ANALISI DEI PROTOCOLLI DI POSTA ELETTRONICA

Prima di effettuare degli esercizi utilizzando Cisco Packet Tracer, Wireshark e Putty, facciamo un riepilogo sul funzionamento della posta elettronica.

La posta elettronica è un mezzo di comunicazione asincrono, infatti a differenza di chat e meet, la comunicazione avviene anche quando gli utenti che devono comunicare non sono contemporaneamente connessi alla rete. Questo servizio è composto essenzialmente da due parti;

- mail user agent (o client di posta) cioè il programma che si occupa di comporre il messaggio e trasmetterlo al server, o di riceverlo dal server per poterlo visualizzare sul client (PC).
- mail transfer agent, cioè il programma presente sul server di posta, che si occupa di smistare i messaggi nelle mail-box degli utenti tra i vari server di posta.

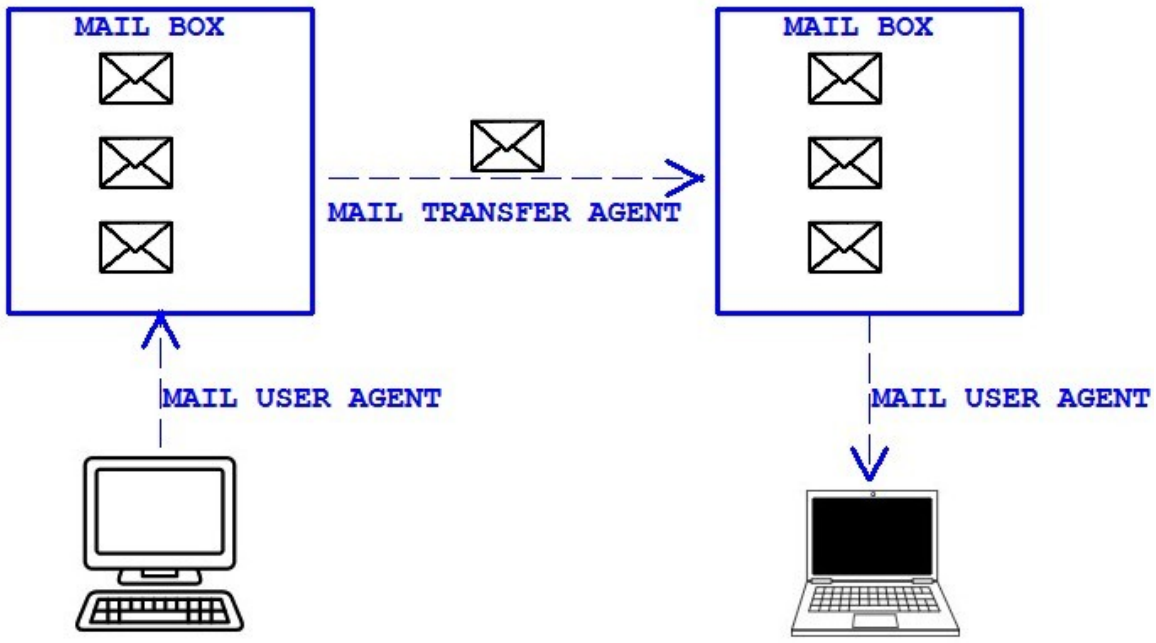
Sul server di posta, risiedono le cartelle degli utent che contengono i messaggi chiamate mail-box.

Sul client del mittente, tramite il mail user agent, viene composto il messaggio ed inviato alla propria mail-box presente nel server.

Il mail transfer agent smista il messaggio presente nella mai-box di uscita del mittente e lo porta alla mail-box di ingresso del destinatario.

Dal client del destinatario, tramite il mail user agent, si accede alla mail-box per scaricare il messaggio.

L'accesso alle mail-box avviene ovviamente tramite autenticazione.



L'invio del messaggio sul server non è immediato in quanto dipende dal transito su altri server, che avviene memorizzando il messaggio su disco per poi inviarlo (store e forward).

La trasmissione e la ricezione del messaggio avviene con differenti protocolli, per la posta in uscita si utilizza il protocollo **SMTP** (**S**imple **M**ail **T**ransfer **P**rotocol) per quella in ingresso ci sono due possibili protocolli, il POP o POP3 e l'IMAP. La comunicazione in entrata ed in uscita è criptata con i protocolli **SSL** (**S**ecure **S**ockets **L**ayer) e **TLS** (**T**ransport **L**ayer **S**ecurity)

Una piccola parentesi sui due protocolli di posta in entrata e cioè **IMAP** (**I**nternet **M**ail **A**ccess **P**rotocol) e **POP** (**P**ost **O**ffice **P**rotocol).

Il protocollo POP prevede che i messaggi vengano scaricati dal server sul pc dove è installato il client e poi cancellati dal server, mentre con il protocollo IMAP i messaggi vengono scaricati ma rimangono salvati sul server.

Nel caso del protocollo POP è comunque possibile selezionare un'impostazione sul web server per mantenere comunque i messaggi sul server.

Nel caso del protocollo IMAP, i messaggi risultano sincronizzati tra il client ed il server, in quanto le modifiche effettuate su un messaggio dal lato client, vengono poi riprodotte anche sul server, in modo che da webserver o da altri client, i messaggi siano sempre aggiornati all'ultima modifica. Il protocollo IMAP è più nuovo e più complesso del POP.

