ANALISI DELLE RETI

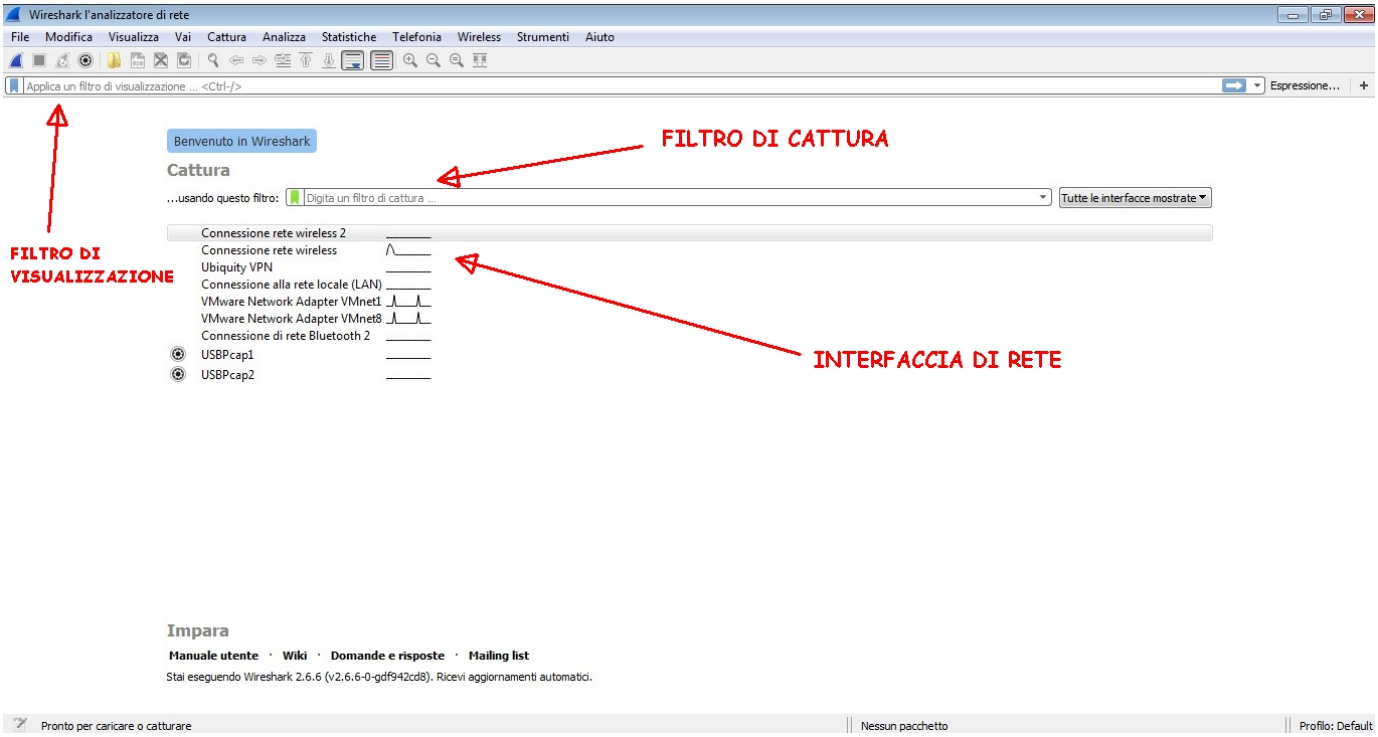
WIRESHARK

Wireshark è uno strumento software per catturare ed interpretare i dati che transitano su una rete.

In una rete transitano dati sottoforma di *frame, datagrammi, pachetti o segmenti* a seconda del loro contenuto, possiamo omunque genericamente riferirci al termine **pacchetti** di dati.
Wireshark, è un programma definito anche **analizzatore di rete**, scaricabile liberamente dal sito www.wireshark.org, e disponibile per diverse piattaforme.

Wireshark cattura i dati dall'interfaccia di rete selezionata e poi suddivide quanto catturato in pacchetti, stabilendo dove iniziano e dove finiscono. Poi interpreta e presenta i dati secondo il protocollo utilizzato. I dati catturati possono essere analizzati, subito o salvati per una successiva valutazione ed analisi ed i protocolli gestiti sono molteplici.

La G.U.I. (**G**raphical **U**ser **I**nterface) di Wireshark all'avvio si presenta in questo modo.

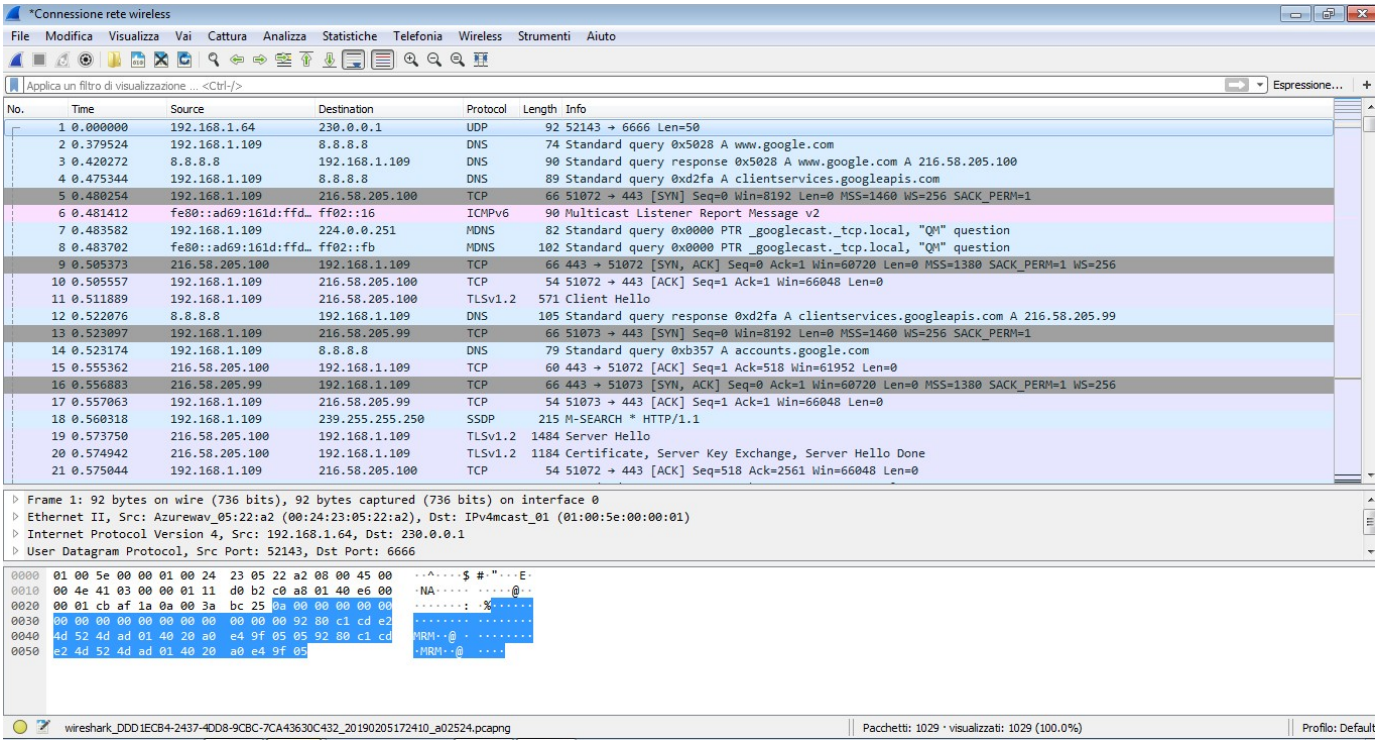


Tramite i 3 campi evidenziati possiamo scegliere:

- su quale interfaccia di rete catturare i pacchetti dati, ad esempio sulla wifi o sulla scheda ethernet
- quale filtro di cattura utiizzare, cioè possiamo scegliere se catturare ad esempio solo i pacchetti dati che utilizzano un protocollo TCP diretti alla porta 22.
- quale filtro di visualizzazione utilizzare, cioè possiamo scegliare cosa visualizzare anche in questo caso in base al protocollo utilizzato o in base ad altre caratteristiche che vedremo più avanti.

Approfondiremo poi come impostare i filtri.

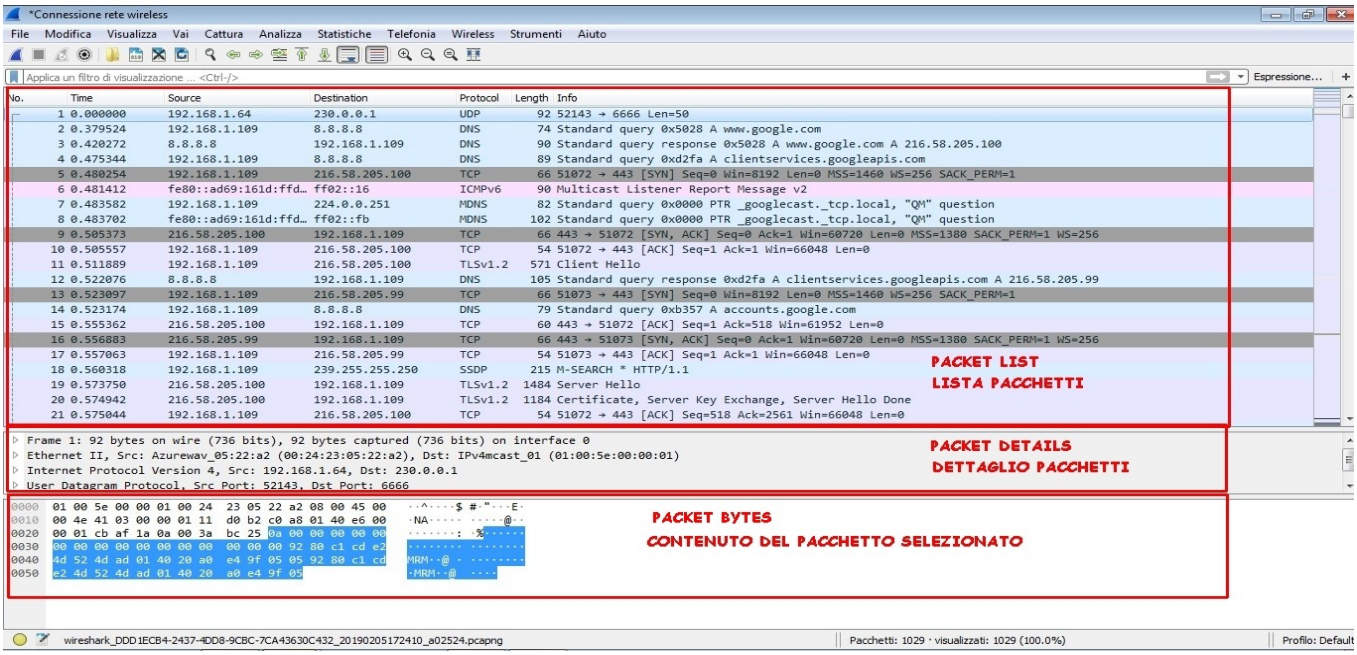
Senza impostare alcun filtro di cattura proviamo a cliccare sull'interfaccia di rete che attualmente stiamo utilizzando, ad esempio la connessione rete Wireless. In questo modo iniziera la cattura dei pacchetti che verranno visualizzati al centro dell pagina.



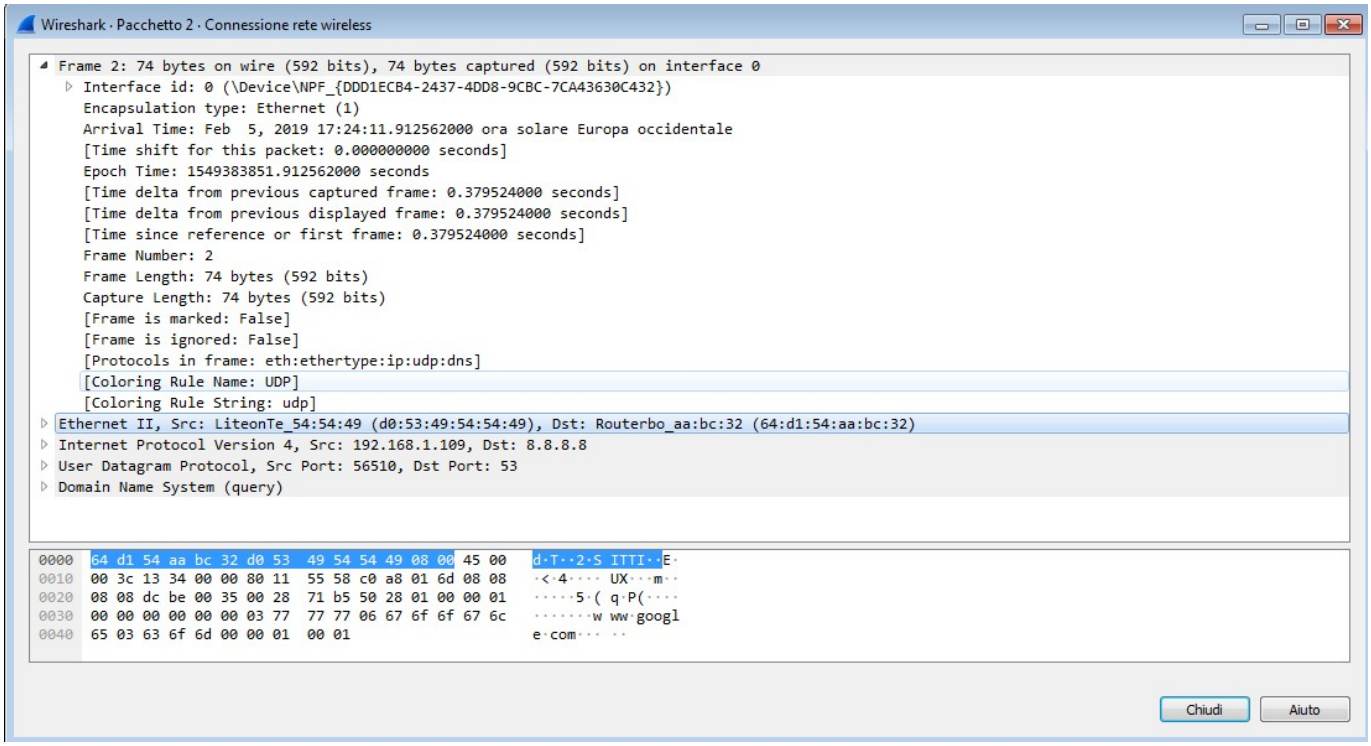
I pacchetti catturati vengono visualizzati di diverso colore a seconda del tipo di protocollo.

