

RETI di CALCOLATORI

Parte III

SOMMARIO

1. Architettura di Rete: livello transport

- **TCP**
- **UDP**

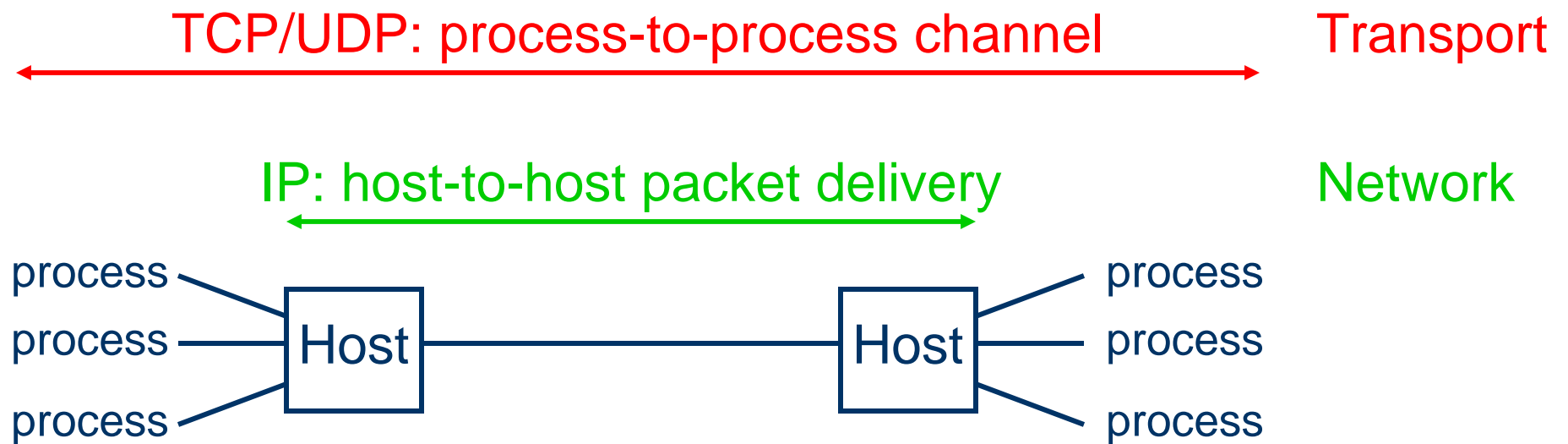
2. Architettura di Rete: livello application

- **DNS**
- **E-Mail**
- **WWW**

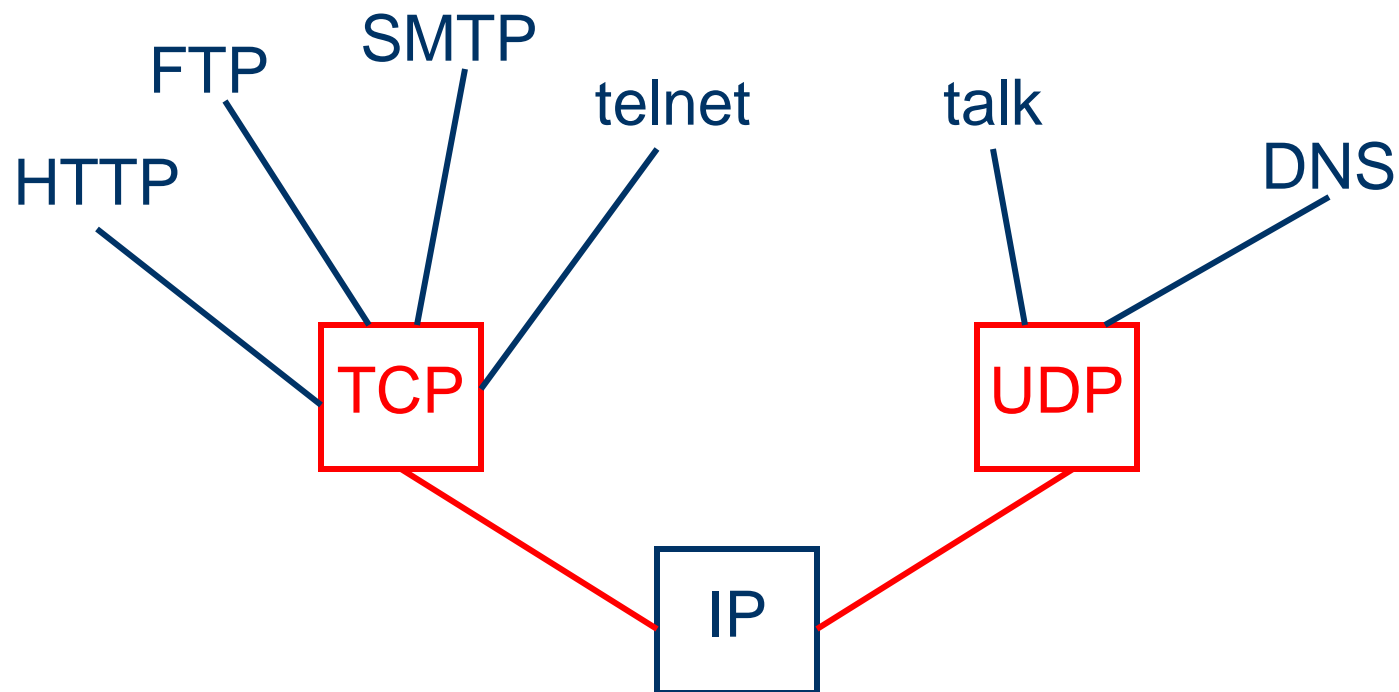
LIVELLO TRANSPORT

TCP / UDP

Livello Transport



Protocolli di livello transport: **TCP** e **UDP**



TCP (Transmission Control Protocol):

Aggiunge al protocollo IP le seguenti funzionalità:

- **error detection** sui dati
- **demultiplexing** verso protocolli di livello application
- **reliability**: assicura l'arrivo ordinato e corretto dei dati

