

# Reutilización de Frecuencias

Cristian Canales - Francisco González

Diciembre, 2017

## Abstract

Se presenta la construcción y los resultados obtenidos de un algoritmo que apunta a la reutilización de frecuencias y que propone un conjunto de estructuras de datos y un *scheduling* como base para la reconstrucción de éste en búsqueda del objetivo planteado.

Se describen las principales funciones que influyen en la ejecución de las pruebas y se presentan los resultados obtenidos y las conjeturas generadas a partir de estos. Por otro lado se proponen posibles mejoras para trabajos posteriores y aspectos necesarios a tener en consideración con el fin de optimizar los resultados generados.

## 1 Algoritmo y funciones implementadas

Para la realización de pruebas se construye un Grafo de forma aleatoria que representa la topología de red de un conjunto de nodos conectados bajo un criterio de distancia predefinido para la conformación de este. Posteriormente se conforma un árbol recorriendo el grafo anterior en BFS, permitiendo distribuir las conexiones entre nodos eliminando conexiones redundantes. Finalmente con los datos generados se crea un *scheduling* para la distribución de espacios de comunicación.

A continuación se describen las funciones implementadas con sus respectivas consideraciones.

1. `generate_graph`: Esta función es la encargada de generar un grafo aleatorio que representa la topología de red base para la creación del *scheduling*.

Se reciben los siguientes parámetros:

- (a) `G`: Instancia inicial del grafo
  - (b) `max_nodes`: Cantidad máxima de nodos para la generación del grafo.
  - (c) `distance`: Valor que determina el tamaño del plano en donde se genera la distribución aleatoria de nodos.
  - (d) `power_level`: Diccionario inicial con los criterios de generación del grafo.
  - (e) `nodes`: Cantidad inicial de nodos.
2. `generate_tree`: Esta función es la encargada de generar un árbol de conexiones que sirve de base para la conformación del *scheduling*.

Este árbol agrega conexiones al nodo root acorde al recorrido BFS realizado, donde todas las primeras conexiones serán en el primer nivel, por ende los con menor distancia respecto al root.

Por otro lado conforme avanza el bfs se generan nuevas conexiones que entregan profundidad al árbol generado.

Se reciben los siguientes parámetros:

- (a)  $H$ : Instancia inicial del árbol generado.
  - (b)  $G$ : Grafo generado
  - (c)  $root$ : Nodo root seleccionado para la conformación del árbol.
3. *scheduling*: Esta función es la encargada de generar la tabla de *scheduling* a partir del árbol de conexiones para designar los tiempos y canales establecidos para llevar a cabo la comunicación entre nodos.

Esta función asigna el primer canal de frecuencia sólo a los primeros hijos del nodo root, y luego distribuye los distintos canales de frecuencia según su profundidad, no asignando pares de conexión en un *timeslot* en donde ya exista comunicación asignada para alguno de los dos nodos.

Se reciben los siguientes parámetros:

- (a)  $H$ : Instancia inicial del árbol generado.
- (b)  $root$ : Nodo root seleccionado para la conformación del árbol.
- (c)  $bfs$ : Resultado obtenido del recorrido del árbol en BFS.
- (d)  $n_{ch}$ : Número de canales designados para el *scheduling*.

## 2 Resultados

Para la obtención de resultados se midieron los tiempos de ejecución de la generación del árbol y del *scheduling* para 15, 30, 40 y 60 nodos. Además se consideraron cuatro canales para la implementación del *scheduling*.

Para las pruebas de 15, 30 y 40 nodos se ocuparon los siguientes parámetros.

- niveles de potencia: 35, 55, 85, 105
- distancia máxima: 300

Para las pruebas de 60 nodos se ocuparon mayores distancias máximas para provocar árboles más profundos

- niveles de potencia: 35, 55, 85, 105
- distancia máxima: 500

Obteniendo los siguientes tiempos de ejecución promedio y largo máximo promedio del *scheduling* con 4 canales de frecuencia.

