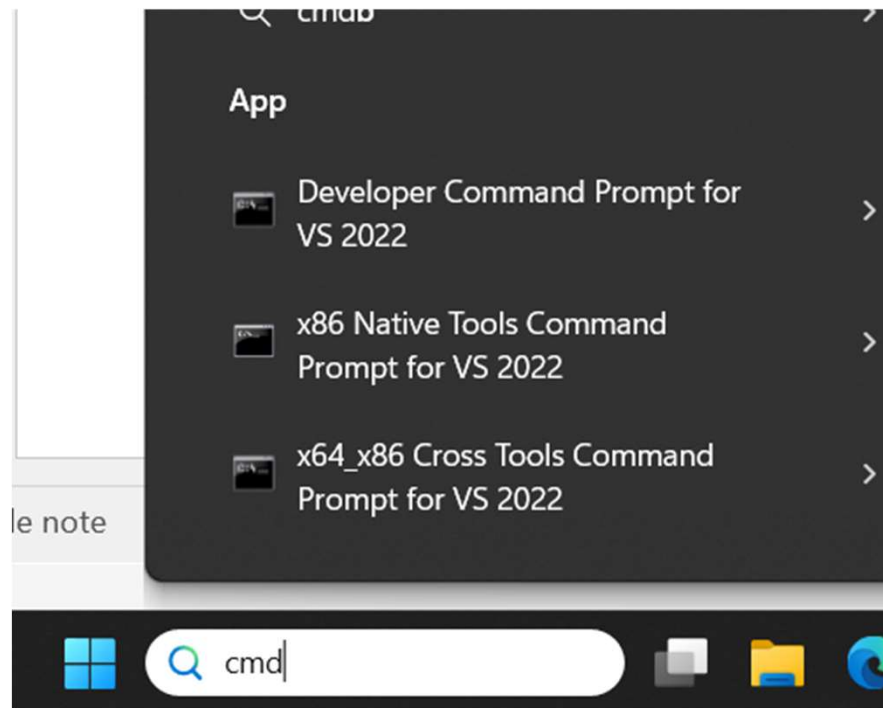


Wireshark

Wireshark è un analizzatore del traffico di rete, possiamo quindi controllare tutte le informazioni che viaggiano in rete nel formato di pacchetti sia nella rete locale che su internet, se vogliamo fare una semplice prova per controllare cosa accade possiamo aprire il prompt dei comandi scrivendo «cmd» nella finestra di ricerca



Wireshark

Possiamo poi (attraverso il comando ping) effettuare il controllo di una connessione con la sezione italiana di Google (SI È NOTATO CHE L'INDIRIZZO DI GOOGLE.IT CAMBIA PERIODICAMENTE, NONOSTANTE L'INDIRIZZO IP RESTI ATTIVO, QUESTO PERCHÉ OVVIAMENTE VENGONO USATI DIVERSI SERVER E QUINDI VENIAMO «REINDIRIZZATI» A QUELLO MAGGIORMENTE LIBERO, POICHÉ IN QUESTO ESERCIZIO DOVREMO EFFETTUARE UN NUOVO PING UTILizzeremo QUINDI L'INDIRIZZO IP E NON «WWW.GOOGLE.IT»)

```
C:\Users\alber>ping www.google.it

Esecuzione di Ping www.google.it [216.58.209.35] con 32 byte di dati:
Risposta da 216.58.209.35: byte=32 durata=58ms TTL=114
Risposta da 216.58.209.35: byte=32 durata=58ms TTL=114
Risposta da 216.58.209.35: byte=32 durata=72ms TTL=114
Risposta da 216.58.209.35: byte=32 durata=58ms TTL=114

Statistiche Ping per 216.58.209.35:
    Pacchetti: Trasmessi = 4, Ricevuti = 4,
    Persi = 0 (0% persi),
Tempo approssimativo percorsi andata/ritorno in millisecondi:
    Minimo = 58ms, Massimo = 72ms, Medio = 61ms
```

Wireshark

Il comando ping come definito da Wikipedia è:

Ping (Packet internet groper) è un'utility di amministrazione per reti di computer usata per misurare il tempo, espresso in millisecondi, impiegato da uno o più pacchetti ICMP a raggiungere un dispositivo di rete (attraverso una qualsiasi rete informatica basata su IP) e a ritornare indietro all'origine. È prettamente utilizzato per verificare la presenza e la raggiungibilità di un altro computer connesso in rete e per misurare le latenze di trasmissione di rete.

