6. Herramientas de Simulación para redes Ad-hoc

Las herramientas de simulación son un elemento clave en el diseño de redes Ad-hoc, debido a que éstas por naturaleza están compuestas por un gran número de nodos, lo que implica un coste de implantación elevado. Es por ello que las herramientas de simulación son fundamentales cuando se quiere proponer nuevos diseños. Es necesario comprobar la idoneidad de una nueva composición de red antes de ponerla en práctica en escenarios reales **¡Error! No se encuentra el origen de la referencia.**.

Las redes de sensores inalámbricas (WSN) pueden ser consideradas un ejemplo particular de de redes Ad-hoc (MANET). Ambas redes comparten problemas parecidos, ambas redes están compuestas por un gran número de nodos, estos nodos pueden ser móviles o estáticos. El consumo energético es un desafío en ambas redes, debido a que se desea que los nodos tengan una larga autonomía.

Hay dos elementos claves que deben ser evaluados a la hora de elegir una herramienta de simulación:

- Comprobar que los modelos utilizados son correctos.
- La idoneidad de la herramienta de simulación para implementar los modelos.

6.1. Modelado

El siguiente esquema, Figura 1 representa el modelo general de una herramienta de simulación de redes.

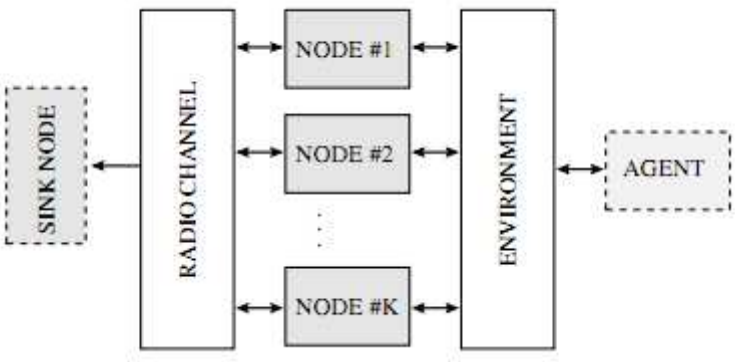

Figura 1: Modelo de simulación de redes de sensores.

Se consideran los siguientes componentes:

- Nodos, cada nodo es un dispositivo físico que mide variables físicas del entorno que le rodea. Cada nodo es capaz de comunicar con sus nodos vecinos. Cada nodo internamente implementa una pila de protocolos.

- Entorno, la principal diferencia con respecto a las redes convencionales es el hecho del modelado del entorno que rodea cada nodo. Además, los nodos medirán alguna variable característica de dicho entorno.
- Radio canal, este componente caracteriza la propagación de la señales a través del medio.
- Nodos “sink”, son los encargados de recolectar toda la información medida por las redes de sensores. Este tipo de nodos normalmente tiene una mayor capacidad de procesamiento, y suelen estar conectados a un nivel superior de red, como puede ser internet.
- Agentes, generador de eventos de interés. Es decir, la herramienta de simulación debe generar los eventos de interés para los nodos. Este componente es útil cuando su comportamiento puede ser implementado de manera independiente al entorno.

En la siguiente Figura 2 se representa un posible modelo de un nodo en una red de sensores. Se pueden destacar los siguientes componentes:

- El stack de protocolos implementados en el nodo, normalmente este stack está compuesto por un protocolo MAC, un protocolo de enrutamiento y protocolo especifico de aplicación.
- Physical-node representa la plataforma hardware. Modela las características físicas de los nodos en la red de sensores. Es importante el modelo energético implementado en el modelo, ya que como se comentó anteriormente el consumo de energía es un parámetro fundamental en redes ad-hoc. Otros aspectos de los nodos también son tenidos en cuenta en este componente, como son, movilidad, posiciones, etc.
- El componente medio, los nodos con el mundo real que les rodea. Los nodos están conectados al mundo real a través de un canal radio, o a través de canales físicos.

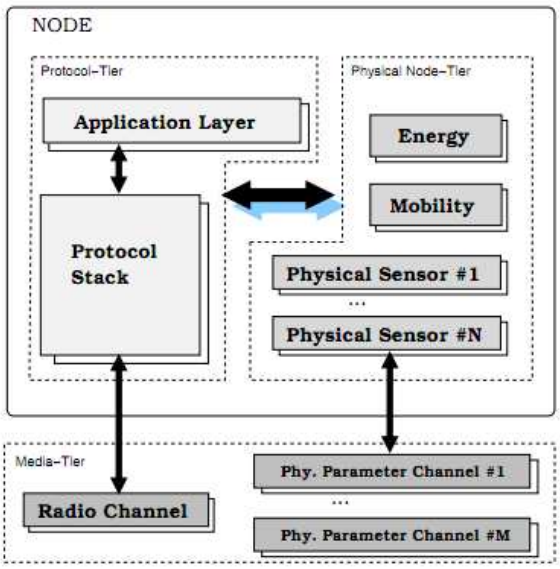

Figura 2: Modelo de los nodos.

