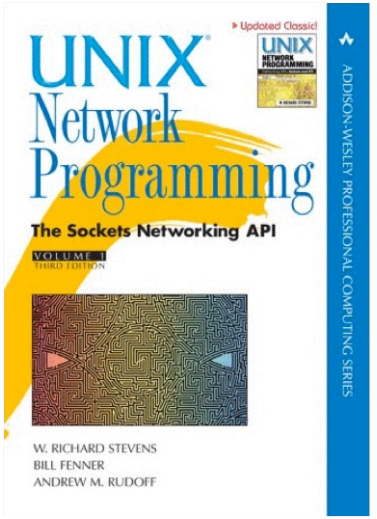
Laboratorio di Reti di Calcolatori

Lezione 1

Materiale didattico

- Slides delle lezioni
- UNIX® Network Programming Volume 1, 3rd ed.: The Sockets Networking API, R. Stevens
- Gapil - Guida alla Programmazione in Linux
  - Disponibile online



Programma

- Introduzione architetture e protocolli di rete: TCP, UDP
- Programmi per la gestione della rete in Unix/Linux
  - ifconfig, arp, route, netstat, ping, traceroute, tcpdump, nslookup
- Applicazioni di rete: DNS, FTP
- Il modello di programmazione client/server e P2P
- Socket di Berkeley:
  - l'interfaccia di programmazione socket
  - socket TCP/UDP
  - server concorrenti
  - IO/Multiplexing
  - conversione di nomi ed Indirizzi

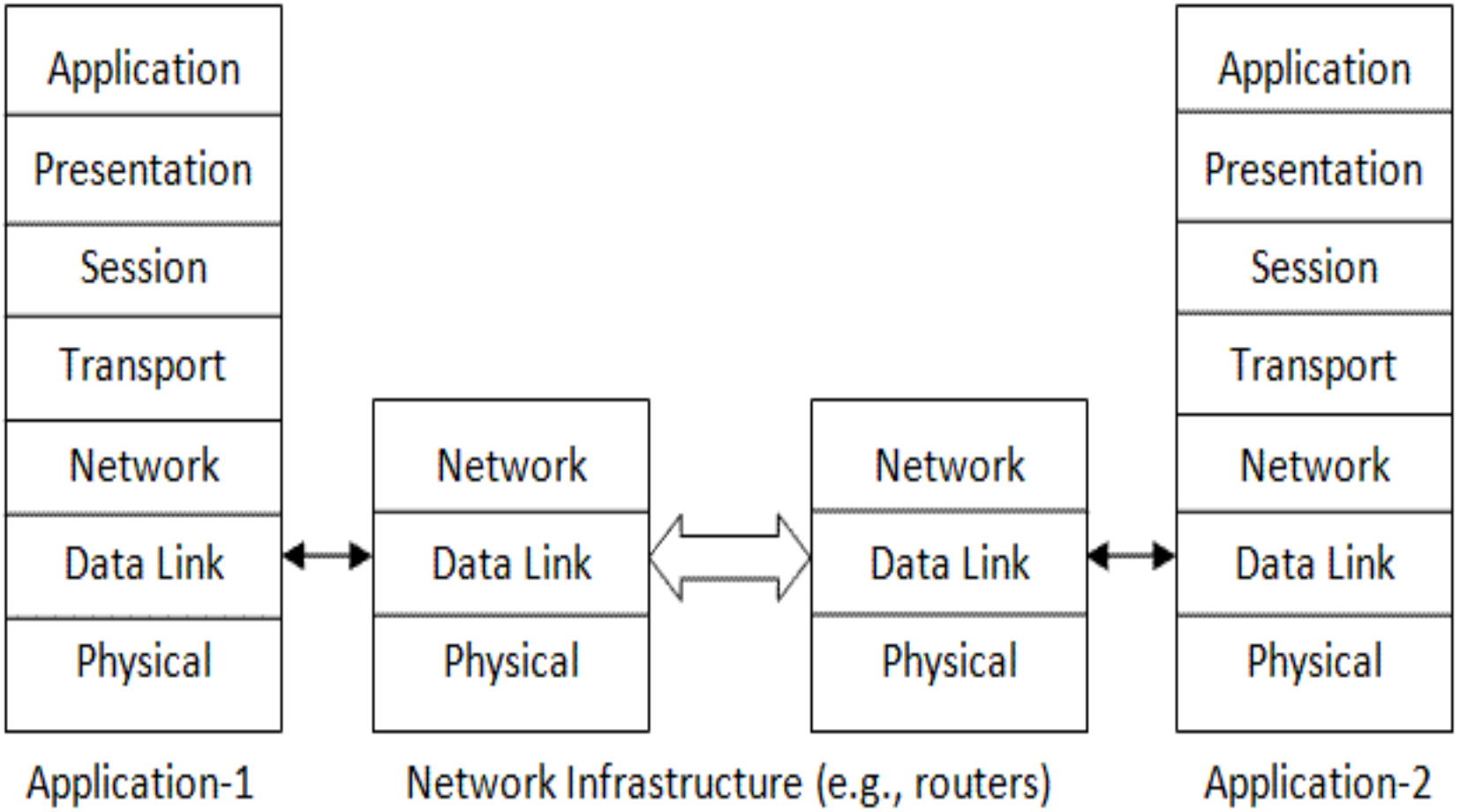
Applicazioni di rete

- Applicazioni di rete
  - processi eventualmente in esecuzione su macchine differenti
  - operano in modo indipendente
  - scambiano informazioni attraverso una rete
- Il sottosistema di comunicazione
  - gestisce lo scambio di informazioni
  - attraverso i propri servizi

