Teoría de las Comunicaciones

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Trabajo Práctico 1

Wiretapping

Grupo ?

Integrante	LU	Correo electrónico
Dabbah, Julián	15/09	djulius@gmail.com
Fernández Abrevaya, Victoria	710/10	vabrevaya@gmail.com
González, Sergio	723/10	sergiogonza90@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Índice

Resumen

Resumen:
Palabras clave: ARP, Teoría de la Información, Entropía, Scapy, Nivel de Enlace.

1. Introducción

El presente trabajo práctico tiene como objetivo estudiar el comportamiento de una red a nivel de enlace (analizando principalmente el vínculo de este nivel con la capa superior). Para ello vamos a observar el funcionamiento del protocolo ARP dentro de una determinada red: buscamos caracterizar la misma observando solamente los mensajes enviados a través este protocolo. A su vez, relacionaremos la red estudiada con los conceptos aprendidos dentro del campo de la teoría de la información: utilizando la entropía de una red local (tomando como fuente de información el modelo explicado más adelante) vamos a caracterizar el funcionamiento de la misma, centrándonos en cuales son los nodos distinguidos en ella.

ARP (*Address Translation Protocol*) es el protocolo encargado de mapear direcciones de nivel 3 a direcciones de nivel 2 (nivel físico). Comunmente se utiliza para traducir direcciones IP a direcciones MAC, aunque el protocolo permite asociar otros tipos de direcciones. En este trabajo nos concentraremos solamente en el caso mencionado.

Para construir asociaciones IP-MAC se envía un paquete ARP consistente, entre otros, de los siguientes campos:

- Tipo de mensaje (*"who-is"* / *"is-at"*)
- IP del host que envía el mensaje
- MAC del host que envía el mensaje
- IP del host destino
- MAC del host destino

Si la máquina A quiere comunicarse con B, conociendo solamente su dirección IP, deberá enviar un mensaje ARP de manera *broadcast*, del tipo *who-is*, con la IP de B dentro del campo IP-destino. El host cuya IP se corresponda con el mensaje contestará entonces con un ARP del tipo *is-at*, indicando su dirección a nivel de enlace. Este último mensaje se envía solamente a la IP que desea conocer la MAC, por lo cual sólo puede ser escuchado por la máquina que preguntó. A través del envío de estos paquetes, cada host puede armar dinámicamente una caché de asociaciones IP-MAC. El hecho de que los mensajes *who-has* sean enviados de manera *broadcast* nos permitirá escucharlos fácilmente, para poder sacar conclusiones a partir del uso que se haga de ellos. En general las tablas ARP tienen vigencia por aproximadamente 15 minutos, por lo que se esperan ver mensajes ARP relativamente seguido.

El primer paso para cumplir con los objetivos mencionados será implementar una función que, dada una dirección IP, pregunte con qué dirección MAC se corresponde, y la muestre en pantalla si es que obtuvo respuesta. Para esto utilizaremos la herramienta *scapy*, un programa de manipulación de paquetes. Con esta función analizaremos qué ocurre al suministrarle distintos tipos de direcciones IP.

A continuación implementaremos una función que capture los mensajes ARP, escuchando pasivamente la red local durante un determinado tiempo. Con esta información analizaremos la entropía de la red, utilizando como modelo de fuente el que se explica en la sección ***. A partir de estos datos graficaremos lo observado: grafos dirigidos en los que se indica qué IP mandó mensaje a cual *** COMPLETAR. Esperamos que estos gráficos nos den una caracterización de cada una de las redes estudiadas.

2. Métodos

2.1. Primera Consigna: Implementación de un cliente ARP

El primer paso consiste en implementar una función que, dada una dirección IP, envíe un mensaje ARP a través de la red local preguntando con qué dirección MAC se corresponde. Si recibe respuesta, se muestra por pantalla dicha dirección; en caso contrario se indica que la IP es inexistente, o no puede ser alcanzada desde la red local.

