Azzolini Riccardo

Reti e protocolli

1 Protocollo IP

Il servizio realizzato da **IP** (**Internet Protocol**) è la consegna di un **datagramma**, che è un pacchetto di bit contenente:

- i dati;
- le informazioni ausiliarie (talvolta chiamate *metadati*), tra cui, ad esempio, l'indirizzo del mittente e quello del destinatario.

In particolare, il protocollo IP fornisce un servizio **connectionless** (senza connessione) di trasmissione **non affidabile** di datagrammi (pacchetti). Esso *non* assicura:

- la consegna,
- l'integrità,
- la non-duplicazione,
- l'ordine di consegna

dei datagrammi.

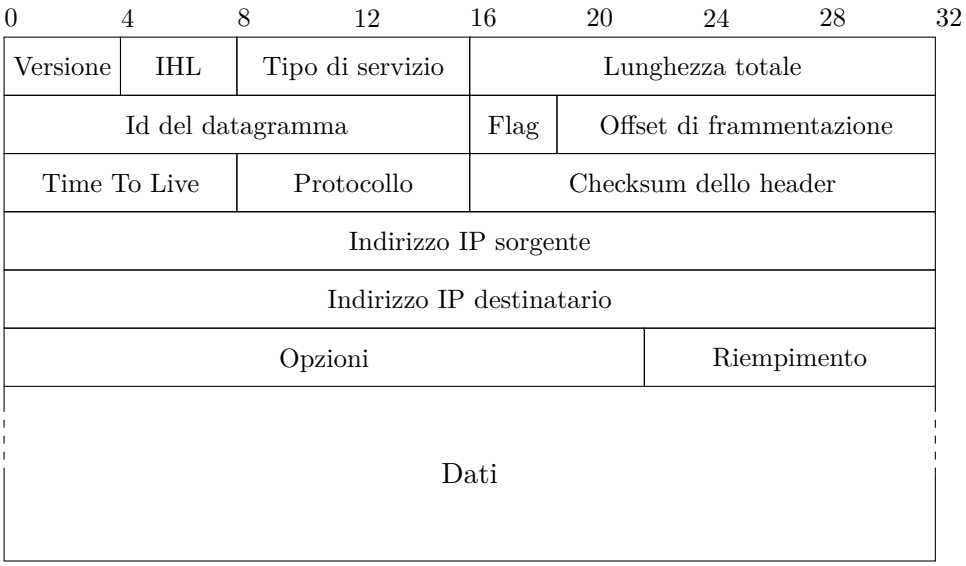
Per realizzare il proprio servizio, IP si può appoggiare a una varietà di protocolli di livello più basso, quali Ethernet, PPP, X.25, Frame Relay, ATM, ecc.

1.1 Elementi del protocollo

Il protocollo IP specifica il formato esatto di tutti i dati, l'insieme di regole che inglobano l'idea di consegna non affidabile, e fornisce le funzioni di instradamento (*routing*). Queste ultime sono realizzate in base agli indirizzi IP: ogni gateway dispone di opportune *tabelle di routing* che, se il destinatario non è direttamente connesso al gateway, permettono di determinare verso quale altra macchina instradare il messaggio in modo da farlo avvicinare alla destinazione, che raggiungerà mediante una serie di *salti* (*hop*).

1.2 Struttura del pacchetto

Un pacchetto / datagramma IP è costituito da un *header* (intestazione, che contiene le informazioni ausiliarie), seguito dai dati "veri e propri":



Alcune delle informazioni più importanti trasmesse nel datagramma sono:

- *Versione* (4 bit): ha il valore 4.[1]
- *Lunghezza totale* (16 bit): la dimensione complessiva del pacchetto (in byte), che è variabile, ma al massimo pari a 64k.
- *Id del datagramma* (16 bit): un identificatore univoco di ciascun pacchetto, che permette al ricevente di distinguerlo dagli altri (ad esempio, per determinare se è un duplicato).
- *Time To Live* (*TTL*, 8 bit): il numero massimo di hop (connessioni attraversate da un gateway al successivo) che il pacchetto può fare prima di essere considerato perso.
- *Protocollo* (8 bit): il protocollo incapsulato nella parte dati del pacchetto (ad esempio TCP). Questo permette di determinare a quale protocollo di livello superiore affidare i dati ricevuti tramite IP.
- *Checksum dello header* (16 bit): un codice che consente di rilevare errori nella trasmissione.