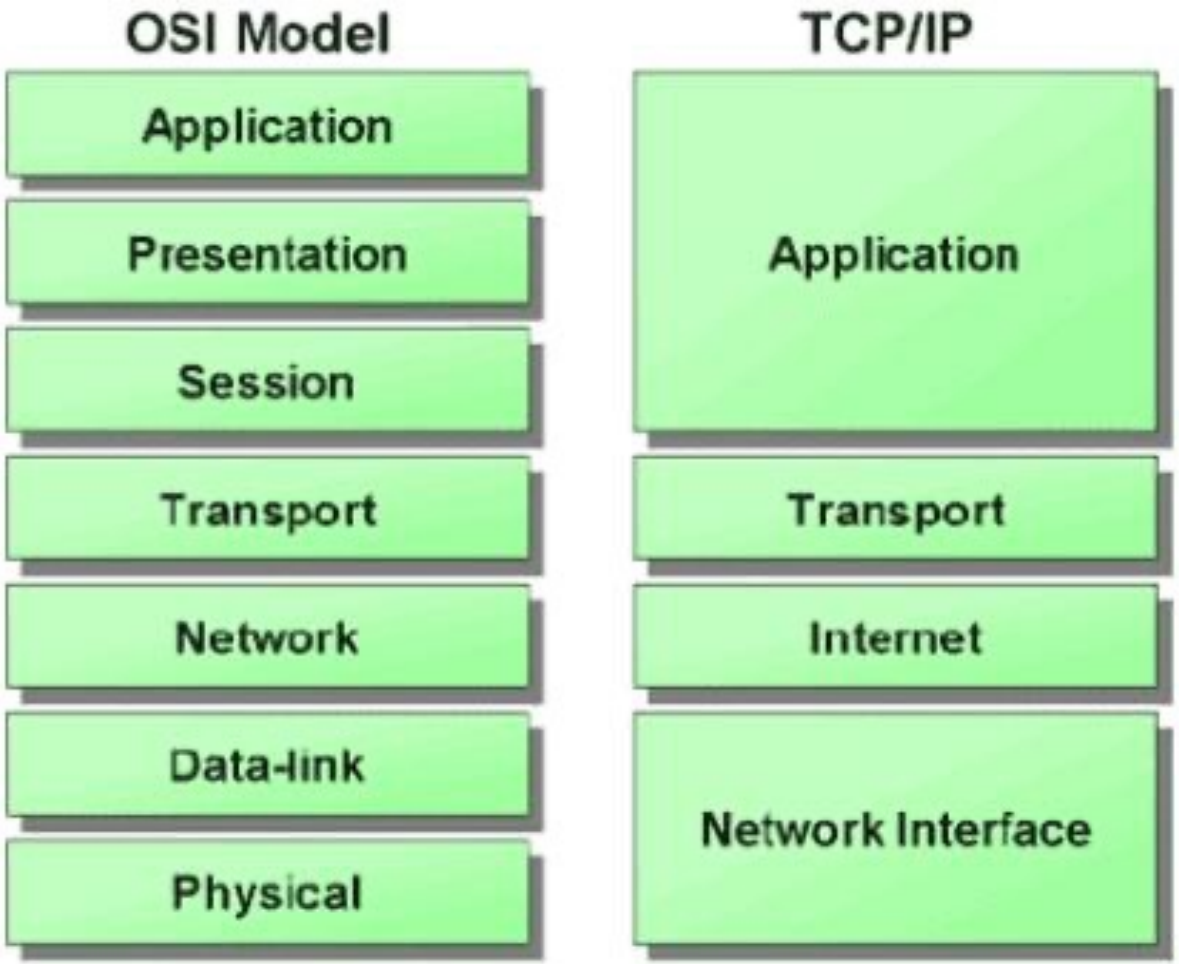
Protocollo

- La comunicazione tra le entità avviene attraverso un protocollo
- Insieme di regole e di convenzioni tra i partecipanti alla comunicazione

Protocolli e stratificazione (ISO/OSI)

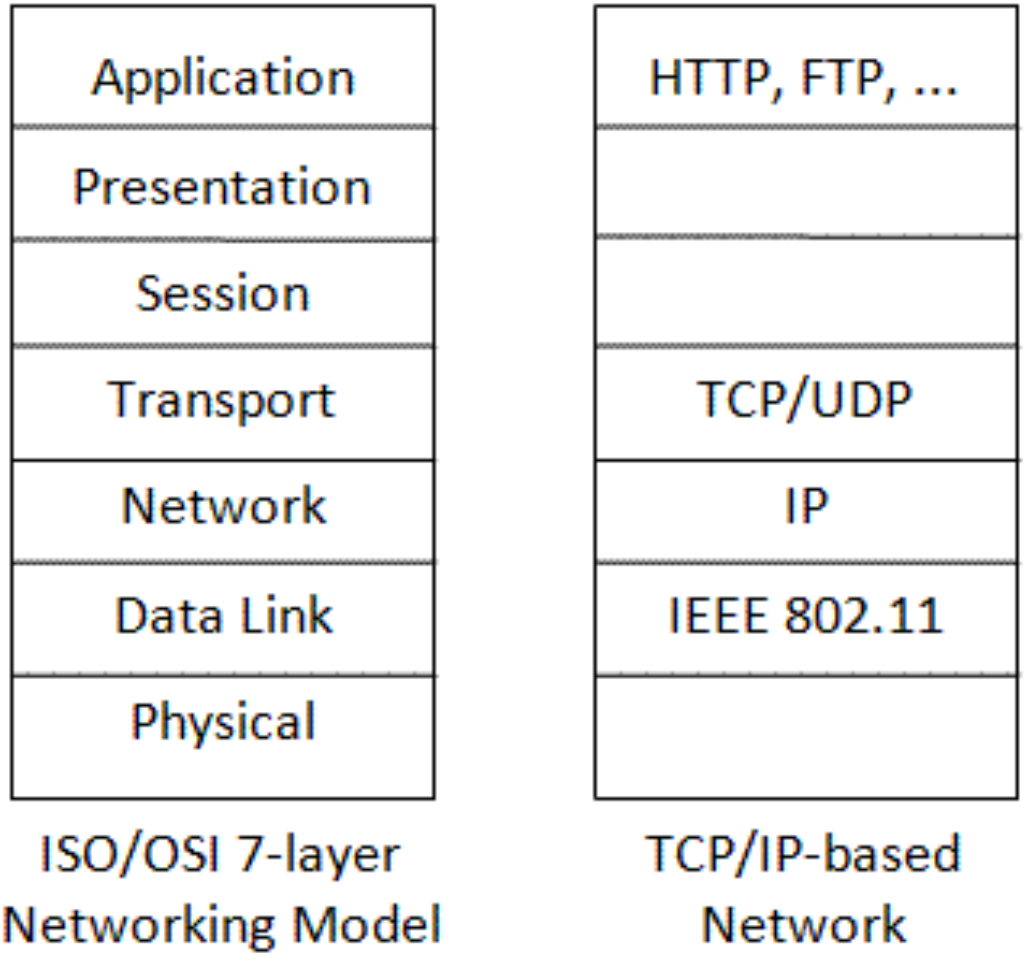


Protocolli e stratificazione (ISO/OSI vs. TCP/IP)

