DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 1

Wiretapping

6 de diciembre de 2015 — Teoría de las comunicaciones

Integrante	LU	Correo electrónico
Ladelfa, Hernán Nahuel	318/04	nahueladelfa@gmail.com
González, Sergio Martín	723/10	sergiogonza90@gmail.com

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

Índice

1. Introducción

El objetivo de este trabajo, es poder realizar un estudio sobre distintas redes, para poder realizar un análisis de los distintos **protocolos** que corren en la misma, asi como la posible caracterización de algunas IPs, como **nodos distinguidos**

Para realizar esto, se implemento una herramienta capas de: poder capturar el trafico de una red, ya sea Wifi o Ethernet 802.3, y quedarse solo con la información de los distintos protocolos encontrados, y los distintos hosts (IP/MAC) que intervienen.

La herramienta también permite calcular la entropia, y la probabilidad de aparición de un nodo, en el trafico de la red en un período de tiempo determinado.

Con esta información, el objetivo es realizar un análisis exhaustivo de las redes estudiadas.

2. Primera Consigna: Caputando tráfico

2.1. Fuentes de información

A continuación se presentan las fuentes de información del modelo teórico, que luego se simularan mediante la herramienta implementada.

2.1.1. Protocolos

En primer lugar, la cátedra planteo una fuente de información que nos ayude a distinguir entre los distintos protocolos que se pueden hallar en una red. La fuente de información es la siguiente:

$$S_{t_i;t_f} = \{s_1 \cdots s_n\} \text{ siendo } s_i = p_i.type / p_i \in P \text{ entre los instantes de tiempo } [t_i; t_f].$$

y P esta definida de la siguiente manera:

$$P_{t_i;t_f} = \{p_1 \cdots p_n\} \text{ siendo } p_i \text{ el i-esimo paquete transmitido en la red entre los instantes de tiempo } [t_i, t_f]$$

Entonces, lo que se hace es, realizar una captura de la red, entre los instantes t_i y t_f, y de esto nos quedamos únicamente con los campos *type* del frame Ethernet. Así obtenemos una fuente de informacion donde cada simbolo es un protocolo utilizao en la red.

2.1.2. Nodos

Ahora, debemos idear una fuente de información, que nos permita poder distinguir los hosts involucrados en la captura de trafico, y a su ves, distinguir entre ellos, cuales son los mas concurridos y con mas apariciones en la red.

Para esto, tomemos la fuente de información P vista en el anterior punto, y tratemos de adaptarla para poder obtener una fuente de información que nos permita encontrar hosts en la red, solo basándonos en paquetes ARP. Tomemos el siguiente subconjunto de P:

$$\overline{P}_{t_i;t_f} = \{p_1 \cdots p_n\} \; \forall \; p_i \in \text{P} \; / \; p_i.type = ARP \text{ entre los instantes de tiempo } [t_i; t_f].$$

Con esta fuente de información, lo que hacemos es quedarnos únicamente con los paquetes ARP de toda la captura de trafico. En base a esto proponemos la siguiente fuente de información.

$S_1 = R_{t_i;t_f} = \{r_{ai} \mid r_{ai} = \overline{p}_i[ARP].ip\_origen\} \bigcup \{r_{bi} \mid r_{bi} = \overline{p}_i[ARP].ip\_destino\}$ entre los instantes de tiempo $[t_i; t_f]$.

Esto quiere decir, nos quedamos con las IPs origen y destino de la capa ARP del paquete (del cual sabemos que tiene una, porque los símbolos son tomados de $\overline{P}$). De esta forma, obtenemos una fuente de información donde cada simbolo es un hosts de la red.

