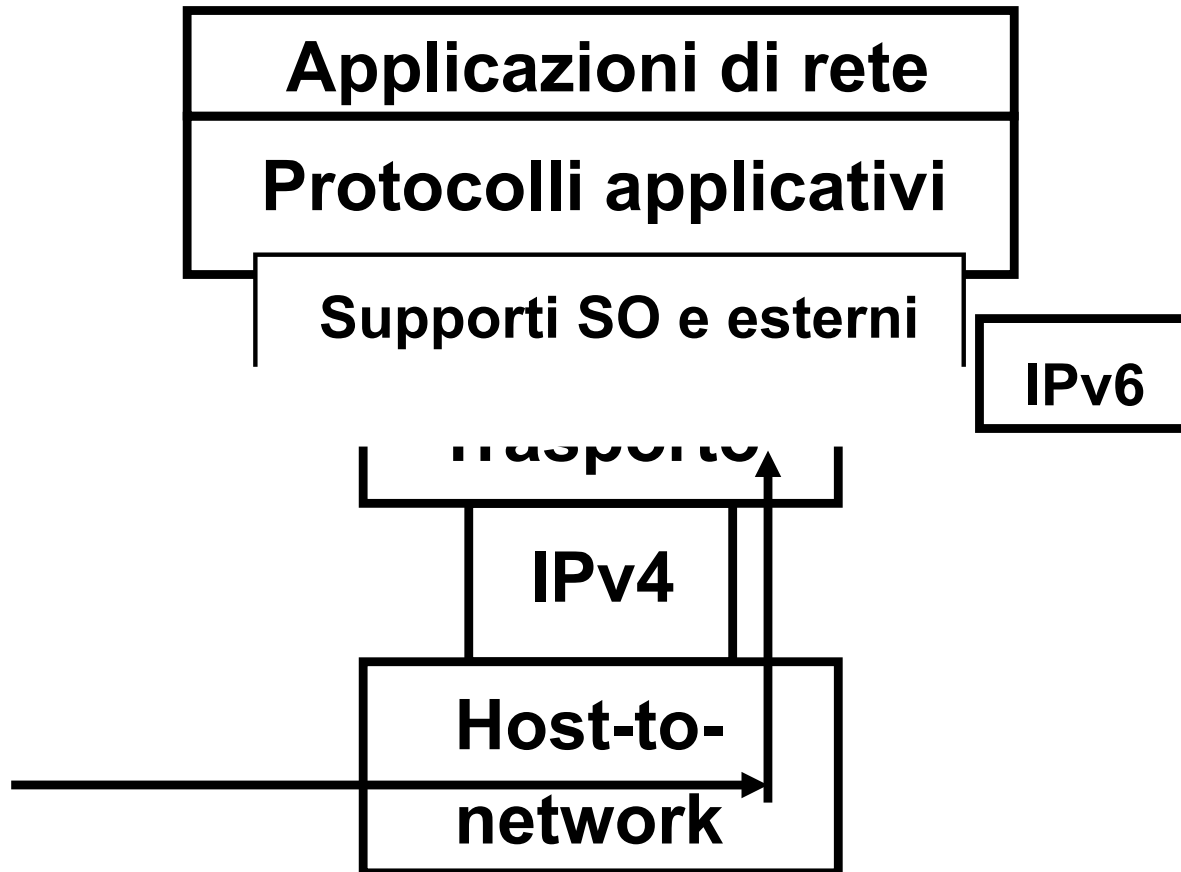


LIVELLO TRASPORTO

Dove ci troviamo?

