Teoría de las Comunicaciones

Primer Cuatrimestre de 2013

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

Taller de Wiretapping

Grupo:

Integrante	LU	Correo electrónico
Matías Capello	006/02	matiascapello@gmail.com
Santiago Hernández	48/11	santi-hernandez@hotmail.com
Andrés Laurito	27/11	andy.laurito@hotmail.com

Índice

Índice de figuras

Abstract

Meter el resumen al final...

Palabras Clave:

1. Introducción

El objetivo de este trabajo práctico es abordar algunas nociones de nivel de enlace, poniendo fundamentalmente el foco en el vínculo con la capa superior y desarrollando un acercamiento analítico. El objetivo será analizar de manera interactiva el protocolo **ARP** y sacar algunas conclusiones de cómo se comportan los hosts en un segumento de red determinado.

1.1. ARP - Address Resolution Protocol

El protocolo **ARP** tiene un papel clave entre los protocolos de capa de Internet relacionados con el protocolo TCP/IP, ya que permite que se conozca la dirección física de una tarjeta de interfaz de red correspondiente a una dirección IP. Por eso se llama Protocolo de Resolución de Dirección (en inglés **ARP** significa Address Resolution Protocol).

Cada equipo conectado a la red tiene un número de identificación de 48 bits. Éste es un número único establecido en la fábrica en el momento de fabricación de la tarjeta. Sin embargo, la comunicación en Internet no utiliza directamente este número (ya que las direcciones de los equipos deberían cambiarse cada vez que se cambia la tarjeta de interfaz de red), sino que utiliza una dirección lógica asignada por un organismo: la dirección IP.

Para que las direcciones físicas se puedan conectar con las direcciones lógicas, el protocolo **ARP** interroga a los equipos de la red para averiguar sus direcciones físicas y luego crea una tabla de búsqueda entre las direcciones lógicas y físicas en una memoria caché.

Cuando un equipo debe comunicarse con otro, consulta la tabla de búsqueda. Si la dirección requerida no se encuentra en la tabla, el protocolo **ARP** envía una solicitud a la red. Todos los equipos en la red comparan esta dirección lógica con la suya. Si alguno de ellos se identifica con esta dirección, el equipo responderá al **ARP**, que almacenará el par de direcciones en la tabla de búsqueda, y, a continuación, podrá establecerse la comunicación.

La Figura 1 muestra el formato del paquete **ARP** para el mapero de dirección IP-a-Ethernet. De hecho, **ARP** puede ser utilizado para muchos otros tipos de mapeos las mayores diferencias se dan en los tamaños de dirección. Además de los OP y las direcciones de capa de enlace de ambos origen y destino, el paquete contiene:

- Un campo HardwareType, que especifica el tipo de red física (por ejemplo, Ethernet).
- Un campo ProtocolType, que especifica el protocolo de la capa superior (por ejemplo, IP).
- Campos HLen (hardware address length) y PLen (protocol address length), que especifican las longitudes de la dirección de capa de enlace y la dirección de la capa superior, respectivamente.
- Un campo Operation, que especifica si se trata de una solicitud o una respuesta.

- Las direcciones de origen y destino del hardware (Ethernet) y protocolo (IP).