Como mencionamos, esta consigna se implementa utilizando la herramienta *scapy*. Ya que la misma funciona sobre python, programamos la función en este lenguaje. Para el envío del paquete ARP instanciamos un objeto del tipo ARP, y modificamos solamente el campo *pdst*, ya que los otros tienen por defecto los valores que necesitamos (la IP de la máquina que envía, tipo de mensaje *who-has*, etc). El envío en sí se realiza a través de la función *sr* (send and receive), utilizando un timeout de 3 segundos. Esta función devuelve una lista cuyo primer elemento son los paquetes que respondieron, y el segundo los paquetes sin respuestas. Si el primer elemento no es vacío, se devuelven las direcciones MAC recibidas[1]; en caso contrario se indica que dicha dirección no puede ser alcanzada en la red local.

Utilizando la función programada, vamos a analizar qué ocurre al suministrarle distintos tipos de direcciones IP. Todos los casos que mencionamos a continuación fueron fueron probados sobre una red wireless, cuya IP era 192.168.0.3. Los casos testeados fueron los siguientes:

1. Direcciones que pertenecen a la red (para conocer las direcciones IP que se encuentran en la red utilizamos la función *arping*, incluída dentro de *scapy*): 192.168.0.1, 192.168.0.4, 192.168.0.6
2. Direcciones con la máscara de red correcta, pero que no pertenecen a hosts conectados a la misma. Por ejemplo: 192.168.0.2, 192.168.0.5, etc.
3. Misma dirección que la máquina de origen (en este caso, 192.168.0.3).
4. Dirección IP correspondiente a la máquina de origen, según se ve de afuera (utilizando por ejemplo http://www.whatismyip.com/). En este caso dicha dirección fue 24.232.212.124.
5. Dirección broadcast de la red (192.168.0.255)
6. Dirección 0.0.0.0
7. Dirección 255.255.255.255
8. Direcciones inválidas (por ejemplo: 123456789)
9. Direcciones que no pertenecen a la red local (por ejemplo, 173.194.42.35)

Al mismo tiempo que probamos estos casos con la función mencionada, utilizaremos también la herramienta *wireshark*, para poder ver con más precisión los mensajes que se envían (como sólo nos interesan los paquetes ARP, utilizaremos un filtro que muestre solamente los mismos).

2.2. Segunda Consigna: Capturando tráfico

Para capturar los mensajes ARP nuevamente utilizaremos la herramienta *scapy*: esta vez haremos uso de la función *sniff*, filtrando solamente los mensajes ARP. Para hacer luego un análisis de los datos obtenidos y poder comparar el comportamiento de distintas redes, tomamos los siguientes datos:

- `Captura Oficina`: Escuchamos durante aproximadamente ocho horas de una jornada laborable la red ethernet interna de una oficina de una empresa de software. Para cada paquete recibido guardamos todos sus campos, agregando el tiempo en el que se produjo la captura.
- `Captura Red ISP`: Escuchamos durante aproximadamente dieciséis horas (13:00 - 5:00) de un día hábil a través de un modem hogareño conectado a directamente al ISP. Relevamos los mismos datos que para el caso anterior.
- `Captura Laboratorio`: Utilizamos además los archivos `pcap` provistos por la cátedra, con información de una red de alto tráfico ARP, y otra de poco tráfico (`big_arp.pcap` y `small_arp.pcap`), correspondiente a capturas realizadas dentro de los laboratorios de la Facultad. En este caso no contamos con información temporal.

