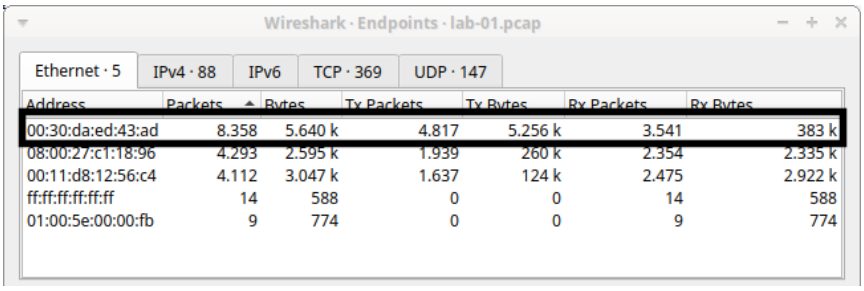
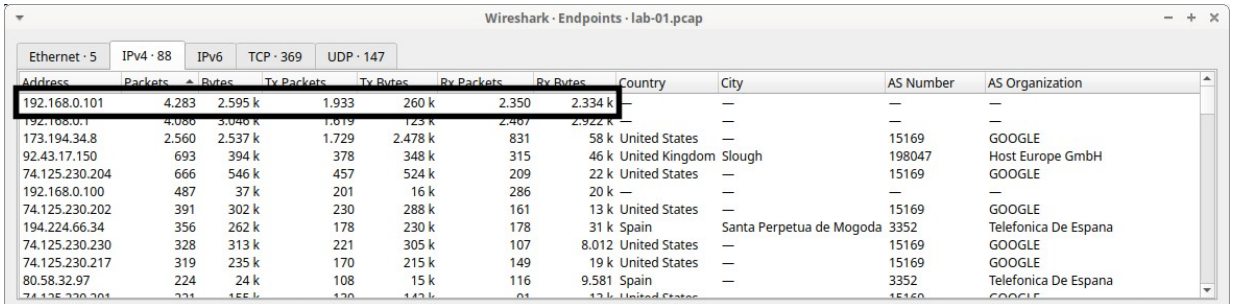
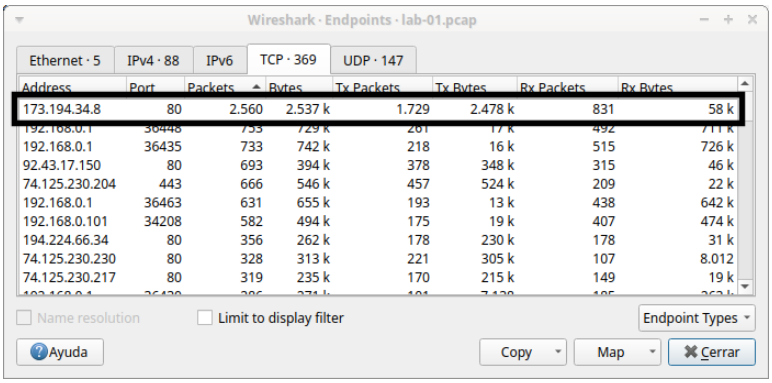
1. Abre con Wireshark el archivo lab_-01.pcap:
   a. ESTADÍSTICAS: Analiza la captura y responde:
      - Nº de equipos que a nivel de enlace intervienen en la captura: **en la captura hay 5 direcciones MAC. Una de ellas es la de broadcast (ff:ff:ff:ff:ff:ff) por lo que serían 4 equipos. NOTA: siendo estrictos y fijándonos en la dirección 01:00:5e:00:00:fb vemos que tiene el bit I/G a 1 y se trata de una dirección multicas; por lo que la respuesta correcta sería 3 equipos.**
      - Qué equipo a nivel de enlace es el más activo (indica nº total de paquetes, paquetes enviados y recibidos, total de Bytes, total de Bytes enviados y total de Bytes recibidos):



