Reti di Calcolatori:
Internet, Intranet e Mobile Computing
a.a. 2007/2008

http://www.di.uniba.it/~lisi/courses/reti/reti0708.htm

dott.ssa Francesca A. Lisi
lisi@di.uniba.it

Orario di ricevimento: mercoledì ore 10-12

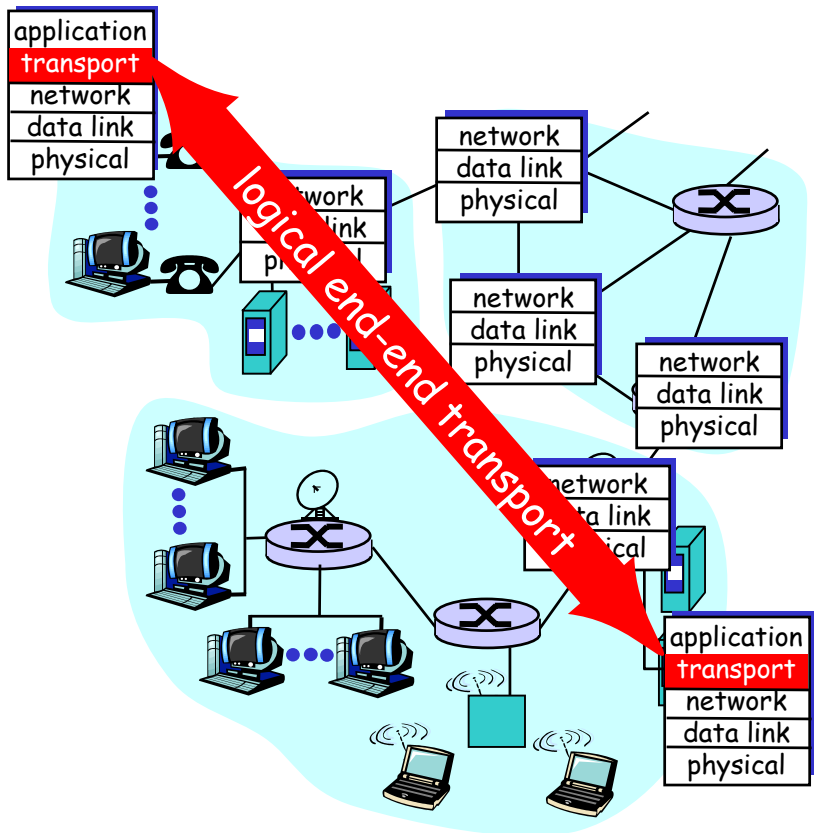
dib

Sommario della lezione di oggi: Lo strato di trasporto (1/3)

- Servizi e protocolli dello strato di trasporto
- Multiplazione e demultiplazione delle applicazioni
- Trasporto senza connessione: UDP
- Trasporto con connessione: TCP
- Il controllo della congestione nel TCP

