



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Sanguijuelas

4 de septiembre de 2014

Métodos Numéricos
Trabajo Práctico Nro. 1

Integrante	LU	Correo electrónico
Martin Carreiro	45/10	martin301290@gmail.com
Kevin Kujawski	459/10	kevinkuja@gmail.com
Juan Manuel Ortíz de Zárate	403/10	jmanuoz@gmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

Índice

1. Resumen	3
2. Introducción teórica	4
2.1. Algoritmo de eliminación gaussiana	4
2.2. Matriz banda	5
3. Desarrollo	6
3.1. Matriz Banda	6
3.1.1. Optimización espacial	6
3.1.2. Optimización temporal	6
3.1.3. Ejemplo	6
3.2. Soluciones	10
3.3. Solución GRASP	10
3.4. Solución Random	10
4. Resultados	11
4.1. Resultados Random	11
5. Discusión	15
5.1. Red del Alto Palermo	15
6. Conclusiones	16
7. Referencias	17

1. Resumen

Mediante el manejo de matrices se buscará modelar y solucionar un problema de la realidad. Cómo esta esta compuesta de infinitas variables dicha modelización implicará una inevitable discretización. Es decir trabajar con una cantidad acotada de variables del problema (sólo las relevantes). Si bien el problema en cuestión consiste en decidir cual/es de las sanguijuelas que están adheridas al parabrisa se debe eliminar la modelización del mismo no es trivial. Es más, podría decirse que este proceso es mucho mas complejo y costoso que la solución en sí. ¿Por qué? porque la creación de la matriz que represente al parabrisa y el cálculo de las temperaturas en cada punto, si no se usa buen método, podría llegar a demorar mucho tiempo, tirar overflows o directamente nunca terminar.

Por esto es que a lo largo de este tp haremos mucho foco en como calcular las temperaturas, como optimizar el espacio ocupado por la matriz obtenida y como optimizar lo mas posible todas las operaciones matriciales.

2. Introducción teórica

Dado que la modelización utilizada para este problema es una matriz y que en base a algunos datos que nos dan de ella (posición de las saniguijuelas, dimensiones y temperaturas de los bordes) tenemos que calcular los valores de muchas celdas, lo que hay que resolver es un sistema de ecuaciones. Lo primero que se nos viene a la mente cuando tenemos esto es el método de eliminación gaussiano. Este transforma la matriz de coeficientes en una matriz triangular superior y luego mediante back substitution se pueden obtener todos sus valores.

Veamos ahora más en detalle como funciona cada uno de estos algoritmos.

2.1. Algoritmo de eliminación gaussiana

Es importante en este punto aclarar que para poder hacer estos cálculos la matriz debe ser cuadrada (7). Los pasos a seguir en este algoritmo son:

- Ir a la columna no cero extrema izquierda
- Si el primer renglón tiene un cero en esta columna, intercambiarlo con otro que no lo tenga
- Luego, obtener ceros debajo de este elemento delantero, sumando múltiplos adecuados del renglón superior a los renglones debajo de él
- Cubrir el renglón superior y repetir el proceso anterior con la submatriz restante. Repetir con el resto de los renglones (en este punto la matriz se encuentra en la forma de escalón)
- Comenzando con el último renglón no cero, avanzar hacia arriba: para cada renglón obtener un 1 delantero e introducir ceros arriba de éste sumando múltiplos correspondientes a los renglones correspondientes

Ejemplo:

$$\begin{cases} 2x & +y & -z & = & 8 \\ -3x & -y & +2z & = & -11 \\ -2x & +y & +2z & = & -3 \end{cases}$$

Figura 1: Sistema de ecuaciones de 3x3

$$\begin{cases} 2x & +y & -z & = & 8 \\ & \frac{1}{2}y & +\frac{1}{2}z & = & 1 \\ & 2y & +z & = & 5 \end{cases}$$

Figura 2: Sistema de ecuaciones con ceros en la primera columna

$$\begin{cases} 2x & -2z & = & 6 \\ & \frac{1}{2}y + \frac{1}{2}z & = & 1 \\ & & -z & = & 1 \end{cases}$$

Figura 3: Sistema de ecuaciones triangulado

En esta última imagen podemos observar que el valor de Z ya lo tenemos. Por lo tanto si reemplazamos el mismo en la ecuación arriba obtenemos Y y lo mismo con la primer ecuación.

