DEPARTAMENTO
DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

TP1: Wiretapping

20 de abril de 2016 — Teoría de las Comunicaciones

Integrante	LU	Correo electrónico
Benitti, Raul	592/08	raulbenitti@gmail.com
Castro, Damian	326/11	ltdicai@gmail.com
Lizana, Helen	118/08	hsle.22@gmail.com
Grenier, Michelle	418/10	michelle.grenier@hotmail.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. Introducción

Las redes de computadoras han dejado de ser una tecnología reservada a ciertos ambitos científicos y militares para convertirse en piezas fundamentales en el desarrollo de casi cualquier actividad, a tal grado que las relaciones humanas, desde el comercio hasta las guerras, han sido profundamente transformadas por la conectividad alcanzada en los últimos años. Es por esto que analizar los distintos aspectos de una red puede proveer información útil para comprender el uso que se esta dando a la red, informacion que sirve tanto para modificar la infrastructura y los protocolos utilizados a fin de mejorar la calidad del servicio como, incluso, manipular las actividades que se esten llevando sobre ella.

En el presente trabajo experimentaremos sobre sistemas basados en dos de las tecnologías de redes más difundidas, Ethernet y WiFi 802.11, y analizaremos los datos obtenidos utilizando dos modelos de fuente de información para extraer conclusiones sobre el uso y la configuración de las redes. Los conceptos teóricos sobre los que basaremos el analisis se presentan a continuación.

1.1. Información y Fuente de información

Una fuente de información es todo aquello que emite mensajes de acuerdo a una ley de probabilidad fija. Los mensajes pertenecen a un conjunto finito de símbolos $S = s_1, ..., s_n$, conocido como el alfabeto de la fuente. La emisión de un símbolo s_i por parte de la fuente S representa un evento que tiene asociada una probabilidad fija $P_S(s_i)$ de ocurrir.

Dado un evento e con probabilidad $P(e)$, se define la **información del evento** e como

$$I(e) = -\log P(e)$$

$I(e)$ es una medida de la cantidad de información que obtenemos por la ocurrencia de E: mientras más improbable sea E, mayor será la información brindada por su ocurrencia (menor sera la incertidumbre sobre el hecho observado). Dicho de otra manera, si sabemos que un evento E tiene alta probabilidad de ocurrir, entonces su ocurrencia no aportará mucha información sobre lo que se esta observando.

1.2. Entropía

Dada una fuente de información $S = s_1, ..., s_n$, se define la entropía de S, $H(S)$, como la suma ponderada de la información de casa simbolo de S

$$H(S) = \sum_{i=1}^{n} P(s_i) * I(s_i)$$

La entropía de una fuente de información mide la cantidad de información esperada al observar la emisión de un nuevo símbolo por parte de la fuente. Dado un evento e

1.3. ARP

Para poder realizar envío de paquetes de capa 3 utilizando los servicios de capa 2 es necesario poder realizar un mapeo entre las direcciones de ambas capas. ARP (*Address Resolution Protocol*) es un protocolo de control que surge como respuesta a esta necesidad. Cada host y switch de una red mantiene una

tabla ARP donde se relaciona una dirección lógica d con la direccion física f a la que debe entregarse cualquier paquete destinado a d (el host con direccion física f no es necesariamente el destinatario de la dirección d: puede ser un intermediario que sabe como hacer llegar el paquete a d). En el caso de redes IP sobre Ethernet, ARP es utilizado para mapear direcciones IP con direcciones MAC. La configuracion de estas tablas ARP se realiza dinámicamente siguiendo un protocolo que consiste básicamente en los siguientes pasos:

1. Un host **A** desea enviar un paquete a una determinada IP. Si **A** conoce la direccion MAC a la que debe enviar los paquetes destinados a esa IP, entonces utiliza esa direccion física. Si no, envía un mensaje broadcast dentro de la red y aguarda la respuesta. Este mensaje se conoce como **ARP request** (WHO_HAS), y lleva la siguiente información:

   IP origen: IP de A

   IP destino: IP a la que se desea enviar un paquete

   MAC origen: MAC de A

   MAC destino: dirección broadcast de Ethernet (FF:FF:FF:FF)

2. Si dentro de la red existe un host B que sabe como direccionar a la direccion IP requerida, entonces responde al mensaje ARP request con un mensaje **ARP reply** (IS_AT) indicando su dirección física. Este host puede ser el dueño de la dirección IP, o un host intermediario (como un router). Además, extrae las direcciones IP origen y MAC origen del paquete ARP request, yactualiza su tabla ARP para relacionarlas. El paquete ARP reply contiene la siguiente información:

   IP origen: IP de B

   IP destino: IP de A

   MAC origen: MAC de B

   MAC destino: MAC de A

3. **A** recibe el ARP reply de **B**, actualiza su tabla ARP y envía el paquete original utilizando la dirección física de **B**.