La visualizzazione dei pacchetti è sequenziale ed in ordine, essi vengono disposti in 5 colonne.

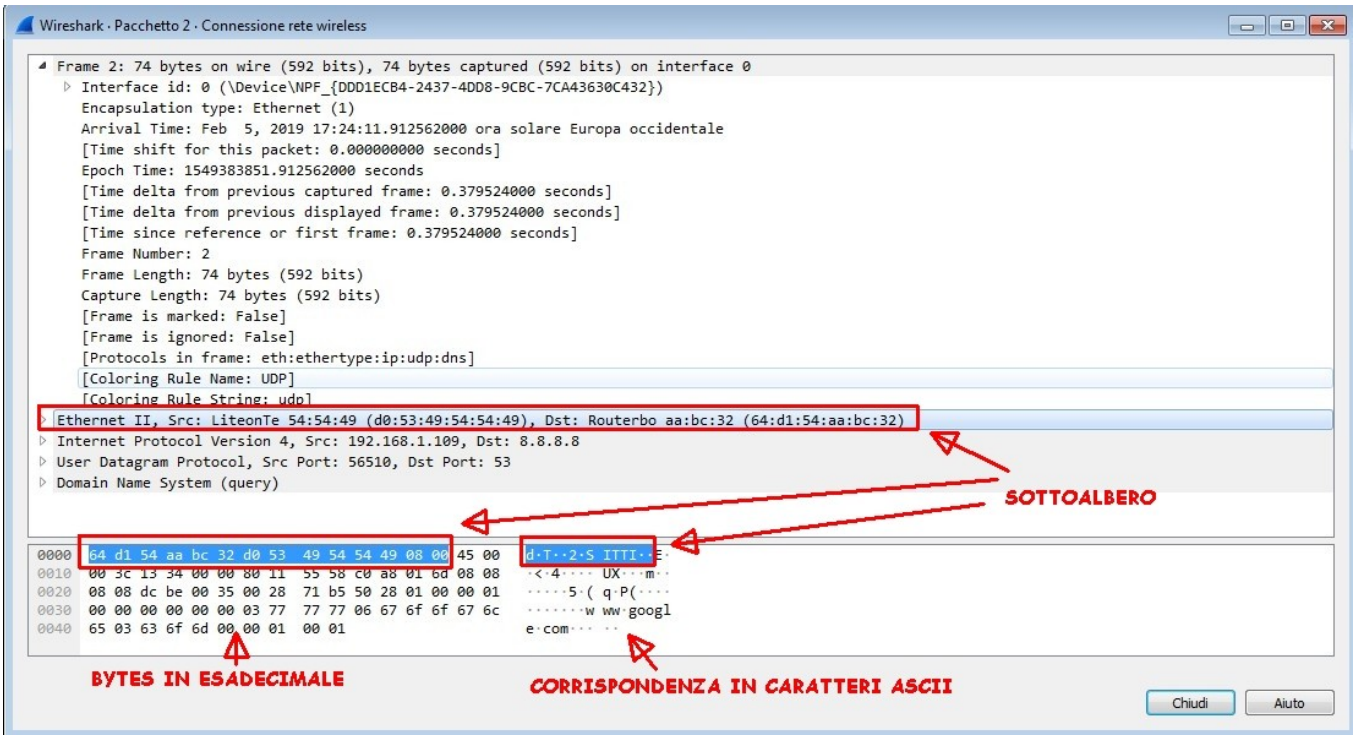
Tempo	*IP Sorgente*	*IP Destinazione*	*Protocollo*	*Lunghezza*	*Informazioni*
0.000000	*192.168.1.64*	*230.0.0.1*	*UDP*	*92*	*52143 + 6666 Len=50*
0.379524	*192.168.1.109*	*8.8.8.8.*	*DNS*	*74*	*Standard Query 0x5028 a www.google.com*



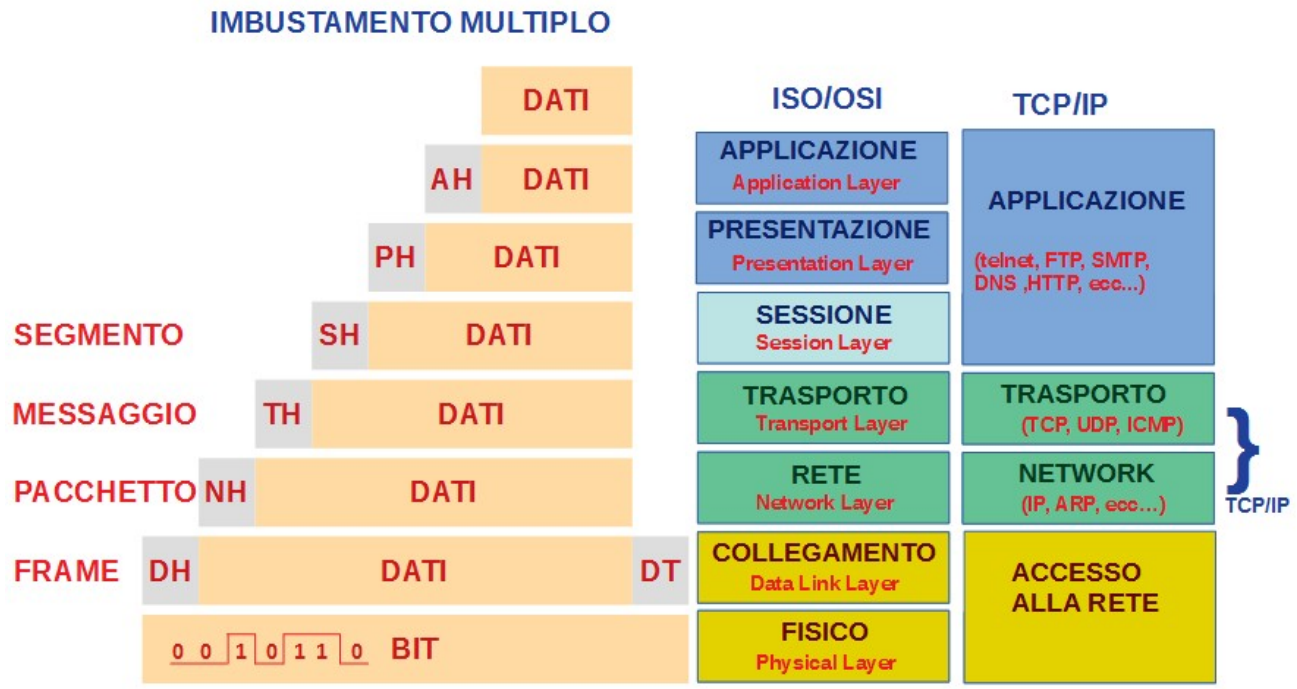
Nelle 3 sezioni sopra evidenziate troviamo una sintesi di ogni pacchetto, cioè il suo contenuto con dettagli che contengono altre informazioni visualizzabili cliccando sopra ad ogni riga, ed in basso il contenuto in byte.
Con la visuale generale è difficile visualizzare ogni informazione, per questo cliccando su uno specifico pacchetto *(ad esempio il secondo che è una richiesta DNS al server per conoscere l'indirizzo del sito www.google.com)* si apre una finestra con I dettagli del pacchetto.



Questo sistema può servire ad esempio per confrontare due pacchetti visualizzandoli contemporaneamente.
All'interno della visualizzazione dei dettagli del pacchetto, troviamo una struttura ad albero, ed ogni sezione viene definita ***"sottoalbero" (subtree)*** . Cliccando sulla freccia a sinistra il sottoalbero si espande o si chiude.
Cliccando su un sottoalbero, vengono inoltre **evidenziati i bytes interessati da quella sezione**.



Cominciamo a fare una distinzione con i termini che stiamo utilizzando, se parliamo di **pacchetto**, intendiamo l'insieme di dati in uscita dal livello 3 *(livello Rete)* del modello ISO-OSI, se invece utilizziamo il termine **frame**, intendiamo l'insieme di dati in uscita dal livello 2 *(livello Collegamento)* dello stesso modello.
Non dobbiamo dimenticare infatti che nel modello ISO-OSI viene effettuato un **imbustamento multiplo**, cioè ai pacchetti in entrata ad ogni livello vengono aggiunti dei dati (intestazioni e code dette anche header e tail).

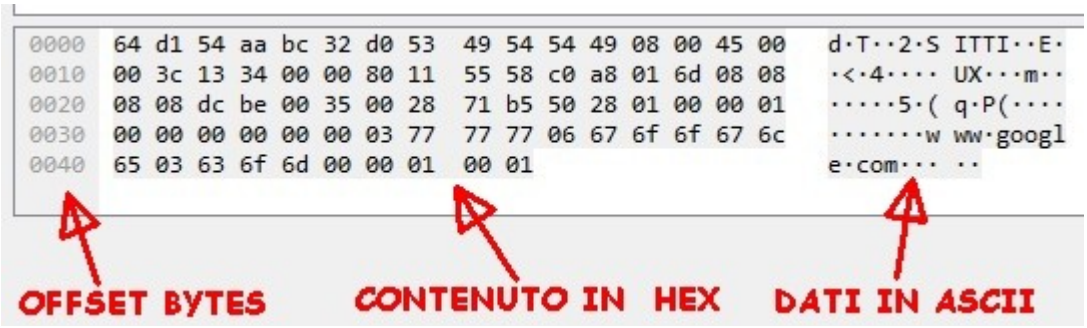


Chiusa questa parentesi torniamo ad analizzare i dettagli del pacchetto, analizziamo ad esempio il contenuto della prima riga.

```
Frame 2: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
  Interface id: 0 (\Device\NPF_{DDD1ECB4-2437-4DD8-9CBC-7CA43630C432})
  Encapsulation type: Ethernet (1)
  Arrival Time: Feb  5, 2019 17:24:11.912562000 ora solare Europa occidentale
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1549383851.912562000 seconds
  [Time delta from previous captured frame: 0.379524000 seconds]
  [Time delta from previous displayed frame: 0.379524000 seconds]
  [Time since reference or first frame: 0.379524000 seconds]
  Frame Number: 2
  Frame Length: 74 bytes (592 bits)
  Capture Length: 74 bytes (592 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:udp:dns]
  [Coloring Rule Name: UDP]
  [Coloring Rule String: udp]
Ethernet II, Src: LiteonTe_54:54:49 (d0:53:49:54:54:49), Dst: Routerbo_aa:bc:32 (64:d1:54:aa:bc:32)
Internet Protocol Version 4, Src: 192.168.1.109, Dst: 8.8.8.8
User Datagram Protocol, Src Port: 56510, Dst Port: 53
Domain Name System (query)
```

Possiamo notare oltre al numero del pacchetto ed all'interfaccia di acquisizione, la quantità di dati sulla linea e la quantità di dati catturati. In questo caso coincidono, ma mettendo dei filtri può accadere che non vengano memorizzati tutti i dati che passano sulla linea (questo se si vuole utilizzare meno spazio sul PC).

Nel pannello ***PACKET BYTES***, in basso troviamo il contenuto in bytes del pacchetto catturato.

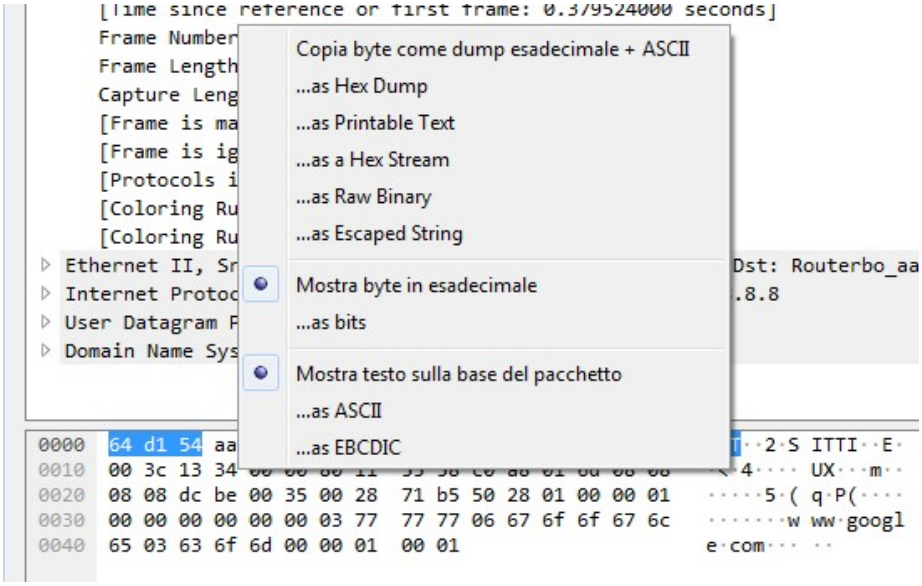


La prima colonna è un indice in esadecimale che rappresenta l'indirizzo di ogni byte. La prima riga inizia da 0000 e ci sono in tutto 16 bytes, pertanto la seconda riga iniziera in 10 (esadecimale).

Nella seconda colonna troviamo il contenuto in byte del pachetto dati.