La complejidad computacional de esto es aproximadamente n^3 . Esto es, el número de operaciones requeridas es n^3 si el tamaño de la matriz es $n \times n$. Por lo tanto puede volverse un cálculo sumamente grande si la matriz es de dimensiones importantes. Por esto decidimos pensar un poco mejor la situación y encontrarle la vuelta para resolverlo en un tiempo razonable.

2.2. Matriz banda

Analizando la matriz del sistema de ecuaciones que se genera a partir de la representación del Parabrisas observamos que presenta las características de un tipo de matriz llamada "banda", la cual se distingue por tener todas las celdas en 0 salvo en la diagonal ensanchada, es decir, que p celdas a la izquierda y q a la derecha de la diagonal a lo sumo tienen algunos o todos los valores distintos de 0. En nuestro caso, al construir la matriz nos basamos en el tipo de celda, si es el borde frío o una sanguijuela el coeficiente respectivo será un 1 en la diagonal igualado a la temperatura correspondiente, pero en el caso de ser una celda vacía en la diagonal irá un -4, a los costados un 1 y representando la fila de "arriba" y "abajo" del parabrisas irán un 1 a m columnas de distancia de la diagonal. Finalmente la matriz resultante quedará con una banda de ancho $m+p$, con m ancho del parabrisas.

Aca podemos ver un ejemplo de una matriz banda:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & 0 & 0 & 0 \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & 0 & 0 \\ 0 & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & 0 \\ 0 & 0 & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} \\ 0 & 0 & 0 & a_{64} & a_{65} & a_{66} & a_{67} \\ 0 & 0 & 0 & 0 & a_{75} & a_{76} & a_{77} \end{pmatrix}$$

Figura 4: Matriz banda 7x7

3. Desarrollo

3.1. Matriz Banda

Como explicamos en la introducción de la matriz banda, llegamos a la conclusión de que la matriz de ecuaciones de la representación del parabrisas tenía forma de la denominada "matriz banda" cuya respectiva banda era de tamaño $m+m+1$ con m siendo el ancho del parabrisas. Este es un hecho importante ya que las matrices banda tienen características especiales de las cuales es posible la optimización temporal y espacial para resolver el sistema de ecuaciones con eliminación gaussiana.

3.1.1. Optimización espacial

El problema de representar la matriz original es que la mayoría de los valores son 0 y solo importan los elementos de la banda de la matriz, por lo tanto una forma de optimizar espacialmente es solo guardar la banda, con lo que se logra reducir considerablemente el espacio en esta representación, ya que suponiendo que se tiene un parabrisas de n filas y m columnas, el tamaño de la matriz original sería de $(n*m)^2$, mientras que con la optimización de matriz banda quedaría de tamaño $(n*m)*(2*m+1)$

3.1.2. Optimización temporal

3.1.3. Ejemplo

por cada posición del parabrisas

```
pos = fila posicion + columna posicion * ancho;
```

```
si en esa posición hay una SANGUIJUELA:
```

```
bandMatrix[pos][ancho] = 1;
bandMatrix[pos][res] = ts;
```

```
si esa posición es borde FRI0:
```

```
bandMatrix[pos][ancho] = 1;
bandMatrix[pos][res] = -100;
```

```
en caso de que esa posición sea VACIA:
```

```
bandMatrix[pos][ancho] = -4;
bandMatrix[pos][res] = 0;
```

```
si la izquierda no es vacía
```

```
bandMatrix[pos][res] -= matrix[i][j-1]->temp;
```

```
else
```

```
bandMatrix[pos][ancho-1] = 1;
```

```
si la de arriba no es vacía bandMatrix[pos][res] -= matrix[i-1][j]->temp;
```

```
else bandMatrix[pos][0] = 1;
```

```

si la de la derecha no es vacía
    bandMatrix[pos][res] -= matrix[i][j+1]->temp;
else bandMatrix[pos][ancho+1] = 1;

si la de abajo no es vacía
    bandMatrix[pos][res] -= matrix[i+1][j]->temp;
else bandMatrix[pos][ancho*2] = 1;

