UNIVERSIDAD SAN FRANCISCO DE QUITO

REDES I

TAREA 2

Protocolo ARP: Fabricantes de Infraestructura de red en la USFQ

Ricardo Muriel 125858

Alejandro Maruri 136515

Objetivos

- Comprender el funcionamiento y la manera de actuar del protocolo ARP a través de un script para conocer características y direcciones físicas de los dispositivos dentro de una red local.
- Reconocer la importancia de una MAC-Address, distinguiendo su exclusividad debido a su estructura para determinar los fabricantes de la infraestructura TI de la USFQ.
- Comprender y analizar la diferencias entre la obtención de resultados mediante la aplicación del protocolo ARP en un red WLAN y una LAN.

Introducción

El protocolo ARP (Address Resolution Protocol) es un protocolo resuelto en la capa de enlace del modelo OSI (capa física del modelo TCP/IP) que trata de responder a "¿Cuál es la dirección física de un nodo de la red dada su dirección IP?". En el trasfondo de este protocolo se encuentra el descubrimiento de los dispositivos conectados al mismo medio físico. De manera básica, el protocolo lo que hace es enviar un datagrama, a manera de broadcast (a todos los nodos de la red), con la dirección IP del nodo que se quiere conocer su dirección física. Si existe un nodo que tenga dicha dirección IP dentro de la red, este responderá a la petición realizada por el nodo que realizó la solicitud. El nodo solicitante almacena IP-MAC hasta que esta caduque (un tiempo de vida). Por otro lado, una dirección física o MAC-Address tiene una estructura especial, lo que garantiza que sea única. La estructura está compuesta por seis pares de caracteres alfanuméricos, donde los primeros tres indican un código para identificar al fabricante, y los restantes al dispositivo.

Procedimiento

Con la intención de recolectar información importante, el procedimiento se efectuó 6 veces: 3 de ellas en LAN y 3 en WLAN. Para cada una se utilizó un timeout distinto, en este caso fue de 15, 30 y 60 segundos. El proceso se realizó en la biblioteca principal de la USFQ el dia Jueves 5 de Marzo entre las 5 y 6 pm. Esto se llevó a cabo para determinar los fabricantes que conforman la infraestructura de red de la USFQ y comprender el funcionamiento del protocolo ARP en distintas redes y circunstancias.

1. **Archivos de Salida:** Se modificó el script "dr_arp_discovery.py" para generar archivos de texto de salida que contengan la IP y MAC-Address de los dispositivos conectados a la red local donde se ejecuta el programa. Además, se utilizó el método *summary* de uno de los objetos en *ans* (*"r"*) del script, el cual retorna un *string* con la respuesta a la petición del protocolo ARP. La 5ta posición de dicho *string* corresponde a la MAC-Address y la 7ma a la IP del nodo que responde a la solicitud.

```
mac_summary = str(r.summary()).split()

# Writing in file
output_file.write(mac_summary[5].replace("'", "") + '\t' + mac_summary[7].replace("'", "") + '\n')
```

Imagen 1. Obtención del MAC Address utilizando el método summary(), para luego escribirlo en un archivo de texto con el formato solicitado

2. **Identificar fabricantes:** Para identificar los fabricantes de las tarjetas de red, se utilizó una API en la dirección "http://macvendors.co/api/" dentro de una función que recibe una dirección MAC y retorna el nombre de la compañía a la que pertenece.

```
def get_mac_vendor(mac_address):
    """Returns the vendor of NIC given a MAC-Address (if exists)"""
    url = "http://macvendors.co/api/"  # API base url,you can also use https if you need
    request = urllib2.Request(url + mac_address, headers={'User-Agent': "API Browser"})
    response = urllib2.urlopen(request)
    # Fix: json object must be str, not 'bytes'
    reader = codecs.getreader("utf-8")
    obj = json.load(reader(response))

    return obj['result']['company']  # Return company name
```

Imagen 2. Obtención de los fabricantes mediante el uso de la API http://macvendors.co/api/ para cada MAC Address.