[1] Sebbene la versione 4 (IPv4) sia ancora la più utilizzata su Internet, esiste anche la versione 6 (IPv6), ormai largamente supportata: in tal caso, questo campo contiene il valore 6, ma il resto del pacchetto ha una struttura completamente diversa.

- *Indirizzi IP sorgente* e *destinatario* (32 + 32 bit): l'indirizzo del destinatario è indispensabile per il routing, mentre quello della sorgente serve alla realizzazione di protocolli di livello superiore, nei quali il destinatario deve conoscere l'indirizzo da cui arrivano i dati per poter rispondere (e/o richiedere altri dati).

Per essere trasmesso, tutto questo datagramma (sia l'header che i dati) viene **incapsulato** in un frame (trama) del protocollo di livello inferiore:

Poi, la macchina che riceverà il frame determinerà che esso contiene un datagramma IP, e si comporterà di conseguenza: in particolare, leggerà l'indirizzo di destinazione, per sapere se è il destinatario del messaggio o se invece lo deve inoltrare (secondo le tabelle di routing).

2 Protocollo UDP

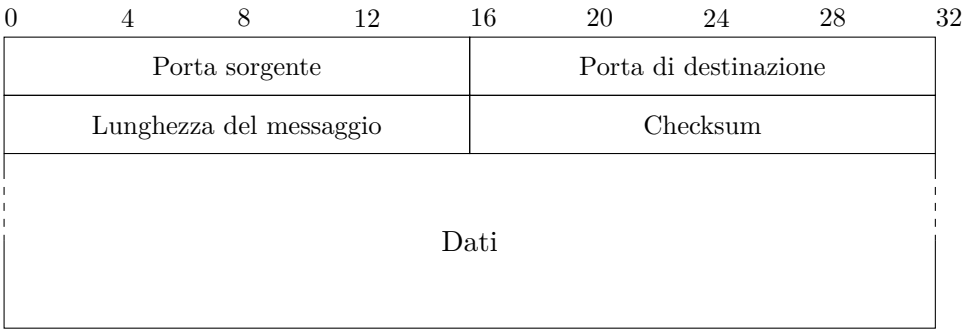
IP non consente di distinguere più destinazioni di datagrammi all'interno della stessa macchina, poiché considera solo il singolo indirizzo IP della macchina. Di per sé, questo imporrebbe una limitazione molto pesante: per ogni macchina, potrebbe comunicare sulla rete una sola applicazione alla volta.

Per superare questo limite, è stato definito **UDP** (**User Datagram Protocol**). Esso si appoggia al protocollo IP, dal quale eredita le caratteristiche del servizio di trasmissione dei pacchetti, che è **connectionless** e **non affidabile**. In aggiunta, però, UDP fornisce:

- un servizio (facoltativo) di protezione dagli errori di trasmissione;
- soprattutto, il concetto di **porta**, che permette di distinguere più sorgenti / destinazioni dei messaggi per uno stesso indirizzo IP.

2.1 Struttura del pacchetto

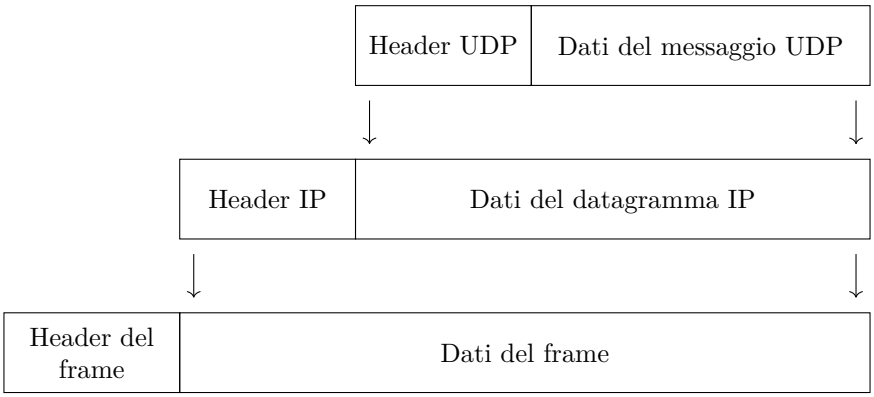
Il pacchetto / datagramma UDP ha la seguente struttura (anch'essa composta, come al solito, da un header seguito dai dati):



Oltre ai numeri di porta (che sono di 16 bit ciascuno, quindi permettono di distinguere fino a 2^{16} applicazioni all'interno di una macchina), l'header UDP contiene solo le seguenti informazioni di servizio:

- *lunghezza del messaggio* (16 bit), misurata in byte;
- *checksum* (16 bit), per il rilevamento di errori nella trasmissione (facoltativa: se non usata, questo campo contiene il valore 0).