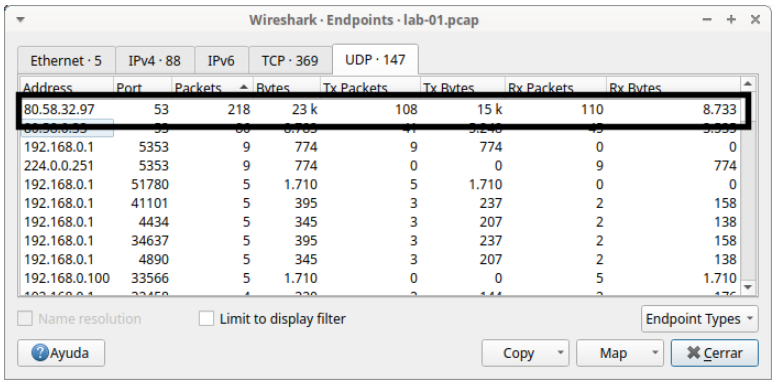
      - Repite el apartado anterior para el nivel de red, TCP y UDP.



Nivel de red

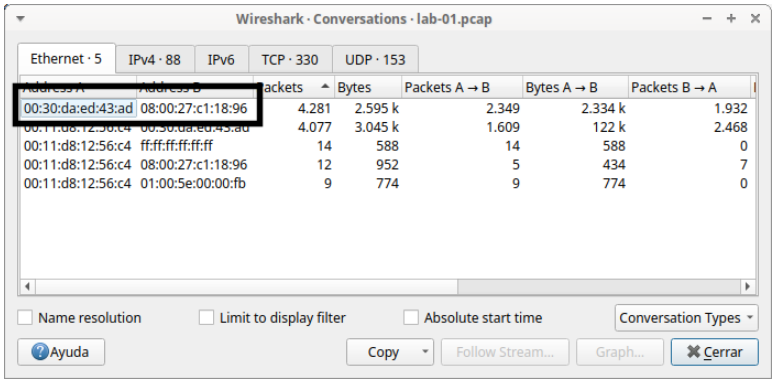


Nivel de transporte TCP

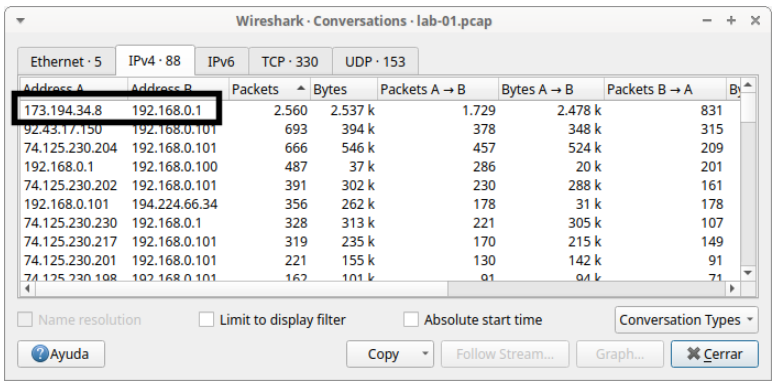


Nivel de transporte UDP

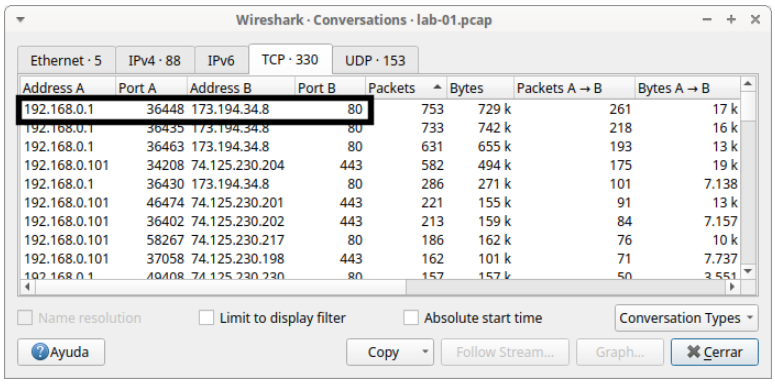
      - Indica los elementos participantes de nivel de enlace, ip, tcp y udp, de la conversación más activa en cada nivel del TCP/IP.