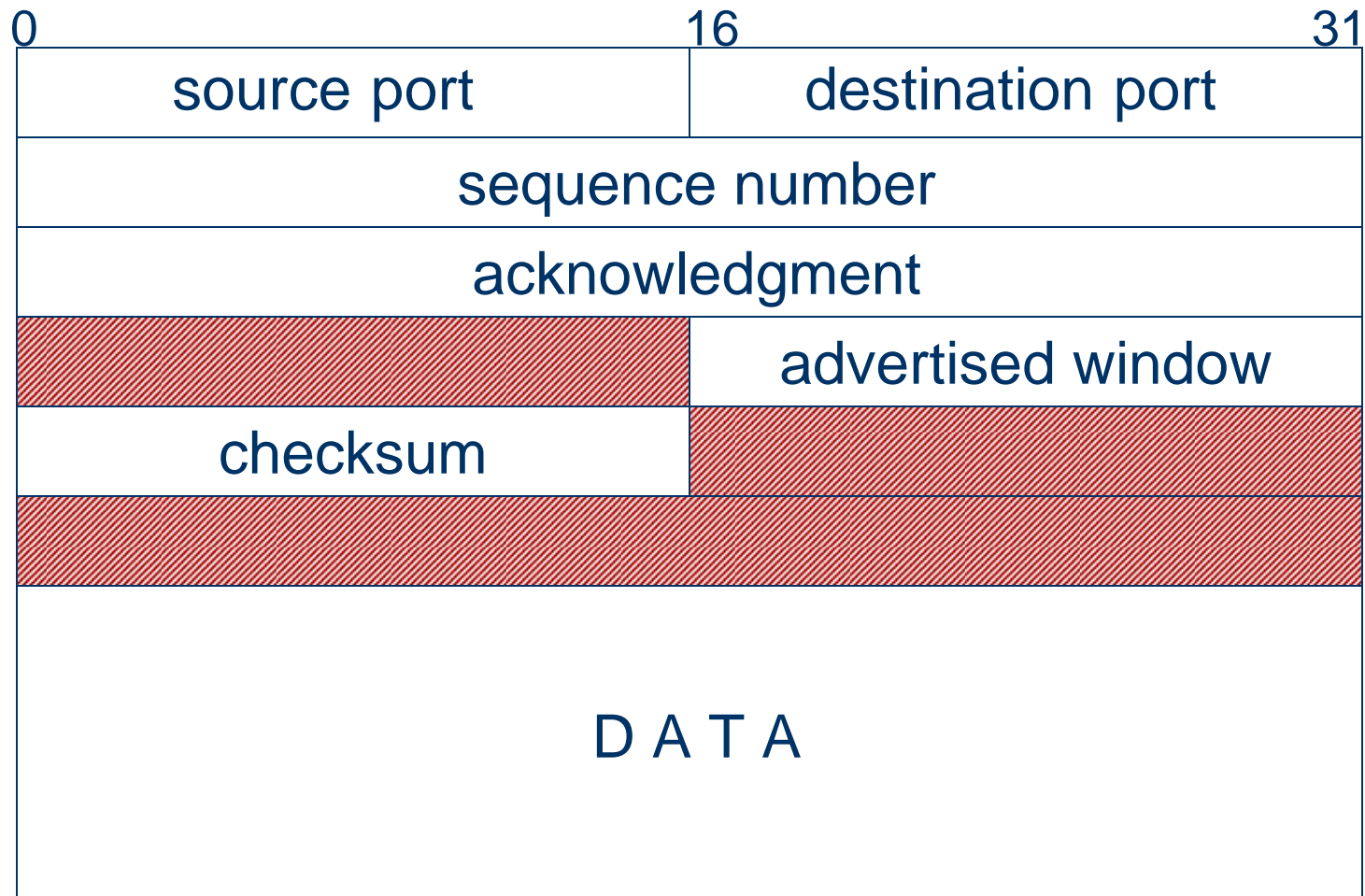
e non pone limiti sulla quantità di dati comunicati

Service Model:

- **connection-oriented**
- **full-duplex**
- **buffered transfer**
- **SLIDING WINDOW MECHANISM**
 - invio ordinato e affidabile di sequenze di byte
 - flow control
 - congestion control

Livello Transport: TCP

Segmento TCP



Livello Transport: TCP

port: numero intero che insieme all' indirizzo IP identifica il processo mittente o destinatario della comunicazione

sequence number: indice del primo byte di dati portato dal segmento (rispetto alla sequenza originale)

acknowledgment: prossimo byte atteso

checksum: rilevamento di errori fatto su tutto il segmento e su alcuni campi dell'header del pacchetto IP (es. indirizzo IP source e destination)

1. Connection-oriented & full-duplex

Fasi della connessione:

Connection establishment

(handshake)



