

Práctica redes Ad-hoc

Ejercicio 1. Estudio del algoritmo AODV con Dynamic Ad Hoc Routing Simulator (DARS)

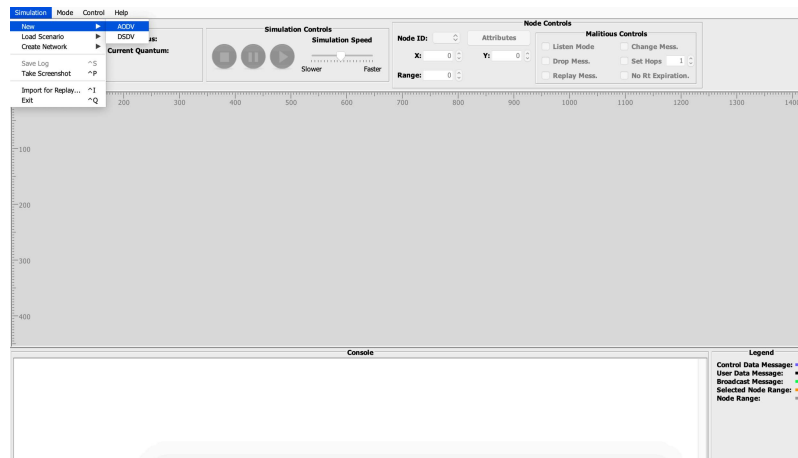
El objetivo del ejercicio es entender el funcionamiento del algoritmo AODV (Ad-hoc On-Demand Distance Vector Routing). Se trata de un algoritmo de encaminamiento uniforme bajo demanda que encamina salto a salto los mensajes a través de una red ad-hoc.

Para estudiar su funcionamiento en una topología de red determinada, usaremos el simulador DARS, por ser multiplataforma (implementado en Java), open source y sencillo de ejecutar. Se pide un documento donde se dé respuesta a las siguientes cuestiones, comentando los hallazgos en cada paso.

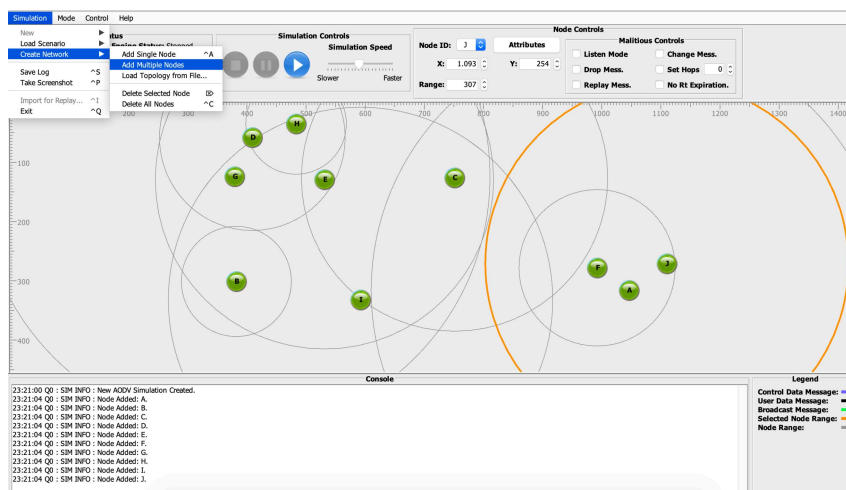
1. Descargar el simulador del repositorio del proyecto (o a través del Campus Virtual):

<https://sourceforge.net/projects/dars/>

2. Ejecutar el fichero DARS-1.1.1.jar desde el explorador de ficheros o a través de línea de comandos: `java -jar DARS-1.1.1.jar`
3. Crear nueva simulación en modo AODV