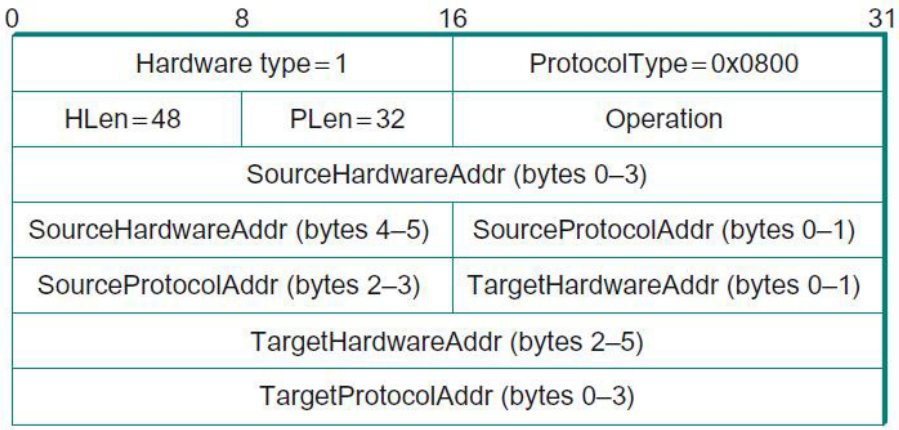

Figura 1: Figura 1 - Formato del paquete ARP para el mapeo de direcciones IP a direcciones Ethernet

1.2. Entropía - Información

En el ámbito de la teoría de la información la entropía, también llamada entropía de la información y entropía de Shannon (en honor a Claude E. Shannon), mide la incertidumbre de una fuente de información.

La entropía también se puede considerar como la cantidad de información promedio que contienen los símbolos usados. Los símbolos con menor probabilidad son los que aportan mayor información; por ejemplo, si se considera como sistema de símbolos a las palabras en un texto, palabras frecuentes como `que`, `el`, `a` aportan poca información, mientras que palabras menos frecuentes como `corren`, `niño`, `perro` aportan más información. Si de un texto dado borramos un `que`, seguramente no afectará a la comprensión y se sobreentenderá, no siendo así si borramos la palabra `niño` del mismo texto original. Cuando todos los símbolos son igualmente probables (distribución de probabilidad plana), todos aportan información relevante y la entropía es máxima.

El concepto de entropía es usado en termodinámica, mecánica estadística y teoría de la información. En todos los casos la entropía se concibe como una *medida del desorden* o la *peculiaridad de ciertas combinaciones*. La entropía puede ser considerada como una medida de la incertidumbre y de la información necesarias para, en cualquier proceso, poder acotar, reducir o eliminar la incertidumbre. Resulta que el concepto de información y el de entropía están ampliamente relacionados entre sí, aunque se necesitaron años de desarrollo de la mecánica estadística y de la teoría de la información antes de que esto fuera percibido.

Shannon ofrece una definición de entropía que satisface las siguientes afirmaciones:

- La medida de información debe ser proporcional (continua). Es decir, el cambio pequeño en una de las probabilidades de aparición de uno de los elementos de la señal debe cambiar poco la entropía.
- Si todos los elementos de la señal son equiprobables a la hora de aparecer, entonces la entropía será máxima.

1.2.1. Definición Formal

Supongamos que un fenómeno (variable aleatoria) tiene un grado de indeterminación inicial igual a k (k estados posibles) y supongamos todos los estados equiprobables, entonces la probabilidad p de que se dé una de esas combinaciones será 1/k. Podemos representar entonces la expresión c_i como:

$$c_i = log_2(k) = log_2\left[1 \div (1 \div k)\right] = -log_2(p)$$

Si ahora cada uno de los k estados tiene una probabilidad p_i, entonces la entropía vendrá dada por la suma ponderada de la cantidad de información:

$$H = -p_1 log_2(p_1) - p_2 log_2(p_2) - ... - p_k log_2(p_k) = -\sum_{i=1}^{k} p_i log_2(p_i)$$

Por lo tanto, la entropía de un mensaje X, denotado por H(X), es el valor medio ponderado de la cantidad de información de los diversos estados del mensaje:

$$H(X) = -\sum_i p(x_i) log_2 p(x_i)$$

2. Desarrollo

2.1. Primera consigna: Implementación de un cliente ARP

Se utilizó *Scapy* para implementar un script que reciba una dirección IP, realice un pedido de su MAC Address y lo devuelva.

