Teoria de las comunicaciones

Segundo Cuatrimestre de 2012

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

Taller de Capa de Enlace

Taller N°1

Integrante	LU	Correo electrónico
Mancuso, Emiliano	597/07	emiliano.mancuso@gmail.com
Mataloni, Alejandro	706/07	amataloni@gmail.com
Curtua, Matias	453/07	curtu$_i$*nfinito*73@*hotmail.com*

Reservado para la catedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Índice

1. Primera consigna

Lo que hicimos fue implementar en *Scapy* un script que, dada una dirección IP, realiza un pedido por la dirección física y recibe y muestra la respuesta en caso de ser recibida.

El script es el siguiente:

```
pkt = ARP(pdst=sys.argv[1], op="who-has");
response = sr1(pkt)
response.show()
```

Lo interesante es ver que ocurre en algunos casos:

- dirección inexistente = el script se cuelga esperando la respuesta que nunca llega.
- dirección de la maquina de origen = el paquete ARP no se envía.
- dirección broadcast = pregunta por la dirección por lo que pasa lo mismo que si fuera una dirección inexistente
- dirección de red = pregunta por la dirección por lo que pasa lo mismo que si fuera una dirección inexistente

2. Segunda consigna

Implementamos un script en *Scapy* para escuchar pasivamente en la red y capturar cada mensaje ARP capturado. En el mismo script cuando capturamos el mensaje leemos los datos para obtener los vendors y exibirlos por pantalla.

El script es el siguiente:

```
# Build the vendor dictionary
ins = open( "vendorsUtil.txt", "r" )
vendors_dict = {}
for line in ins:
    vendors_dict[line[0:8]] = line[9:-1]
ins.close()

# Print pretty vendor
def arp_monitor_callback(pkt):
vendor_prefix = pkt[ARP].hwsrc[0:8].upper()
strr =  pkt[ARP].psrc + ": \t" + vendors_dict[vendor_prefix]
print strr

if len(sys.argv) == 1:
print "Listening for 5 seconds.."
to = 5  # Default value
else:
to = int(sys.argv[1])

sniff(prn=arp_monitor_callback, filter="arp", store=0, timeout=to)
```

3. Tercera consigna

Para esta parte lo que hicimos fue capturar los paquetes *ARP*, y crear un grafo dirigido de IPs. Cada nodo representa una dirección IP y existe un eje entre los nodos x e y si y solo si se observo un request ARP con source el IP del nodo x y target igual al IP que representa el nodo y. Consideramos que esta es la mejor forma para extraer informacion en cuanto a la topologia de la red en cuestión, asi como también sacar datos interesantes de la misma.

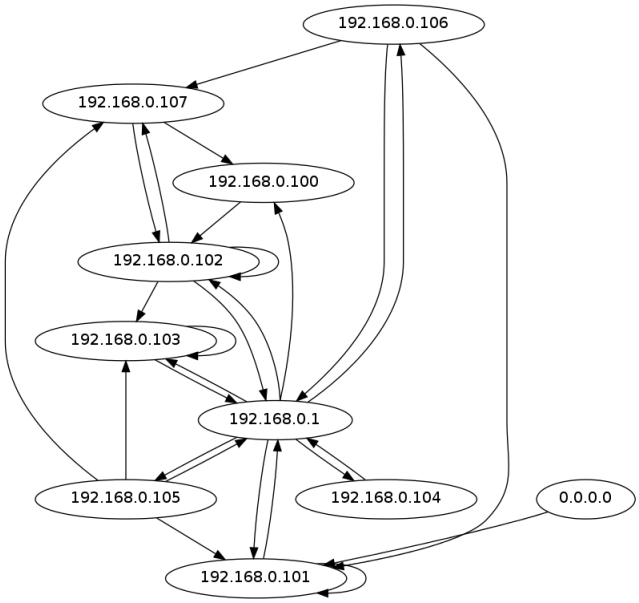

Figura 1: Gráfico dirigido de los distintos request ARP observados.