las tramas RTS en cada nodo.

Actores que participan:

El usuario de la aplicación

Precondición:

No hay una simulación en curso y el usuario ha seleccionado un nodo.

Flujo de eventos:

1.-El usuario selecciona el dispositivo.

2.-El sistema permite al usuario la activación de RTS.

3.-El usuario activa dicho parámetro.

4.-El sistema solicita al usuario el tamaño en bytes del umbral de activación de las tramas RTS.

Configurar tiempo de creación de tramas:

Descripción general:

Este caso de uso permite al usuario configurar el tiempo que transcurre entre que se genera una trama y la siguiente. Una trama generada que no puede ser enviada permanecerá en la cola de envío.

Actores que participan:

El usuario de la aplicación

Precondición:

No hay una simulación en curso y el usuario ha seleccionado un nodo.

Flujo de eventos:

1.-El usuario selecciona el dispositivo.

2.-El sistema permite al usuario configurar un valor denominado “mean time” sobre el cual se crearan los diversos tiempos (la utilización de este valor se verá detallado más adelante).

3.-El usuario introduce este valor

4.-El sistema solicita al usuario la elección de una función de generación que podrá ser:

A: constante

B: uniforme

C: exponencial

5.-El usuario elige con qué función trabajará.

Configurar tamaños de creación de tramas:

Descripción general:

Este caso de uso permite al usuario configurar el tamaño que tendrán las tramas que generaran y enviaran los nodos con el fin de simular los diversos tamaños de tramas que requerirán las aplicaciones que se encuentran en la capa aplicación del modelo OSI .

Actores que participan:

El usuario de la aplicación

Precondición:

No hay una simulación en curso y el usuario ha seleccionado un nodo.

Flujo de eventos:

1.-El usuario selecciona el dispositivo.

2.-El sistema permite al usuario configurar un valor denominado “mean size” sobre el cual se generaran las diversas tallas de trama (la utilización de este valor se detalladamente mas adelante).

3.-El usuario introduce este valor

4.-El sistema solicita al usuario la elección de una función de generación que podrá ser:

A: constante

B:uniforme

C:exponencial

5.-El usuario elige con que función trabajará.

# Subsistema “Gestión de recogida de resultados”:

Ver diagrama de secuencia:

Descripción general:

Permite al usuario ver una representación grafica del comportamiento de los dispositivos durante la ejecución de la simulación.

Actores que participan:

El usuario de la aplicación

Precondición:

Ha terminado de ejecutarse una simulación.

Flujo de eventos:

1.- El sistema termina de ejecutar una simulación.

2.- El usuario puede interactuar con el diagrama desplazándolo de arriba abajo y de derecha a izquierda para recorrerlo.

Ver situación de nodos en el espacio:

Descripción general:

Este caso de uso permite al usuario ver la distribución espacial de los dispositivos.

Actores que participan:

El usuario de la aplicación

Precondición:

Ha terminado de ejecutarse una simulación.

Flujo de eventos:

1.- El sistema termina de ejecutar una simulación.

2.- El usuario puede observar a través del diagrama la situación de los nodos, además de interactuar con él para desplazarse , girar la imagen generada o ver las conexiones que se han generado en la simulación.

Ver resultados:

Descripción general:

Este caso de uso permite al usuario ver los resultados obtenidos (troughput , jitter …etc...) por cada dispositivo a través de una tabla.

Actores que participan:

El usuario de la aplicación

Precondición:

Ha terminado de ejecutarse una simulación.

Flujo de eventos:

1.- El sistema termina de ejecutar una simulación y recoge los datos para mostrarlos en una tabla.

2.- El usuario además de poder ver los resultados más representativos de una simulación podrá ver una gráfica comparativa por cada parámetro medido.

**3.2.- Arquitectura Software:**

**3.2.1.-Decisiones de diseño:**

En el siguiente diagrama se muestran los paquetes empleados y su distribución (según tu empleo) en el modelado de tres capas siguiendo el patrón de diseño DAO:

**Preservación de datos**

**Lógica de negocio**

**Presentación**

ObstacleConfigurationPanel.java

NodeConfigurationPanel.java

AccessPointConfigurationPanel.java

DataSet.java

ConfigSensitivity.java

GraphicRepresentation.java

Graphic3D.java

InterFace.java

Line.java

Point2D.java

Point3D.java

TaskForJProgressBar.java

ViewPreSimulation.java

**Interfaz**

Diagram.java

ElementPlotted.java

ObjectPlotted.java

**Diagram**

**Java swing**

**Java chart**

**JXL**

**File**

AccessPoint.java

AssociationResponse.java

BeaconFrame.java

Cannel.java

Frame.java

Node.java

RequestFrame.java

WirelessChannel.java

**ElementsForSim**

Para la implementación del proyecto decidí utilizar el lenguaje de programación java por ser un lenguaje multiplataforma y totalmente orientado a objetos y para su diseño me he basado en el patrón DAO (Data Access Object) para facilitar la gestión de distintas fuentes de datos. Por ello la aplicación se ha dividido en tres módulos bien diferenciados:

1.-Presentacion : interface de usuario.

2.-Capa de modelo de negocio : encargada de realizar la simulación.(algoritmos csma/ca , algoritmo de back-off y medidas de la interacción con el medio )

3.-Capa de conservación de datos: encargada de recuperar datos de archivos “.excel” y archivos “.txt” con resultados de la simulación.

Para la creación de la interface de usuario he utilizado Java Swing. Java Swing es una biblioteca gráfica para Java que incluye “widgets” (componente o control visual reutilizable) para la creación de la interfaz de usuario tales como tablas, cajas de texto, despegables…etc…

Para la conexión de la capa de presentación con la capa de negocio utilizo la clase DataSet , la cual permite la creación de objetos que contienen todos los parámetros necesarios para ejecutar una simulación y que son recogidos de la interfaz.

La capa denominada como lógica de negocio representa a todas las clases implicadas en la simulación: AccessPoint, Node, Channel…etc. La clase “main” en la encargada de recoger el objeto DataSet mencionado anteriormente y ejecutar la función generateSimulation (también se encuentra en la clase main).

La capa preservación de datos implica a todas aquellas clases encargadas de acceder a ficheros de configuración (ficheros excel) estos son:

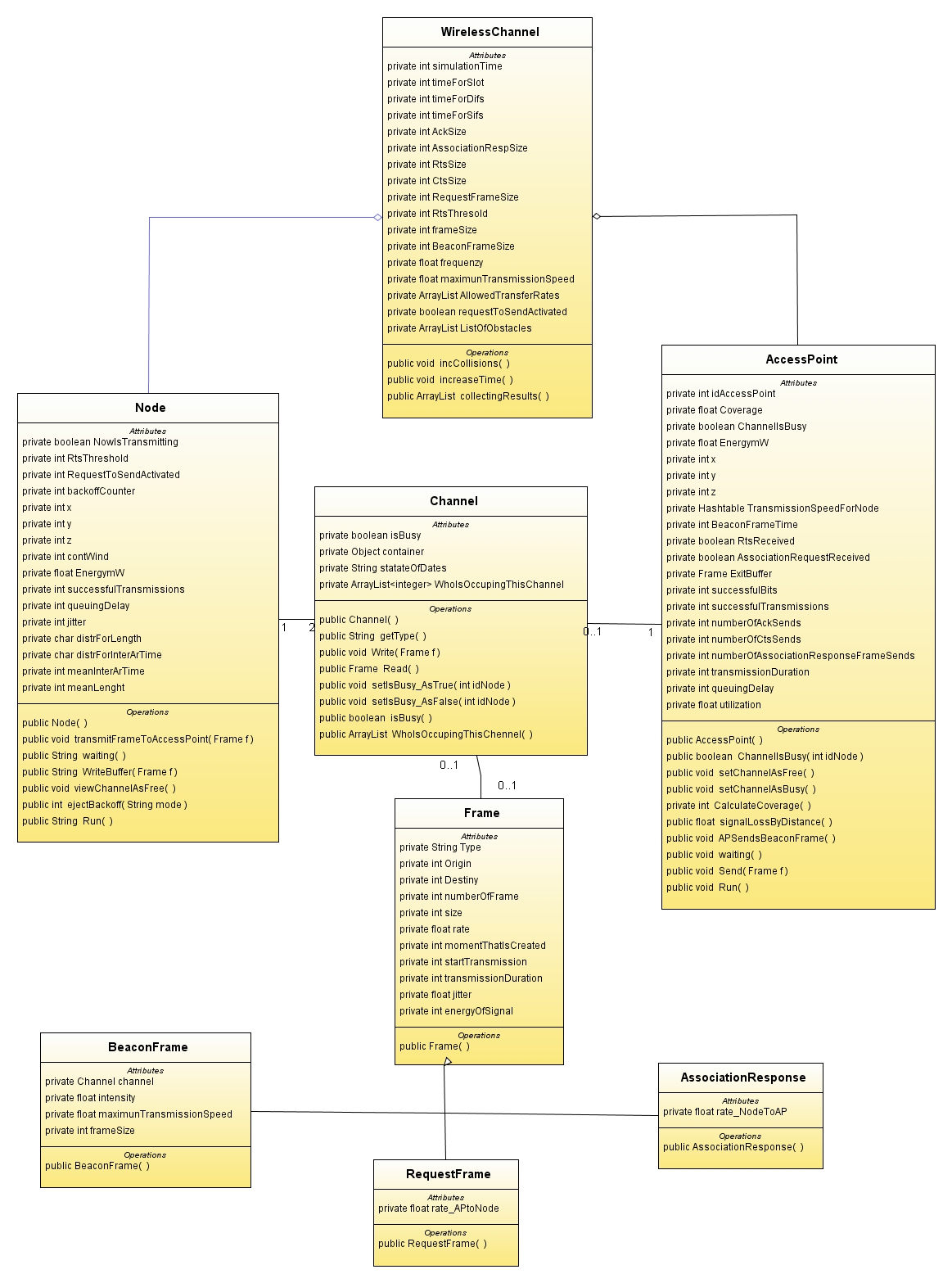
-Materials.xls: Lista de materiales con sus respectivas atenuaciones.

