

*Reti di calcolatori e Internet:
Un approccio top-down*

7^a edizione
Jim Kurose, Keith Ross

Pearson Paravia Bruno Mondadori Spa

Capitolo 3: Livello di trasporto

Obiettivi:

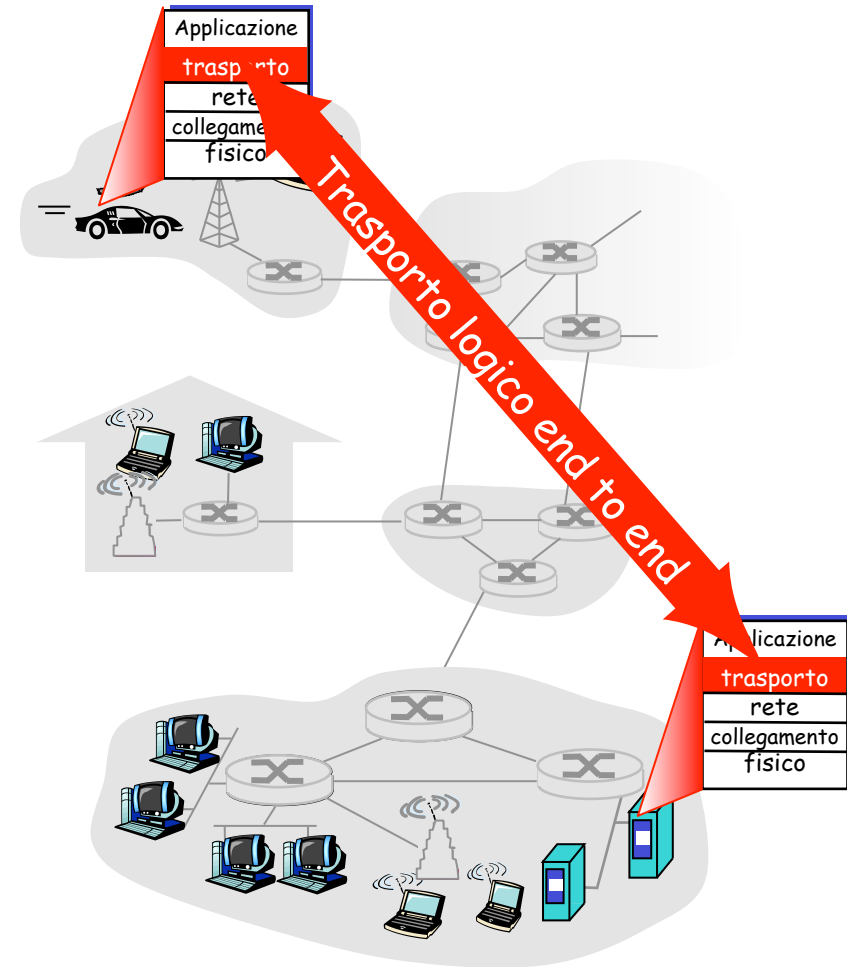
- ❑ Capire i principi che sono alla base dei servizi del livello di trasporto:
 - multiplexing/demultiplexing
 - trasferimento dati affidabile
 - controllo di flusso
 - controllo di congestione
- ❑ Descrivere i protocolli del livello di trasporto di Internet:
 - UDP: trasporto senza connessione
 - TCP: trasporto orientato alla connessione
 - controllo di congestione TCP

Capitolo 3: Livello di trasporto

- ❑ 3.1 Servizi a livello di trasporto
- ❑ 3.2 Multiplexing e demultiplexing
- ❑ 3.3 Trasporto senza connessione: UDP
- ❑ 3.4 Principi del trasferimento dati affidabile
- ❑ 3.5 Trasporto orientato alla connessione: TCP
 - struttura dei segmenti
 - trasferimento dati affidabile
 - controllo di flusso
 - gestione della connessione
- ❑ 3.6 Principi del controllo di congestione
- ❑ 3.7 Controllo di congestione TCP

Servizi e protocolli di trasporto

- ❑ Forniscono la *comunicazione logica* tra processi applicativi di host differenti
- ❑ I protocolli di trasporto vengono eseguiti nei sistemi terminali
 - lato invio: scinde i messaggi in *segmenti* e li passa al livello di rete
 - lato ricezione: riassembla i segmenti in messaggi e li passa al livello di applicazione
- ❑ Più protocolli di trasporto sono a disposizione delle applicazioni
 - Internet: TCP e UDP



