

TRABALLO FIN DE GRAO
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN TECNOLOXÍAS DA INFORMACIÓN



Práctica 3: MANETs en INET

Estudiante 1: Óscar Olveira Miniño

Estudiante 2: Alejandro Javier Herrero Arango

A Coruña, diciembre de 2024.

Índice general

1	AODV	1
1.1	Ejercicio 1.1	1
1.2	Ejercicio 1.2	2
1.3	Ejercicio 1.3	2
1.4	Ejercicio 1.4	2
1.5	Ejercicio 1.5	3
1.6	Ejercicio 1.6	3
1.7	Ejercicio 1.7	3
1.8	Ejercicio 1.8	3
2	DSDV	4
2.1	Ejercicio 2.1	4
2.2	Ejercicio 2.2	5
2.3	Ejercicio 2.3	5
2.4	Ejercicio 2.4	6
2.5	Ejercicio 2.5	6
2.6	Ejercicio 2.6	6

Índice de figuras

1.1	Logs que muestran el envío del primer RREQ y los nodos que lo reciben	1
1.2	Nodos que reenvían el primer RREQ (A la izquierda, mobile[10]; A la derecha, mobile[12])	2
2.1	Log del nodo que manda el primer Hello con hopdistance 3	4
2.2	Tabla de enrutamiento nodo 8	5

1.1 Ejercicio 1.1

1.1.1 ¿Qué nodos reenvían el primer paquete RREQ enviado por static1? ¿Y el segundo RREQ? ¿Por qué?

SEMILLA A USAR DURANTE LA PRACTICA: 2701

El primer paquete RREQ enviado por static1 es recibido y reenviado únicamente por los nodos mobile[10] y mobile [12], a pesar de enviarse con intención de alcanzar todos los nodos de la red (Figura 1.1). Esto ocurre porque el primer envío contiene un TTL igual a 2 (aparece en la documentación de Inet), por lo que solamente van a responder los dispositivos a los que le llegue un TTL > 1 para así poder hacer el reenvío.