6.2. Principales paquetes de simulación

A continuación se van a describir las principales herramientas de simulación utilizadas en el ámbito de la investigación en redes ad-hoc.

NS-2

Es un simulador de eventos desarrollado en C++**¡Error! No se encuentra el origen de la referencia.**. Es uno de los simuladores más populares en el ámbito académico. Soporta un gran abanico de protocolos en todas las capas del modelo OSI. NS-2 usa el lenguaje de programación de scripts Otcl como interface de configuración. Proporciona el mayor soporte en protocolos de comunicación de todos los paquetes no comerciales del mercado. Una de las desventajas de NS-2 es su pobre interfaz gráfica comparada con otros simuladores.

OMNET++

Simulador modular implementado en C++ **¡Error! No se encuentra el origen de la referencia.**. Esta herramienta de simulación está tomando protagonismo en los últimos años, debido a su fácil interfaz gráfica, y a que proporciona una poderosa GUI para animación y depuración. Su mayor desventaja es la carencia de protocolos disponibles en sus librerías. Esto implica que el usuario debe desarrollar los protocolos con el esfuerzo que esto supone. Sin embargo en los últimos años esta carencia se está mejorando gracias a las contribuciones de los desarrolladores. Existen numerosos framworks para aplicaciones específicas.

J-Sim

Este simulador está completamente desarrollado en Java. La mayor ventaja que proporciona J-Sim es la gran cantidad de protocolos soportados, incluyendo un framework para WSN. Los modelos en J-Sim son reusables e intercambiables ofreciendo un amplia flexibilidad.

OPNET

Herramienta comercial de simulación de redes. Proporciona un entorno virtual de red que modela el comportamiento de una red por completo, incluyendo sus pasarelas (routers), conmutadores (switches), protocolos, servidores y aplicaciones en red. Esta herramienta es ampliamente usada en ambientes profesionales de diseño de redes. Existen versiones gratuitas para estudiantes, como la versión GURU.

Otros Simuladores

A continuación se listan otras herramientas de simulación, que si bien son menos usadas que las anteriores, también hay trabajos desarrollados usando estos paquetes.

- NCTUns, **¡Error! No se encuentra el origen de la referencia.** simulador de eventos discretos cuyo núcleo está embebido en el kernel de una máquina UNIX.
- GloMoSim, **¡Error! No se encuentra el origen de la referencia.** simulador para redes inalámbricas escrito en Parsec (lenguaje de programación derivado del c). La principal ventaja que aporta es la ejecución de procesos en paralelo.

6.3. Elección de la herramienta de simulación

Durante la primera fase de este trabajo se evaluaron diferentes herramientas de simulación de redes antes de ir decididamente a por una en concreto. En primer lugar se evaluó OPNET, en concreto la versión de estudiantes disponible en la Universidad John Moores de Liverpool. El principal problema que se encontró fue que OPNET no permite comunicaciones multicast. Esto quiere decir, que en los escenarios simulados con OPNET todos los nodos comunican con todos los nodos. Además el tiempo requerido por la simulación es muy elevado, llevándose horas para simular algunos escenarios.

Una vez descartado OPNET, se comenzó a evaluar simuladores open software, el primero en evaluarse fue OMNET++. La interfaz de usuario de este simulador es bastante amigable, pero como ya se ha mencionado anteriormente, este simulador tiene una carencia bastante importe la cual es que no contiene protocolos de comunicación incluidos en el paquete inicial. Esto implica un esfuerzo de desarrollo que no se consideró oportuno realizar en primera instancia, debido a que no se disponía de suficiente tiempo.

Finalmente el simulador elegido es NS-2. Este simulador requiere un esfuerzo inicial ya que la composición de los escenarios se debe realizar en OTcl, lo implica un periodo de

aprendizaje. Pero la mayor ventaja que proporciona NS-2 es que contiene diversos protocolos de enrutamiento, así como modelos de propagación en su paquete básico, lo que hace que rápidamente se puedan obtener resultados.