3. **Archivos para graficar:** Con la información obtenida a través de la API, se generó archivos de texto con los nombres de los fabricantes y su porcentaje, de acuerdo a su frecuencia en la red. Cada archivo corresponde a cada caso de red y timeout distinto.

```
for vendor in mac_vendor_dict:
    frec = mac_vendor_dict[vendor]
    vendor_percent[vendor] = (frec / total * 100)
    mac_vendor_dict[vendor] = (frec / total * 100)  # Percent of a vendor
    output_file_vendor_percent.write(str(counter) + '\t' + vendor + '\t' + str(mac_vendor_dict[vendor]) + '\n')
    counter = counter + 1
```

Imagen 3. Obtención del porcentaje de frecuencia para cada fabricante

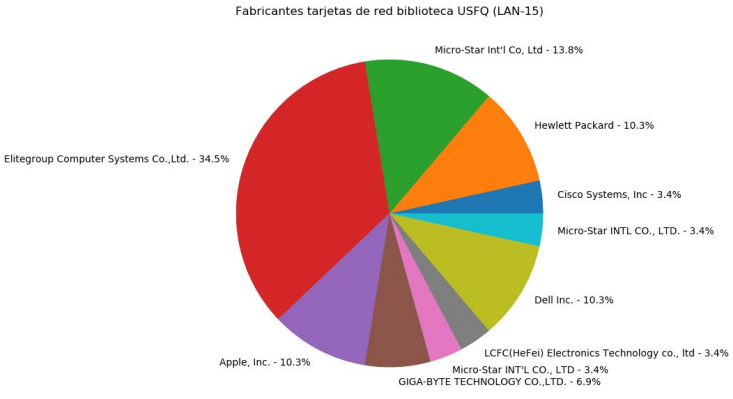
4. **Graficar resultados:** Se creó un script en python que interpreta los archivos de texto generados en el paso anterior para graficar los resultados en forma de pastel

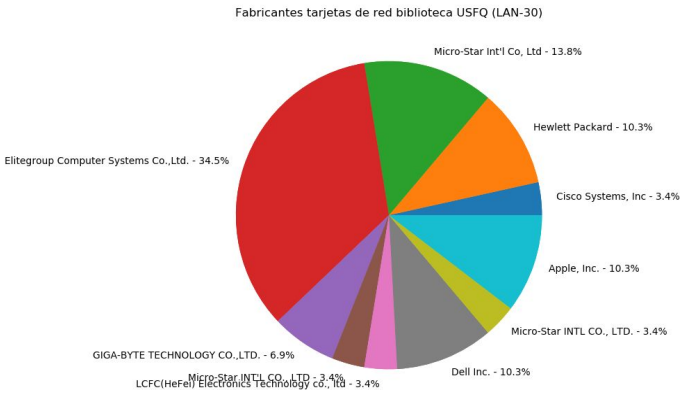
```
wedges, texts, autotexts = ax.pie(percents, autopct=lambda pct: func(pct, percents), textprops=dict(color="w"))
ax.pie(percents, labels=vendors)
```

Imagen 4. Plot de los resultados de cada fabricante para los distintas redes.

Resultados

LAN-15 (s) y LAN-30 (s):

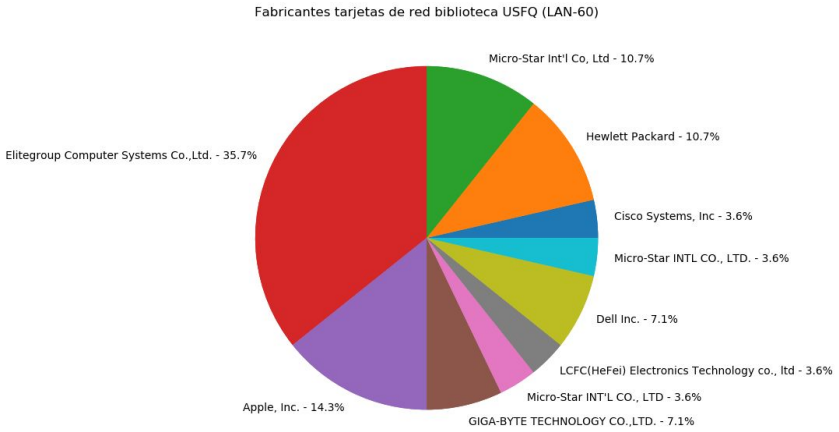




Fabricante con mayor frecuencia: Elitegroup Computer Systems con 34.5 %.