Además, cada entrada de las tablas ARP tiene seteado un tiempo de vida. Una vez agotado ese tiempo, la entrada se descarta y debe volver a aprenderse.

Observación	Eventos contabilizados
Paquete WHO_HAS, MAC origen	1 evento
Paquete WHO_HAS, MAC destino	0 evento
Paquete IS_AT, MAC origen	1 evento
Paquete IS_AT, MAC destino	1 evento

Cuadro 1: Contabilización de eventos para $S1$

2. Experimentos

Como mencionamos anteriormente, el análisis de paquetes de una red puede utilizarse para inferir información sobre la actividad y topología de la red. En este trabajo aprovecharemos esta capacidad para dilucidarqué protocolos se distinguen del resto, cuál es la incidencia de los paquetes ARP y cuáles son los nodos destacados de las redes. Realizamos cuatro experimentos para obtener datos, uno sobre cada una de las siguientes redes:

- Red1: Red wiFi de un laboratorio del DC
- Red2: Red Wifi de un bar Starbucks
- Red3: Red Ethernet en un ámbito laboral
- Red4: Red Ethernet en un ámbito laboral

Modelamos estas redes como dos fuentes de información distintas:

1. S: este modelo fue dado por la cátedra. El alfabeto se define como los protocolos enviados dentro de los paquetes Ethernet capturados durante el experimento. Así mismo, consideramos como función de probabilidad a la frecuencia de cada símbolo dentro del experimento, donde marcamos como ocurrencia de un evento a la observación de un protocolo al capturar un paquete.
2. $S1$: con este modelo deseamos poder distinguir los nodos relevantes de una red dada. Para ello, definimos el alfabeto de $S1$ como las direcciones MAC de los paquetes del protocolo ARP. En este caso también tomamos como función de probabilidad a la frecuencia de cada dirección MAC dentro del total observado, pero contabilizando las ocurrencias de cada MAC segun se muestra en el Cuadro **??**. La decisión de utilizar direcciones MAC en lugar de direcciones IP radica en el hecho de querer identificar los nodos físicos dentro de la red observada. Utilizar directamente direcciones IP podría llevar a malinterpretar la topología de la red: por ejemplo, podríamos considerar como relevantes a varios host con distintas IP que se encuentran fuera del sistema en estudio, y no notar que todo el táfico debe pasar por un único nodo propio. Por otra parte, decidimos utilizar las direcciones MAC tal como se muestra en el Cuadro **??** pués consideramos que brindan la mayor información acerca de la actividad de cada nodo en la red: enviar un paquete ARP nos da información sobre la existencia del host, y de recibir un paquete ARP (es decir, la existencia de un paquete IS_AT destinado a un nodo particular) podemos deducir que el nodo destino seguramente continue con envío de más paquetes (es decir, tenga actividad inmediata).

2.1. Herramientas de sniffing

Para capturar y procesar la información, utilizamos tanto el programa *Wireshark* como dos script (capturar.py, identificar.py), escritos en Python, utilizando la librería para análisis de redes *scapy*. Ambas herramientas hacen uso del modo promíscuo de la placa de red, en el cual se capturan no solo los paquetes dirigidos a el host que esta capturando, sino todos los paquetes que se envíen por el medio.

2.1.1. Implementación de S: capturar.py

En su forma de ejecución básica, el script muestra por pantalla cada paquete que captura hasta que sea detenido con una interrupción (CTRl+C). Al finalizar, se muestra

1. el total de paquetes capturados
2. los protocolos observados (junto con la cantidad de veces que se observo cada uno)
3. la entropía correspondiente modelo S.

Si bien incorporamos varias opciones de ejecución (ejecutar el comando con la opción *-h.*), la forma más sencilla corresponde a

```
sudo python capturar.py -i <interfaz_de_captura>
```

2.1.2. Implementación de $S1$: identificar.py

Este script es similar a capturar.py, pero en lugar de analizar los protocolos de cada paquete, filtra solo los paquetes ARP e implementa el modelo de fuente $S1$. Al igual que capturar.py, el script muestra por pantalla cada paquete que captura hasta que sea detenido con una interrupción (CTRl+C). Al finalizar, devuelve

1. un diccionario donde se mapea direcciones MAC con direcciones IP
2. las direcciones MAC observadas (junto con la cantidad de veces que se observo cada una)
3. la entropía correspondiente modelo $S1$.

Para ver opciones de ejecución, ejecutar el comando con *-h.*). la forma más sencilla corresponde a

```
sudo python identificar.py -i <interfaz_de_captura>
```

2.2. Análisis de entropía de una red

Uno de los ejercicios solicitaba calcular la entropía de una fuente. Para ello debemos definir con presicion dos cosas, la **fuente de informacion** y el **evento**, para luego calcular su probabilidad y de allí la entropía.