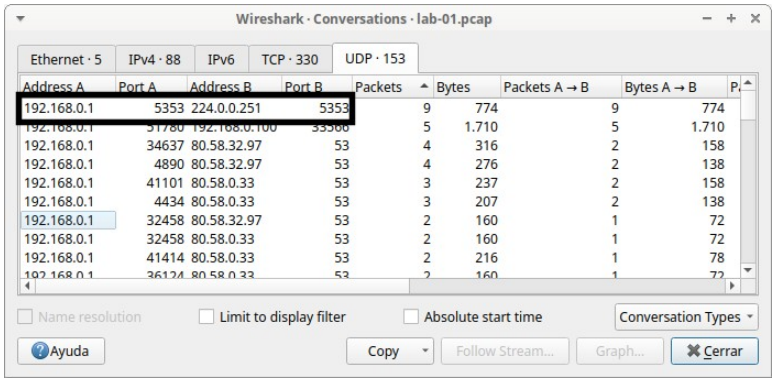
Nivel de enlace: 00:30:da:ed:43:ad ↔ 08:00:27:c1:18:96



Nivel de red: 173.194.34.8 ↔ 192.168.0.1



Nivel de transporte TCP: 192.168.0.1:36448 ↔ 173.194.34.8:80



Nivel de transporte UDP: 192.168.0.1:5353 ↔ 224.0.0.251:5353

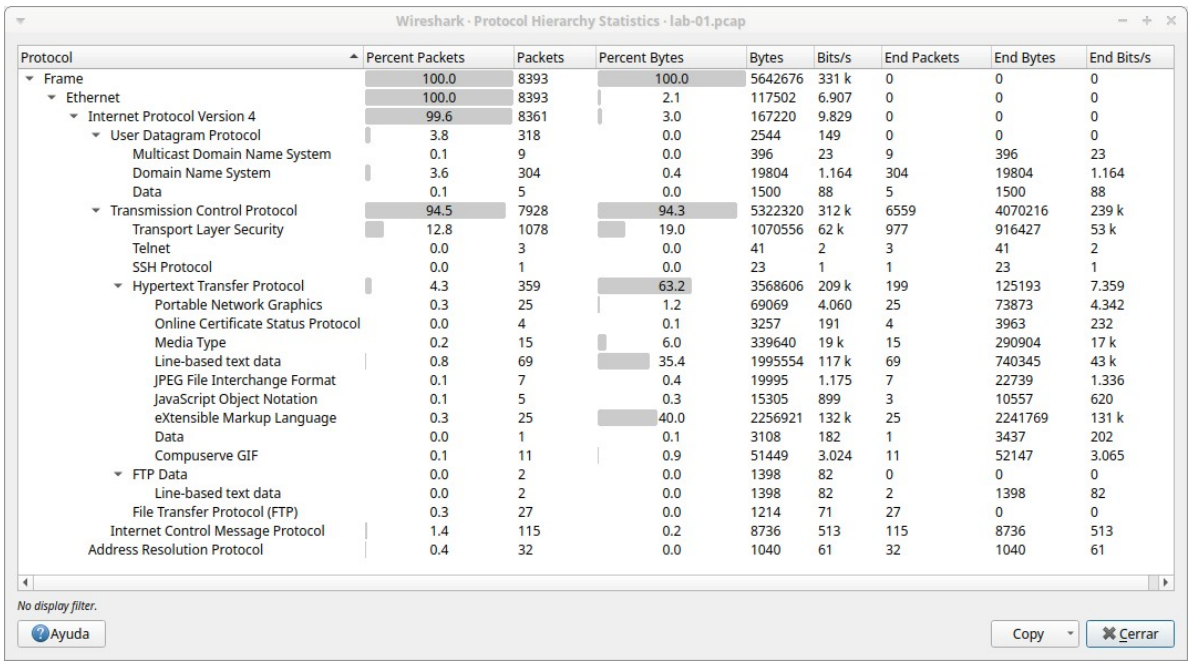
NOTAS:

- A nivel de transporte se trabaja con sockets[1] por lo que hay que indicar IP:puerto
- En el caso de UDP, la *conversación* más activa son envíos multicast lanzados por el equipo 192.168.0.1. Entre equipos reales sería la que aparece en segundo lugar, que

1 Socket: <protocolo de transporte, dirección IP, puerto>. Por ejemplo: <tcp, 18.75.4.1, 443>

como se ve son dirección de tipo unicast (de equipos).