Strato di trasporto: Servizi e protocolli

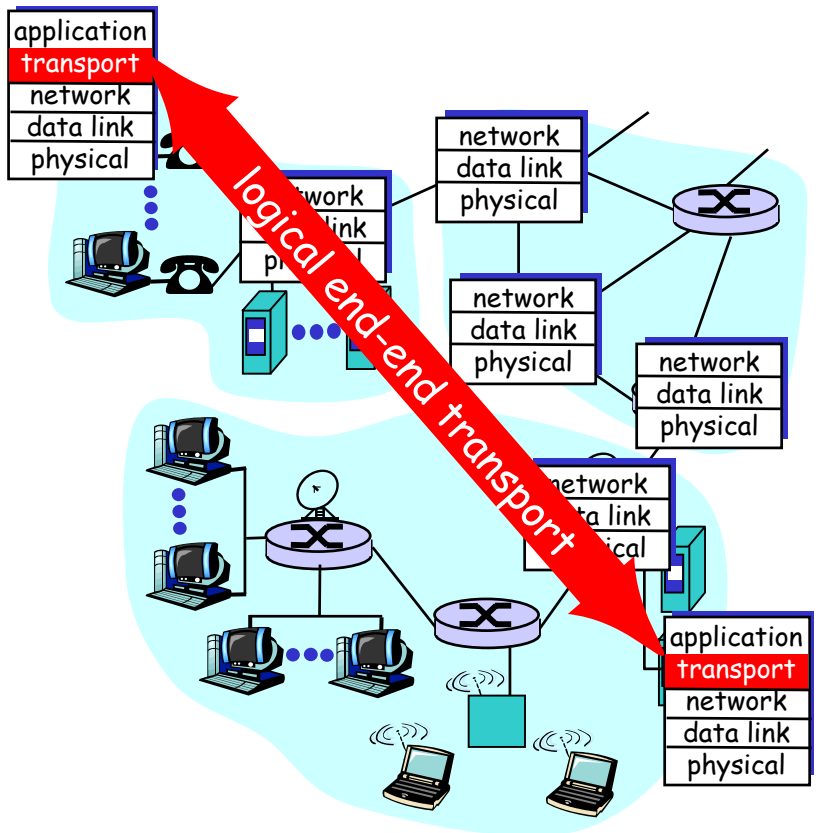
- I protocolli di trasporto:
  - forniscono *comunicazione logica* fra processi applicativi in esecuzione su host distinti
  - sono in esecuzione sui terminali
- servizi dello strato di transporto vs servizi dello strato di rete:
  - *strato di rete:* trasferimento dati fra terminali
  - *strato di trasporto:* trasferimento dati fra processi (basandosi su servizi dello strato di rete)
- analogia del servizio postale!



Strato di trasporto: Servizi e protocolli (cont.)

Servizi di trasporto Internet:

- consegna affidabile, in-order e unicast (TCP)
  - controllo della congestione
  - controllo del flusso
  - setup della connessione
- consegna inaffidabile ("best-effort"), disordinata e unicast o multicast: UDP
- servizi non disponibili:
  - real-time
  - garanzie sull'ampiezza di banda
  - multicast affidabile



