Taller de Telecomunicaciones
Modalidad SDR

Informe
Proyecto Final

Capa MAC con Implementación CSMA/CA

Mariana Gelós, Martín Randall
Tutor: Víctor González Barbone

23 de diciembre de 2016

Índice

1. Introducción **2**

2. Protocolo CSMA/CA **2**

3. Máquina de Estados **3**

4. Instalación de GR-GWN **5**

5. Creación del bloque **5**

6. Codificación del bloque **5**

7. Testeo del bloque **6**

8. Conclusiones **7**

1. Introducción

El presente proyecto consistió en desarrollar un bloque OTT (Out Of Tree) de GNU Radio que implemente una capa MAC utilizando el protocolo CSMA/CA.

Este trabajo se encuentra en el marco del proyecto GR-GWN (GNU Radio GNU Wireless Network), desarrollado por docentes del grupo ARTES del Instituto de Ingeniería Eléctrica. El mismo es un proyecto en desarrollo cuyo objetivo es la construcción de un framework destinado a implementar redes de datos inalámbricas sobre GNU Radio. En el siguiente enlace, https://github.com/vagonbar/gr-gwn, se encuentra el repositorio del proyecto junto con una wiki con información relevante para la utilización del mismo.

Por otro lado, cabe destacar que la capa MAC implementada en este proyecto será utilizada en el proyecto de fin de carrera RoCo (Comunicación Inalámbrica de Robots Móviles sobre GNU Radio).

El presente informe describe las etapas realizadas para llegar a la implementación del bloque. En primer lugar, se realizó un estudio teórico del protocolo CSMA/CA, lo que permitió construir diagramas de flujo y de máquina de estados. Luego se procedió a la instalación del proyecto GR-GWN y a la creación del bloque. Por último se codificó el bloque, tomando como referencia el diagrama de máquna de estados, y se lo testeó para distintos casos de prueba.

2. Protocolo CSMA/CA

El protocolo CSMA/CA pertenece a la familia de protocolos CSMA (Carrier Sense Multiple Access). Estos protocolos permiten arbitrar el acceso al medio en canales compartidos. Basan su funcionamiento en la posibilidad de las estaciones de sensar el medio para detectar si otros nodos están transmitiendo, y actuar acorde a ello. De esta forma, se limita la probabilidad de ocurrencia de colisiones.

Los medios inalábricos son medios compartidos que presentan problemas particulares, tales como el de la estación oculta y el de la estación expuesta. Para resolver estos problemas se crearon diversos protocolos, entre ellos el CSMA/CA (CSMA Collision Avoidence), el cual fue definido en el estándar IEEE 802.11.

Existe diversas versiones del protocolo CSMA/CA, con detección del canal físico o del canal virtual (esta última utiliza mensajes de Request to Send y Clear to Send). En este proyecto se utilizó la primer versión, la cual se puede describir de la siguiente forma:

- **Carrier Sense:** Previo a la transmisión, la estación sensa el canal para determinar si otro nodo está transmitiendo.
- **Collision Avoidence:**
  - Si el canal está ocupado, se sortea un slot de tiempo dentro de un conjunto de slots posibles, antes de volver a sensar el medio. En el siguiente punto se detallará cómo se determina el conjunto de slots posibles. Cada slot tiene una longitud igual al tiempo de propagación de ida y vuelta de una trama.
  - Si el canal está libre, se transmite la trama y se espera el acuse de recibo (ACK). Si el ACK no llega en tiempo y forma, se asume que la trama colisionó y se aplica el retroceso exponencial binario, esto es, se duplica la cantidad de slots posibles antes de volver a sensar el medio. Inicialmente se espera 1 slot, luego de una colisión se sortea entre 1 y 2 slots, luego de una segunda colisión se sortea entre 1 y 4 slots, y así sucesivamente.

El funcionamiento del protocolo se puede representar con el diagrama de flujo que se observa a continuación:

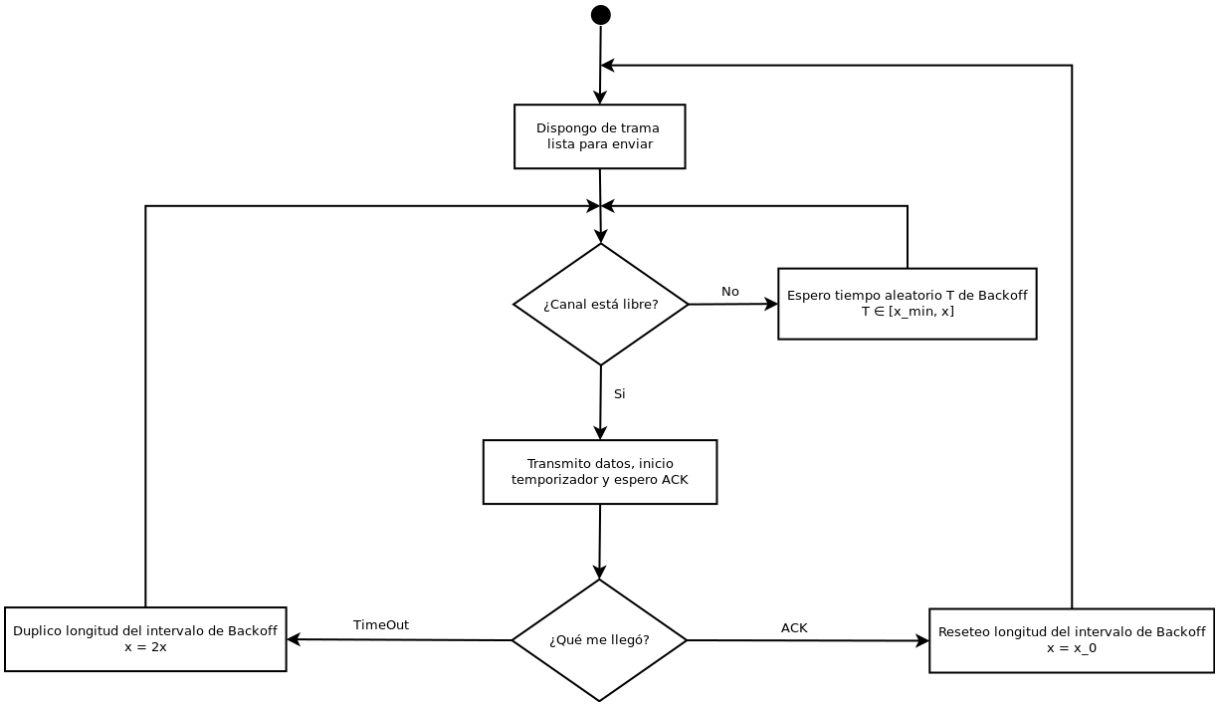

Figura 1: Diagrama de flujo del protocolo CSMA/CA

3. Máquina de Estados

El proyecto GR-GWN cuenta con el módulo `gwnfsm.py` que permite crear una máquina de estados finita, lo cual resulta muy útil para implementar protocolos de comunicaciones, como el caso de CSMA/CA. Por lo tanto fue necesario abstraer el protocolo a un diagrama de máquina de estados, para luego facilitar el proceso de codificación del bloque.

Para lograr el diseño de la máquina de estados, se utilizó la información recabada durante el estudio del protocolo, en particular del diagrama de flujo, para identificar los estados y las transiciones entre ellos. En la siguiente figura se puede observar el diagrama de la máquina:

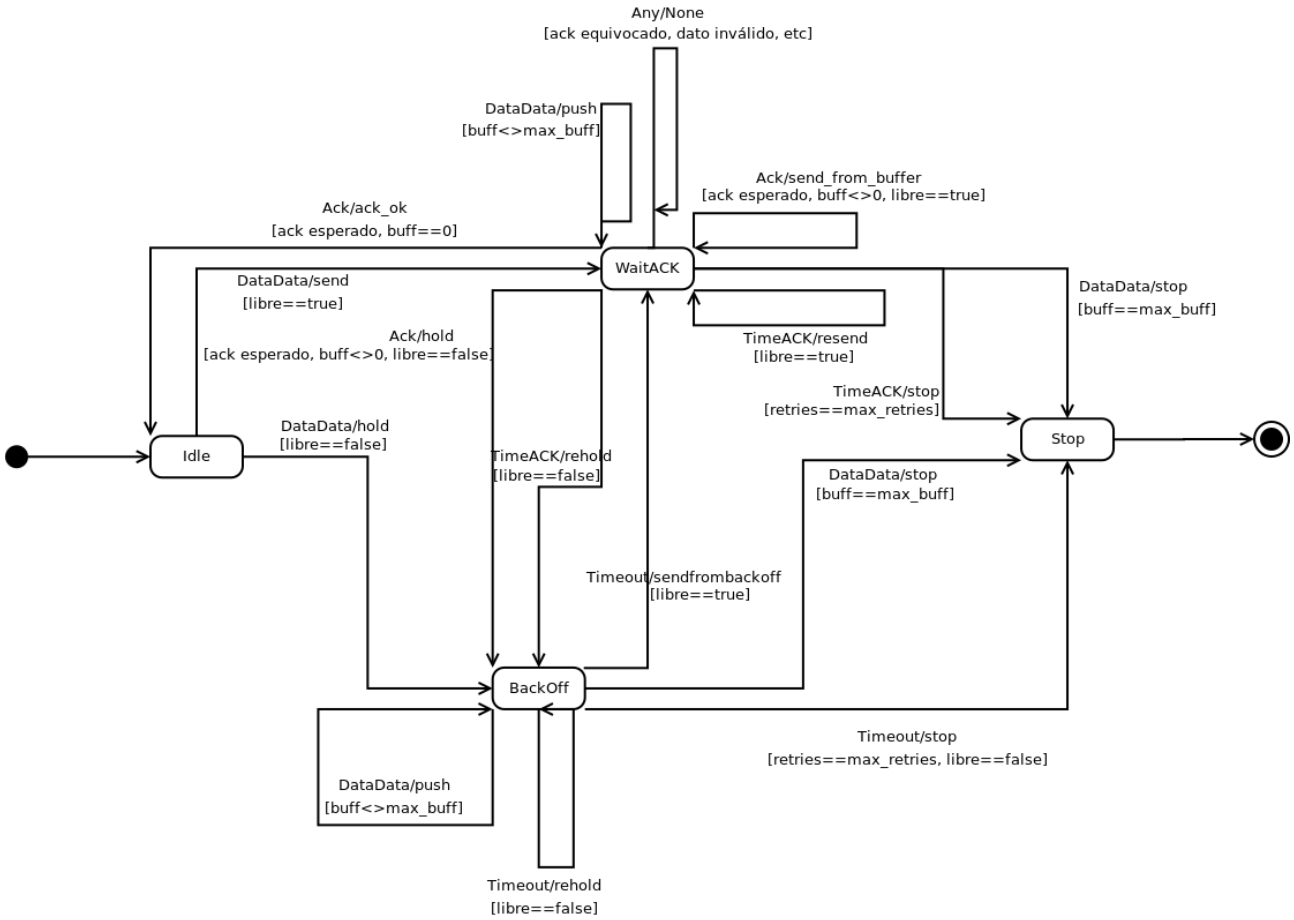

Figura 2: Diagrama de la máquina de estados

Como se puede observar, se identificaron cuatro estados:

- **Idle:** Estado inactivo. La máquina se encuentra en este estado mientras espera que llegue una trama para transmitir.
- **WaitACK:** Estado de espera de ACK. la máquina se mantiene en este estado hasta que llegue el ACK esperado en tiempo y forma.
- **BackOff:** Estado de retroceso. En este estado se espera un tiempo aleatorio antes de volver a sensar el canal.
- **Stop:** Estado de parada. Se llega a este estado cuando se supera la capacidad del buffer o la cantidad máxima de intentos de retransmisión.

Cada transición de la máquina de estados se compone de tres factores:

- El evento que llega, el cual puede ser un dato para enviar, un timeOut o un ACK. En este punto es importante aclarar que la máquina de estados maneja dos timeOuts distintos: uno representa el tiempo máximo de espera de un ACK (en el diagrama se encuentra bajo el nombre de timeACK) y otro representa el tiempo de backOff sorteado (se encuentra bajo el nombre de timeOut).

- La función que se ejecuta antes que la máquina pase al siguiente estado.
- Las condiciones que se deben verificar. Se pueden mencionar algunas a modo de ejemplo: que el canal esté libre, que el buffer no esté lleno o que el ACK sea el esperado.

4. Instalación de GR-GWN

Para poder trabajar con las herramientas implementadas por el proyecto GR-GWN, es necesario instalar el repositorio. Para esto se ejecutaron los siguientes comandos indicados en la wiki del proyecto (https://github.com/vagonbar/gr-gwn/wiki/Installation):

```
git clone https://github.com/vagonbar/gr-gwn.git
cd gr-gwn
mkdir build
cd build
cmake ../
make && sudo make install
sudo ldconfig
```

5. Creación del bloque

El script de GNU Radio `gr_modtool` realiza todas las tareas necesarias para crear un nuevo bloque y mantenerlo integrado al módulo `gr-gwn`. Se ejecuta con el siguiente comando:

```
gr_modtool add -t sync -l python
```

Con esto se agrega un bloque del tipo `sync` usando lenguaje Python.

Al crear el bloque se generan dos archivos Python: uno corresponde al bloque propiamente dicho (en este caso se encuentra bajo el nombre `CSMA_FSM.py`) y el otro al código de prueba (`qa_CSMA_FSM.py`), el cual permite testear el bloque para los casos de prueba que se definan.

6. Codificación del bloque

La codificación del bloque consistió en traducir la máquina de estados diseñada a lenguaje Python. Se utilizó como referencia otro protocolo implementado anteriormente por el proyecto GR-GWN: el protocolo Stop and Wait.

Por lo tanto, para codificar el bloque, se fueron modificando y/o agregando las funciones, transiciones y variables necesarias. Para realizar este proceso resultó fundamental contar con el diagrama correcto de la máquina de estados.