-Basic configuration.xls: Presenta la configuración por defecto.

-Relationship between rate and power received.xls: Contiene las relaciones de sensibilidad de los dispositivos para cada tecnología (en la aplicación y antes de lanzar una ejecución , estos datos pueden ser modificados para cada dispositivo de manera independiente).

**3.2.2.-Diagrama de las clases más representativas:**

## Paquete “ElementsForSim”:



**Descripción de las clases:**

**Clase WirelessChannel:**

**Esta clase contendrá todos los datos necesarios para la configuración además de referencias a todos los dispositivos.**

**Clase Node:**

**Esta es la clase que representará durante la simulación a los dispositivos inalámbricos. Contiene un conjunto de variables de configuración como son las requeridas para fijar su posición (int x, int y, int z) o la activación de tramas RTS, y otro conjunto de variables para la recogida de datos del rendimiento desarrollado, como son el numero de tramas mandadas con éxito (numberOfSuccessfulFrame) o la suma de los tiempos que han pasado las tramas en la cola de envío (queuingDelay).**

**Clase AccessPoint:**

**Esta clase representará los puntos de acceso que el usuario añadirá a la simulación, además de contener variables para su configuración y para la recogida de su rendimiento , podemos observar que contiene además una tabla (Hashtable) que cada índice contendrá una referencia a un nodo al que podrá mandar datos acompañado de la velocidad optima a la que se podrán enviar estos datos .Esta velocidad es previamente calculada teniendo en cuenta parámetros como la intensidad que llega al nodo y la sensibilidad de este (esta velocidad es calculada en el objeto nodo al recibir un beaconFrame y percibir su intensidad , y es comunicada al AP con una trama de solicitud de conexión RequestFrame).**

**Clase Channel:**

**Representará el buffer de entrada y de salida de cada dispositivo .Cada nodo estará asociado a un objeto Channel que será su buffer de entrada y otro que será el buffer de salida y buffer de entrada de un AP , mientras que cada AP tendrá asociado solo objeto Channel como buffer de entrada y uno más por cada nodo asociado a él (buffers de entrada de cada nodo).**

**Clase Frame:**

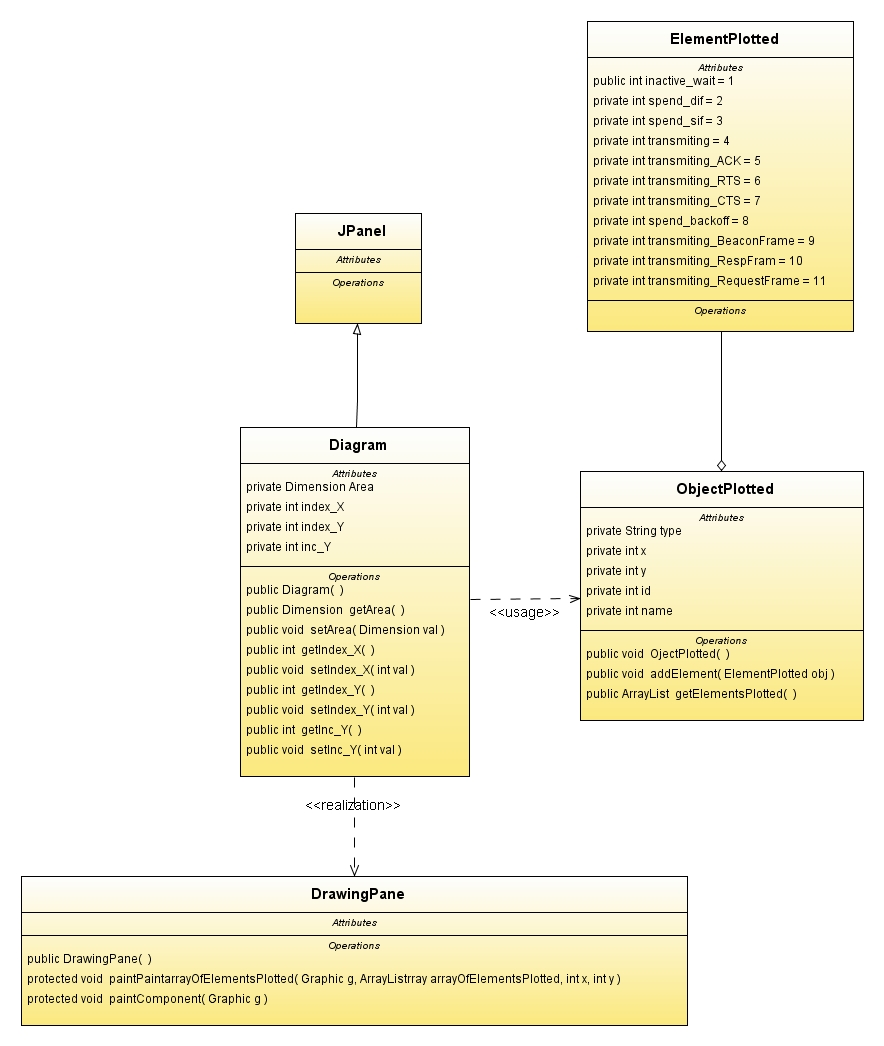
**Representará las tramas que se intercambian los dispositivos de la red.De esta clase extienden las siguientes:**

**BeaconFrame: Empleada por el punto de acceso para avisar de su presencia a los nodos más cercanos.**

**RequestFrame: Usada por los nodos para solicitar una conexión al AP una vez recibida una trama beacon.**

**AssociationResponse: Contestación de un AP a un nodo al recibir una trama RequestFrame.**

## Paquete “Diagram”:



El paquete Graphic contiene las clases necesarias para representar de manera gráfica sobre un elemento “JPanel” (librería java swing) el comportamiento de cada dispositivo de la red durante la ejecución del algoritmo CSMA/CA.

Las clases implicadas son las siguientes:

**Diagram:**

Es el panel sobre que el que se pintará, por ello entiende de la clase JPanel . A su vez esta clase contiene a la clase “**DrawingPane”** para poder dotar al panel de la capacidad de bibujar sobre el y actualizarse (en tamaño) cada vez que se ejecute le función paint() o repaint().

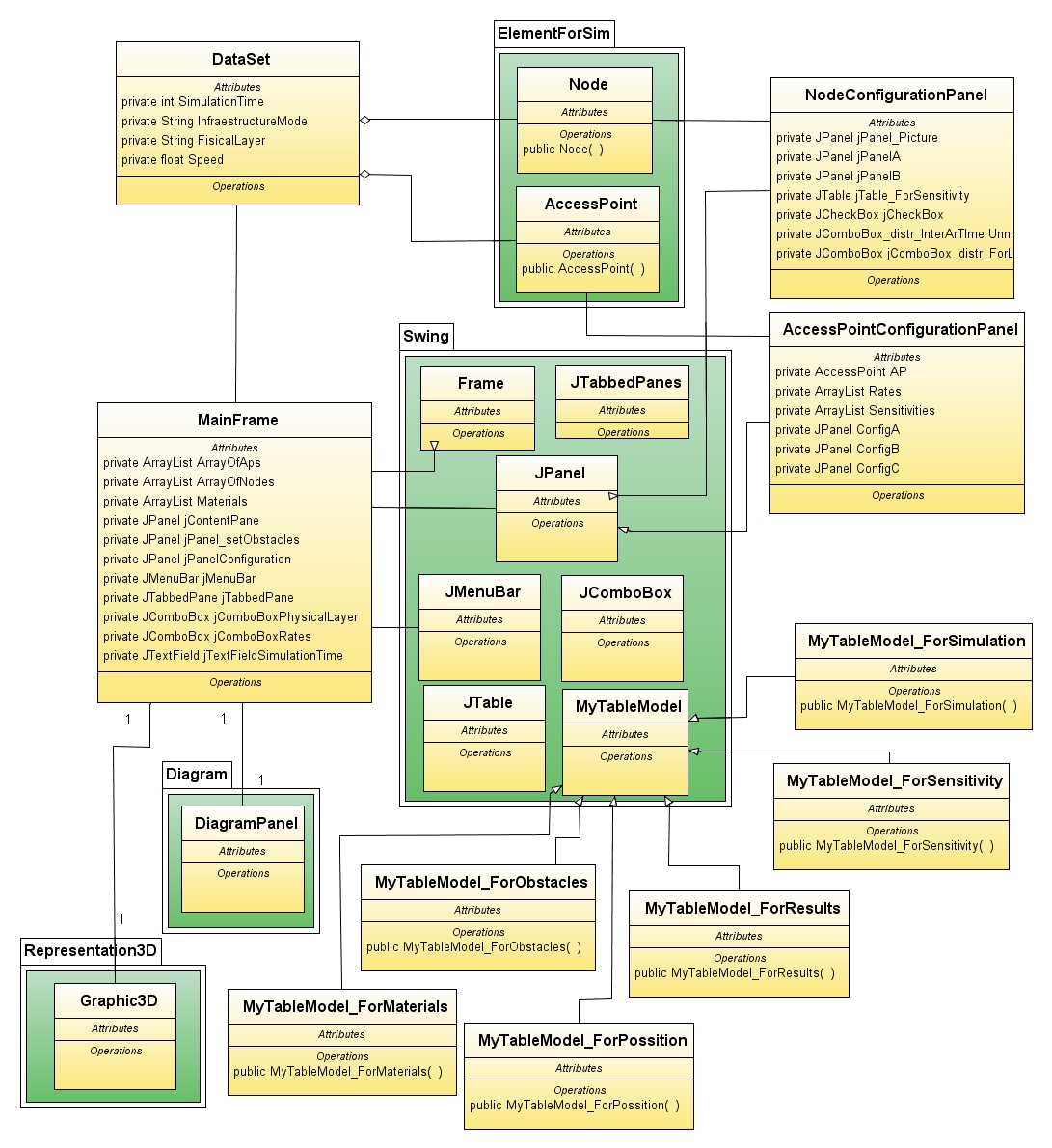
**DrawingPane:**

Sera la clase utilizada como “pincel” para dibujar sobre el componente Jpanel que define la clase Diagram , por ello esta clase contiene las funciones que recibirán como parámetro un objeto de la clase Graphic dotada de las funciones paint y repaint.