Nella terza colonna troviamo lo stesso contenuto, visualizzato però in caratteri ASCII, se un valore esadecimale, non corrisponde ad alcun carattere ASCII, viene visualizzato un punto "."

Un clic con il pulsante destro sul pannello, apre una finestra consente di visualizzare il contenuto in bytes o in bit, e la terza colonna in ASCII (**A**merican **S**tandart **C**ode for **I**nformation **I**nterchange, *codice standard americano per lo scambio di informazioni*) o in EBCDIC (**E**xtended **B**inary **C**oded **D**ecimal **I**nterchange **C**ode, *sistema di codifica ad 8 bit utilizzato dalla IBM*).



Come detto prima, evidenziando una riga nel ***PACKET DETAILS,*** si evidenziano bytes corrispondenti nel ***PACKET BYTES.***

Caratteristica quest'ultima molto utile per analizzare eventuali problemi nella **dissezione**.

Per dissezione si intende l'operazione che Wireshark effettua *(mediante i **dissettori**, propri strumenti software)* per effettuare l'analisi sintattica di un protocollo al fine di decodificarlo e presentarlo nell'interfaccia.

I dissettori sono la parte fondamentale di Wireshark.

FILTRI

Altra parte importante di Wireshark, sono i filtri. Mediante questo strumento riusciamo a ridurre la quantità di dati catturati, concentrando l'attenzione su ciò che è di nostro interesse.
Su Wireshark ci sono due tipi di filtri, i filtri di **cattura** *(capture filters)* e quelli di **visualizzazione** *(display filters)*.
I primi agiscono sulla cattura dei pacchetti trascurando i pacchetti che non soddisfano i criteri stabiliti nel filtro.
I secondi agiscono solo sula visualizzazione, pertanto i pacchetti vengono catturati ma visualizzati secondo i criteri indicati.
Altra differenza tra i due, è la sintassi utilizzata, nel caso dei filtri di cattura si utilizza una sintassi di basso livello denominata **B**erkeley **P**acket **F**ilter BPF, mentre invece i filtri di visualizzazione utilizzano una sintassi logica più comune ed utilizzata anche da altri programmi.

FILTRI DI CATTURA

Ricordiamo che l'utilizzo dei filtri di cattura riduce di molto la quantità di pacchetti catturati e memorizzati, e ciò influisce anche su ciò che possiamo visualizzare tramite i filtri di visualizzazione *(se un pacchetto non è stato catturato non può essere ovviamente visualizzato).*
Lasciando vuoto il campo dei filtri di cattura Wireshark acquisirà tutto, opzione questa che a volte può creare problemi causati dalla quantità di dati *(vedi ad esempio l'utilizzo di Wireshark in remoto)*.
Gli elementi che costituiscono un filtro di cattura sono; **protocollo**, **direzione** e **tipo**.
Ad esempio, ***tcp dst port 22****,* indica che si vogliono catturare i pacchetti trasmessi secondo il protocollo TCP, con destinazione la porta 22.
In questo caso il protocollo è il **TCP**, la direzione è **dst** (destiinazione) ed il tipo è **porta 22.**
La direzione può essere di tipo **src** o **dst**, il primo src per catturare da uno specifico indirizzo di origne, il secondo dst per catturare un pacchetto indirizzato ad uno specifico indirizzo di destinazione.

Oltre a definire una destinazione, si possono usare dei modificatori come ***src or dst*** oppure ***src and dst***.

I tipi possibili sono: **host**, **port**, **net**, **portrange**. Se il tipo viene omesso si considera il primo host.

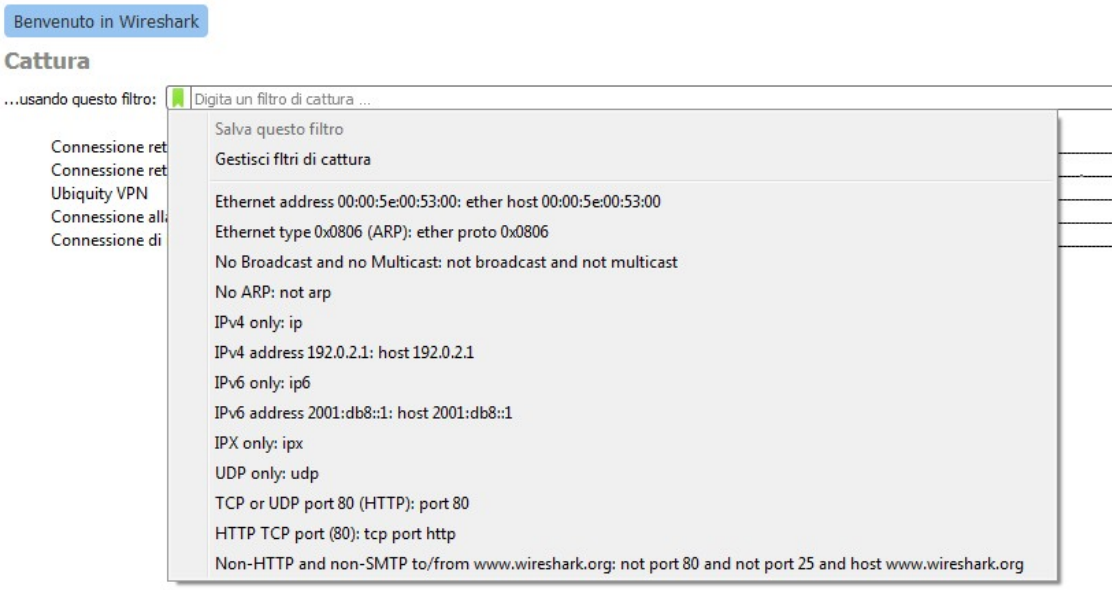
Direzione e protocollo possono essere omessi, pe cercare un tipo nell'origine o nella destinazione in tutti i protocolli. Per esempio ***dst host 192.168.0.100*** mostrerebbe solo il traffico che va a questo indirizzo IP.
Se si omette l'elemento dst verrebbe mostrato il traffico in ingresso ed in uscita dall'indirizzo IP indicato.

Il programma ci impedisce di inserire dei filtri con la sintassi sbagliata, pertanto riusciamo ad accorgerci subito di eventuali errori.

Di seguito una serie di esempi, tratti dal manuale ufficiale di wireshark https://wiki.wireshark.org/

FILTRO	SIGNIFICATO
host 172.18.5.4	*Cattura il traffico in entrata ed uscita dall'indirizzo 172.18.5.4*
net 192.168.0.0/24 **oppure** net 192.168.0.0 mask 255.255.255.0	*Cattura tutto il traffico della rete indicata, cioè tutto il traffico da e per gli indirizzi compresi tra 192.168.0.1 a 192.168.0.255*
src net 192.168.0.0/24 **oppure** src net 192.168.0.0 mask 255.255.255.0	*Cattura il traffico proveniente dalla rete indicata, cioè tutto il traffico proveniente dagli indirizzi compresi tra 192.168.0.1 a 192.168.0.255*
dst net 192.168.0.0/24 **oppure** dst net 192.168.0.0 mask 255.255.255.0	*Cattura il traffico destinato alla rete indicata, cioè tutto il traffico diretto agli indirizzi compresi tra 192.168.0.1 a 192.168.0.255*
port 53	*Cattura tutto il traffico indirizzato alla porta 53*
Host www.example.com and not (port 80 or port 25) **oppure** host www.example.com and not port 80 and not port 25	*Cattura tutto il traffico diretto all'indirizzo www.example.com escluso quello SMTP (porta 25) e HTTP (porta 80)*
port not 53 and not arp	*Cattura tutti i pacchetti indirizzati alla porta 53 escludendo quelli ARP*
(tcp[0:2] > 1500 and tcp[0:2] < 1550) or (tcp[2:2] > 1500 and tcp[2:2] < 1550) **oppure** tcp portrange 1501-1549	*Cattura i pacchetti TCP diretti a due gruppi di porte quelle superiori a 1500 e quelle inferiori a 1550*
ip	*Cattura solo il traffico IPV4*
ether	*Cattura i protocolli ethernet*
tcp	*Cattura i protocolli TCP*
ip6	*Cattura i protocolli IPV6*
arp	*Cattura i protocolli ARP*
Tcp port 80	*Cattura i protocoll TCP diretti alla porta 80*
ether broadcast	*Cattura il traffico broadcast (indirizzato a tutti i dispositivi della rete) su rete ethernet*

Cliccando comunque sulla bandierina verde accanto alla barra dei filtri di catture, Wireshark ci mostra dei filtri già pronti e ci consente di gestire i filtri modificandoli o aggiungendone altri.



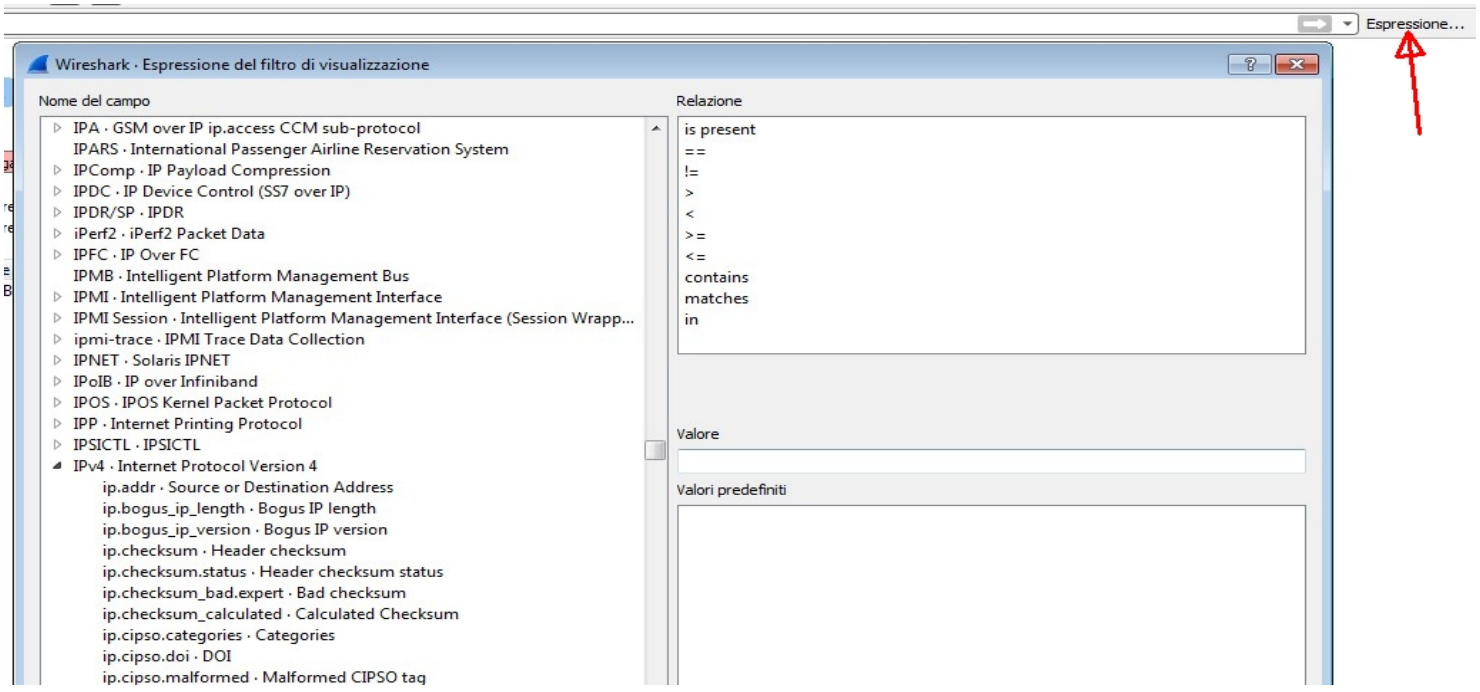
FILTRI DI VISUALIZZAZIONE

La sintassi del filtro di visualizzazione si basa su espressioni che restituiscono i valori TRUE o FALSE.

Queste espressioni utilizzano operatore di confronti che si possono combinare con operatori BOOLEANI, per esprimere criteri complessivi ed ottenere i risultati desiderati.

Operatore	In linguaggio C	Descrizione
eq	==	*Uguale a*
ne	!=	*Diverso da*
gt	>	*Maggiore di*
lt	<	*Minore di*
ge	>=	*Maggiore o uguale di*
le	<=	*Minore o uguale di*
Contains		*Verifica se il campo contiene un dato valore*
Matches		*Esamina un campo sulla base di un'espressione regolare in stile PERL*
and	&&	*Restituisce TRUE se entrambi le espressioni sono viere*
or	||	*Restituisce TRUE se una o entrambe le espressioni è vera*
xor	^^	*Restituisce TRUE solo se una delle due estressioni è vera*
not	!	*Nega l'espressione seguente*
	[]	*Accede ad un sottostringa della stringa, es. dns.resp.name[1..4] accede i primi 4 caratteri del nome di risposta DNS*
	()	*Raggruppa le espressioni*

Wireshark ci offre un grande aiuto per comporre i filtri di visualizzazione, cliccando sulla scritta "espressione" in alto a destra vicino la riga di comando del filtro di visualizzazione, si apre una finestra con una miriade di filtri possibilie e personalizzabili. Ogni tipo di filtro ha una struttura ad albre e cliccando sull frecci si aprono altre sottotipologie.