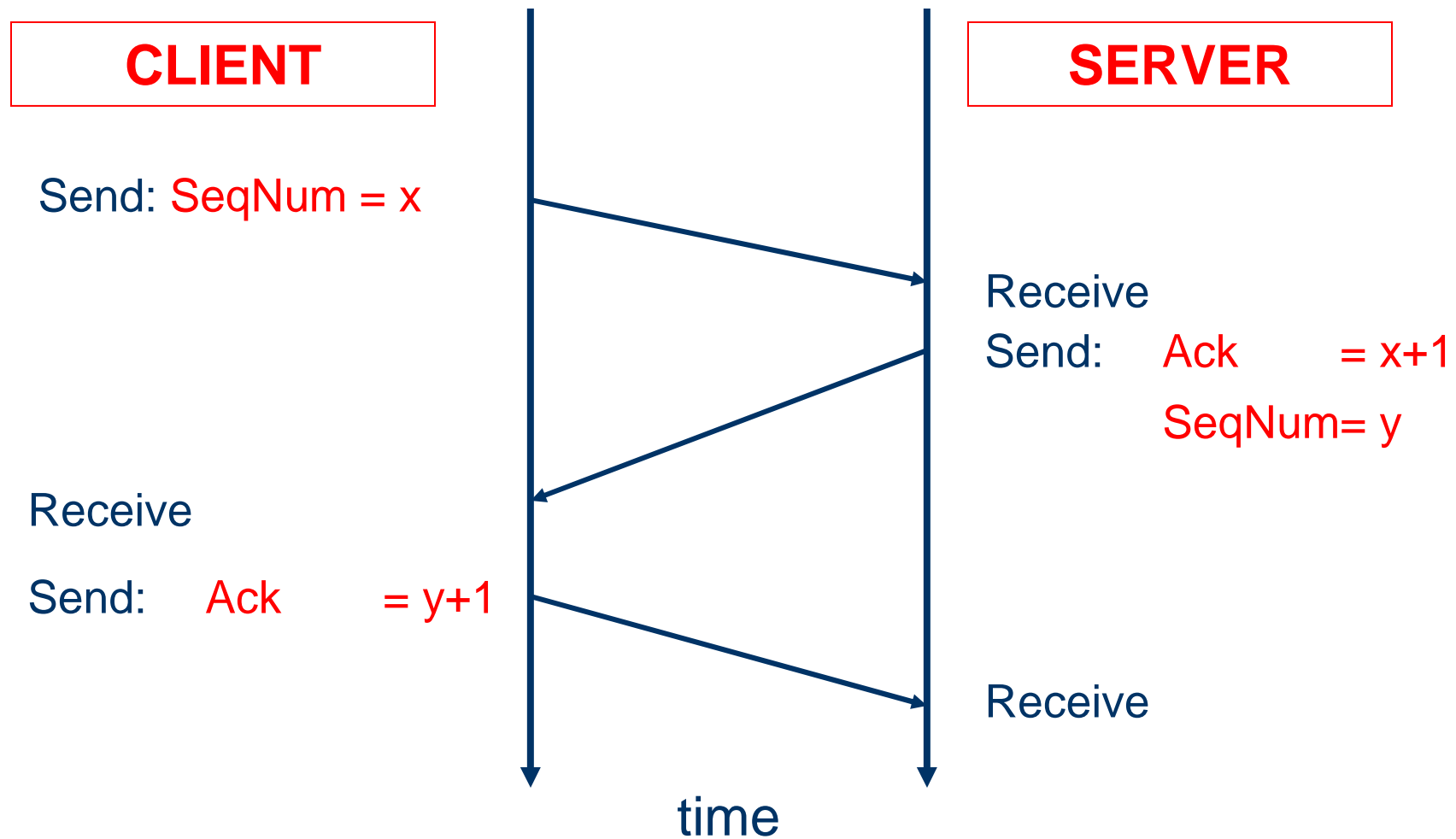
Full-duplex data exchange



Connection termination

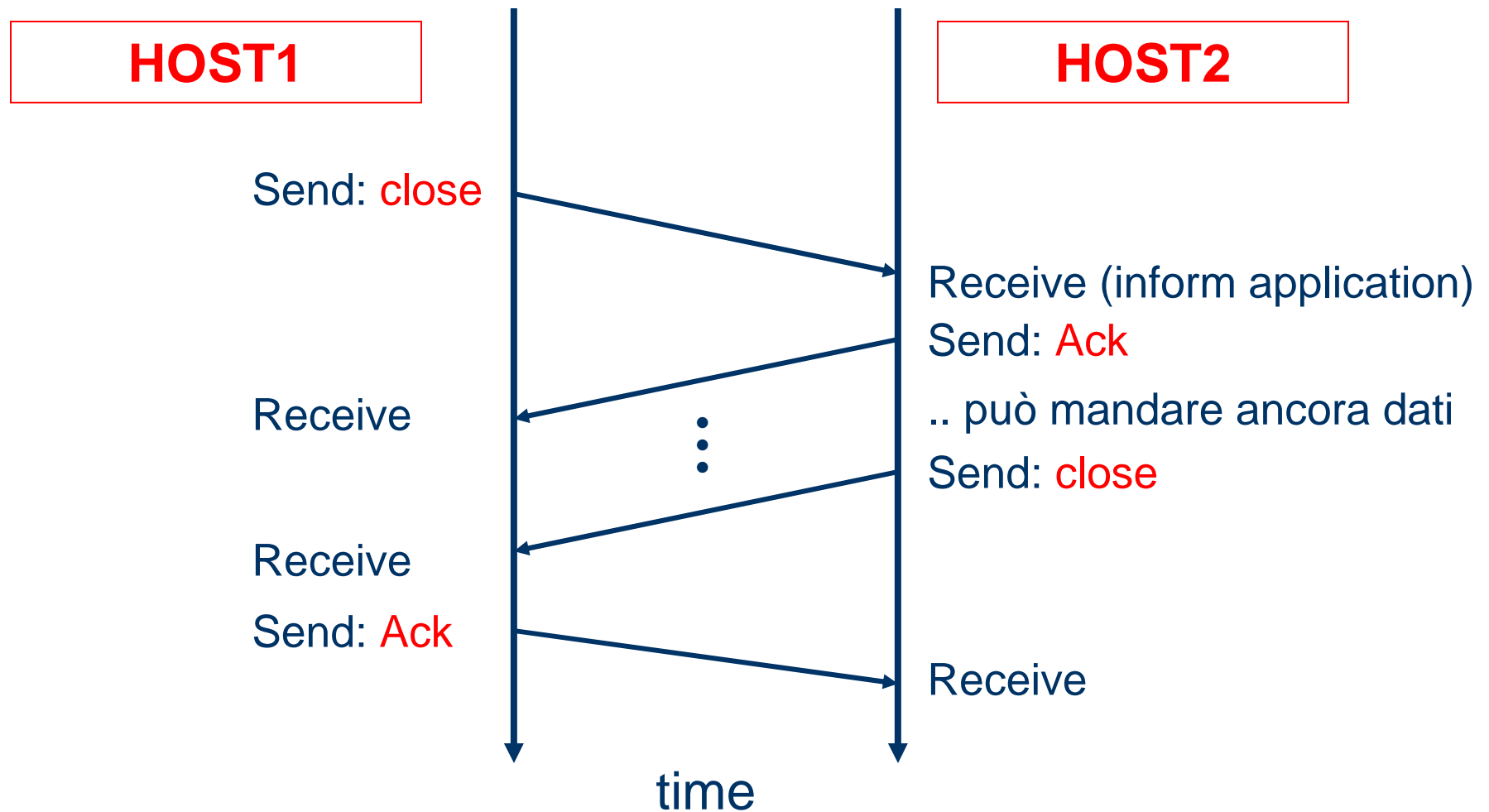
(handshake)