Teniendo esto, podemos calcular, tanto la entropía de ambas fuentes, así como la probabilidad y cantidad de información de cada símbolo, tanto para S_1 como para $S_{t_i;t_f}$. La cantidad de información de cada símbolo s, se calcula de la siguiente forma:

$$I(s) = -log_2(P(s))$$

en donde $P(s)$ es la probabilidad de ocurrencia de s, en todo el espacio muestral S, en este caso, todos los símbolos emitidos por S_1 y $S_{t_i;t_f}$. Luego, la entropía de la fuente es la que se obtiene mediante la siguiente formula:

$$H(S) = \sum_{s \in S} P(s) * I(s)$$

2.2. Resumen de la herramienta

Para la simulación de las fuentes de información anteriormente presentadas. Desarrollamos una aplicación utilizando Scapy, la cual escucha pasivamente la red, captura los paquetes y vuelca la información pertinente en una serie de archivos. Los mismos son los siguientes:

- Un archivo que indica: El origen y el destino (MAC) de la capa Ethernet, el "type", y las IPs origen y destino (solo cuando es pertinente)
- Un archivo con la probabilidad de ocurrencia de cada "type", y la entropía de los "type"
- Un archivo con la probabilidad de ocurrencia de cada "host" (ip), y la entropía de los host (solo teniendo en cuenta paquetes ARP)
- Una imagen con un grafo indicando los "request" (who-has) y "replys" (is-at) que se enviaron los "host" entre si (opcional).

2.3. Ejecución

La ejecución del a herramienta debe ser realizada en un entorno Linux, y es necesario tener instalado: *Python 3.0* (o superior), *Scapy*, y si ademas se quiere realizar grafos, es necesario tener instalado *graphviz*.

Para ejecutar la herramienta, se debe abrir una consola en la carpeta "src" adjunta a este informe y ejecutar el comando:

```
$ sudo ./WiretappingTool.py
```

Esto comenzara a capturar, y volcara la información en los archivos configurados como default. Adicionalmente, la aplicación cuenta con ciertos paramentos para personalizar la ejecución de la captura. Por ejemplo:

```
$ sudo ./WiretappingTool.py -f salida.out -t 60 --console --arp
```

Esto realiza lo siguiente:

- (−t o −−timeout) indica un tiempo en segundos para finalizar la captura, en esta caso 60 segundos.
- (−−arp) Captura solo paquetes ARP.
- (−−console) Ademas de volcar el resultado de la captura a un archivo, se muestran los paquetes en la consola, en tiempo real.
- (−f) Vuelca la captura en *salida.out* en lugar del archivo configurado por defecto (out/sniff.out).

Si ademas de la captura, se desea realizar un grafo con los nodos de la red, junto con los "request" y "replys" realizados por los mismos, se debe pasar el parámetro "–graph". Si se hace esto, al terminar la ejecución, la imagen con el grafo se abrirá al instante. Para ver con detalle cada uno de los parametros que dispone la aplicacion, se puede utilizar:

```
$ sudo ./WiretappingTool.py -h
```

3. Segunda Consigna: Gráficos y Análisis

Para poder simular las fuentes de información antes mencionadas, lo que hicimos fue correr la herramienta implementada en diferentes redes con distintas topologias. Todas las capturas fueron realizadas en redes Wifi, ya que en redes Ethernet "switcheadas" no se pueden ver los mensajes ARP *"is-at"* que no son hacia nuestro host, debido a que estos son unicast, y no es posible capturarlos debido a que no llegan a la interfaz de la pc que realiza la captura.

En primer lugar, queremos presentar las redes que utilizamos, y ademas mostrar el grafo que genero nuestra herramienta, con los mensajes ARP "*who-has*" y "*is-at*"

3.1. Red Laboral

Esta red es la red del trabajo de uno de los integrantes del grupo. Dicha red cuenta con diversos routers Wifi, por lo que es difícil saber de cuantos hosts es posible capturar el trafico. Por los datos obtenidos se puede estimar que al momento de realizar la captura, se encontraban conectados alrededor de 53 hosts. A continuación se muestra el grafo resultante de la captura.

Como se puede ver a simple vista en la figura 1, hay 3 IPs muy concurridas: 192.168.28.131 (A), 192.168.28.143 (B), y 192.168.29.254 (C). Debido a que la IP (C) termina con el numero 254, y que ademas es el nodo mas concurrido de la red (al momento de esta captura), podemos intuir que esta IP es el router al que esta conectada la PC que capturo el trafico. Luego, esta captura se realizó desde una maquina virtual corriendo Linux, hosteada en una pc con Windows 7, y las IPs (B) y (A) son el host real y la Virtual respectivamente.