Fabricante con menor frecuencia: LCFC HeFei Electronics Technology y Cisco Systems, ambos con 3.4%.

LAN-60 (s):

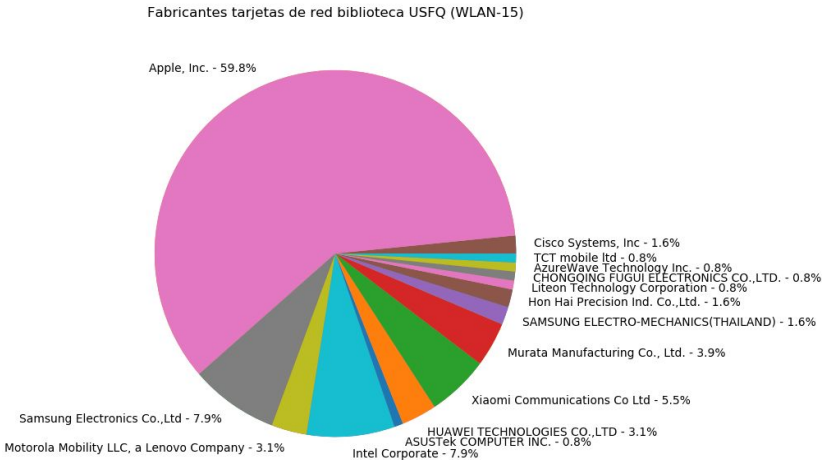


Fabricante con mayor frecuencia: Elitegroup Computer Systems con 35.7 %

Fabricante con menor frecuencia: LCFC HeFei Electronics Technology y Cisco Systems ambos con 3.6%.

Se puede observar que, para todos los casos, el fabricante con mayor frecuencia es Elitegroup Computer Systems; y los de menor son LCFC HeFei Electronics Technology y Cisco Systems. Estos resultados son consistentes en las 3 pruebas, debido a que al ser un medio físico cableado, es posible obtener con mayor precisión la información de cada dispositivo en la red, ya que las conexiones y desconexiones fortuitas ocurren raramente, existe escasa interferencia y es un medio rápido de transmisión. La única excepción ocurrió en la tercera prueba (60 s. timeout), en la que se encontró un dispositivo menos (28) que en los casos anteriores.

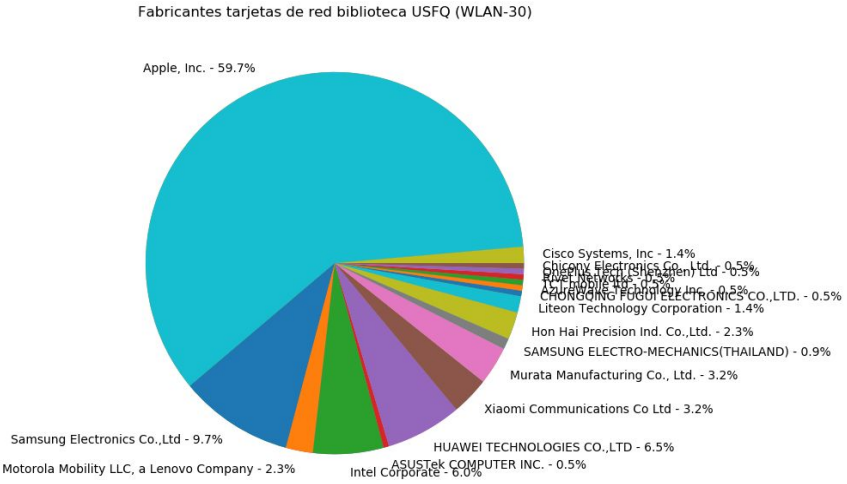
WLAN-15 (s):



Fabricante con mayor frecuencia: Apple con 59.8 %.