Connection establishment

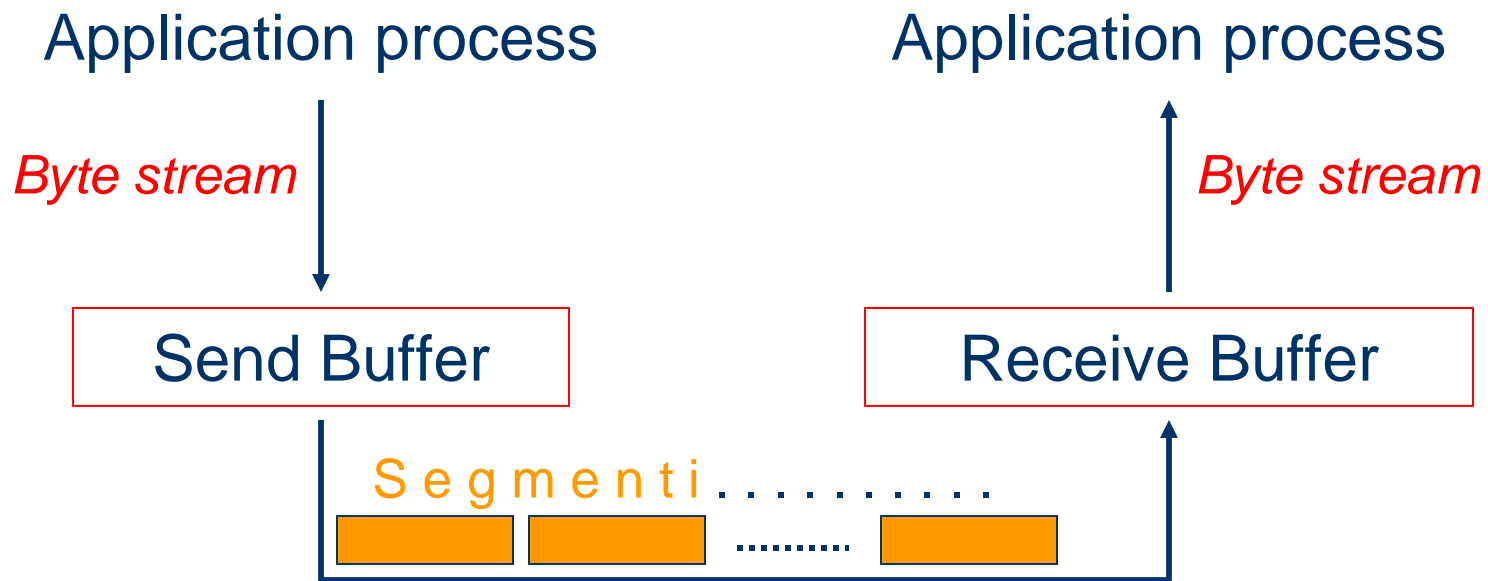


Osservazione: le sequenze dati del client e del server iniziano da byte di indice random, rispettivamente **x** e **y**, per ridurre la probabilità di sovrapposizioni con comunicazioni precedenti

Connection termination



2. Buffered transfer



Livello Transport: TCP

I byte accumulati nel *send buffer* sono incapsulati in un segmento e inviati quando:

- raggiungono la **Maximum Segment Size** (MSS)

$$\text{MSS} = \text{MTU} - |\text{IP Header}| - |\text{TCP Header}|$$

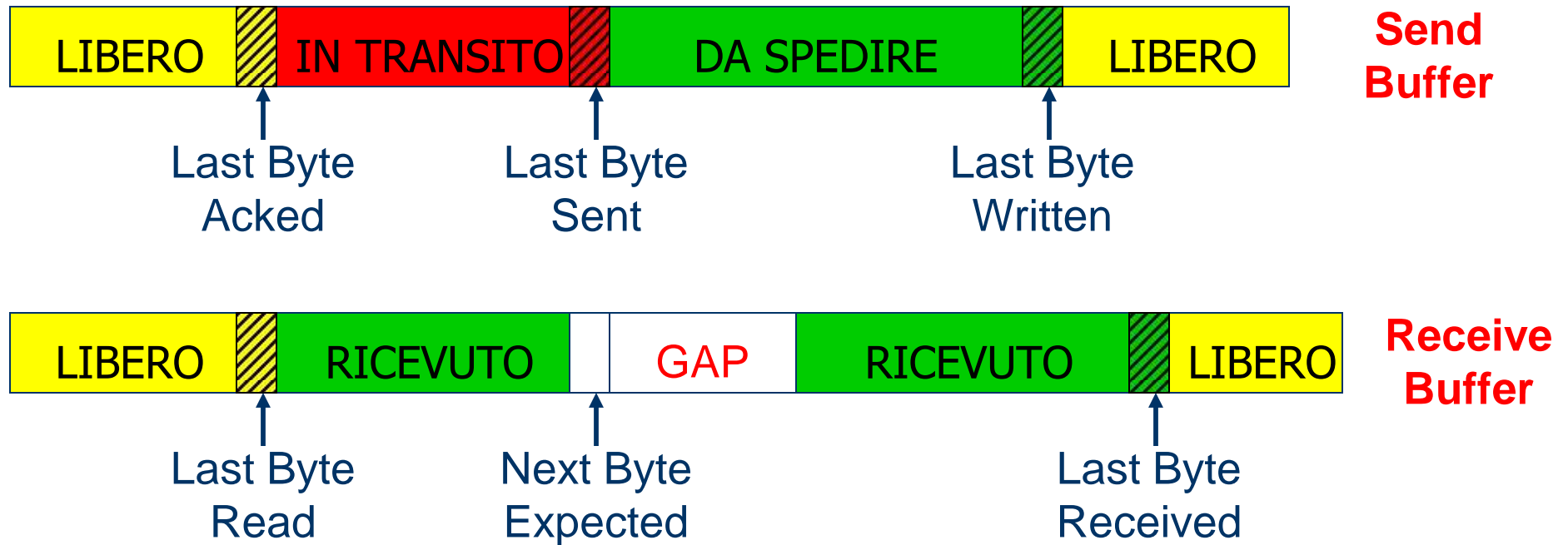
- esplicitamente richiesto dal sender
- dopo un certo tempo

