Abbiamo visto alcuni aspetti del protocollo SMTP (porta 25, protocollo di tipo asincrono best effort, basato sul TCP). Dal punto di vista applicativo la limitazione più stringente è il limite sulla lunghezza massima delle righe che costituiscono il messaggio (che è 72 caratteri, dopo i quali ci deve essere per forza l’a capo). L’altra limitazione è il poter usare solo i caratteri ASCII a 7 bit (motivo per cui esiste il MIME che traduce i file da allegare in ascii).

Per cercare di aggirare questa limitazione al protocollo SMTP è stato introdotto il metodo di codifica MIME, la cui descrizione si trova negli RFC dal numero 2045 (e segue in altri RFC con la descrizione di alcuni dettagli ulteriori). Lo standard MIME permette di codificare sequenze di byte usando una codifica, che è la RADIX 64; questa codifica è un insieme di caratteri (di cardinalità 64) che mi permette di scrivere dei numeri in base 64 (con un carattere per ogni cifra in base 64, le cifre decimali, le lettere dell’alfabeto maiuscole e minuscole + alcuni altri caratteri per arrivare a 64). Si tratta dunque di un sottoinsieme degli ASCII stampabili a 7 bit.

La codifica avviene in questo modo: se io ho una sequenza di byte da inviare, questi byte saranno composti da 8 bit ciascuno, e saranno in uno STREAM su una connessione TCP (Stabilita dal protocollo SMTP). Questa sequenza di byte sarà convertita in RADIX 64, suddividendo lo stream di bit in sottogruppi di 6 bit (6 bit = 2^6 opzioni = numeri in base 64) consecutivi, facendo corrispondere 24 bit (3 byte) a 4 caratteri RADIX in base 64 (3 byte corrispondono a 4 caratteri, che sono scritti come testo del messaggio). Le sequenze di caratteri RADIX saranno lunghe 72 caratteri, in quanto questa è la limitazione delle righe del messaggio. Quando il messaggio arriva a destinazione, la decodifica avviene in direzione opposta (4 caratteri vengono convertiti in 3 byte). In questo modo si può descrivere un trasferimento di dati in forma binaria a gruppi di 24 bit.

Il problema di questo tipo di codifica è che causa un’espansione nel messaggio di posta (se vogliamo mandare x kb, le dimensioni del payload del messaggio sono di 4/3 x ). Questo tipo di codifica però permette di allegare qualsiasi file a un messaggio di posta elettronica (anche se il trasferimento è inefficiente).

Dal punto di vista di chi vuole inviare il messaggio le complicazioni finiscono qui, tuttavia, chi vuole ricevere deve poter sapere a priori come interpretare la rappresentazione binaria dell’attachment (è un file audio, è un file che contiene un’immagine…? Chi lo sa): per facilitare il compito di chi riceve, lo standard prevede che prima della codifica venga posta un’intestazione che specifica che tipo di file si sta trasmettendo in versione codificata. Nello Standard MIME sono quindi stati definiti dei tipi di file, specificati come stringhe di caratteri divise in due parti (prima parte: classe di file, poi c’è una barra e poi il tipo specifico di file; esempio: text/ascii, questo esempio ha poco senso, ce l’ha solo se vogliamo inviare messaggi con righe poco più lunghe di 72 caratteri. C’è la categoria Image con i tipi gif, png, jpeg… Image/gif). Grazie a questa intestazione, il ricevente è in grado di decodificare correttamente i file in allegato.

Se si mette un padding, allora bisogna dire anche al ricevente qual è la lunghezza effettiva del file (che non è necessariamente un multiplo di 3 byte, potremmo dover aggiungere dei bit inutili di padding per renderlo tale): dunque nell’intestazione, sotto forma di metadato, è presente anche la lunghezza effettiva del file espressa in byte.

Quello che noi siamo abituati a usare è un client di posta elettronica che usa contemporaneamente i protocolli SMTP e MIME (senza che noi notiamo nulla), quando noi scriviamo un messaggio possiamo anche metterci i caratteri speciali, che vengono convertiti a nostra insaputa in formato RADIX 64 quando inviamo la mail.

Quando un client invia un messaggio, lo fa collegandosi via stream su porta 25 a un Server, utilizzando il protocollo SMTP e un socket di tipo TCP. Quando il server riceve il messaggio (dopo che il mittente si è autenticato, per evitare lo spam) il server controllerà l’indirizzo e-mail del destinatario, traducendo la parte dopo la @ in un indirizzo IP (tramite servizio DNS), il messaggio verrà quindi inviato al server specificato da tale IP. Nonostante si usi il protocollo TCP, esiste la possibilità che il messaggio resti sempre sul server intermedio e non arrivi mai a quello finale (quindi il collegamento TCP garantisce al mittente che il messaggio sia arrivato al server intermedio e al server intermedio che sia arrivato al server destinazione, ma dal server destinazione non viene inviata alcun acknowledgement al client mittente, questo anche perché il protocollo SMTP è asincrono, dunque il mittente potrebbe aver spento il pc subito dopo l’invio del messaggio). Questo è il motivo per cui l’SMTP è un protocollo Best Effort: perché fa più tentativi per inviare il messaggio, ma se fallisce troppe volte invia al mittente la notifica di fallimento nell’invio del messaggio.