- Abbiamo trattato il livello h2n:
- Abbiamo trattato il livello IPv4

DA FARE:



Livello 4 (*transport*)

- Il livello trasporto estende il servizio di consegna con impegno proprio del protocollo IP tra due host terminali ad un servizio di consegna a due processi applicativi in esecuzione sugli host
- Nello stack TCP/IP, se IP è il protocollo di rete, TCP sarà l'unico protocollo di trasporto?
NO!
- **TCP (*Transfer Control Protocol*)** è solo un protocollo del livello di trasporto della suite TCP/IP
- Un altro protocollo fondamentale è costituita dal protocollo **UDP (*User Datagram Protocol*)**

Livello 4 (*transport*)

- Il livello transport affronta la comunicazione fra processi applicativi in esecuzione sugli host
- **Servizi aggiuntivi minimi rispetto a IP**
 - *multiplazione e demultiplazione messaggi tra processi*
 - *rilevamento dell'errore (mediante checksum)*
- Protocolli *transport* standard dello stack TCP/IP
 - **UDP** (*User Datagram Protocol*)
 - **TCP** (*Transmission Control Protocol*): offre servizi aggiuntivi rispetto a UDP
- Potrebbero esistere altri protocolli trasporto in contesti particolari e/o proprietari

Panoramica servizi di UDP e TCP

Servizi comuni a UDP e TCP:

Estensione del servizio di consegna del protocollo IP tra due nodi terminali ad un servizio di consegna a due processi applicativi in esecuzione sui nodi terminali

- *multiplazione e demultiplazione*
- *rilevamento dell'errore (non correzione!)*

Alcuni (principali) servizi aggiuntivi di TCP:

- **Trasferimento affidabile** dei dati
 - controllo di flusso, numeri di sequenza, acknowledgement e timer
- **Controllo di flusso e congestione**
 - regola il tasso di invio dei segmenti da parte del mittente