NB: Il livello trasporto agisce a livello locale sui sistemi terminali

Relazione tra livello di trasporto e livello di rete

- ❑ *livello di rete:*
comunicazione logica tra host
- ❑ *livello di trasporto:*
comunicazione logica tra processi
 - si basa sui servizi del livello di rete e li potenzia

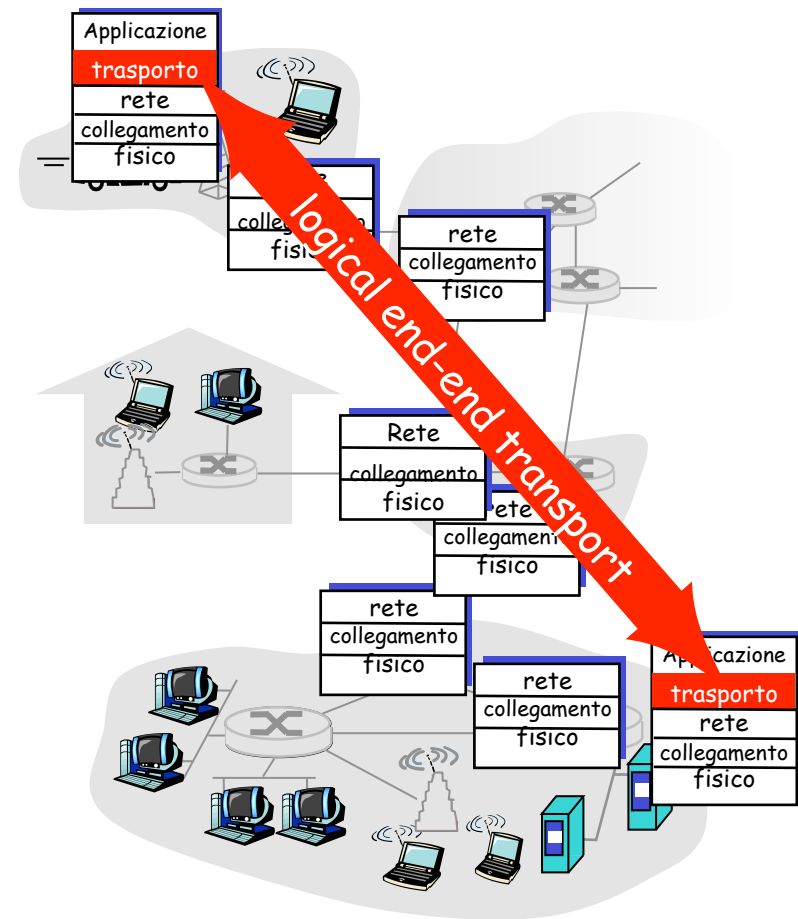
Analogia con la posta ordinaria:

12 ragazzi inviano lettere a 12 ragazzi

- ❑ processi = ragazzi
- ❑ messaggi delle applicazioni = lettere nelle buste
- ❑ host = case
- ❑ protocollo di trasporto = Anna e Andrea
- ❑ protocollo del livello di rete = servizio postale

Protocolli del livello di trasporto in Internet

- ❑ Affidabile, consegne nell'ordine originario (TCP)
 - controllo di congestione
 - controllo di flusso
 - setup della connessione
- ❑ Inaffidabile, consegne senz'ordine: UDP
 - estensione senza fronzoli del servizio di consegna best effort
- ❑ Servizi non disponibili:
 - garanzia su ritardi
 - garanzia su ampiezza di banda



Capitolo 3: Livello di trasporto

- ❑ 3.1 Servizi a livello di trasporto
- ❑ 3.2 Multiplexing e demultiplexing
- ❑ 3.3 Trasporto senza connessione: UDP
- ❑ 3.4 Principi del trasferimento dati affidabile
- ❑ 3.5 Trasporto orientato alla connessione: TCP
 - struttura dei segmenti
 - trasferimento dati affidabile
 - controllo di flusso
 - gestione della connessione
- ❑ 3.6 Principi del controllo di congestione
- ❑ 3.7 Controllo di congestione TCP