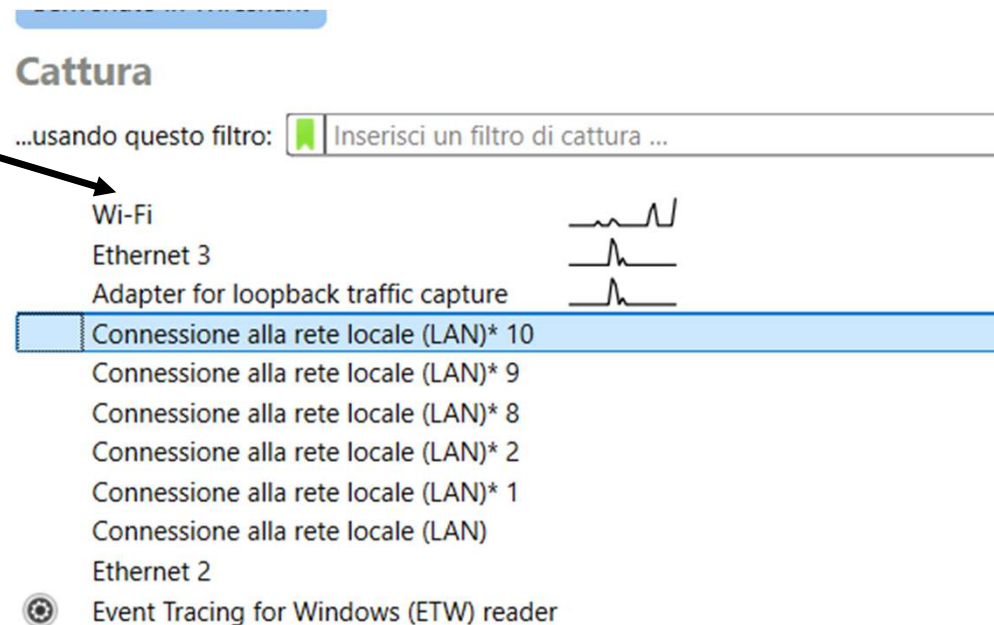
Un pacchetto ICMP è:

In telecomunicazioni e informatica l'Internet Control Message Protocol (ICMP) è un protocollo di servizio per reti a pacchetto che si occupa di trasmettere informazioni riguardanti malfunzionamenti, informazioni di controllo o messaggi tra i vari componenti di una rete di calcolatori.

In altre parole quando si vuole controllare se un host di rete risponde o meno si può usare questo comando e il pacchetto che viene inviato, una volta ricevuto viene «rispedito» al mittente

Wireshark

SENZA CHIUDERE IL PROMPT,
possiamo quindi aprire il software
wireshark e scegliere con il doppio
click l'interfaccia di connessione
della ethernet (se siamo collegati a
internet tramite una ethernet)
oppure wifi (io sto usando il wifi)
per fare partire una ricerca dei
pacchetti. LE RETI CHE STANNO
MOVIMENTANDO DATI
OVVIAMENTE VENGONO POSTE IN
PRIMO PIANO PER MOSTRARE
L'ANDAMENTO DEL TRAFFICO