**ObjectPlotted:**

Cada dispositivo (nodo o punto de acceso) está relacionado con un objeto ObjectPlotted el cual con tiene la imagen que se mostrará en el diagrama además de una lista de instancias de la clase **ElementsPlotted** .

## Paquete “InterFace”:



(\*)Para hacer más sencillo el diagrama he obviado las operaciones de tipo get/set

El paquete interfaz contiene a todas las clases utilizadas para la construcción de la interfaz gráfica de usuario, por ello la mayoría de estas clases contiene objetos del paquete swing. Este, junto a los paquetes “Graphic3D” y “Diagram” forman la capa de presentación con la que interactuará el usuario.

Las clases implicadas son las siguientes:

**DataSet:**

Esta es la clase utilizada para conectar la capa de presentación con la lógica de negocio .Todos sus atributos hacen referencia a algún parámetro de configuración de una simulación.

**MainFrame:**

Esta clase contiene la ventana principal del programa ( por ello extiende de una clase “Frame” del paquete swing).

**NodeConfigurationPanel:**

Esta clase permitirá al usuario configurar un nodo concreto que previamente haya creado.

**NodeConfigurationPanel:**

Esta clase permitirá al usuario configurar un nodo concreto que previamente haya creado.

**Clases del tipo “MyTableModel”:**

Estas clases son utilizadas por los componentes de java swing Jtable para configurar las tablas , en ellas se especifican aspectos como el numero de filas y columnas , asi como el nombre de las mismas y la posibilidad de ser editadas

**3.3.-Implementación:**

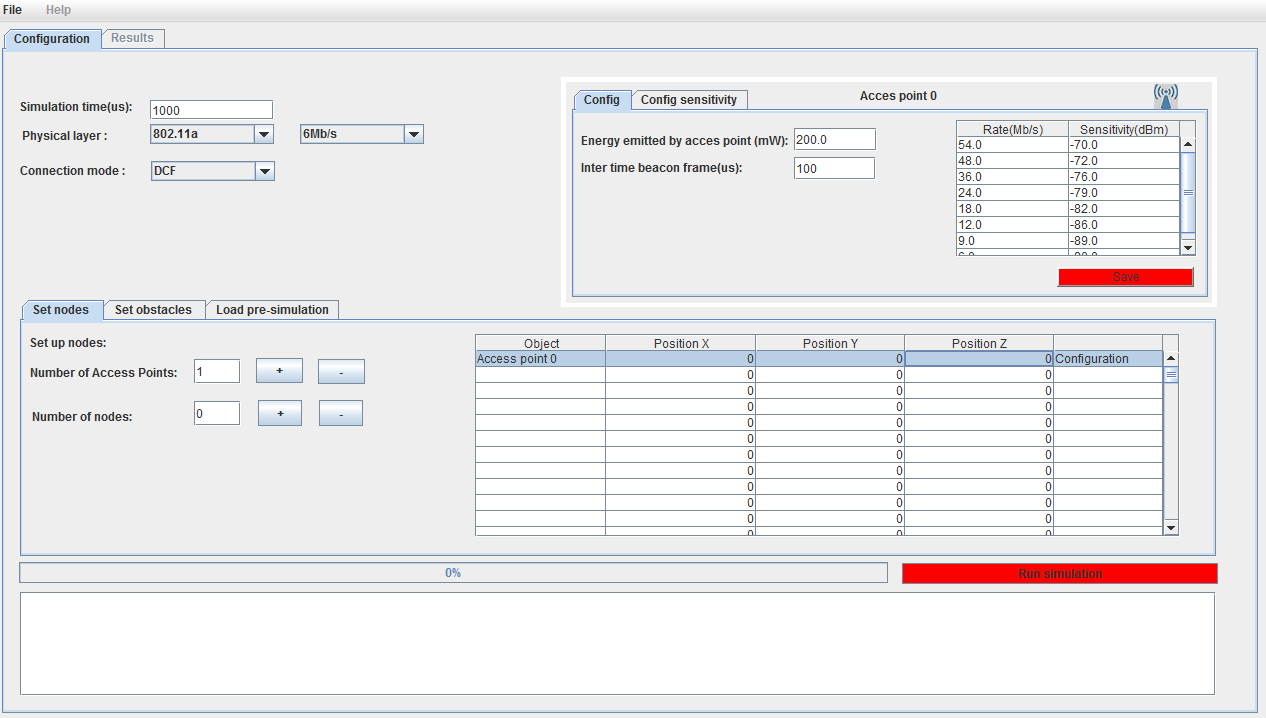
**3.3.1.- Descripción de la interfaz de usuario:**

Este es el medio por el cual el usuario podrá elegir y configurar todos los parámetros de la simulación.

Ha sido diseñada en “eclipse “con la ayuda del plugin Visual Editor de Java.

La interfaz de usuario está dividida en dos pestañas: configuración y resultados. La primera de estas pestañas permite al usuario la configuración total de la simulación, mientras que la segunda parte permanecerá inactiva hasta que termine de ejecutarse la simulación y mostrara al usuario todos los resultados obtenidos.

Al iniciar la aplicación el usuario se encuentra con la primera de las pestañas del panel principal que está diseñada para permitir la configuración de la simulación .Desde este primer panel pueden configurarse aspectos como el tiempo de simulación en microsegundos , la tecnología que se va a implementar ,así como la velocidad máxima permitida para la transferencia de datos.

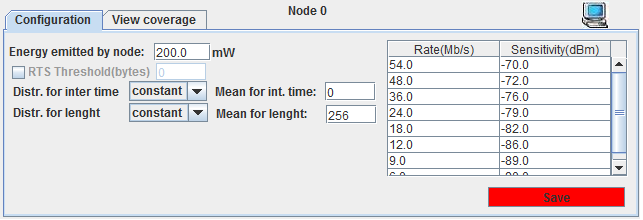


(Interfaz de usuario del simulador)

Según la tecnología escogida variará el rango de velocidades permitidas y la frecuencia empleada en las emisiones:

|  |  |  |  |
| --- | --- | --- | --- |
| **Estándar** | **Velocidades permitidas** | **Frecuencia** | **Especificación radio** |
| 802.11 | 1,2 Mbps | 2.4 Ghz | FHSS/DSSS |
| 802.11a | 6,9,12,18,24,36,48,54 Mbps | 5 Ghz | OFDM |
| 802.11b | 1,2,5.5,11 Mbps | 2.4 Ghz | DSSS |
| 802.11g | 6,9,12,18,24,36,48,54 Mbps | 2.4 Ghz | OFDM/DSSS |
| 802.11n | 1,6,11,54,108,130,150 Mbps | 2.4 Ghz / 5 Ghz | OFDM/MIMO |

Más abajo, en esta misma pestaña, podrá introducir los nodos (que representan dispositivos con tarjetas de red inalámbricas) y puntos de acceso wifi. Cada uno de estos dispositivos podrá ser configurado de manera independiente .



(Panel para la configuración de un nodo).

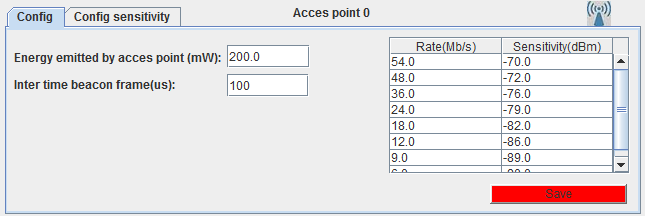
Los nodos permiten la configuración de los tiempo de creación de tramas así como el tamaño variable de las mismas .Estos dos tipos de valores vendrán dados por tres tipos de funciones: constante, uniforme y exponencial (centradas en un valor medio configurable también desde la interfaz):

-*Constante* : El tamaño de trama o los tiempos de creación de las misma permanecerán constantes e iguales al valor principal.

-*Uniforme* : El tamaño de trama será un valor escogido de manera aleatoria entre 256 y 2\*(valor principal)-1 , y el resultado supera el valor 2.312(tamaño máximo en bytes permito para una trama) devolverá ese mismo valor. Para la generación del tiempo de creación entre tramas , los extremos a la hora de escoger el valor aleatorio son 1 y 2\*(valor principal)-1.