La función implementada para resolver esto es `preguntarMAC`, que recibe una dirección IP por parámetro Dentro de dicha función se ejecuta lo siguiente:

```
mensajeARP = srp1(Ether(dst="ff:ff:ff:ff:ff:ff")/ARP(pdst=ip),timeout=2)
```

Mediante srp1, se envía el paquete ARP en broadcast y se acepta solo la primer respuesta (suponemos que solo uno contesta ya que solo uno debería tener la ip). `Ether(dst="ff:ff:ff:ff:ff:ff")` es para mandarlo en modo broadcast y `ARP(pdst=ip)`, la IP de la que le pregunto su MAC Address.

Se probó esta funcionalidad con diferentes direcciones de IP, para analizar el resultado obtenido:

- Dirección ip existente: Se obtiene la respuesta con la dirección MAC.
- Dirección ip inexistente: No hay respuesta, se cae por timeout.
- Dirección ip de la máquina origen: No hay respuesta, se cae por timeout.
- Dirección ip Broadcast: No hay respuesta, se cae por timeout.

También se probaron los siguientes casos con scapy obteniendo estos resultados:

- Dirección ip existente con su MAC address: Se obtiene la respuesta con la dirección MAC.
- Dirección ip existente con la MAC address ”00:00:00:00:00:00”: No hay respuesta, se cae por timeout.

- Dirección ip existente con una MAC address distinta a la suya y a la de broadcast: No hay respuesta, se cae por timeout.
- Dirección ip de nuestro router con nuestra MAC address y una IP distinta a la nuestra como source: Se obtiene la respuesta con la dirección MAC normalmente. No notamos cambios.
- Dirección ip equivocada a una MAC address conocida: No hay respuesta, se cae por timeout.

2.2. Segunda consigna: Capturando tráfico

Para la primera parte se nos pedía realizar un programa que escuche en la red por un período de tiempo y capture los mensajes ARP que circulen por la misma.

Para ello utilizamos la función de Scapy `sniff` de la siguiente manera:

```
sniff(prn=arp_monitor_callback, filter=''arp'', store=0)
```

El parámetro `store` se setea en 0 para que la función `sniff()` no guarde nada (como lo haría de otra manera) y por lo tanto corra de forma ilimitada. El parámetro `filter` se usa para mejorar la performance cuando hay alta carga de datos: El filtro se aplica dentro del kernel, por lo que Scapy sólo verá trafico ARP.

La función callback `arp_monitor_callback` se aplica a los paquetes que se sniffean, y se define asi:

```
def arp_monitor_callback(pkt):
  global salida
  if ARP in pkt and pkt[ARP].op in (1,2):
    salida.write(pkt.sprintf(''%ARP.hwsrc%%ARP.psrc%%ARP.hwdst%%ARP
    .pdst% '')+str(datetime.now())+'\ n')
    return pkt.sprintf(''%ARP.hwsrc%%ARP.psrc%%ARP.hwdst%%ARP.pdst%'')
```

La idea es que si los paquetes ARP son de tipo `who-has` o `is-at`, guarda en el archivo de salida y muestra por pantalla los datos de MAC source, IP source, MAC dest e IP dest. Datos que podrán usarse luego para obtener información de la red.

Para la segunda parte de esta consigna se pide analizar la entropía de la red. Para esto primero se dejó corriendo el script anterior para capturar una buena cantidad de paquetes ARP en una red con tráfico considerable. El lapso de tiempo usado para la captura fue de 2 horas.

Se plantearon dos diferentes modelos para la fuente de información: En el primero, se definió como conjunto de símbolos a las direcciones de IP destino que aparecieran en los paquetes ARP capturados. La idea sería la de poder distinguir direcciones muy solicitadas de otras poco buscadas. El segundo modelo una como símbolos a las direcciones de IP origen. Esta vez la idea es caracterizar a los nodos por generar gran cantidad o poca cantidad de requests ARP.

Para cada modelo el calculo consiste en obtener la probabilidad estadística de cada simbolo sobre el total y en base a eso obtener la información de cada símbolo y la entropía de la fuente con las fórmulas presentadas en la sección de desarrollo.