3.2. Red Starbucks - FibertelZone

La siguiente captura se realizo también desde una maquina virtual corriendo Linux, desde una notebook corriendo Windows 8.1, en una red (Wifi) de Starbucks. De todas maneras, al momento de conectarse, la red Wifi pertenecía a la red de FibertelZone, sin contraseña, por lo que es muy probable que los hosts no se encuentren solo en el establecimiento, sino también en los alrededores.

Hay varias cosas a destacar en el grafo. Primero hay 2 redes visibles, una con IPs privadas (10.0.0.0) (A), y otra con IPs publicas de la red 169.254... (B) (no es posible determinar la mascara debido a que no hay suficientes IPs). Otra cosa a destacar, que es bastante interesante, es que hay paquetes ARP "who-has" desde IPs de la red (A) hacia una IP que podría legar a ser la IP broadcast de la red (B). Si esto fuera así, habría hosts preguntando por la dirección MAC de una IP broadcast, lo cual es bastante absurdo. Ademas el hecho de que esta IP no responda los "who-has", aumenta la posibilidad de que sea una IP broadcast. De todas maneras, al no saber la mascara de esta red, esa IP podría no ser broadcast. Por ejemplo si la mascara fuera 255.254.0.0, entonces la IP 169.254.255.255 seria una IP utilizable por algún host o un router, y no una IP broadcast.

Luego, tambien se puede visualizar paquetes "who-has" desde la ip 0.0.0.0, en ambas redes. Estos paquetes ARP, luego de investigar un poco, son denominadas Gratuitous ARP[1], las cuales sirven (entre otras cosas) para que un host verifique que no haya otro dispositivo usando su propia IP.