6.4. Entorno de Simulación NS-2

NS-2 es un simulador orientado a objetos desarrollado como parte del proyecto VINT en la Universidad de California en Berkeley. Este proyecto fue financiado por DARPA en colaboración con XEROX Palo Alto Research Center (PARC) y Lawrence Berkely National Laboratory (LBNL).

NS-2 es ampliamente usado por la comunidad investigadora en el contexto de networking. El simulador es orientado a eventos y funciona off line.

El simulador consiste en un núcleo de C++, donde se define tanto el funcionamiento del simulador, como se describen los protocolos de comunicaciones disponibles en el simulador. Existen protocolos definidos para todos los niveles del modelo OSI. El núcleo C++ hace uso de clases de Otcl, éste se usa para definir los parámetros de los escenarios que se desean simular. Es decir, características tales como topología, movilidad, tráfico, etc. El motivo de usar dos lenguajes de programación distintos es que C++ es un lenguaje compilado, es cual es rápido de ejecutar una vez compilado, pero es lento para realizar modificaciones. Las características de Otcl son totalmente opuestas.

En la siguiente Figura 3 se muestra el flujo de simulación de NS-2.

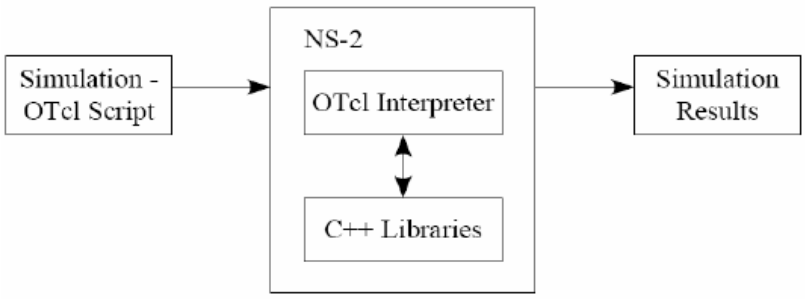

Figura 3: Flujo de simulación en NS-2.

El estándar 802.11, conocido comercialmente como WIFI, está implementado en NS-2. Otros nuevos protocolos como 802.11p (Wave), Zigbee, y WiMAX, están siendo implementados en NS-2 como parte de nuevos proyectos de investigación.

En el siguiente esquema, Figura 4 se muestra como está estructurado el proceso de diseño en NS-2.

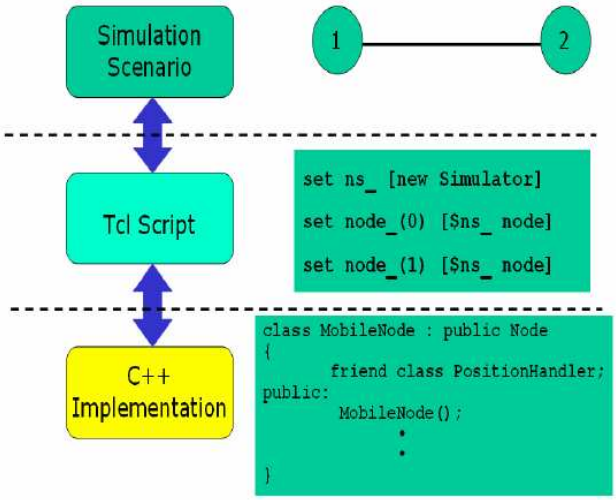

Figura 4: Estructura de capas de NS-2 desde el punto de vista del diseñador.

El escenario de simulación se describe usando un script escrito en Tcl. Este script hace uso de las clases implementadas en C++.

NS-2 suministra varios tipos de salidas. En primer lugar, NS-2 proporciona un archivo de trazas, el cual recopila toda la información de los sucesos producidos en la red durante el tiempo de simulación. Este archivo de trazas deber ser procesado para poder separar la información de interés según los resultas que se quieran obtener de la red. Este archivo de trazas lo define el usuario en el script Otcl. En cuanto a la visualización de las simulaciones existe la posibilidad de definir un archivo de trazas para ser usado por la herramienta de visualización Nam, la cual será descrita en el siguiente apartado.