3. SLIDING WINDOW

Il meccanismo di sliding window garantisce:

- **reliable delivery:** il sender riceve acknowledgment cumulativi per i byte inviati. Ogni segmento i cui byte non sono stati “acknowledged” entro un certo TIMEOUT viene rispedito.
- **flow control:** il ricevente regola il flusso di dati in arrivo tramite la *advertised window*
- **congestion control:** evita che il sender invii troppi dati quando il ricevente non può riceverli

Livello Transport: TCP



Per ogni segmento arrivato il ricevente invia un segmento di risposta con

Acknowledgment = Next Byte Expected

Advertised Window = |Receive Buffer| - (L.B.Rcvd - L.B.Read)

Livello Transport: TCP

Numero massimo di dati inviabili:

Effective Window = Advertised Window - (L.B.Sent - L.B.Acked)

Osservazioni:

1. Il campo acknowledgment nel segmento preparato dal ricevente non riflette necessariamente l'ultimo segmento di dati ricevuto
2. Se Advertised Window = 0 il sender continua a inviare segmenti di 1 byte che costringono il ricevente a inviare segmenti di con valori di A.W. aggiornati, sino a quando $A.W. > 0$

UDP (User Datagram Protocol)

Aggiunge al protocollo IP le seguenti funzionalità:

- **error detection** sui dati
- **demultiplexing** verso protocolli di livello application

Service Model:

- **unreliable** (non recupera dati perduti)
- **connectionless**

APPLICAZIONI di RETE

DNS / E-mail / WWW

Applicazioni di Rete

Paradigma **CLIENT-SERVER** :

- un programma **server** attende passivamente di essere contattato, mentre un programma **client** inizia attivamente la comunicazione
- il **server** gira su una macchina potente (classe server) e offre un *singolo servizio* a più client
- il **client** è invocato dall'utente sulla propria macchina (es. PC) e può richiedere *diversi servizi* ad diversi server

Applicazioni

API (Application Program Interface): insieme di procedure standard utilizzate da un programma applicativo (server o client). Ad esempio (procedure di interazione con il sw di rete):

- indicare il protocollo di livello transport da usare
- mettersi in attesa di un collegamento (server)
- iniziare un collegamento (client)
- specificare lo spazio di memoria in cui risiedono i dati da spedire o ricevuti

Es. Socket API (la più utilizzata)

DNS (Domain Name System)

- Base di dati distribuita che definisce il mapping
indirizzi IP \leftrightarrow nomi mnemonici

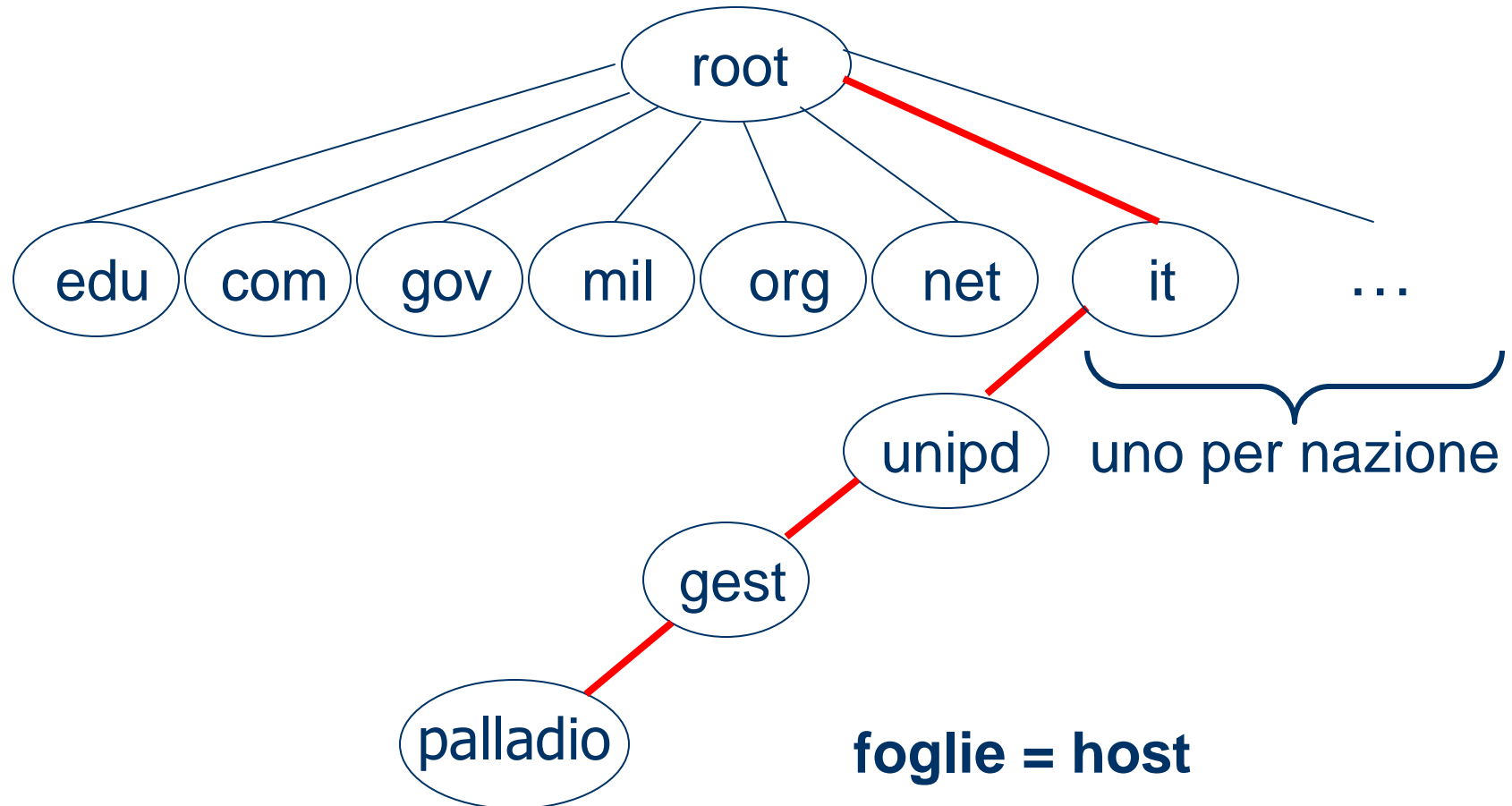
Es. 147.162.25.223 \equiv palladio.gest.unipd.it