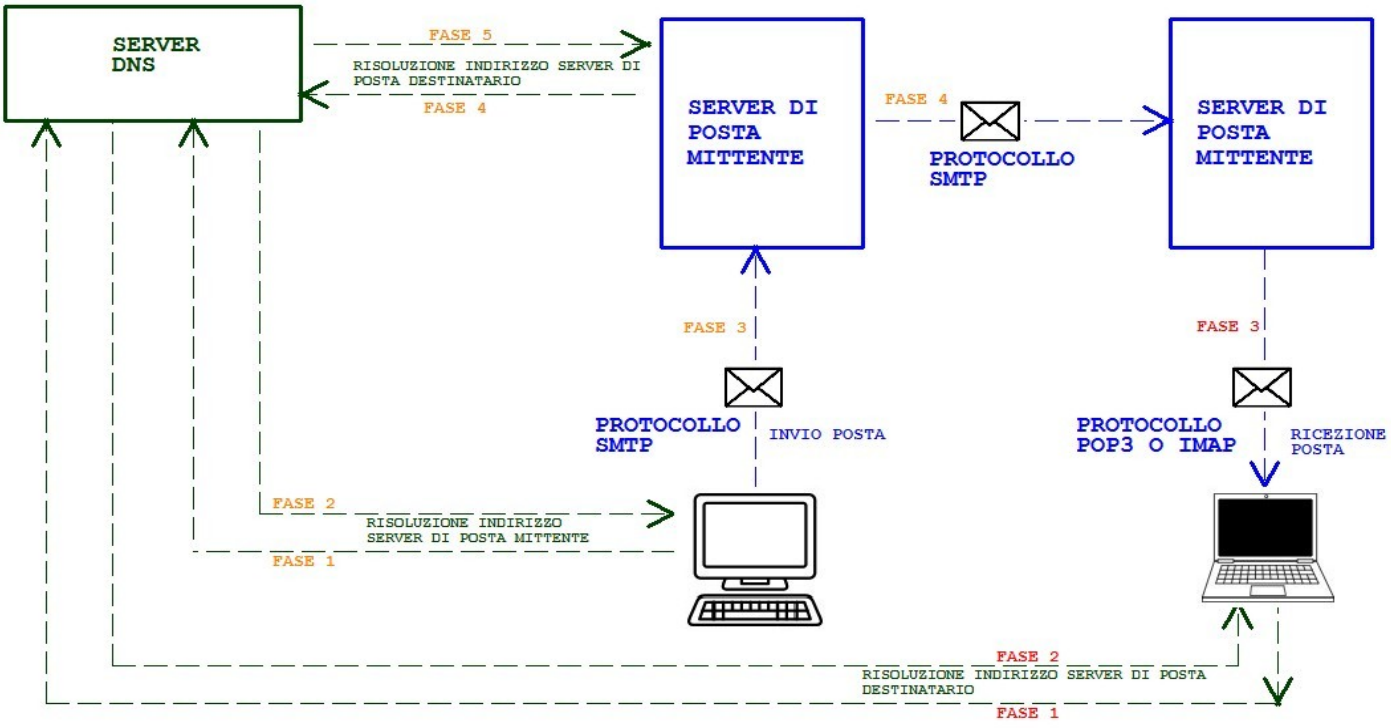
Caratteristiche del protocollo POP

- Le email vengono scaricate dal client e sono sempre accessibili sul PC , anche senza connessione.
- La connessione serve solo durante l'invio e la ricezione dei messaggi.
- Il server non deve mantenere i messaggi, risparmio dello spazio sul Server.
- I messaggi possono comunque rimanere sul Server.

Caratteristiche del protocollo IMAP

- La posta rimane sul Server ed è accessibile da più postazioni o client diversi.
- La connessione ad internet è sempre necessaria per accedere alla posta
- Fino a quando non viene richiesto il messaggio intero, vengono scaricate solo le intestazioni per una più rapida panoramica dei messaggi.
- I messaggi restano sul server, perciò si risparmia spazio sui dispositivi e sui PC.
- Non si perdono i messaggi in quanto la posta rimane sul server.
- Le mail possono comunque essere scaricate tramite il client sul PC.

Nella seguente immagine la struttura di un'invio e ricezione di una email, con la sequenza delle varie fasi
In arancione le fasi dell'invio, in rosso le fasi della ricezione.



Esercizio 1:

SIMULAZIONE DEL SERVIZIO DI POSTA ELETTRONICA CON CISCO PACKET TRACER

Seguendo la dispensa al link:
https://danielepostacchini.it/wp-content/uploads/2020/04/Mail-con-Cisco-Packet-Tracer.pdf
Proviamo ad inserire nella rete aziendale dell'esercizio, un server di posta elettronica esterna ed interna.

Per la posta esterna creare due utenti con il vostro nome e cognome nel seguente modo:

- **nome.cognome@alfaproject.it** password **1234**
- **cognome.nome@alfaproject.**it password **4321**

Per la posta interna creare due utenti con il vostro nome e cognome nel seguente modo:

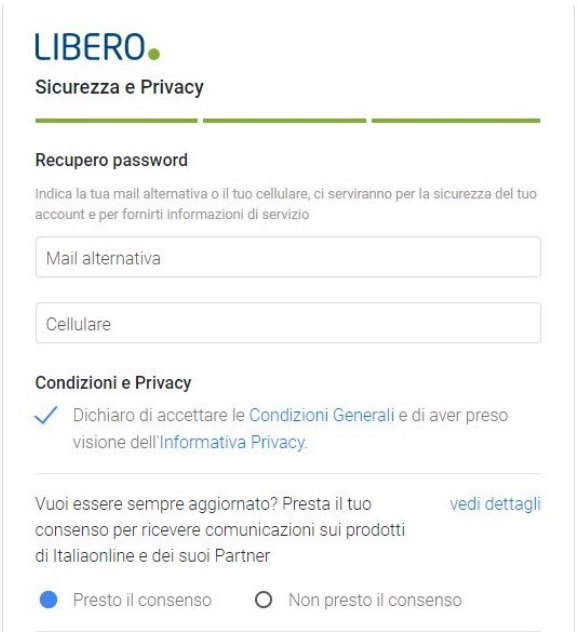
- **nome.cognome@intmail.it** password **1234**
- **cognome.nome@intmail.it** password **4321**

Impostare i 4 utenti su 4 PC della rete e provare ad inviare la posta tra i due utenti esterni e poi tra i due utenti interni.
Scrivere nel file di Cisco Packet Tracer vicino ad ogni PC il nome utente configurato nel PC.