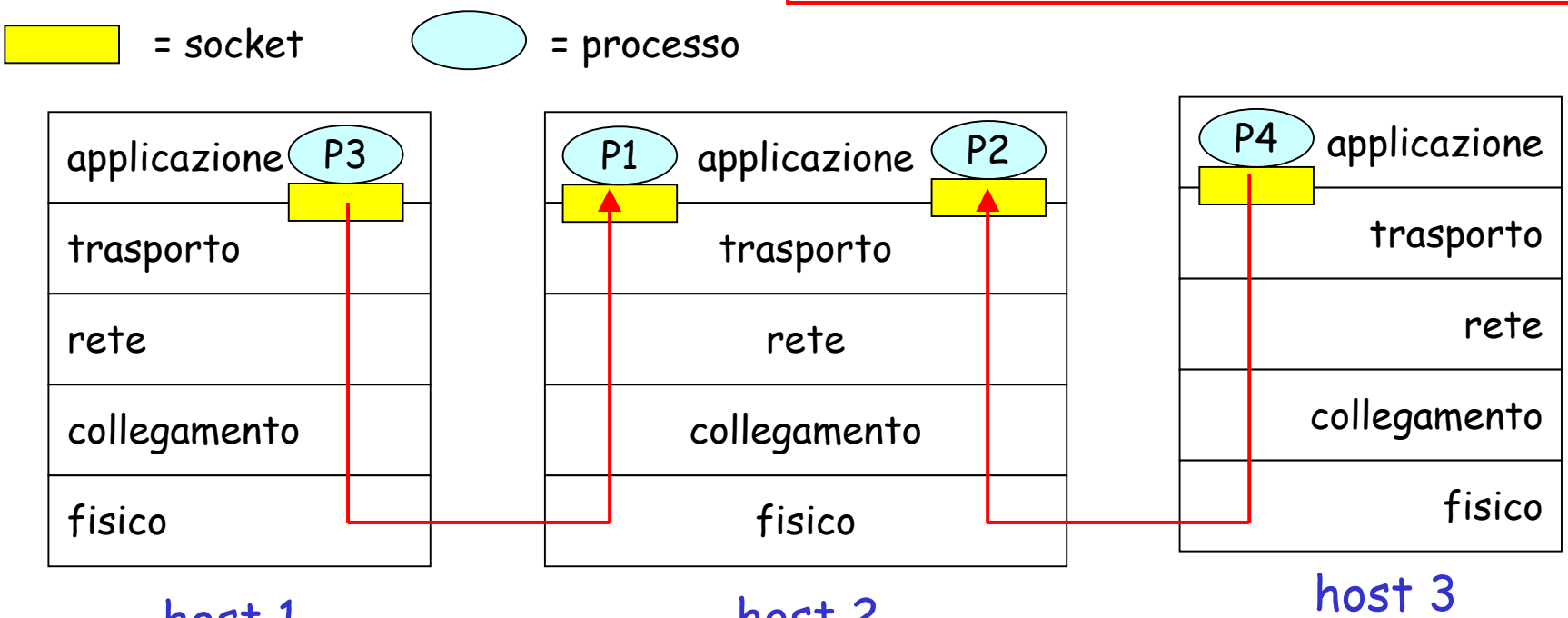
Multiplexing/demultiplexing

Demultiplexing nell'host ricevente:

consegnare i segmenti ricevuti alla socket appropriata

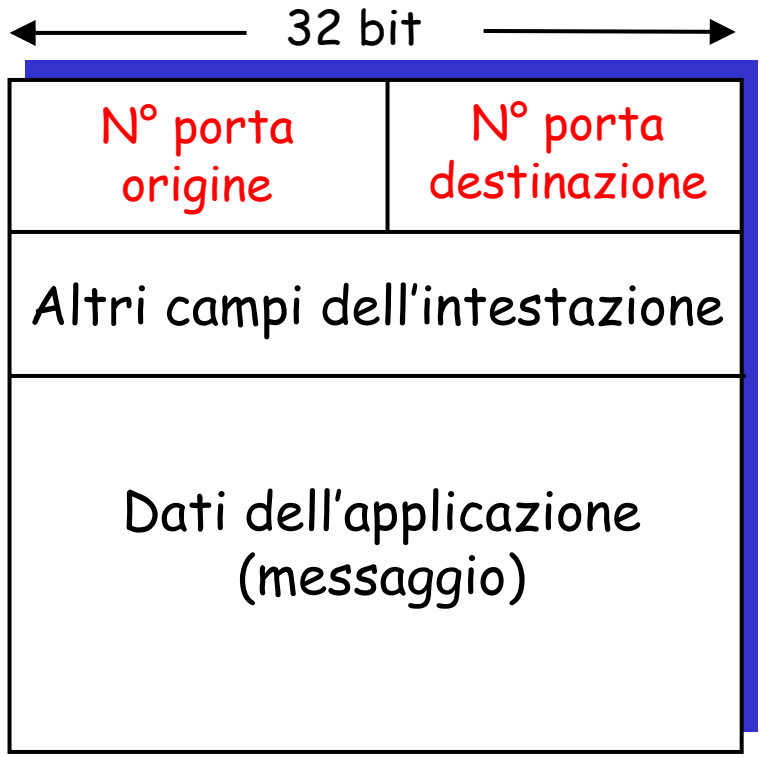
Multiplexing nell'host mittente:

raccogliere i dati dalle socket associate ad un processo ed incapsularli con l'intestazione (utilizzati poi per il demultiplexing)



Demultiplexing: come funziona

- L'host riceve i datagrammi IP
  - ogni datagramma ha un indirizzo IP di origine e un indirizzo IP di destinazione
  - ogni datagramma trasporta 1 segmento a livello di trasporto
  - ogni segmento ha un numero di porta di origine e un numero di porta di destinazione
- L'host usa gli indirizzi IP e i numeri di porta per inviare il segmento alla socket appropriata



Struttura del segmento TCP/UDP

Demultiplexing senza connessione

- Crea le socket con i numeri di porta:

```
DatagramSocket mySocket1 = new
   DatagramSocket(99111);
DatagramSocket mySocket2 = new
   DatagramSocket(99222);
```