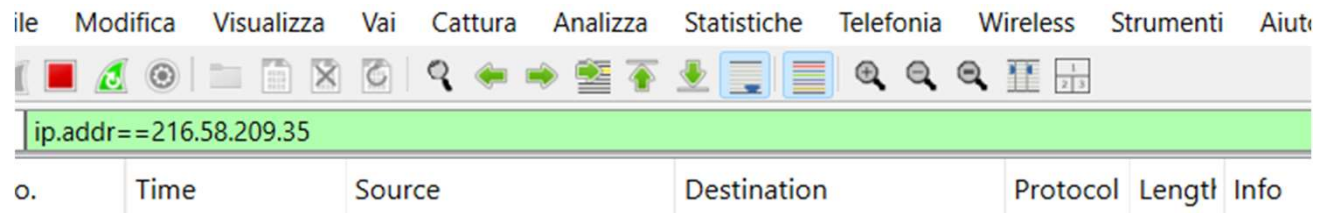
Wireshark

ANCHE SE NON ABBIAMO FATTO ALCUNA OPERAZIONE MANUALE se la cattura ha avuto inizio possiamo cominciare già a notare del traffico di rete di vario tipo, quel traffico per ora non ci interessa, a noi interessa il traffico con google.it. Possiamo quindi dal prompt copiare l'indirizzo ip di questo server di google usando il Ctrl+C Ctrl+V

```
C:\Users\alber>ping www.google.it  
  
Esecuzione di Ping www.google.it [216.58.209.35] con 32 byte di dati:  
Risposta da 216.58.209.35: byte=32 durata=58ms TTL=114  
Risposta da 216.58.209.35: byte=32 durata=58ms TTL=114  
Risposta da 216.58.209.35: byte=32 durata=72ms TTL=114  
Risposta da 216.58.209.35: byte=32 durata=58ms TTL=114
```

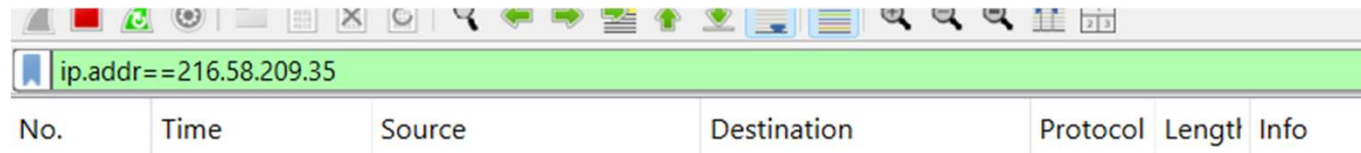
Wireshark

Nella riga dei filtri di visualizzazione di Wireshark si inserisce questa istruzione che permette di filtrare i risultati del traffico in base all'indirizzo (sia in entrata che in uscita)



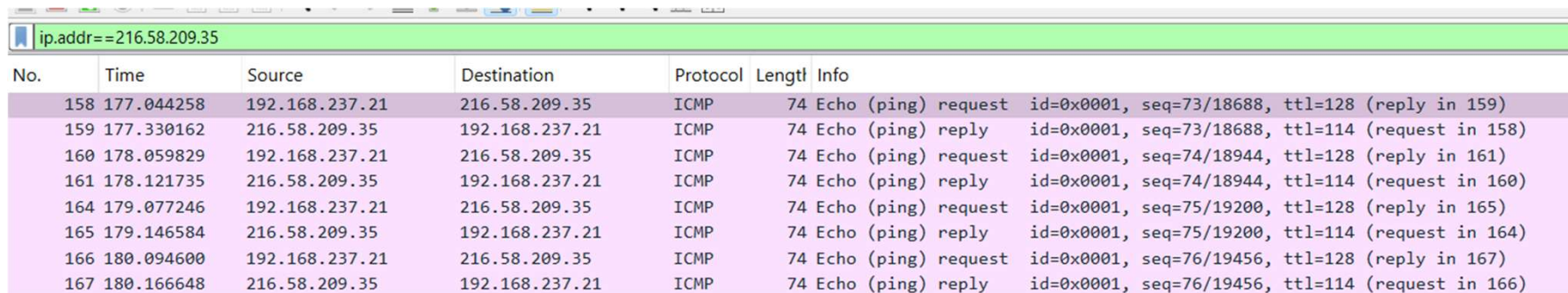
Wireshark

E poi premere Invio, se presente altro traffico **questo sparirà** perché abbiamo detto a Wireshark che vogliamo visualizzare solo quel traffico



Wireshark

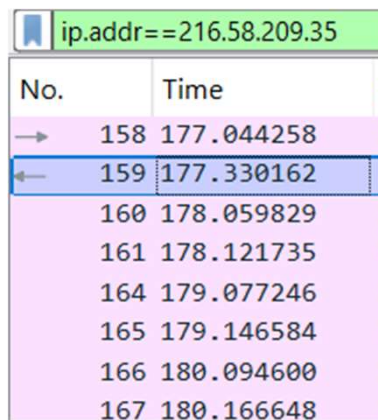
Quindi possiamo provare NELLA STESSA FINESTRA DEL PROMPT a pingare di nuovo QUESTA VOLTA L'INDIRIZZO IP E NON IL NOME WWW.GOOGLE.IT per ottenere il traffico di rete relativo alla comunicazione che ovviamente sarà solo quella del controllo della connessione (che è il più semplice cioè Ping), i pacchetti ICMP del Ping sono visualizzati in un colore indaco (i tipi di pacchetti vengono colorati in forme diverse)