---

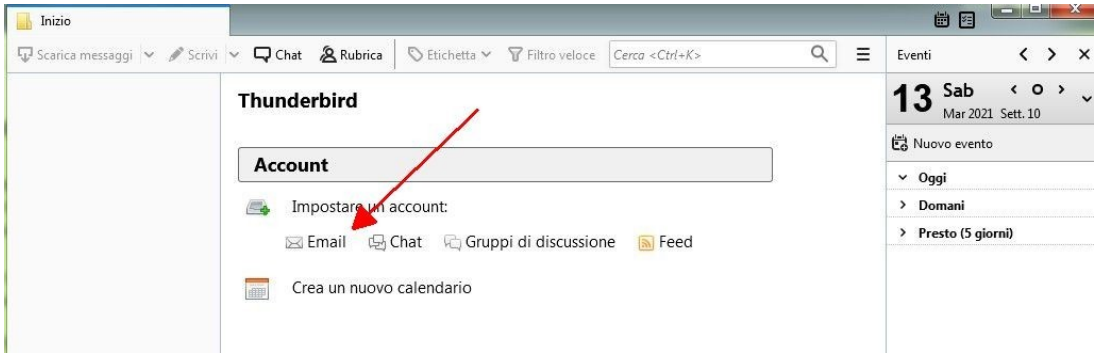
2) SNIFFING DEI PROTOCOLLI SMTP E POP CON WIRESHARK

Per poter intercettare il traffico in uscita ed in entrata durante l'invio e la ricezione di una mail, dobbiamo innanzitutto installare un client di posta elettronica.
Procediamo innanzitutto con l'installazione del client di posta "Mozilla Thunderbird".
Scaricare ed installare il client di posta all'indirizzo **https://www.thunderbird.net/it/**
Ora creiamo una mail sul dominio libero.it. Accedere all'indirizzo **https://login.libero.it/** e cliccare su "CREA ACCOUNT". Seguire tutte le indicazioni fino al termine.



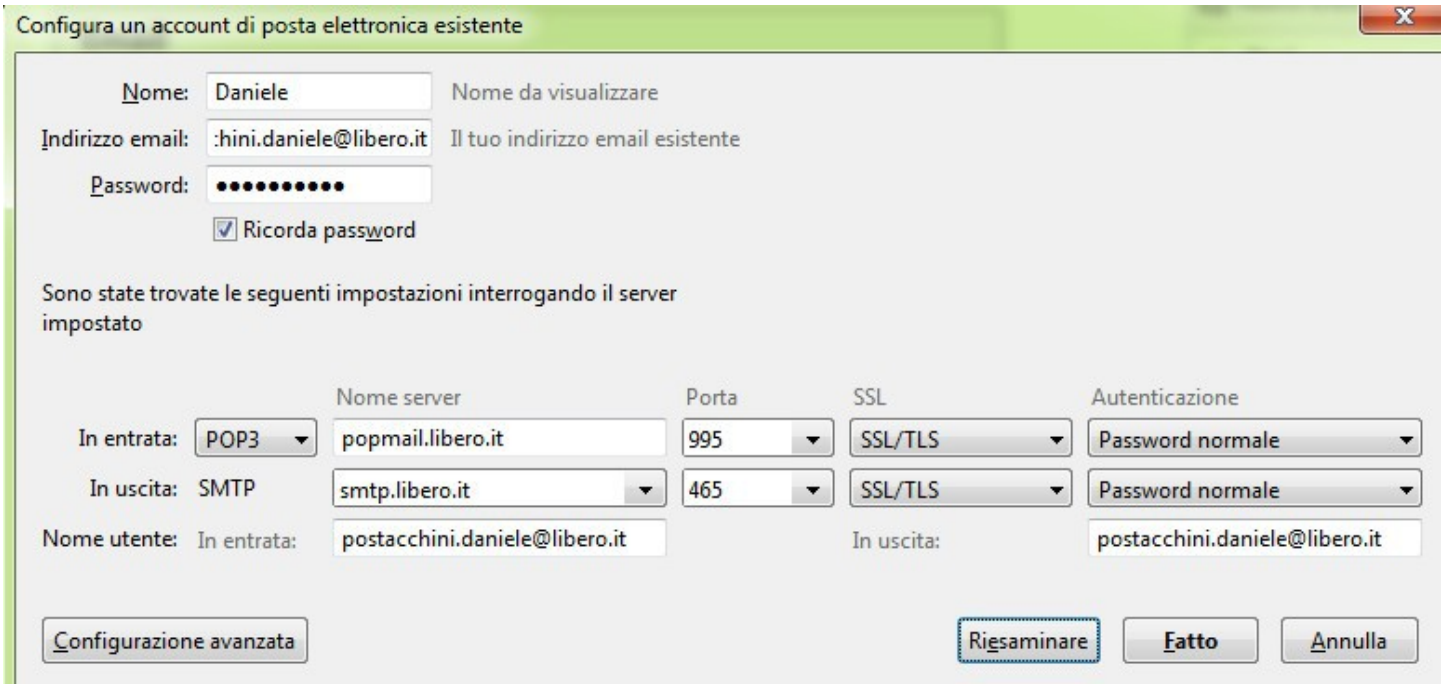
Dopo aver creato l'account chiudere browser ed aprire Mozilla Thunderbird.

Sulla sezione Account,
fare clic su Email,
sotto Impostazioni account.



Inserire l'indirizzo e la password creata su libero.it e fare clic su "Configurazione Manuale"

Nella successiva finestra introdurre i seguenti valori, impostando POP3 come protocollo del server posta in entrata e SMTP come protocollo del server posta in uscita. Fare clic su "RIESAMINA" e se non ci sono errori fare clic su "FATTO".



Queste schermate possono essere differenti con le versioni più recenti di Thunderbird, ma le impostazioni restano le stesse.

Ora il client di posta è pronto per inviare e ricevere messaggi.