[1] https://wiki.wireshark.org/Gratuitous_ARP

Figura 1: Red Laboral

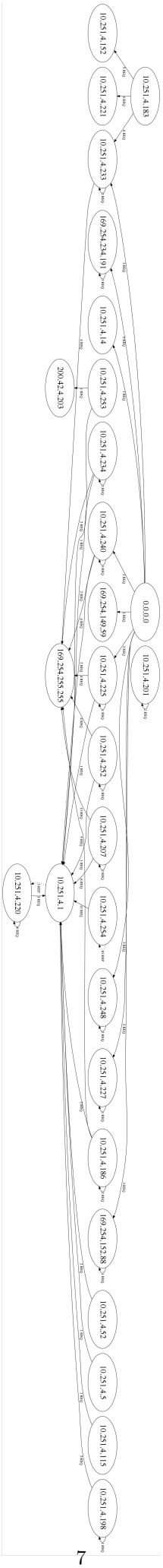

Figura 2: Red Starbucks - FibertelZone

3.3. Cantidad de información por nodo

3.3.1. Red Laboral

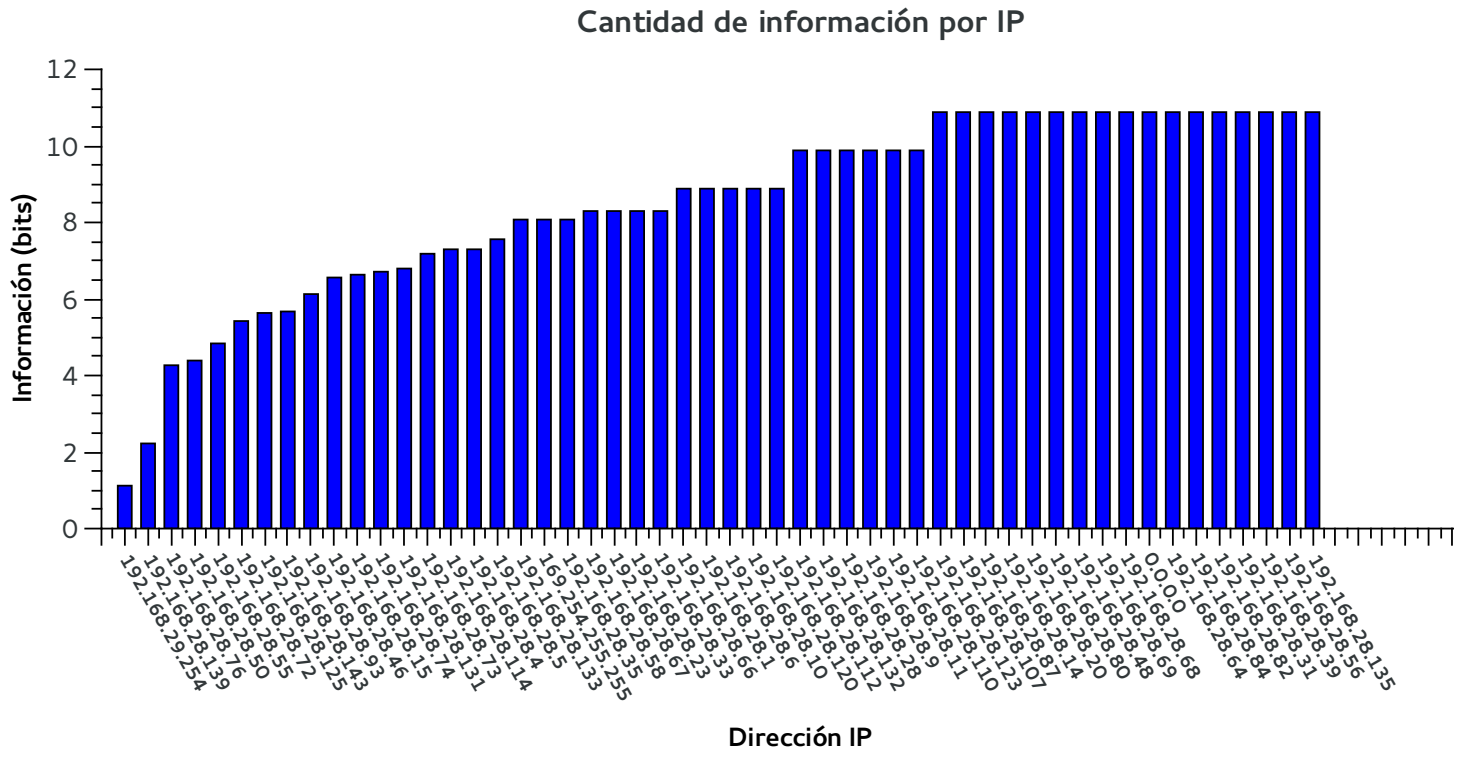

Figura 3

3.3.2. Red Starbucks

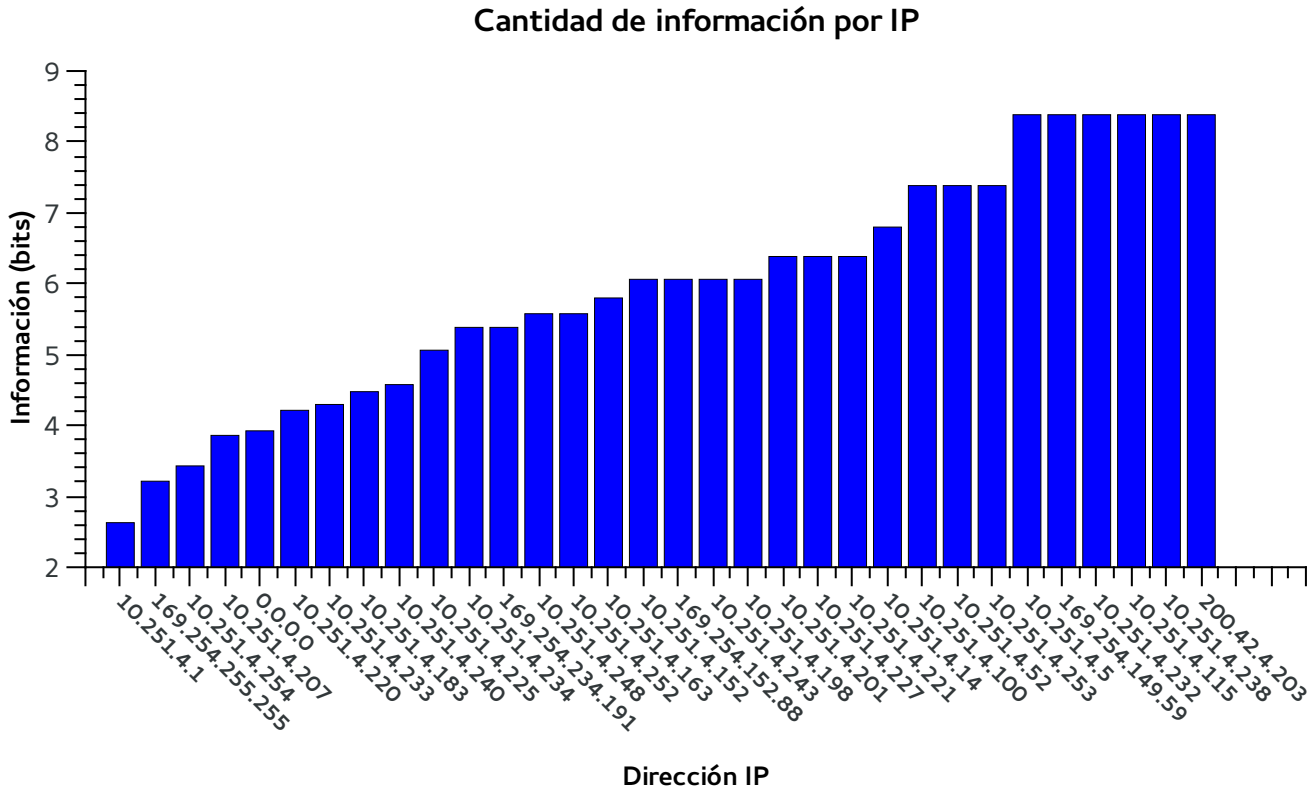

Figura 4

3.4. Paquetes capturados de cada protocolo

