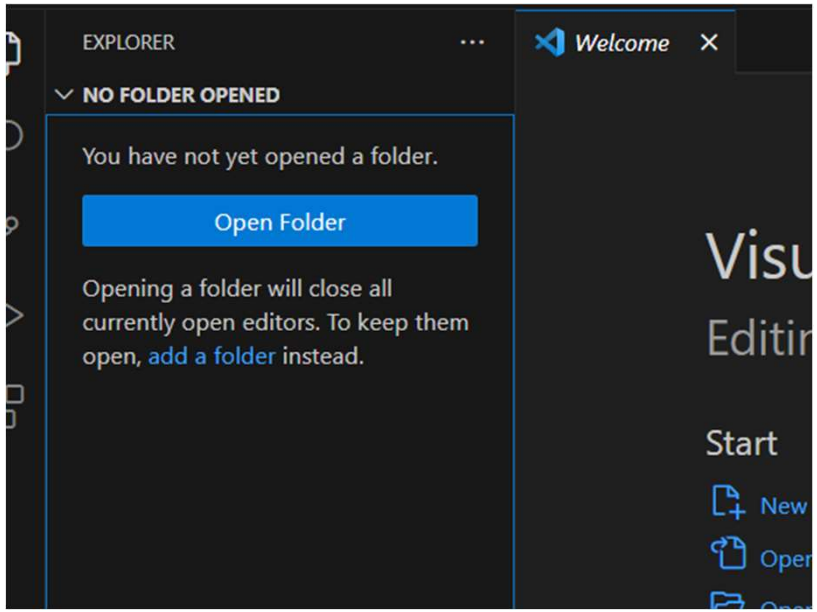
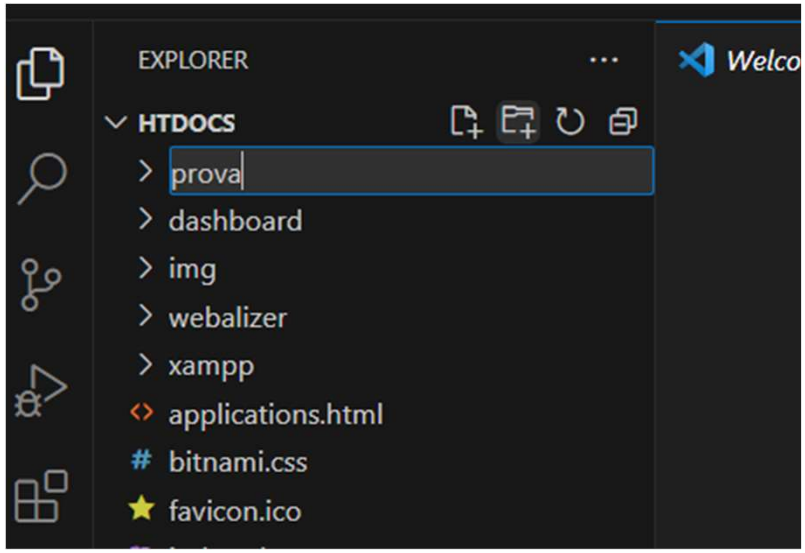
Wireshark

Con Wireshark siamo anche ovviamente in grado di analizzare il traffico di rete relativo allo scaricamento di una pagina web di un sito sviluppato in locale, il sito lo possiamo per esempio creare aprendo Visual Studio Code e puntando alla cartella c:\xampp\htdocs premendo su Open Folder



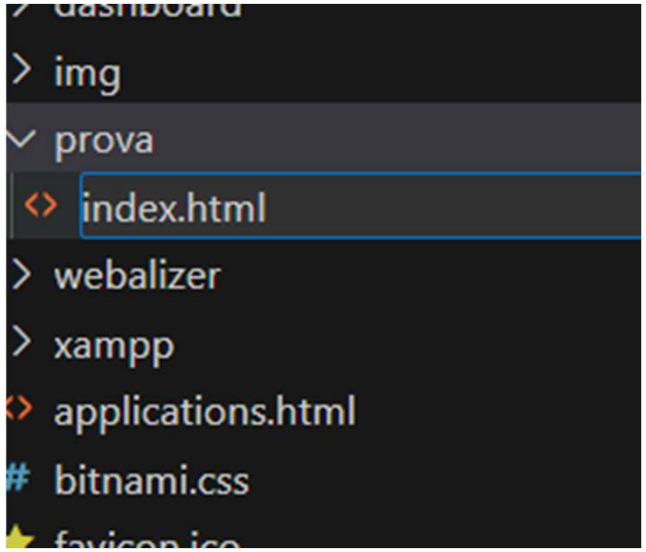
Wireshark

E creando una cartella «prova» al suo interno



Wireshark

All’interno della cartella «prova» possiamo inserire un file html «index.html»