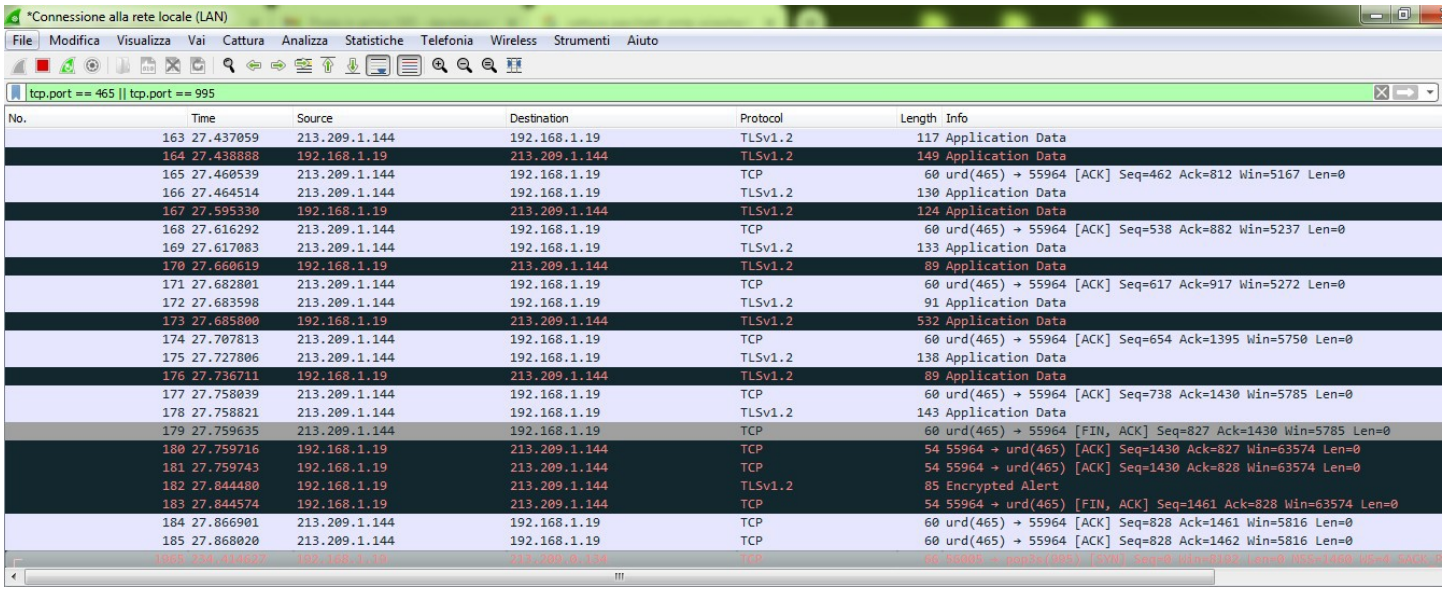
Avendo configurato il client di posta Thunderbird con il protocollo POP3, intercetteremo questo tipo di protocollo, essendo la comunicazione criptata, Wireshark ci consentirà di catturare tutti i pacchetti relativi all'invio ed alla ricezione delle mail, ma non ci permetterà di vederne il contenuto.

Dopo aver lanciato Wireshark ed aver avviato la cattura dei pacchetti sulla porta di rete connessa ad internet, impostiamo il filtro di visualizzazione nei seguenti modi:

- tcp.port == 465 visualizzo i traffico indirizzato alla porta 465 (SMTP)
- tcp.port == 995 visualizzo il traffico indirizzato alla porta 995 (POP3)
- tcp.port == 465 || tcp.port == 995 visualizzo il traffico indirizzato alla porta 465 o alla porta 995

In quest'utimo caso visualizzo sia il traffico della posta in uscita (SMTP) che della posta in entrata (POP3).

Ovviamente per vedere il traffico dobbiamo inviare e ricevere una mail mediante il client Thunderbird.
Un esempio di cattura è il seguente:



Si può notare che i protocollo di criptaggio TLS viene utilizzato per i pacchetti identificati con la scritta "Application Data" che sono quelli che contengono le informazioni dei messaggi che vengono criptate (contenuto, indirizzi email, ecc.…) mentre invece i messaggi non criptati non contengono informazioni importanti, ma sono necessari per la comunicazione con i server come ad esempio per la gestione dell'handshaking. All'interno del pacchetto "Application Data" è indicato il protocollo utilizzato SMTP o POP3.

Esercizio 2:

Installare Thunderbird, creare una casella di posta su libero, e provare a catturare l'invio di un pacchetto con Wireshark.

3) ANALISI DEL CONTENUTO DI UN MESSAGGIO DI POSTA ELETTRONICA

Una piccola parentesi

L'**IETF** (**I**nternet **E**ngineering **T**ask **F**orce) è un organismo internazionale che si occupa di redigere delle norme tecniche per standardizzare il mondo delle telecomunicazioni e di Internet. L'ente composto da tecnici ed esperti programmatori, lavora suddiviso in gruppo preparando dei documenti definiti RFC (**R**equest **F**or **C**omments) validati poi da un altro ente certificatore denominato **IESG** (Internet Engineering Steering Group) quest'ultimo provvedere a promuovere questi documenti come standard ufficiali a livello mondiale.

Tutti gli standard ed i documenti prodotti sono open source e disponibili gratuitamente.

Il documento dell'**IETF RFC2822** , definisce la struttura del messaggio di posta nel seguente modo:

Busta (envelope) contiene le informazioni necessarie per il trasporto: indirizzi, priorità ed altre informazioni.

Intestazione *(header)* contiene informazioni di controllo è costituita da campi posti su più righe contenenti una parola chiave che è il nome del campo, ed un valore rappresentato con una stringa ASCII. Minimo ha due campi indirizzo mittente e destinatario.

Corpo (body) contiene il contenutoi del messaggio inizialmente codificato in ASCII a 7 bit, poi esteso per supportare allegati multimediali, definiti tramite lo standard **MIME** (**M**ultipurpose **I**nternet **M**ail **E**xtensions).

	Busta <*indirizzo mittente*> <*indirizzo destinatario*>
Intestazione	<*nome campo1*>: valore
	<*nome campo2*>: valore
	...
	<*nome campon*>: valore
	(Riga Vuota)
	<*Corpo Del Messaggio*>

Lo standard MIME viene regolamentato dalle RFC 1521, 1341 e 822. In aggiunta alla codifica ASCII questo standard prevede una struttura aggiuntiva al corpo del messaggio con nuove regole per i messaggi non ASCII. Lo standar prevede l'invio di nuove intestazioni contenenti la versione del MIME, il contenuto (Content Type) la descrizione (Content Description) e l'ID (Conten ID).