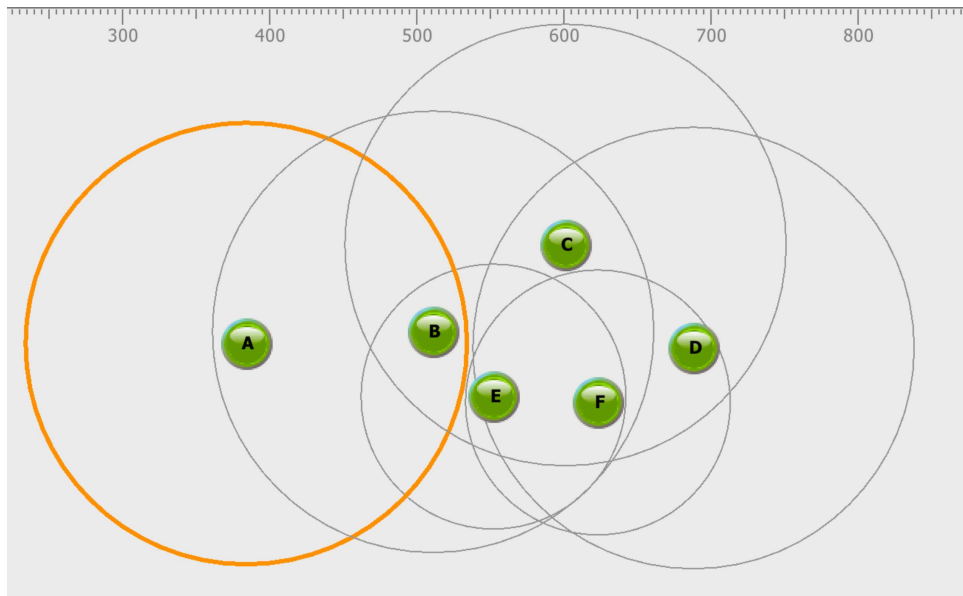


4. Juega con las opciones que ofrece el simulador para crear una red, ya sea manualmente (añadiendo cada nodo individualmente) o añadiendo varios al mismo tiempo. Hazte con los controles básicos del simulador.



Date cuenta de que puedes controlar el rango de cobertura de cada uno de los nodos, simulando así una red ad-hoc inalámbrica. De esta forma, para que dos nodos puedan comunicarse (y poderse reenviar mensajes) ambos han de estar mutuamente incluidos en sus círculos dentro de la topología.

5. Crea la siguiente topología e inicia la simulación. Puedes controlar la velocidad del simulador y pausarla en cualquier momento.



Nota: como podrás comprobar, puede parecer que los nodos intercambian información de control periódica, lo que a priori no concuerda con el comportamiento propio de uno de estos algoritmos bajo demanda (que solo buscan la ruta en el momento en que se necesitan). No obstante, el simulador muestra estas interacciones para reflejar el hecho de que cada nodo mantiene viva la conexión con los vecinos que tiene a su alrededor (para tener constancia de su proximidad), de forma que cuando alguien realiza una petición, la red se inunda reenviándola a través de esos nodos.

6. Envía un mensaje de A a D (haciendo click derecho en A). En la pantalla observa la secuencia de eventos que suceden (mensajes intercambiados, log, actualización de la tabla de encaminamiento de los nodos, etc.). ¿Es posible el envío del mensaje? ¿Qué ruta siguen los mensajes?
7. ¿Qué nodo(s) envía(n) un RREQ? ¿Cuál es la función de este mensaje?
8. ¿Qué nodo(s) envía(n) un RREP? ¿Cuál es la función de este mensaje?
9. Inmediatamente después del intento de enviar el mensaje (representado con un flujo de mensajes en negro entre los nodos en el simulador), pausa la simulación y echa un vistazo a la tabla de encaminamiento (vectores distancia en caché) del nodo B. ¿Qué significa cada entrada en la tabla?
10. Fíjate en el campo EXP de esa tabla de encaminamiento. Sin cerrar esa ventana, reanuda la simulación (y acelera la velocidad si es necesario). ¿Qué ocurre con esos valores? Verás que tras cierto tiempo, las entradas de la tabla cambian. Según el

funcionamiento del algoritmo AODV que hemos visto en clase, ¿cómo se explica este suceso?

11. Elimina el nodo C y a continuación envía el mismo mensaje desde A hasta D. ¿Cuál es el resultado de aplicar ahora el algoritmo de encaminamiento?
12. Justo tras enviar el mensaje anterior pausa la simulación y, con ella pausada, encarga un envío de un mensaje con el mismo emisor y destinatario, para cuando la simulación se reanude. No obstante, antes de reanudar la simulación (con objeto de que la caché de los nodos sea aún válida), aleja el nodo F lo suficiente para que no esté en el rango de los demás nodos. Ahora sí, reanuda la simulación.

¿Qué tipo de mensaje se genera para notificar ante este cambio inesperado en la ruta? ¿Quién genera este mensaje para avisar al emisor? ¿Es posible efectuar el envío? Apóyate en el log del programa si es necesario.