Se il messaggio arriva fino al server destinazione, questo può copiare il contenuto del messaggio nella mail-box del destinatario (usando la parte prima della @ dell’indirizzo email suppongo?). Se il destinatario Effettua il log-in nel server destinatario allora potrà tranquillamente accedere alla sua mail-box e leggere il messaggio.

Per facilitare la ricezione da parte del destinatario sono stati introdotti altri protocolli di comunicazione, anche se il destinatario non effettua il log-in sul server. I protocolli in questione sono il protocollo POP e il protocollo IMAP; lo scopo di questi due protocolli è poter accedere alla mail-box senza dover fare il login sul server SMTP ma attraverso l’uso di un’altra macchina client, che si connette al server usando uno di quei due protocolli.   
La versione più semplice è quella del POP (Post Office Protocol), la versione più recente di questo protocollo è la versione 3, definita nell’RFC 1939. Per utilizzarlo è ovviamente necessario che il server-mailbox oltre al server SMTP sulla porta 25 abbia attivo anche un server POP sulla porta 110. Quando il client si autentica collegandosi alla porta 110 (dando lo username = indirizzo email e la password, definita sul server mail-box), connettendosi esattamente come se fosse loggato al server SMTP. Una volta Loggato può dare diversi comandi, come vedere tutti i messaggi ricevuti e/o scaricare uno o più (anche tutti) di questi messaggi. Usando il POP il contenuto di ciò che si vuole scaricare sarà riportato nel client POP, effettuando così una copia dalla mail-box principale alla mail-box secondaria. A questo punto il POP 3 prevede due opzione possibili: la prima è che quando si scarica un messaggio dal server, il messaggio originale viene cancellato (rimane una sola copia, quella del client), la seconda opzione, che si ha inviando un comando al server POP, è di scaricare la copia ma non cancellare il messaggio originale (in tal caso si hanno due copie del messaggio, una sul server e una sul client).

Se non cancello il messaggio dopo aver scaricato i dati sul client ho vantaggi e svantaggi: lo svantaggio è che prima o poi si riempirà la mail-box sul server (e quindi ne dovrò cancellare per poter tornare a ricevere i messaggi), ma il vantaggio è che se per caso viene persa la copia del messaggio sul client POP, rimane il Back-up del messaggio sul server.

Le limitazioni del POP 3 sono, oltre alle limitazioni del SMTP (aggirabili con l’estensione MIME), abbiamo la limitazione di poter solo scaricare interamente o non scaricare affatto un messaggio (non si ha la possibilità di vedere i metadati, come subject, mittente ecc. del messaggio). Quindi per poter sapere se un messaggio era interessante o no, andava per forza scaricato prima: ai tempi in cui la banda di connessione non era particolarmente larga poteva essere frustrante aspettare diversi minuti per poi scoprire che il messaggio scaricato era di SPAM. Per questo motivo è stata introdotta la possibilità di indicare al POP quanti byte del messaggio scaricare (quindi scaricando i primi 1000 byte si poteva avere un’idea del subject del messaggio e si poteva decidere se lo si voleva scaricare tutto) tramite un estensione opzionale (poteva pure capitare che ci fossero server che non riconoscevano questa estensione).

Per ovviare a queste problematiche si è introdotto il protocollo IMAP, la cui versione attuale è la numero 4.2 ed è descritta dall’RFC 9051.  
L’idea è che, anziché scaricare una copia del file mail-box su una macchina locale, con questo protocollo facciamo sempre riferimento alla copia contenuta sul server. La porta usata dai server IMAP è la 143 e il protocollo permette di usare un meccanismo di copia dei dati che utilizza una tecnica di caching (tenendo una copia in locale delle ultime cose viste). Il protocollo è realizzato in modo da supportare l’accesso di più client alla stessa mail-box (quindi se due client si collegano alla stessa mail-box e uno dei due cancella un messaggio, quel messaggio sarà stato cancellato anche per l’altro, oppure se uno dei due legge un messaggio, l’altro vedrà tale messaggio come “messaggio già letto”).   
Questo permette di avere l’accesso alla mail-box per messaggi di posta elettronica tramite macchine che non sono permanentemente connesse alla rete; si affida tutto a dei server che sono perennemente connessi a Internet e poi ci si connette con quante macchine si vuole ai server contenenti la nostra mail-box.

--Nota: entrambi i protocolli appena descritti usano il protocollo TCP al livello 4 (anche se esistono versioni SICURE dei protocolli che usano anche la cifratura dei dati per mantenere la riservatezza delle informazioni)—

Se siamo interessati a mantenere riservati i contenuti dei messaggi durante tutta la trasmissione si usa una cifratura end-to-end (in questo modo solo il destinatario che ha la chiave può decifrarlo).  
Un altro aspetto legato alla sicurezza dei sistemi è quello di proteggere le credenziali usate dagli utenti, in questo caso la cifratura non è end-to-end, ma è su un canale tra client e server (e in tal caso tale tecnica di cifratura è implementata sulla versione sicura dell’IMAP e del POP, in tal caso si ha la versione TLS dei protocolli).