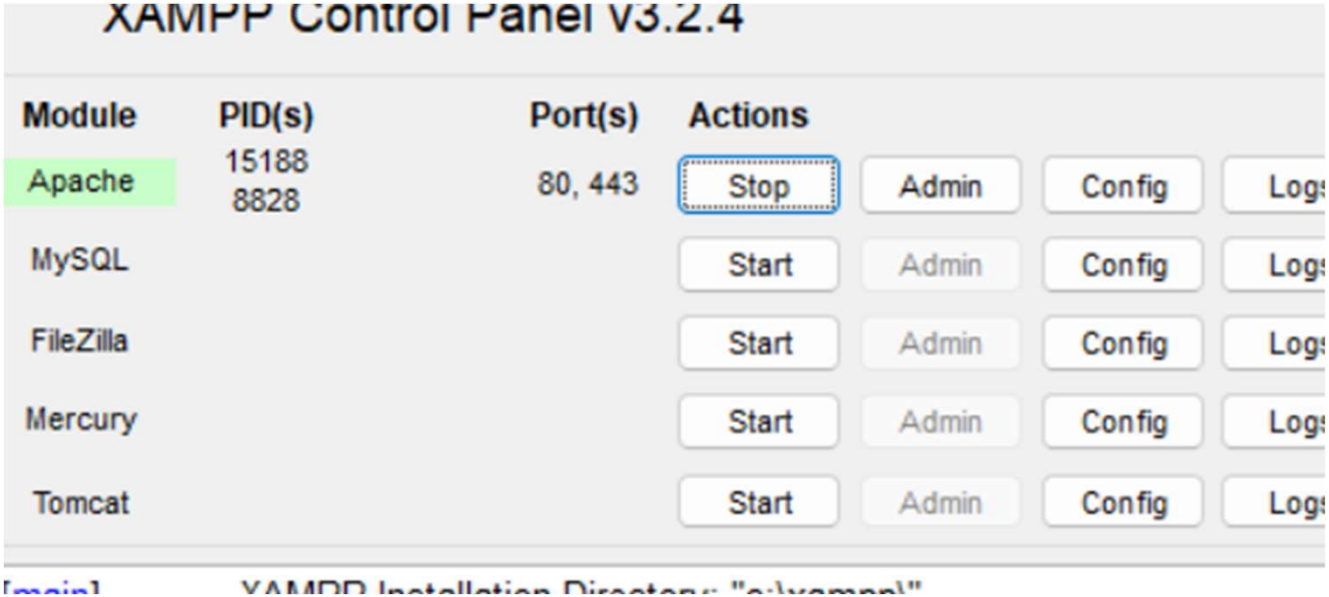


Wireshark

E popolarlo con un po di codice di base html (non è importante ciò che si scrive, basta che ci sia un pò di html)

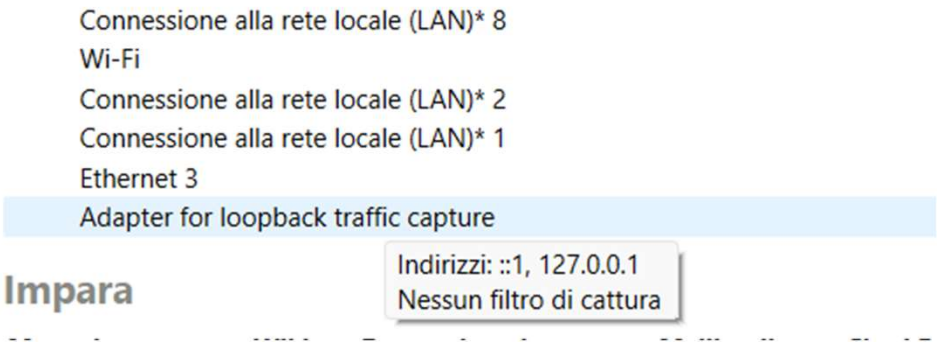
Wireshark

Dopo aver salvato la pagina web possiamo chiudere Visual Studio Code e avviare il web server (la prova viene fatta con il pacchetto xampp)



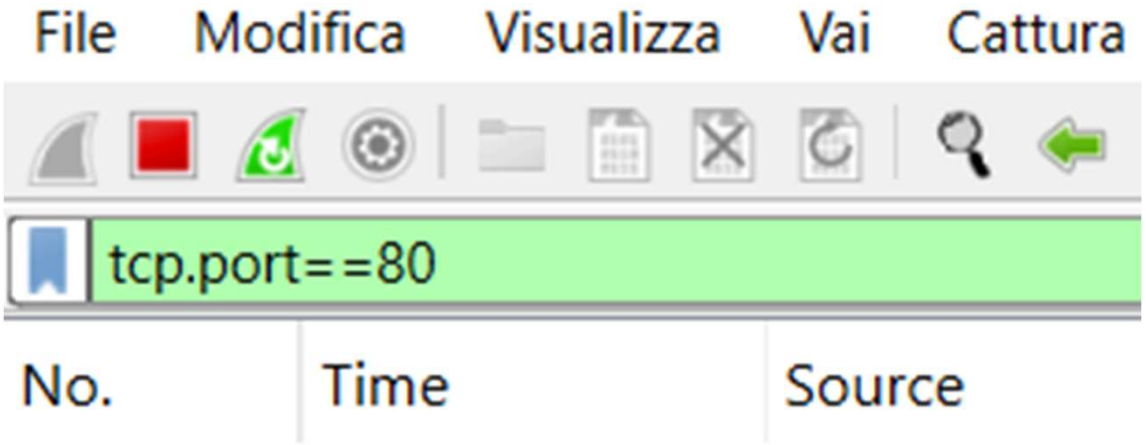
Wireshark

Si apre quindi Wireshark, e si sceglie come interfaccia di rete «Adapter for loopback traffic capture»