Multiplexing/demultiplexing

Demultiplexing

nell'host ricevente:

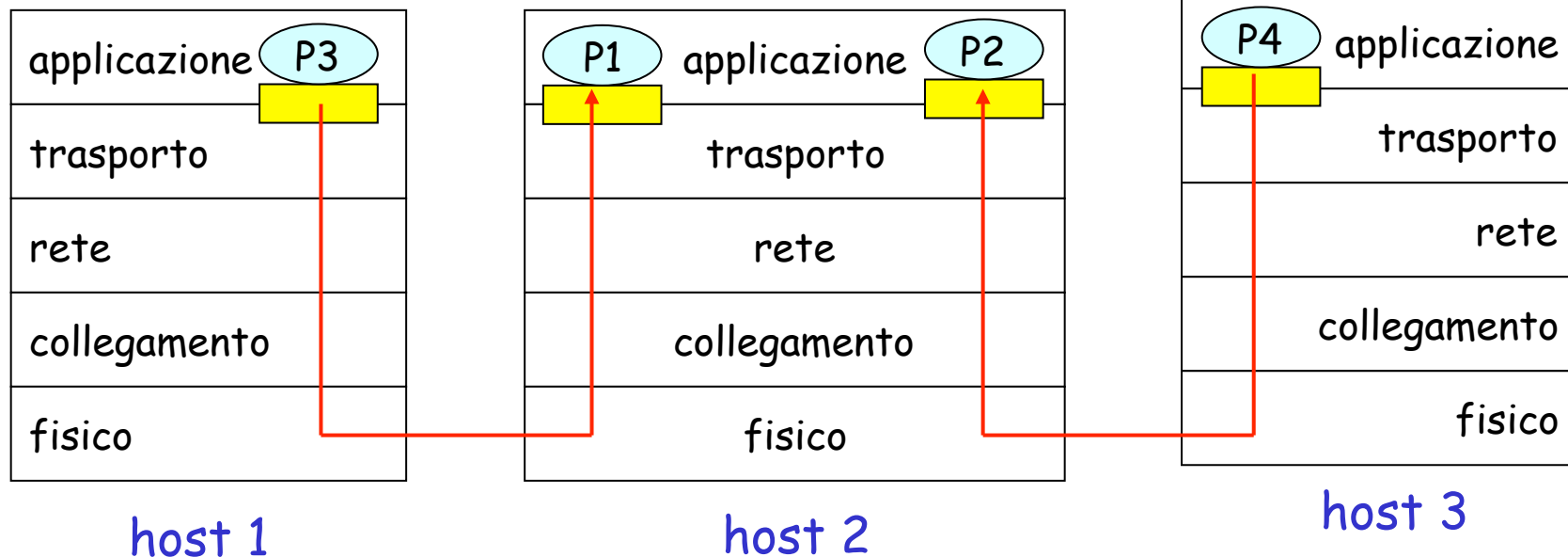
consegnare i segmenti ricevuti
alla socket appropriata

Multiplexing

nell'host mittente:

raccogliere i dati da varie
socket, incapsularli con
l'intestazione (utilizzata poi
per il demultiplexing)