Per vedere la struttura di un messaggio di posta elettronica con Mozilla Thunderbird, è sufficiente selezionare il messaggio e "visualizzare il sorgente", o su gmail "visualizza originale.

Esercizio 3:

Cercare il documento RFC2822 e descrivere sinteticamente la struttura prevista, successivamente inviare e ricevere una mail con Thunderbird, e visualizzare la struttura del messaggio con i relativi campi.

PROTOCOLLO SMTP (**S**imple **M**ail **T**ransfer **P**rotocol)

SMTP è un protocollo Client-Server definito dalle RFC 882 e 5321, utilizza il TCP come livello trasporto ed opera sulla porta 25, per connessioni crittografate opera invece sulla porta 465. La sua funzione è quella di stabilire una connessione tra le due macchine al fine di trasferire un messaggio di posta.

Dopo aver stabilito la connessione, il client invia il messaggio al server, il quale mette il messaggio nella casella di posta dell'utente (mailbox) per poi inoltrarlo al server del destinatario.

L'invio del messaggio è articolato in 3 fasi:

- handshaking, cioè l'apertura della connessione,
- trasferimento del messaggio,
- chiusura della connessione.

Client e Server comunicano tramite stringhe di testo composte in genere da comandi di 4 caratteri per il client e risposte di 3 caratteri per il server che descrivono un codice di stato.

Una panoramica dei comandi è la seguente:

- HELO sta per "Hello", avvio della sessione, il client si connette con il suo nome di PC
- MAIL FROM il client indica il mittente,
- RCPT TO il client indica il destinatario,
- DATA il client inoltra l'email,
- RSET il client interrompe la trasmissione, ma mantiene attiva la connessione,
- VRFY/EXPN verify/expand, il client verifica se la casella di posta può ricevere il messaggio,
- NOOP il client chiede la risposta al server, per evitare la chiusura connessione per un time-out,
- QUIT il client chiude la connessione.

Di seguito una panoramica dei codici di stato inviati come risposta dal server:

- 200 richiesta accettata,
- 214 risposta al comando HELP, contiene informazioni,
- 220 il server è pronto,
- 221 il server chiude il canale di trasmissione, risposta al comando QUIT
- 250 il server indica che il mittente o il destinatario è stato ricevuto, risposta a MAIL TO o RCPT TO
- 354 il server risponde al comando data.

Ovviamente ci sono altri comandi ed altri codici di stato definiti nelle RFC nominate precedentemente.

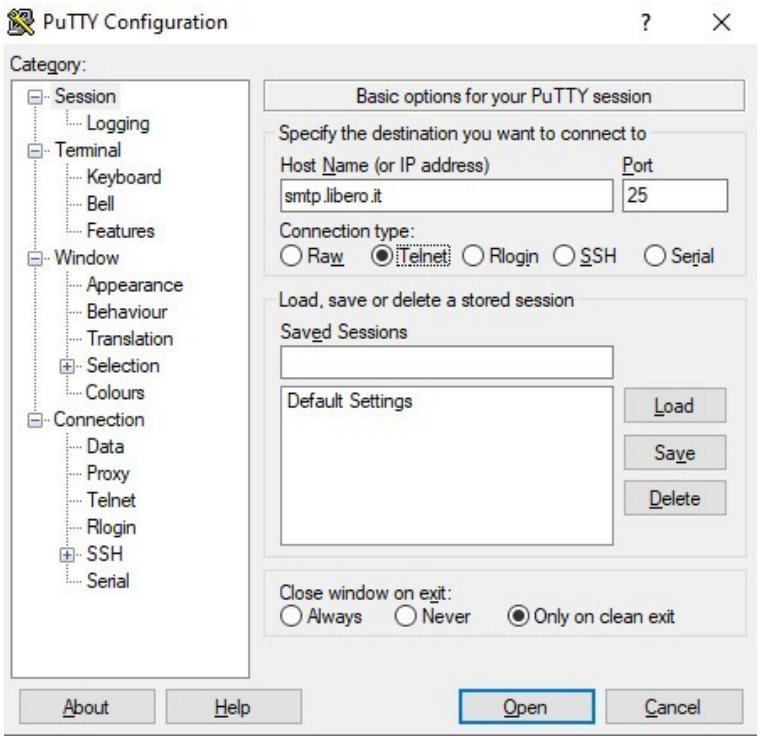
Il protocollo SMTP si è poi evoluto introducendo l'autenticazione, sono stati pertanto aggiunti dei comandi descritti nella RFC 4954 ((SMTP Service Extension for Authentication).

In questo caso il primo comando da inviare al server è HELO, il server risponderà con l'elenco dei comandi aggiuntivi previsti nell'estensione, tra cui AUTH LOGIN, necessario per l'autenticazione.

Esercizio 4 – invio manuale di una email

Proviamo ad utilizzare PUTTY con protocollo TELNET per inviare una mail usando i comandi SMTP in modo manuale.

TELNET è un protocollo client-server basato su TCP/IP, Putty invece è un software lato client che può operare con diversi tipi di connessioni, tra cui TELNET ed SSH.
La prima operazione da fare è lanciare Putty da riga da comando ed inserire il nome del server SMTP con la relativa porta 110 (comunicazione non crittografata) selezionando il protocollo TELNET.



Una volta avviata la connessione con Telnet, digitiamo il comando **EHLO server**, il server risponderà con i comandi aggiuntivi tra cui quello necessario per l'autenticazione **AUTH LOGIN**.
Digitando quest'ultimo comando ci verrà chiesto il nome utente e la password, **codificati in base64**.
Senza addentrarci nei particolari della codifica base64 che possiamo trovare al seguente link:
https://it.wikipedia.org/wiki/Base64#:~:text=Base64%20%C3%A8%20un%20sistema%20di,i%20dati%20nel%20formato%20ASCII
Convertiamo il nome utente e la password con il convertitore presente al seguente link:
https://toolset.mrw.it/dev/base64-encoder-decoder.html

Le stringhe ottenute andranno inserite dopo il comando AUTH LOGIN, una per volta.
L'ultima riga contiene il punto, e successivamente dovremo dare il comando QUIT.