No.	Time	Source	Destination	Protocol	Length	Info
158	177.044258	192.168.237.21	216.58.209.35	ICMP	74	Echo (ping) request id=0x0001, seq=73/18688, ttl=128 (reply in 159)
159	177.330162	216.58.209.35	192.168.237.21	ICMP	74	Echo (ping) reply id=0x0001, seq=73/18688, ttl=114 (request in 158)
160	178.059829	192.168.237.21	216.58.209.35	ICMP	74	Echo (ping) request id=0x0001, seq=74/18944, ttl=128 (reply in 161)
161	178.121735	216.58.209.35	192.168.237.21	ICMP	74	Echo (ping) reply id=0x0001, seq=74/18944, ttl=114 (request in 160)
164	179.077246	192.168.237.21	216.58.209.35	ICMP	74	Echo (ping) request id=0x0001, seq=75/19200, ttl=128 (reply in 165)
165	179.146584	216.58.209.35	192.168.237.21	ICMP	74	Echo (ping) reply id=0x0001, seq=75/19200, ttl=114 (request in 164)
166	180.094600	192.168.237.21	216.58.209.35	ICMP	74	Echo (ping) request id=0x0001, seq=76/19456, ttl=128 (reply in 167)
167	180.166648	216.58.209.35	192.168.237.21	ICMP	74	Echo (ping) reply id=0x0001, seq=76/19456, ttl=114 (request in 166)

Wireshark

All'inizio non appaiono, ma se facciamo click in una riga qualsiasi (in questo caso la seconda), notiamo prima di tutto che quando selezioniamo un pacchetto appaiono sulla sinistra due frecce che indicano quella verso destra l'uscita (il client richiede qualcosa al server) e quella verso sinistra una risposta

A screenshot of the Wireshark packet list pane. At the top, a filter bar shows 'ip.addr==216.58.209.35'. Below it is a table with two columns: 'No.' and 'Time'. The table contains several rows of network traffic. The second row, with packet number 159, is highlighted in blue and has a double-headed arrow icon to its left. The other rows have single-headed arrows: right-pointing for outgoing traffic and left-pointing for incoming traffic. The background of the packet list is light pink.

No.	Time
→ 158	177.044258
↔ 159	177.330162
160	178.059829
161	178.121735
164	179.077246
165	179.146584
166	180.094600
167	180.166648

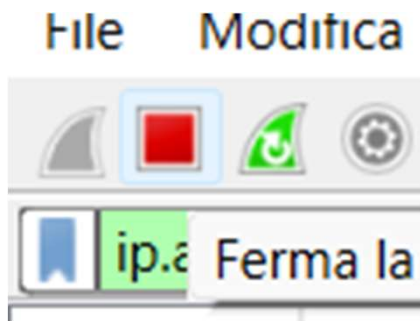
Wireshark

Nella colonna Time vediamo i tempi a partire ovviamente da un cronometro che parte da 0 e va per microsecondi, poi abbiamo l'indirizzo IP dal quale il pacchetto parte (Source), quello nel quale arriva (Destination), il numero di pacchetto progressivo (in questo caso notiamo che Wireshark prima di quelli di Ping aveva già catturato altri pacchetti), il tipo di pacchetto (il ping usa i pacchetti ICMP), la lunghezza del pacchetto in byte (74 per tutti), il tipo di richiesta (Request che ovviamente parte dal nostro pc, Reply a una richiesta che ovviamente arriva al nostro Pc), poi abbiamo un id che ci dice che lo scambio dei dati avviene con la modalità big endian (0X0001) cioè la sequenza dei bit avviene con quello più significativo a sinistra e quello più piccolo a destra, la scritta «Seq» indica il numero di richiesta di tipo Ping catturata fra tutte (tant'è vero che sono raggruppate in numero progressivo) e la indica con modalità big endian/little endian cioè con i bit nell'ordine più piccolo più grande e il contrario. Il ttl è il tempo di «vita» previsto per il pacchetto, ovvero il tempo previsto entro il quale il pacchetto non può più essere ritenuto valido (per la richiesta 128 microsecondi, per la risposta 114), la scritta «Reply in» comunica il numero di pacchetto (tra tutti) nel quale troviamo la risposta o la richiesta in base al fatto che la riga selezionata sia di richiesta o di risposta.