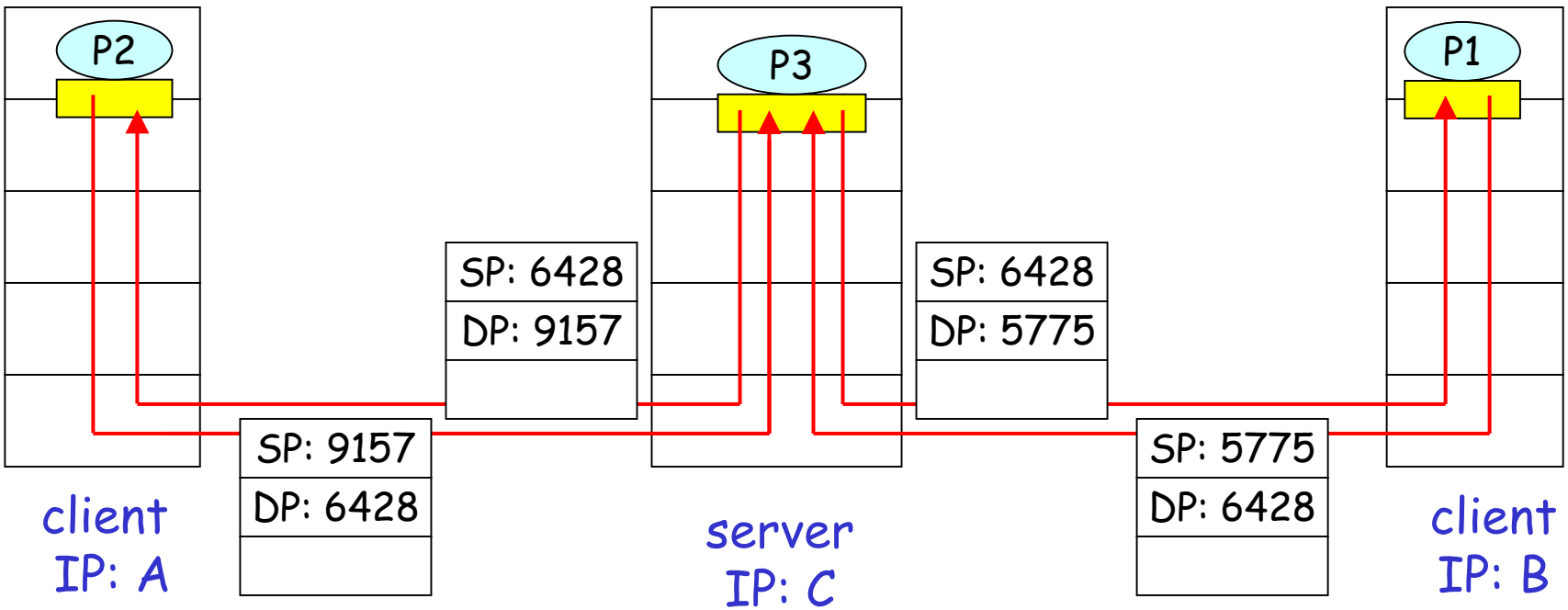
- La socket UDP è identificata da 2 parametri:

  (indirizzo IP di destinazione, numero della porta di destinazione)

- Quando l'host riceve il segmento UDP:
  - controlla il numero della porta di destinazione nel segmento
  - invia il segmento UDP alla socket con quel numero di porta
- I datagrammi IP con indirizzi IP di origine e/o numeri di porta di origine differenti vengono inviati alla stessa socket

Demultiplexing senza connessione (cont.)

```
DatagramSocket serverSocket = new DatagramSocket(6428);
```