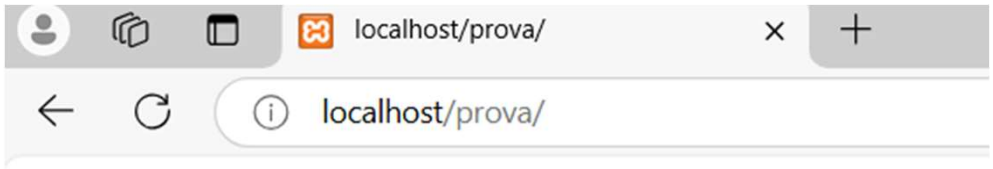
Wireshark

A questo punto per filtrare solo le comunicazioni che avvengono verso un web server locale (naturalmente se non impostato diversamente), poiché le connessioni al server per http avvengono tramite la porta 80 possiamo impostarla come filtro e premere invio



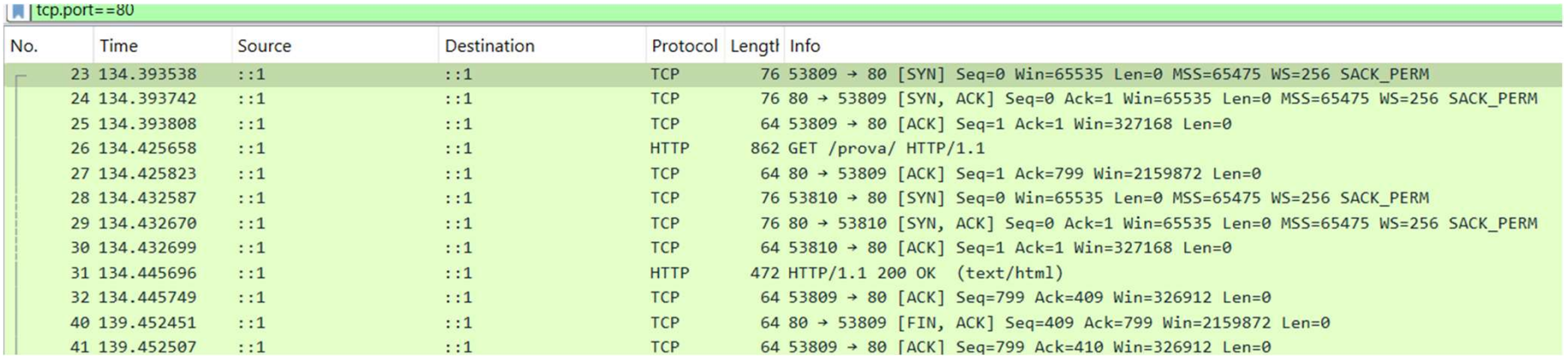
Wireshark

Per collegarci al sito web basta aprire il browser e scrivere



Wireshark

Il browser naturalmente apparirà vuoto ma wireshark ha intercettato tutti i pacchetti di trasmissione con la porta 80 del server

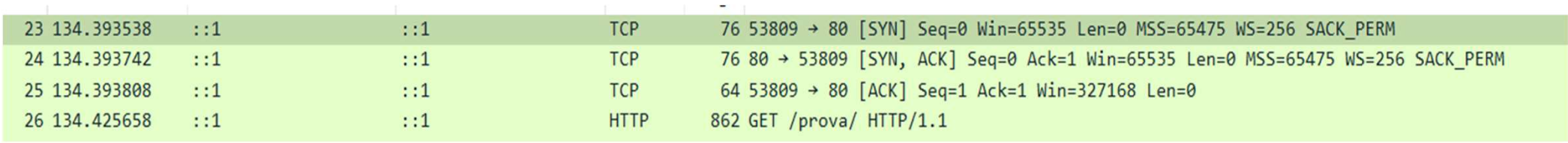


tcp.port==80

No.	Time	Source	Destination	Protocol	Length	Info
23	134.393538	::1	::1	TCP	76	53809 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=65475 WS=256 SACK_PERM
24	134.393742	::1	::1	TCP	76	80 → 53809 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65475 WS=256 SACK_PERM
25	134.393808	::1	::1	TCP	64	53809 → 80 [ACK] Seq=1 Ack=1 Win=327168 Len=0
26	134.425658	::1	::1	HTTP	862	GET /prova/ HTTP/1.1
27	134.425823	::1	::1	TCP	64	80 → 53809 [ACK] Seq=1 Ack=799 Win=2159872 Len=0
28	134.432587	::1	::1	TCP	76	53810 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=65475 WS=256 SACK_PERM
29	134.432670	::1	::1	TCP	76	80 → 53810 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65475 WS=256 SACK_PERM
30	134.432699	::1	::1	TCP	64	53810 → 80 [ACK] Seq=1 Ack=1 Win=327168 Len=0
31	134.445696	::1	::1	HTTP	472	HTTP/1.1 200 OK (text/html)
32	134.445749	::1	::1	TCP	64	53809 → 80 [ACK] Seq=799 Ack=409 Win=326912 Len=0
40	139.452451	::1	::1	TCP	64	80 → 53809 [FIN, ACK] Seq=409 Ack=799 Win=2159872 Len=0
41	139.452507	::1	::1	TCP	64	53809 → 80 [ACK] Seq=799 Ack=410 Win=326912 Len=0