ip.addr==216.58.209.35							
No.	Time	Source	Destination	Protocol	Length	Info	
→ 158	177.044258	192.168.237.21	216.58.209.35	ICMP	74	Echo (ping) request	id=0x0001, seq=73/18688, ttl=128 (reply in 159)
← 159	177.330162	216.58.209.35	192.168.237.21	ICMP	74	Echo (ping) reply	id=0x0001, seq=73/18688, ttl=114 (request in 158)
160	178.059829	192.168.237.21	216.58.209.35	ICMP	74	Echo (ping) request	id=0x0001, seq=74/18944, ttl=128 (reply in 161)
161	178.121735	216.58.209.35	192.168.237.21	ICMP	74	Echo (ping) reply	id=0x0001, seq=74/18944, ttl=114 (request in 160)
164	179.077246	192.168.237.21	216.58.209.35	ICMP	74	Echo (ping) request	id=0x0001, seq=75/19200, ttl=128 (reply in 165)
165	179.146584	216.58.209.35	192.168.237.21	ICMP	74	Echo (ping) reply	id=0x0001, seq=75/19200, ttl=114 (request in 164)
166	180.094600	192.168.237.21	216.58.209.35	ICMP	74	Echo (ping) request	id=0x0001, seq=76/19456, ttl=128 (reply in 167)
167	180.166648	216.58.209.35	192.168.237.21	ICMP	74	Echo (ping) reply	id=0x0001, seq=76/19456, ttl=114 (request in 166)



Wireshark

La cattura può essere salvata in un file, prima basta fermarla premendo il pulsante rosso



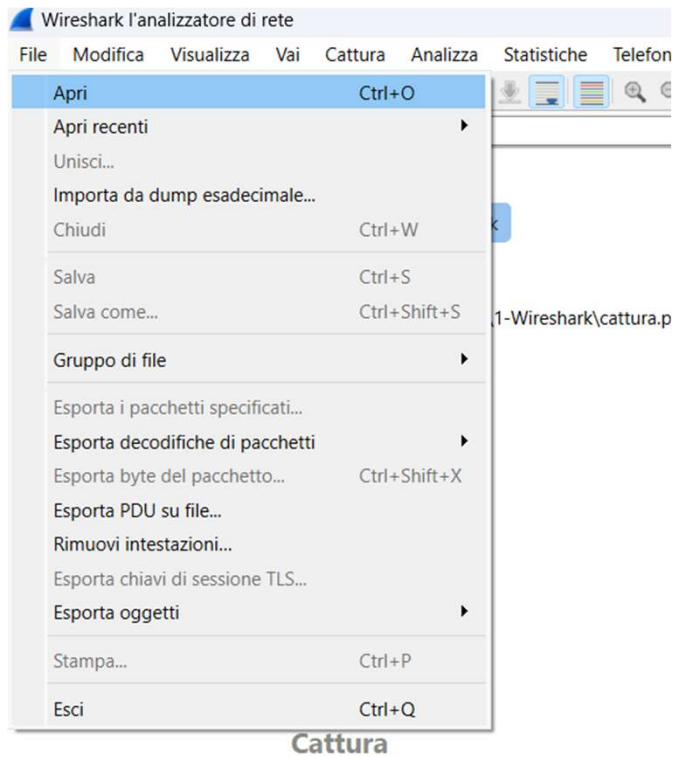
Wireshark

E poi salvata in un file per la sua riapertura

Apri	Ctrl+O	
Apri recenti		
Unisci...		
Importa da dump esadecimale...		
Chiudi	Ctrl+W	
Salva	Ctrl+S	
Salva come...	Ctrl+Shift+S	
Gruppo di file		
Esporta i pacchetti specificati...		
Esporta decodifiche di pacchetti		
Esporta byte del pacchetto...	Ctrl+Shift+X	
Esporta PDU su file...		
Rimuovi intestazioni...		
Esporta chiavi di sessione TLS...		
Esporta oggetti		

Wireshark

Ovviamente se chiudiamo e riapriamo Wireshark, dalla finestra iniziale possiamo riaprire anche il file