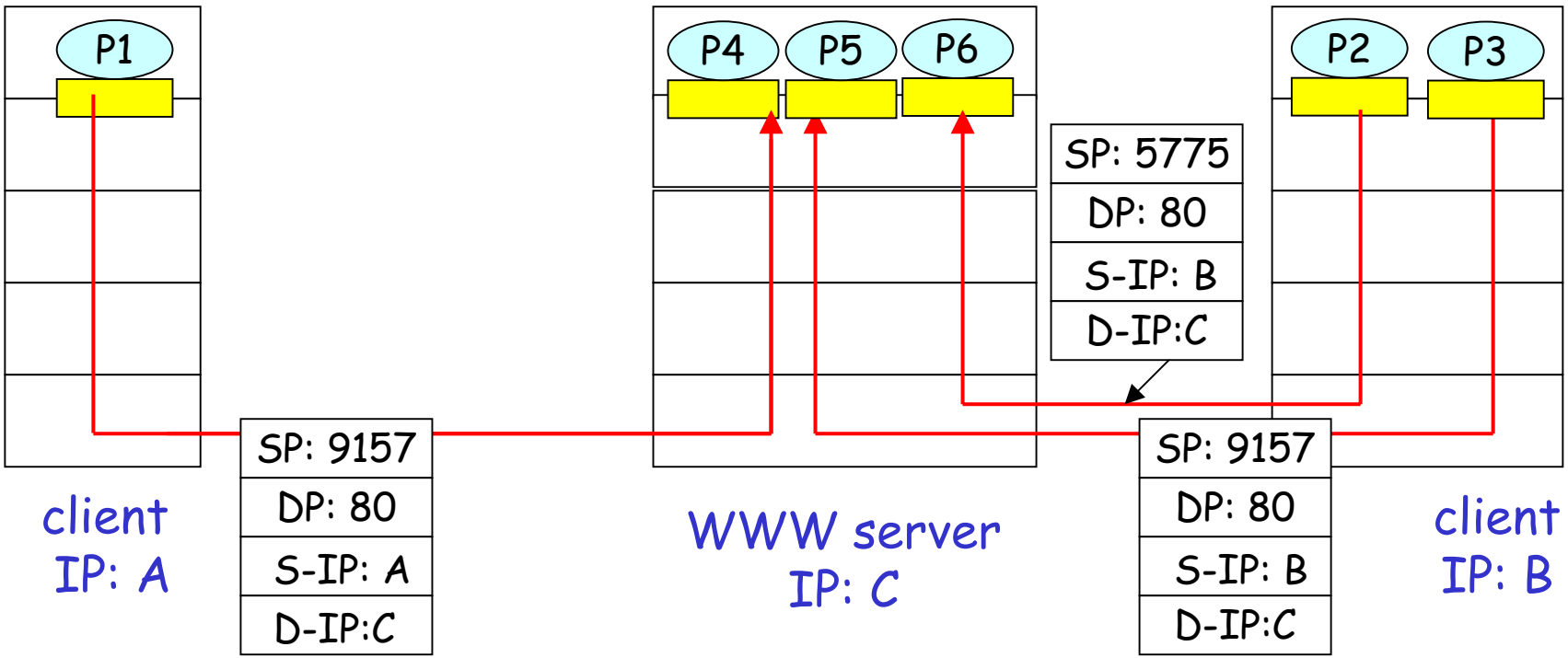


SP fornisce "l'indirizzo di ritorno"

Demultiplexing con connessione

- La socket TCP è identificata da 4 parametri:
  - indirizzo IP di origine
  - numero di porta di origine
  - indirizzo IP di destinazione
  - numero di porta di destinazione
- L'host ricevente usa i quattro parametri per inviare il segmento alla socket appropriata
- Un host server può supportare più socket TCP contemporanee:
  - ogni socket è identificata dai suoi 4 parametri
- N.B. I server WWW hanno socket differenti per ogni connessione client
  - con HTTP non-persistente si avrà una socket differente per ogni richiesta

Demultiplexing con connessione (cont.)