2.3. Tercera consigna: Gráficos y análisis

En primer lugar, con los datos recolectados generamos un grafo dirigido (red_topologia.png) para visualizar la forma de nuestra red. Los nodos contienen las IPs de las maquinas involucradas y las aristas simbolizan un mensaje ARP donde el origen de la arista es la IP solicitante y el destino la IP solicitada del mensaje. (El grafo no se incluye en el informe por su gran tamaño, pero está en la carpeta con los archivos entregados).

Nota: Para el siguiente análisis solo se expondrán los símbolos más distinguidos basados en la información del mismo y la entropía de la fuente, debido a la gran cantidad de IPs.

2.3.1. Modelo 1: direcciones IP destino como símbolos

Entropia de la red: 3.39847619304

IP destino	Información
9.6.162.15	1.74553111348
9.6.162.1	1.85952788434
...	...
9.6.162.179	10.6733090756
9.6.162.138	10.6733090756
9.6.162.133	10.6733090756
9.6.162.173	10.6733090756
9.6.162.87	10.6733090756
192.168.1.3	10.6733090756
9.6.162.26	10.6733090756
9.6.162.60	10.6733090756
9.6.162.63	10.6733090756
9.6.162.103	10.6733090756

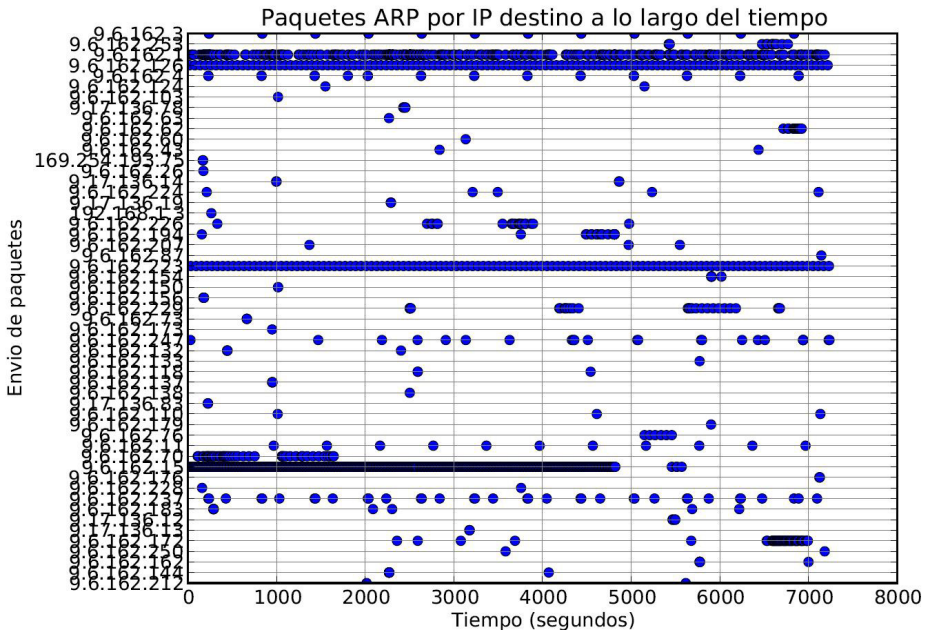

Figura 2: Actividad IPs solicitadas

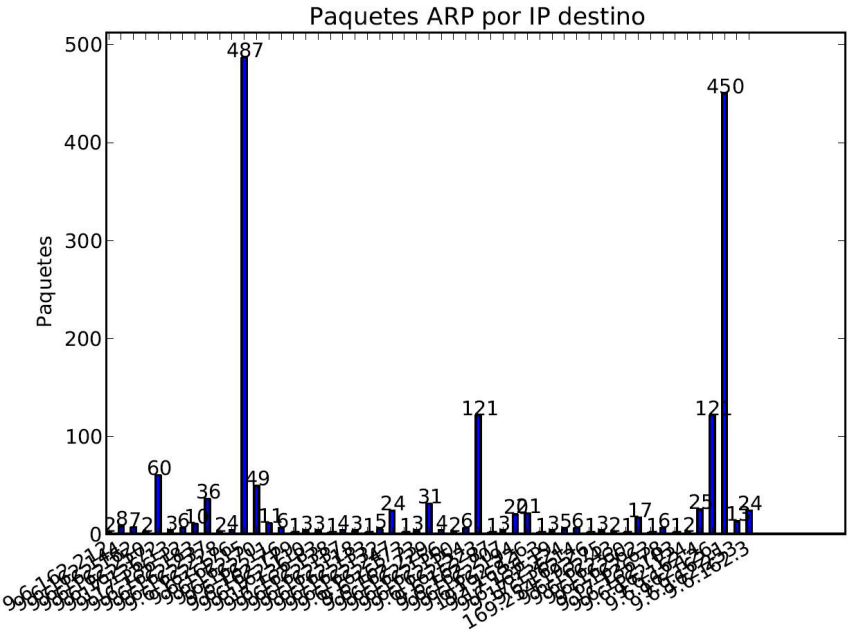

Figura 3: Histograma IPs solicitadas

2.3.2. Modelo 2: direcciones IP origen como símbolos

Entropia de la red: 3.71219167365

IP origen	Información
9.6.162.3	1.25545656068
...	...
9.6.162.12	9.08834657484
9.6.162.62	9.08834657484
9.6.162.15	9.67330907556
9.6.162.221	9.67330907556
9.6.162.150	9.67330907556
192.168.1.3	10.6733090756

