DEPARTAMENTO
DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 1

ARP Sniffing

Teoría de las Comunicaciones
Primer Cuatrimestre de 2017

Integrante	LU	Correo electrónico
Borgna, Agustín	079/15	aborgna@dc.uba.ar
Gonzalez Benitez, Juan	324/14	gonzalezjuan.ab@gmail.com
Lancioni, Gian Franco	234/15	gianflancioni@gmail.com
Vazquez, Cristian	056/10	(cristianvazquez4@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

A modo muy general, con el objetivo de analizar distintas redes locales, vamos a modelar ciertos comportamientos en el tráfico como fuentes de Teoría de la Información y poder extraer conclusiones concretas partiendo de las herramientas conocidas, particularmente los conceptos de entropía de la fuente e información de un símbolo en dicha fuente, para estudiar dichas fuentes.

1. Introducción

1.1. ARP

En particular nos interesa analizar las interacciones que se dan entre hosts de la red con el fin resolver direcciones de capa de red hacia capa de enlace, usualmente IP a MAC. Estas interacciones son propias del *ARP (protocolo de resolución de direcciones)* [1], el cual cumple un rol crucial en lo que conocemos como *internetworking*.

Se trata de un protocolo de *request* y *response*, implementados a partir de paquetes *'who-has'*, que preguntan a un dominio de broadcast por la *MAC address* de una IP particular, y paquetes *'is-at'* que responden de manera unicast al emisor del paquete 'who-has'.

El tipo del paquete se determina por el campo de operación de dicho paquete, del cual haremos amplio uso.

Por supuesto, dicho protocolo no se ejecuta cada vez que se quiera efectuar una transmisión, sino que los resultados se almacenan temporalmente una *cache* para agilizar tiempos.

1.2. Herramientas y resolución de primer etapa

Cada una de las capturas de red que hicimos (una por cada integrante del grupo), se hizo con el programa *TShark*, versión *terminal-based* del packet sniffer *Wireshark*.

Los paquetes obtenidos los manipulamos con el framework *Scapy* de *Python* que nos permite acceder a los campos de cada uno y, por ejemplo, determinar si tiene capa *ARP* y (de ser así) su destino buscado.

1.2.1. Primer ejercicio

Lo primero que se nos pide es modelar cada red como una fuete de información binaria S de memoria nula con símbolos en $\{S_{unicast},\ S_{broadcast}\}$.

Para implementar la fuente simplemente contaremos la cantidad de paquetes p tales que $p.dst =$ 'ff:ff:ff:ff:ff:ff' (i.e la dirección MAC de destino de broadcast). Dicha implementación está en *entro.py* y se encarga de imprimir por pantalla la entropía de la fuente y las ocurrencias de cada símbolo para una secuencia dada de paquetes.

Por lo tanto dicha fuente se abstrae de la identidad de los nodos de la red y los trata indiscriminadamente como emisores de símbolos según la dirección de capa de enlace de sus paquetes.

A modo analítico, la fuente S de cada red permite entender de qué manera se comunican los hosts de la red.

Por ejemplo, al tratarse de conexiones predominantemente dirigidas, resultaría extraño que los paquetes de broadcast fueran mayoría en el tráfico de la red. Y en caso contrario, la fuente haría visibles escenarios particulares de topologías o comunicaciones entre hosts.

1.2.2. Segundo ejercicio

El segundo ejercicio consistió en modelar una nueva fuente de información S_1 de memoria nula, con el objetivo de distinguir, en lugar de tipos de destinos como lo hacía S, los propios nodos (hosts) de la red.

Dicha distinción se hizo a partir de los paquetes *ARP*, y la implementación se encuentra en *distinguidos.py*.

Lo que hicimos fue considerar como símbolos las *IPs* de request de los mensajes 'who-has'. De esta manera, los símbolos con menos información son aquellos más solicitados.

Entonces los nodos distinguidos, con la finalidad de exponer aquellos más 'importantes' en la red, los consideramos como aquellos símbolos s tales que $I(s) < H(S_1)$ siendo $I(s)$ la información del nodo s en la fuente S_1 y $H(S_1)$ la entropía de la fuente.

Siguiendo con esta idea, es de esperarse que los nodos más solicitados, como los *default gateways* aparezcan siempre entre los distinguidos.

[1] https://en.wikipedia.org/wiki/Address_Resolution_Protocol

2. Mediciones realizadas

Ahora mostraremos los datos obtenidos en todas las mediciones, considerando las herramientas obtenidas en los dos ejercicios de la primera etapa mas un conjunto de gráficos que aporten nuevos datos para poder analizar.

2.1. Red 'Tatooine-lan'

Esta medición se trata de una red Wi-Fi doméstica de 4 dispositivos activos con un tiempo de escucha de 20 minutos. Dos de las máquinas estaban usando servicios de streaming de video.