3.4.1. Red Laboral

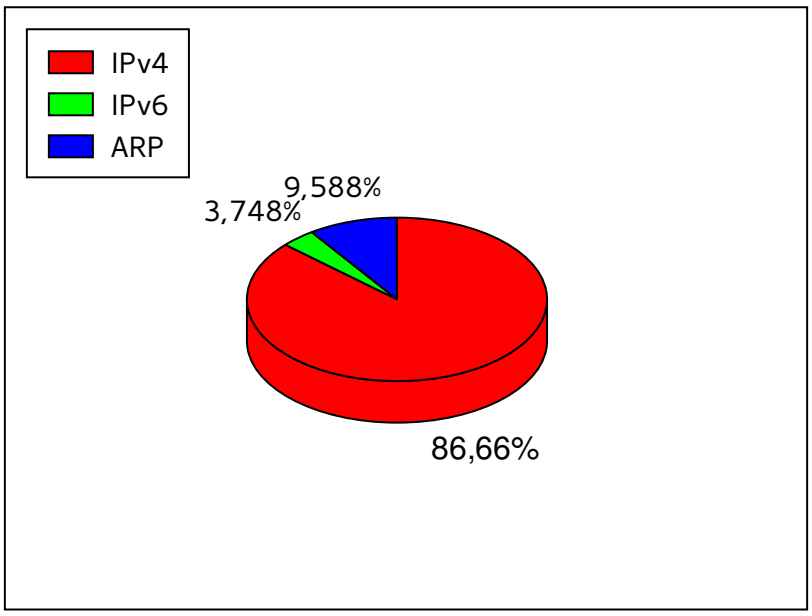

Figura 5

3.4.2. Red Starbucks

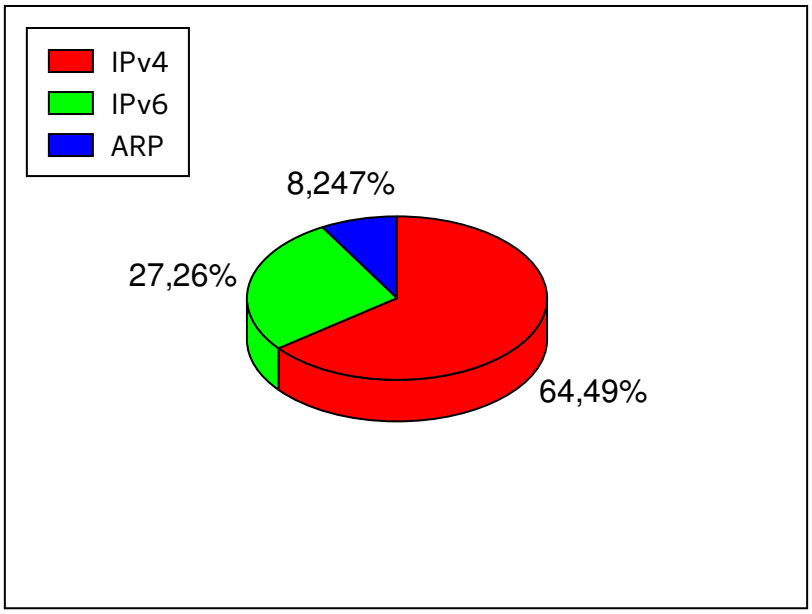

Figura 6

3.5. Variación de la etropía a lo largo del tiempo