```

	0	1	2	3	4	5	6
0	F	F	F	F	F	F	F
1	F	S					F
2	F						F
3	F			.			F
4	F						F
5	F						F
6	F	F	F	F	F	F	F

Figura 5: Vista de la representación de Parabrisas del ejemplo

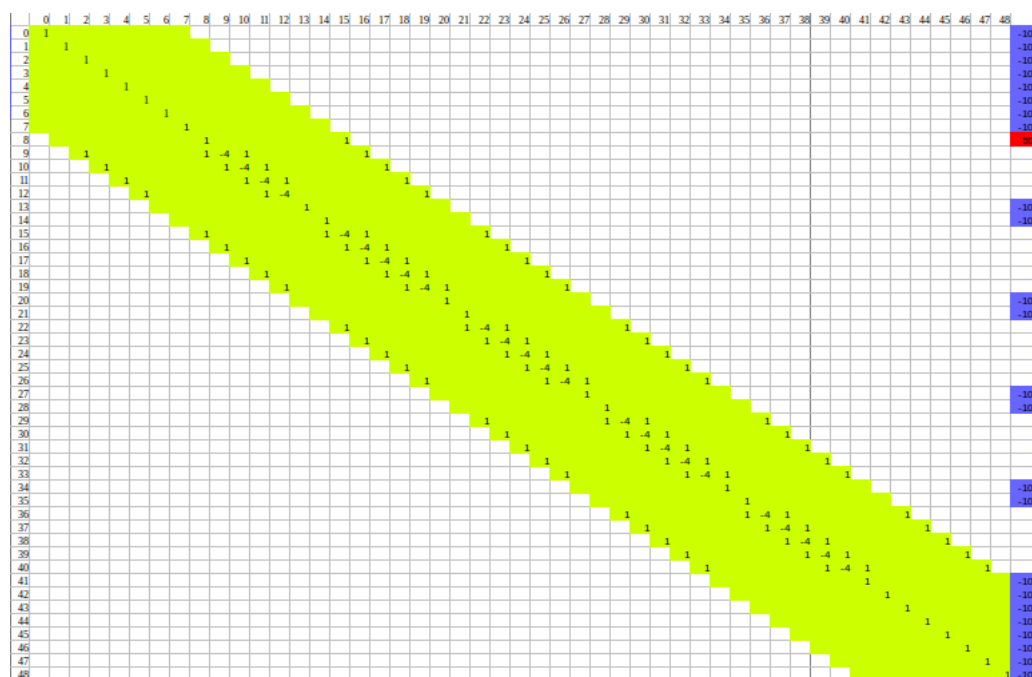


Figura 6: Matriz de ecuaciones del ejemplo

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0							1									-100
1							1									-100
2							1									-100
3							1									-100
4							1									-100
5							1									-100
6							1									-100
7							1									-100
8							1									-200
9							-4	1							1	-400
10						1	-4	1							1	100
11						1	-4	1							1	100
12						1	-4								1	200
13						1										-100
14						1										-100
15						-4	1								1	-400
16	1					1	-4	1							1	0
17	1					1	-4	1							1	0
18	1					1	-4	1							1	0
19	1					1	-4								1	100
20						1										-100
21						1										-100
22	1					-4	1								1	100
23	1					1	-4	1							1	0
24	1					1	-4	1							1	0
25	1					1	-4	1							1	0
26	1					1	-4								1	100
27						1										-100
28						1										-100
29	1					-4	1								1	100
30	1					1	-4	1							1	0
31	1					1	-4	1							1	0
32	1					1	-4	1							1	0
33	1					1	-4								1	100
34						1										-100
35						1										-100
36	1					-4	1								1	200
37	1					1	-4	1							1	100
38	1					1	-4	1							1	100
39	1					1	-4	1							1	100
40	1					1	-4									200
41						1										-100
42						1										-100
43						1										-100
44						1										-100
45						1										-100
46						1										-100
47						1										-100
48						1										-100

Figura 7: Matriz banda del ejemplo

3.2. Soluciones

Para resolver el problema que se nos pide, seleccionar que sanguijuelas matar, pensamos una solución Greedy (7) + Local Search . Esta consiste, a grandes rasgos, en ir matando las sanguijuelas del medio (ya que son las que mayor temperatura generan en el punto crítico) hasta que el punto central este por debajo del 235 C°. Para tener algún parámetro de referencia pensamos en buscar otra solución posible, y así compararlas y chequear que esta sea mejor. Esta solución alternativa consiste básicamente en seleccionar aleatoriamente que sanguijuelas aniquilar. Procederemos ahora a explicar en detalle las implementaciones de estas soluciones.