Una vez obtenidos los datos, analizamos la entropía de la red. Para ello definimos una fuente de información cuyos símbolos consisten en pares `<ip_fuente, ip_destino>`[2]. Estos símbolos son tomados de los mensajes ARP del tipo *who-has* que envía cada host[3]. Recordemos que los mensajes *is-at* se envían de manera privada al nodo que preguntó por esa IP, por lo tanto sólo podemos escuchar los del tipo *who-has*, o los *is-at* envíados a nuestra máquina. Para evitar considerar que la máquina que corre el sniffer es más sigificativa de lo que realmente es, sólo observaremos los paquetes *who-has*, dado que son los únicos mensajes que podemos ver en su totalidad. Con estas salvedades, consideramos que

[1] Si bien en condiciones normales, la asocicación IP-MAC debería ser única, en esta instancia consideramos que se pueden obtener múltiples respuestas. Para el análisis de los datos obtenidos en las secciones siguientes esto no será relevante pues sólo analizaremos relaciones entre direcciones IP

[2] Consideramos los pares de manera ordenada, por lo cual no es lo mismo una combinación de IPs, que la misma combinación de manera invertida.

[3] El sniffer programado captura ambos tipos de mensaje ARP; al calcular la entropía filtramos primero los mensajes *who-has*.

contamos con suficiente información, utilizando únicamente el protocolo ARP, para conocer y caracterizar la red en cuestión.

En estas condiciones, cada símbolo posible tiene una probabilidad de ocurrencia que se relaciona con la importancia del nodo dentro de la red: los nodos *significativos* (como, por ejemplo, el router) serán aquellos que aparezcan con mayor frecuencia en el campo `ip_destino`. La probabilidad de cada símbolo se calculará de manera empírica, en base a la cantidad de apariciones sobre el total de símbolos presentados (es decir, su frecuencia relativa)). Además, podemos considerar que la fuente, definida de esta manera, es una fuente de memoria nula, ya que suponemos que los símbolos emitidos son estadśticamente independientes[4].

Una vez conseguidos los datos de captura, procesamos estos archivos con sendos scripts en *Python*. Para todos los casos trazamos un grafo de red en base a los mensajes escuchados y calculamos la entropía total de la red. Esto lo realizamos trivialmente desde la definición, con la salvedad dicha anteriormente, de aproximar la probabilidad de ocurrencia de los símbolos por su frecuencia relativa. Para ajustar esta aproximación, consideramos períodos de tiempo prolongados. En los casos en que contábamos con información temporal sobre la captura de los paquetes, realizamos, además, el cálculo de la entropía para subintervalos disjuntos de una hora, para observar la variabilidad de la red a lo largo del tiempo de captura. En todos los casos, calculamos la entropía en bits, es decir, tomando logaritmo en base 2.

2.3. Tercera Consigna: Gráficos y Análisis

Para caracterizar una red buscamos, por un lado, encontra nodos relevantes de la misma. Contamos para esto con los símbolos mencionados en la sección anterior (pares `<ip_fuente, ip_destino>`), y con el valor de la información de cada nodo (en este caso, función de su frecuencia relativa), así como la entropía de la red[5]. A partir de estos datos podemos caracterizar cada una de las redes a estudiar observando la siguiente información:

1. Grafos dirigidos, en donde cada nodo representa una IP, y existe un eje desde el nodo A hasta el nodo B, si A hizo un pedido *who-has* a B (es decir, si el símbolo `<A, B>` apareció durante la captura). Dado que el trazado de aristas resulta imposible de interpretar a simple vista, lo simplificamos gráficamente, representando el grafo de manera circular y concéntrica, donde los nodos que se encuentran más al centro tienen mayor grado de entrada que los más distantes. No realizamos esto en un continuo, sino separándolo en clases en una escala logarítmica.

2. Histograma relacionando cada IP, con la cantidad de veces que fue consultada (es decir, que apareció en el campo `ip_destino`).

3. Cantidad de información de cada IP. En este caso puede compararse la información con la entropía total, y sacar conclusiones a partir de esto de cuáles son los nodos significativos.