7. Testeo del bloque

Como fue mencionado anteriormente, al crear el bloque de GNU Radio se genera un programa Python para realizar el testeo. El mismo permite definir un flowgraph que sea adecuado a las pruebas que se pretende realizar. En este caso se implementó el siguiente flowgraph:

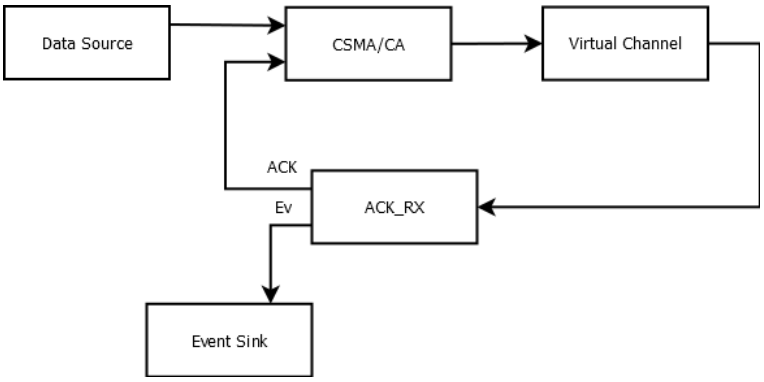

Figura 3: Flowgraph de prueba

Para validar el funcionamiento del bloque en todas las situaciones posibles, fue necesario simular distintos escenarios que permitan que la máquina pase por todas las transiciones. Para emular estas situaciones, en lugar de usar el bloque real de detección de ocupación del medio (bloque `Probe Medium`), se utilizó una función que devuelve True o False en base a un vector booleano recibido por la función como parámetro, en cada invocación sucesiva devuelve el siguiente item del vector.

A continuación se detalla la serie de pruebas realizadas. Estas pruebas abarcan todas las transiciones de la máquina de estados. A mediada que se fue realizando las pruebas, se encontraron errores en la máquina de estados, principalmente en la definición de las funciones. Luego de depurar los errores encontrados se logró que la máquina pasara correctamente todos los tests.

Evento	Dato1	ACK	Dato2	TimeOut	ACK	Dato3	TimeOut	Dato4
Estado de partida	Idle	WaitACK	Idle	BackOff	WaitACK	Idle	BackOff	BackOff
Función	Send	ack_ok	hold	sendformbackoff	ack_ok	hold	rehold	push
Canal libre	True	True	False	Tue	True	False	False	False

Cuadro 1: Test 1 - prob_pérdida = 0

Evento	TimeOut	ACK	ACK	
Estado de partida	BackOff	WaitACK	WaitACK	Idle
Función	sendfrombackoff	send_from_buffer	ack_ok	
Canal libre	True	True	True	

Cuadro 2: Test 1 (continuación) - prob_pérdida = 0

Evento	Dato1	TimeACK	Dato2	TimeACK	TimeOut	TimeOut	...	TimeOut	
Estado de partida	Idle	WaitACK	WaitACK	WaitACK	BackOff	BackOff	...	BackOff	stop
Función	Send	resend	push	rehold	rehold	rehold	...	stop	
Canal libre	True	True	True	False	False	False	...	False	

Cuadro 3: Test 2 - prob_pérd = 1

Evento	Dato1	TimeACK	TimeACK	...	TimeACK	
Estado de partida	Idle	WaitACK	WaitACK	...	WaitACK	stop
Función	Send	resend	resend	...	stop	
Canal libre	True	True	True	...	False	

Cuadro 4: Test 3 - prob_pérd = 1

Evento	Dato1	TimeACK	Dato2	TimeACK	Dato3		TimeACK	
Estado de partida	Idle	WaitACK	WaitACK	WaitACK	WaitACK	...	WaitACK	stop
Función	Send	resend	push	resend	push		stop	
Canal libre	True	True	True	True	True		True	

Cuadro 5: Test 4 - prob_pérd = 1

Evento	Dato1	Dato2	TimeACK	TimeOut	ACK	ACK	
Estado de partida	Idle	WaitACK	WaitACK	BackOff	WaitACK	WaitACK	Idle
Función	Send	push	hold_ack	sendformbackoff	send_from_buffer	ack_ok	
Canal libre	True	True	True	True	True	True	

Cuadro 6: Test 5 - prob_pérd = 0,8

8. Conclusiones

Este proyecto permitió tener un primer acercamiento al diseño, implementación y testeo de un bloque OTT de GNU Radio en el contexto del proyecto GR-GWN. Para lograr el objetivo final de obtener una capa MAC con implementación CSMA/CA fue fundamental seguir un procedimiento ordenado, construyendo a partir del diagrama de la máquina de estados.

A partir de los resultados obtenidos en la etapa de testeo, se puede concluir que el bloque construido implementa correctamente el protocolo CSMA/CA.

Como trabajo a futuro queda la integración del bloque `Probe Medium` para sensar el canal y la implementación de sockets para lograr que GNU Radio reciba del Sistema Operativo los datos a transmitir.