3.3. Solución GRASP

3.4. Solución Random

En esta solución seleccionamos de nuestro array de posiciones de sanguijuelas (que es el array recibido por parametro, osea con las posiciones sin discretizar) una al azar y la eliminamos. Luego ejecutamos de nuevo el cálculo de las temperaturas y chequeamos si el punto crítico esta por debajo del 235 C°. Si no lo está, elegimos otra al azar y repetimos el proceso hasta que lo esté.

Pseudocodigo:

```
Class Windshield{
    randomSolution(){
        while(!this->isCooledDown()){
            this->randomKill()
        }
    }
    isCooledDown(){
        return (matrix.centerPoint < Ts)
    }
    randomKill(){
        randomRemove(posSanguijuelas)
        this->recalculateTemps()
    }
}
```

Vale aclarar que el recalculateTemps utiliza el metodo band matrix, ya que este es más rápido.

4. Resultados

4.1. Resultados Random

A continuación observaremos una matriz en la que la única sanguijuela que se debe eliminar es la del centro. Esta se encuentra exactamente en el punto central. Adicionalmente hay otras 4 sanguijuelas en las puntas. Matar a estas no soluciona nada ya que la del centro esta aplicando una temperatura constante de 400 C°.

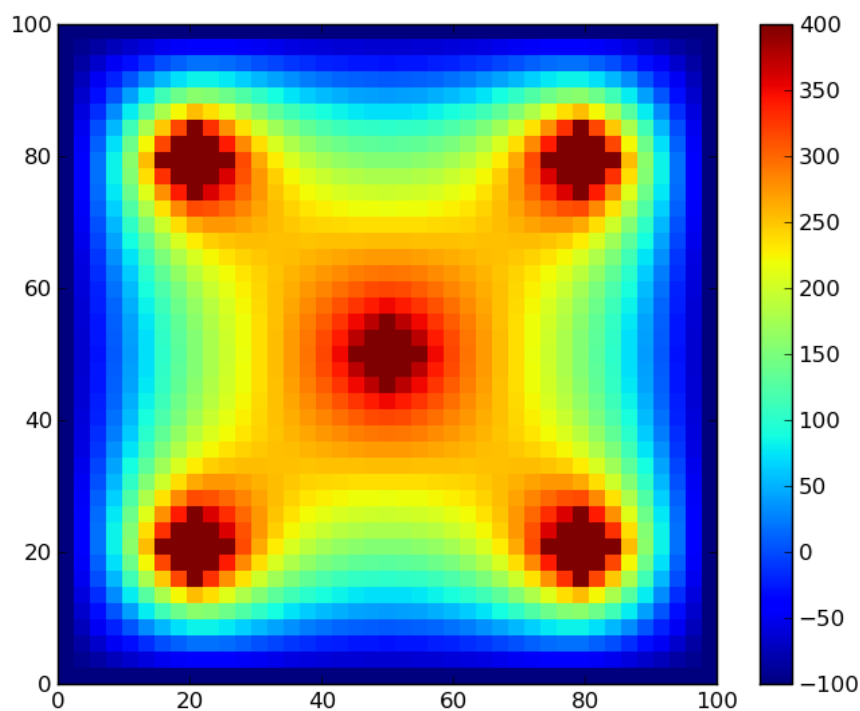


Figura 8: Parabrisas con 5 sanguijuelas

Ahora veamos que obtenemos al aplicarle distintas veces el algoritmo de solución random explicado en el desarrollo (3.4).