En la tabla 1 se observan los tiempos de ejecución de la creación de un árbol y en la tabla 2 se observan los tiempos de ejecución de la creación del *scheduling* del árbol creado anteriormente.

Table 1: Tiempos de ejecución para creación del árbol

	15	30	40	60
1	0.0004601478577	0.001034021378	0.00282907486	0.005336046219
2	0.000155210495	0.0009319782257	0.001850128174	0.005040884018
3	0.0004501342773	0.001448869705	0.001394033432	0.003200054169
4	0.0005431175232	0.001314878464	0.002809047699	0.005397081375
5	0.0002708435059	0.001981973648	0.001549005508	0.003618955612
<b>Total</b>	<b>0.0003758907318</b>	<b>0.001342344284</b>	<b>0.002086257935</b>	<b>0.004518604279</b>

Table 2: Tiempos de ejecución para creación del scheduling

	15	30	40	60
1	0.0002009868622	0.0001859664917	0.0002508163452	0.0005569458008
2	0.0000006914138794	0.0002160072327	0.0001919269562	0.0009210109711
3	0.0001499652863	0.0004029273987	0.0002009868622	0.0007932186127
4	0.0001230239868	0.0001730918884	0.0004830360413	0.0007581710815
5	2.91E-05	0.0002670288086	0.0002479553223	0.0008828639984
<b>Total</b>	<b>0.0001007509232</b>	<b>0.000249004364</b>	<b>0.0002749443054</b>	<b>0.0007824420929</b>

## 2.1 Demostración de conjetura

Luego de la generación del árbol de conexiones y el posterior *scheduling*, ver Fig. 3, se requiere realizar el análisis pertinente para implementar la reutilización de frecuencias en el esquema propuesto. Para esto se considera que es posible asegurar que cualquier nodo con más de dos niveles de diferencias respecto a su padre, puede hacer uso de un mismo canal y *timeslot* con otro par de nodos de forma paralela, siempre y cuando la comunicación no se realice a través de su mismo padre.

La propuesta se basa en que a partir de una topología de mayor dispersión se obtiene un árbol de conexiones de mayor profundidad, permitiendo relacionar la distancia entre nodos conforme a la profundidad del árbol generado, favoreciendo de este modo la reutilización de frecuencias.

Como ejemplo de se presenta el árbol de la Fig. 2, en donde se observa que el nodo 11 es capaz de reutilizar cualquier canal de frecuencia en desde el segundo nivel, lo cual incluye a los nodos 3,2,9,17,13,7 y los nodos conectados al nodo root 0,4,6,8,10,14,15,16. Dentro del segundo nivel, el nodo 11 podría también reutilizar frecuencia con el nodo 18 y 19 pero no es recomendado dada la cercanía con su nodo padre.

Además es posible asegurar que el nodo 1 y 12 tienen la capacidad de conectarse con cualquier nodo que no sea su padre, teniendo en consideración que es necesario encontrarse a más de dos niveles de distancia, esto quiere decir que es capaz de reutilizar canales de frecuencia en el mismo *timeslot* con el nodo 0,4,6,10,14,15,16 además del nodo 8, lo cual no es recomendado debido a la cercanía con su padre.

Es posible asegurar la reutilización de dichos canales de frecuencia debido a la forma en la que se crea el árbol, donde los nodos conectados a un nivel inferior son los que se encuentran más alejados

del nodo root en cada subárbol. Esto queda demostrado también en la Fig.?? donde con los mismos colores del árbol se resaltan las posiciones de los nodos respecto al nodo central, y además es posible observar la distancia de los nodos demarcados como capaces de reutilizar frecuencia.