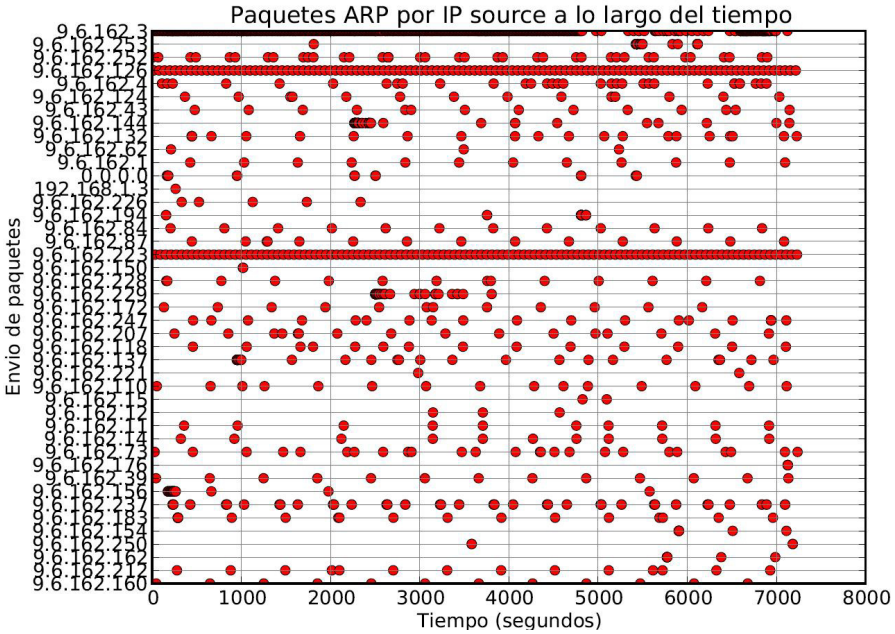

Figura 4: Actividad IPs solicitantes

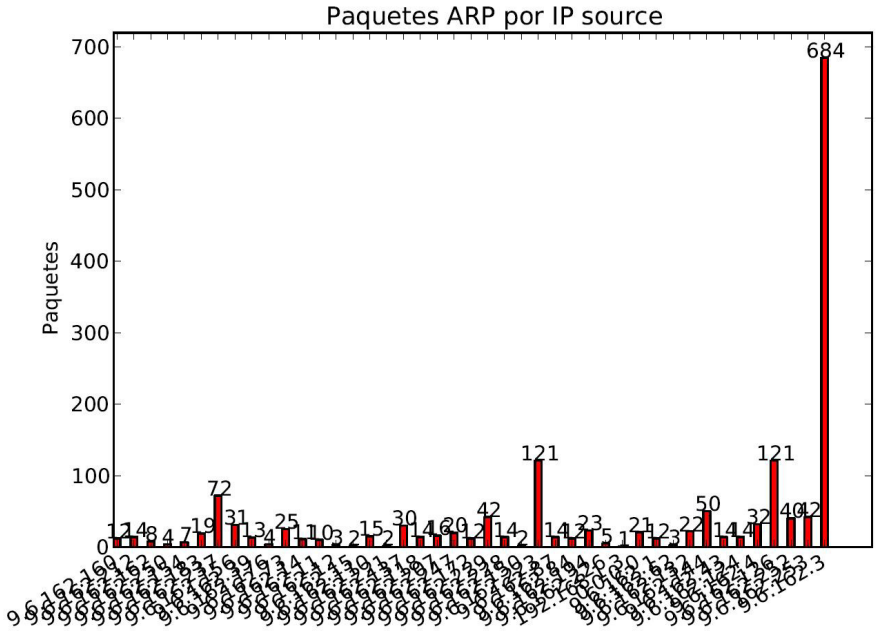

Figura 5: Histograma IPs solicitantes

3. Análisis de resultados

3.0.3. Modelo 1: direcciones IP destino como símbolos

Segun el histograma de IPs destino, las más solicitadas son: 9.6.162.1, 9.6.162.126, 9.6.162.223 y 9.6.162.15. En el gráfico de actividad en el tiempo, se ve que se solicitan en forma aproximadamente constante durante el transcurso de la captura de paquetes, salvo 9.6.162.15 que a los aprox. 5000 s deja de ser tan solicitada.

Si buscamos dichos nodos en el gráfico dirigido de la red, vemos lo siguiente:

- El nodo 9.6.162.1 recibe flechas desde casi todos los demás nodos. Esto nos indica que muy probablemente se trate del router.
- El nodo 9.6.162.15 es buscada desde los nodos 9.6.162.3 y 9.6.162.4.
- Los nodos 9.6.162.126 y 9.6.162.223 solo están conectados con si mismo. Es decir, lo unico que hacen es mandar ARP con su propia IP en broadcast a todos, como anunciándose.

Con respecto al último item, el que un nodo mande ARP con su IP, es un buen método para encontrar potenciales IPs duplicadas:

Si el nodo no obtiene respuesta, entonces es el único en la red con ese IP. Si recibe respuesta, quiere decir que hay otra computadora con la misma IP, lo que es un problema.

Además, esto permite al router y vecinos actualizar sus tablas ARP para que puedan comunicarse con dicho nodo.

Sin embargo, no nos queda claro por qué ambos nodos tienen este comportamiento de manera constante durante toda la captura de paquetes.

3.0.4. Modelo 2: direcciones IP origen como símbolos

Segun el histograma de IPs origen, las más solicitadas son: 9.6.162.3, 9.6.162.126 y 9.6.162.223 y 9.6.162.15.

No es sorpresa ver a las últimas 2 IPs, ya que del análisis anterior surge que son nodos que no pararon de mandar ARP con su propia IP.

El nodo 9.6.162.3 sin embargo, presenta gran actividad y se conecta con varios nodos. Uno de los nodos con los que más interacción tiene es el 9.6.162.15. Esto ayuda a explicar por qué el nodo 9.6.162.15 aparece tantas veces como destino, en el gráfico del caso anterior. Sin embargo no pudimos explicar por qué este nodo en particular posee tanta actividad.