El segundo RREQ es reenviado por 10, 12, 3, 1, 7, 2. Al hacer el segundo reenvío, el TTL del paquete pasa a ser 4 (según la documentación de INET, en los siguientes paquetes, el TTL se suma 2 con respecto al anterior ya que el objetivo es poder llegar lo más lejos posible). Como ahora el TTL es 4, pasa por 10 y 12 otra vez (el TTL pasa a ser 3), 12 no alcanza ningún objetivo pero 10 logra mandar ese paquete a 1 y 3 (el TTL pasa a ser 2) y como 3 llega a alcanzar a 7 y 2, se los envía también. Llegados a este punto, el TTL es 1 por lo que ya que se acabaría todos los posibles reenvíos del segundo RREQ.

```

** Event #1719 t=10.002964223082 Manet.static1.wlan[0].radio (Ieee80211ScalarRadio, id=79) on aodv::Rreq (inet::Packet, id=1642)
INFO: Transmission started: (inet::physicallayer::WirelessSignal)aodv::Rreq (58 us 99 B) (inet::Packet)aodv::Rreq (99 B) (inet::Seq
INFO: Changing radio transmission state from IDLE to TRANSMITTING.
INFO: Changing radio transmitted signal part from NONE to WHOLE.
** Event #1720 t=10.002964558393 Manet.mobile[10].wlan[0].radio (Ieee80211ScalarRadio, id=860) on aodv::Rreq (inet::physicallayer
INFO: Reception started: attempting (inet::physicallayer::WirelessSignal)aodv::Rreq (58 us 99 B) (inet::Packet)aodv::Rreq (99 B) (ir
INFO: Changing radio reception state from IDLE to RECEIVING.
INFO: Changing radio received signal part from NONE to WHOLE.
** Event #1721 t=10.002965000243 Manet.mobile[12].wlan[0].radio (Ieee80211ScalarRadio, id=990) on aodv::Rreq (inet::physicallayer
INFO: Reception started: attempting (inet::physicallayer::WirelessSignal)aodv::Rreq (58 us 99 B) (inet::Packet)aodv::Rreq (99 B) (ir
INFO: Changing radio reception state from IDLE to RECEIVING.
INFO: Changing radio received signal part from NONE to WHOLE.
** Event #1722 t=10.002965127308 Manet.mobile[1].wlan[0].radio (Ieee80211ScalarRadio, id=275) on aodv::Rreq (inet::physicallayer
INFO: Reception started: not attempting (inet::physicallayer::WirelessSignal)aodv::Rreq (58 us 99 B) (inet::Packet)aodv::Rreq (99 B)
** Event #1723 t=10.002965194789 Manet.mobile[3].wlan[0].radio (Ieee80211ScalarRadio, id=405) on aodv::Rreq (inet::physicallayer
INFO: Reception started: not attempting (inet::physicallayer::WirelessSignal)aodv::Rreq (58 us 99 B) (inet::Packet)aodv::Rreq (99 B)
** Event #1724 t=10.002965706956 Manet.mobile[7].wlan[0].radio (Ieee80211ScalarRadio, id=665) on aodv::Rreq (inet::physicallayer
INFO: Reception started: not attempting (inet::physicallayer::WirelessSignal)aodv::Rreq (58 us 99 B) (inet::Packet)aodv::Rreq (99 B)
** Event #1725 t=10.002965764806 Manet.mobile[2].wlan[0].radio (Ieee80211ScalarRadio, id=340) on aodv::Rreq (inet::physicallayer
INFO: Reception started: not attempting (inet::physicallayer::WirelessSignal)aodv::Rreq (58 us 99 B) (inet::Packet)aodv::Rreq (99 B)
** Event #1726 t=10.002965801036 Manet.mobile[11].wlan[0].radio (Ieee80211ScalarRadio, id=925) on aodv::Rreq (inet::physicallayer
INFO: Reception started: not attempting (inet::physicallayer::WirelessSignal)aodv::Rreq (58 us 99 B) (inet::Packet)aodv::Rreq (99 B)
** Event #1727 t=10.002966033624 Manet.mobile[6].wlan[0].radio (Ieee80211ScalarRadio, id=600) on aodv::Rreq (inet::physicallayer
INFO: Reception started: not attempting (inet::physicallayer::WirelessSignal)aodv::Rreq (58 us 99 B) (inet::Packet)aodv::Rreq (99 B)
** Event #1728 t=10.002966245726 Manet.mobile[13].wlan[0].radio (Ieee80211ScalarRadio, id=1055) on aodv::Rreq (inet::physicallayer
INFO: Reception started: not attempting (inet::physicallayer::WirelessSignal)aodv::Rreq (58 us 99 B) (inet::Packet)aodv::Rreq (99 B)
** Event #1729 t=10.002966271749 Manet.mobile[0].wlan[0].radio (Ieee80211ScalarRadio, id=210) on aodv::Rreq (inet::physicallayer
INFO: Reception started: not attempting (inet::physicallayer::WirelessSignal)aodv::Rreq (58 us 99 B) (inet::Packet)aodv::Rreq (99 B)
** Event #1730 t=10.002966524973 Manet.mobile[5].wlan[0].radio (Ieee80211ScalarRadio, id=535) on aodv::Rreq (inet::physicallayer

```

Figura 1.1: Logs que muestran el envío del primer RREQ y los nodos que lo reciben

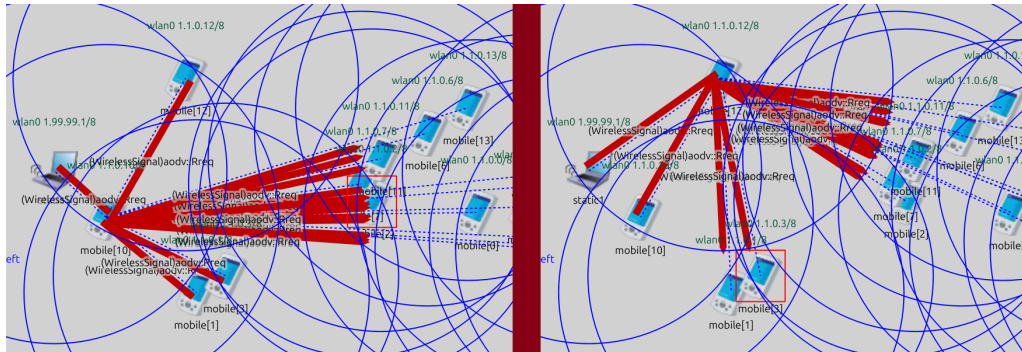


Figura 1.2: Nodos que reenvían el primer RREQ (A la izquierda, mobile[10]; A la derecha, mobile[12])