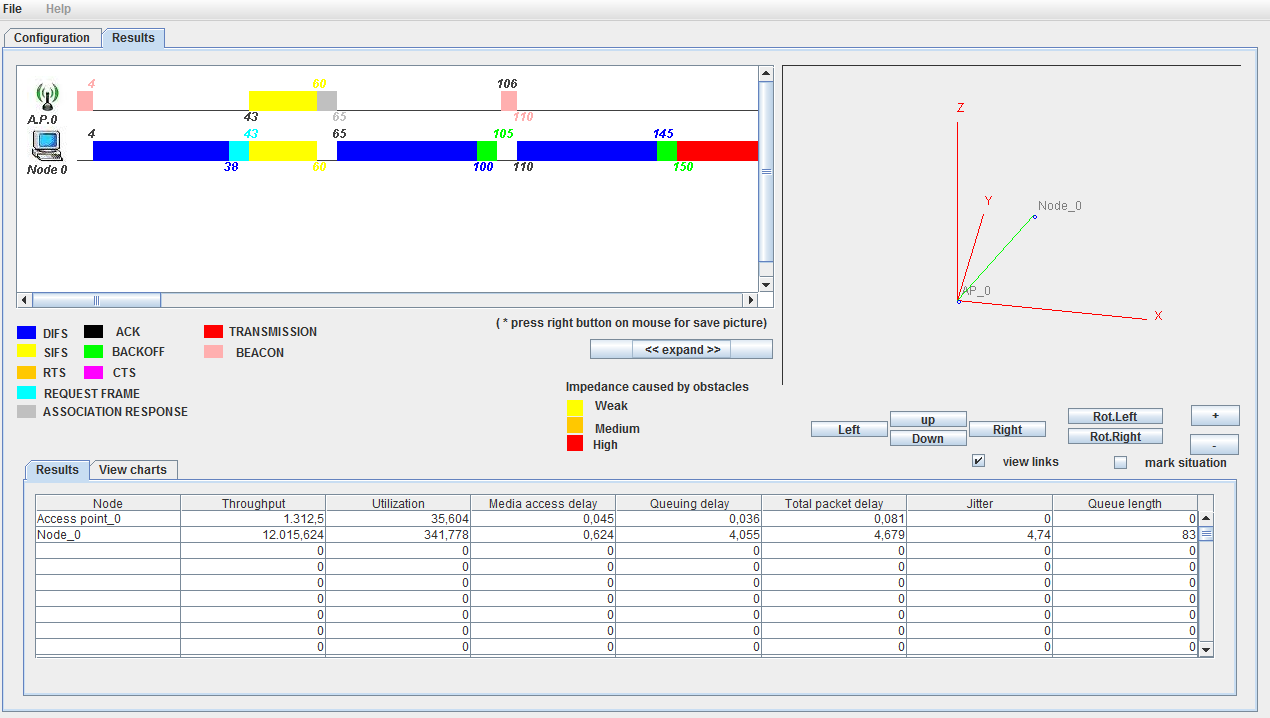
-*Exponencial*: En este caso se toma un valor aleatorio comprendido entre cero y uno , y después se hace la resta 1-(random(0,1)).Para el tamaño de las tramas si el valor sale fuera del rango [256, 2.312] , tomará el valor del extremo mas cercano.

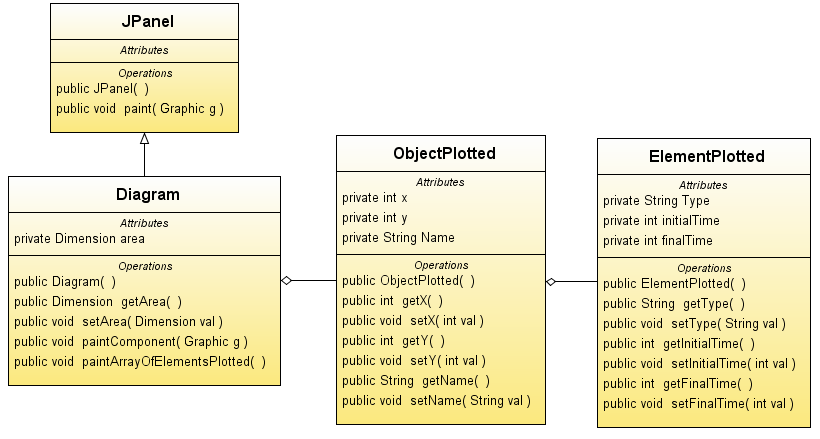
Los puntos de acceso permitirán configurar el tiempo de creación de las tramas de baliza ( beacon frames ) y ambos tipos de dispositivos permitirán configurar , sensibilidades de señal al medio (dependientes del tipo de tecnología escogida:802.11,802.11a…) y potencias empleadas para le emisión de sus señales .Para cada dispositivo el usuario tendrá la opción de poder ver una grafica donde se describe la potencia de la señal en función de la distancia dependiendo de la configuración de la potencia emitida definida por el usuario.



(Panel para la configuración de un punto de acceso).

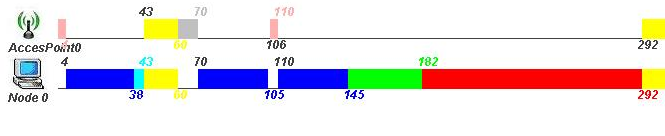
En la segunda de las pestañas principales ,el usuario podrá recoger el resultado de la simulación realizada, además podrá ver toda la traza resultante de la ejecución del algoritmo CSMA/CA entre los dispositivos involucrados así como ver la distribución en un espacio tridimensional y las asociaciones resultantes de la simulación de los mismos ,y una tabla que recoge para cada dispositivo una serie de medidas que se explican en el apartado 3.3.6. A continuación se explica como ha sido construido cada uno de estos componentes.

 (Ejemplo de resultados obtenidos tras una simulación)

** 3.3.2.-Construcción del diagrama de secuencia:**

Para su construcción he utilizado las siguientes clases:

Cada nodo o punto de acceso implementa una lista de objetos de la clase ”ElementPlotted“. Cada objeto de esta clase tiene un atributo que especifica de que tipo es y representa el estado por el que está pasando el dispositivo (ap o nodo) en cada microsegundo es por ello que cada nodo y punto de acceso implementa una lista de ElementPlotted la cual contiene un eslabón por cada tarea ejecutada por el dispositivo. Los distintos tipos que pueden tomar los objetos de esta clase son los siguientes: inactive\_wait , spend\_dif , spend\_sif , transmiting, transmiting\_ACK, transmiting\_RTS, transmiting\_CTS, spend\_backoff, transmiting\_BeaconFrame , transmiting\_RespFrame, transmiting\_RequestFrame.



(Ejemplo gráfico del resultado de una ejecución)

**3.3.3.-Construcción del diagrama tridimensional:**

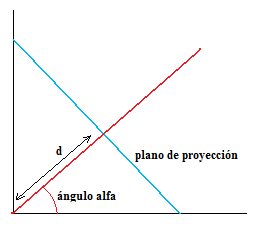
Para la representación tridimensional de la posición de los nodos y los puntos de acceso en el espacio decidí utilizar la Api 3D de java pero descarte esta idea porque debido al coste computacional que suponía ralentizaba mucho el programa. La opción elegida para ello ha sido una representación isométrica sobre un plano .

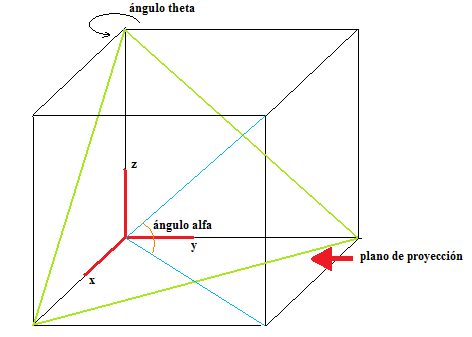
Una proyección isométrica de un objeto tridimensional es la proyección de dicho objeto sobre un plano que se encuentra a una cierta distancia y con cierto grado de inclinación. Las operaciones realizadas para situar cada punto del espacio en el plano han sido las siguientes:

X = (d\***Radio**\*Cos(Theta+Ang))/(d+f+**Radio**\*Sen(Theta+Ang)\*Cos(Alfa)-p.z\*amp\*Sen(**Alfa**));

Y = (d\*p.z\*amp\*Cos(Alfa)+d\*Radio\*Sen(Theta+Ang)\*Sen(Alfa))/(d+f+Radio\*Sen(Theta+Ang)\*Cos(Alfa)-p.z\*amp\*Sen(Alfa));

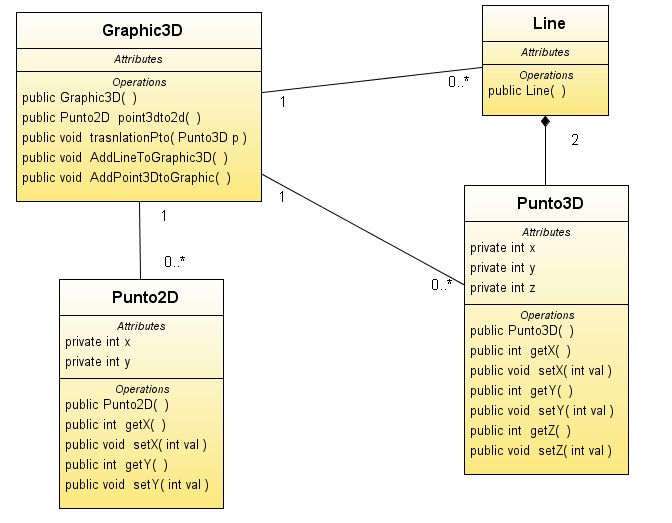
Donde **Radio** es la distancia desde la coordenada Y a la coordenada X del punto p(x,y,z), **Theta** es el ángulo formado entre el plano de proyección y el eje X ,**Ang** es el ángulo entre el punto **p** y el plano xy , y **Alfa** es el ángulo de inclinación del plano de proyección y el plano XY.