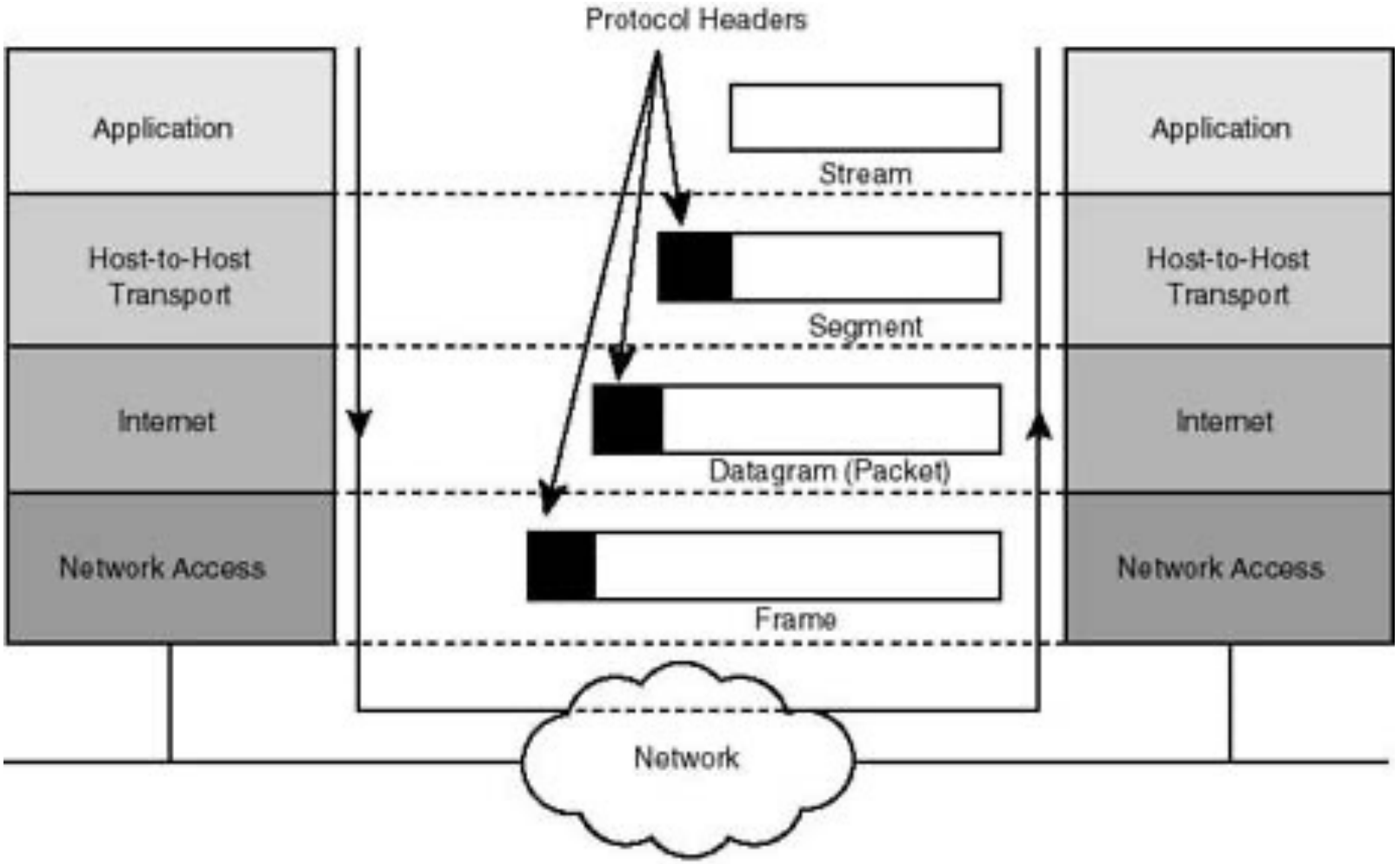


TCP/IP and the OSI model

Protocolli e stratificazione (ISO/OSI vs. TCP/IP)



TCP/IP headers



Identificazione endpoint

- Ogni qual volta due endpoint vogliono comunicare devono identificarsi univocamente
- Tale identificazione avviene attraverso due livelli di indirizzamento:
  - Il primo determina l'**host** su cui e' in esecuzione il processo
  - Il secondo determina il **processo** con cui si vuole comunicare

Identificazione del Server

- Ogni qual volta due endpoint vogliono comunicare devono identificarsi univocamente
- Tale identificazione avviene attraverso due livelli di indirizzamento:
  - Il primo determina l'host su cui e' in esecuzione il processo
  - Ad ogni host di una rete IP e' associato un indirizzo IP
  - Il secondo determina il processo con cui si vuole comunicare

Indirizzi IP

- Un Indirizzo IP è un numero che identifica univocamente un dispositivo collegato ad una rete informatica che utilizza lo standard IP (Internet Protocol)
- Gli indirizzi IPv4 sono costituiti da 32 bit (4 byte), e vengono descritti con 4 numeri decimali rappresentati su 1 byte (quindi ogni numero varia tra 0 e 255) separati da un punto

Indirizzi IP

- Un esempio di indirizzo IPv4 è il seguente: 192.167.11.34
- Questa rappresentazione limita lo spazio di indirizzamento a 4,294,967,296 indirizzi univoci possibili
- La rete internet esclude 18.000.000 indirizzi utilizzati per le reti private
- Per ovviare al problema della mancanza di indirizzi IP dovuta alla costante crescita di Internet è stato introdotto l'IPv6

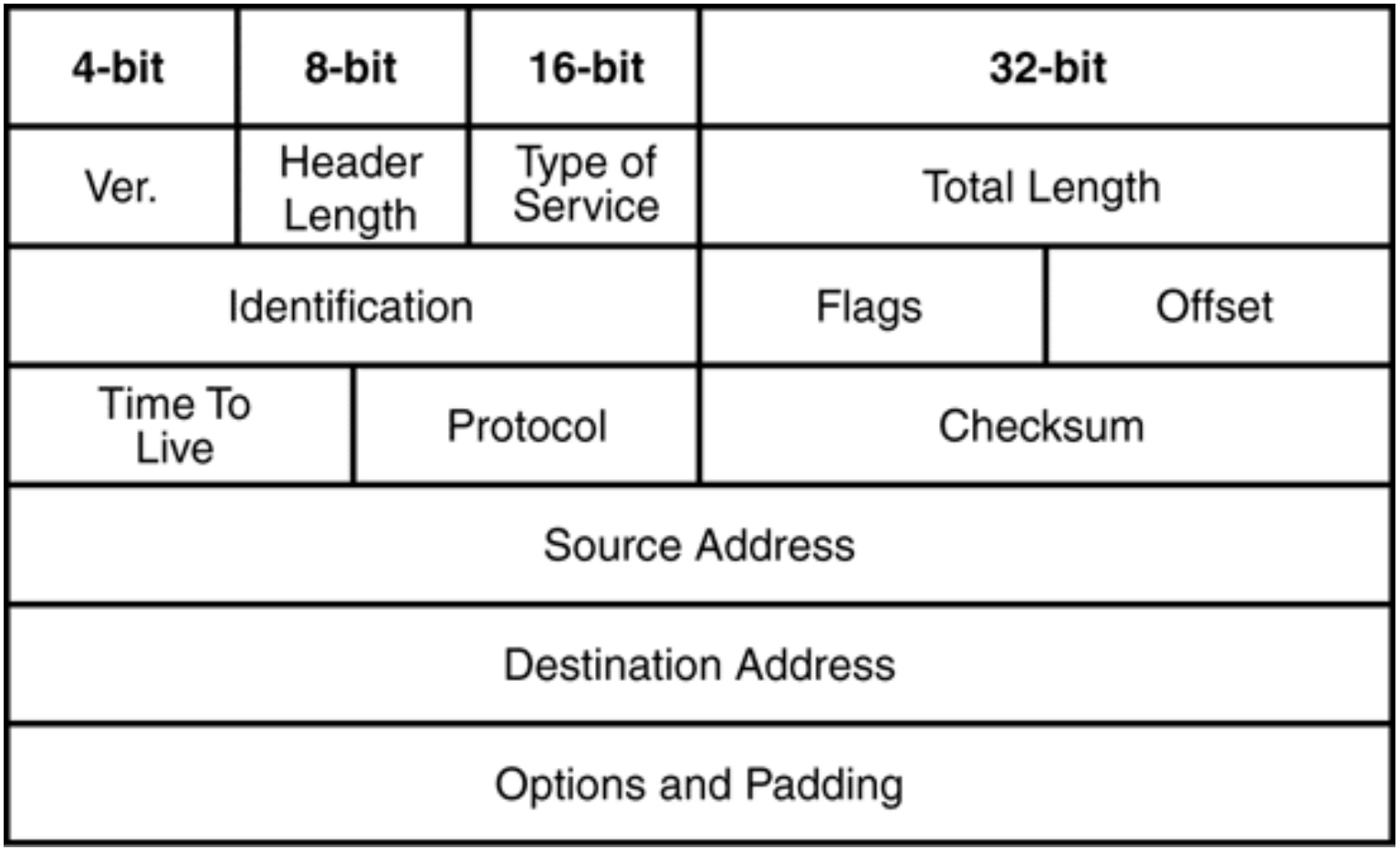
Indirizzamento IPv6

- Indirizzi lunghi 128 bit anziché 32
  - **2^{128} indirizzi**
  - **circa 10^{38} indirizzi**
- Più precisamente
  - **340.282.366.920.938.463.463.374.607.431.768.211.456 indirizzi**
- Alcune stime:
  - **superficie della terra: 511.263.971.197.990 mq**
  - **655.570.793.348.866.943.898.599 indirizzi IPv6 per mq**

Indirizzamento IPv6

- L'indirizzo viene suddiviso in 8 blocchi di 16 bit ciascuno. I blocchi sono separati da ":" e vengono rappresentati in notazione esadecimale
  - **`3ffe:1001:0001:0100:0a00:20ff:fe83:5531`**
  - **`3ffe:1001:0001:0000:0000:0000:0000:0001`**
- Esistono delle semplificazioni:
  - **si possono omettere gli zeri iniziali**
    - **`3ffe:1001:1:100:a00:20ff:fe83:5531`**
  - **si possono sostituire gruppi di zeri con "::"**
    - **`3ffe:1001:1::1`**
- Gli indirizzi IPv6 compatibili IPv4 si scrivono:
  - **`::163.162.170.171`**

