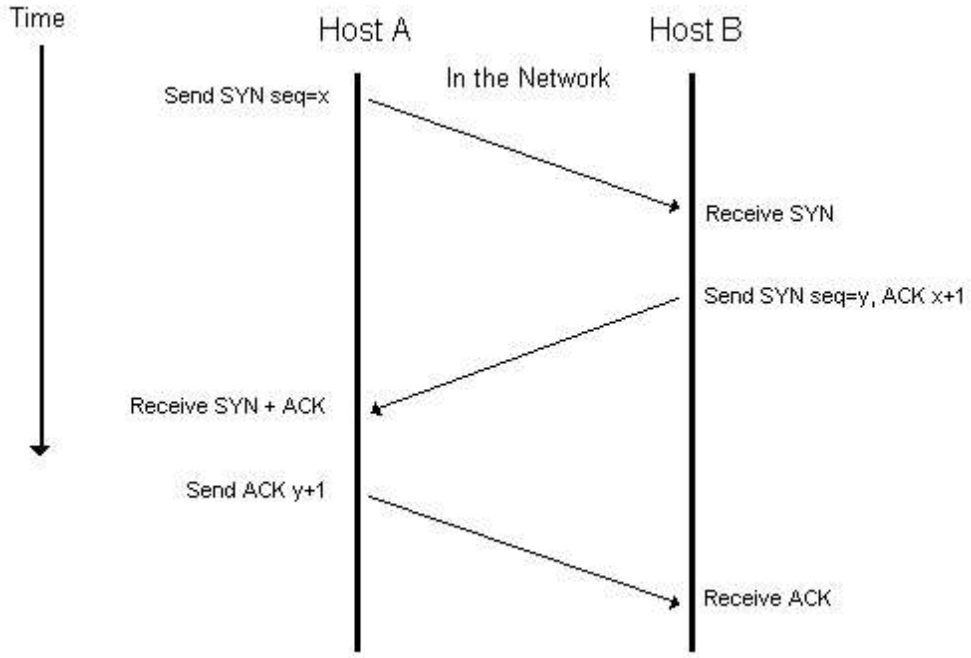
Approfondimento TCP

- TCP è un protocollo **orientato alla connessione**, ovvero prima di poter trasmettere dati deve stabilire la comunicazione, negoziando una connessione tra mittente e destinatario, che viene esplicitamente chiusa quando non più necessaria. Esso quindi possiede le funzionalità per creare, mantenere e chiudere/abbattere una **connessione**.
- Il servizio offerto da TCP è il trasporto di un **flusso di byte** bidirezionale tra due applicazioni in esecuzione su host differenti. Il protocollo permette alle due applicazioni di trasmettere contemporaneamente nelle due direzioni, quindi il servizio può essere considerato "**FULL-Duplex**" anche se non tutti i protocolli applicativi basati su TCP utilizzano questa possibilità.
- Il *flusso di byte* viene **frazionato** in blocchi per la trasmissione dall'applicazione a TCP (che normalmente è implementato all'interno del sistema operativo), per la trasmissione all'interno di *segmenti* TCP, per la consegna all'applicazione che lo riceve, ma questa divisione in blocchi non è necessariamente la stessa nei diversi passaggi.
- TCP garantisce che i dati trasmessi, se giungono a destinazione, lo facciano in ordine e una volta sola ("**at most once**"). Più formalmente, il protocollo fornisce ai livelli superiori un servizio equivalente ad una connessione fisica diretta che trasporta un flusso di byte. Questo è realizzato attraverso vari meccanismi di ***acknowledgment*** e di ritrasmissione su ***timeout***.
- TCP offre funzionalità di **controllo di errore** sui pacchetti pervenuti grazie al campo ***checksum*** contenuto nella sua PDU (**Protocol Data Unit**).
- TCP possiede funzionalità di ***controllo di flusso*** (facendo mantenere al mittente una variabile chiamata **finestra di ricezione** - *receive window* - che fornisce un'indicazione sullo spazio libero disponibile nel buffer del destinatario) tra terminali in comunicazione e ***controllo della congestione*** sulla connessione, attraverso il meccanismo della *finestra scorrevole*. Questo permette di ottimizzare l'utilizzo della rete anche in caso di congestione (quando il traffico offerto alla rete è vicino o superiore alla capacità della rete), e di condividere equamente la capacità disponibile tra diverse sessioni TCP attive su un collegamento.
- TCP fornisce un servizio di *multiplazione* delle connessioni su un host, attraverso il meccanismo delle porte.



Dimensioni consigliate dei segmenti (TPDU – Transport Protocol Data Unit)

È necessario definire la lunghezza del segmento oltre che **in funzione delle capacità di trasmissione del mittente e di ricezione del destinatario**, anche e soprattutto **in funzione delle caratteristiche della rete**, come per esempio la grandezza massima del frame fisico, o Maximum Transfer Unit (MTU).