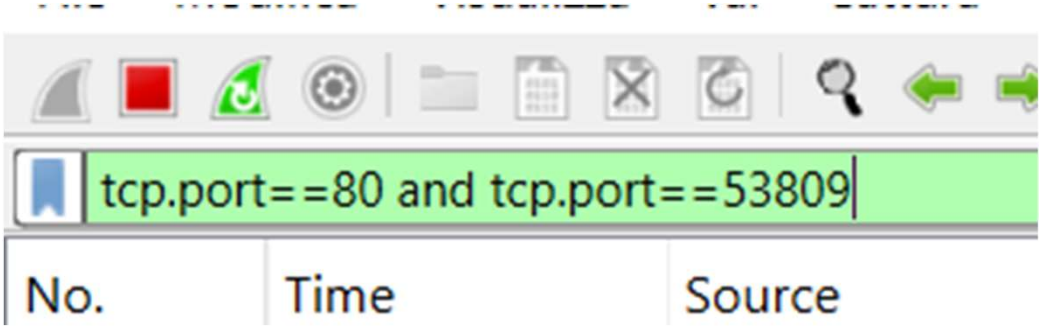
Wireshark

ATTENZIONE ALLE COMUNICAZIONI PERCHÉ I RISULTATI POTREBBERO ESSERE FALSATI, INFATTI LA COMUNICAZIONE DA CONSIDERARE È SOLO QUELLA TRA LA PORTA CHE RICHIEDE LA PAGINA WEB E LA PORTA 80 DEL SERVER, LA COMUNICAZIONE DI ALTRE PORTE NON DEVE ESSERE TENUTA IN CONSIDERAZIONE, In altre parole si tratta delle porte usate per le comunicazioni iniziali di sincronizzazione e la richiesta di GET



Wireshark

Per ottenere quindi una ricerca «pulita» si aggiunge come filtro quello della porta 53809



Wireshark

Le prime tre comunicazioni sono richieste di sincronizzazione:

1. Il client verifica se può connettersi al server per una connessione http
2. Il server risponde
3. Il client risponde al server che ha ricevuto la risposta

23	134.393538	::1	::1	TCP	76 53809 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=65475 WS=256 SACK_PERM
24	134.393742	::1	::1	TCP	76 80 → 53809 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65475 WS=256 SACK_PERM
25	134.393808	::1	::1	TCP	64 53809 → 80 [ACK] Seq=1 Ack=1 Win=327168 Len=0

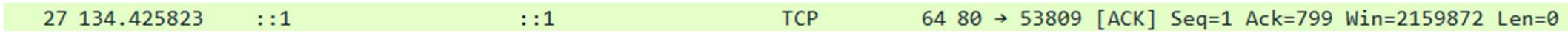
Wireshark

Il client richiede al server l’accesso al sito contenuto nella sua sottocartella prova

26	134.425658	::1	::1	HTTP	862 GET /prova/ HTTP/1.1

Wireshark

Il server gliela invia



Wireshark

Il contenuto iniziale del sito (la pagina iniziale del sito) è stata scaricata correttamente (questa è un'annotazione di wireshark)

31	134.445696	::1	::1	HTTP	472 HTTP/1.1 200 OK (text/html)

Wireshark

Il client dice al server che ha ricevuto tutto senza errori

32	134.445749	::1	::1	TCP	64 53809 → 80 [ACK] Seq=799 Ack=409 Win=326912 Len=0

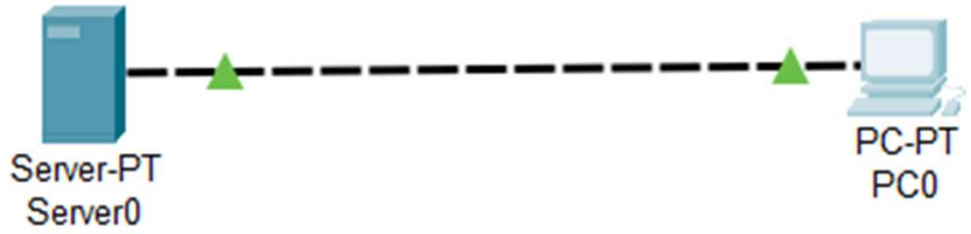
Wireshark

Gli ultimi 4 messaggi sono relativi alla chiusura della connessione, il server risponde che ha ricevuto dal client la comunicazione che tutto è comunica al client la fine dell'attività di comunicazione, il client risponde che ha ricevuto la risposta del server, il server comunica al client la chiusura definitiva della connessione, il client risponde che ha ricevuto la comunicazione del server. Se il client vorrà comunicare con il server, dovrà ripetere il procedimento di sincronizzazione già visto nei primi tre pacchetti (three handshake)

40	139.452451	::1	::1	TCP	64 80 → 53809 [FIN, ACK] Seq=409 Ack=799 Win=2159872 Len=0
41	139.452507	::1	::1	TCP	64 53809 → 80 [ACK] Seq=799 Ack=410 Win=326912 Len=0
42	139.660817	::1	::1	TCP	64 53809 → 80 [FIN, ACK] Seq=799 Ack=410 Win=326912 Len=0
43	139.660862	::1	::1	TCP	64 80 → 53809 [ACK] Seq=410 Ack=800 Win=2159872 Len=0