1.2 Ejercicio 1.2

1.2.1 Elige el nodo intermedio de la ruta que sigue el primer paquete RREQ que llega a static2. Muestra su tabla de enrutamiento (vector `<Ipv4route *>` dentro del módulo `ipv4.routingTable`) justo antes y justo después de recibir el primer RREQ. Explica las diferencias y cómo se crean las entradas que aparecen (incluyendo los campos más importantes).

1.3 Ejercicio 1.3

1.3.1 Haz lo mismo justo antes y justo después del primer RREP.

1.4 Ejercicio 1.4

1.4.1 Tras aplicar la nueva configuración, ¿Cuál es el primer nodo en darse cuenta de la caída? ¿Cómo? Muestra una captura del log del nodo que se da cuenta que muestre el motivo. ¿Notifica este nodo la caída del nodo?

Hay que aclarar que con la semilla que usamos, a partir de $t=15$ el nodo alternativo para establecer la ruta se alejaba un poco de static2 por lo que hicimos las pruebas en $t=12s$.

1.5 Ejercicio 1.5

- 1.5.1 Muestra el contenido del paquete RERR en Wireshark explicando los campos más importantes. ¿Qué IP tiene como destino? ¿Por qué?

1.6 Ejercicio 1.6

- 1.6.1 Explica cómo se propaga el RERR por la red. ¿Qué nodos lo reenvían? ¿Cómo sabe un nodo si debe reenviar el RERR?

1.7 Ejercicio 1.7

- 1.7.1 Muestra capturas de la tabla de enrutamiento de un nodo antes y después de recibir un RERR y explica en qué cambia.

1.8 Ejercicio 1.8

- 1.8.1 ¿Qué hace static1 al recibir el RERR? Muestra el contenido del siguiente RREQ en Wireshark. ¿En qué cambia con respecto al de la pregunta 1?