Algo particular que también se puede observar en los gáficos, es que aparece la IP 0.0.0.0 como nodo origen. Esto podría explicarse si se tratara de nodos que aún no tenían otorgada una dirección IP.

4. Conclusiones

En la primera parte, las pruebas realizadas sobre el script implementado en Scapy nos permitieron conocer particularidades del ARP, como por ejemplo la posibilidad de enviar la propia IP del nodo solicitante para detectar direcciones redundantes en la red.

Luego, en la segunda y tercer parte, podemos ver la utilidad del análisis del tráfico de mensajes ARP para obtener información relevante de una red. El gráfico topológico que puede obtenerse con las direcciones origen y destino de los mensajes capturados, permite ver de manera simple qué nodos tienen un comportamiento distinguido. El análisis de este gráfico, complementado con el cálculo de la entropía de los nodos, permite identificar fácilmente al router, por ejemplo, asi como otros nodos que tengan un comportamiento anormal y que puedan perjudicar el funcionamiento de la red.

Con respecto al cálculo de la entropía en los dos modelos planteados: Como vimos en la sección de Análisis de Resultados, concluimos que el primero, en el que los símbolos son las direcciones IP destino, nos sirve para

hallar los nodos más solicitados, entre los cuales es esperable que se encuentre el router. El segundo modelo, en cambio, nos muestra qué nodos tienen alta actividad en la red, y entre ellos podemos encontrar algunos con comportamiento anómalo, como los que hallamos en la red analizada que enviaban requests ARP con su propia IP de manera contínua.

Referencias

[1] Computer Networks - A Systems Approach 5th ed. - L. Peterson, et. al

[2] http://es.wikipedia.org/wiki/Entropía_(información)

[3] http://serverfault.com/questions/219837/why-my-laptop-sends-arp-request-to-itself

5. Anexo A: Código entregado

Junto al informe se entregó la captura de datos de la red analizada y el código pedido para completar las consignas y realizar el análisis.
La entrega consiste en los siguientes archivos:

1. redes.py: archivo principal que contiene las funciones para cumplir con las consignas.
2. analisis.py: contiene las funciones para el calculo de la entropía y el gráfico de los datos capturados.
3. __init__.py: archivo necesario por python.
4. oficina.txt: archivo con la captura de datos de la red analizada.
5. Figuras: carpeta con las figuras presentadas en el informe para mayor visibilidad.

Para utilizar el código entregado es necesario ejecutar **redes.py** el cual presenta el siguiente menú:
****

Redes - wiretapping
1. Buscar MAC address de una ip
2. Sniffear paquetes ARP
3. Calcular entropia paquetes sniffeados
4. Graficar paquetes sniffeados
0. Salir, volver atras
****

Opcion:

1. Se corresponde con la primer consigna del enunciado, pide una dirección de ip y al ingresarla busca la MAC asociada mediante un mensaje ARP.
2. Se corresponde con la segunda consigna del enunciado, pide el nombre del archivo de salida y guarda allí los datos capturados de la red con el siguiente formato: "MAC origenIP origenMAC destinoIP destino Tiempo".
3. Pide como entrada un archivo con datos capturados de una red utilizando el formato de la opción 2 y genera archivos de salida con el cálculo de la entropía de la red. los, símbolos y la información de cada uno de ellos.
4. Genera histogramas, gráficos de actividad a lo largo del tiempo y de topología de la red, a partir de un archivo con datos capturados de una red.

Requisitos:

- Para utilizar las opciones 1 y 2 es necesario tener instalado **scapy**.
- Para utilizar la opcion 4 es necesario tener los siguientes módulos de python instalados: **matplotlib**, **pydot**, **numpy** y **graphviz**.