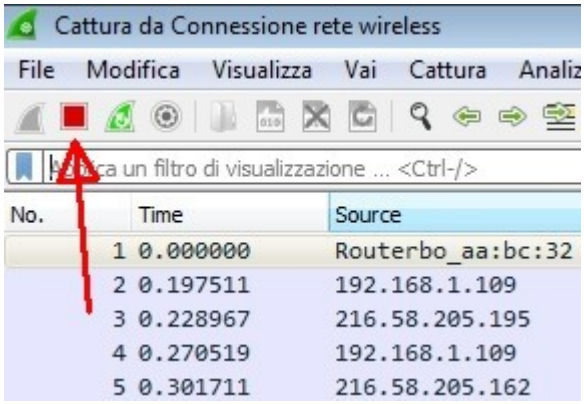


CATTURA DI PACCHETTI DALLA RETE LOCALE

Proviamo ora ad effettuare una semplice cattura di pacchetti dalla rete locale, per poi analizzare quanto acquisito. Dopo aver lanciato Wireshark ed aver selezionato l'interfaccia di rete attiva, l'acquisizione ha inizio. Per avere più pacchetti da analizzare, digitiamo nel browser l'indirizzo di un sito ad esempio www.ebay.it , ed avviamo un breve navigazione tra le sue pagine.

Non essendoci filtri attivi, con questa semplice operazione verranno catturati diversi pacchetti dati.

Conviene pertanto bloccare la cattura dei pacchetti con l'apposito pulsante di STOP.

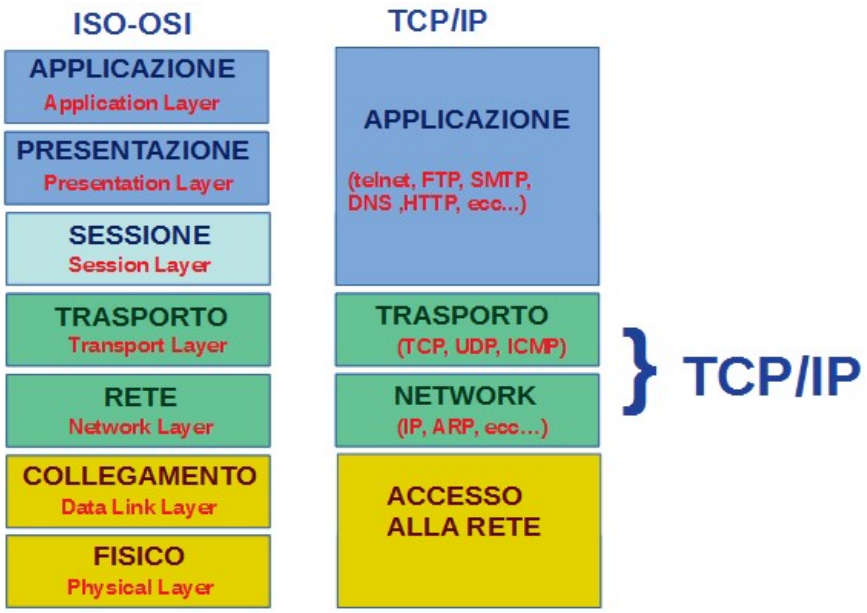


Una delle caratteristiche più interessanti di Wireshark, sta proprio nel funzionamento dei dissettori, e cioè quegli strumenti software in grado di decodificare un pacchetto dati e presentarlo nell'interfaccia.

Nel fare questa operazione infatti, i pacchetti verranno mostrati secondo l'ordine del modello TCP/IP.

Internet infatti si basa su questo protocollo che è differente dallo standard ISO/OSI per il numero di livelli e per la distribuzione dei vari protocolli in ogni strato.

In questa immagine possiamo distinguere i 4 livelli del modello TCP/IP, con qualche esempio di protocollo appartenente ad ogni livello.



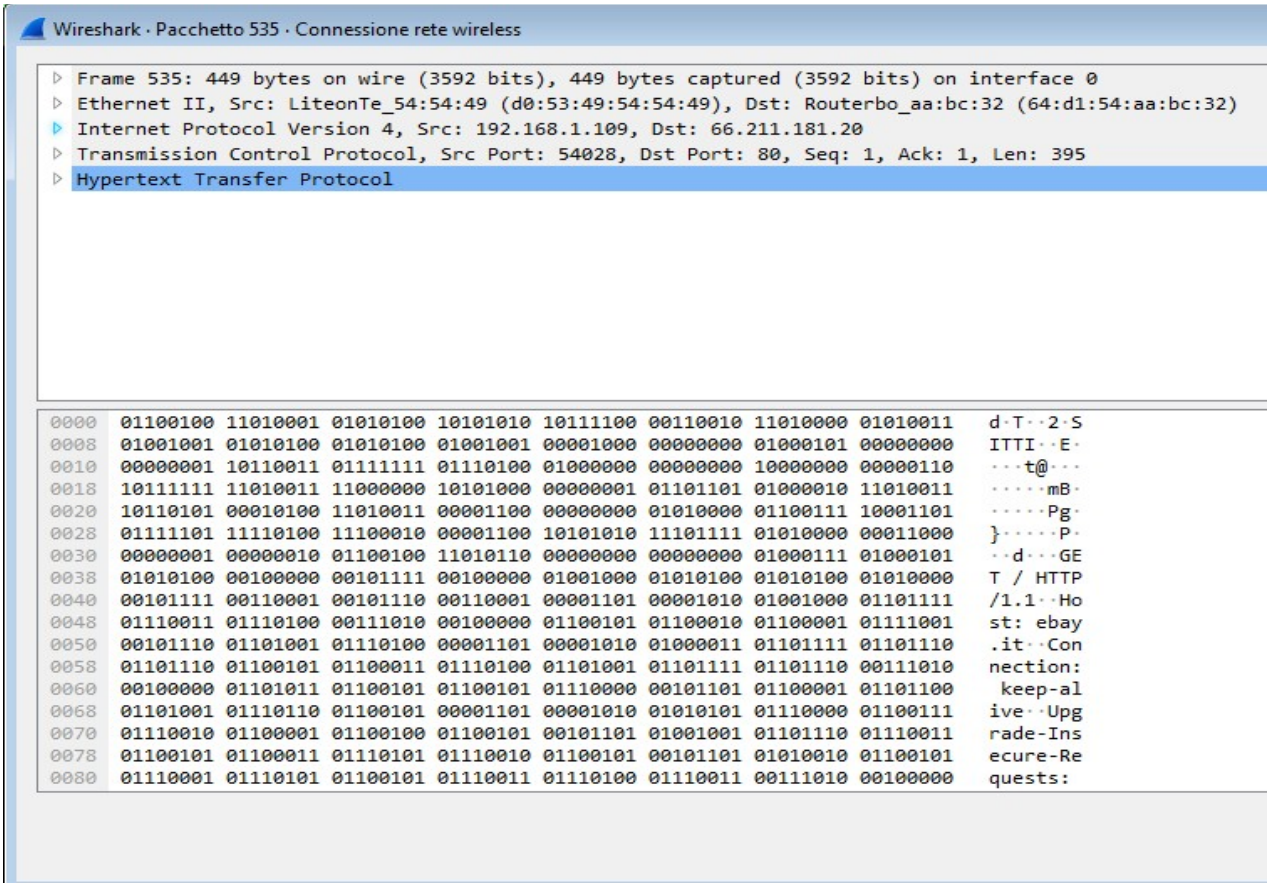
Ora per selezionare qualche pacchetto proviamo a scrivere nel campo dei filtri di visualizzazione "http" , in questo modo verranno visualizzati solo i pacchetti che utilizzano questo protocollo, con un po' di fortuna potremmo aprire uno dei pacchetti indirizzati o provenienti dal sito selezionato prima.

Ma come dicevamo prima sulla finestra del dettaglio del pacchetto (cliccando su un pacchetto si apre questra finestra) abbiamo la visualizzazione dei dati secondo il modello TCP/IP della figura precedente.

Perciò oltre al primo livello rappresentato dal frame completo, avremo in ordine gli altri 4 livelli.

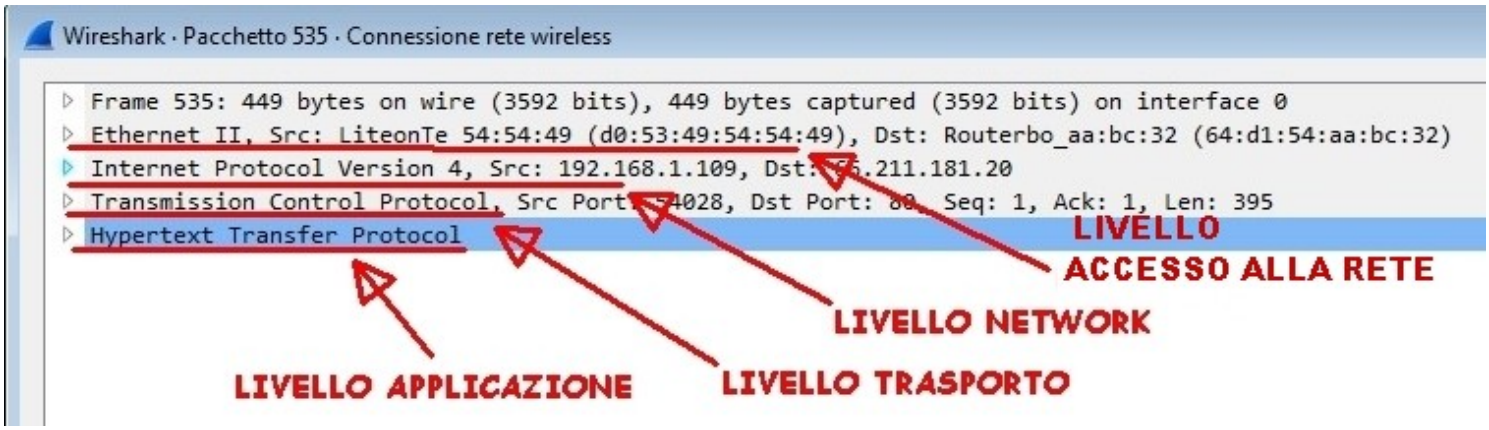
Nella figura seguente vediamo il dettaglio di un pacchetto http, si possono notare le seguenti righe:

- **Frame 535** (il 535 esimo pacchetto catturato) come si legge nella riga, esso è composto da 449 bytes.
- **Ethernet II**, dove troviamo l'indirizzo MAC del sorgente (Src) e del destintario (Dst).
- **Internet Protocol IP**, dove invece troviamo gli indirizzi IP del sorgente (Src) e del destintario (Dst).
- **Trasmission Control Protocol TCP**, dove troviamo anche le porte del PC utilizzate.
- **HyperText Transfer Protocol HTTP**, che invece contiene i dati a livello applicazione.

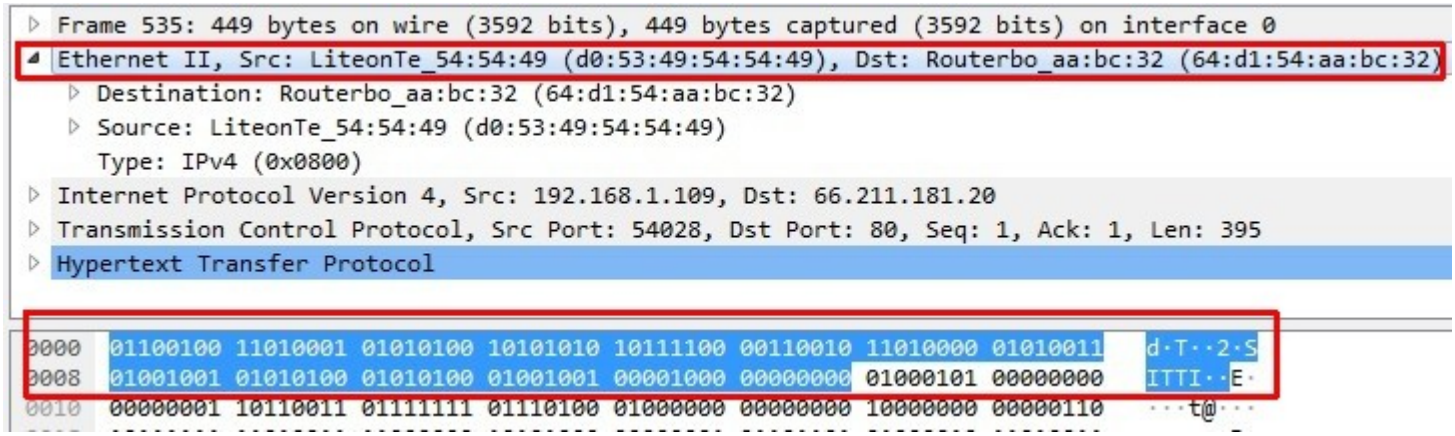


In pratica partendo dall'alto dopo il frame abbiamo:

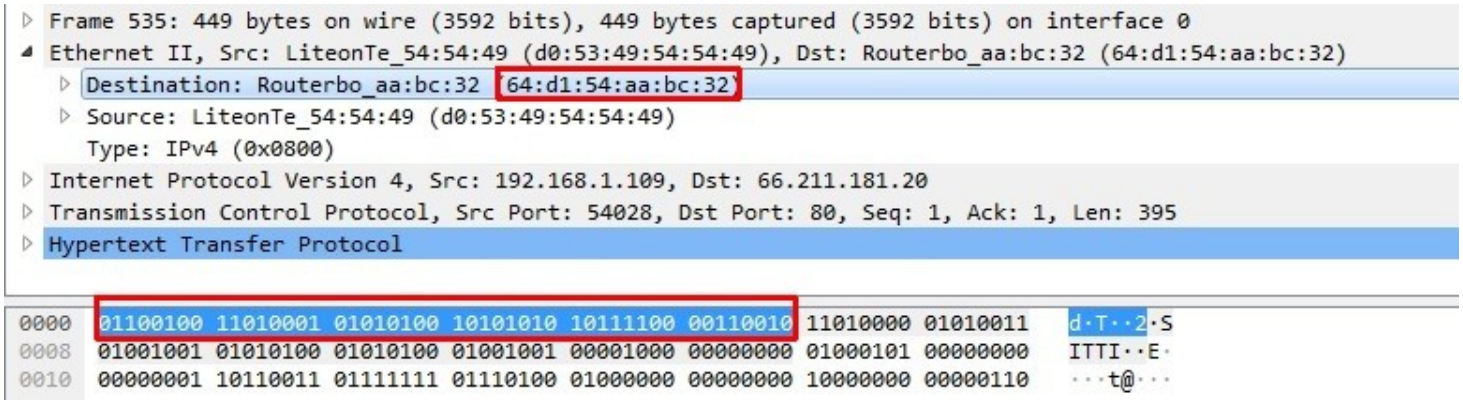
Livello Accesso alla rete	**-**	**Ethernet II,**
Livello Network	**-**	**Internet Protocol IP**
Livello Trasporto	**-**	**Trasmission Control Protocol TCP**
Livello Applicazione	**-**	**HyperText Transfer Protocol HTTP**