Non è invece presente l'indirizzo IP, dato che esso è già presente nell'header del datagramma IP, all'interno del quale i pacchetti UDP vengono incapsulati per essere trasmessi:



3 Protocollo TCP

TCP (**Transmission Control Protocol**) è un protocollo **connection-oriented**: esso realizza una connessione **full-duplex** e **affidabile** tra due applicazioni, identificate da un indirizzo IP e un numero di porta.

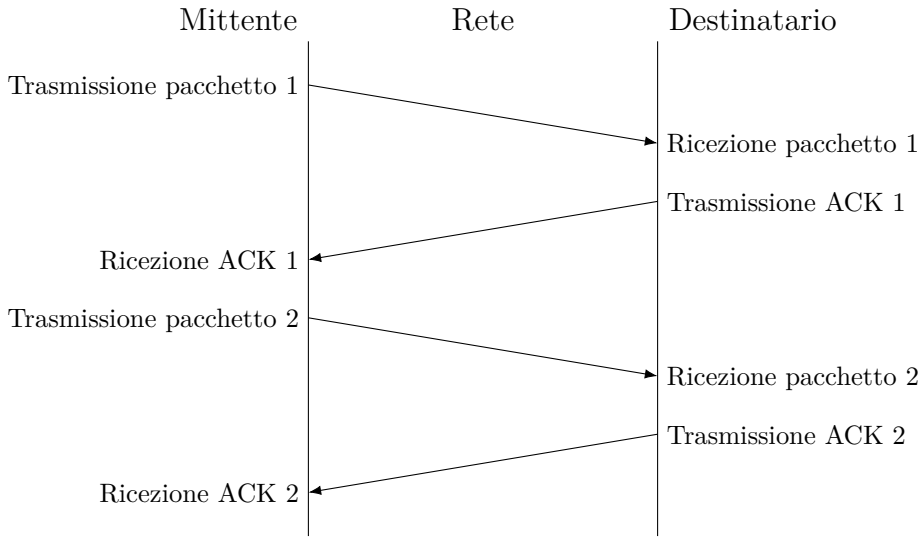
TCP costituisce l'infrastruttura di comunicazione della maggior parte dei sistemi basati su scambio di messaggi su Internet.

3.1 Affidabilità

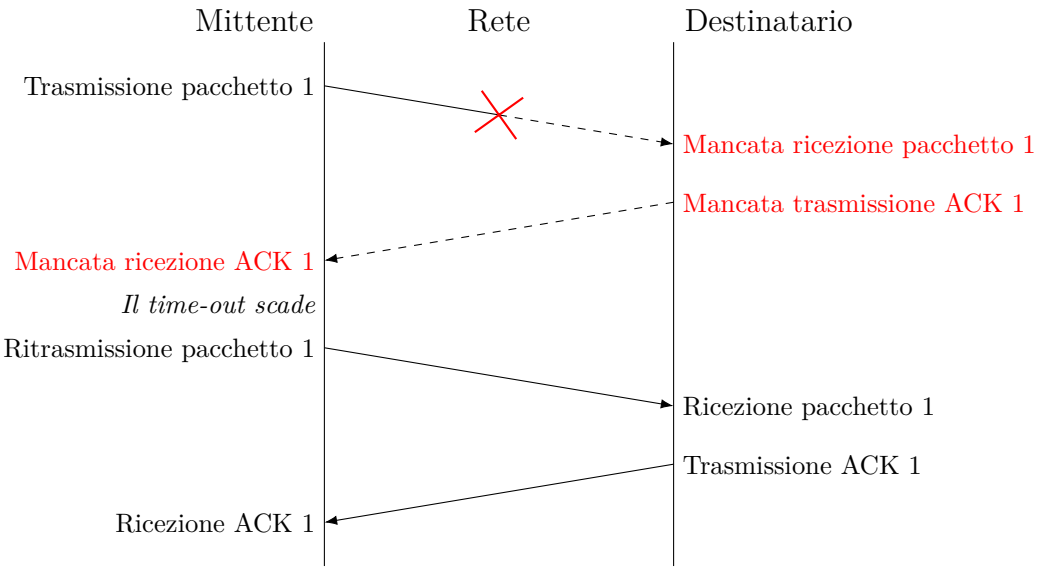
L'affidabilità di TCP è una delle sue caratteristiche fondamentali. Siccome il protocollo sottostante (IP) non è affidabile, per realizzare l'affidabilità TCP utilizza un meccanismo detto **PAR**: **Positive Acknowledgement with Retransmission** (riscontro positivo di ricezione con ritrasmissione).