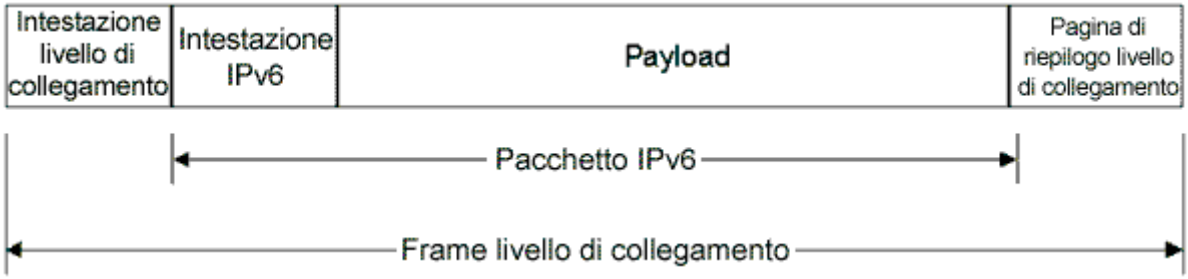
Valori della MTU (maximum transfer unit) per una serie di reti diverse.

Rete	MTU
Hyperlink	65535
Token Ring IBM (16 Mbit/sec)	17914
Token Ring IEEE 802.5 (4 Mbit/sec)	4464
FDDI	4532
Ethernet	1500
X.25	576

La MTU più piccola fra due stazioni viene in genere chiamata *path MTU*, che dice quale é la lunghezza massima oltre la quale un pacchetto inviato da una stazione ad un'altra verrebbe senz'altro frammentato. Si tenga conto che non è affatto detto che la path MTU sia la stessa in entrambe le direzioni, perché l'instradamento può essere diverso nei due sensi, con diverse tipologie di rete coinvolte.

Il protocollo TCP definisce una lunghezza massima di un segmento o *maximum segment size* MSS (**Maximum Segment Size**) che annuncia all'altro capo della connessione la dimensione massima del segmento di dati che può essere ricevuto, così da evitare la frammentazione. E' indipendente dalla semantica (un indirizzo può essere spezzato in più segmenti) e di norma viene impostato alla dimensione della MTU dell'interfaccia meno la lunghezza delle intestazioni di IP e TCP se entrambi gli estremi della connessione si trovano nella stessa rete fisica, altrimenti lo standard raccomanda di utilizzare un valore di **536 byte**, equivalente alla dimensione normale di un datagramma IPv4 meno le dimensioni standard delle intestazioni IP e TCP sommate insieme, 40 byte appunto.

Nel caso della più attuale versione 6 (IPv6) si richiede un livello di Data Link che supporti una dimensione minima del pacchetto IPv6 di 1280 byte (con 40 byte di intestazione IP) e si consiglia una dimensione MTU a 1500 byte (tipica d'incapsulamento ETHERNET II) preferendo collegamenti con MTU configurabile[1].



[1] Un esempio di collegamento con MTU configurabile è il PPP con Maximum Receive Unit

Il 3-Way Handshaking

La s**essione** TCP si **apre** attraverso il metodo di Handshake a tre vie. Ogni segmento del protocollo TCP è numerato, e precisamente ogni segmento contiene:

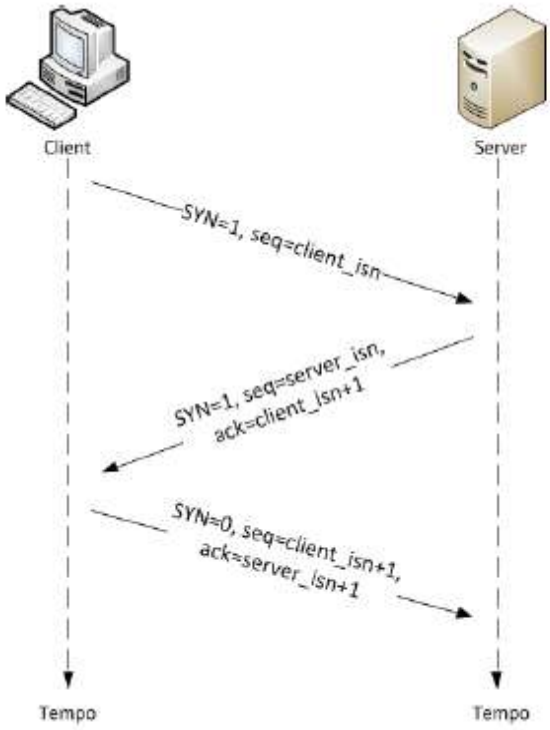
- Il numero del pacchetto (SEQN)
- Il numero dell'ultimo pacchetto ricevuto dall'host mittente aumentato di uno (ACKN)

Ogni host tiene quindi in memoria due contatori che contengono il numero del successivo pacchetto da inviare e quello da ricevere.

Il 3-Way Handshaking serve a sincronizzare i due contatori fra gli host.

Esso funziona in questo modo:

- L'host mittente invia un pacchetto con il flag SYN impostato su 'on'.
- Il ricevente riceve la richiesta e risponde a sua volta con un pacchetto con il flag di sincronizzazione SYN 'on', il numero di sequenza per i segmenti da inviare, il numero di sequenza per quelli da ricevere.
- L'host mittente, ricevuti questi pacchetti, risponde con un segmento contenente i due numeri sequenziali dei segmenti.