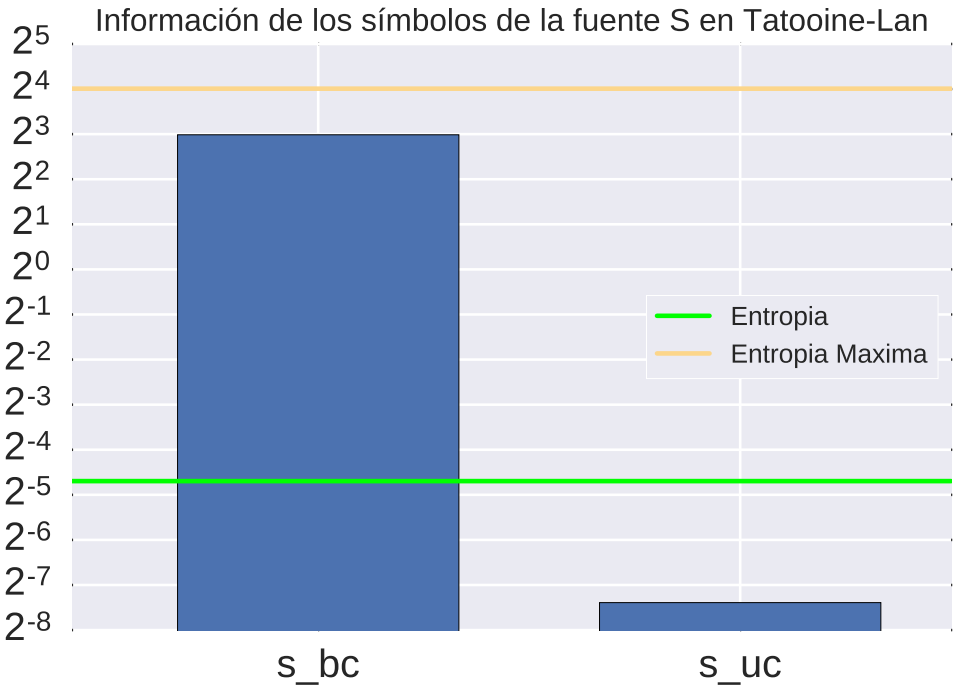
Mientras se realizaba la medición también se borraron los registros (comando 'arp -d $< host\_ip >$' en linux) la caché ARP en el dispositivo cuya IP pública es 190.168.0.14 y privada 192.168.0.12 con el objetivo de forzar un handshake en dicho protocolo con el default gateway de la red.

2.1.1. Fuente S

En la medición se contaron entre todos los paquetes del archivo *.pcap* con la herramienta el ejercicio 1:

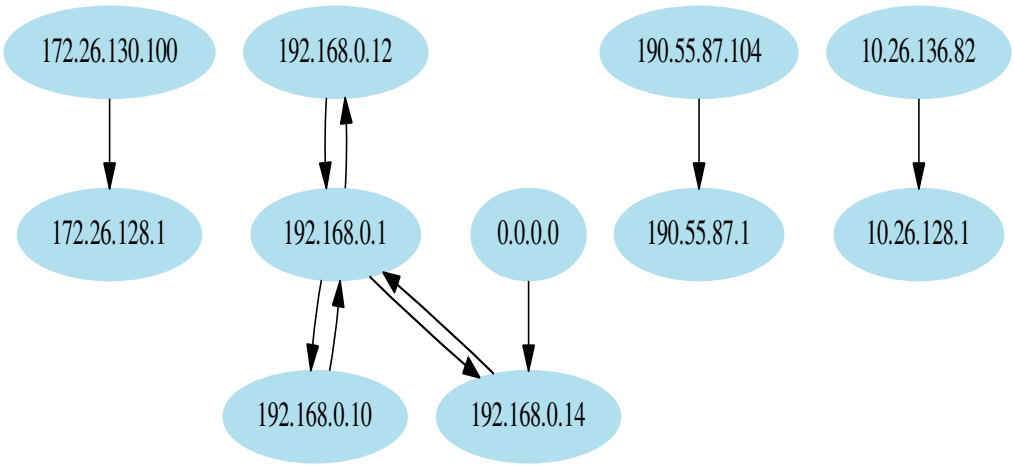
- 69834 paquetes unicast
- 289 paquetes broadcast (p.dst = 'ff:ff:ff:ff:ff:ff')
- 70123 paquetes en total

Dejando una entropía $H(S) = 0{,}03858550573957047$ y la siguiente distribución en la información de los símbolos $S_{broadcast}$ y $S_{unicast}$:



2.1.2. Grafo de conectividad de la red

Con el fin de analizar de mejor manera la topología de la red también armamos un grafo entre los nodos de la red que muestre el tráfico de paquetes ARP sensado. Cada nodo está asociado a una IP y un eje entre una IP_1 y IP_2 significa que se sensó un paquete *'who-has'* emitido por IP_1 consultando por la dirección MAC de IP_2.



Se puede notar que ninguno de los otros dispositivos mecionados, al margen de aquel para el cual flusheamos la caché ARP, hizo un request de tipo *'who-has'* durante el sniff usando su IP pública. Dicho comportamiento se repite en otras redes no privadas aledañas en las cuales un único nodo envía un paquete *'who-has'* a una IP con el último octeto en 1 (candidatos a routers).

Pero, por otro lado, se puede ver buena actividad en una red que parece corresponderse con el rango 192.168.0.0/16 reservado para IPs privadas.

2.1.3. Información y nodos distinguidos

Haciendo uso de la herramienta del ejercicio 2 para distinguir nodos de la red respecto de sus respectivos símbolos en la fuente S_1 en caso de que aporten menor información que la entropía de dicha fuente, encontramos únicamente al nodo 192.168.0.1 como nodo distinguido.

Las direcciones con los dos últimos octetos en 0.1 en redes 192.168.0.0/16 son comunes para IPs de interfaces de routers. Lo que significaría que nuestro código fue capaz de encontrar al menos un default gateway, particularmente por el que más nodos consultaron.