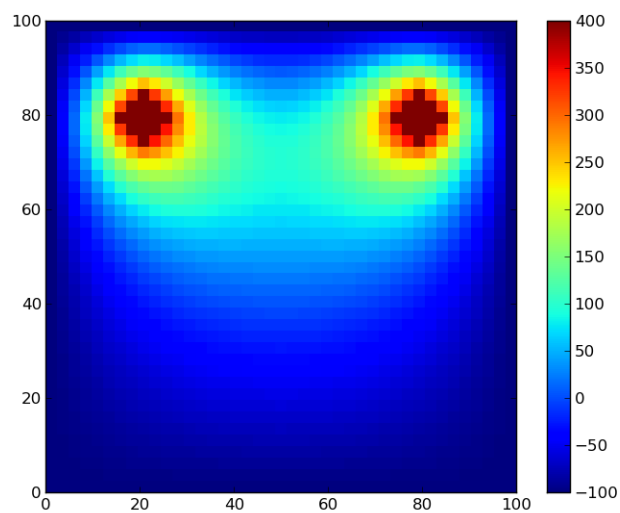


Figura 9: Resultado primer corrida

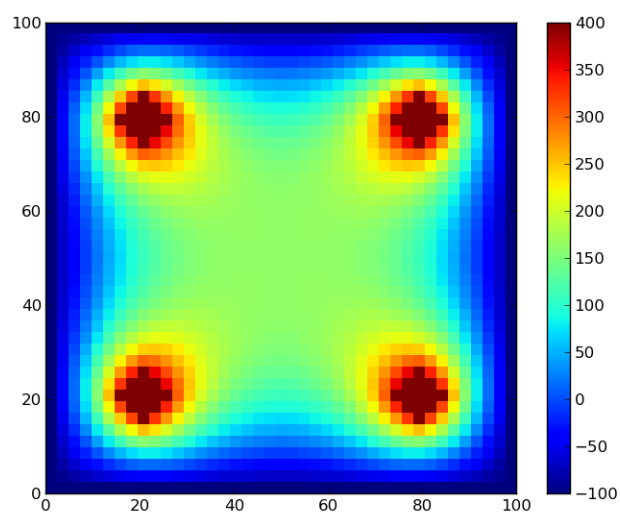


Figura 10: Resultado segunda corrida

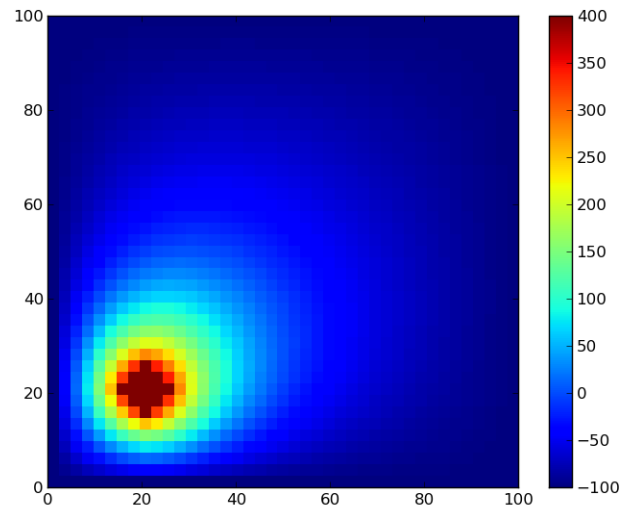


Figura 11: Resultado tercer corrida

Como podemos observar la solución óptima (la que menos sanguijuelas elimina), es la segunda. Pero como esta solución es completamente random en la primera corrida mata 2 sanguijuelas antes de elegir la correcta y en la tercera 4. Sólo en la segunda elije en el primer intento la sanguijuela correcta.

Veamos ahora cuanto demoró cada una de las soluciones. Es importante aclarar que acá el algoritmo utilizado para recalcular las temperaturas fue el gaussiano sin banda matriz. Para que la diferencia de tiempos sea mas clara.