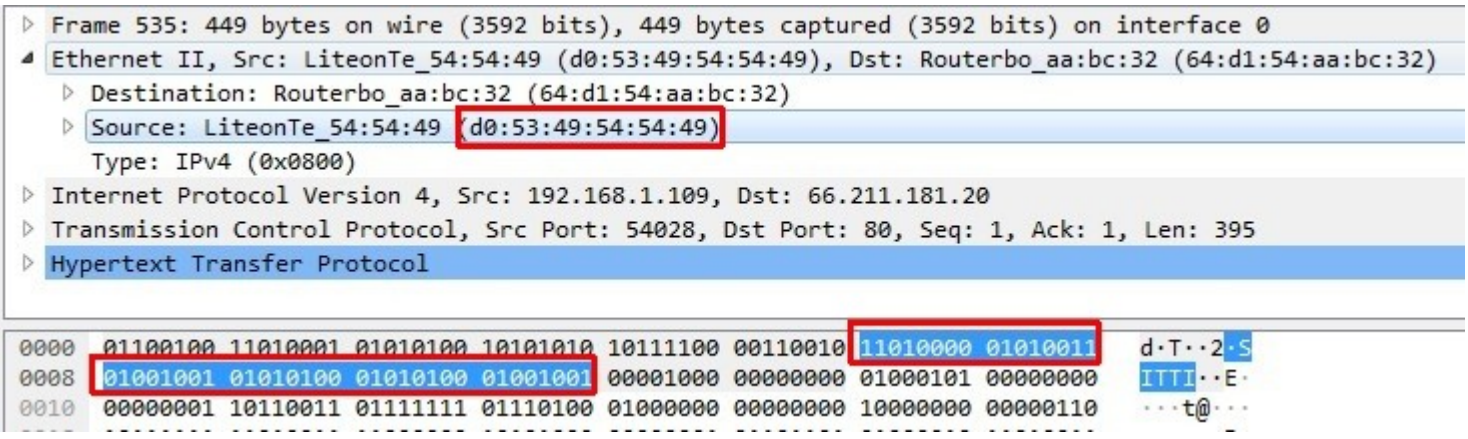


Possiamo ovviamente scendere maggiormente nel dettaglio di ogni livello, andando ad aprire la struttura ad albero cliccando sopra alla riga scelta. In questo modo potremo vedere il contenuto ad esempio del livello Ethernet che verrà anche evidenziato nel pannello dei bytes in basso, dove possiamo vedere il contenuto in binario o in esadecimale.

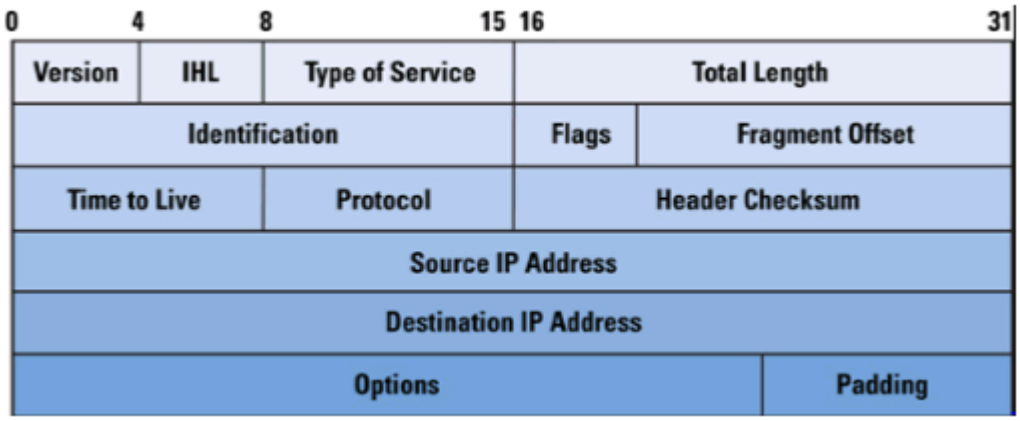


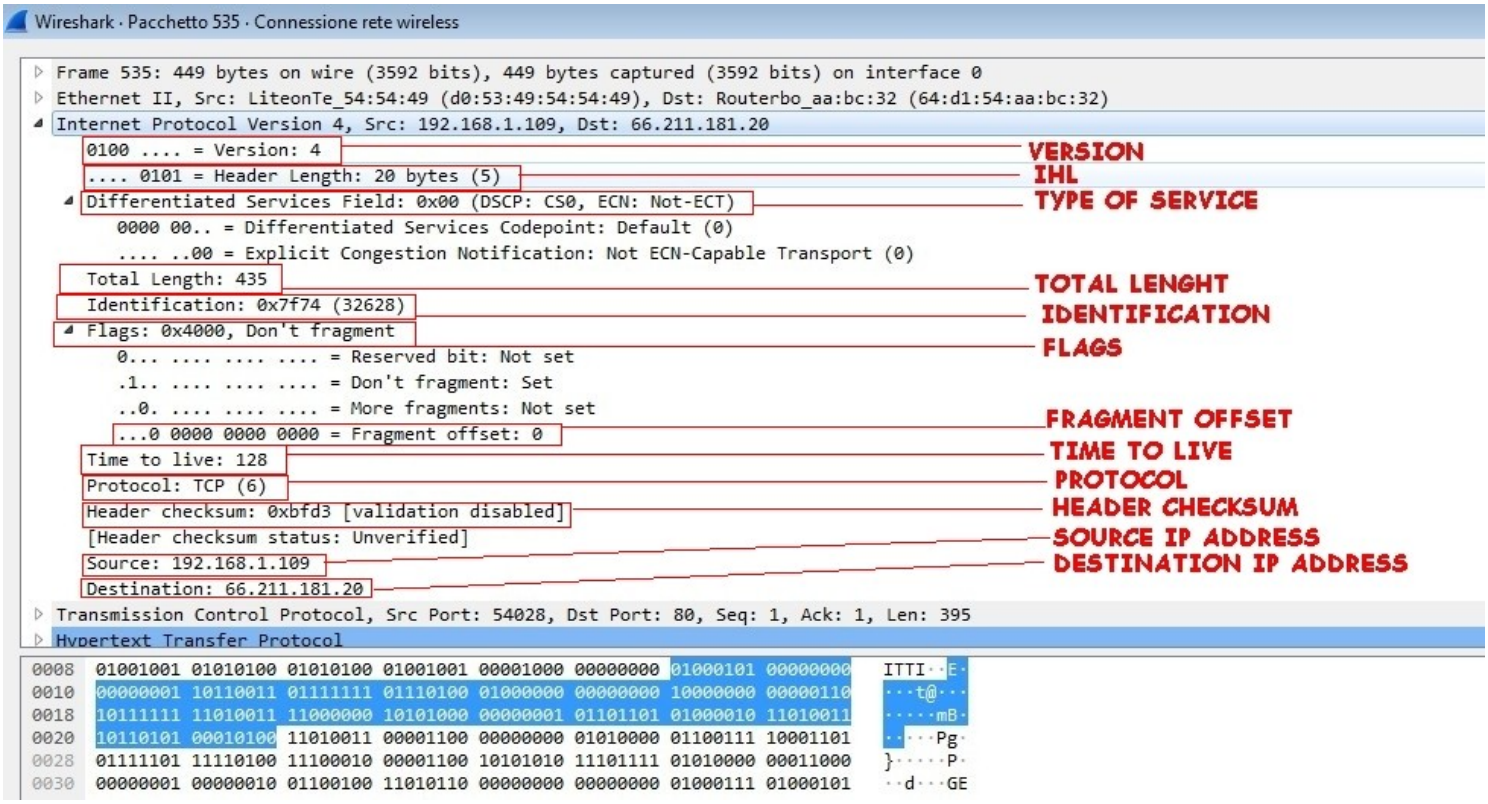
Cliccando ad esempio sulla destinazione, o sulla sorgente, nel pannello dei bytes si evidenzieranno i dati, che corrispondono all'indirizzo in eadecimale scritto sopra.





Allo stesso modo possiamo analizzare il datagramma IP, dove troviamo la struttura nota di questo pacchetto dati, composta dai seguenti campi:





Allo stesso modo possiamo aprire la parte relativa al TCP ed all'HTTP per analizzare tutti i singoli segmenti di dati che costituiscono il frame complessivo.

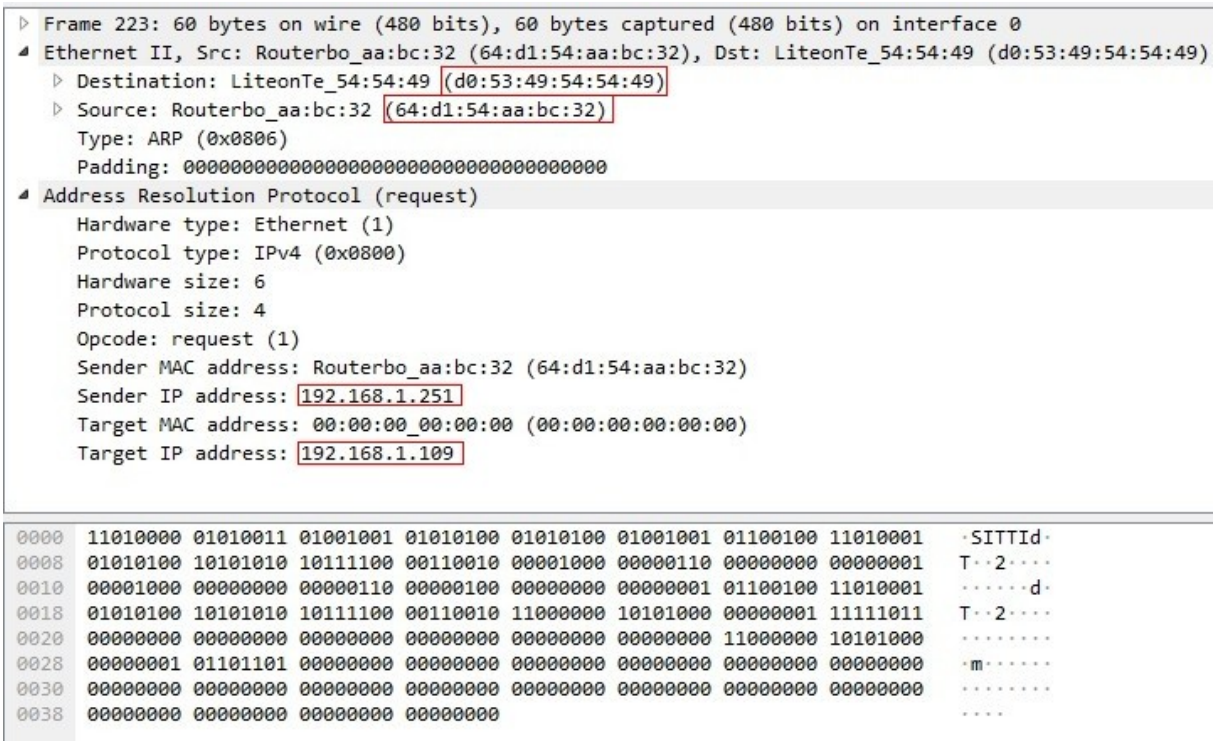
Nel Pannello dei bytes in basso, troviamo sempre evidenziati i dati corripondenti a ciò che evidenziamo in alto. Possiamo notare, cliccando su ogni strato, il modo di "**imbustare**" i dati classico del modello ISO/OSI e del TCP/IP.

Un altro pacchetto dati semplice da analizzare, è quello del protocollo ARP, digitiamo sul filtro di visualizzazione **“arp”**, e clicchiamo su uno dei pacchetti.

In questo caso abbiamo solo 2 livelli, Ethernet (Livello di accesso alla rete) e ARP (Livello Network)

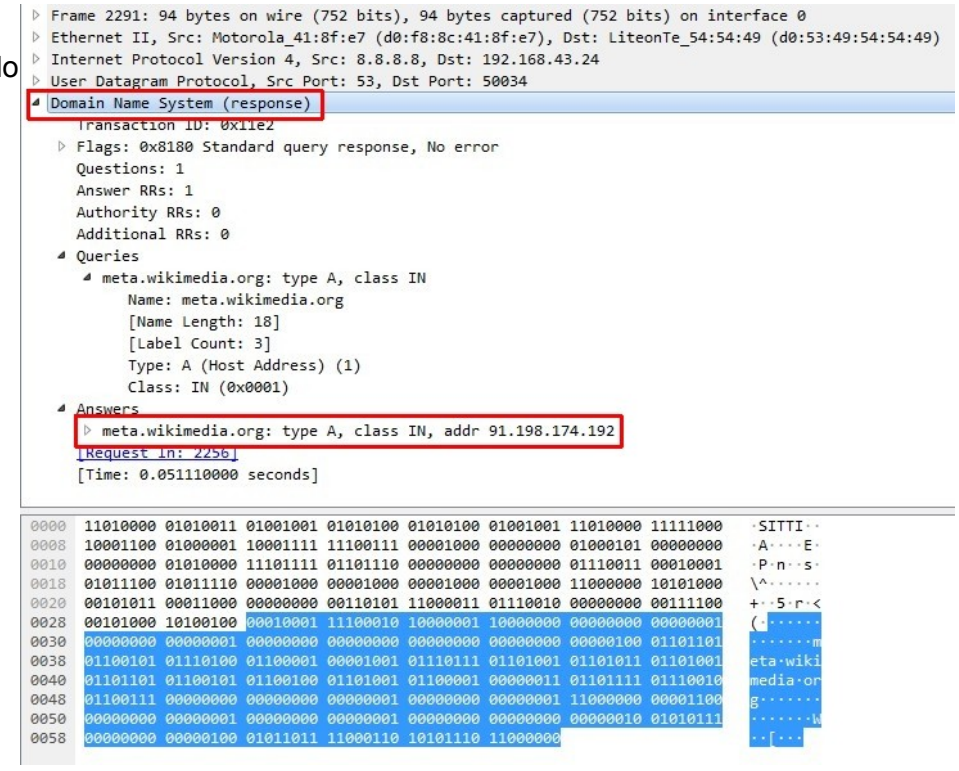
ARP si occupa di associare ad ogni indirizzo MAC dei dispotivi, il corrispondente indirizzo IP.