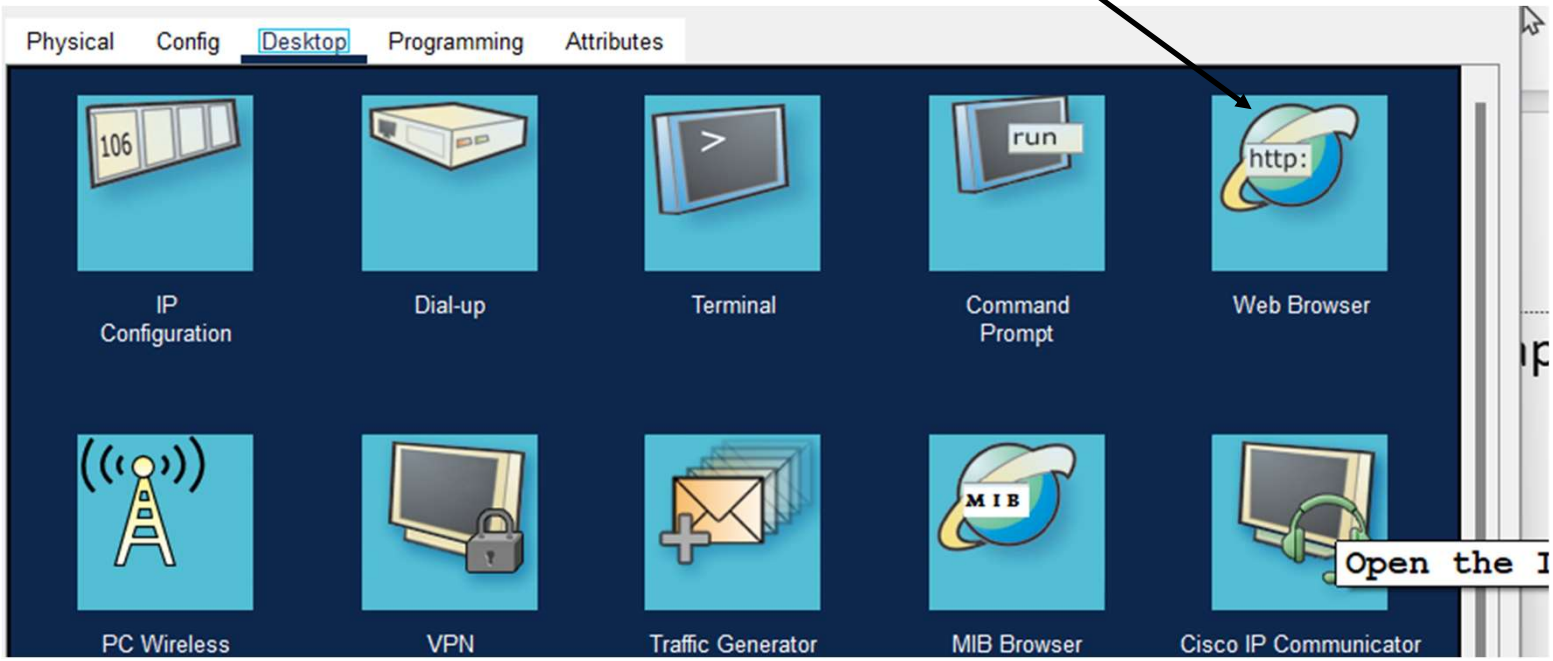
Wireshark

Con Packet Tracer il discorso cambia perché fare un collegamento in locale si può sicuramente ma la cattura dei pacchetti è più difficoltosa e quindi si presenta un collegamento fatto tra client e un server, i passaggi e le procedure di indirizzamento sono molto semplici (uguali a quelli di connessione di due pc) e non vengono spiegati