A questo punto possiamo verificare il corretto invio del messaggio, andando sulla posta del destinatario, controllando la presenza del messaggio, ed aprendolo con "Mostra originale" o "Visualizza sorgente".

```
EHLO server
250-smtp-31.iol.local hello [188.153.99.97], pleased to meet you
250-HELP
250-AUTH LOGIN PLAIN
250-SIZE 50000000
250-8BITMIME
250-STARTTLS
250 OK
AUTH LOGIN
334 VXNlcm5hbWU6
cG9zdGFjY2hpbmkuZGFuaWVsZUBsaWJlcm8uaXQ=
334 UGFzc3dvcmQ6
TWFudWVsYV83
235 ... authentication succeeded
MAIL FROM: postacchini.daniele@libero.it
250 <postacchini.daniele@libero.it> sender ok
RCPT TO: daniele.postacchini@gmail.com
250 <daniele.postacchini@gmail.com> recipient ok
DATA
354 OK
From:daniele_libero
To:daniele_gmail
Subject:Prova SMTP
Sto provando l'invio manuale
.
250 DwL9nAWJe2tp3DwLenQ9NZ mail accepted for delivery
QUIT
```

Nome utente codificato in base64

Password codificata in base64

Provare a ripetere le operazioni catturando i pacchetti su wireshark ad ogni comando. Per visualizzare i pacchetti mettere il seguente filtro di visualizzazione tcp.port == 25. Verificare l'invio di comandi e la ricezione dei codici di stato.

PROTOCOLLO POP3 (Post Office Protocol 3)

Per leggere i messaggi di posta il destinatario accede mediante il suo client di posta alla propria mail box.

Uno dei protocolli per svolgere questa funzione è il POP3, definito dalle RFC 1734,1939,1957,2449. Questo protocollo si appoggia come nel caso dell'SMTP al TCP ed opera sulla porta 110, per le connessioni crittografate sulla porta 995. Anch'esso ha una struttura client-server ed utilizza dei comandi per eseguire le operazioni necessarie per ricevere e leggere i messaggi di posta. La comunicazione si articola in 3 fasi:

1. apertura della connessione,
2. trasferimento del messaggio,
3. chiusura della connessione.

La fase del trasferimento è composta a sua volta da 3 passaggi:

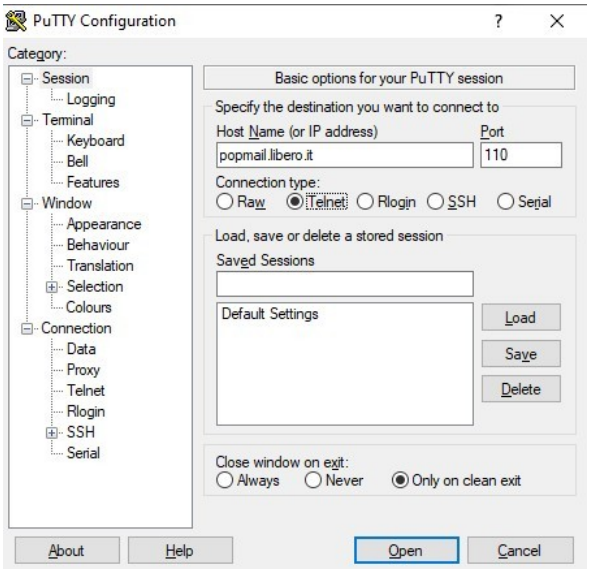
1. autenticazione; il client accede alla propria mail-box mediante nome e password,
2. transazione; il client riceve i messaggi di posta che vengono segnati per poi essere eliminati nella fase successiva,
3. aggiornamento; il server cancella i messaggi e chiude la connessione.

Anche in questo caso vediamo una panoramica del comandi più comuni:

- USER il comando serve per inviare il nome dell'utente, ovvero il suo indirizzo email,
- PASS invia la password,
  (in questo caso il nome utente e la password non devono essere codificati come nel caso dell'SMTP, ma scritti in formato ASCII)
- LIST serve per richiedere l'elenco dei messaggi presenti nella mail box,
- STAT il server risponde con il numero dei messaggi e lo spazio occupato,
- RETR scarica un messaggio, occorre indicare il numero del messaggio che si vuole recuperare subito dopo il comando,
- DELE contrassegna il messaggio per indicare al server di eliminarlo, occorre indicare il numero del messaggio che si vuole eliminare subito dopo il comando,
- QUIT chiude la connessione.

Esercizio 5 – ricezione manuale di una mail con POP3

Anche in questo caso proviamo ad utilizzare Putty per ricevere i messaggi di posta, sempre utilizzando TELNET indichiamo il nome del server e la porta.



Ora inviamo in sequenza i comandi per l'autenticazione, il comando LIST, il comando STAT, ed il comando RETR seguito dal numero del messaggio che si vuole recuperare. L'immagine di seguito non contiene l'intera risposta del server al comando RETR, perché troppo lunga.

```
USER postacchini.daniele@libero.it
+OK
PASS [redacted]
+OK Logged in.
LIST
+OK 2 messages:
1 4083
2 4015
.
STAT
+OK 2 8098
RETR 1
+OK 4083 octets
Return-Path: <postacchini.daniele@istitutomontani.edu.it>
Delivered-To: postacchini.daniele@libero.it
Received: from dcd-18 ([10.103.10.24])
        by dcbackend-24.iol.local with LMTP id OIMvAT7o9mHgJAIAKhIJZg
        for <postacchini.daniele@libero.it>; Sun, 30 Jan 2022 20:34:22 +0100
```

Dopo aver recuperato il messaggio, con il comando **DELE** indicheremo al server di cancellarlo dopo la chiusura della connessione. Se proviamo a rientrare dopo il comando **QUIT** non troveremo più il messaggio scaricato.

```
DELE 1
+OK Marked to be deleted.
QUIT
```

Provare a ripetere le operazioni catturando i pacchetti su wireshark ad ogni comando. Per visualizzare i pacchetti mettere il seguente filtro di visualizzazione tcp.port == 110.