- Resolution scheme: procedura distribuita per la traduzione dei nomi mnemonici in indirizzi IP

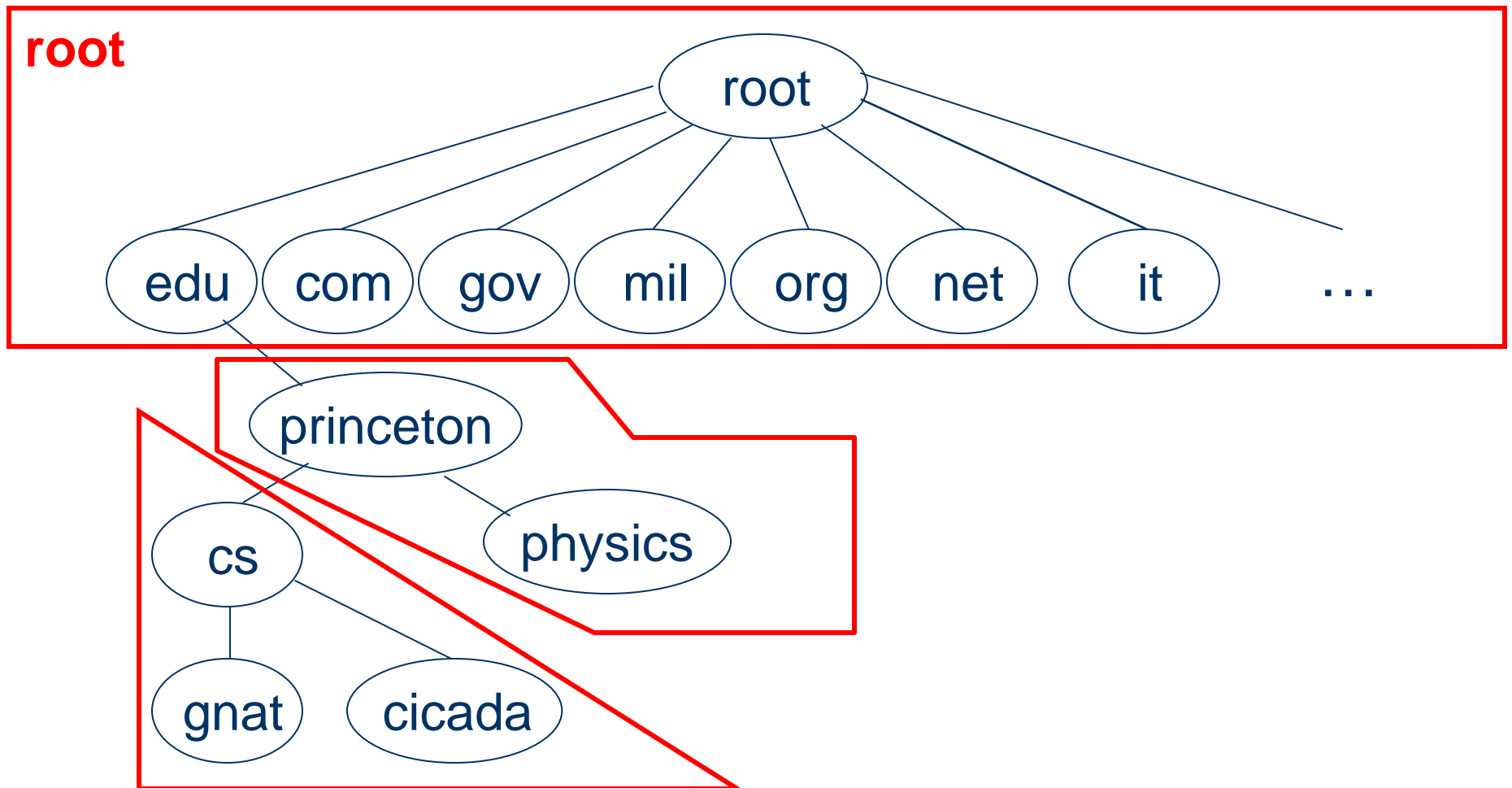
LABELs: palladio, gest, unipd, it

DOMAINs: it, unipd.it, gest.unipd.it, palladio.gest.unipd.it

Domain Hierarchy



Partizione in zone



Name Server: software che utilizza una collezione di **Resource Record** (RR) per tradurre nomi in indirizzi IP.