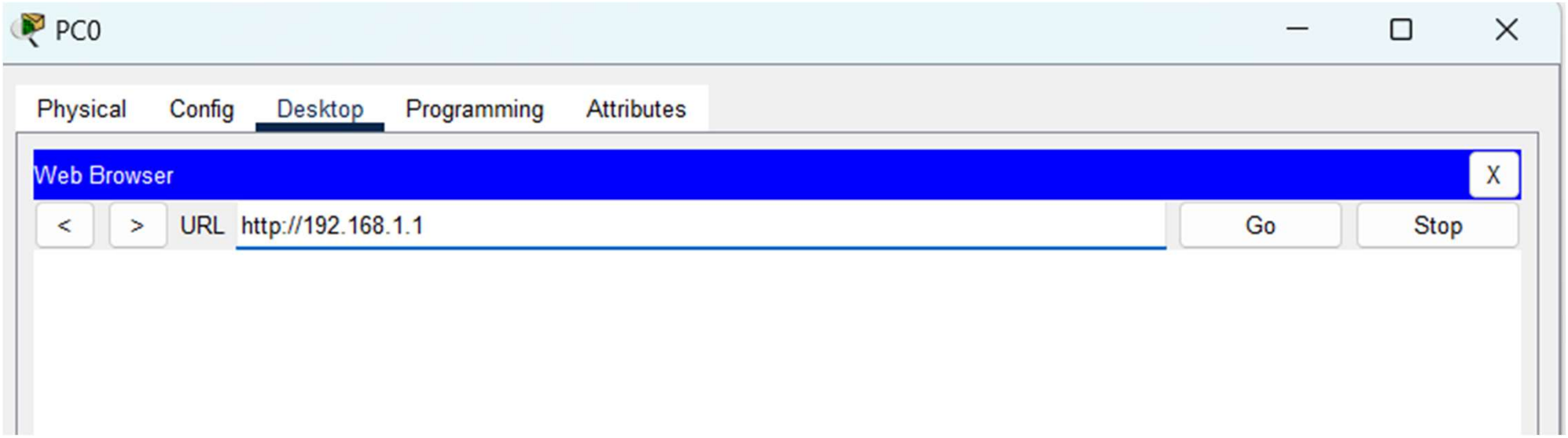
Wireshark

A questo punto è possibile fare click sul pc che si deve collegare al server e aprire la scheda «Desktop» dalla quale si sceglie «Browser»