 = socket  = processo

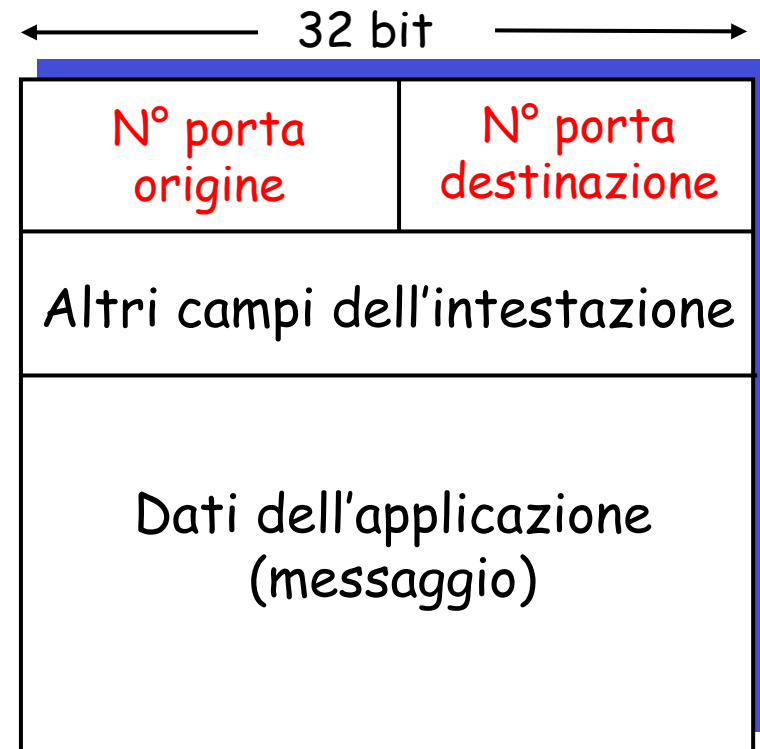


Come funziona il demultiplexing

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

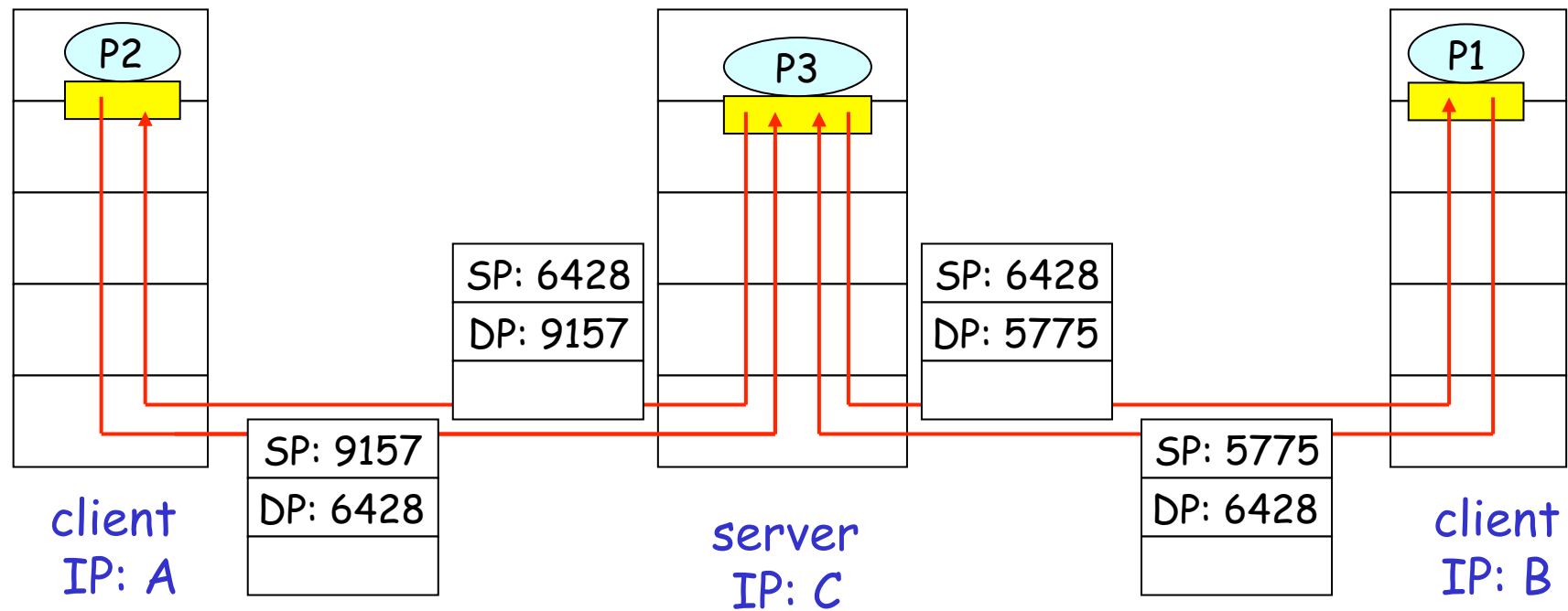


Struttura del segmento TCP/UDP

Demultiplexing senza connessione

- ❑ Crea le socket con i numeri di porta
- ❑ 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
- ❑ Datagrammi IP con indirizzi IP di origine e/o numeri di porta di origine differenti vengono inviati alla stessa socket