Para pasar de un espacio a otro necesitamos realizar transformaciones de coordenadas cartesianas . Estas operaciones son realizadas en la función “pto3Da2D” de la clase Graphic3D.

Para su construcción he utilizado las siguientes clases:



**3.3.4.-Construcción de la tabla de resultados:**

En ella se recogen los resultados recogidos después de una simulación .

## Throughput (kbits/s):

Es la cantidad de datos transferidos de un lugar a otro, o la cantidad de datos procesados en un determinado espacio de tiempo . El throughput puede ser traducido como la tasa de transferencia efectiva de un sistema.

El cálculo para obtener este dato será el siguiente:

## Utilización (us):

## Retardo de acceso medio (us):

Tiempo medio que el nodo ha tenido ocupado el canal.

(\*Durante la simulación cada nodo lleva la cuenta del tiempo que qué tiene ocupado el canal)

## Retardo en cola (us):

Es la suma de los tiempos de retardo en cola de cada trama que cada nodo ha enviado. El tiempo de retardo en cola de cada nodo es el tiempo desde que es creada la trama hasta que consigue enviarse.

## Retardo total de tramas(us):

Es la suma total de los tiempos de retardo en cola más retardo de acceso al medio de las tramas que un nodo ha logrado transmitir con éxito.

## Jitter (us):

El jitter se define como la variación del tiempo de tránsito de las tramas y es calculado de la siguiente manera:

Donde es el instante en el que la trama j fue creada , es el instante en el que la trama j fue enviada , n es el número total de tramas enviadas con éxito y tDel es retardo total de trama calculado previamente.

## Longitud de cola:

Número de tramas restantes en la cola de envío de cada nodo una vez acabada la simulación.

**3.3.5.- Empleo de obstáculos:**

Una vez añadidos los dispositivos que intervendrán en la simulación , tendremos la posibilidad de añadir obstáculos entre ellos para simular la atenuación de la señal a través de dicho obstáculo. Si el usuario lo necesita podrá añadir nuevos materiales al programa especificando además la atenuación (en decibelios) que provocarán por absorción y dispersión de la señal.

Las ondas de radio, de cualquier clase, son atenuadas o debilitadas mediante la transferencia de energía al medio en el cual viajan cuando éste no es el vacío . La potencia de la onda decrece exponencialmente en el medio, correspondiendo a un decrecimiento lineal en dB .Para ver como se han realizado estos cálculos vaya al punto 4.1.1.

**¿Cómo afecta la presencia de obstáculos a las redes wireless?**

Debido a la naturaleza de la tecnología de radio, las señales de radio frecuencia pueden desvanecerse o bloquearse por materiales medioambientales. El motivo es que las ondas de radio no recorren la misma distancia en todas las direcciones. Las paredes, las puertas, los huecos de ascensores, las personas y otros obstáculos suponen distintos grados de pérdida de señal (atenuación) que provocan que el patrón de la radiación de radiofrecuencia (RF) sea irregular e imprevisible. Por ejemplo, una pared de una oficina que cambie la propagación de una señal de RF de un nivel de potencia de 200 mili vatios (entrada) a 100 milivatios (salida) se traduce en una atenuación de 3 dB.

En general, hay una fuerte absorción en los materiales conductores, sobre todo en metales. Otro material absorbente para las ondas, en las frecuencias relevantes a las redes inalámbricas,(rango de microondas) es el agua en todas sus formas (lluvia, neblina, y la contenida en el cuerpo humano) , es por ello que si se desea simular alguno de estos factores se les debe atribuir un valor muy bajo en decibelios de entre -60 y -90 dB .Encontramos absorción intermedia en rocas, ladrillos y concreto, dependiendo de la composición de los materiales. Lo mismo funciona para la madera, árboles y otros materiales; su comportamiento es fundamentalmente determinado por su concentración de agua. Este tipo de materiales de absorción intermedia rondara una atenuación de -15 dB

A continuación se muestra una tabla detallada de materiales y como de importante es la atenuación que se produce. Esta tabla podrá ser utilizada como guía al usuario de la aplicación a la hora de crear nuevos obstáculos:

|  |  |  |
| --- | --- | --- |
| **Material** | **Ejemplo** | **Atenuación** |
| Madera | Tabiques | Baja |
| Vidrio | Ventanas | Baja |
| Amianto | Techos | Baja |
| Yeso | Paredes interiores | Baja |
| Ladrillo | Paredes interiores y exteriores | Media |
| Hojas | Árboles y plantas | Media |
| Agua | Lluvia / Niebla | Alta |
| Cerámica | Tejas | Alta |
| Papel | Rollos de papel | Alta |
| Vidrio con alto contenido en plomo | Ventanas | Alta |
| Metal | Vigas, armarios | Muy alta |

No solo los obstáculos físicos afectan a la transmisión, también existen obstáculos no físicos, como otras ondas pertenecientes a otras redes o a otros dispositivos, como microondas, Bluetooth o algunos teléfonos DECT inalámbricos. Esto es debido a que las redes inalámbricas operan en el espectro de frecuencias de estas tecnologías, y por tanto pueden existir interferencias que atenúen nuestra señal, afectando negativamente al rendimiento. (Debido a la dificulta en el cálculo que implica , este tipo de interferencias no han sido tenidas en cuentas en el programa simulador).

**3.3.6.-Implementacion del algoritmo CSMA/CA:**

En esta sección se describirá el funcionamiento de el algoritmo que coordina el acceso al medio compartido , que es el canal inalámbrico, por parte de los nodos y los puntos de acceso , y se explica como ha sido abordada su implementación.

**Algoritmo CSMA/CA:**

Una vez que una estación inalámbrica se asocie a un AP, puede comenzar a enviar y recibir las tramas hacia y desde el punto de acceso .Pero dado que múltiples estaciones pueden desear transmitir tramas de datos en el mismo tiempo y en el mismo canal , un protocolo múltiple de acceso es necesario para coordinar las transmisiones.El algoritmo básico de acceso a este nivel es llamado CSMA/CA (Carier Sense Multiple Access / Collision Avoidance), que significa que cada estación monitorea el canal antes de transmitir , y se detiene cuando detecta el canal ocupado .Este protocolo funciona tal y como se describe a continuación:

1.-Si un nodo detecta inicialmente que el canal está libre , transmite su trama después de un periodo de tiempo conocido como DIFS (Distributed Inter-frame Spacing).

2.-En caso contrario , la estación elige un valor al azar de backoff (tiempo de espera) y cuenta de regreso de este valor cuando el canal está libre.Cuando el canal es detectado como ocupado , el valor del contador continuara como congelado.

3.-Cuando el contador alcanza cero(nota que esto puede ocurrir solamente cuando el canal es detectado como libre), las estaciones transmiten la trama entera y después esperan un reconocimiento de la misma.

4.-Si se recibe un reconocimiento (trama ACK) ,la estación que transmite sabe que su trama se ha recibido correctamente en la estación de destino. Si la estación tiene otra trama a enviar, comienza el protocolo CSMA/CA en el paso 2.Si el reconocimiento no se recibe, la estación que transmite entra la fase del backoff de nueve en el apso 2, con el valor al azar elegido en un intervalo mas grande.

El protocolo MAC del 802.11 también incluye de manera opcional un esquema efectivo de reserva para evitar el problema del nodo oculto. Dicha solución permite que una estación utilice una trama de control llamada RTS (Request-To-Send) y otra trama de control llamada CTS ( Clear-To-Send )para reservar el acceso al canal. Siguiendo el protocolo CSMA/CA especificado el comportamiento de un nodo durante la simulación seria el reflejado en la siguiente figura:

Diagrama de ejecución de un nodo:

Espera la recepción de un beacon frame

Carga en su buffer de salida una trama de solicitud de conexión (request frame) .

¿Recibe beacon frame?

Comprueba si el medio esta libre

Consume un tiempo DIF

¿Medio ocupado ?

Consume tiempo de backoff almacenado

Amplia la ventana de backoff

Espera a que el medio esté desocupado

Envía trama almacenada en buffer

¿Medio ocupado ?

Consume un tiempo SIF

Consume un tiempo denominado time-out antes de dar por descartada la trama:

-ResponseFrame: en el caso de haber mandado una solicitud de conexión

-CTS: en el caso de haber mandado una trama RTS

-ACK: en el caso de haber mandado una trama de datos.

No

No

No

Si

Si

Si

Como puede observarse en el diagrama de ejecución de un estación wireless la primera tarea es la de conectarse al punto de acceso más cercano. Este proceso de conexión ha sido diseñado de forma pasiva, es decir, será el nodo el que escuche al medio en busca de una baliza de señalización transmitida por un punto de acceso. En la realidad un nodo o dispositivo inalámbrico, hace un barrido por todos los canales (frecuencias) en busca de tramas beacon , este proceso y los tiempos de barrido por cada frecuencia no están especificados en el estándar 802.11 y es propio de cada empresa. Para simularlo cada dispositivo tendrá asociado un objeto “cannel” el cual tendrá funciones asociadas de escritura y lectura y será utilizado como buffer de escritura y lectura de datos.