Trasporto senza connessione: UDP [RFC 768]

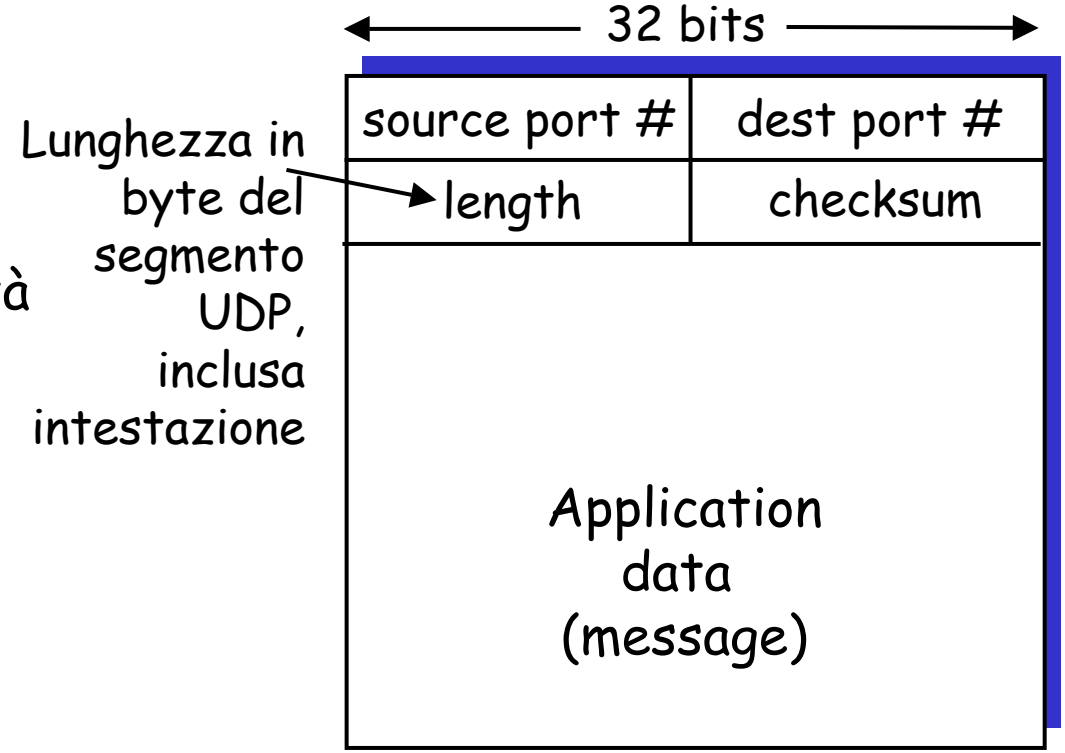
- Protocollo di trasporto "senza fronzoli" per Internet
- servizio "best effort", i segmenti UDP possono:
  - andare persi
  - essere consegnati in disordine
- *senza connessione:*
  - niente handshaking fra mittente e destinatario
  - ogni segmento gestito indipendentemente dagli altri

Perché esiste un UDP?

- Nessun ritardo dovuto al setup di connessione
- semplicità dovuta ad assenza di stato di connessione
- breve intestazione del segmento
- nessun controllo di congestione: UDP può "sparare a raffica" con velocità desiderata

UDP: struttura del segmento

- Spesso usato per far scorrere applicazioni multimediali
  - tollerante alla perdita di dati
  - sensibile alla velocità di trasmissione
- altri usi di UDP:
  - DNS
  - SNMP
- affidabilità su UDP va gestita a livello applicativo (politica di recupero degli errori)



Formato del segmento UDP

UDP: il checksum

Obiettivo: rilevare "errori" nel segmento trasmesso

Mittente:

- tratta i contenuti del segmento come una sequenza di interi a 16 bit
- checksum: somma (in complemento ad 1) dei contenuti del segmento
- il mittente pone il valore di checksum nel campo "UDP checksum"

Destinatario:

- calcola il checksum del segmento ricevuto
- controlla se il checksum calcolato equivale al valore del campo "UDP checksum":
  - NO - errore rilevato
  - YES - nessun errore rievato. Ma *potrebbero esserci errori comunque?* Lo vedremo più in là….

Sommario della prossima lezione: Lo strato di trasporto (2/3)

- Servizi e protocolli dello strato di trasporto
- Multiplazione e demultiplazione delle applicazioni
- Trasporto senza connessione: UDP
- Trasporto con connessione: TCP
- Il controllo della congestione nel TCP