

802.11 MAC DCF

Práctica 4 – Diseño de Redes

Instrucciones

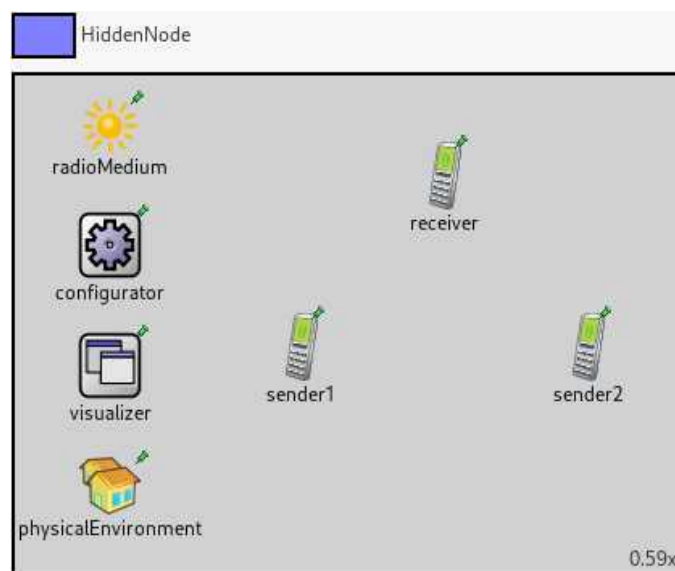
Elabore una **memoria** explicando los resultados de simulación obtenidos al seguir los pasos indicados. Incluya todas las figuras que se piden, así como cualquier figura adicional que considere importante para explicar los resultados de simulación. Las explicaciones deben hacerse a la luz de lo explicado en la teoría de la asignatura.

- El único formato aceptado es **PDF**.
- Utilice la plantilla **LaTeX Overleaf** oficial del Trabajo Fin de Grado disponible en la Wiki FIC.
- **Referencie todas las figuras** en el texto.
- El contenido de las figuras debe ser legible **sin necesidad de hacer zoom**.
- Cuide la **redacción** y la **ortografía**. Ambas serán tenidas en cuenta.
- Utilice el **formato vectorial SVG** para almacenar las figuras siempre que sea posible. Para incluirlas en la memoria, puede convertirlas a PDF con la utilidad *rsvg-convert*, incluida en el paquete *librsvg2-bin* de Ubuntu. El siguiente comando convierte a PDF todos los archivos SVG de una carpeta:

```
$ for i in *.svg; do rsvg-convert -f pdf -o ${i%.*}.pdf $i; done
```

Escenario

La red constará de dos nodos inalámbricos (*sender1* y *sender2*) que enviarán datagramas UDP a un nodo común (*receiver*). Los tres nodos serán de tipo *inet.AdhocHost*, variante de *inet.WirelessHost* que elimina la necesidad de utilizar puntos de acceso. Los nodos estarán a una distancia de aproximadamente 300m entre sí.



Se simularán los siguientes 4 escenarios **durante 5 segundos**:

- CSMA/CA **sin** RTS/CTS
 - **Sin** obstáculo (1)
 - **Con** obstáculo (2)
- CSMA/CA **con** RTS/CTS
 - **Con** obstáculo (3)
 - **Sin** obstáculo (4)

En las configuraciones con obstáculo (2 y 3) *sender1* y *sender2* no serán visibles entre sí debido al obstáculo, pero *receiver* seguirá siendo visible para ambos. Para definir el obstáculo se utiliza el parámetro config de *physicalEnvironment* (ver plantilla abajo). En todas las configuraciones cada nodo debe estar dentro de la cobertura de los otros dos (asegúrese activando *displayCommunicationRange* y ajustando la potencia).

El nodo *sender1* enviará paquetes siguiendo una distribución exponencial de media 3 ms. El nodo *sender2* los enviará siguiendo una distribución truncnormal de media 3 ms y desviación típica 1 ms. Los datagramas serán de 500 bytes y la velocidad de transmisión (*** wlan[*].bitrate*) será de 9 Mbps. Utilice *UdpBasicApp* en los nodos *sender[12]* y *UdpSink* en *receiver*. Para que los nodos transmitan paquetes UDP desde el primer momento configure *startTime=0* para todas las apps y *** arp.typename="GlobalArp"* para **evitar el tráfico ARP** al principio de la simulación. **Fije la semilla** con *seed-set=s* (donde *s* es un valor entero) para que las simulaciones sean repetibles. Elija para *s* la salida de ejecutar *echo \$RANDOM* en un terminal.