Cuando un punto de acceso desee mandar alguna trama a un nodo solo deberá escribir en su objeto “Channel” asociado.

Node channel

AP channel

D

Node channel

Node channel

*(Cada nodo tendrá una lista de identificadores de canal , uno por cada nodo que haya logrado asociarse a el.)*

De esta forma ,los nodos conectados a un mismo punto de acceso mandarán sus tramas al objeto canal asociado al punto de acceso y el punto de acceso mandara sus tramas a los canales de los nodos asociados .

A continuación se explica de manera mas detallada el funcionamiento de cada nodo para simular protocolo CSMA/CA:

Si un nodo recibe un “beacon frame “ , lo analiza es decir debe comprobar que la señal recibida tenga una potencia superior a la menor de las sensibilidades en las que está configurado , también deberá recoger el canal por el que se va a transmitir (será el identificador de AP),y acto seguido deberá mandarle una trama de solicitud de conexión “request frame”. (Como puede verse en el diagrama de clases del paquete “ElementForSim” , tanto la potencia de la señal con la que es enviada cada trama como la velocidad a la que ha sido enviada , son parámetros de cada objeto “Frame”).Si pasado un tiempo “time\_out” no llega la respuesta del AP , el nodo descarta la conexión , en caso contrario recibirá una trama de tipo “association response” especificando al nodo la velocidad de transmisión a la que mandar los datos .Esta velocidad ha sido calculada en base a la sensibilidad del punto de acceso y a la potencia recibida por parte de este nodo..A partir de entonces el nodo queda enlazado a dicho punto de acceso.

Un nodo detecta el medio como desocupado si al testear el canal asociado al punto de acceso después de haber contado un tiempo DIFS, este le contesta que está ocupado en el caso de que otro nodo haya escrito en él , y este esté bajo la misma cobertura del nodo que hizo la solicitud .Esto significa que el nodo a accedido al medio y ha detectado que otro nodo estaba ocupando el medio .En el caso en el que el nodo que esté ocupando el canal no estuviera bajo la misma cobertura del nodo que quiere ocuparlo , el objeto canal asociado al AP , avisaría a dicho nodo de que el medio estaría desocupado , simulando así el problema de el nodo oculto.

Es interesante explicar cómo simular colisiones con el sistema de “objetos canales” diseñado para la simulación. Pongamos el ejemplo de dos nodos conectados a un mismo punto de acceso .Si ambos nodos ven el canal del AP libre en el mismo instante, ambos tendrán permiso para mandar datos en el mismo momento, pero al ser la simulación un proceso secuencial, uno de ellos pondrá el canal como ocupado antes que el otro y el nodo que se ejecute más tarde lo verá como ocupado y no transmitirá. Para poder simular la colisión, es decir el acceso simultaneo de ambos al medio , el objeto canal tiene asociada una variable que controla el numero de dispositivos que están escribiendo a la vez en él , y una variable booleana adicional que indica el estado corrupto de los datos si en algún instante se ha llegado a contabilizar más de un dispositivo escribiendo en el mismo instante.

Para simular el problema del nodo oculto se dota al objeto canal de una lista de índices de dispositivos .Si un nodo ejecuta una operación de comprobación de estado de canal, este devolverá libre si está siendo ocupado por otro nodo que no se encuentra bajo la cobertura del nodo que ha hecho la solicitud. Para explicar mejor como se realiza la tarea de comprobar cuando uno nodo esta bajo la cobertura de otro , se muestra a continuación un extracto de código de la función “deviceAunderCoverageOfDeviceB” , de la clase WirelessChannel .

//1.-primero calculamos la potencia que llega al device A del device B

**float** energyOfDeviceB\_dB = *converterWtodB*(nodeB.getEnergyEmitedW());

ArrayList possitionNodeA = nodeA.getPossition();

ArrayList possitionNodeB = nodeB.getPossition();

**float** signalLoss\_dBm = *calculateEnergyReceivedByDevice\_A*(possitionNodeA,possitionNodeB,nodeB.getEnergyEmitedW());

**float** energyOfDeviceB\_dBm = *converterWtodBm*(nodeB.getEnergyEmitedW());

//Impedance devuele la suma en decibelios de la atenuacion povaocada por los //posibles obstaculos entre el dispositivo A y el dispositivo B.

**float** impedance = *Impedance*("Node "+nodeA.getIdNode(),"Node "+nodeB.getIdNode());

**if**(impedance>0){//restamos 30 para pasar de dB a dBm.

impedanced-=30;

}

//2.- a la pérdida de potencia debida a la distancia debemos restarle la pérdida debido a obstáculos.

**float** energyReceivedByDeviceA = energyOfDeviceB\_dB - signalLoss\_dBm-impedanced;

//3.- como el array de sensibilidades del nodo esta ordenado de mayor a menor , cogemos la menor de las sensibilidades.

ArrayList sensitivity = nodeA.getSensitivity();

**float** lowerSensitivity\_dBm = (Float) sensitivity.get(sensitivity.size()-1);

**if**(energyReceivedByDeviceA < lowerSensitivity\_dBm)**return** **false**;

**else** **return** **true**;

-Dispositivo punto de acceso:

Diagrama de ejecución de un puto de acceso

Manda tramas beacon en difusión

Espera recibir tramas

¿Recibe solicitud de conexión?

Recibe una trama

¿Consume tiempo entre tramas beacon?

¿Canal libre?

Consume un tiempo SIF

Envía trama CTS

Consume un tiempo SIF

Envía trama ACK

Consume un tiempo SIF

Envía trama

response

¿Recibe

RTS?

¿Recibe trama de datos ?

La primera tarea en ejecutar un AP será el envió en difusión de tramas de baliza (beacon frames) para avisar de sus existencia , en la simulación el AP comprobará que nodos están a su alcance es decir que nodos tienen una sensibilidad de recepción superior a la potencia emitida por el nodo menos la atenuación debida a la distancia y a obstáculos intermedios, una vez obtenida esta lista nodos, colocará en sus buffers de entrada (objeto canal) dicho frame.

Una vez haya enviado la trama “beacon” en difusión , pasará escuchar el medio en busca de tramas de solicitud de conexión por parte de los nodos cercanos. Al recibir una trama de solicitud de conexión comprueba que la potencia con la que ha sido envida menos la atenuación debida a la distancia y a obstáculos intermedios sea superior a la sensibilidad mínima a la que ha sido configurado. En función de la potencia recibida se establecerá una velocidad de transmisión que se le comunicara al nodo a través de la trama de aceptación de solicitud “association response”.

El Comportamiento de un AP al recibir una trama de datos es semejante al anterior descrito, solo que este tipo de tramas será contestada con una trama ACK.

En el caso de que reciba una trama RTS , contestará después de un tiempo SIF al nodo con una trama CTS , solo en el caso en el que el punto de acceso no este siendo ocupado por otro nodo .

En la siguiente tabla se muestran los tiempos para difs y sifs utilizados en cada tecnología así como la ventana de contienda inicial y el tamaño del slot utilizados al ejecutar el algoritmo de back off cada nodo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Estándar** | **DIFS** | **SIFS** | **Content wind.** | **Slot** |
| 802.11 | 34 | 16 | 16 | 9 |
| 802.11a | 34 | 16 | 16 | 9 |
| 802.11b | 50 | 10 | 32 | 20 |
| 802.11g | 30 | 10 | 32 | 20 |
| 802.11n | 34 | 16 | 34 | 9 |

Capítulo 4 :

**4.1.-Experimentación :**

**4.1.1.- Medición de la cobertura de un dispositivo:**

Llamamos cobertura a la distancia entre el dispositivo y el punto en donde la potencia de la señal de dicho dispositivo es superior a la menor de las sensibilidades de los dispositivos de la misma red , es decir es la distancia máxima permitida entra dos dispositivos para poder comunicarse.

Primero realizaremos el cálculo suponiendo unas condiciones ideales y posteriormente estimaremos unas pérdidas adicionales por falta de condiciones ideales , representadas en la simulación por los obstáculos que el usuario a seleccionado.

La pérdida de propagación es la cantidad de señal necesaria para llegar de un extremo de la conexión wireless al otro. Es decir la cantidad de señal que se pierde al atravesar un espacio. Para calcularla he utilizado la siguiente fórmula:

**Pp = 20\*log10(d/1000) + 20\*log10(f\*1000) + 32,4**

 Donde **Pp** indica la perdida de propagación en decibelios (dB), **d** es la distancia en metros y **f** es la frecuencia en GHz. EL valor de la frecuencia depende del canal en el que se tenga configurado el equipo.

También podemos resumirla como:

**Pp = 20log10(d) + 20log10(f) + 32,4**

Pero en este caso, **Pp** indica la perdida de propagación en decibelios (dB), **d** es la distancia en kilómetros y **f** es la frecuencia en MHz.

(\*)La función dedicada a este cálculo se encuentra en la clase WirelessChannel con el nombre signalLostByDistance.

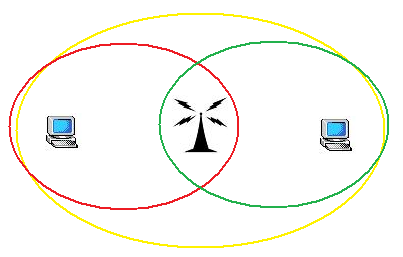
Una vez calculada la pérdida de señal debida a la dispersión por distancia tenemos que calcular la perdida debida la presencia de obstáculos , para ello simplemente restamos a **Pp**  la suma de las atenuaciones (en dB) de los obstáculos.