Demultiplexing senza connessione (continua)

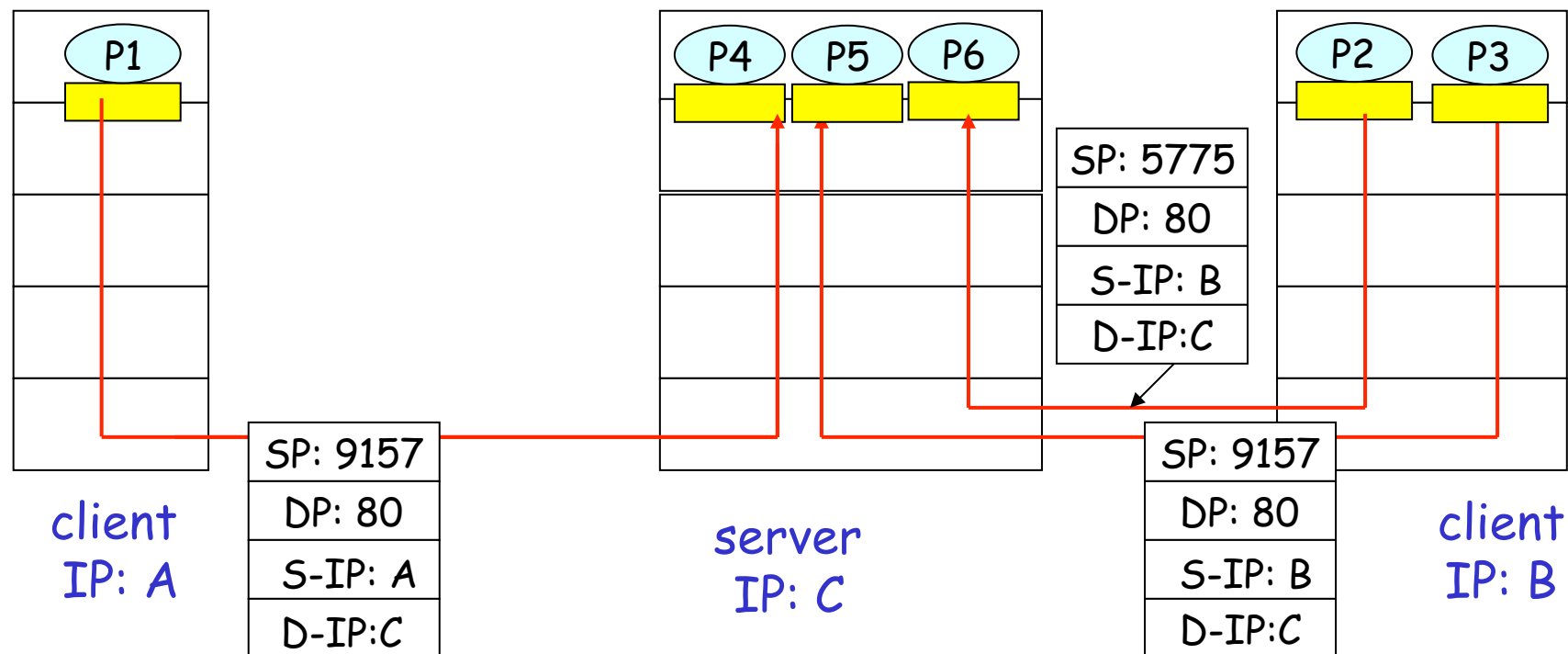


SP fornisce "l'indirizzo di ritorno"