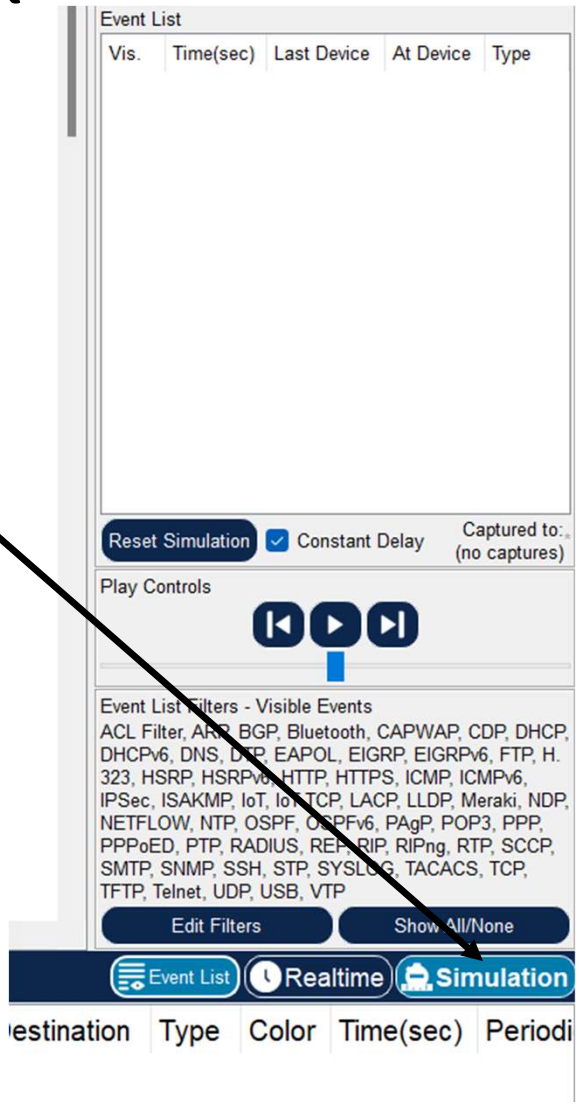
Wireshark

Si inserisce l’indirizzo IP del server da visitare



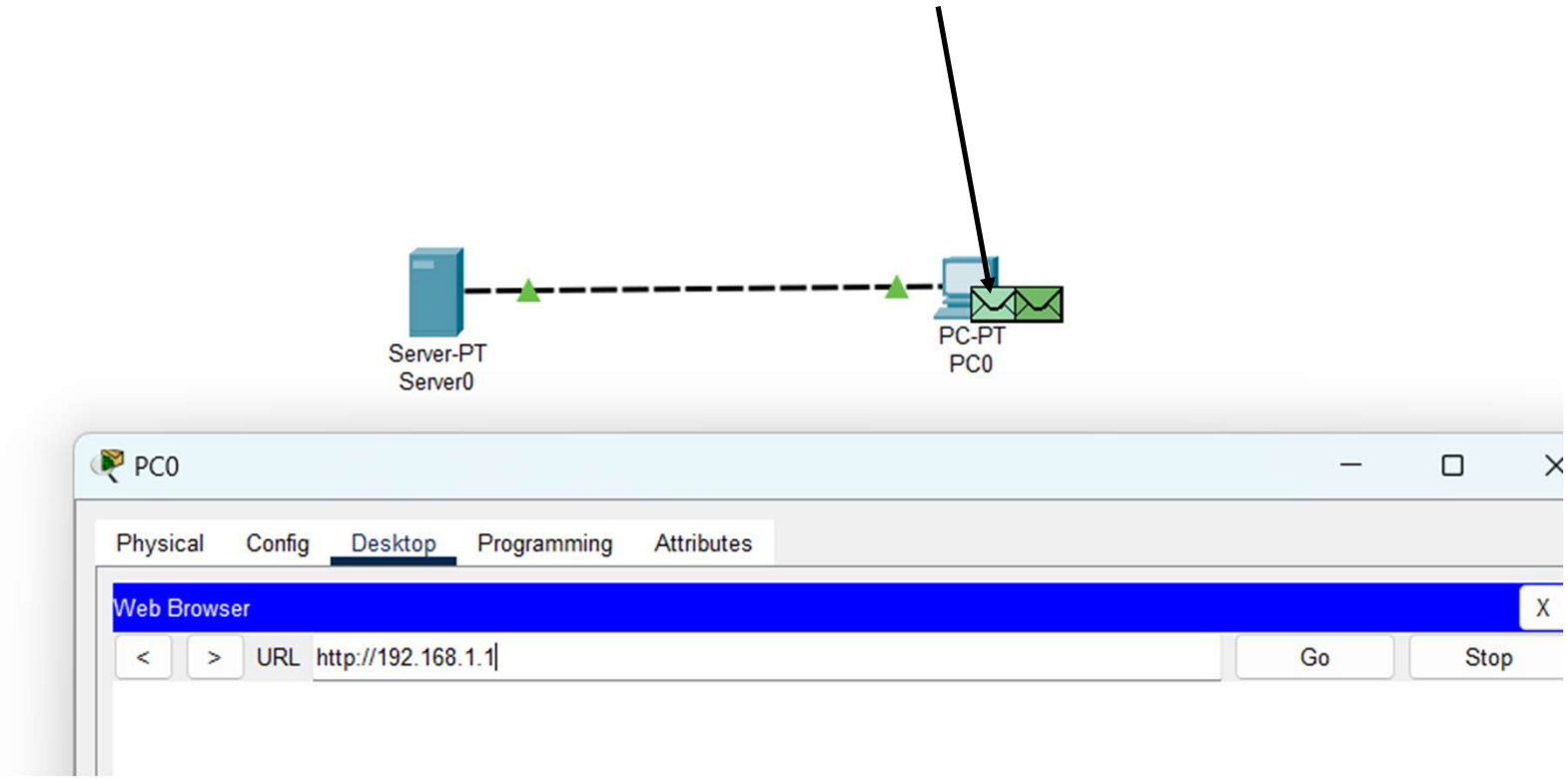
Wireshark

E si apre la modalità simulazione



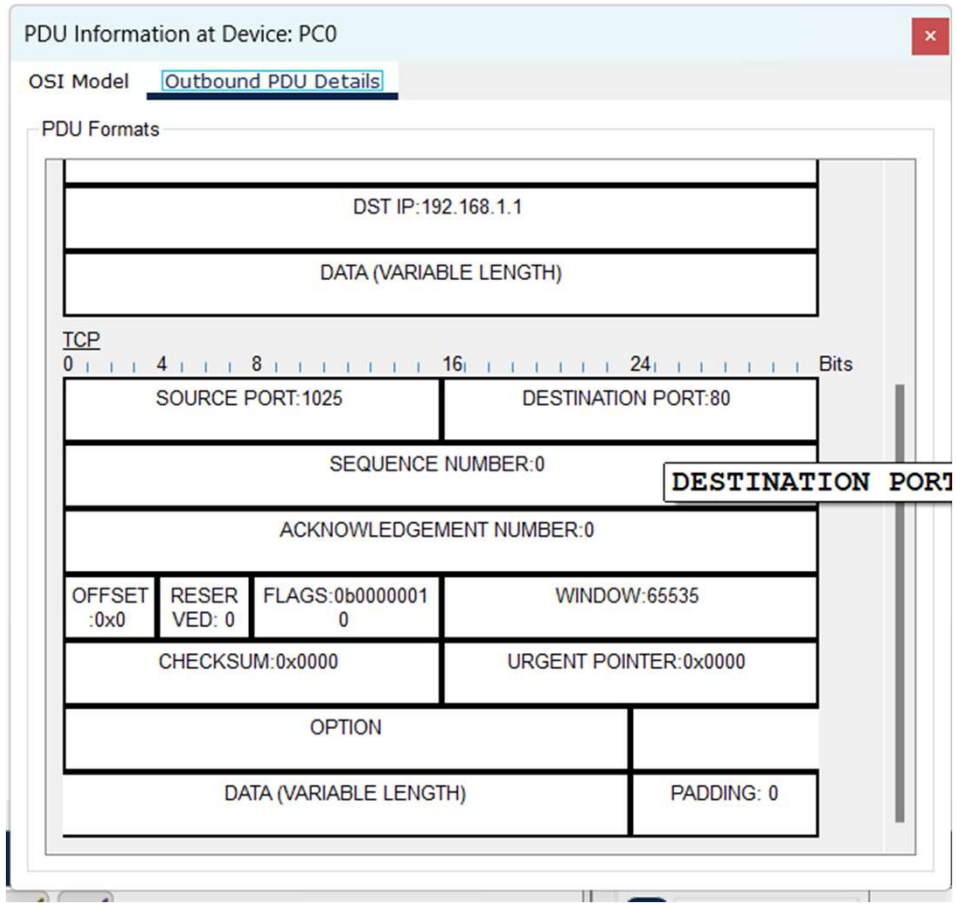
Wireshark

Quindi si preme invio sull'indirizzo da visitare e appaiono sul pc due lettere (pacchetti), per ora ci interessa solo quello di sinistra (il più chiaro)