2.1 Ejercicio 2.1

2.1.1 Avanza la simulación hasta el instante $t = 7$ s. Busca el primer paquete Hello transmitido a partir a ese instante con un valor de hopdistance de al menos 3 y muestra una captura del contenido. Explica el significado de los campos srcAddress y nextAddress, utilizando para explicarlos una captura de la tabla de enrutamiento del nodo que está transmitiendo el paquete (i.e., no el que consta en srcAddress)

```
** Event #46281 t=7.005012955107 Manet.mobile[12].wlan[0].radio (Ieee80211ScalarRadio, id=990) on selfmsg receptionTimer (omnetpp::cMessage, id=50572)
INFO: Reception ended: ignoring (inet::physicalLayer::WirelessSignal)Hello (54 us 87 B) (inet::Packet)Hello (87 B) (inet::SequenceChunk) length = 87 B WHOLE as :
** Event #46282 t=7.005013271783 Manet.mobile[10].wlan[0].radio (Ieee80211ScalarRadio, id=860) on selfmsg receptionTimer (omnetpp::cMessage, id=50573)
INFO: Reception ended: ignoring (inet::physicalLayer::WirelessSignal)Hello (54 us 87 B) (inet::Packet)Hello (87 B) (inet::SequenceChunk) length = 87 B WHOLE as Scalar
** Event #46283 t=7.005013464089 Manet.static1.wlan[0].radio (Ieee80211ScalarRadio, id=79) on selfmsg receptionTimer (omnetpp::cMessage, id=50574)
INFO: Reception ended: ignoring (inet::physicalLayer::WirelessSignal)Hello (54 us 87 B) (inet::Packet)Hello (87 B) (inet::SequenceChunk) length = 87 B WHOLE as Scalar
** Event #46284 t=7.007887772166 Manet.radioMedium (Ieee80211ScalarRadioMedium, id=3) on selfmsg removeNonInterferingTransmissions (omnetpp::cMessage, id=0)
** Event #46285 t=7.0136600356 Manet.mobile[8].ipv4.ip (Ipv4, id=765) on Hello (inet::Packet, id=45441)
INFO: Received (inet::Packet)Hello (16 B) (inet::DsdvHello) srcAddress = 1.1.0.7, sequencenumber = 6, nextAddress = 1.1.0.8, hopdistance = 3 from upper layer.
DETAIL: Sending datagram 'Hello' with destination = 255.255.255.255
DETAIL: destination address is broadcast, sending packet to broadcast MAC address
INFO: Sending (inet::Packet)Hello (36 B) (inet::SequenceChunk) length = 36 B to output interface = wlan0.
INFO (MessageDispatcher)Manet.mobile[8].ipv4.lp: Dispatching packet to interface, interfaceId = 101, inGate = (omnetpp::cGate)in[1] ← ip.queueOut, outGate = (omnetpp::cGate)in[0]
INFO (MessageDispatcher)Manet.mobile[8].nl: Dispatching packet to interface, interfaceId = 101, inGate = (omnetpp::cGate)in[0] ← ipv4.ifOut, outGate = (omnetpp::cGate)in[1]
INFO (MessageDispatcher)Manet.mobile[8].cb: Dispatching packet to interface, interfaceId = 101, inGate = (omnetpp::cGate)in[1] ← nl.out[1], outGate = (omnetpp::cGate)in[0]
INFO (MessageDispatcher)Manet.mobile[8].bl: Dispatching packet to interface, interfaceId = 101, inGate = (omnetpp::cGate)in[0] ← cb.out[0], outGate = (omnetpp::cGate)in[1]
INFO (MessageDispatcher)Manet.mobile[8].li: Dispatching packet to interface, interfaceId = 101, inGate = (omnetpp::cGate)in[0] ← bl.out[1], outGate = (omnetpp::cGate)in[1]
** Event #46286 t=7.0136600356 Manet.mobile[8].wlan[0].llc (Ieee80211LlcLpd, id=726) on Hello (inet::Packet, id=45441)
```

Figura 2.1: Log del nodo que manda el primer Hello con hopdistance 3

Como se puede ver la imagen, el nodo que manda el primer mensaje Hello con hopdistance 1 es el nodo 8. Vamos a fijarnos en su tabla de enrutamiento:

```

ownedObjects[6] (omnetpp::cOwnedObject)
├── [0] (vector<Ipv4Route *>) routes: size=17
│   └── elements[17] (inet::Ipv4Route *)
│       ├── [0] ??? 1.1.0.0/32 gw:1.1.0.0 metric:1 if:wlan0 seqNo:6 expiryTime:11.725265486533
│       ├── [1] ??? 1.1.0.1/32 gw:1.1.0.0 metric:4 if:wlan0 seqNo:4 expiryTime:9.686108756567
│       ├── [2] ??? 1.1.0.2/32 gw:1.1.0.5 metric:2 if:wlan0 seqNo:6 expiryTime:11.652760364534
│       ├── [3] ??? 1.1.0.3/32 gw:1.1.0.0 metric:3 if:wlan0 seqNo:4 expiryTime:9.036773380074
│       ├── [4] ??? 1.1.0.4/32 gw:1.1.0.4 metric:1 if:wlan0 seqNo:6 expiryTime:11.555479839635
│       ├── [5] ??? 1.1.0.5/32 gw:1.1.0.5 metric:1 if:wlan0 seqNo:6 expiryTime:11.859803172991
│       ├── [6] ??? 1.1.0.6/32 gw:1.1.0.13 metric:2 if:wlan0 seqNo:6 expiryTime:11.630463083307
│       ├── [7] ??? 1.1.0.7/32 gw:1.1.0.0 metric:2 if:wlan0 seqNo:6 expiryTime:11.6636600356
│       ├── [8] ??? 1.1.0.9/32 gw:1.1.0.9 metric:1 if:wlan0 seqNo:6 expiryTime:11.394929766779
│       ├── [9] ??? 1.1.0.10/32 gw:1.1.0.0 metric:4 if:wlan0 seqNo:4 expiryTime:9.389669147591
│       ├── [10] ??? 1.1.0.11/32 gw:1.1.0.0 metric:2 if:wlan0 seqNo:6 expiryTime:11.528981602141
│       ├── [11] ??? 1.1.0.12/32 gw:1.1.0.0 metric:5 if:wlan0 seqNo:4 expiryTime:9.942589852095
│       ├── [12] ??? 1.1.0.13/32 gw:1.1.0.13 metric:1 if:wlan0 seqNo:6 expiryTime:11.064080361441
│       ├── [13] ??? 1.1.0.14/32 gw:1.1.0.14 metric:1 if:wlan0 seqNo:6 expiryTime:11.974955863522
│       ├── [14] ??? 1.99.99.1/32 gw:1.1.0.0 metric:4 if:wlan0 seqNo:4 expiryTime:9.423779215448
│       ├── [15] ??? 1.99.99.2/32 gw:1.1.0.14 metric:2 if:wlan0 seqNo:6 expiryTime:11.967831552786
│       └── [16] C 127.0.0.0/8 gw:* metric:1 if:lo0