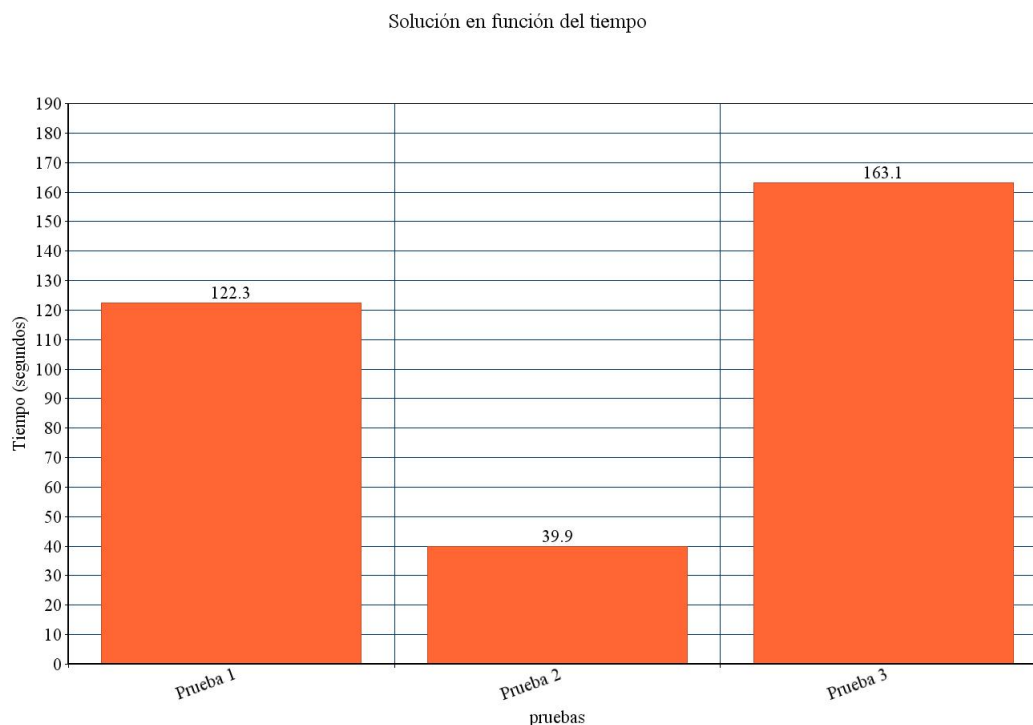


Figura 12: Tiempos de corrida de cada prueba

Acá podemos observar que, como era de esperarse, cuanto más sanguijuelas elimina mas tiempo tarda, ya que debe recalcular las temperaturas.

5. Discusión

Espiral vs centrico

no contar los bordes ni las sanguijuelas para crear la matriz banda

¿porque random?

5.1. Red del Alto Palermo

En esta red se observa una gran interacción con el router, cuya IP es 172.17.0.1. Esto sucede porque el router es el dispositivo por el que pasan en general la mayoría de los pedidos. Un aspecto interesante que se ve en esta comunicación es que no parece haber un pedido fuera de la red, es decir, a otro dominio. Lo que se puede deducir es que el dominio de esta red parece ser 172.17.0.0 /16 donde la IP del router es 172.17.0.1, el cuál se comunica con el resto de los dispositivos bajo este dominio.

6. Conclusiones

Se pudo observar que capturar los paquetes ARP en distintas redes locales da bastante información sobre la red. Es decir, con un análisis estadístico sobre los paquetes ARP *who-has* tomando como referencia las fuentes de información que se pueden definir a partir de las IPs que emiten mensajes y las IPs destinatarias, se puede por ejemplo deducir cuál es el router de la red. Otra manera de verlo es que si ya se sabe cuál es la IP del router, los experimentos sirven para ver que efectivamente el router es el dispositivo más solicitado, como se puede ver en la experimentación que se realizó en el departamento de un miembro del equipo.

Se ejecutó la herramienta implementada en distintas redes, lo que supuso distintos resultados. Lo interesante de esto es detectar nodos distinguidos o un comportamiento especial que hizo que tuviéramos que averiguar ciertas cosas, como sucedió en las capturas de la red del McDonald's en donde aparece una forma de asignar IPs cuando se levantan interfaces. Asimismo, con los histogramas hechos de las fuentes correspondientes se puede ver el grado de actividad de ciertos dispositivos en la red (por lo menos en el momento en que fue realizado el experimento; para mayor precisión se debería realizar una captura más amplia y de más duración).

Como última observación cabe destacar que no se debió participar 'agresivamente' en la red. Es decir, la escucha de los paquetes ARP fue pasiva. Sólo con recopilar información durante un tiempo determinado se puede deducir con bastante confiabilidad qué dispositivo hace de servidor para el resto de los nodos de la red.

7. Referencias

- Una matriz de n por m elementos, es una matriz cuadrada si el número de filas es igual al número de columnas, es decir, $n = m$ y se dice, entonces que la matriz es de orden n
- En matemáticas una matriz se le llama matriz banda cuando es una matriz donde los valores no nulos son confinados en un entorno de la diagonal principal, formando una banda de valores no nulos que completan la diagonal principal de la matriz y más diagonales en cada uno de sus costados.
- Un algoritmo Greedy (también conocido como ávido, devorador o goloso) es aquel que, para resolver un determinado problema, sigue una heurística consistente en elegir la opción óptima en cada paso local con la esperanza de llegar a una solución general óptima. Este esquema algorítmico es el que menos dificultades plantea a la hora de diseñar y comprobar su funcionamiento.