DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

TP 1: Wiretapping

Teoría de las Comunicaciones
Segundo Cuatrimestre de 2016

Integrante	LU	Correo electrónico
Axel Straminsky	769/11	axelstraminsky@gmail.com
Jorge Quintana		jorge.quintana.81@gmail.com
Florencia Zanollo	934/11	florenciazanollo@gmail.com
Luis Toffoletti	827/11	luis.toffoletti@gmail.com

Índice

1. Introducción

El objetivo del presente trabajo es descubrir la topología de distintas redes utilizando captura de paquetes ARP. Para la clasificación de los datos así obtenidos se modelaran por cada red dos fuentes de memoria nula, que identificaremos como S y $S1$. Para las fuentes S se distinguirán los paquetes broadcast vs. los paquetes unicast y para las fuentes $S1$ los símbolos se distinguirán basados en las direcciones IP de los orígenes de los paquetes "Who-Has" ARP.

ARP es un protocolo de capa 2.5 que se encarga de traducir direcciones IP (Nivel de red) a direcciones físicas de los dispositivos o "MAC addresses" (Nivel de Enlace). Este protocolo distingue dos tipos de mensajes, "Who has" e "Is At".
"Who has" son típicamente mensajes de "pedido" (request) enviados a toda la red (Broadcast) preguntando a los dispositivos, identificados por MAC address, quién poseé cierta dirección IP.
"Is At" son mensajes de "respuesta" (reply) enviados a un sólo nodo (unicast) que es el nodo que efectuó el pedido, indicando que el dispositivo con la IP buscada se encuentra en la dirección física que envia la respuesta.

Un dispositivo comunicándose en una red a nivel capa de enlace, necesita conocer la dirección MAC del dispositivo con el que desea comunicarse, pero el protocolo IP utiliza direccionamiento por dirección IP. Para traducir de un tipo de direccionamiento al otro de manera eficiente, los dispositivos mantienen una tabla ARP que "cachea" la información. Estas tablas ARP son actualizadas cada cierto tiempo, lo que genera los mensajes de protocolo ARP que capturaremos.

La distinción entre tipos de paquetes que soporta el protocolo ARP nos conduce de forma natural a la primera distinción entre símbolos que utilizaremos para modelar la fuente S, que distinguirá entre paquetes de tipo Broadcast y paquetes de tipo Unicast.

Para el modelado de la fuente $S1$ el criterio de distinción entre los símbolos de la fuente se justifica matemáticamente de acuerdo a la cantidad de información que cada símbolo trae aparejado y su comparación con la entropía total del sistema.
La cantidad de información que aporta un evento E que ocurre con probabilidad $P(E)$ se define como

$$I(E) = log\frac{1}{P(E)}$$

Para calcular la cantidad promedio de información de una fuente de memoria nula S, tenemos que cuando el simbolo s_i ocurre, obtenemos una cantidad de información

$$I(s_i) = log\frac{1}{P(s_i)}$$

y la probabilidad de que esto ocurra es directamente $P(s_i)$ con lo cual la cantidad promedio de información por cada símbolo de la fuente S será

$$H(S) = \sum_S P(s_i)I(s_i)$$

A esta cantidad se la conoce como la entropía de la fuente S: $H(S)$.
La primera conclusión que se puede sacar de la definición es que los eventos que más información aportan son aquellos con menor probabilidad de ocurrir.

Es claro que en nuestro modelo no estamos trabajando con fuentes de memoria nula ideales, sino que estamos utilizando redes reales para modelar las mismas, con lo cual las probabilidades serán en realidad estadísticos obtenidos en base a la experimentación (captura de paquetes). Los estadísticos utilizados serán el ratio entre la cantidad de ocurrencias de un evento y el total de los eventos capturados, por esta razón y con la intención que el estadístico sea representativo de la probabilidad de ocurrencia de los eventos se efectuarán capturas durante intervalos de tiempo mayores a diez minutos.

2. Desarrollo

Para el presente trabajo práctico se desarrolló un programa en Python utilizando la libreria Scapy que captura los paquetes que escucha la interfaz definida y filtra los mismos quedandose solamente con aquellos que son paquetes del protocolo ARP, el mismo programa acepta como parámetros la ruta de un archivo en formato ".pcap" que puede ser generado por medio de capturas anteriores o utilizando software alternativo como Wireshark.

Tenemos dos variantes del programa, uno para cada fuente explicadas más adelante. Con los datos obtenidos por el programa se calculan la entropía y la cantidad de información de cada símbolo en cada fuente.

Adicionalmente se utilizaron programas auxiliares para generar los gráficos, se crearon archivos dot por medio de scripts en python y luego se graficaron mediante GraphViz.

Se capturaron redes de distintos tamaños utilizando distintas tecnologías a nivel enlance: Switched Ethernet y Wireless LAN

2.1. Fuente S

Esta fuente binaria está compuesta por los símbolos $s_{Broadcast}, s_{Unicast}$.