Multiplicazione e demultiplicazione host-applicazioni

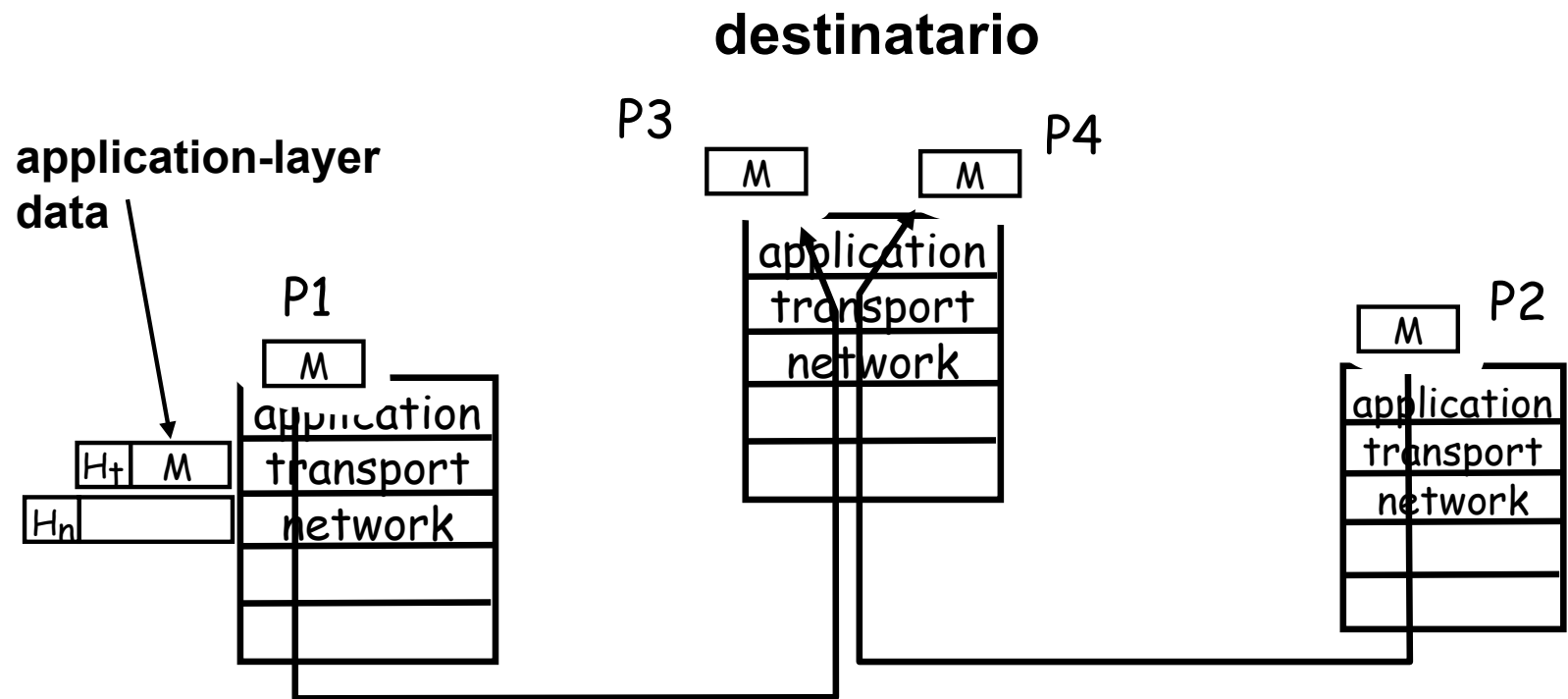
1. Multiplazione e demultiplazione

Il protocollo IP non consegna i dati tra processi applicativi in esecuzione sui nodi terminali (un indirizzo IP per identificare ogni nodo terminale) → compito del protocollo di trasporto

Creazione dei segmenti provenienti dai messaggi di diversi processi applicativi → multiplazione

Ogni segmento dello strato di trasporto possiede un campo contenente l'informazione usata per determinare a quale processo deve essere consegnato il segmento → demultiplazione

Esempio

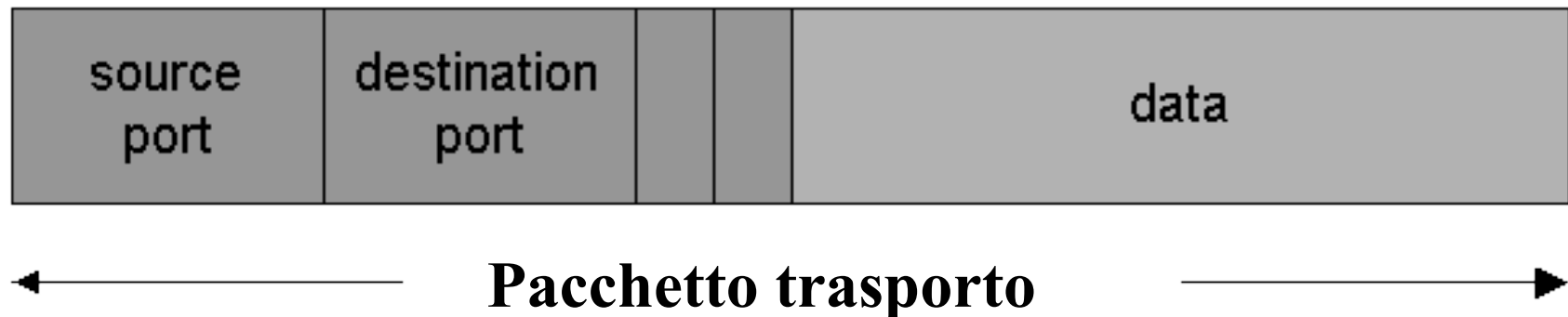


Multiplicazione e demultiplicazione (*cont.*)

UDP e TCP attuano la moltiplicazione/demultiplicazione includendo due campi speciali nell'header del segmento:

- il *numero di porta del mittente*
- il *numero di porta del destinatario*

Permettono di identificare in modo univoco i due processi applicativi, residenti su due nodi terminali e comunicanti tra loro



Paradigma client/server [1]

Tipicamente descriviamo una comunicazione a livello transport tramite il paradigma client/server

1. Un processo server richiede l'apertura di una porta «in ascolto» (connessione passiva, *listening port*) per un certo protocollo trasporto al proprio sistema operativo
 - Ogni volta che il SO riceve un pacchetto con protocollo LV4, consulta la porta destinazione, i processi in ascolto «registrati», e associa il pacchetto al processo (nota: il SO potrebbe non inoltrare subito il pacchetto al processo, in alcuni casi potrebbe esserci una conservazione presso un buffer, vedi TCP)