- gira di solito su macchine dedicate (name server).
- risponde a query inviate da client via rete

A ogni zona sono associati almeno 2 name server, ciascuno dei quali deve conoscere almeno:

- indirizzo IP di un root server
- indirizzo IP di tutti gli host della sua zona
- indirizzo IP dei name server delle zone figlie

Esempi di Resource Record:

<princeton.edu, cit.princeton.edu, NS>

<cit.princeton.edu, 128.196.128.233, A>

<www.cs.princeton.edu, cicada.cs.princeton.edu, CNAME>

<cs.princeton.edu, gnat.cs.princeton.edu, MX>

NS: Name Server

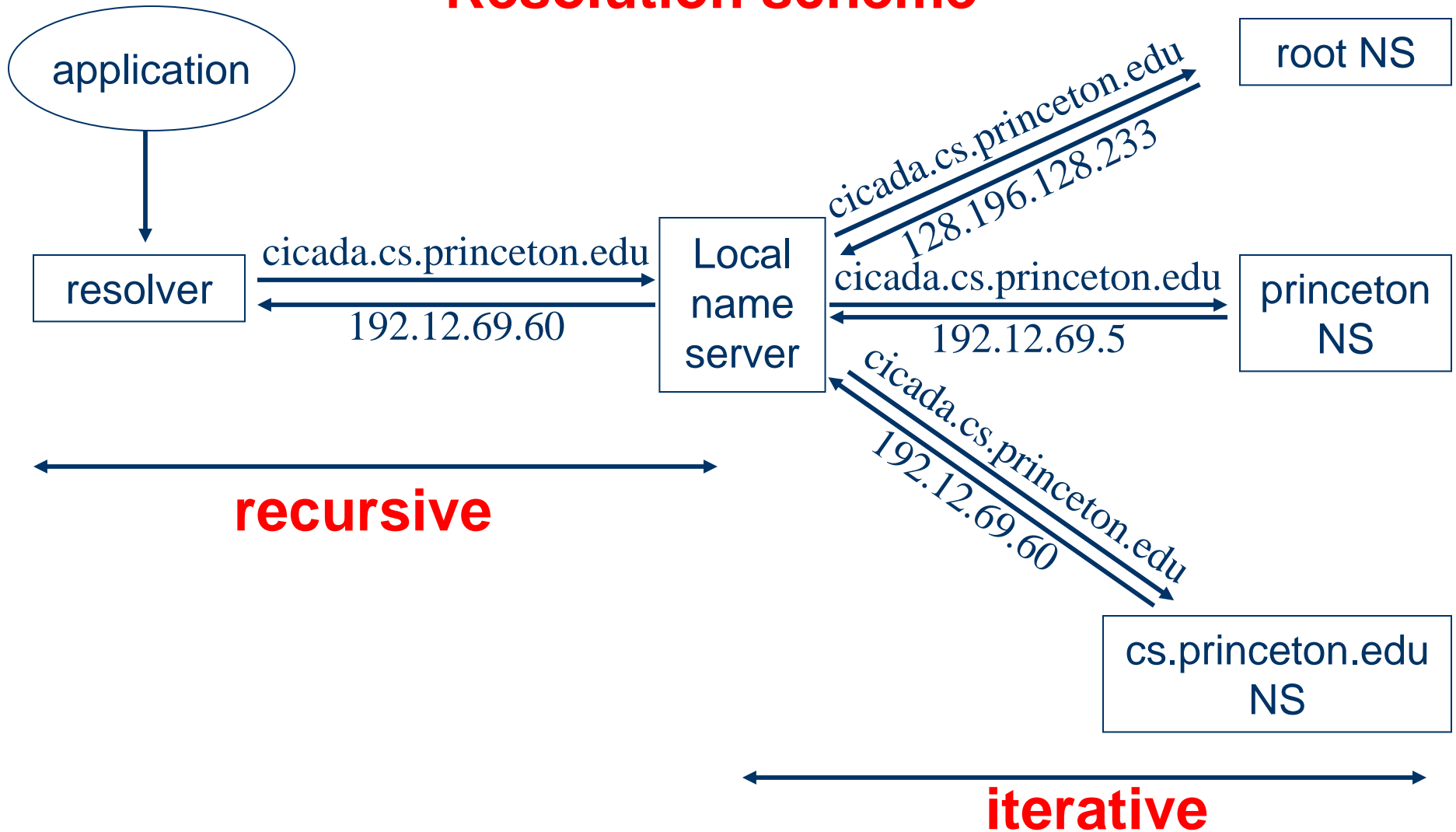
A: Address

CNAME: Alias