PROTOCOLLO IMAP (Internet **A**ccess **M**essage **P**rotocol**)**

Questo protocollo è definito dalla RFC 2060, a differenza del POP3 che permette l'accesso di un solo client, l'IMAP consente l'accesso da client diversi. Utilizza la porta 143, e per le connessioni sicure crittografate la porta 993.

In questo protocollo la ricezione del messaggio prevede 4 fasi:

1. Instaurazione della connessione con il server,
2. Risposta del server con un messaggio di benvenuto,
3. Autenticazione e gestione delle mail da parte del client,
4. Termine della connessione, o timeout.

Durante la connessione ci possiamo trovare nei seguenti 4 stati possibili:

1. **Non Authenticate**; appena si avvia la connessione siamo in questo stato,
2. **Authenticated**; saremo in questo stato dopo aver inserito il comando di autenticazione con nome utente e password,
3. **Selected**; saremo in questo stato dopo aver scelto su quale casella di posta lavorare;
4. **Logout state**; saremo in questo stato dopo aver richiesto la chiusura della connessione.

A seconda dello stato saranno possibili comandi differenti, i comandi vanno sempre preceduti da un tag composto da una stringa alfanumerica, come vedremo nell'esempio più avanti.

Comandi nello stato Non Authenticate:

- Authenticate: *indica il metodo da utilizzare per l'autenticazione,*
- STARTTLS richiede *la connessione sicura con il server,*
- Login *nome password* *invia nome e password al server per l'autenticazione,*

Comandi nello stato Authenticated:

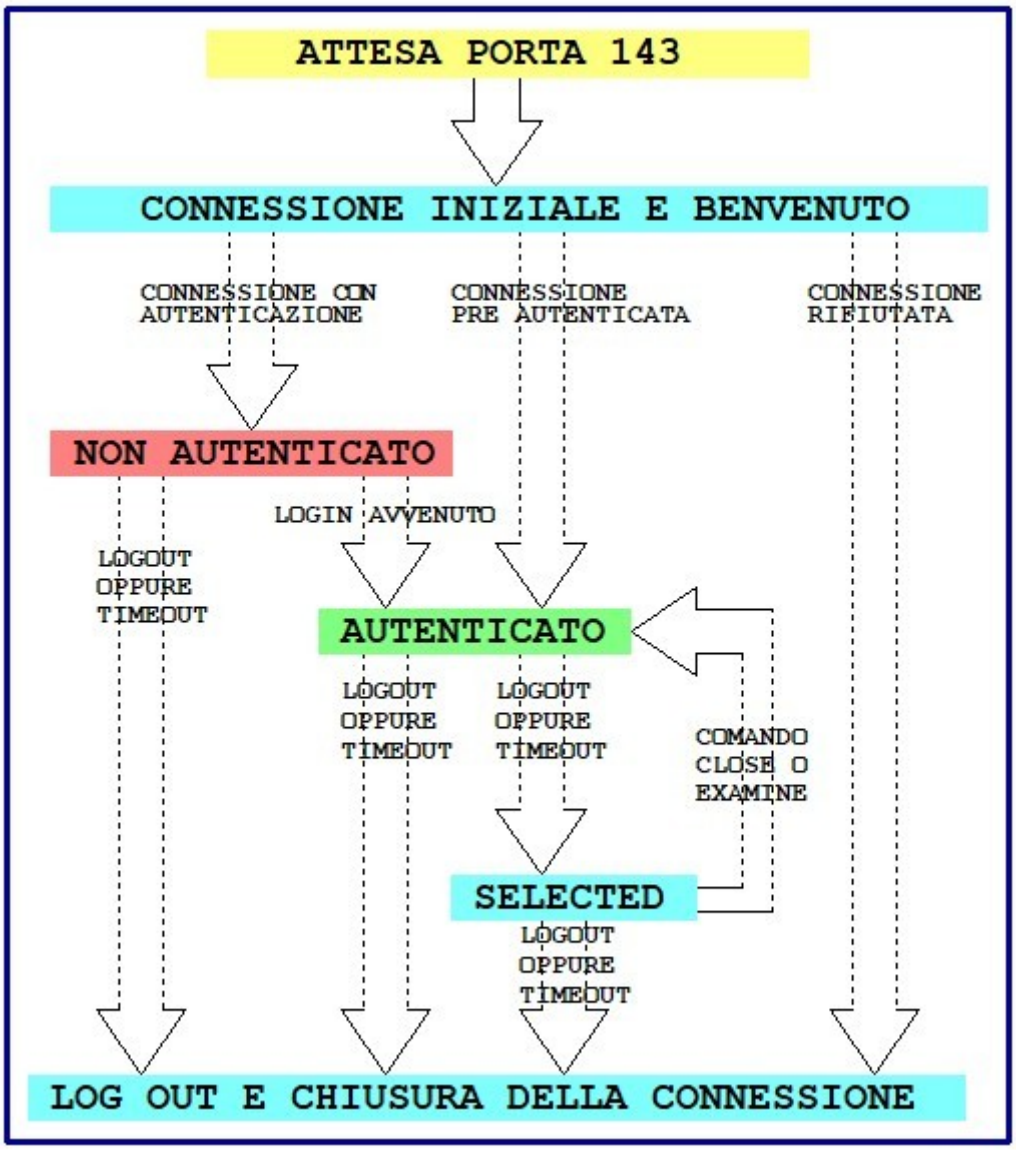
- Select *mailbox* *seleziona la mailbox su cui operare,*
- *Examine mailbox* *seleziona la mailbox con accesso in sola lettura,*
- Create *mailbox* *crea una mailbox,*
- Delete *mailbox* *cancella la mailbox,*
- Rename *nome_mailbox nome_nuovo_mailbox* *rinomina la mailbox,*
- Subscribe *mailbox* *aggiunge la mailbox alla lista delle caselle da visualizzare con il client,*
- Unsubscribe *mailbox* *rimuove la mailbox alla lista delle caselle da visualizzare con il client.,*
- List *mailbox sottocartella* *mostra l'elenco delle mailbox e cartelle,*
- LSUB *mailbox sottocartella* *come il precedente ma solo con le cartelle che il client può visualizzare (Subscribe)*
- Status *mailbox stato* *richiede lo stato della mailbox, oltre alla mailbox bisogna indicare cosa richiedere con il campo "stato". Questo campo può valere ad esempio i seguenti valori; MESSAGES (per chiedere il numero dei messaggi) RECENT (per richiedere il numero dei messaggi identificati come recenti) UNSEEN (per chiedere i messaggi identificati come non letti)*
- Append *mailbox messaggio* *richiede di aggiungere un messaggio alla mailbox.*