IP header


4-bit
8-bit
16-bit
32-bit
Ver.
Header Length
Type of Service
Total Length
Identification
Flags
Offset
Time To Live
Protocol
Checksum
Source Address
Destination Address
Options and Padding

Identificazione del server

- Ogni qual volta due endpoint vogliono comunicare devono identificarsi univocamente
- Tale identificazione avviene attraverso due livelli di indirizzamento:
  - Il primo determina l'host su cui e' in esecuzione il processo
  - Ad ogni host di una rete IP e' associato un indirizzo IP
  - Il secondo determina il processo con cui si vuole comunicare
  - Ad ogni applicazione in esecuzione su un host e' associato un numero di porta

Numeri di Porta

- In un ambiente multitasking più processi in esecuzione su uno stesso host devono poter comunicare mediante lo stesso sottosistema di rete
- E' necessario consentire più connessioni simultaneamente
- Per poter tenere distinte le diverse connessioni su uno stesso host si utilizzano le porte

Numeri di Porta

Le porte sono interi a 16 bit da 0 a 65535

- Da 0 a 1023: porte riservate (ai processi di root)
- Da 1024 a 49151: porte registrate
- Da 49152 a 65535: porte effimere (per i client, ai quali non interessa scegliere una porta specifica)

Porte Riservate

- Esempi di porte riservate
- 21 ftp (trasferimento file)
- 22 ssh (login remoto sicuro)
- 25 smtp (invio email)
- 80 http (web)
- 143 imap (lettura email)
- Lista ufficiale su: http://www.iana.org/
- La corrispondenza tra nomi simbolici e numeri si trova nel file /etc/services

TCP e UDP

- I due principali protocolli relativi al livello di trasporto sono
  - TCP (Transport Control Protocol)
  - UDP (User Datagram Protocol);

- TCP
  - servizio con connessione
  - comunemente usato per i servizi a livello applicazione:
    - Telnet
    - HTTP
    - SMTP
    - FTP-data

- UDP
  - usato se le due parti A e B utilizzano un'interazione del tipo richiesta/risposta
  - per la sincronizzazione o il controllo del traffico dati (FTP)
  - oppure per ragioni di efficienza (NFS)

UDP

• UDP (User Datagram Protocol) è un protocollo standard, descritto nella RFC 768

• E' presente in ogni implementazione TCP/IP che non sia utilizzata esclusivamente per il routing

• UDP rappresenta un'interfaccia ad IP per le applicazioni
  • realizza un meccanismo di multiplexing e di demultiplexing
  • per l'invio e la ricezione dei datagram ai processi
  • grazie al concetto di porta

• E' un protocollo molto semplice, con basso overhead

UDP

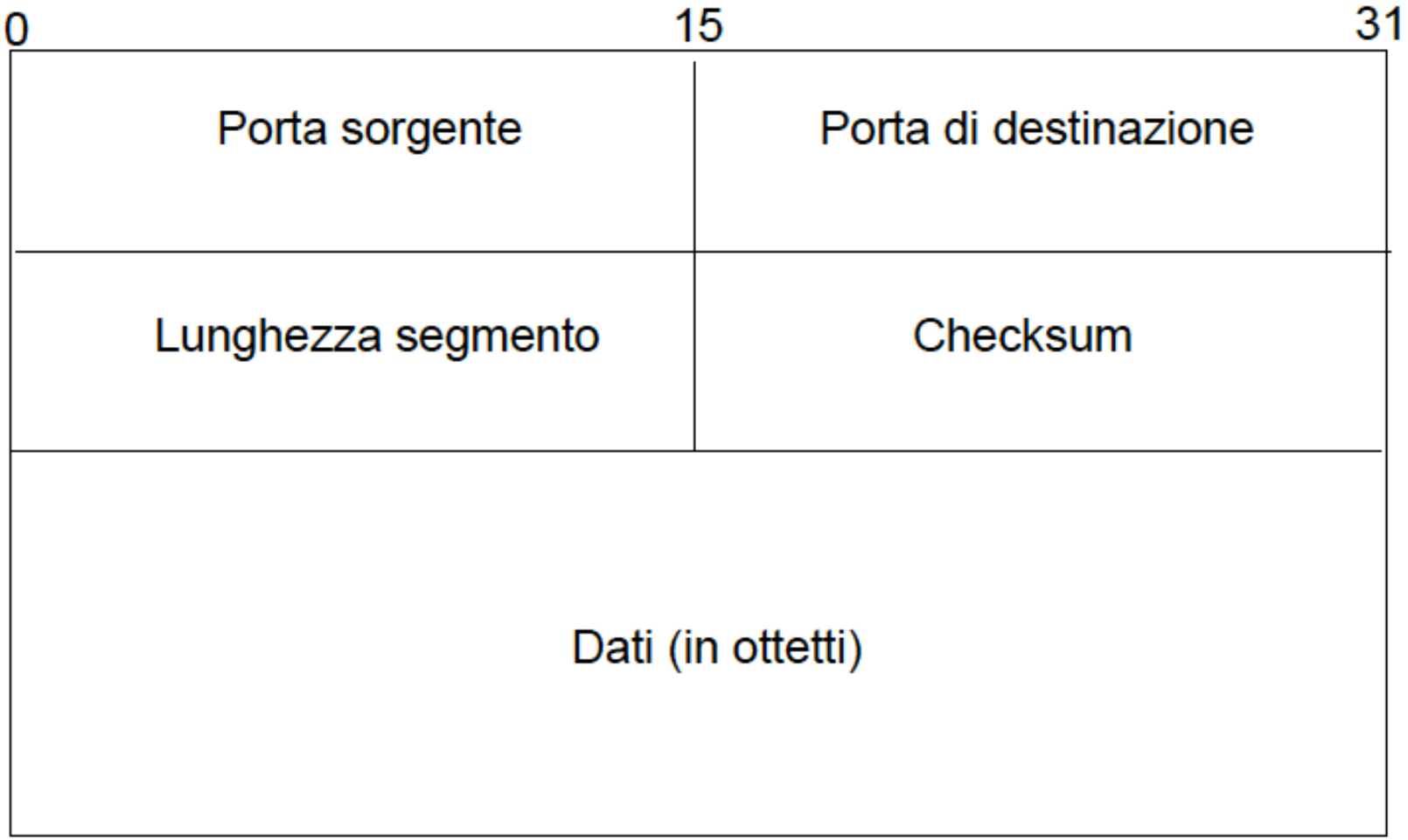
- Ogni datagramma UDP è inviato in un singolo datagramma IP
  - quest'ultimo può venire frammentato durante una trasmissione
  - viene riassemblato dall'IP ricevente
  - prima di essere presentato al livello UDP
- Tutte le implementazioni IP devono accettare datagrammi di almeno 576 byte
  - accetta datagrammi UDP di almeno 516 byte
  - header IP è di al più 60 byte
- E' compito dell'applicazione
  - suddividere i dati inviati in datagrammi UDP di lunghezza appropriata
  - riassemblare i datagram ricevuti

UDP

- Un datagram UDP consiste in un header di 8 byte (64 bit), seguito dai dati

- Il preambolo contiene quattro campi ciascuno della lunghezza di 2 byte (16 bit):
  1. porta sorgente (sprt) intero a 16 bit (da 0 a 65535)
  2. porta destinazione (dprt) intero a 16 bit (da 0 a 65535)
     - indicanti il punto di accesso nell'host sorgente e nell'host di destinazione;
  3. lunghezza UDP
     - espressa in byte
     - comprende sia il preambolo che i dati
  4. checksum UDP
     - stringa di 16 bit
     - utilizzata per il controllo degli errori

Datagramma UDP



TCP

• TCP (Transmission Control Protocol) è un protocollo standard descritto nella RFC 793;

• Come per UDP, TCP è in pratica presente in ogni implementazione TCP/IP che non sia utilizzata esclusivamente per il routing

• TCP è molto più complesso di UDP

• Fornisce un analogo meccanismo di multiplexing e di de-multiplexing

• TCP offre a livello applicazione le seguenti funzionalità
  - **trasferimento dati a flusso**
  - **affidabilità**
  - **controllo del flusso**
  - **trasferimenti full-duplex**.