Wireshark

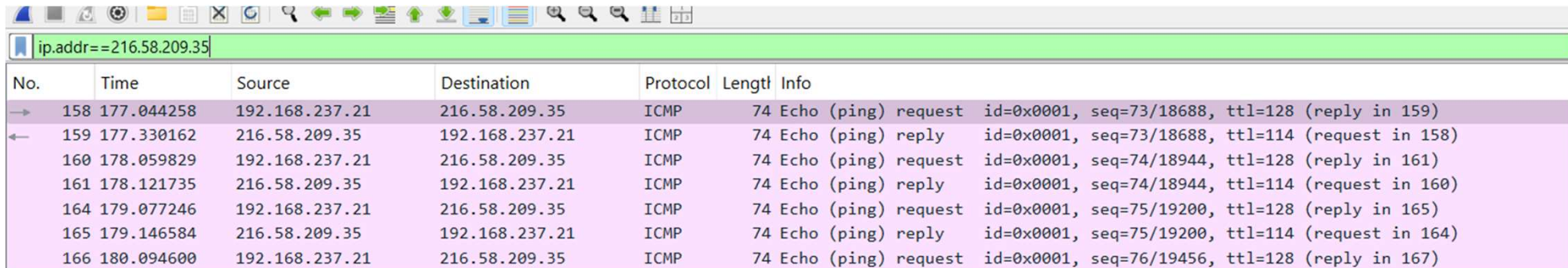
Attenzione però: nel file che viene salvato vengono salvati TUTTI I DATI DELLE CATTURE, quindi quando riapro il file senza usare di nuovo il filtro, questo inizialmente mi fa apparire tutti i movimenti

Applica un filtro di visualizzazione ... <Ctrl>F

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	13.107.246.254	192.168.237.21	TLSv1.2	78	Application Data
2	0.000088	192.168.237.21	13.107.246.254	TCP	54	49955 → 443 [ACK] Seq=1 Ack=25 Win=1019 Len=0
3	0.027614	13.107.246.254	192.168.237.21	TCP	54	443 → 49955 [FIN, ACK] Seq=25 Ack=1 Win=83 Len=0
4	0.027614	13.107.246.254	192.168.237.21	TCP	54	[TCP Retransmission] 443 → 49955 [FIN, ACK] Seq=25 Ack=1 Win=83 Len=0
5	0.027670	192.168.237.21	13.107.246.254	TCP	54	49955 → 443 [ACK] Seq=1 Ack=26 Win=1019 Len=0
6	4.958747	192.168.237.21	20.250.77.142	TCP	55	49776 → 443 [ACK] Seq=1 Ack=1 Win=260 Len=1 [TCP PDU reassembled in 55]
7	5.079276	20.250.77.142	192.168.237.21	TCP	66	443 → 49776 [ACK] Seq=1 Ack=2 Win=251 Len=0 SLE=1 SRE=2
8	8.670001	192.168.237.21	20.189.173.3	TCP	66	49967 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
9	8.938033	20.189.173.3	192.168.237.21	TCP	66	443 → 49967 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1310 WS=256 SACK_PERM
10	8.938168	192.168.237.21	20.189.173.3	TCP	54	49967 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
11	8.939412	192.168.237.21	20.189.173.3	TLSv1.3	445	Client Hello (SNI=self.events.data.microsoft.com)
12	9.152475	20.189.173.3	192.168.237.21	TLSv1.3	153	Hello Retry Request, Change Cipher Spec
13	9.152550	192.168.237.21	20.189.173.3	TCP	54	49967 → 443 [ACK] Seq=392 Ack=100 Win=261888 Len=0
14	9.172526	192.168.237.21	20.189.173.3	TLSv1.3	516	Change Cipher Spec, Client Hello (SNI=self.events.data.microsoft.com)
15	9.418467	20.189.173.3	192.168.237.21	TLSv1.3	1364	Server Hello
16	9.418467	20.189.173.3	192.168.237.21	TCP	1364	443 → 49967 [ACK] Seq=1410 Ack=854 Win=4193536 Len=1310 [TCP PDU reassembled in 16]
17	9.418467	20.189.173.3	192.168.237.21	TCP	1364	443 → 49967 [ACK] Seq=2720 Ack=854 Win=4193536 Len=1310 [TCP PDU reassembled in 17]
18	9.418467	20.189.173.3	192.168.237.21	TCP	1364	443 → 49967 [ACK] Seq=4030 Ack=854 Win=4193536 Len=1310 [TCP PDU reassembled in 18]
19	9.418594	192.168.237.21	20.189.173.3	TCP	54	49967 → 443 [ACK] Seq=854 Ack=5340 Win=262144 Len=0