- Teniendo en cuenta la jerarquía de protocolos haz un dibujo del modelo TCP/IP usando los protocolos presentes en la captura.



Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	8393	100.0	5642676	331 k	0	0	0
Ethernet	100.0	8393	2.1	117502	6.907	0	0	0
Internet Protocol Version 4	99.6	8361	3.0	167220	9.829	0	0	0
User Datagram Protocol	3.8	318	0.0	2544	149	0	0	0
Multicast Domain Name System	0.1	9	0.0	396	23	9	396	23
Domain Name System	3.6	304	0.4	19804	1.164	304	19804	1.164
Data	0.1	5	0.0	1500	88	5	1500	88
Transmission Control Protocol	94.5	7928	94.3	5322320	312 k	6559	4070216	239 k
Transport Layer Security	12.8	1078	19.0	1070556	62 k	977	916427	53 k
Telnet	0.0	3	0.0	41	2	3	41	2
SSH Protocol	0.0	1	0.0	23	1	1	23	1
Hypertext Transfer Protocol	4.3	359	63.2	3568606	209 k	199	125193	7.359
Portable Network Graphics	0.3	25	1.2	69069	4.060	25	73873	4.342
Online Certificate Status Protocol	0.0	4	0.1	3257	191	4	3963	232
Media Type	0.2	15	6.0	339640	19 k	15	290904	17 k
Line-based text data	0.8	69	35.4	1995554	117 k	69	740345	43 k
JPEG File Interchange Format	0.1	7	0.4	19995	1.175	7	22739	1.336
JavaScript Object Notation	0.1	5	0.3	15305	899	3	10557	620
eXtensible Markup Language	0.3	25	40.0	2256921	132 k	25	2241769	131 k
Data	0.0	1	0.1	3108	182	1	3437	202
Compuserve GIF	0.1	11	0.9	51449	3.024	11	52147	3.065
FTP Data	0.0	2	0.0	1398	82	0	0	0
Line-based text data	0.0	2	0.0	1398	82	2	1398	82
File Transfer Protocol (FTP)	0.3	27	0.0	1214	71	27	0	0
Internet Control Message Protocol	1.4	115	0.2	8736	513	115	8736	513
Address Resolution Protocol	0.4	32	0.0	1040	61	32	1040	61

Nivel aplicación	Telnet, SSH, HTTP, FTP, TLS	DNS, MDNS
Nivel de transporte	TCP	UDP
Nivel de internet	IP, ICMP, ARP	
Network interface and hardware	Ethernet	

b. Haz los siguientes DISPLAY FILTERS:

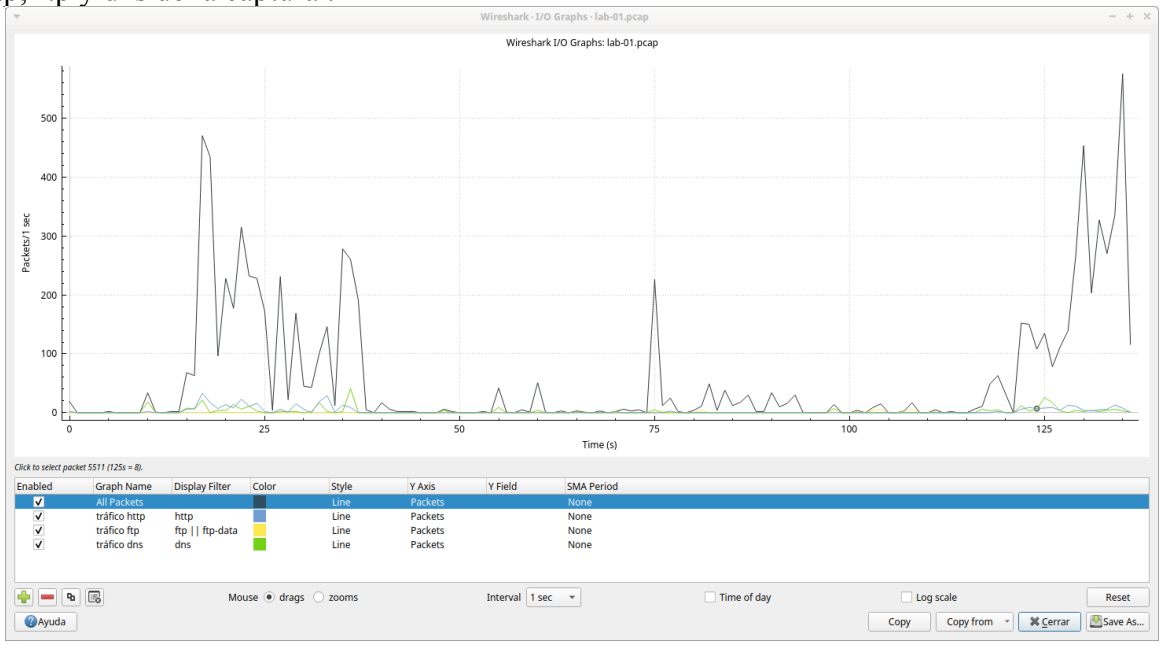
Nombre	**Acción**
Broadcast	eth.addr == ff:ff:ff:ff:ff:ff
Eth. origen PC1	eth.src == 00:11:d8:12:56:c4
Eth. destino PC2	eth.dst == 00:30:da:ed:43:ad
Eth. PC3	eth.addr == 01:00:5e:00:00:fb
Eth. PC2 ↔ PC3	eth.addr == 00:11:d8:12:56:c4 && eth.addr == 01:00:5e:00:00:fb
ARP	arp
ARP Request	arp.opcode == 0x0001
Eth. NO PC1	! eth.addr == 00:11:d8:12:56:c4
IP origen 50.57.203.250	ip.src == 50.57.203.250
IP destino 192.16.192.166	ip.dst ==192.16.192.166
IP_101 ↔ IP_80.58.32.97	ip.addr == 192.168.0.101 && ip.addr == 80.58.32.97
IP_80.58.32.97	ip.addr == 80.58.32.97
NO IP_80.58.32.97	! ip.addr == 80.58.32.97
ICMP	icmp
ICMP Echo Request	icmp.type == 8

ICMP Tiempo Excedido	icmp.type == 11
TTL 1	ip.ttl == 1
TTL 6	ip.ttl <= 6
ICMP IP_74 ↔ IP_1	(ip.addr == 192.168.0.100 && ip.addr == 192.168.0.20) && (icmp.type == 0 || icmp.type == 8)
DNS	udp.srcport == 53
www.atareao.es	ip.dst == 92.43.17.150 && tcp.dstport == 80
Atareao.es	ip matches “atareao.es”
Atareao.es_js	http.request.full_uri matches "js$" Siendo más estrictos: http.request.full_uri matches "\\.js$"
FTP-Control	Sol. #1: ip.addr == 192.168.0.1 && ip.addr == 130.206.1.5 && ftp Sol. #2: ip.addr == 192.168.0.1 && ip.addr == 130.206.1.5 && tcp.port==21 La diferencia está en que la Sol.#1 únicamente muestra los mensajes FTP de control que transportan datos ftp y la segunda muestra todos los mensajes incluyendo los tcp de establecimiento de conexión, fin, ...
FTP-Data	ip.addr == 192.168.0.1 && ip.addr == 130.206.1.5 && ftp-data

c. Crea, siguiendo el orden indicado, las siguientes reglas de coloreado:

Nombre	**Acción**
ARP	arp
Errores ICMP	icmp.type == 3 || icmp.type == 11 || icmp.type == 5 || icmp.type == 4
ICMP	icmp
TTL	ip.ttl <= 10

d. GRÁFICOS: Haz un gráfico donde se vea a la vez la evolución temporal del tráfico de los protocolos http, ftp y dns de la captura .



e. GeoIP: Instala y activa las librerías necesarias para tener Geolocalización en Wireshark e indica el filter string para ver únicamente los paquetes enviados desde España
ip.geoip.src_country == "Spain"