TCP

- Dal punto di vista dell'applicazione
  - TCP trasferisce un flusso continuo di byte lungo la rete
  - l'applicazione non si deve occupare di suddividere i dati in datagrammi
  - TCP raggruppa i bytes di dati in segmenti, passandoli poi al livello rete

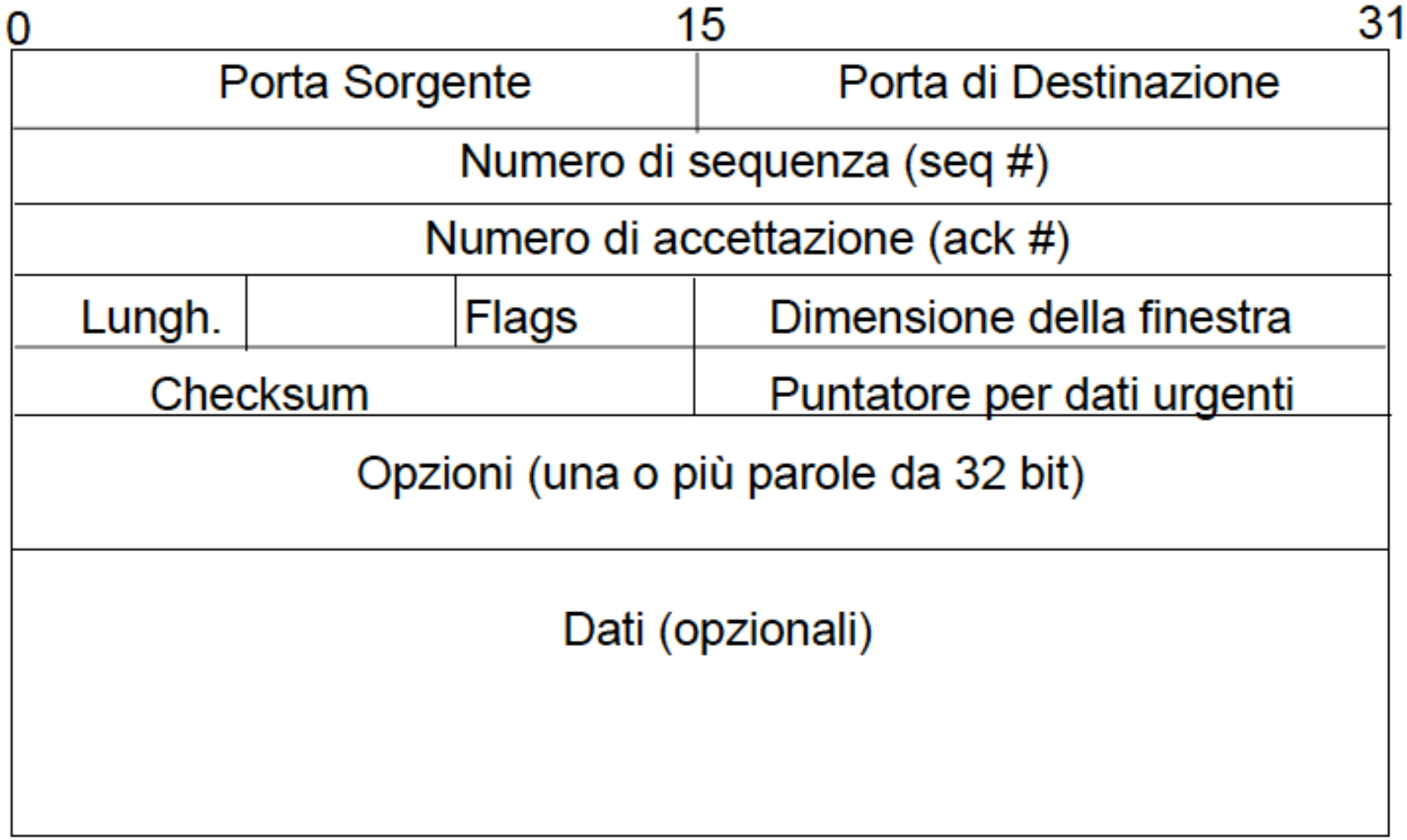
- TCP stabilisce (entro certi limiti) come segmentare ed inviare i dati al livello rete

- TCP assegna un numero di sequenza ad **ogni byte trasmesso**, attendendo una conferma di avvenuta ricezione (**ACK**) da parte del TCP di destinazione

- Il TCP ricevente, quando restituisce un ACK al mittente, invia anche il **numero massimo di sequenza** che può ricevere nella prossima trasmissione

- Un segmento consiste in un preambolo di 20 byte, più un campo opzionale di 4 byte, seguito da zero o più byte di dati;