3.5.1. Red Laboral

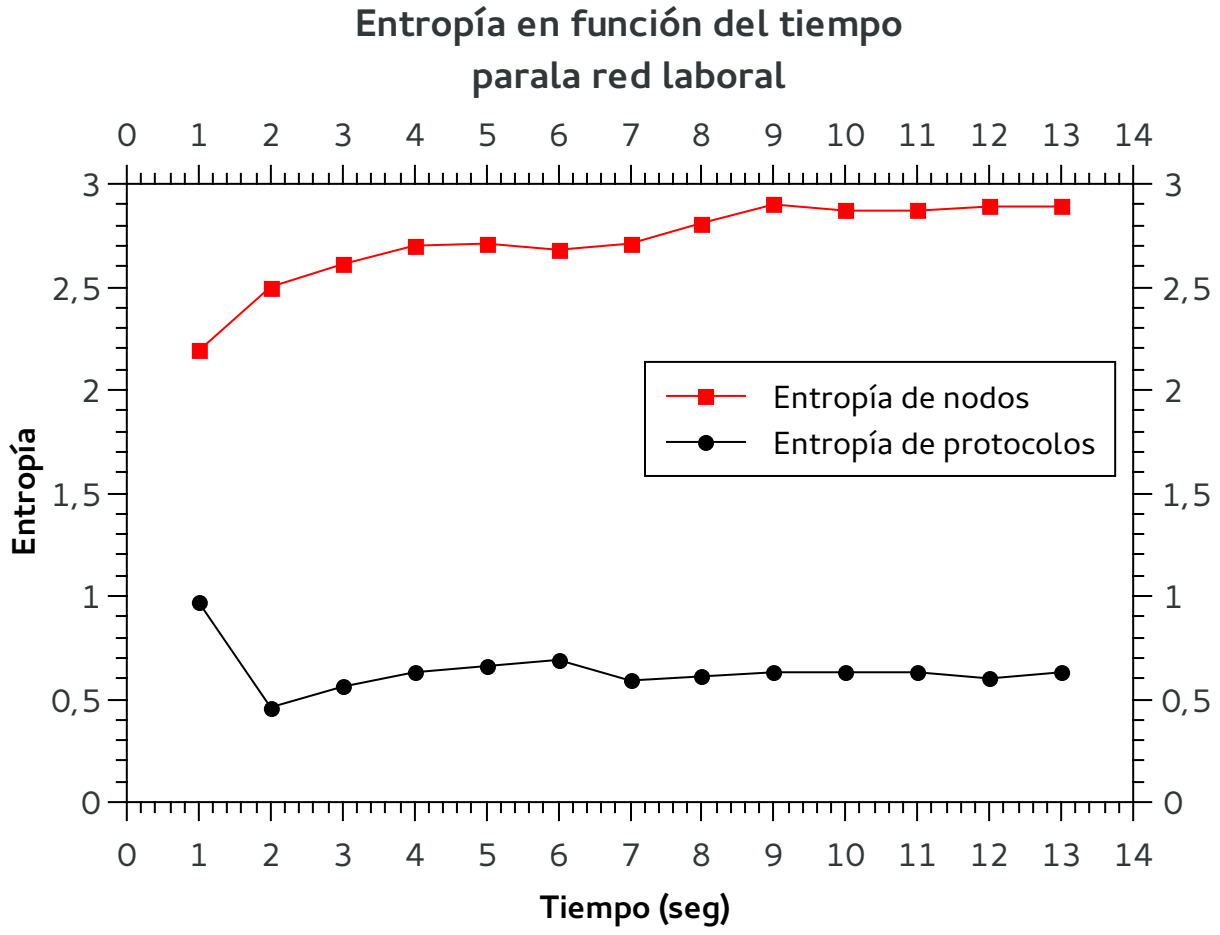

Figura 7

3.5.2. Red Starbucks

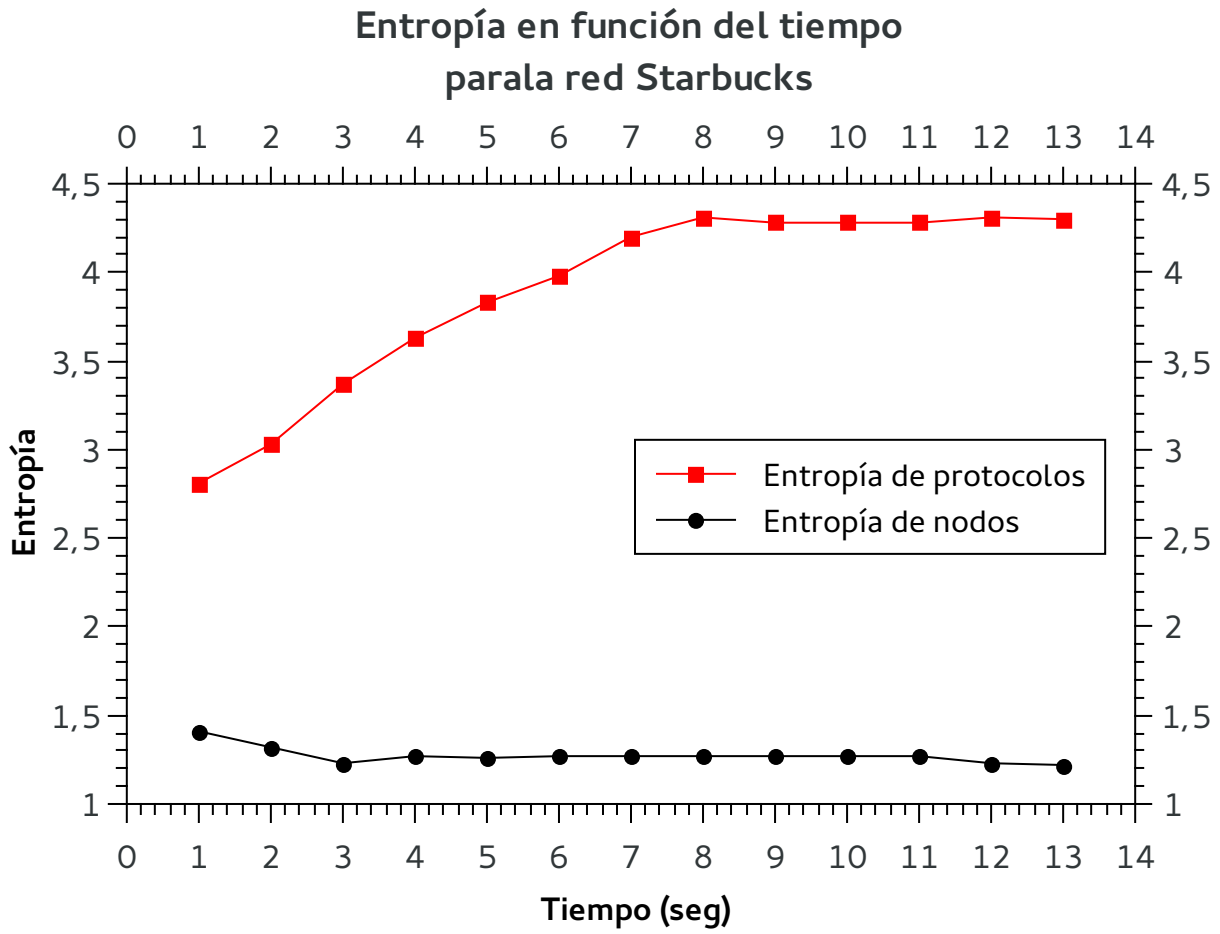

Figura 8

4. Conclusiones