En mi simulador todos estos parámetros son configurables; desde la presencia de obstáculos , como la atenuación que provocarán los mismos , hasta las sensibilidades de todos los dispositivos y la potencia de señal de los mismos , el usuario puede además ver antes de la simulación una gráfica del descenso de la potencia de la señal con respecto a la distancia , y si elige la opción ver pre-simulación también podrá ver si la cobertura de alguno de los dispositivos no alcanza a otros implicados en la simulación. Esta función esta implementada en la función WirelessChannel con el nombre “deviceAunderCoverageDeviceB” , y su función será la de comprobar que el dispositivo B posea una sensibilidad superior a la potencia que le llega del dispositivo A.

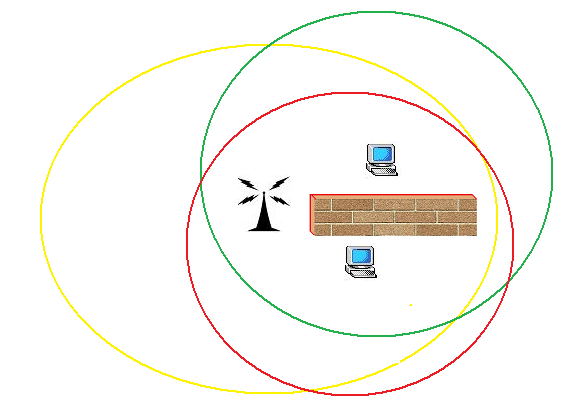
4.1.2.- Descubrimiento del problema del nodo oculto:

Este problema surge en el momento en el que uno de los nodos conectados a un punto de acceso , no es capaz de observar si los demás nodos conectados , tienen ocupado el medio . Esto se produce por dos motivos:

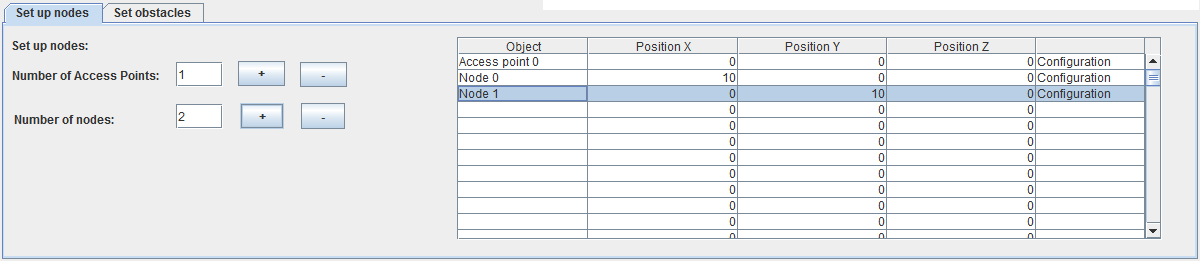
1.- Uno de los nodos no se encuentra bajo la cobertura de algún otro nodo de la red y por ello ninguno es capaces de ver cuando el otro está ocupando el medio .La siguiente imagen muestra un ejemplo.



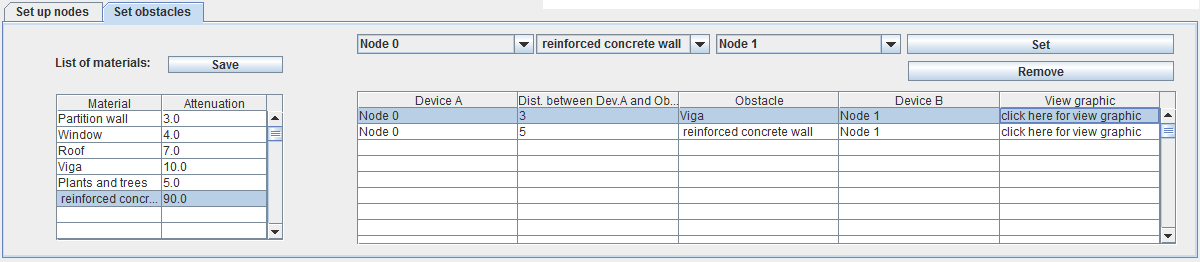
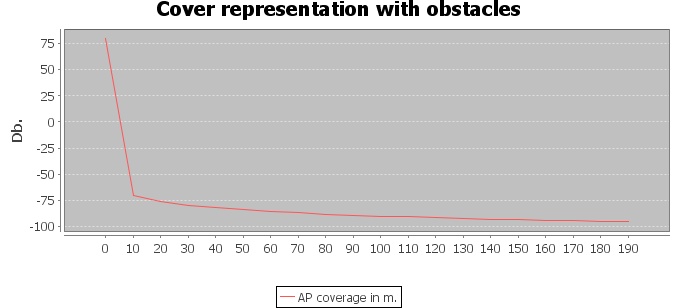
2.- Uno de los nodos no se encuentra bajo la cobertura de algún otro nodo de la red debido a la presencia de obstáculos que atenúan las señales de los nodos implicados e impiden que sean capaces de ver cuando el otro está ocupando el medio .La siguiente imagen muestra un ejemplo.



A continuación se muestra como realizar una simulación de dicho problema para ver el comportamiento de ambos nodos .Utilizaremos un punto de acceso situado en la posición (0,0,0) y dos nodos situados respectivamente en las posiciones (10,0,0) y (0,10,0).

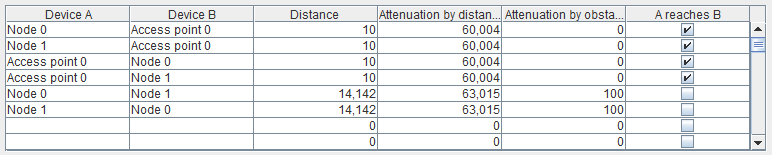


Aislaremos ambos nodos utilizando obstáculos .Con esto queremos conseguir que un nodo no pueda percibir el medio como ocupado cuando lo ocupa el otro nodo.

 Como podemos observar hemos añadido una viga que se encuentra a 3 metros del nodo 0 y un muro de hormigón a 5 metros .Si queremos ver como afecta alguno de estos obstáculos a la potencia de la señal del dispositivo del la primera columna de la tabla debemos hacer doble “click” en la quinta columna .Como ejemplo vamos a ver como afecta el muro de hormigón que hemos añadido.

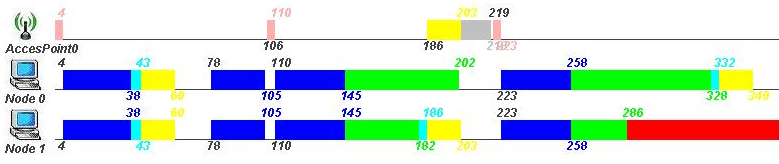
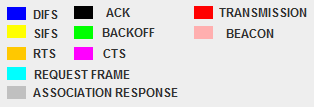
Como podemos observar se ha producido la mayor atenuación en la posición en la que se encuentra el muro , a cinco metros de la posición del nodo 0.

Ahora comprobaremos si estos obstáculos introducidos han sido suficientes para aislar ambos nodos para ello debemos lanzar una pre simulación. Obtenemos los siguientes resultados:



Como podemos observar la los nodos pueden comunicarse con el A.P. pero no pueden verse entre ellos .Esto es debido a que la atenuación de la señal de cada uno de ellos debido a al distancia que los separa menos la atenuación provocada por absorción por los obstáculos que hemos insertado, es inferior (en decibelios) a la sensibilidad (también en decibelios) de cada uno de los nodos.

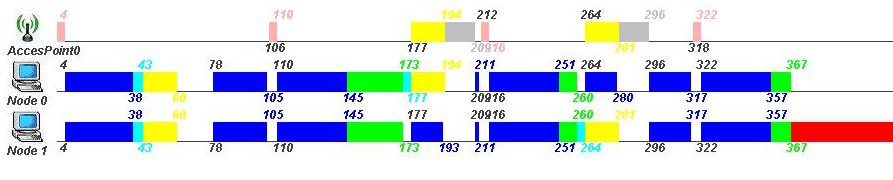
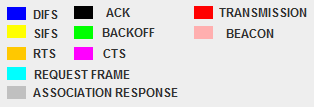
Por ultimo lanzaremos la ejecución completa de la simulación:



La primera colisión que se produce esta en el us 38.Esta colisión es normal ya que ambos nodos han empezado a ejecutarse a la vez y han observado el medio como libre en el mismo instante, esto provocará que ambas señales interfieran entre si y como consecuencia de ello el punto de acceso no reciba ningún tipo de información.

El primer síntoma del problema del nodo oculto aparece en el us 182.El nodo 1 a ocupado el medio para mandar una solicitud de conexión al punto de acceso , el nodo 2 debería haber visto el medio como ocupado y haber bloqueado su cuenta de backoff hasta percibir el medio como libre. Sin embargo vemos que esta tarea la realiza cuando el punto de acceso ocupa el medio en el us 202 ya que el punto de acceso si está en su línea de visión. Podemos ver también como en el us 285 el nodo 1 termina su cuenta de backoff y comienza a transmitir ya que en ese instante ningún otro dispositivo tiene ocupado el medio.

A continuación se muestra el diagrama de ejecución de una simulación ejecutada con los mismos parámetros pero sin obstáculos , es decir cada nodo será capaz de ver cuando el medio ha sido ocupado por otro nodo y por lo tanto no hay nodos ocultos.