Quando il destinatario riceve un messaggio, invia al mittente una conferma di ricezione (**acknowledgement**, ACK). Se il mittente non ottiene tale riscontro entro un certo tempo (time-out), deduce che il pacchetto sia andato perso, e lo ritrasmette. Questo procedimento si ripete finché non è confermata la ricezione del pacchetto da parte del destinatario.

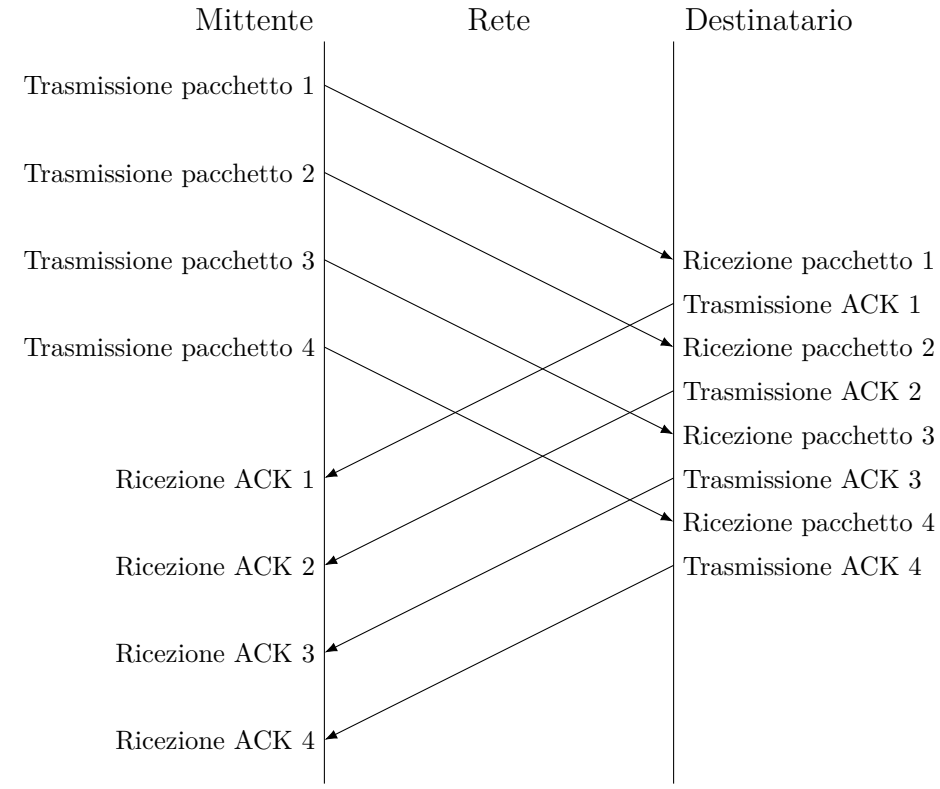
Una tipica sequenza di messaggi TCP è la seguente:



Invece, quello mostrato in seguito è un caso in cui si ha la perdita, e successiva ritrasmissione, di un pacchetto:



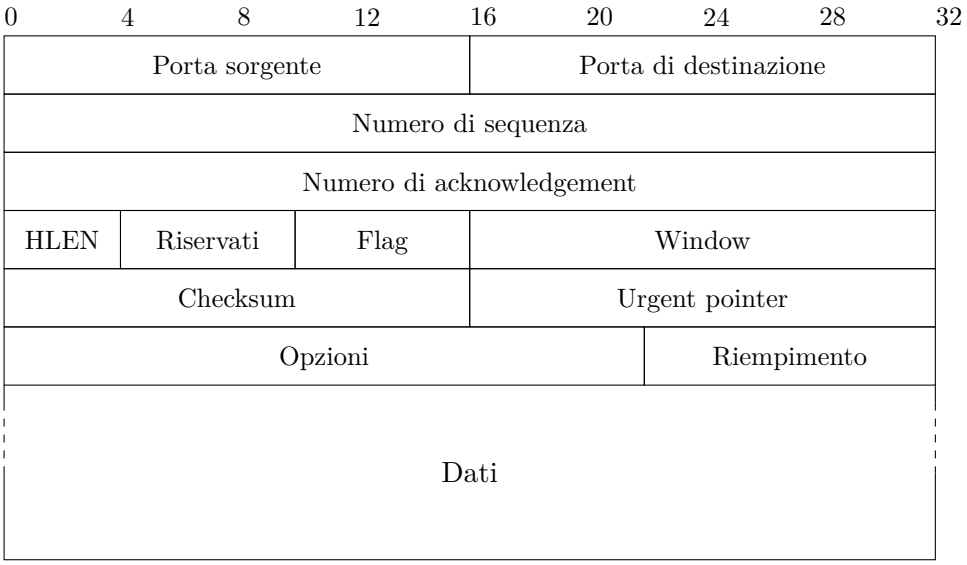
Per ottimizzare lo sfruttamento della rete, il mittente può inviare direttamente una sequenza di più pacchetti, e poi verificare che arrivi un acknowledgement (ACK) per ciascuno di essi, eventualmente ritrasmettendo quelli per cui ciò non avviene entro il time-out fissato:



Ogni riscontro indica anche il numero di byte che il ricevente è in grado di accettare. Esso serve al **controllo di flusso**: il ricevente può segnalare al trasmittente se si ha congestione sulla linea, e addirittura annullare la trasmissione di altri pacchetti, indicando di essere disposto a ricevere 0 byte. Senza un meccanismo del genere, se un pacchetto arrivasse in ritardo a causa della congestione, il mittente potrebbe pensare che esso si sia perso, e quindi ritrasmetterlo, ma ciò aumenterebbe ulteriormente la congestione sulla rete, in un circolo vizioso.

3.2 Struttura del pacchetto

Il pacchetto (segmento) TCP è composto da:

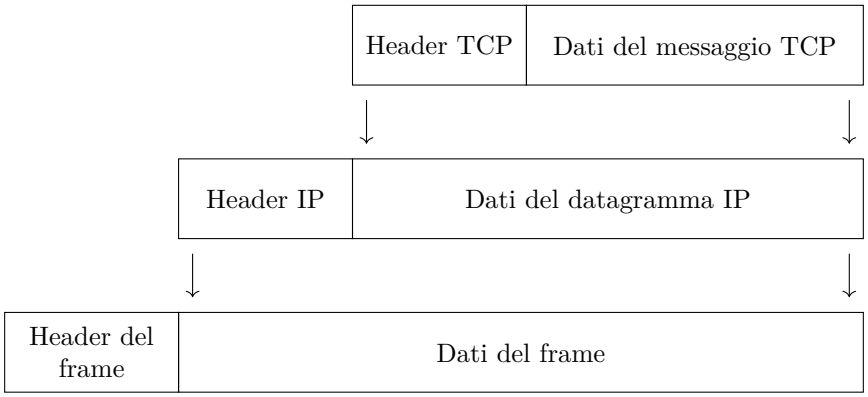


Le informazioni più importanti contenute nell'header sono:

- *Porta sorgente* e *porta di destinazione* (16 + 16 bit): i numeri di porta, analoghi a quelli presenti nel pacchetto UDP. La quaterna formata dagli indirizzi IP del pacchetto IP e dai numeri di porta del pacchetto TCP specifica univocamente il circuito virtuale a cui il pacchetto appartiene.
- *Numero di sequenza* (32 bit): indica la posizione dei dati del pacchetto nello stream di comunicazione, cioè nel messaggio (più grande) che è stato suddiviso in pacchetti per trasmetterlo.
- *Numero di acknowledgement* (32 bit): indica il numero del byte che il mittente di questo pacchetto si aspetta di ricevere dal destinatario nel prossimo pacchetto che quest'ultimo invierà. Ciò permette di determinare a quale pacchetto si riferisca l'acknowledgement.

- *HLEN* (4 bit): lunghezza dello header, come numero di parole di 32 bit.
- *Flag* (6 bit): usati per gestire le fasi di apertura e chiusura delle connessioni, e contrassegnare i messaggi che invece fanno parte della comunicazione vera e propria. Ciascuno dei flag (bit) contenuti in questo campo ha un proprio nome (SYN, ACK, FIN, ecc.).