2.2. Fuente S1

Para S1 teníamos varias opciones dentro de los paquetes ARP. Podíamos ver los paquetes Who-Has o Is-At, así como también podíamos centrarnos en source o destino. Es decir, cuatro combinaciones. Para decidirnos por una de ellas lo que hicimos fue experimentar con todas y analizar los resultados.

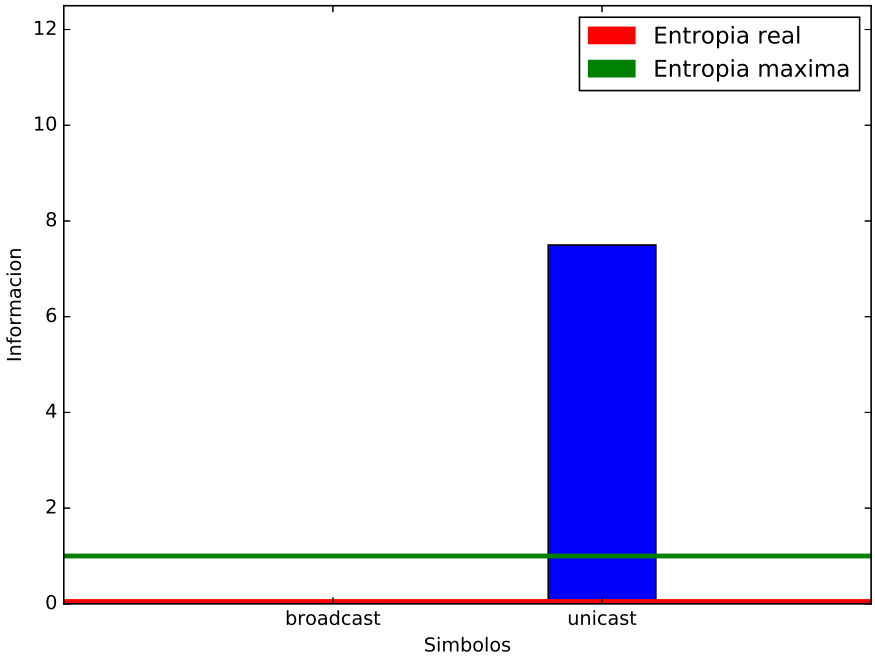
[EXPLICACIÓN DE CON CUÁL NOS QUEDAMOS Y POR QUÉ]

[SIMBOLOS DE LA FUENTE]

3. Resultados

3.0.1. Red corporativa de empresa de Telecomunicaciones - Ethernet switcheada

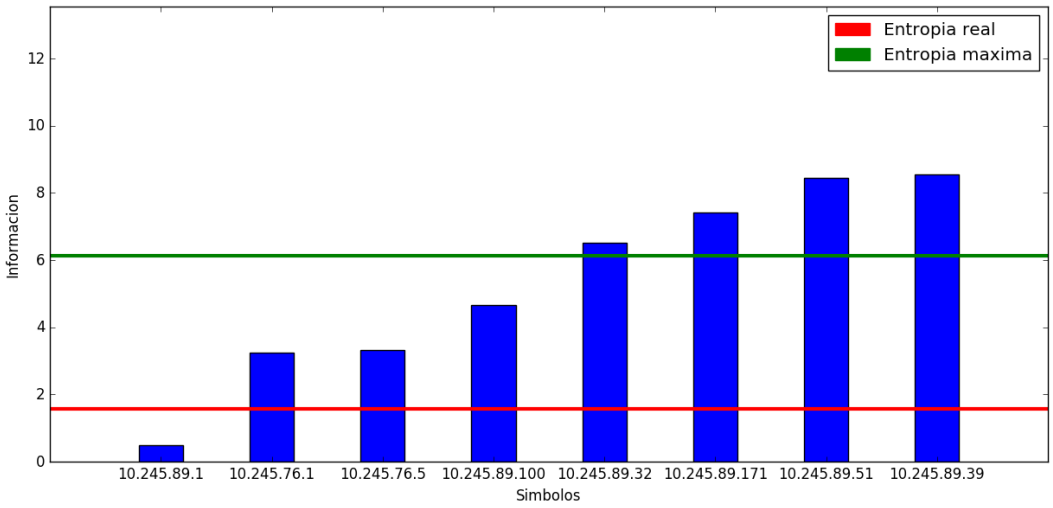
Datos obtenidos con la fuente S:



Como se puede observar hay muchísimos más broadcast que unicast, es por ello que el último dá más información. Las probabilidades son las siguientes: broadcast 0.99, unicast 0.01. La entropía es casi nula

Los broadcast son visibles para todos los dispositivos, mientras que los unicast no. Al ser una red grande, nosotros estamos viendo los broadcast de todos los hosts y sólo nuestros unicast (respuestas enviadas al host con el cuál se hizo la captura), es por esto que la diferencia es tan grande.

Datos obtenidos con la fuente $S1$:



3.0.2. McDonalds - wireless LAN

3.0.3. Red doméstica - wireless LAN

3.0.4. Red corporativa de empresa de Telecomunicaciones - wireless LAN

4. Conclusiones

5. Referencias

1. http://www.secdev.org/projects/scapy/

2. http://www.wireshark.org