Paradigma client/server [2]

2. Un processo client apre invece una connessione attiva indicando la porta destinazione in cui è in ascolto il processo server (oltre all'indirizzo IP)
 - Anche il processo client viene associato a una porta per poter gestire correttamente eventuali risposte da parte del processo server
 - La porta del processo client è spesso non indicata esplicitamente dal processo applicativo, e in questi casi è scelta in modo autonomo dal SO

Multiplazione e demultiplazione (*cont.*)

Numero di porta: numero di 16 bit compreso tra 0 e 65535

Numeri di porta noti (*well-known ports, assigned numbers* in **[RFC 1700]**): tra 0 e 1023 riservati per protocolli applicativi noti (ad es., HTTP e FTP)

- **HTTP**: numero di porta 80
- **Telnet**: numero di porta 23
- **SMTP**: numero di porta 25
- **DNS**: numero di porta 53

Quando si realizza un nuovo servizio di rete ad ampia diffusione è necessario assegnargli un nuovo numero di porta

Occorre sia il numero di porta del mittente sia quello del destinatario per distinguere processi dello stesso tipo ed in esecuzione negli stessi istanti

Categorie di numeri di porta

0-1023 → Well Known Ports

NON DEVONO essere usate senza una precedente autorizzazione da IANA [RFC4340]. Nella maggior parte dei sistemi, possono essere usate solo da processi con privilegi di root o simili

1024-49151 → Registered Ports

L'uso di queste porte è registrato a beneficio degli utenti della rete, ma non esistono vincoli restrittivi. Tuttavia, NON DEVONO essere usate senza una precedente autorizzazione da IANA [RFC4340]. Nella maggior parte dei sistemi, possono essere usate da qualsiasi processo

49152-65535 → Dynamic or Private Ports

NON è applicato alcun controllo per cui possono essere usate liberamente

Porte note

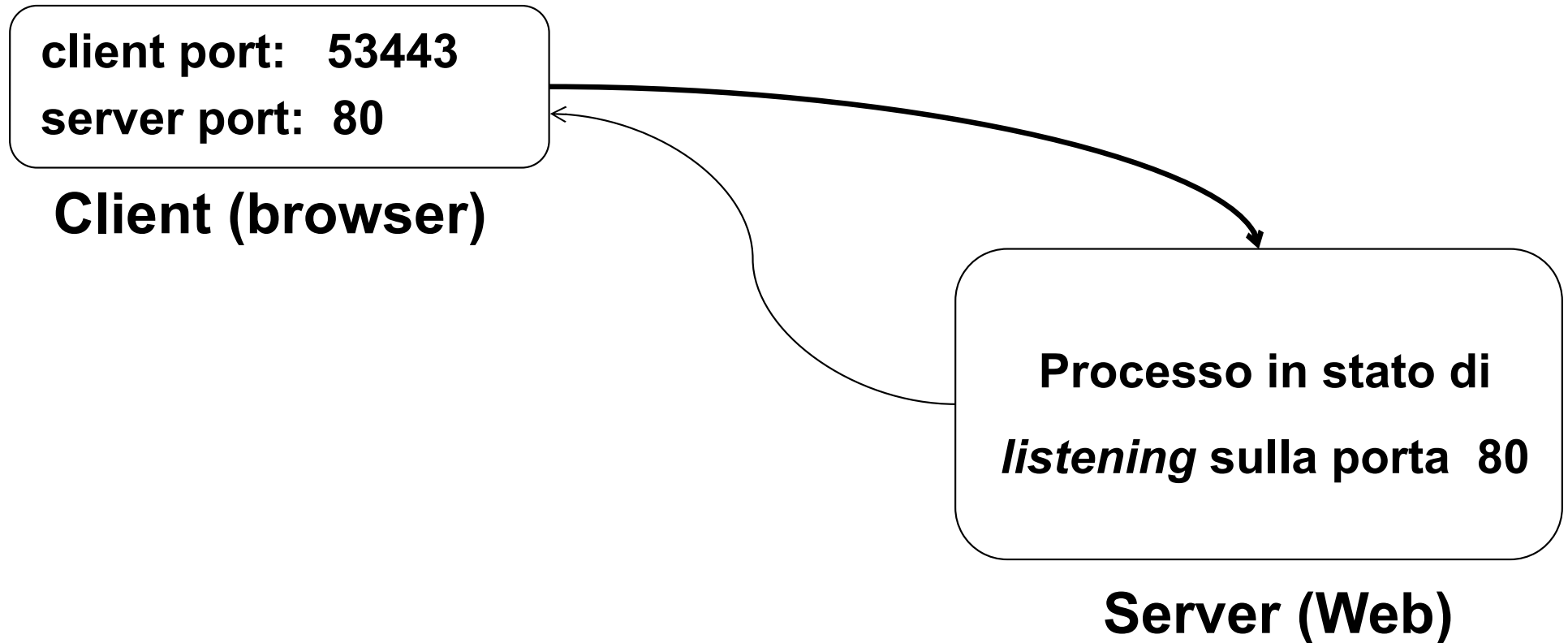
Le porte associate a servizi standard sono solitamente note ai sistemi operativi ed accessibili tramite file di configurazione (file 'services', tipicamente presente in /etc/
\Windows\System32\drivers\etc in sistemi basati su Linux e Windows)