Comandi nello stato Selected:

- Check *controllo del server IMAP,*
- Close *chiede di tornare nello stato Authenticated e rimuove i messaggi identificati da cancellare,*
- *Expunge* *rimane nello stato Selected e rimuove i messaggi identificati da cancellare,*
- Search *parametri* *ricerca i messaggi che soddisfano i parametri di ricerca, i parametri di ricerca possibili sono diversi, riferirsi alla RFC2060,*
- Fetch *messaggi dati* *visualizza i dati richiesti di un messaggio, anche qui i dati possibili da richiedere sono diversi, riferirsi alla RFC2060,*
- Store *messaggi dati* *cambia i dati (flag di stato) di un messaggio,*
- Copy *messaggi mailbox* *copia un messaggio in una mailbox,*
- Uid *comandi argomenti* *indica al server di utilizzare gli Uid (Identificatori univoci) anziché il numero dei messaggi.*

In ogni stato sono comunque disponibili i seguenti comandi: **Capability** per elencare le funzioni del server, **Noop** (no operation) e **Logout** per terminare la connessione. Per i chiarimenti dei comandi visti sopra riferirsi alla RFC2060, https://www.rfc-editor.org/rfc/pdfrfc/rfc2060.txt.pdf

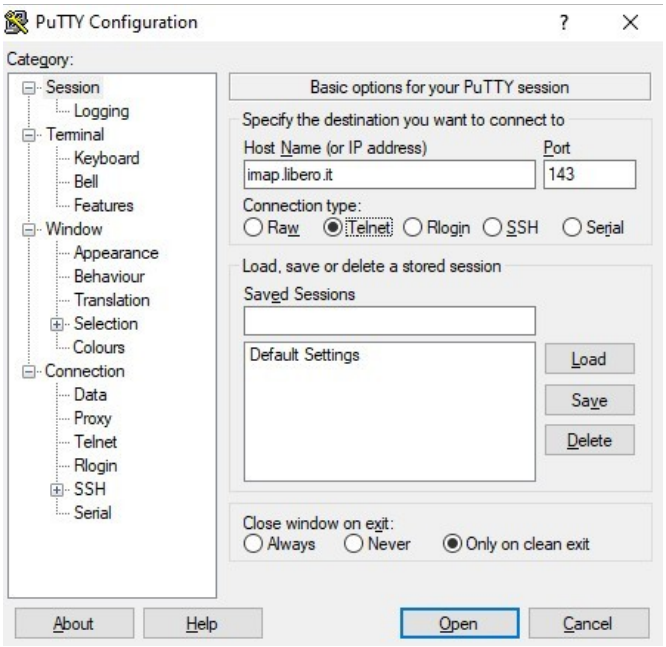
Lo schema del passaggio tra stati può essere riassunto con la seguente immagine:



Esercizio 6– ricezione manuale di una mail con IMAP

Utilizzando Putty effettuare una connessione Telnet al server imap.libero.it porta 143.

Inviare una mail alla propria casella di posta e tramite i comandi IMAP leggere l'oggetto ed il corpo, nella prossima pagina un esempio.



Nell'esempio di seguito, effettuiamo un'autenticazione con il comando LOGIN, successivamente selezioniamo la mail box di ingresso con il comando SELECT, poi con due comandi FETCH, richiediamo l'oggetto ed il corpo del messaggio della email numero 1.
Ogni comando è preceduto da stringhe alfanumeriche differenti (a01, a02, a03 ed a04).
Al termine la sessione si chiude con il comando LOGOUT.

```
* OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE IDLE AUTH=
PLAIN AUTH=LOGIN] Dovecot ready.

a01 Login postacchini.daniele@libero.it [redacted]
a01 OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE IDLE SOR
T SORT=DISPLAY THREAD=REFERENCES THREAD=REFS THREAD=ORDEREDSUBJECT MULTIAPPEND U
RL-PARTIAL CATENATE UNSELECT CHILDREN NAMESPACE UIDPLUS LIST-EXTENDED I18NLEVEL=
1 CONDSTORE QRESYNC ESEARCH ESORT SEARCHRES WITHIN CONTEXT=SEARCH LIST-STATUS BI
NARY MOVE SNIPPET=FUZZY SEARCH=X-MIMEPART XDOVECOT NOTIFY METADATA SPECIAL-USE Q
UOTA] Logged in
a02 Select inbox
* FLAGS (\Answered \Flagged \Deleted \Seen \Draft $smartnletter $offerte $dem_in
box)
* OK [PERMANENTFLAGS (\Answered \Flagged \Deleted \Seen \Draft $smartnletter $of
ferte $dem_inbox \*)] Flags permitted.
* 1 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1615665295] UIDs valid
* OK [UIDNEXT 175] Predicted next UID
* OK [HIGHESTMODSEQ 324] Highest
a02 OK [READ-WRITE] Select completed (0.002 + 0.000 + 0.001 secs).
a03 Fetch 1 (Flags body[header.fields (subject)])
* 1 FETCH (FLAGS (\Seen) BODY[HEADER.FIELDS (SUBJECT)] {23}
Subject: prova imap

)
a03 OK Fetch completed (0.001 + 0.000 secs).
a04 Fetch 1 (Flags body[text])
* 1 FETCH (FLAGS (\Seen) BODY[TEXT] {254}
--000000000000dceade05d6e4325f
Content-Type: text/plain; charset="UTF-8"

corpo del messaggio

--000000000000dceade05d6e4325f
Content-Type: text/html; charset="UTF-8"

<div dir="ltr">corpo del messaggio</div>

--000000000000dceade05d6e4325f--
)
a04 OK Fetch completed (0.001 + 0.000 secs).
```

RISPOSTE ALLA RICHIESTA