Los componentes que se necesitan describir en NS-2 son:

- La apariencia de la red. Esto conlleva la posición de todos los nodos, descrita bajo coordenadas (x,y,z). El movimiento de los nodos, así como el tiempo en el que se producen estos movimientos. Velocidad de movimiento y dirección.
- Tráfico de los nodos. Definición de los nodos que actúan como fuentes y como destinos. Definición de las conexiones.
- Otros parámetros relacionados con el desarrollo de la simulación, como por ejemplo el tiempo de simulación, o definición del archivo de trazas.
- NS-2 requiere una curva de aprendizaje debido a los numerosos factores que intervienen conjuntamente, pero una vez que se adquiere cierta destreza en el uso del simulador, los resultados pueden ser muy satisfactorios.
- Existen numerosos manuales y tutoriales de NS-2 disponibles en la red. En este trabajo se quiere destacar el siguiente manual **¡Error! No se encuentra el origen de la referencia.**, el cual proporciona una descripción detallada de las principales características de NS-2, así como una gran cantidad de ejemplos de aplicación, los cuales permiten adquirir práctica en el uso del simulador y al mismo tiempo

obtener resultados. Además este manual proporciona ciertas nociones básicas y ejemplos para el uso de lenguajes de procesamiento de texto. Estos lenguajes son muy útiles cuando se quiere procesa el archivo de trazas generado por NS-2.

6.4.1. Visualización de las simulaciones: Nam

Una vez se ha definido el script donde definimos nuestro escenario de simulación, se puede correr una simulación gráfica usando el simulador gráfico Nam "Network Animator" que incorpora NS-2. Con esta herramienta se puede observar la transmisión de los paquetes y todos los eventos relacionados con estas transmisiones, como puede ser la perdida de paquetes, etc. Además NS-2 permite aplicaciones móviles. Estos movimientos se pueden visualizar utilizando Nam.

La herramienta Nam utiliza un archivo de traza específico el cual es generado por NS-2. Este archivo de trazas es definido en el script.

La siguiente imagen, Figura 5 es una captura de pantalla de una de las simulaciones llevadas a cabo en este estudio. En esta simulación se puede observar un nodo móvil denominado tren, el cual se está aproximando a dos balizas estáticas, denominadas beacon 1 y beacon 2. Este escenario corresponde al escenario inicial, el cual se detallará en el capítulo de resultados de simulación de la aplicación objetivo.

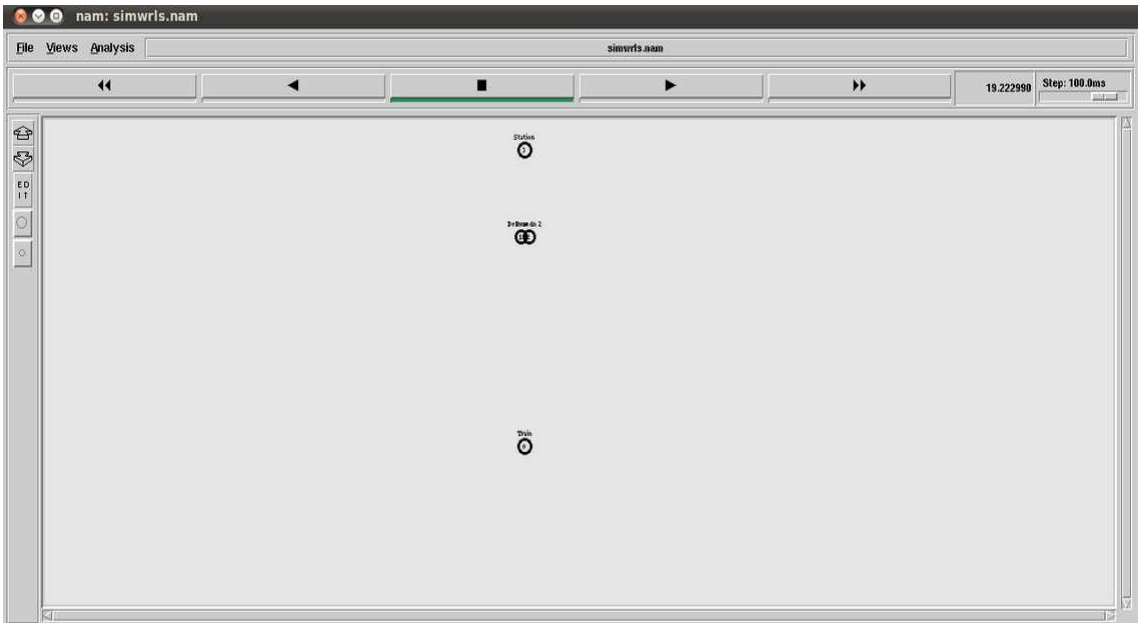

Figura 5: Network animator.