4. Entropía de la fuente en función de un rango horario. En este caso no estamos observando tanto los nodos significativos, sino cómo distintos tipos de redes pueden variar la cantidad de información presente según el momento del día, y el tipo de red que sea -hogareña, oficina, etc. Es de esperar que en un momento de mucha actividad de la red la variabilidad de los mensajes sea mayor, ya que las posibilidades de comunicación entre equipos aumenta considerablemente sólo por el hecho de que haya mayor cantidad de hosts conectados. Por otro lado, contrastaremos esta variabilidad con la entropía total calculada sobre toda la captura: si los valores correspondientes a subintervalos se corresponden con la entropía total, podremos deducir que la entropía de la red es relativamente constante, más allá del momento calculado, mientras que las discrepancias nos indicarán los períodos de mayor abundancia o escasez de información en la red.

5. Histograma de cantidad de apariciones de mensajes. Contamos cuántas veces aparece cada mensaje, y graficamos en un histograma estas cantidades. Esto nos permite diferenciar la variedad de mensajes escuchados: la cantidad de observaciones de un mensaje es creciente de izquierda a derecha, con lo cual, un pico hacia la izquierda indicaría variedad de mensajes escuchados frecuentemente (pocas apariciones pero de muchos mensajes distintos), mientras que un equilibrio en el histograma indicaría que existe la misma abundancia de mensajes poco frecuentes que de usuales. Nos extenderemos al respecto en el análisis de los gráficos en concreto, relacionándolo con los valores de entropía medidos en subunidades de tiempo.

[4] Esta suposición descarta casos patológicos, en los que la respuesta (o su ausencia) de un pedido pudiera generar nuevos, por ejemplo, o tráfico intencionado más allá del normal funcionamiento del protocolo.

[5] Para caracterizar la red, consideramos que la asociación entre una IP y una dirección MAC no es relevante. Lo importante en este caso es la comunicación entre los distintos hosts, los cuales pueden representarse a través de su IP o de su dirección MAC. Elegimos utilizar la IP, pero podría haber sido cualquiera de las dos opciones.

3. Resultados

3.1. Implementación de un cliente ARP

De todos los casos mencionados en la sección 2.1, el único en que se recibió una respuesta con la dirección MAC solicitada es en el caso (1), es decir, al pedir direcciones MAC de nodos presentes dentro de la red local. En todos los otros casos el mensaje ARP no recibió respuesta alguna.

A través del uso de *Wireshark* pudimos observar con un poco más de detalle qué ocurrió en cada una de estas situaciones.

El caso (1) funcionó como se esperaba: al paquete *who-has* (enviado de manera broadcast) le sigue un paquete *is-at*, con el host cuya MAC se quiere conocer como origen, y dirigido únicamente a la máquina que preguntó.

Si se pregunta por una IP que no existe, pero que tiene una máscara correspondiente con la red local (por ejemplo, 192.168.0.2), no se observa ningún tipo de respuesta. Sin embargo, si se pregunta por un dirección cuya máscara no se corresponde con la red (casos 4, 6, 7, 8[6] y 9) se observa un intercambio de mensajes ligeramente diferente. En vez de enviar un paquete ARP de manera broadcast, la máquina pregunta primero quién tiene la dirección (en nuestro caso) 192.168.0.1 (dirección del router). Al recibir la respuesta *is-at* con la MAC correspondiente, pregunta específicamente al router quién tiene la dirección buscada. Es decir, pareciera ser que antes de enviar mensajes ARP, automáticamente se calcula si dicha IP se encuentra dentro de la red local. Si no se encuentra, se ejecuta el intercambio mencionado, enviando el pedido por la IP buscada directamente al router (del cual no se recibe respuesta, al menos en los casos probados).

Por último, si se

3.2. Captura de tráfico

3.3. Gráficos y Análisis

4. Conclusiones

[6] En el caso de una direccón inválida, la dirección enviada se traduce a una dirección IP correcta (en este caso se envió el paquete preguntándo por la dirección 7.91.205.21).