```

Figura 2.2: Tabla de enrutamiento nodo 8

Según la imagen 2.1, el campo srcAddress es el nodo que generó el paquete Hello, en este caso es el 1.1.0.7 y el campo nextAddress es la siguiente dirección que va a retransmitir la trama, en este caso 1.1.0.8.

2.2 Ejercicio 2.2

2.2.1 ¿Qué valor tiene de sequencenumber? ¿Qué quiere decir ese valor?

Como se puede ver en la imagen 2.1, el campo sequencenumber tiene valor 6. Este valor ayuda a identificar entradas obsoletas para así evitar bucles. Para ver si una ruta es válida o obsoleta llega con ver si este valor es par (ruta disponible) o impar (ruta obsoleta/caída). De esta forma, nos indica como de actualizado está la información de la ruta, el estado de la ruta y evita posibles conflictos en la red.

2.3 Ejercicio 2.3

2.3.1 ¿Cómo se modifica? ¿Qué nodo lo modifica, y cuándo lo hace?

Este valor lo modifica el propio nodo en su tabla de enrutamiento, incrementándolo en 2 cuando la ruta que es válida o si es inválida solamente lo incrementa a 1. El nodo actualiza este valor cuando descubre una ruta mejor hacia el destino. Si el nuevo número de secuencia recibido es mayor (y par), el nodo adopta ese valor, lo registra en su tabla y actualiza el próximo salto (nextAddress) con el nodo que recibió la información (en este caso mobile[8]). En el caso de una ruta no válida, genera un sequencenumber impar y establece el costo de la ruta como infinito para indicar que esa ruta no se puede usar.

2.4 Ejercicio 2.4

- 2.4.1** Muestra la tabla de enrutamiento del nodo que recibe el Hello de la pregunta anterior justo antes y justo después de recibirlo, relacionándola con el contenido del paquete. Si se actualiza la tabla, explica por qué se actualiza y las entradas que se crean. Si no se actualiza, explica por qué no se actualiza y di qué entrada se crearía (destino, gateway, métrica) si se actualizase con la información del paquete.

2.5 Ejercicio 2.5

- 2.5.1** Avanza hasta la caída del nodo en $t = 15$ s. Ten en cuenta que la ruta en ese momento puede ser diferente a la de AODV, y por lo tanto el nodo a desactivar también. ¿Cuál es el primer nodo en darse cuenta de la caída? ¿Notifica la caída del nodo de alguna forma?

Hay que aclarar que hemos cambiado de semilla ya que con DSDV no funcionaba bien con la semilla que usamos en los anteriores apartados.

2.6 Ejercicio 2.6

- 2.6.1** ¿Cómo se repara la ruta entre static1 y static2? ¿En qué momento?

non recibe toda a tabla de enrutamiento cando si que deberia, algunhas veces os nodos mandan unha entrada da sua tabla, outras veces manda varias entradas...