Gli indirizzi sono evidenziati nella seguente immagine.

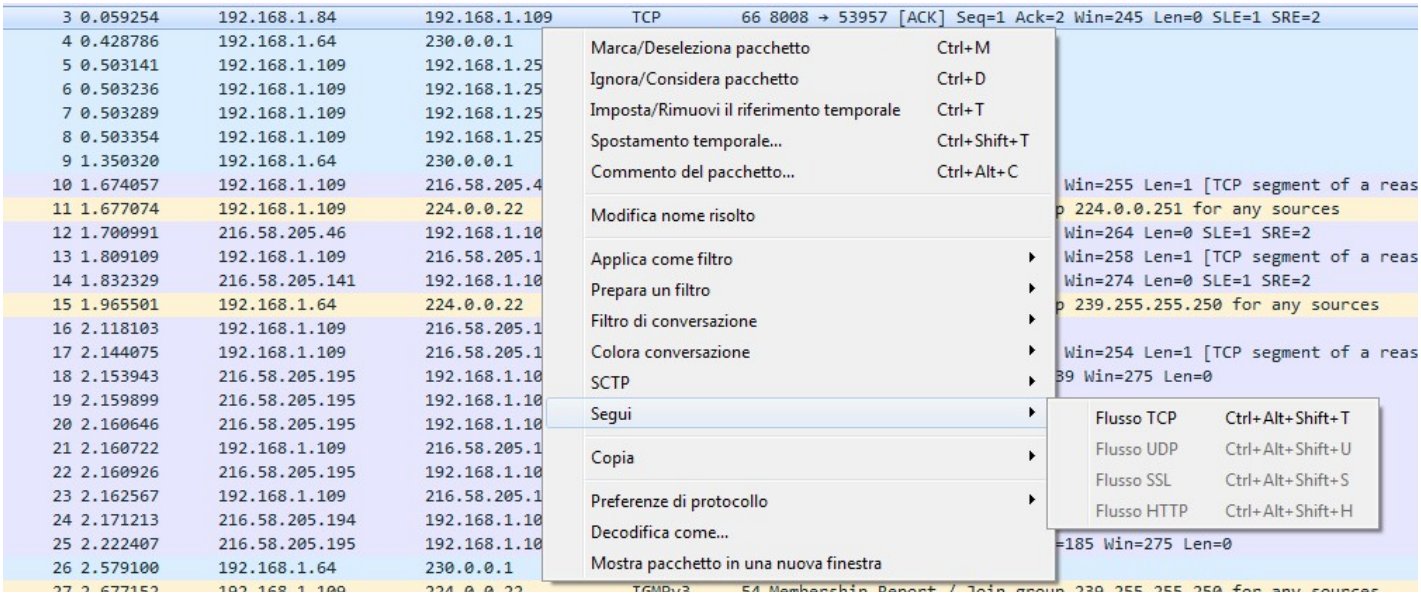


Un’altro pacchetto facilmente analizzabile è quello relativo al protocollo DNS (Domain Name System).

In questo caso possiamo notare nel contenuto del pacchetto DNS a livello applicazione, il nome del sito di wikipedia e il relativo indirizzo 91.168.174.192



Cliccando inoltre su un qualsiasi pacchetto dati con il tasto destro del mouse si può seguire il flusso dati tra client e server



Seguendo un flusso dati automaticamente viene creato il relativo filtro di visualizzazione.

Anche senza seguire il flusso dati con il pulsante destro si può cliccare sopra un pacchetto dati ed impostare un filtro di visualizzazione.

Quanto detto finora è solo una breve introduzione all'utilizzo di un software molto potente per l'anaisi del flusso dati di una rete.
Ovviamente l'utilizzo di Wireshark non è utile solo per affrontare l'analisi dei pacchetti di rete da un punto di vista didattico, ma anche e soprattutto per studiare le tecniche di attacco e di difesa di una rete al fine di migliorarne la sicurezza.

Per quest'importante utilizzo dobbiamo però mettere in campo altre conoscenze ed altri strumenti che ci consentiranno di simulare ogni situazione e verificare tutte le criticità che si possono avere.

Per questo possiamo anticipare senza entrare nei dettagli, la necessità di installare un potente sistema operativo indispensabile per testare e mettere a dura prova la sicurezza di una rete e cioè Kali Linux.
Questo sistema operativo può essere installato su una macchina ma come primo approccio potrebbe essere utile installarlo in una macchina vituale (VM) con Virtual Box.
In questo modo avremo il sistema operativo Kali Linux dentro al nostro PC, che potrà essere avviato insieme al sistema operativo già avviato, senza alcun rischio.
Di seguito le istruzioni per l'installazione di Virtual Box e di Kali Linux.

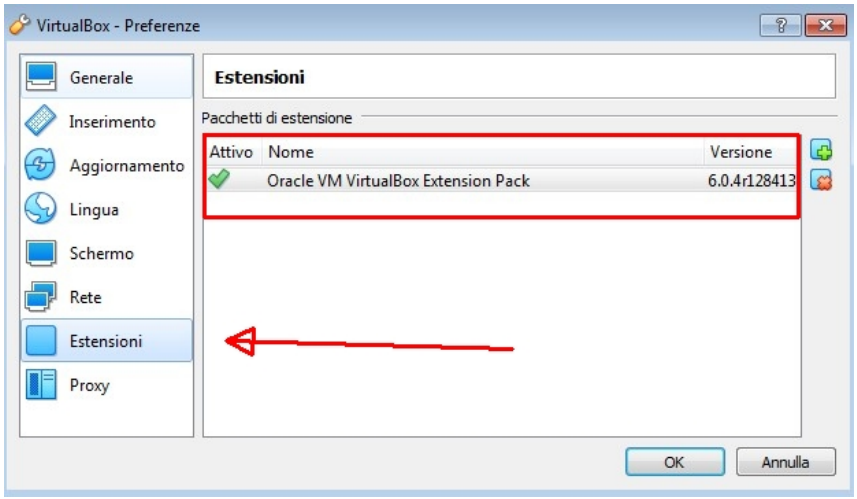
VIRTUAL BOX

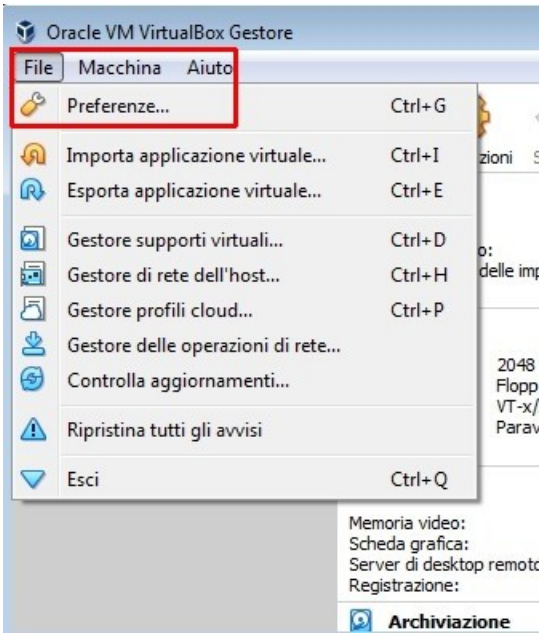
La prima operazione da fare è quella di installare l'ultima versione di Virtual Box, scaricandola per il proprio sistema operativo dal link https://www.virtualbox.org/wiki/Downloads

L'installazione è praticamente automatica cliccando sul file scaricato, basta confermare tutte le opzioni richieste.

Dopo aver installato Virtual Box occorre scaricare ed installare (sempre cliccando sul file scaricato) le **Virtual Box Extension Pack**, sempre allo stesso link.

Dopo l'installazione lanciando Virtual Box e cliccando su ***File-Preferenze***, dovremmo trovare le Etension Pack installate.

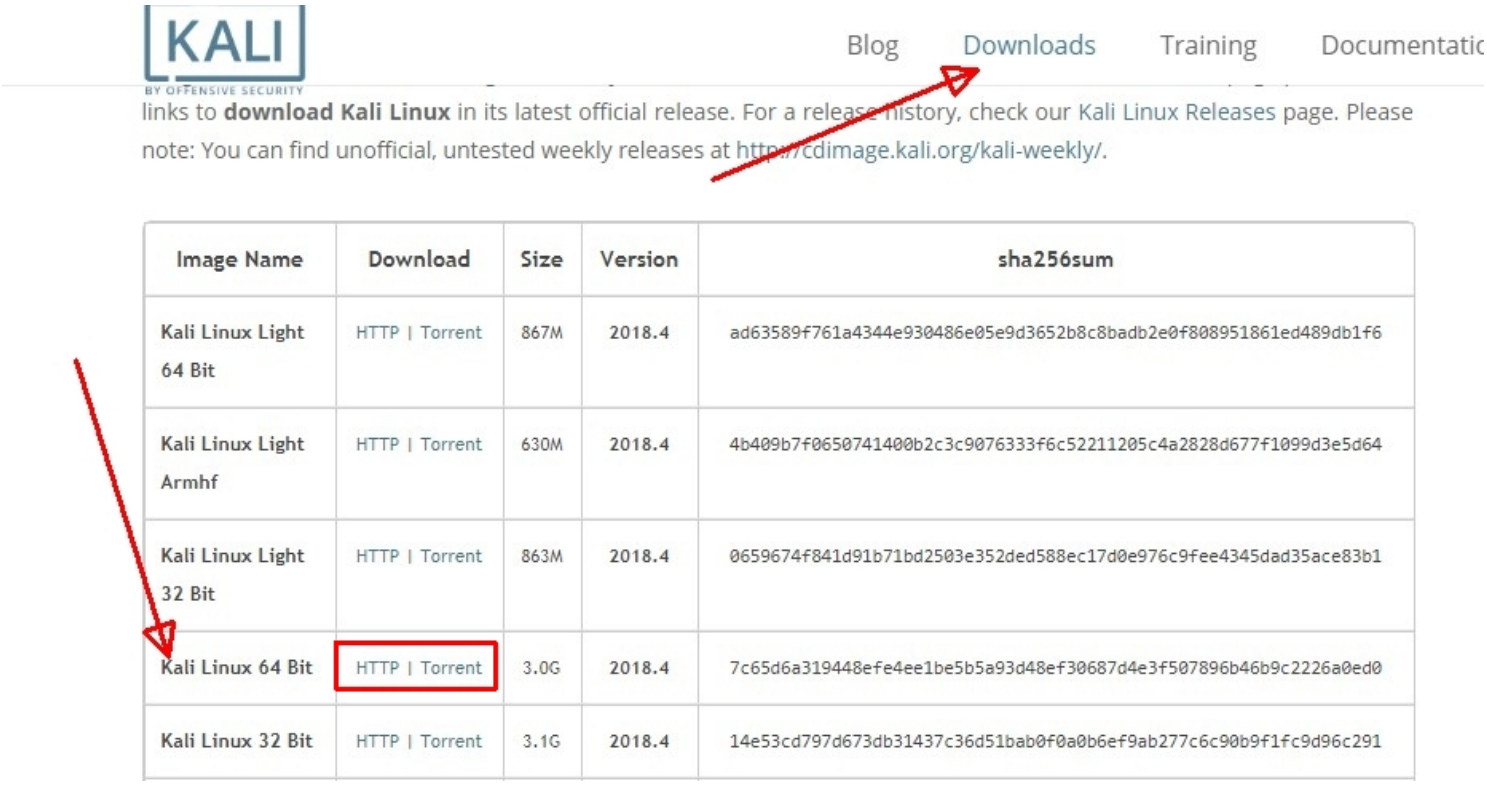




KALI LINUX

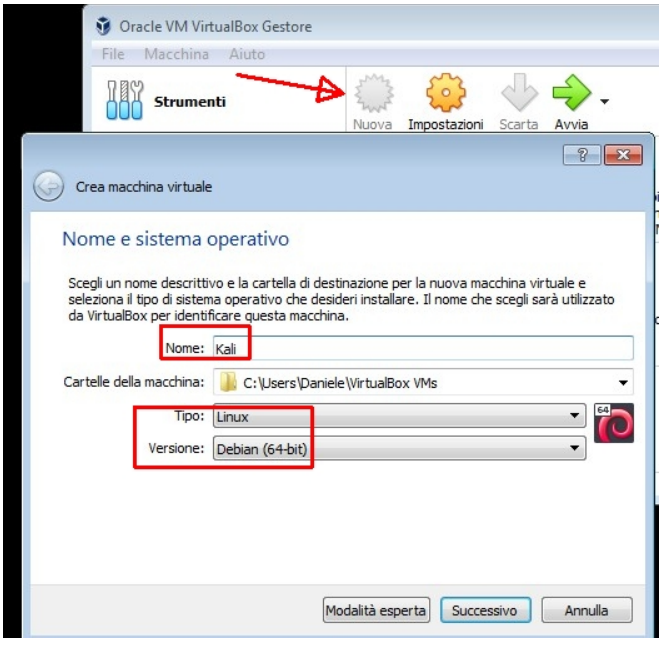
Seconda cosa da fare è scaricare la ISO del sistema operativo KALI, reperibile al link:
https://www.kali.org/downloads/

.Nella sezione downloads, troviamo le varie versioni di KALI Linux, consiglio di installare l'ultima versione completa (non light) a 64 bit.
Il file ISO (Immagine del Sistema Operativo) si può scaricare direttamente cliccando su HTTP, o indirettamente cliccando su Torrent e scaricando così il file torrent da dare in pasto al programma bitTorrent o ad altri programmi del genere che gestiscono questo tipo di condivisione di file.