Capítulo 5 :

5.1.-Conclusiones y desarrollos futuros:

Todo lo que me ha aportado este proyecto de fin de carrera ha sido muy positivo . Gracias a este proyecto he profundizado aún más en lo estudiado durante la carrera acerca de redes wireless y el estándar de comunicación 802.11 y comprender mejor su complejo funcionamiento y espero poder ayudar con mi trabajo a muchas personas que estén interesadas en comprender el funcionamiento de una tecnología en auge como es la tecnología wifi, la cual ha llegado a convertirse en algo cotidiano en nuestras vidas.

Además me ha ayudado a mejorar mi planificación para abordar un proyecto y trabajar con componentes del lenguaje de programación java que eran totalmente nuevos para mí como la librería gráfica swing y las apis jxl para tratar con hojas de cálculo y JFreeChart para mostrar y tratar gráficas.

Finalmente expondré en varios puntos posibles desarrollos futuros :

1. Añadir la posibilidad de simular redes wireless ad-hoc. Este tipo de redes son poco utilizadas y se diferencian principalmente de las redes DFC por no poseer de punto de acceso wifi.
2. Utilizar applet de java para la implementación de la interfaz gráfica para permitir la ejecución de la aplicación a través de un navegador web.

Anexo:

Manual de usuario:

1.-Ejecución:

<<Falta explicar como se ejecutara la aplicación>>

2.-Configurando el simulador:

Ejecute la aplicación y una vez que este cargada podrá visualizar la pestaña de configuración. Todos los parámetros son configurables desde esta pestaña.

* Grupo de configuración general:

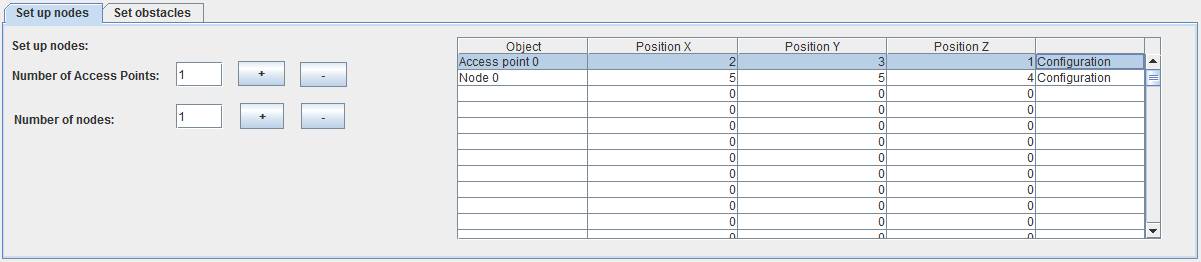
Permitirá al usuario configurar la duración de la simulación en microsegundos , elegir la tecnología con la que trabajará (802.11, 802.11a, 802.11b, 802.11g y 802.11n). Para cada tecnología tendremos una lista de posibles velocidades, aquella que seleccionemos será la velocidad máxima a la que trabajaran los dispositivos.

Nota: Ya que 802.11 puede trabajar con dos frecuencias (2.5 y 5 Ghz) si escogemos esta tecnología también podremos seleccionar con qué frecuencia trabajará.



* Grupo de configuración de dispositivos

Permitirá al usuario seleccionar el número de nodos y puntos de acceso con los que trabajará la simulación.

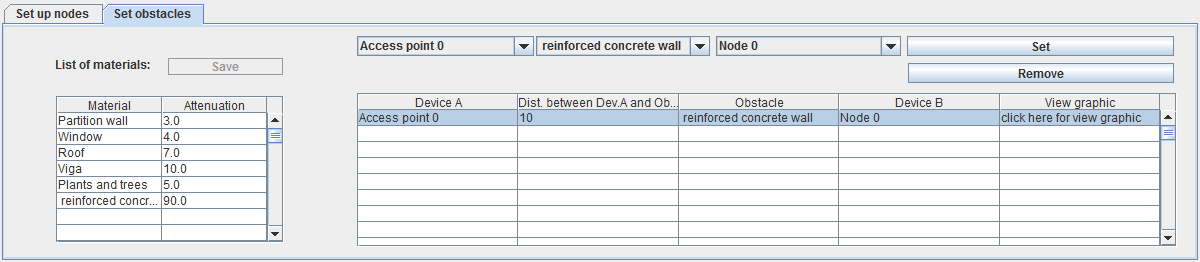


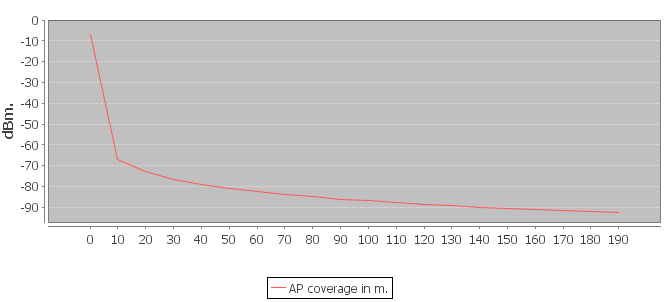
Si el usuario pulsa en la cuarta columna de la tabla podrá configurar de manera independiente cada dispositivo.

* Grupo de configuración de obstáculos:

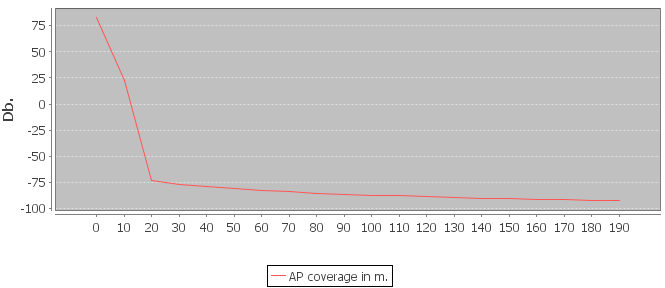
Desde esta pestaña podremos modificar y añadir obstáculos entre los dispositivos previamente insertados en la simulación. Si se desea modificar o añadir los obstáculos con los que trabajar , solo tiene que modificar la tabla que muestra la lista de obstáculos y a continuación pulsar “save”.

La segunda columna de la tabla de inserción de obstáculos , permite configurar la distancia a la que se encuentra el obstáculo del primer dispositivo .A continuación el usuario podrá ver pulsando en la quinta celda como el obstáculo variara la cobertura del primer device de la tupla y podrá observar cómo se produce la mayor atenuación a la distancia donde se indique que se encuentra el obstáculo.





Cobertura de un punto de acceso en la dirección de un obstáculo situado a 10 metros.

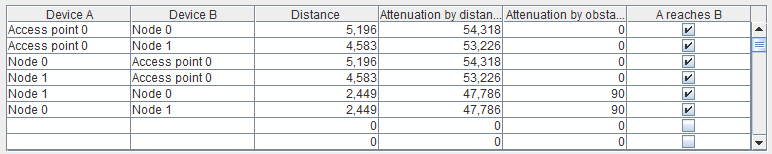


Cobertura de un punto de acceso en ausencia de obstáculos.

3.-Ejecutando una pre-simulación:

Una vez haya configurado una simulación y haya colocados los obstáculos , podrá ver antes de realizar la simulación una tabla informativa haciendo click en el botón “pre-simuulation”, que le indique la distantica entre los dispositivos añadidos , las atenuación por distancia de la señal entre ellos y las atenuaciones debidas a obstáculos con el fin de ver el alcance entre estos dispositivos y poder prever un posible problema de nodo oculto no deseado antes de ejecutar la simulación. A continuación se muestra un ejemplo:

Primero añadiremos a la simulación un punto de acceso y dos nodos en las siguientes posiciones respectivamente (0,0,0) , (3,3,3) y (1,3,4) y añadiremos como obstáculos entre ambos nodos un muro de hormigón , posteriormente lanzaremos la pre-simulación :



Las dos primeras columnas muestran los dispositivos implicados para los cuales se muestra a continuación la distancia entre ellos , la atenuación de la señal debida a la distancia , la atenuación debida a obstáculos implicados en la simulación y por ultimo se indica si la cobertura del primer dispositivo alcanza al segundo dispositivo.

4.-Ejecutando una simulación:

Una vez que la configuración de los parámetros de la simulación haya sido completada , pulse el botón “Run Simulation” para iniciar la ejecución de la simulación. Este proceso puede durar desde unos segundos a varios minutos dependiendo de la complejidad de la simulación y de la velocidad de la CPU y del tiempo de de simulación establecido .

Tras terminar el proceso de simulación se activará la pestaña “Results” , que mostrara el diagrama de ejecución de dicha simulación , un gráfico que muestra la distribución espacial de los dispositivos y sus conexiones entre ellos, además de una tabla que recoge el rendimiento de la simulación medido a través de varios parámetros.

5.-Modificación:

Si el usuario desea modificar el código para insertar sus propios algoritmos le aconsejo tener instalado el compilador para java “eclipse galileo” con el plugin visual editor para facilitar la modificación de los componentes de java swing (estos pueden ser ventanas , botones, paneles…etc..) .

Ya que este proyecto también trabaja con las apis jFreeChart y jExcelApi también deberá tener estas apis agregadas al proyecto. A continuación le indico dos direcciones donde podrá descargarlas de manera gratuita:

<http://www.jfree.org/jfreechart/>

<http://jexcelapi.sourceforge.net/>

Bibliografía:

* **REDES WIRELESS 802.11 (ANAYA MULTIMEDIA O REILLY)**  
  de GAST, MATTHEW.
* IEEE Standard for Information technology- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.
* <http://www.wi-fiplanet.com>.
* Apuntes de la asignatura de Redes de cuarto curso de ingeniería superior informática.