Figure 1: Grafo Gerado

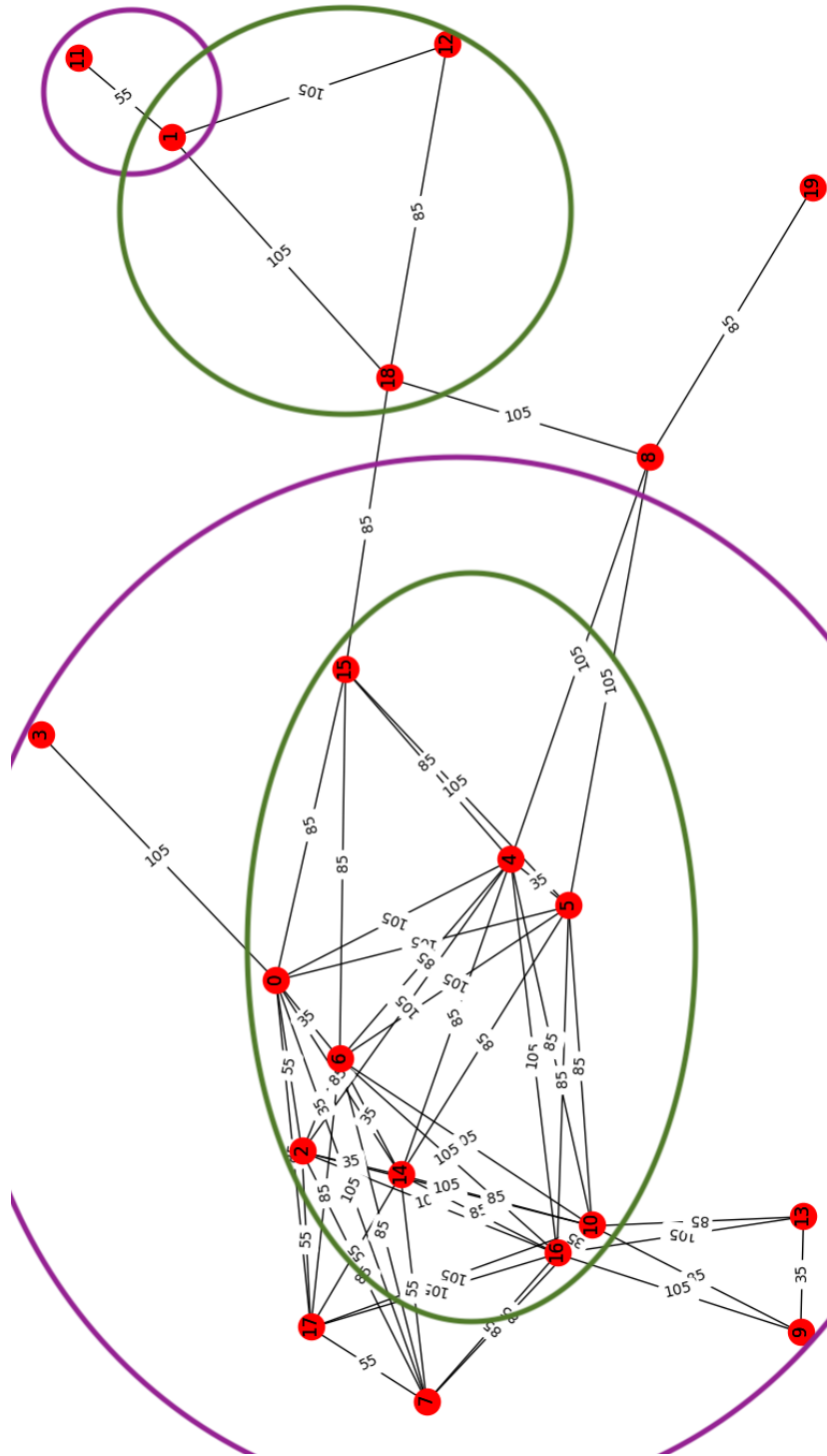


Figure 2: Árbol de conexiones generado

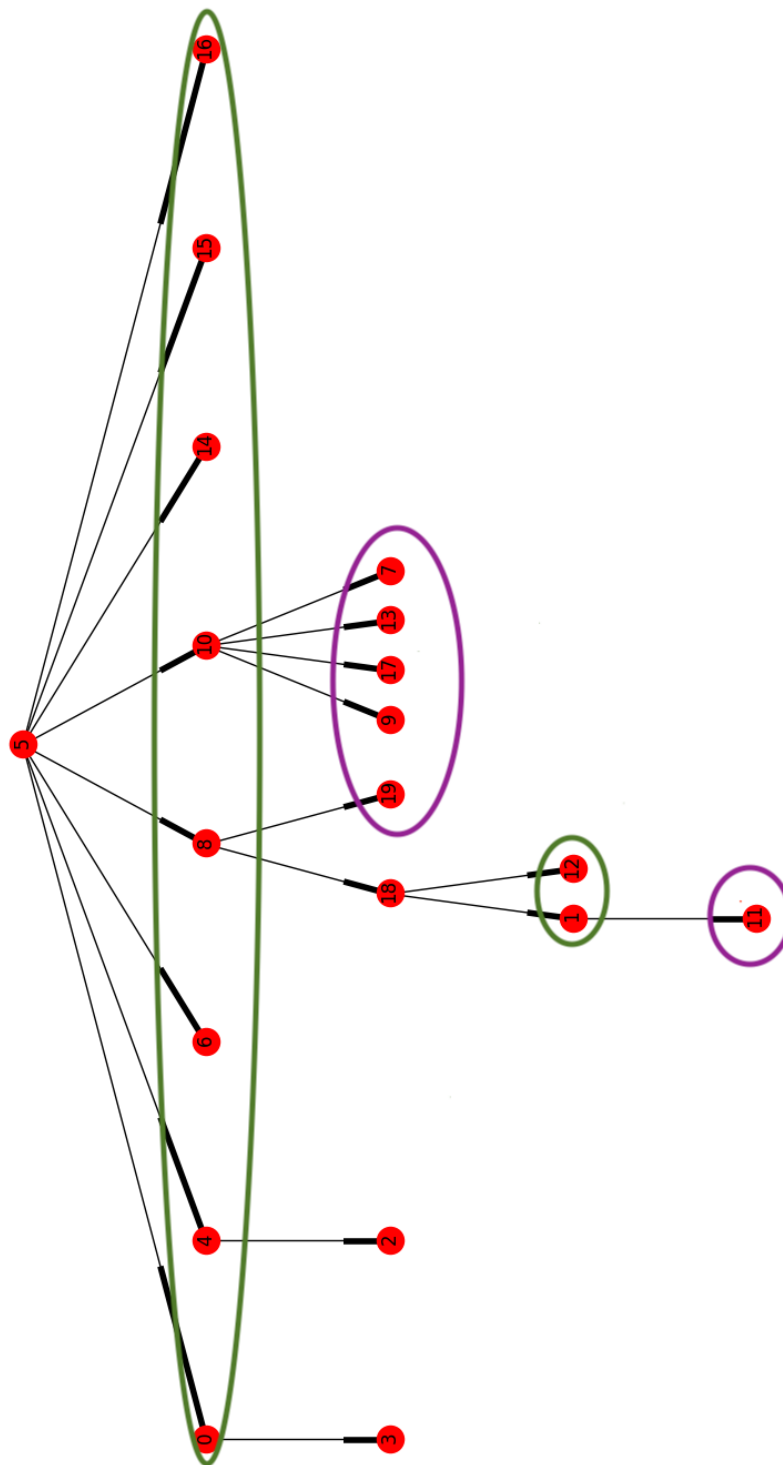


Figure 3: Scheduling generado

Channel/Timeslot	0	1	2	3	4	5	6	7	8	9
Canal: 0	(5, 0)	(5, 4)	(5, 6)	(5, 8)	(5, 10)	(5, 14)	(5, 15)	(5, 16)		
Canal: 1	(None, None)	(0, 3)	(4, 2)	(None, None)	(8, 18)	(8, 19)	(10, 9)	(10, 17)	(10, 13)	(10, 7)
Canal: 2	(18, 1)	(18, 12)								
Canal: 3	(None, None)	(1, 11)								