Come UDP, anche il pacchetto TCP viene incapsulato in un datagramma IP per essere trasmesso:

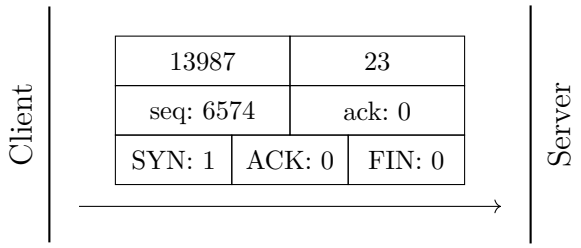


3.3 Sessione TCP

La sessione di comunicazione TCP prevede tre fasi: setup, scambio di dati, e shutdown.

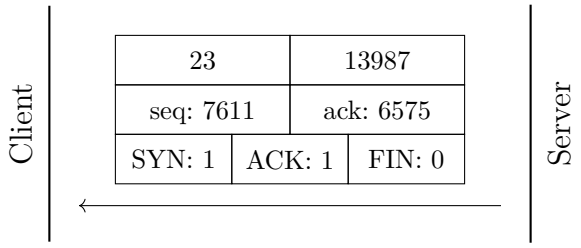
3.3.1 Setup

Un server, in ascolto su una determinata porta, riceve una richiesta di connessione da parte di un client. Tale richiesta è costituita da un segmento TCP che è marcato con il bit di sincronismo SYN, e contiene un numero casuale s_c come numero di sequenza.

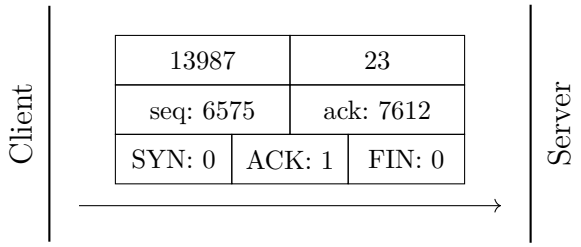


Il server risponde con un segmento, marcato sia con il bit di sincronismo SYN che con il bit di ACK (acknowledgement), nel quale:

- il numero di sequenza è un altro numero casuale, s_s;
- il numero di acknowledgement è il numero di sequenza del client, incrementato di uno ($s_c + 1$), per indicare che questo pacchetto è la risposta a quello inviato precedentemente dal client.



Il client manda un segmento avente numero di sequenza $s_c + 1$ e numero di acknowledgement $s_s + 1$, e contrassegnato con il bit di ACK.



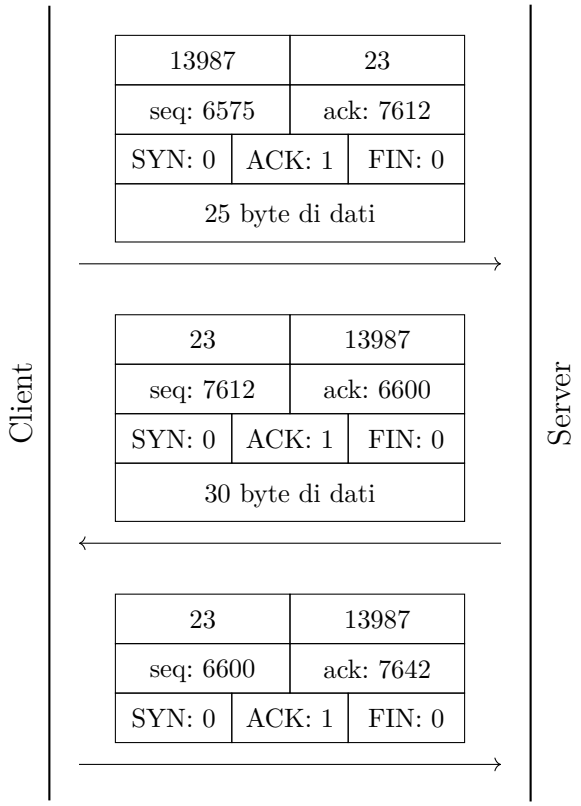
3.3.2 Scambio di dati

Una volta finita la fase di setup, si è instaurato un circuito virtuale, attraverso il quale avviene la comunicazione.