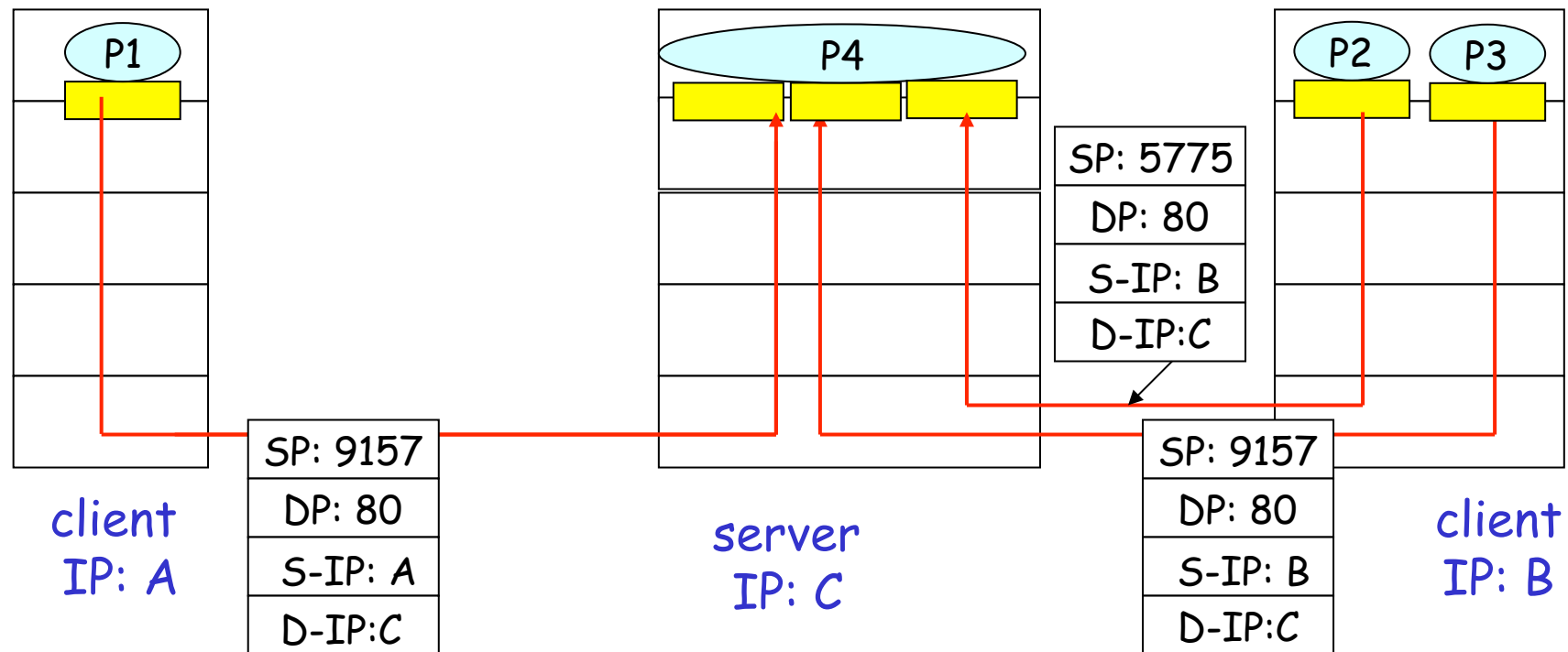
Demultiplexing orientato alla connessione

- ❑ 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
- ❑ I server web hanno socket differenti per ogni connessione client
 - con HTTP non-persistente si avrà una socket differente per ogni richiesta

Demultiplexing orientato alla connessione (continua)



Demultiplexing orientato alla connessione: thread dei server web



Capitolo 3: Livello di trasporto

- ❑ 3.1 Servizi a livello di trasporto
- ❑ 3.2 Multiplexing e demultiplexing
- ❑ 3.3 Trasporto senza connessione: UDP
- ❑ 3.4 Principi del trasferimento dati affidabile
- ❑ 3.5 Trasporto orientato alla connessione: TCP
 - struttura dei segmenti
 - trasferimento dati affidabile
 - controllo di flusso
 - gestione della connessione
- ❑ 3.6 Principi del controllo di congestione
- ❑ 3.7 Controllo di congestione TCP

UDP: User Datagram Protocol [RFC 768]