La herramienta de visualización permite parar y continuar la simulación cuando se deseé. Además todos los eventos de transmisión y recepción de paquetes son visualizados mediante líneas que entran y salen de los nodos. Sin ser una herramienta

gráfica muy potente, ésta nos permite obtener una idea de los eventos ocurridos durante el tiempo de simulación.

6.4.2. Modelo Energético en NS-2

NS-2 contiene un modelo energético que puede ser activado. Este modelo energético es activado en el script Otcl, donde se modelan todos los componentes de la red.

Cuando el modelo energético es activado, todos los nodos tienen prefijados un nivel de energía, el cual se va consumiendo conforme los nodos realizan diversas tareas en la red, como son el envío o la recepción de paquetes. El consumo de los nodos en estas tareas puede ser configurado por el usuario.

Una vez que el nivel energético de un nodo llega al mínimo, este nodo se vuelve inservible en la red.

Cuando el modelo energético está activado y se utiliza la herramienta de visualización Nam, se puede observar como los nodos cambian de color conforme la energía de éstos se va agotando. Además toda la información relativa al consumo energético en cada evento se refleja en el archivo de trazas generado por NS-2.

En el archivo de trazas se puede observar el consumo asociado a cada evento.

6.5. Arquitectura de simulación

Como ya se ha comentado anteriormente el entorno de simulación utilizado en este estudio es NS-2. Esta herramienta de simulación te permite obtener resultados de simulación para redes Ad-hoc.

Para poder realizar simulaciones paramétricas se ha desarrollado una arquitectura de simulación, la cual hace uso de diversos lenguajes de programación. En primer lugar la descripción de los parámetros de red tales como, topología, movilidad y la definición de las comunicaciones, son programadas en Otcl. La implementación de los protocolos de comunicación, así como los modelos de propagación, están realizados en NS-2 utilizando C++. Hay que decir que en este proyecto no se ha realizado ninguna modificación de la codificación en C++.

Como ya se conoce, NS-2 proporciona como salida un archivo de trazas que debe ser procesado para obtener información de los eventos sucedidos en la red durante el tiempo de simulación. Para procesar dichos archivos se ha utilizado el lenguaje de procesamiento de textos PERL (Practical extraction and report language). Perl es un lenguaje de programación diseñado por Larry Wall en 1987. Perl toma características del lenguaje C, del lenguaje interpretado shell (sh), AWK, sed, Lisp y, en un grado inferior, de muchos otros lenguajes de programación. Utilizando PERL se pueden definir ciertas métricas para medir el rendimiento de la red, como son el throughput

(rendimiento), retardos, etc. Los resultados de procesamiento del archivo PERL son almacenados en archivos de texto.

Por último para automatizar todo el proceso de simulación, se han realizado diversos programas en C, los cuales hacen uso tanto de NS-2 a través de llamadas a los archivos Otcl, así como de los archivos PERL de procesamiento. Los programas en C realizan llamadas al sistema operativo mediante la función system, para llamar tanto a NS-2, como al código escrito en PERL.

Por último y con el objetivo de representar gráficamente los resultados de simulación, se ha utilizado el paquete ofimático open software denominado Open Office.

En la siguiente Figura 6 se representa la arquitectura de simulación descrita en este capítulo.