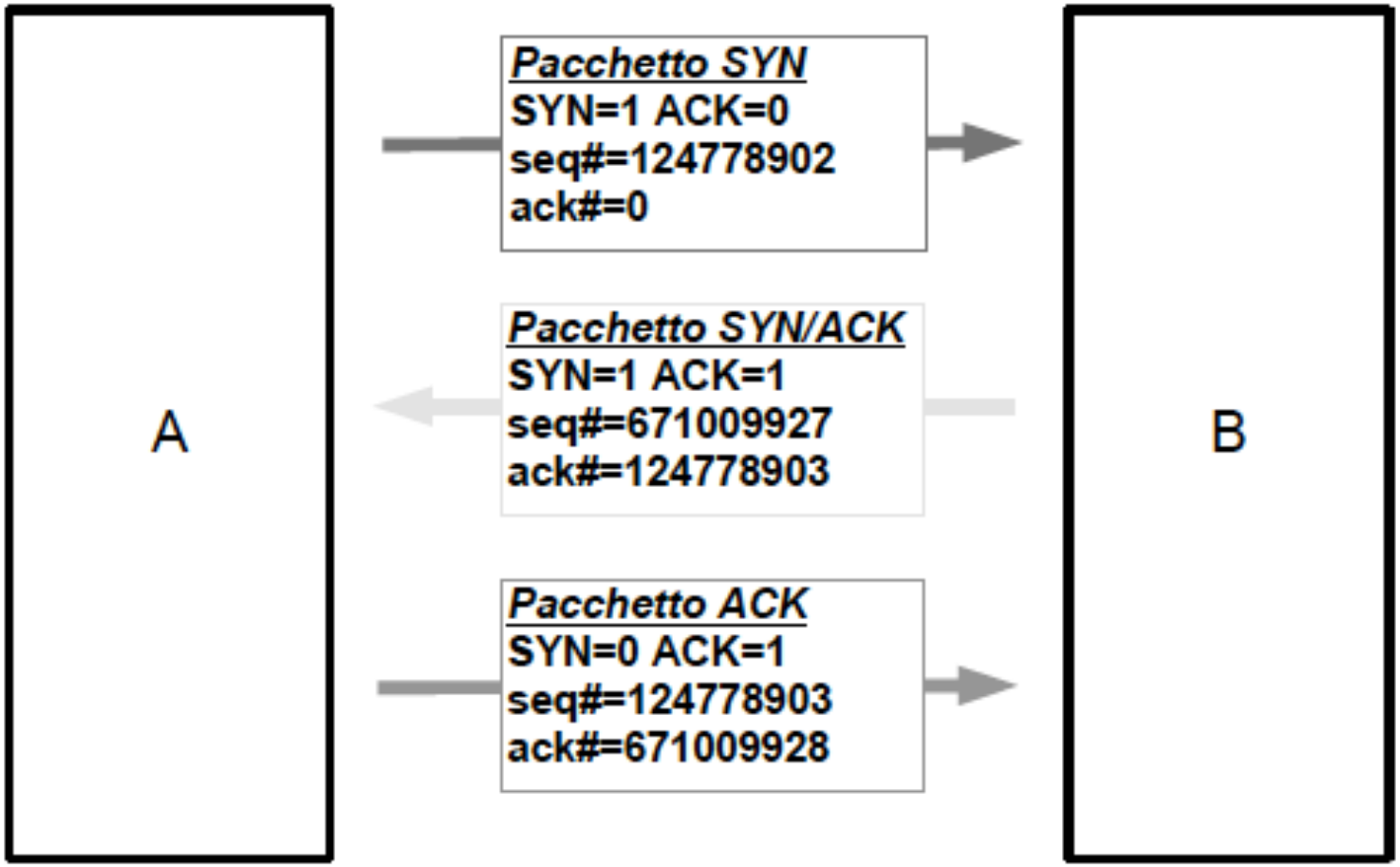
Segmento TCP



Three way handshake

- Quando un processo A sull'host C desidera aprire una connessione TCP con un processo B sull'host S:
  - C genera un numero di sequenza iniziale N ed invia a S un segmento TCP in cui il flag SYN è impostato su 1, il flag ACK su 0 ed i campi seq# e ack# sono impostati ad N ed a 0, rispettivamente
  - S genera un numero di sequenza iniziale M e risponde con un segmento in cui i flag SYN e ACK sono entrambi 1, mentre seq#=M e ack#=N+1
  - Infine, C invia ad S un segmento in cui SYN=0, ACK=1, seq#=N+1 e ack#=M+1
- Dopo tali tre fasi la connessione è stabilita, ed i due processi A e B possono scambiare dati

Three way handshake



Modello Client-Server

- Nel modello client-server si distinguono due entità
  - I programmi che forniscono un servizio, chiamati server
  - I programmi di utilizzo, detti client che effettuano le richieste

- Un server può (di norma deve) essere in grado di rispondere a più di un client

Modello Client-Server

- Distinguiamo due classi di server:
  - concorrenti
  - Iterativi

- Seguono questo modello tutti i servizi fondamentali di internet:
  - http, ftp, telnet, ssh, etc.

Identificare una Comunicazione

- La classica implementazione di un server concorrente prevede ad ogni nuova richiesta client venga generato un processo che la gestisce
- Come si distinguono processi server che forniscono uno stesso servizio a client diversi se utilizzano la stessa porta per comunicare e sono in esecuzione sullo stesso host?

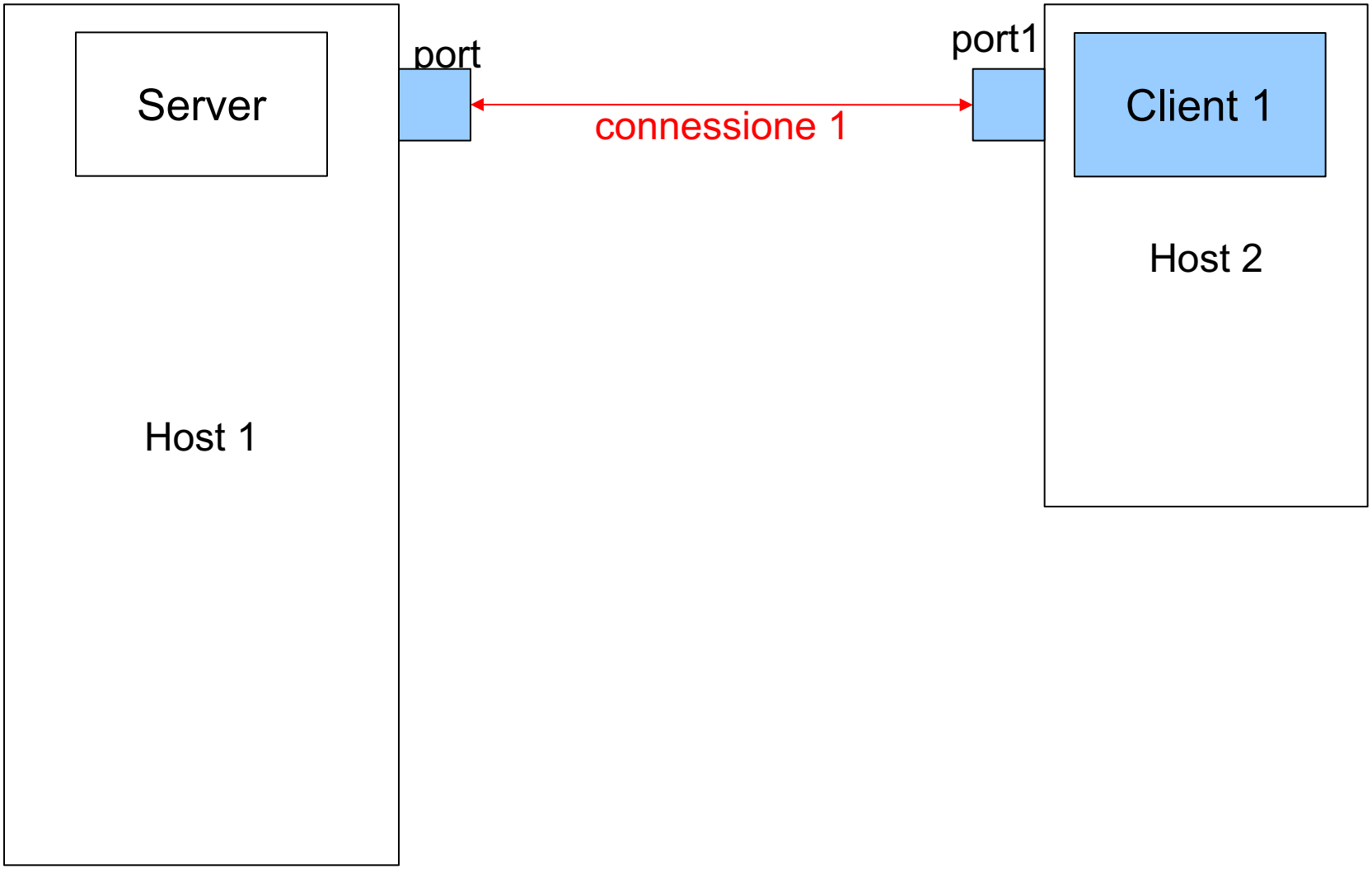
Identificare una Comunicazione

- TCP e UDP usano 4 informazioni per identificare una comunicazione
  - Indirizzo IP del server
  - Numero di porta del servizio lato server
  - Indirizzo IP del client
  - Numero di porta del servizio lato client