MACCHINA VIRTUALE KALI LINUX

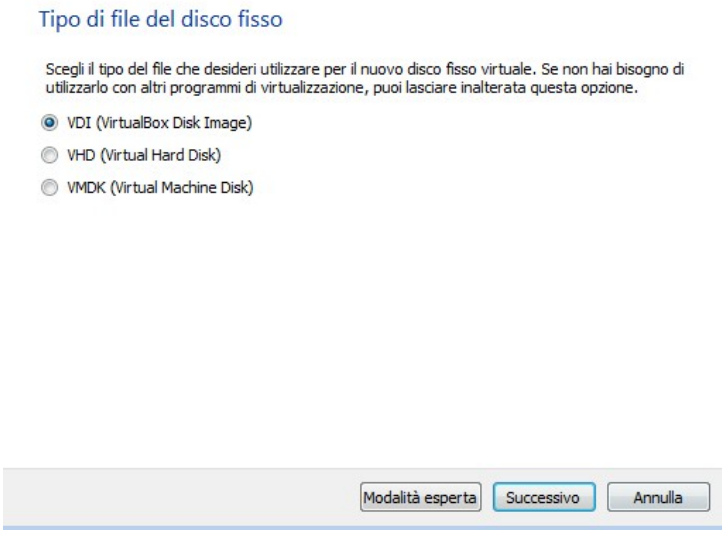
Ora ocorre creare la macchina virtual di KALI Linux nell'ambiente di Virtual Box.
Lanciando Virtual Box clicchiamo su nuova ed alla finestra successiva impostiamo il tipo di sitema operativo "Linux" e la versione "Debian 64bit".
Possiamo dare un nome qualsiasi alla macchina virtuale scegliamo magari un nome che ci indica il S.O.
e cioè Kali.
Poi proseguiamo con il tasto "successivo".



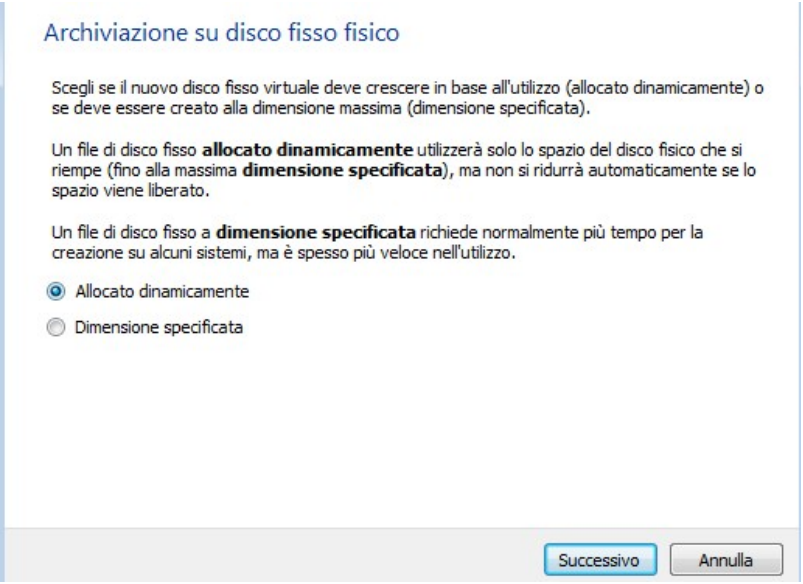
Successivamente verrà richiesta la RAM da utilizzare, si consigliano almeno 2GB.

Verrà chiesto di creare un disco fisso.

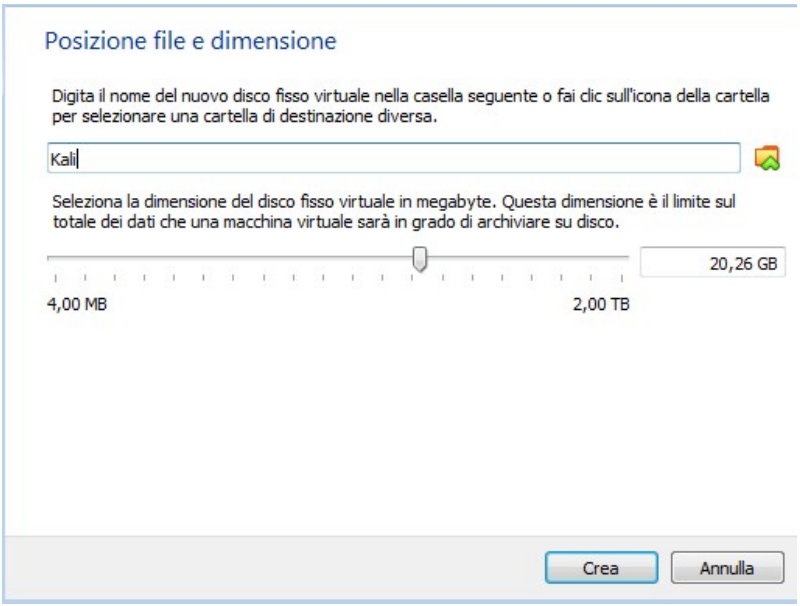
Verrà chiesta la tipologia del disco fisso. Lasciare l'opzione di default.



Verrà richiesto se utilizzare l'allocazione dinamicadel disco (scelta consigliata).
Con l'allocazione dinamica le dimensioni sul disco fisico del PC della macchina virtuale, crescono fino alla dimensione massima che scegieremo poi.
In caso contrario verrebbe utilizzato uno spazio fisso sul disco rigido delle dimensioni indicate.

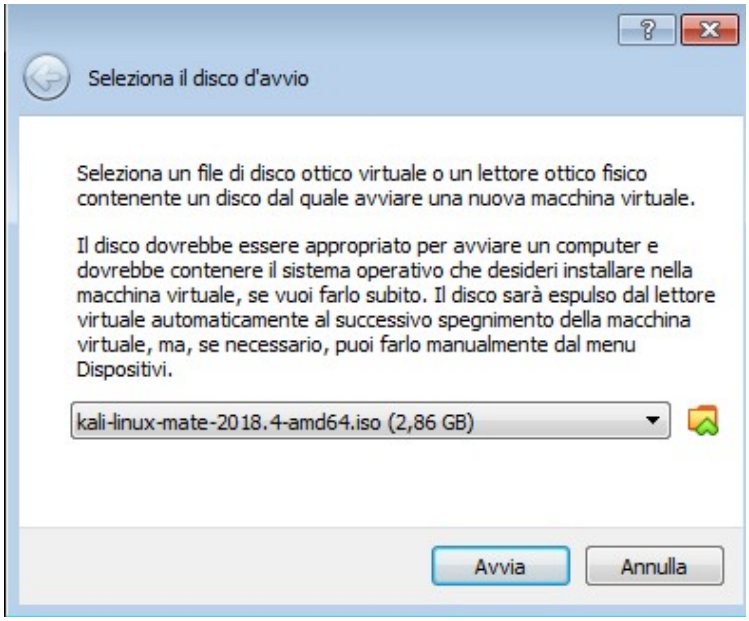


Infine scegliamo la dimensione dei disco della macchina virtuale, si consigliano almeno 20GB.

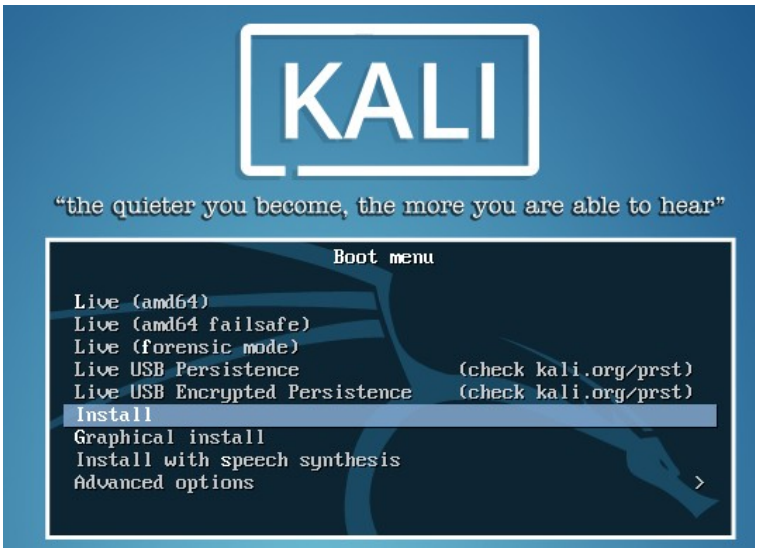


Al termine cliccando sul tasto "Crea" la nostra macchina virtuale è pronta per l'installazione. E' come se avessimo assemblato un PC con tutte le caratteristiche richieste, RAM, disco fisso. La parte successiva sarà l'installazione del sistema operativo dall'imagine precedentemente scaricata.

Cliccando sul tasto AVVIA di virtual box, ci verrò chiesto inizialmente dove trovare l'immagine ISO precedentemente scaricata. Cliccando sul pulsante con la cartella a destra andare a selezionare il file ISO scaricato.



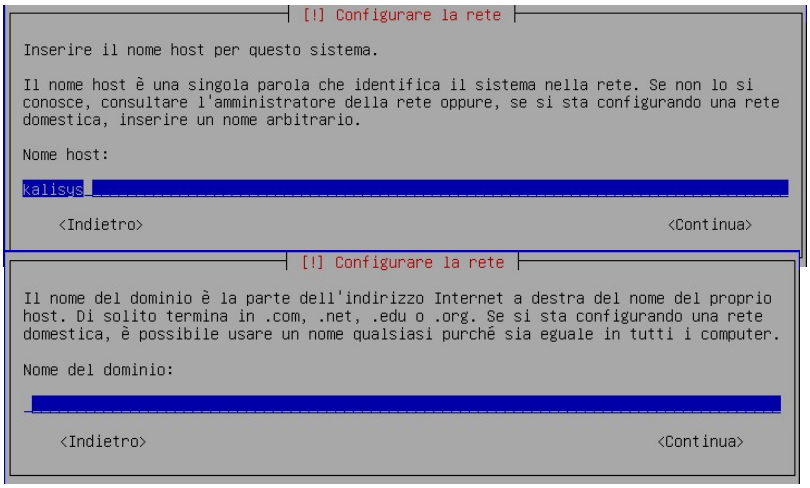
Successivamente si avvierà il classico menù di ogni sistema Linux che prevede la possibilità di una Live, cioè l'avvio del sistema operativo senza installazione. Noi invece selezioneremo subito l'installazione.



Le prime richieste fatte durante l'installazione riguardano la lingua del sistema e della tastiera, impostare ovviamente **l'Italiano.**

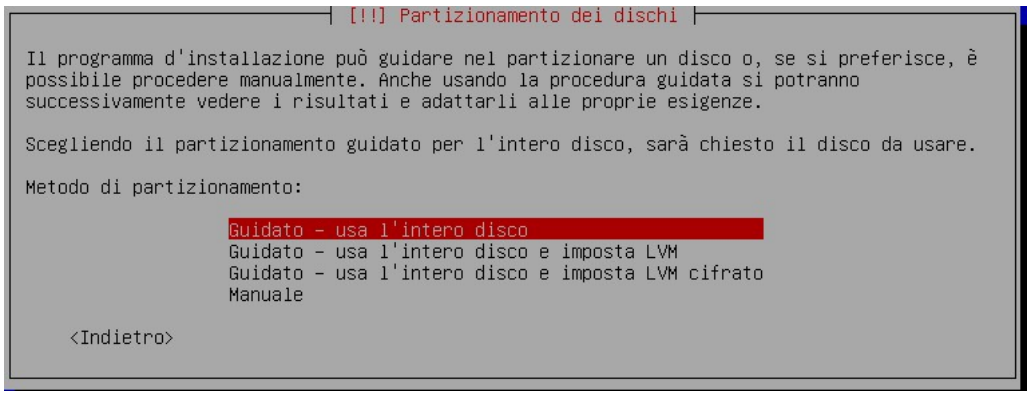
Verrà poi chiesto il nome del sistema, possiamo scegliere il nome che vogliamo, in questo caso **kalisys**.

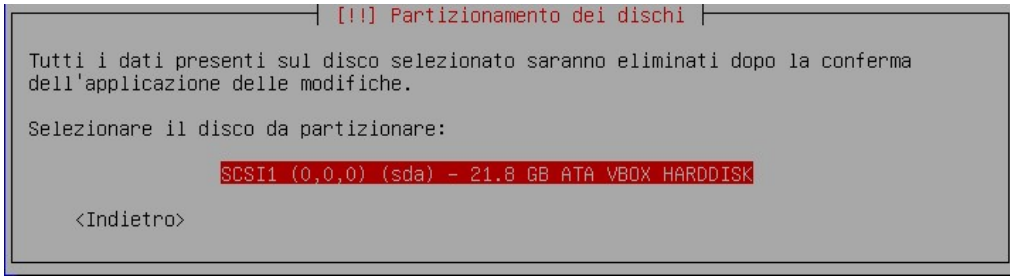
Possiamo lasciar vuoto il nome del dominio.