Wireshark

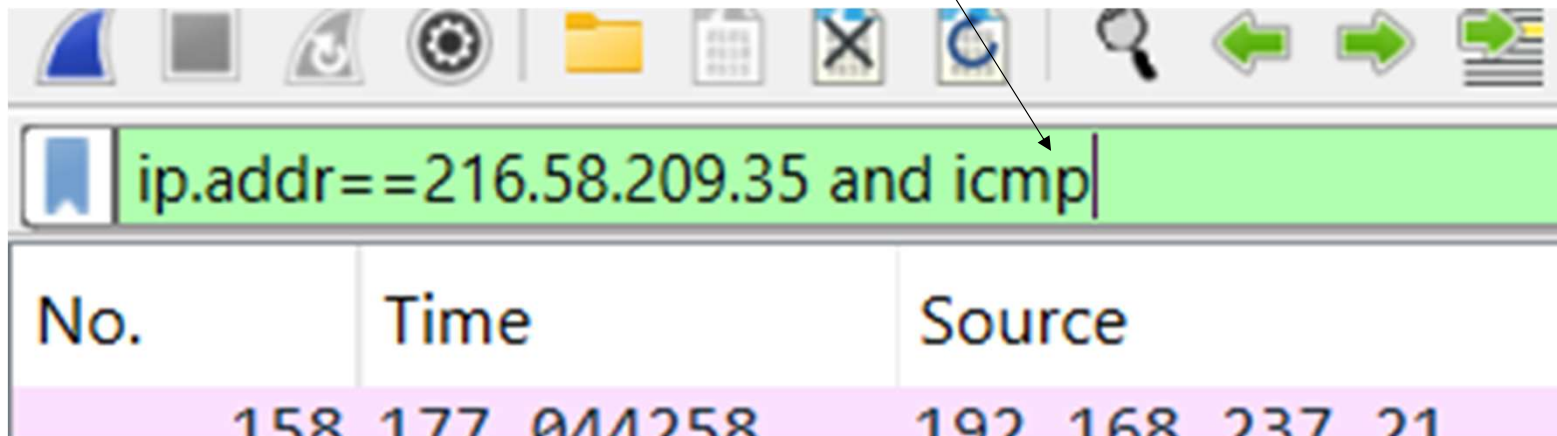
Per ripristinare quindi le sole comunicazioni con l'indirizzo 216.58.209.35 (IN QUESTO CASO OVVIAMENTE) bisogna quindi applicare di nuovo il filtro



No.	Time	Source	Destination	Protocol	Length	Info
→ 158	177.044258	192.168.237.21	216.58.209.35	ICMP	74	Echo (ping) request id=0x0001, seq=73/18688, ttl=128 (reply in 159)
← 159	177.330162	216.58.209.35	192.168.237.21	ICMP	74	Echo (ping) reply id=0x0001, seq=73/18688, ttl=114 (request in 158)
160	178.059829	192.168.237.21	216.58.209.35	ICMP	74	Echo (ping) request id=0x0001, seq=74/18944, ttl=128 (reply in 161)
161	178.121735	216.58.209.35	192.168.237.21	ICMP	74	Echo (ping) reply id=0x0001, seq=74/18944, ttl=114 (request in 160)
164	179.077246	192.168.237.21	216.58.209.35	ICMP	74	Echo (ping) request id=0x0001, seq=75/19200, ttl=128 (reply in 165)
165	179.146584	216.58.209.35	192.168.237.21	ICMP	74	Echo (ping) reply id=0x0001, seq=75/19200, ttl=114 (request in 164)
166	180.094600	192.168.237.21	216.58.209.35	ICMP	74	Echo (ping) request id=0x0001, seq=76/19456, ttl=128 (reply in 167)

Wireshark

Il filtro per essere più preciso può avere anche un AND che permette di selezionare i pacchetti scambiati con l'indirizzo IP scegliendo tra tutti solo quelli ICMP di Ping