Assegnazione dei numeri di porta

Modello client/server

1. Numero di porta del **destinatario** nel pacchetto inviato dal client al server corrisponde al numero di porta del servizio richiesto (ad es., 80 per HTTP)
2. Numero di porta del **mittente** nel pacchetto inviato dal client al server corrisponde ad un numero di porta scelto tra quelli non in uso sul client (solitamente fra le porte alte)
3. Numero di porta del **mittente** nel pacchetto inviato dal server al client corrisponde al numero di porta del servizio richiesto (ad es., 80 per HTTP)
4. Numero di porta del **destinatario** nel pacchetto inviato dal server al client corrisponde al numero di porta indicato dal client nel messaggio precedentemente inviato

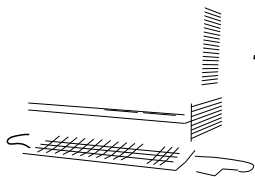
Esempio: *http*



Indirizzo IP e numero di porta (1)

Due processi client, residenti su host diversi per comunicare con lo stesso servizio applicativo sono distinti in base al loro indirizzo IP (sorgente), al numero di porta sorgente (SP) e *al protocollo di trasporto della porta (discussione successiva)*:

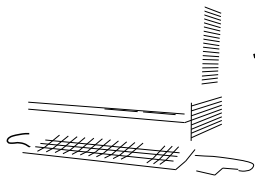
IP =190.85.14.32



host client con
connessione *http*

**IP =190.85.14.32, IP =98.21.45.33
SP=50433, DP=80 (TCP)**

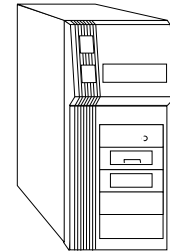
IP =144.21.67.11



host client con
connessione *http*

**SP=53556, DP=80 (TCP)
IP =144.21.67.11, , IP =98.21.45.33**

IP =98.21.45.33

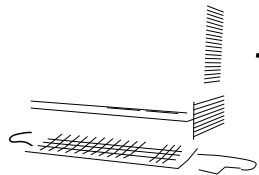


host server che gestisce
multiple connessioni *http*

Indirizzo IP e numero di porta (2)

Nel caso dello stesso numero di porta, l'indirizzo IP può comunque permettere di distinguere l'entità della comunicazione

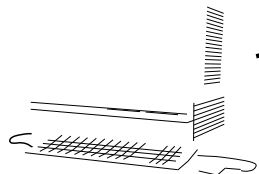
IP =190.85.14.32



host client con
connessione *http*

IP =190.85.14.32
SP=15433, DP=80 (TCP)