6. Anexo B: Topología de la red

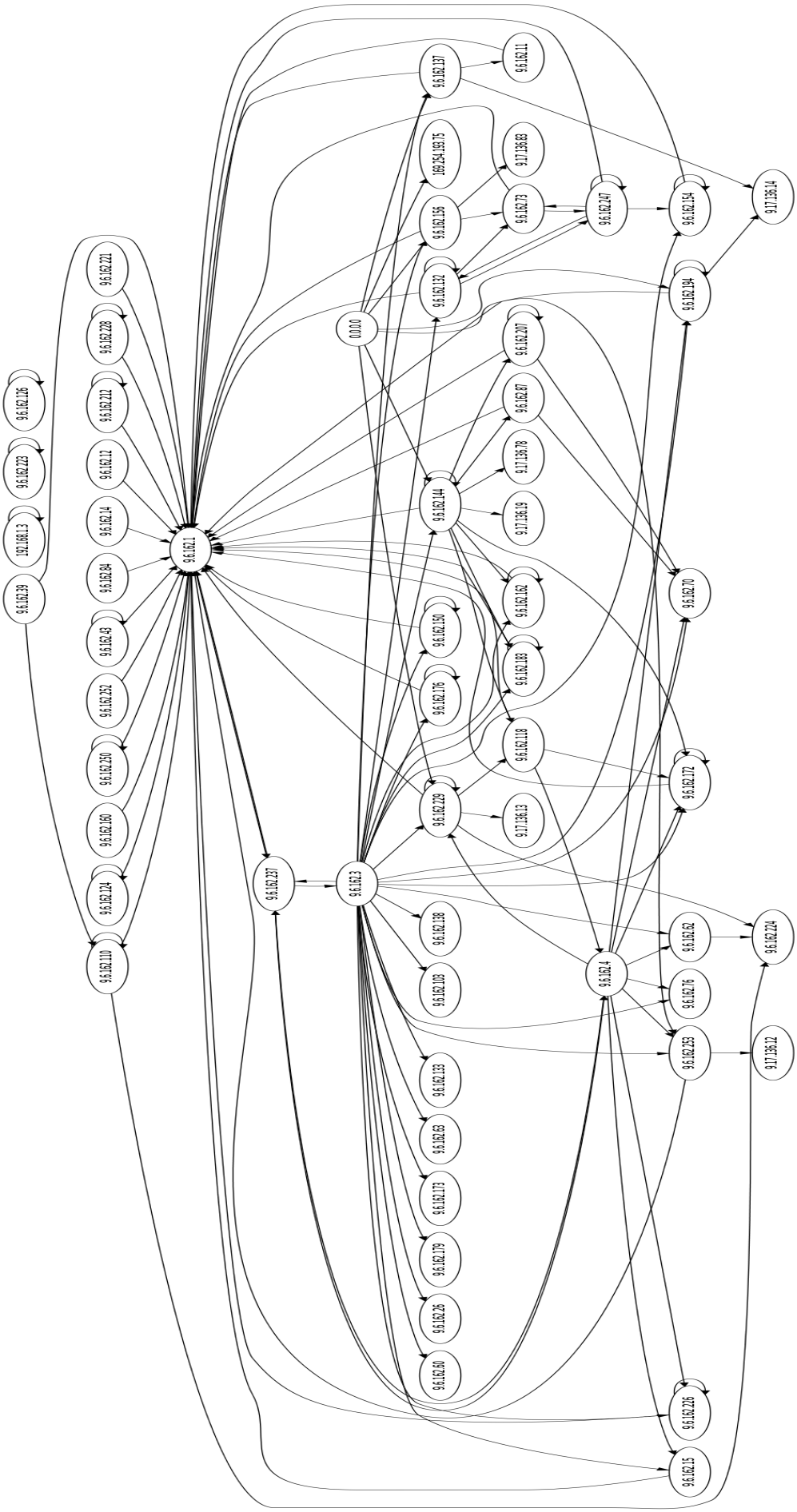

Figura 6: Topología de la red analizada