La siguiente plantilla define los 4 escenarios anteriores y el obstáculo (los valores **x** e **y** deben ser sustituidos por su posición, que dependerá de las coordenadas elegidas para los nodos *sender1* y *sender2* en el fichero *.ned*):

```
[General]
record-eventlog=true

[Config _Wall]
*.physicalEnvironment.config = xml("<environment> \
<object position='min x y 0' orientation='0 0 0' shape='cuboid 10 200 40' \
material='concrete'/> \
</environment>")
*.radioMedium.obstacleLoss.typename = "IdealObstacleLoss"

[Config _RTS]
** wlan[*].mac.dcf.rtsPolicy.rtsThreshold = 100B

[Config No_Wall_No_Rts]

[Config Wall_No_Rts]
extends=_Wall

[Config Wall_Rts]
extends=_Wall,_RTS

[Config No_Wall_Rts]
extends=_RTS
```

Nota: para incorporar la grabación de los vectores *cw* y *backoffSlots* solicitados en algunos apartados debe modificar los ficheros de INET `inet/linklayer/ieee80211/mac/contention/Contention.*` aplicando el parche disponible en la sección de moodle en la que se encuentra este enunciado (`contention.diff`).

Incluya el contenido completo de los ficheros .ned y .ini en la memoria y responda a las siguientes cuestiones:

1. Simule el escenario 1 y obtenga **gráficas en función del tiempo** de:

- Paquetes descartados por *receiver* por recibirse incorrectamente (*packetDropIncorrectlyReceived*). Esto se corresponde con las colisiones entre los nodos *sender1* y *sender2*.
- Contention window (*cw*) y backoff (*backoffSlots*) en el nodo *sender1*.
- Retardo extremo a extremo (*endToEndDelay*) medido en el *receiver*. Escala: **ms** (por defecto está en segundos, por lo tanto: botón derecho → *Apply* → *Other...* → *multiply-by* → 1000).

Explique lo que se observa en las gráficas, relacionando las 3 últimas con la primera.

2. Obtenga un **diagrama de secuencia** de la simulación **durante una colisión**. Para simplificar el diagrama filtre por tipo NED → `inet.AdhocHost`, elimine los *self-message* y las dependencias y ordene los nodos verticalmente de forma que el *receiver* se encuentre entre los dos *senders*. Muestre el intercambio de paquetes entre los nodos durante la colisión y lo que ocurre tras la colisión (reenvío). A la vista del diagrama, ¿por qué ocurren las colisiones?

(**Nota:** para evitar la ralentización del IDE debido al tamaño del fichero `.elog` que resulta de realizar los 5 minutos de simulación completos, se recomienda realizar una simulación parcial. Para ello busque el **primer** mensaje “*Received an erroneous frame from PHY*” en el **log** durante una ejecución normal, apunte el instante de tiempo de la colisión y repita la simulación **deteniéndola** 10 ms después de ese instante.)

3. Simule ahora el escenario 2 y obtenga las 4 gráficas del apartado 1. ¿Qué se observa? ¿Qué diferencias hay con el escenario 1 en cada una?

4. Obtenga un diagrama de secuencia de una colisión ocurrida en el escenario 2. ¿Qué diferencia hay con respecto al escenario 1? ¿Por qué ocurren ahora las colisiones?

5. Muestre y **explique** los siguientes valores para los dos escenarios anteriores:

- Número de paquetes UDP que bajan de la capa de aplicación (*packetSent*) en cada nodo *sender*.
- Número de paquetes extraídos de cola MAC para ser transmitidos (*packetPopped*) en cada *sender*.
- Número de transmisiones de capa MAC (*packetSentToLower*) en cada *sender*.
- Número de retransmisiones de capa MAC (*packetSentToLower* – *packetPopped*) en cada *sender*.
- Número de paquetes UDP recibidos por *receiver*.

6. Simule el escenario 3 y muestre las mismas gráficas que en la cuestión 1. ¿Qué se observa ahora en comparación con el escenario 2?

7. Obtenga un diagrama de secuencia de una colisión en el escenario 3. ¿Cuándo ocurren ahora las colisiones? ¿Deberían ser más, menos o las mismas (en media) que en el escenario 2? ¿Por qué?



8. Simule el escenario 4 y compare los resultados con los del escenario 1. ¿Se obtiene algún beneficio al usar RTS/CTS en un escenario sin nodos ocultos? ¿Cuál es el retardo medio en comparación con el escenario 1?
9. Obtenga un diagrama de secuencia de una colisión en el escenario 4. ¿Cuándo ocurren las colisiones? ¿Deberían ser más, menos o las mismas (en media) que en el escenario 1? ¿Por qué?
10. Repita los escenarios 1, 2 y 3 con una velocidad de transmisión de 6 Mbps y obtenga el **número de paquetes recibidos por receiver**, el **número de colisiones** y el **retardo extremo a extremo medio** en cada uno. ¿Cómo afecta a cada escenario la disminución de la velocidad? ¿Por qué?
11. Haga lo mismo con una velocidad de 18 Mbps. ¿Afecta a algún escenario el aumento de velocidad? ¿Por qué?