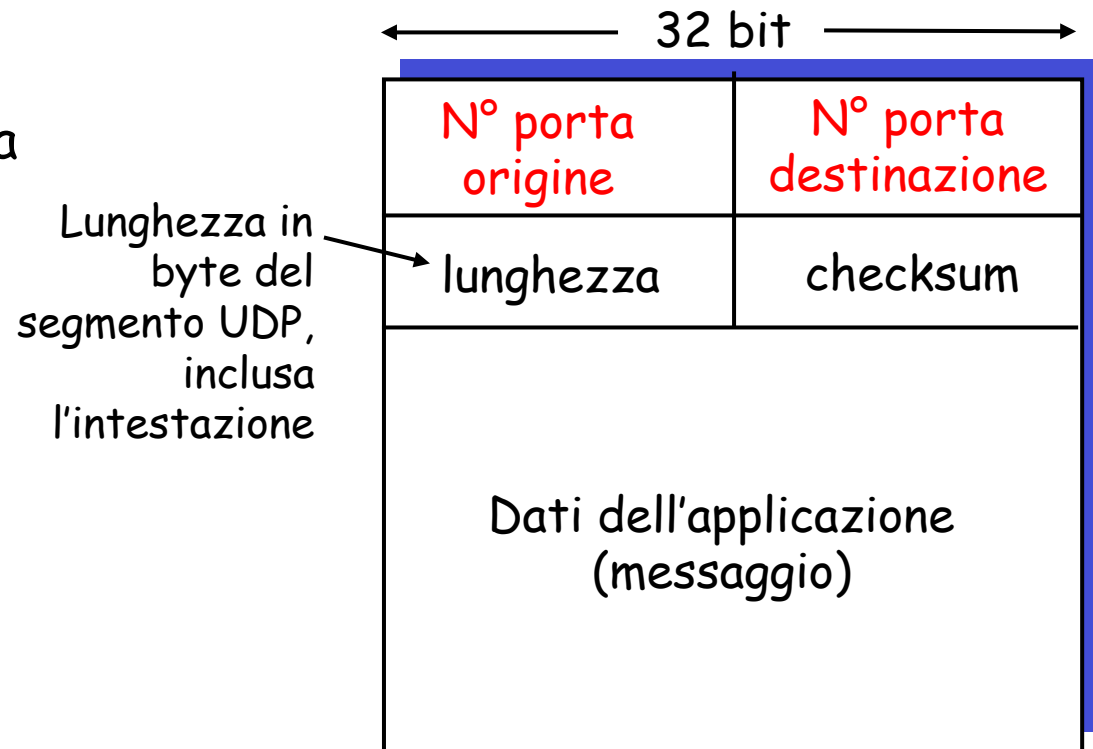
- ❑ Protocollo di trasporto "senza fronzoli"
- ❑ Servizio di consegna "a massimo sforzo", i segmenti UDP possono essere:
 - perduti
 - consegnati fuori sequenza all'applicazione
- ❑ *Senza connessione:*
 - no handshaking tra mittente e destinatario UDP
 - ogni segmento UDP è gestito indipendentemente dagli altri

Perché esiste UDP?

- ❑ Nessuna connessione stabilita (che potrebbe aggiungere un ritardo)
- ❑ Semplice: nessuno stato di connessione nel mittente e destinatario
- ❑ Intestazioni di segmento corte
- ❑ Senza controllo di congestione: UDP può sparare dati a raffica

UDP: ulteriori informazioni

- ❑ Utilizzato spesso nelle applicazioni multimediali
 - tollera piccole perdite
 - sensibile alla frequenza
- ❑ Altri impieghi di UDP
 - DNS
 - SNMP
- ❑ Trasferimento affidabile con UDP: aggiungere affidabilità al livello di applicazione
 - Recupero degli errori delle applicazioni!



Struttura del segmento UDP

Checksum UDP

Obiettivo: rilevare gli “errori” (bit alterati) nel segmento trasmesso

Mittente:

- ❑ Tratta il contenuto del segmento come una sequenza di interi da 16 bit
- ❑ checksum: somma (complemento a 1) i contenuti del segmento (i riporti si sommano al primo bit)
- ❑ Il mittente pone il valore della checksum nel campo checksum del segmento UDP

Ricevente:

- ❑ calcola la checksum del segmento ricevuto
- ❑ controlla se la checksum calcolata è uguale al valore del campo checksum:
 - No - errore rilevato
 - Sì - nessun errore rilevato. *Ma potrebbero esserci errori nonostante questo? Lo scopriremo più avanti ...*

Esempio di checksum

□ Nota

- Quando si sommano i numeri, un riporto dal bit più significativo deve essere sommato al risultato

□ Esempio: sommare due interi da 16 bit

		1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
		1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
<hr/>																	
a capo	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1
<hr/>																	
somma		1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	0
checksum		0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1