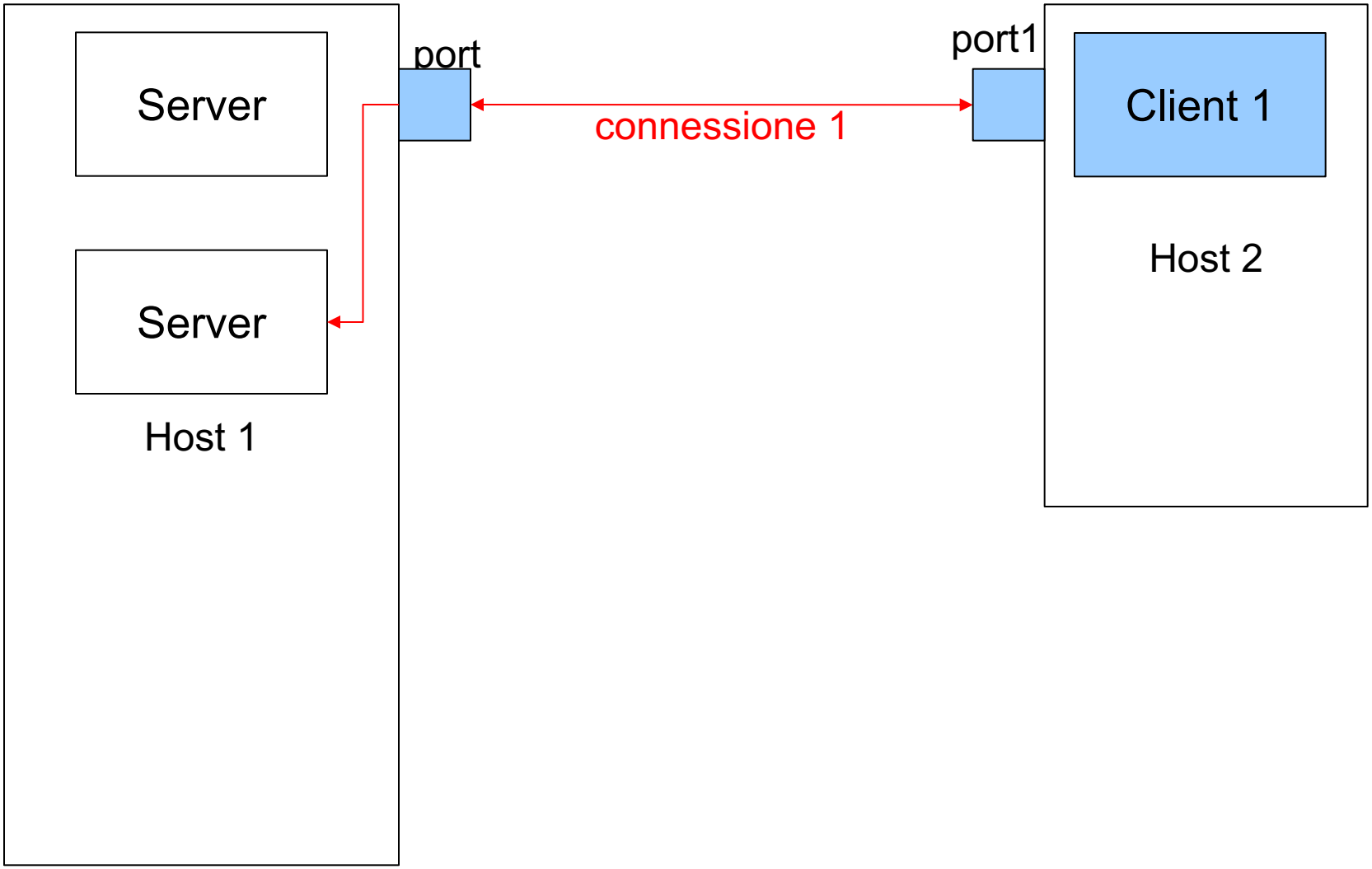
Endpoint

- Un endpoint è una coppia
  - (indirizzo IP, porta)
- Una connessione è una coppia di endpoints
  - (endpoint sorgente,endpoint destinazione)