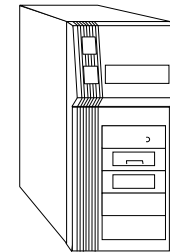
IP =144.21.67.11



host client con
connessione *http*

SP=15433, DP=80 (TCP)
IP =144.21.67.11

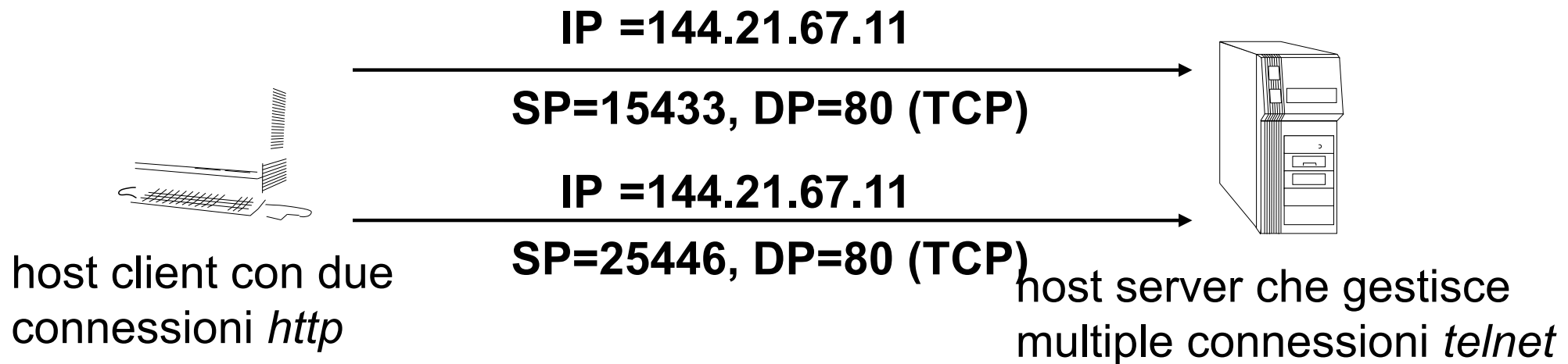
IP =98.21.45.33



host server che gestisce
multiple connessioni *telnet*

Indirizzo IP e numero di porta (3)

Due processi client in esecuzione sullo stesso host (o su host differenti in una rete privata, ma che impiegano lo stesso indirizzo pubblico), vengono distinti solo sulla base della porta sorgente



Indirizzi

- A livello network (IP)
→ [indirizzo IP_x , indirizzo IP_y]
- A livello trasporto (UDP, TCP)
→ [porta_w, porta_z, <protocollo trasporto>]

L'univocità di ciascun flusso a livello trasporto in Internet è dato dalla quintupla

**[(indirizzo IP_x , porta_w), (indirizzo IP_y , porta_z),
<protocollo trasporto>]**

NOTA 1

- Usiamo solitamente il termine **flusso di pacchetti** (meglio **packet flow**) per identificare tutti i pacchetti che appartengono a una stessa comunicazione
- A livello 4 possiamo utilizzare la quintupla per identificare che un certo insieme di pacchetti appartengono a uno stesso flusso
- In alcuni protocolli applicativi complessi che utilizzano più porte, possiamo analizzare informazioni ai livelli superiori per associare più «quintuple» allo stesso flusso
 - Multiplazione da parte del protocollo applicativo

NOTA 2

- Ricordarsi che invece nei protocolli di livello 3 (ad esempio, ICMP) non esistono le porte!
- In generale, l'unica informazione negli header che ci permette di distinguere le comunicazioni fra diversi host sono l'indirizzo IP mittente e destinazione
- In alcuni casi, possiamo utilizzare informazioni nel payload del protocollo per ottenere ulteriori informazioni
 - Esempio: ID dei pacchetti ICMP Echo request/response