In ogni pacchetto, il client (e anche il server) inserisce:

- il proprio numero di sequenza, incrementato del numero di byte trasmessi in precedenza;
- l'acknowledgement del pacchetto precedente, dato dal numero di sequenza di tale pacchetto incrementato del numero di byte che esso conteneva (così, si conferma di aver ricevuto tutti i dati).

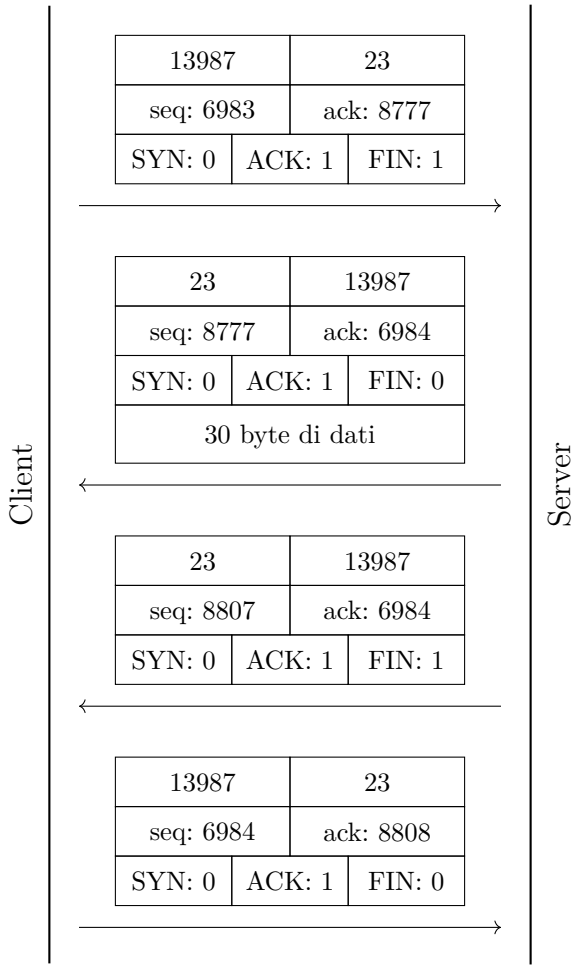
Ad esempio:



Inoltre, il ricevente indica una *finestra di ricezione*, cioè quanti pacchetti il mittente può spedire prima di dover aspettare di ricevere un acknowledgement. Questo serve a evitare che il ricevente venga "inondato" di una quantità di messaggi che non è in grado di gestire.

3.3.3 Shutdown

Il client (o il server) può indicare la fine della trasmissione con un pacchetto marcato dal bit di FIN. Il server (o il client) risponderà con un segmento di acknowledgement, e, prima o poi, indicherà che anch'esso ha finito di trasmettere, mandando a sua volta un pacchetto FIN, e ricevendo come risposta un ultimo ACK. Allora, il circuito virtuale verrà interrotto.



4 Porte riservate

In generale, i numeri di porta inferiori al 1024 sono considerati privilegiati, in quanto riservati a servizi standard (HTTP, FTP, Telnet, ecc.). Alcuni esempi di porte riservate sono:

Porta	Servizio	Descrizione
20	FTP-DATA	Connessione dati FTP
21	FTP-CONTROL	Connessione di controllo FTP
23	TELNET	Terminale remoto
25	SMTP	Simple Mail Transfer Protocol
80	WEBSERVER	Web server (HTTP)

5 Applicazioni distribuite

In generale, si può definire un'applicazione come un insieme di programmi coordinati per svolgere una data funzione. Allora, un'applicazione è distribuita se prevede più programmi eseguiti su calcolatori diversi, connessi tramite una rete. Un esempio sono le applicazioni web, che prevedono l'esistenza di un web server e dei web browser.

Le regole per la comunicazione in un'applicazione distribuita sono dette **protocollo applicativo** (ad esempio, il protocollo applicativo della navigazione web è HTTP, HyperText Transfer Protocol). Tale protocollo deve essere definito opportunamente, e adottato da tutti i programmi dell'applicazione.

I programmi applicativi utilizzano i protocolli (ad esempio TCP) mediante opportune interfacce (*API*, *Application Programming Interface*), fornite dal sistema operativo e dal software di rete. Come spesso capita, ci sono tante interfacce diverse per utilizzare le stesse funzionalità. Grazie all'astrazione fornita da tali interfacce, i programmatori possono sostanzialmente trascurare il funzionamento a basso livello dei protocolli di trasporto (e inferiori), e concentrarsi sul protocollo applicativo.