Fabricante con menor frecuencia: TCT mobile, AzureWave Technology, ChongQing Fugui Electronics, y AsusTek Computer, todos con 0.8%.

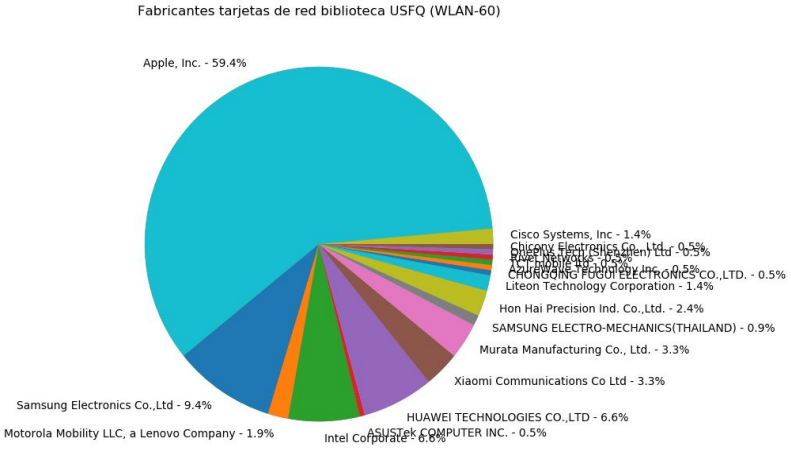
WLAN-30 (s)



Fabricante con mayor frecuencia: Apple con 59.7 %.

Fabricante con menor frecuencia: Chicony Electronics, OnePlus Tech, Rivet Networks, TCT mobile, AzureWave Technology, ChongQing Fugui Electronics, y AsusTek Computer, todos con 0.5%.

WLAN-60 (s):



Fabricante con mayor frecuencia: Apple con 59.4 %.

Fabricante con menor frecuencia: Chicony Electronics, OnePlus Tech, Rivet Networks, TCT mobile, AzureWave Technology, ChongQing Fugui Electronics, y AsusTek Computer, todos con 0.5%.

Se observa que los resultados obtenidos en la red WLAN son menos consistentes, ya que fabricantes como Chicony Electronics, Oneplus Tech , y Rivet Networks, que aparecen en los timeouts de 30 y 60, no aparecen en el de 15. Esto es debido a que Wi-Fi es un medio de transmisión compartido (aire) en el que pueden ocurrir más interferencias o conexiones y desconexiones fortuitas. Por lo tanto, es infrecuente reportar los mismos resultados una y otra vez.

Conclusiones

- Se observó ciertamente el funcionamiento del protocolo ARP, que se encarga de averiguar la dirección física asociada a una dirección IP dentro de la red local actual (mismo medio físico).
- Los resultados obtenidos durante el procedimiento con la red LAN tienen mayor consistencia que los resultados de la red WLAN. Esto se debe a que la conexión LAN es realizada mediante un medio físico cableado y por lo tanto existe una mayor precisión al momento de consultar las direcciones MAC, ya que los dispositivos no varían mucho. Por otro lado, la conexión WLAN presenta menor estabilidad por ser un medio de transmisión compartido lo cual genera ciertas variantes en los resultados.
- Al momento de realizar el proceso en la red LAN, se observó que la primera vez existió un tiempo de ejecución de 10 minutos, pero en las pruebas restantes, el tiempo eran menor a 10 segundos. Este resultado se presenta porque primero existe una conexión inicial donde se obtiene un conocimiento del estado de la red y los dispositivos conectados a esta, por lo que, para las siguientes iteraciones, ya conoce su ambiente; lo que resulta en un menor tiempo de ejecución.