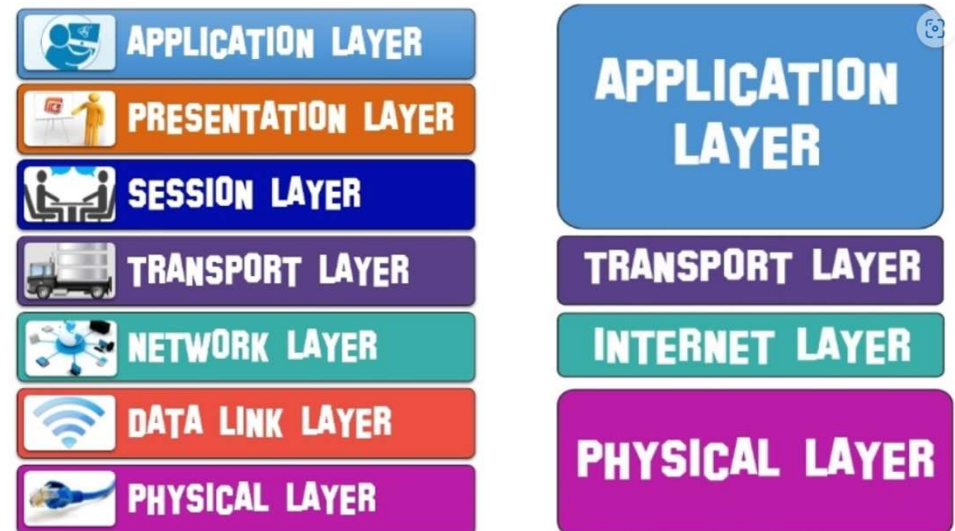
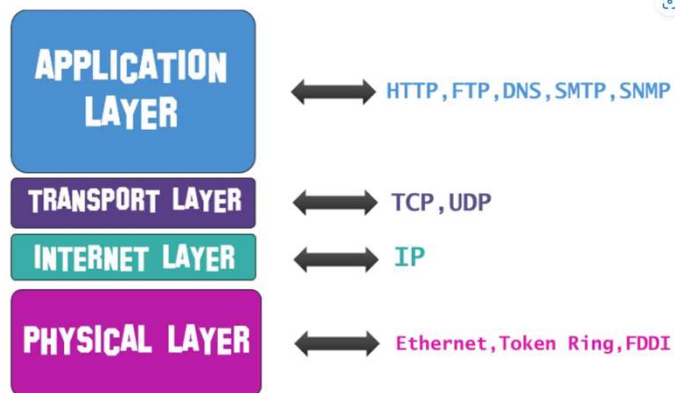
Wireshark

Nella sezione sottostante sono presenti informazioni sui dati contenuti nel pacchetto (nelle righe principali sono presenti le informazioni sul movimento, qui sotto sono presenti invece i dettagli)

```
> Frame 160: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{A3499A7C-...}
> Ethernet II, Src: ChongqingFug_20:f4:69 (4c:d5:77:20:f4:69), Dst: a6:53:56:9e:f9:72 (a6:53:56:9e:f9:72)
> Internet Protocol Version 4, Src: 192.168.237.21, Dst: 216.58.209.35
> Internet Control Message Protocol
```

Wireshark

La suddivisione è fatta secondo i livelli del TCP/IP, (l'immagine di sinistra mostra cosa usa il TCP/IP, l'immagine a destra lo confronta con l'ISO/OSI)



Wireshark

Il protocollo ICMP per navigare in rete viene aiutato dagli altri due protocolli di livello inferiore IP e Ethernet (se usata una rete cablata) oppure wireless 802.11 (se si usa una rete wireless) che aggiungono informazioni per la il recapito del pacchetto. Si può provare a vederla in maniera più strutturata usando Packet Tracer e facendo un semplice collegamento tra due pc inviando un ping e controllando la pdu facendo doppio click sulla busta che viene creata. Una nota: nel ping su internet i pacchetti di ping hanno lunghezza 74 byte, in quello locale 64 byte, è possibile verificarlo direttamente anche con Wrireshark anche **senza** avviare un web server, facendo ping all'indirizzo 127.0.0.1 . LE OPERAZIONI DA SVOLGERE IN PACKET TRACER NON VENGONO QUI ELENCAE PERCHÉ MOLTO SEMPLICI DA COMPIERE.