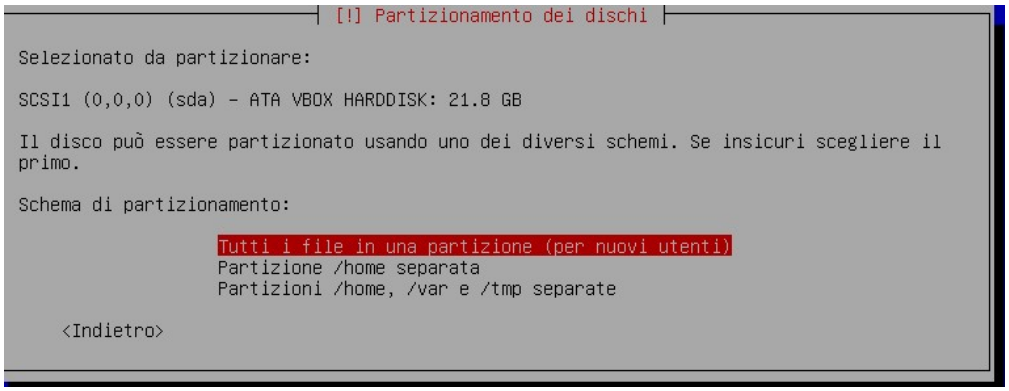


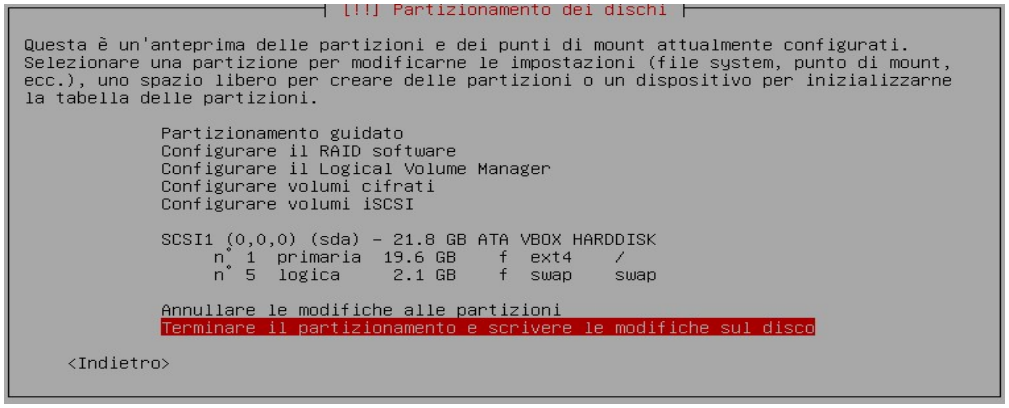
Verrà poi chiesta la **password** di root con relativa conferma, si consiglia di **salvare la password di root**, in quanto indispensabile in ogni operazione.

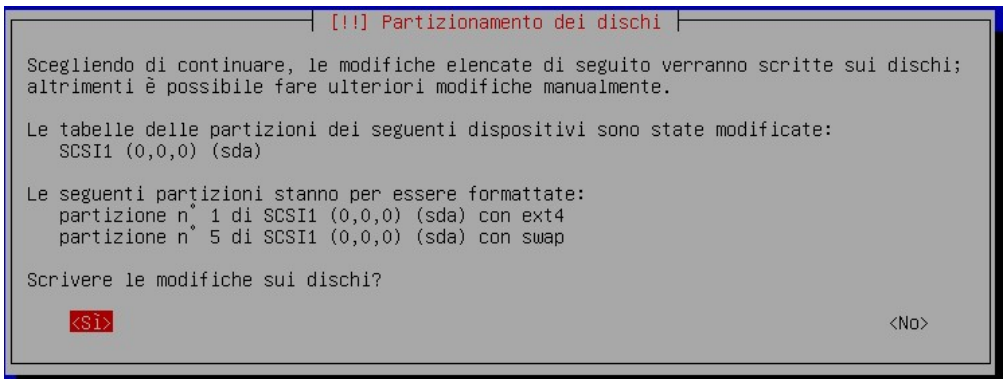
Successivamente dovremo partizionare il disco fisso, si congislia di lasciare le seguenti impostazioni di default.



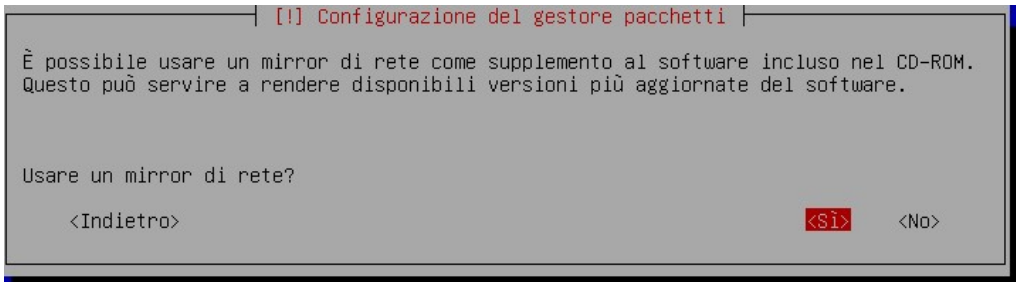


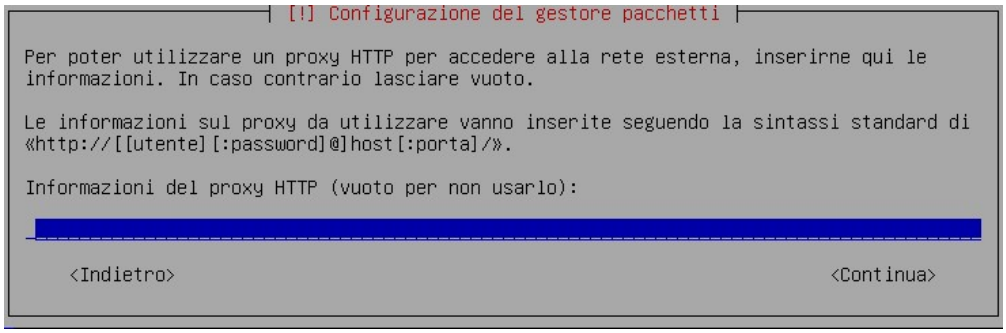






Successivamente verrò chiesto se usare un mirror di rete (scelta consigliata) ed eventualmente presente, le impostazioni del Server Proxy. Nel nostro caso non essendoci il Proxy il campo va lasciato vuoto.

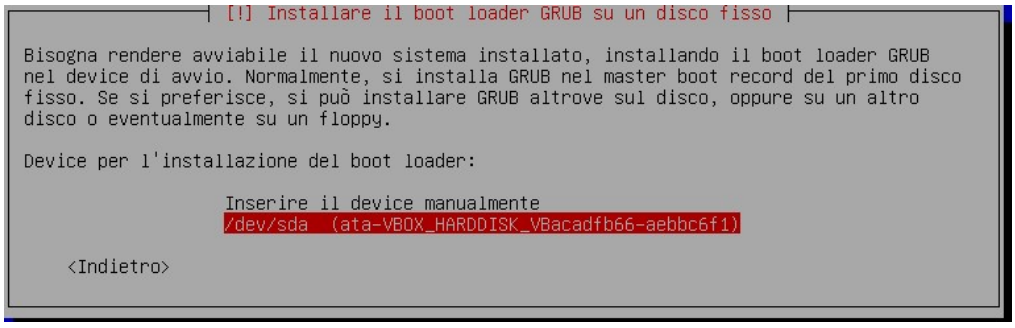




La fase successiva riguard il GRUB (GNU GRand Unified Bootloader) e cioè quella parte del sistema operativo che si occupa della fase di BOOT (avvio) del sistema stesso. Il GRUB normalmente va a modificare l'MBR (Master Boot Record) cioè quella zona fisica dell'Hard Disk del PC che viene letta all'avvio per far partire il sistema operativo della macchina.

Se installassimo Kali su un PC reale dovremmo stare attenti a questa impostazione, nel caso nostro trattandosi di una macchina virtuale, possiamo eseguire questa operazione a cuor leggero.

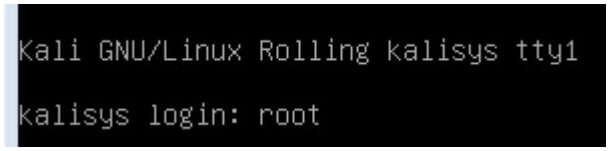
Perciò confermiamo le opzioni secondo queste schermate.



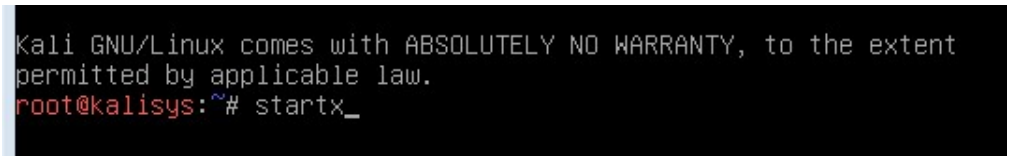
L'installazione è ora terminata e la macchina virtuale con Kali è pronta per essere avviata.

Avviando Kali Linux da Virtual Box con il pulsante “AVVIA”, il sistema operativo partirà nella finestra di Virtual Box, ed una volta avviato chiederà nome utente e password.

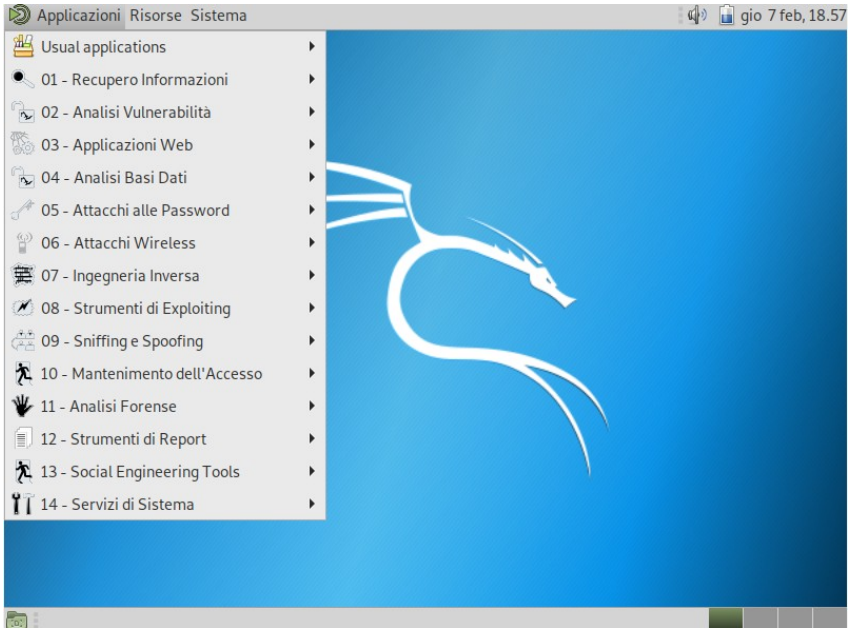
Il nome utente da inserire è **root** la password è quella scelta in fase di installazione.



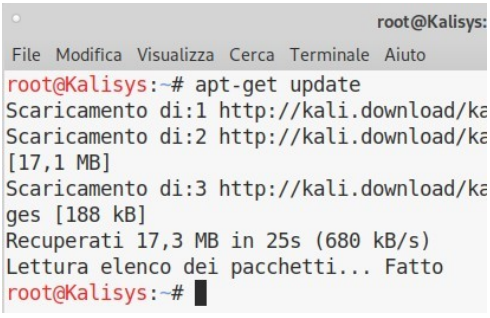
Successivamente ci troveremo il terminale con il prompt dei comandi, a questo punto possiamo decidere se far partire l’interfaccia grafica con il comando “**startx**”



Consiglio di verifficare l’accesso ad internet, tramite l’avvio del browser **firefox** presente nelle applicazioni “**usual application**” e “**internet**”.

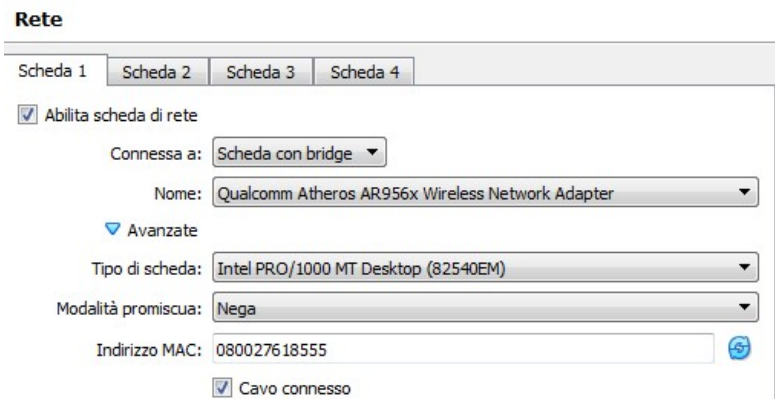


e accedendo al terminale tramite “**usual application**” “**strumenti di sistema**” e “**terminale di mate**”, dobbiamo digitare **apt-get update,** che accede ad internet per controllare la presenza delle liste dei pacchetti presenti nelle repository del sistema operativo, indicate nel file **/etc/apt/sources.list.**



In caso non ci fosse l’accesso ad internet, occorre configurare correttamente la rete dell macchina virtuale accedendo alle impostazioni, nel nostro caso abbiamo questa situazione.

Ovviamente ogni caso andrà valutato singolarmente e con un po’ di conoscenza delle basi ed un po’ di intuito si riuscirà a rendere funzionante la connessione ad internet.



CONCLUSIONI

Possiamo definire quanto detto finora una premessa di quello che sarà il vero e complesso lavoro dell'analisi di una rete. Seguendo questa guida, possiamo però muovere i primi passi e conoscere anche superficialmente i complessi strumenti descritti prima.
In seguito affronteremo nel dettaglio determinate situazioni che ci consentiranno maggiormente di comprendere le varie problematiche delle reti, utilizzando proprio gli strumenti precedentemente descritti.