

## Livello trasporto

- ❑ Servizi
- ❑ Multiplexing e demultiplexing
- ❑ Trasporto senza connessione: UDP
- ❑ Principi di trasferimento affidabile
- ❑ TCP
  - Struttura del segmento
  - Gestione della connessione
  - Trasferimento dati affidabile
  - Controllo del flusso
- ❑ Controllo della congestione in TCP

## TCP (1/2)

- ❑ Definito negli RFC [793](#), [1122](#), 1323, ...
- ❑ *Point-to-point*: connessione tra un sender ed un receiver
- ❑ *Full-duplex*: flusso di dati bidirezionale nella stessa connessione
- ❑ *Connection-oriented*: handshake a tre vie prima dell'invio dei dati e chiusura della connessione
- ❑ *Pipelined*: controllo del flusso e della congestione regolano la finestra di trasmissione



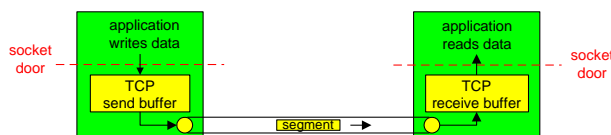
Trasferimento affidabile

## TCP (2/2)

- ❑ Basato su una versione del protocollo sliding window con dimensione variabile della finestra di trasmissione
- ❑ La PDU prende il nome di *segmento*
- ❑ La quantità massima di dati del segmento (*MSS*, *Maximum Segment Size*) è negoziata quando si attiva la connessione

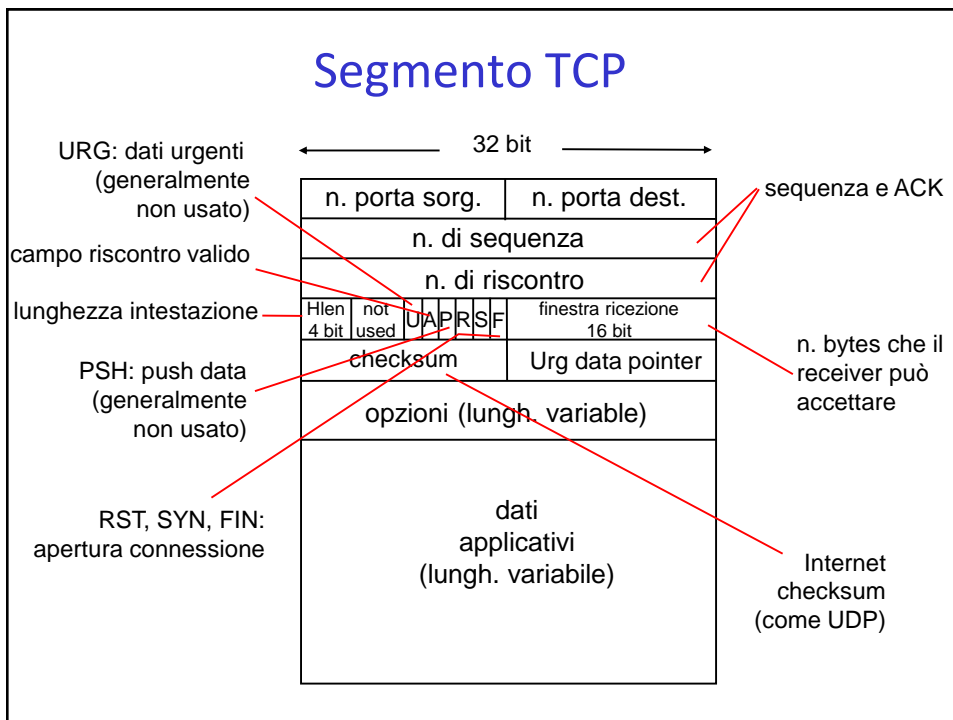
## TCP: buffer di ricezione/trasmissione

- ❑ I dati vengono accumulati in buffer sia in trasmissione sia in ricezione:
  - I dati vengono passati attraverso il socket ad un *buffer di trasmissione*, da cui vengono prelevati per creare un segmento
  - I dati ricevuti vengono memorizzati in un *buffer di ricezione*



# TCP

- ❑ In trasmissione il protocollo deve:
  - Accettare un flusso dati da livello applicazione
  - Spezzare il flusso dati in segmenti; ogni segmento è incapsulato in un pacchetto a livello rete
  - Trasmettere i segmenti tramite un sottostante livello rete inaffidabile
  - Gestire gli errori di trasmissione con la ritrasmissione dei segmenti
- ❑ In ricezione il protocollo deve:
  - Ricevere i segmenti dal livello rete
  - Segnalare i segmenti danneggiati
  - Riordinare i segmenti ricevuti
  - Consegnare a livello applicativo i dati



## Opzioni

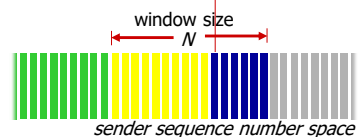
- ❑ Definite durante l'apertura della connessione, o durante la connessione
  - Negoziazione MSS (default 536 byte, max 65,535 byte)
  - Negoziazione Window scale, fattore di scala della finestra di ricezione, che può aumentare fino a  $2^{30}$  bytes (per *long fat networks*)
  - Timestamp: usato per il calcolo del RTT e per evitare riuso delle sequenze (**PAWS –Protection Against Wrapped Sequence numbers RFC 1323**)

## TCP – numeri di sequenza e ACK

- ❑ Dati: sequenza ordinata di byte
- ❑ **Sequenza**: posizione del primo byte del segmento, rispetto alla sequenza di byte
- ❑ **ACK**: prossimo byte atteso
- ❑ *Riscontro cumulativo*: la perdita di un ACK non blocca la trasmissione se confermato da un ACK successivo

outgoing segment from sender

source port #	dest port #
sequence number	
acknowledgement number	
	rwnd
checksum	urg pointer



sent ACKed    sent, not-yet ACKed but not usable ("in-flight")    usable but not yet sent    not usable

incoming segment to sender

source port #	dest port #
sequence number	
acknowledgement number	
A	rwnd
checksum	urg pointer