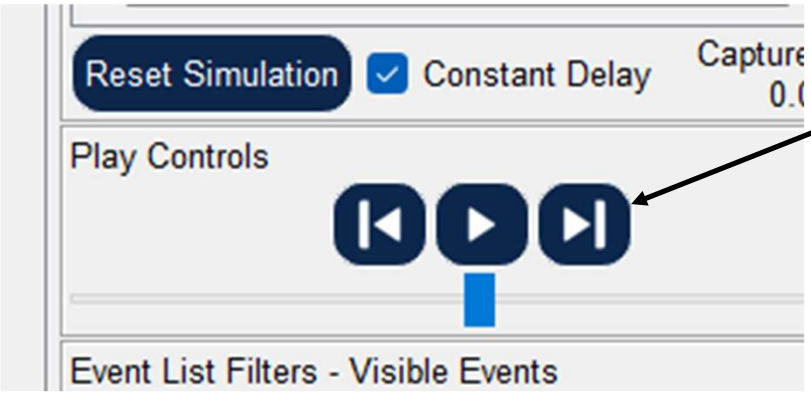
Wireshark

Se si fa click su outbound pdu details si può visionare la struttura del pacchetto che da luogo ad una richiesta di connessione, da lì partirà poi lo scambio dei messaggi



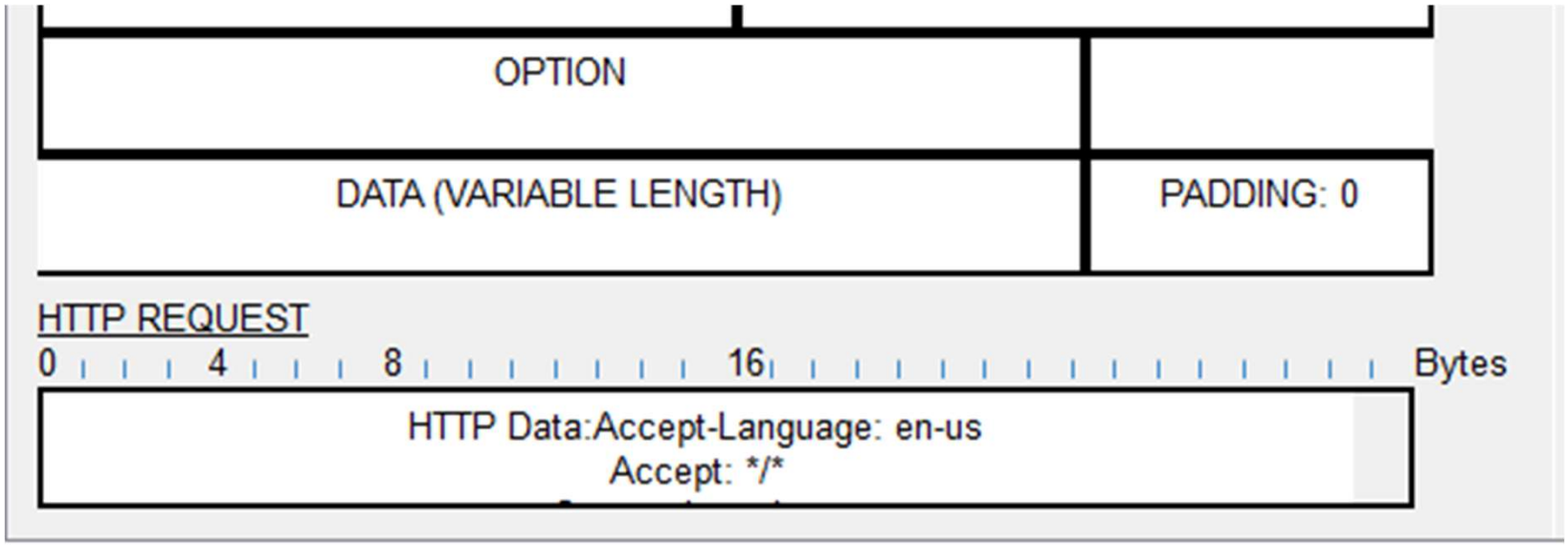
Wireshark

Se si fa click sul pulsante della simulazione per andare avanti di un passaggio, il pacchetto passerà dal client al server e così via, per ogni passaggio sarà possibile visionare la struttura dei pacchetti fino a quando la conversazione terminerà



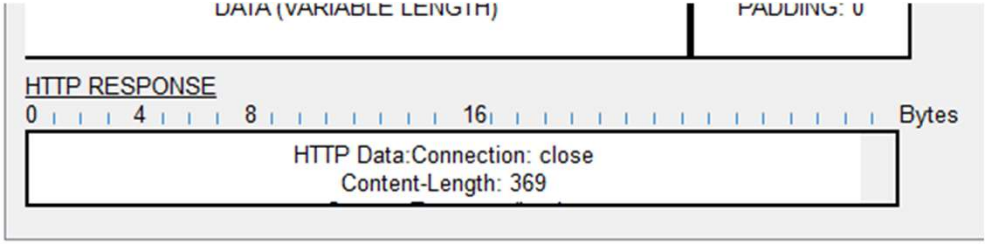
Wireshark

Ad un certo punto noteremo però che il pacchetto inviato dal client avrà questa parte in più



Wireshark

Questa è la richiesta vera e propria al sito web per ottenere la risorsa, il server risponderà inviando al client la pagina principale del sito web (la index)



Wireshark

Le ultime operazioni sono quelle di chiusura della connessione