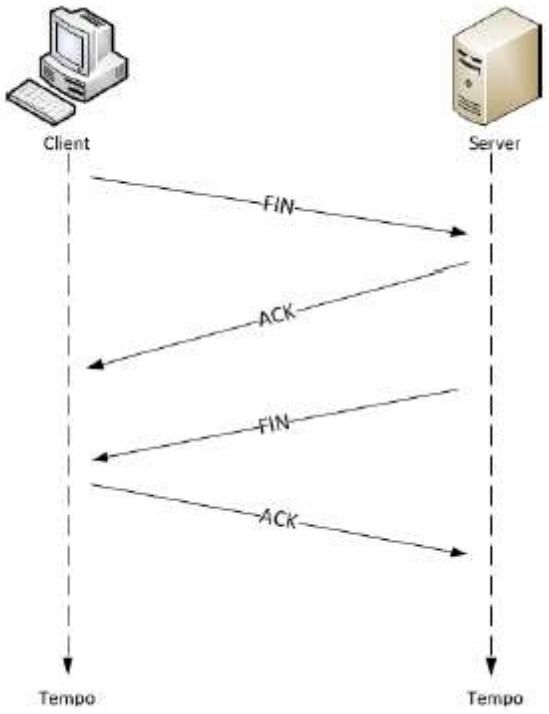
In modo analogo il TCP utilizza questo sistema per terminare **la connessione** assicurandosi che la trasmissione dei dati sia realmente terminata.

In sintesi: se in risposta al Syn arriverà un Syn/Ack, sapremo che la porta è aperta e che qualcuno dall'altra parte è in ascolto. In caso contrario, riceveremo un Rst e quindi nessuna possibile porta cui collegarsi.

Questo metodo ha il vantaggio di essere estremamente semplice e rapido e nello stesso tempo è facilmente rilevabile. Infatti, alla ricezione del Syn/Ack di risposta, la funzione connect() manderà l'Ack che completerà la connessione, la cui traccia finirà nel log di qualsiasi daemon ci sia dall'altra parte, per cui, un eventuale hacker, a meno di mascherarsi attraverso un proxy socks, sarà già alla fine del viaggio nel tentativo di scan.

Chiusura della connessione TCP/IP: handshake a quattro vie (o a tre vie (FIN+ACK))

Fasi:

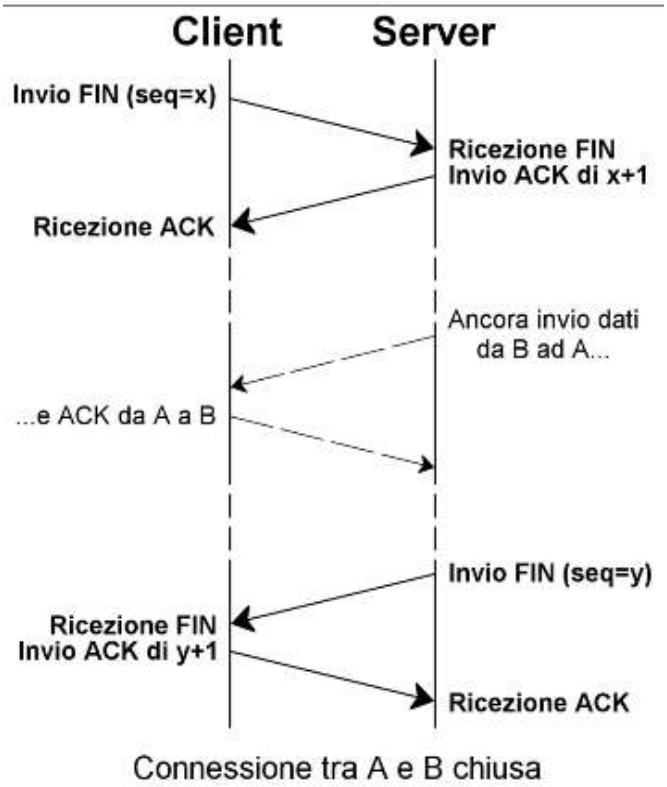


1.Il client invia al server un segmento con bit **FIN**=1, entra nello stato FIN_WAIT_1

2.Il server invia al client un riscontro, il client entra nello stato FIN_WAIT_2

3.Il server invia al client un segmento con il bit FIN=1

4.Il client riscontra il segmento ed entra nello stato TIME_WAIT. Dopo un tempo che varia dai 30 secondi ai 2 minuti la connessione viene conclusa



FIN - se settato a 1 indica che l'host mittente del segmento vuole *chiudere la connessione TCP* aperta con l'host destinatario. Il mittente attende la conferma dal ricevente (con un FIN-ACK). A questo punto la connessione è ritenuta *chiusa per metà*: l'host che ha inviato FIN non potrà più inviare dati, mentre l'altro host ha il canale di comunicazione ancora disponibile.

Quando anche l'altro host invierà il pacchetto con FIN impostato la connessione, dopo il relativo FIN-ACK, sarà considerata *completamente chiusa*.