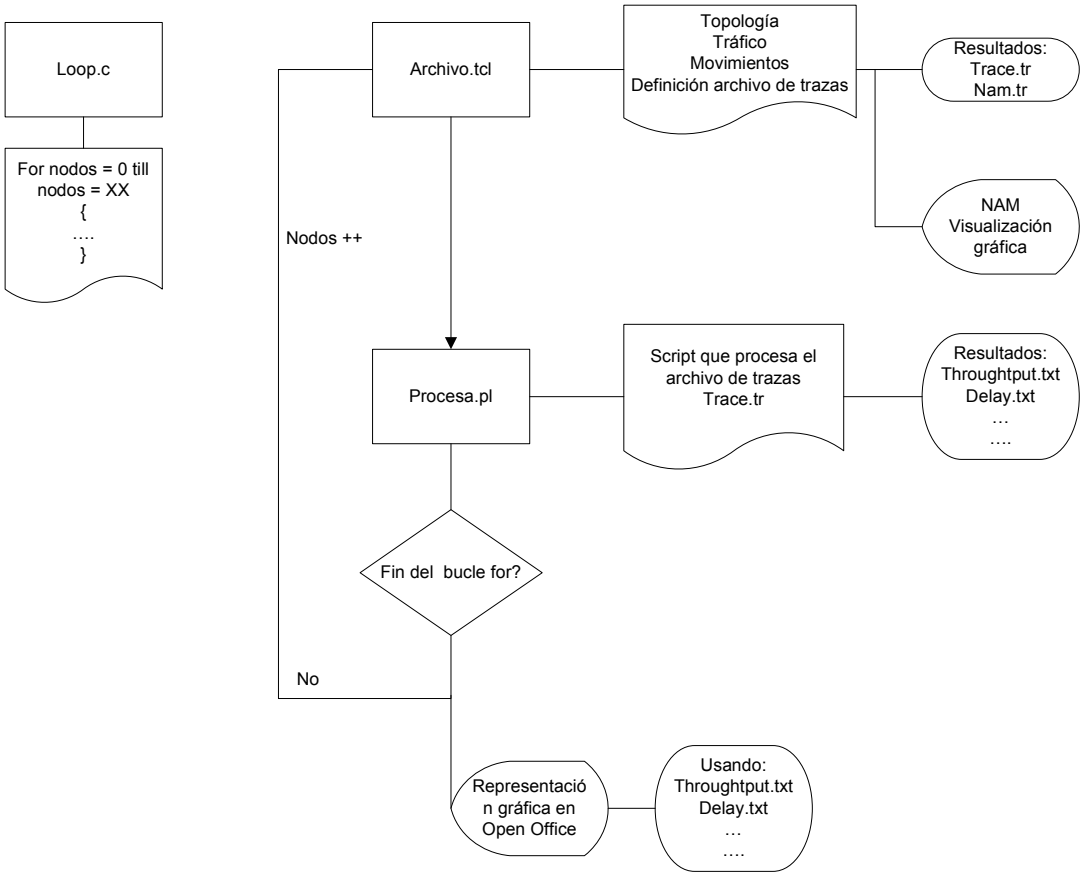

Figura 6: Arquitectura de simulación.

En la anterior figura se puede observar todo el flujo de ejecución de la arquitectura de simulación implementada. Las salidas de NS-2 son básicamente de dos tipos. Por un lados el archivo de trazas donde se recogen todos los eventos ocurridos en las red durante el tiempo de simulación, y archivo de trazas Nam para realizar un visualización de la simulación.

En la siguiente imagen, Figura 7 se ilustra un archivo de trazas.

```
r 2.556838879 _1_ RTR  --- 0 chr 512 [0 0 0 0] ------- [1:0 2:0 32 0] [0] 0 1
s 2.556838879 _1_ RTR  --- 0 AODV 48 [0 0 0 0] ------- [1:255 -1:255 30 0] [0x2 1 1 [2 0] [1 4]] (REQUEST)
r 2.557779023 _36_ RTR  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 30 0] [0x2 1 1 [2 0] [1 4]] (REQUEST)
r 2.557779058 _37_ RTR  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 30 0] [0x2 1 1 [2 0] [1 4]] (REQUEST)
r 2.557779080 _21_ RTR  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 30 0] [0x2 1 1 [2 0] [1 4]] (REQUEST)
r 2.557779135 _41_ RTR  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 30 0] [0x2 1 1 [2 0] [1 4]] (REQUEST)
r 2.557779214 _45_ RTR  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 30 0] [0x2 1 1 [2 0] [1 4]] (REQUEST)
r 2.557779280 _29_ RTR  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 30 0] [0x2 1 1 [2 0] [1 4]] (REQUEST)
```

Figura 7: Archivo de trazas proporcionad por NS-2.

Como se puede observar en la figura la información está divida en filas y columnas. La información contenida en esas filas y columnas está relacionado con:

- Tipo de evento, envío de paquete (S), recepción del paquete (R), o pérdida del paquete (D).
- Tiempo de simulación en el que ocurre el evento.
- Capa que genera el evento. El evento puede ser generado por la capa de aplicación, por la capa de red, etc.
- Tipo de paquete.
- Nodo en el que se produce el evento.
- Tamaño del paquete.
- Dirección IP de la fuente y del destino. Acompañado del puerto de comunicación.
- Otros.

Hay que decir que la información proporcionada por el archivo de trazas puede ser elegida por el diseñador. Ya que existe la posibilidad de deshabilitar ciertas capas, de forma que si por ejemplo, no se quiere obtener información de la capa MAC, se puede deshabilitar dicha capa y en consecuencia el archivo de trazas no proporcionará los eventos relacionados con dicha capa.

En el apéndice B de este trabajo se encuentran todos los códigos utilizados para llevar a cabo esta arquitectura de simulación.