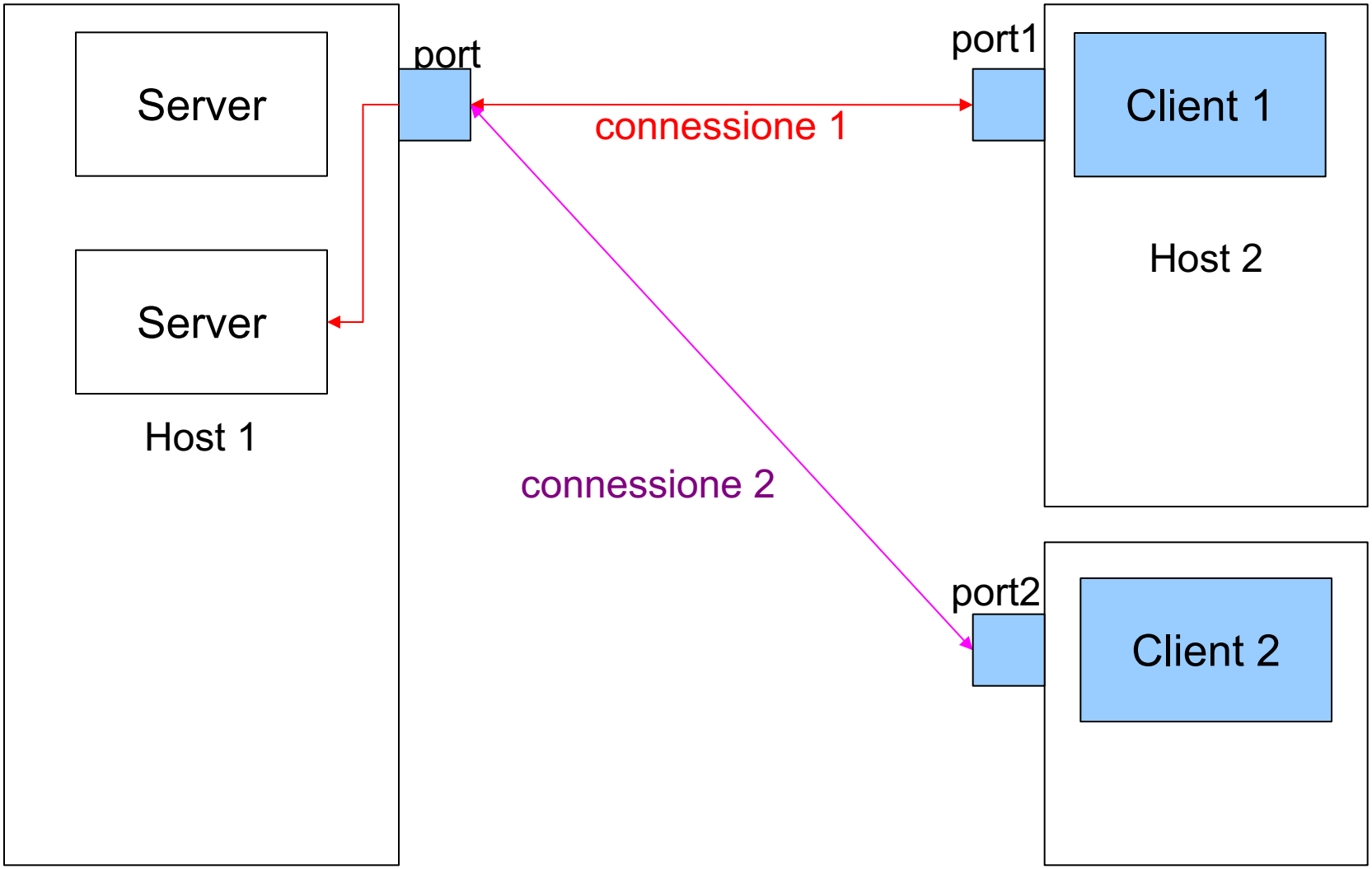
Server Concorrenti



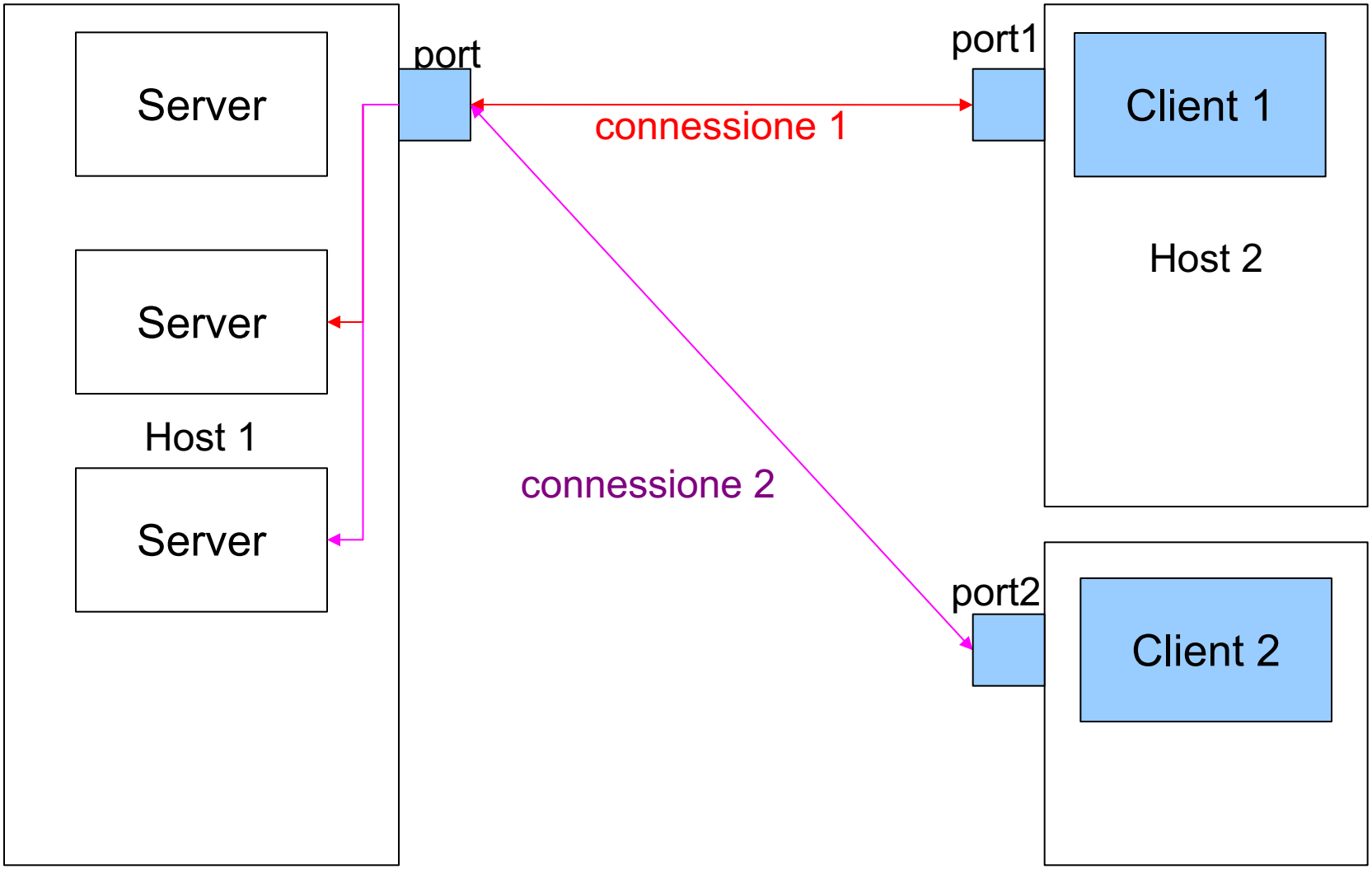
Server Concorrenti



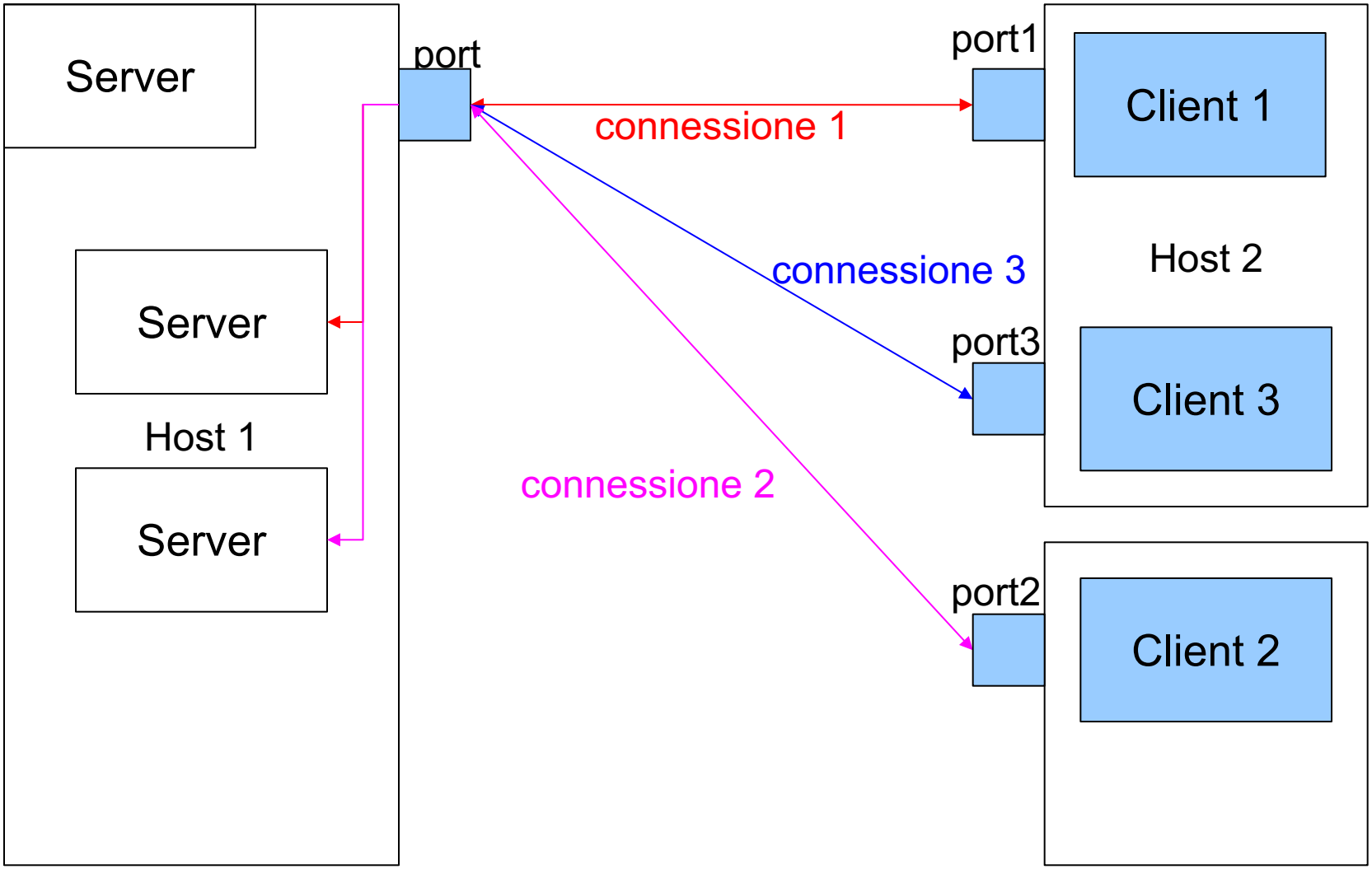
Server Concorrenti



Server Concorrenti



Server Concorrenti



Server Concorrenti