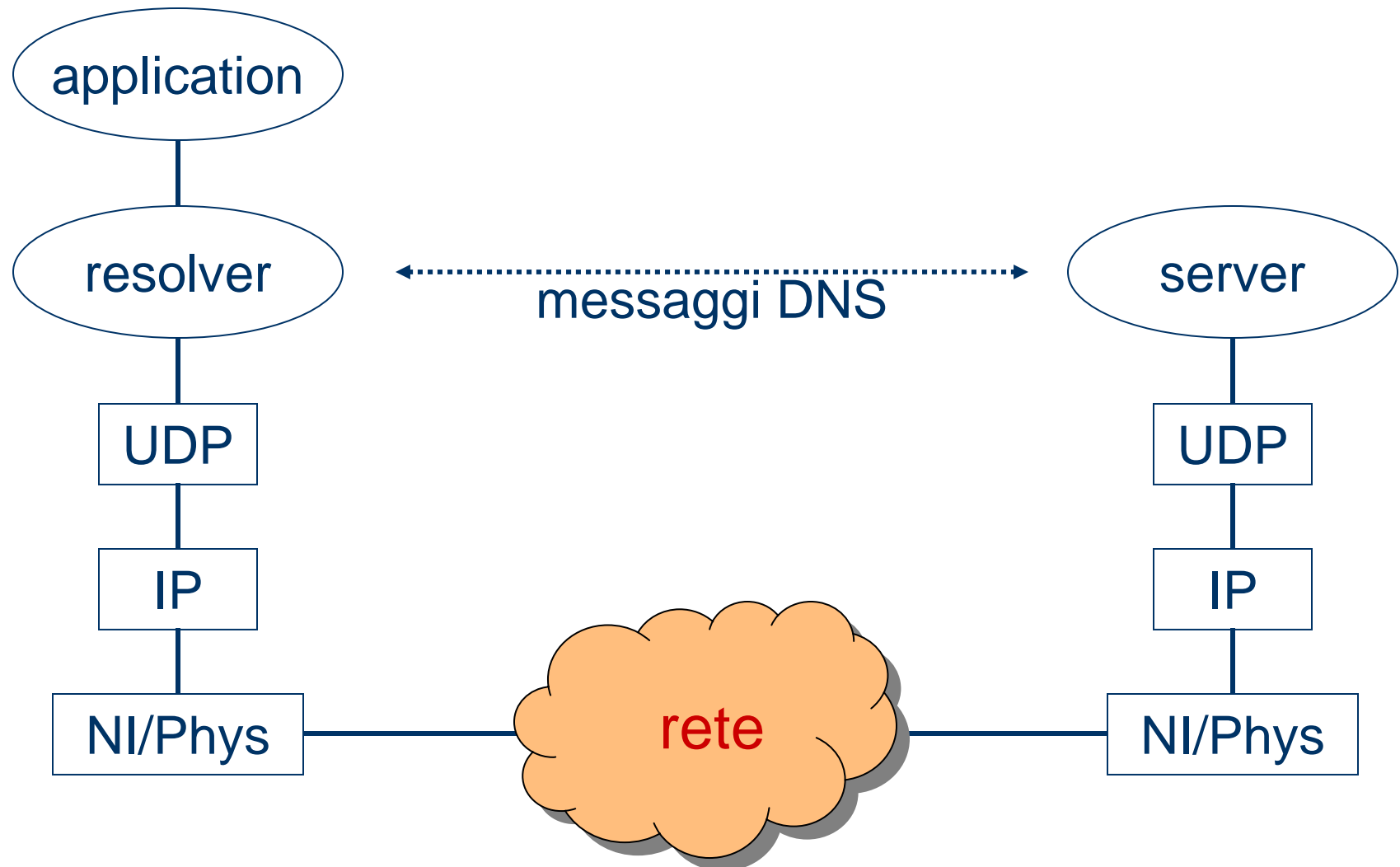
MX: Mail Server

DNS

Resolution scheme



layering



- ogni host è configurato impostando il nome di uno o più *local name server*
- ogni name server mantiene in una *memoria cache* le traduzioni fatte con il ricorso ad altri server in modo da rispondere più efficientemente a richieste successive
- la traduzione mantenuta da un name server per un nome non appartenente alla propria zona si dice *non authoritative*

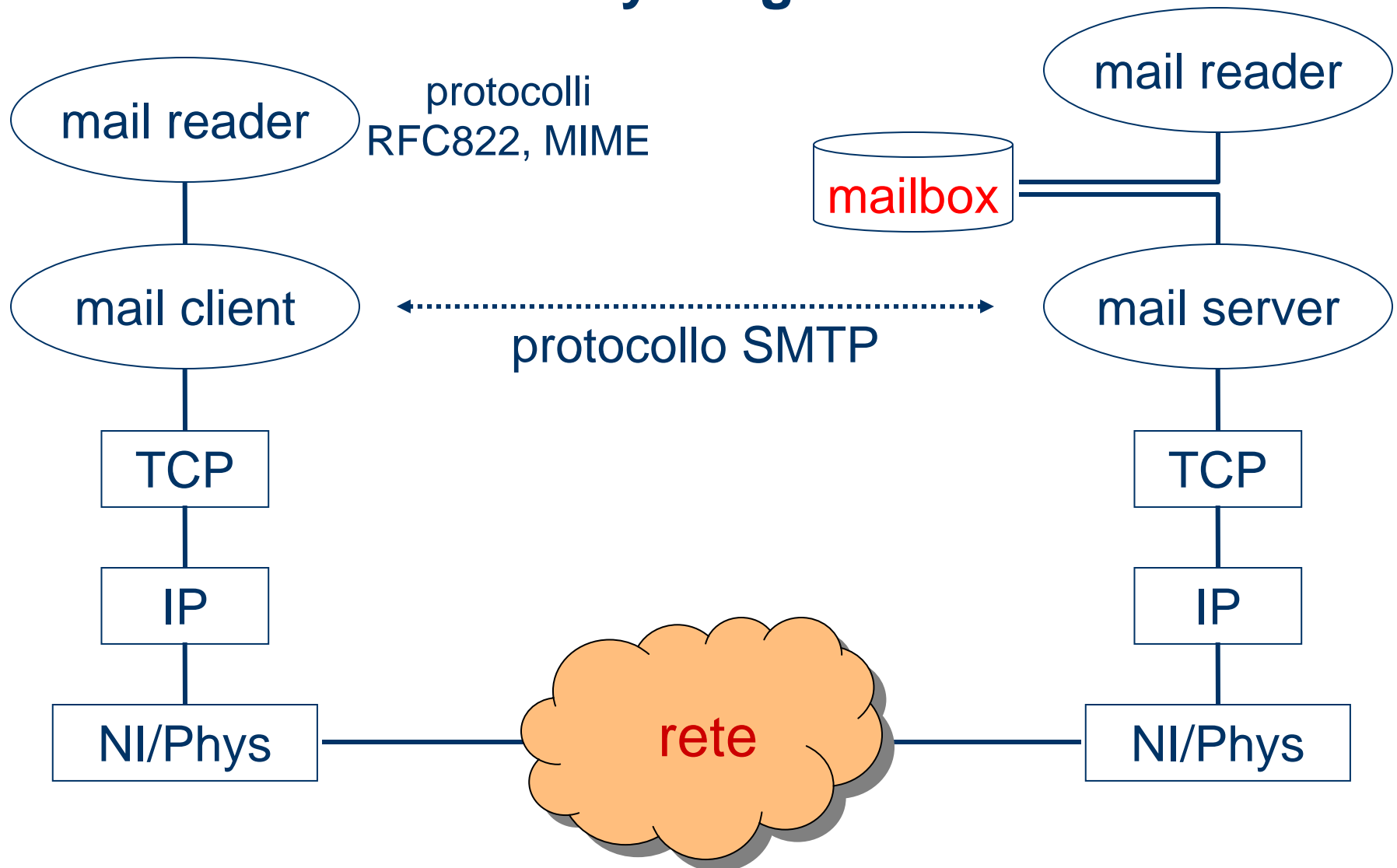
E-mail (Electronic mail)

- **indirizzo di E-mail:** mailbox@domain

Es. andrea.pietracaprina@unipd.it

- **mail reader:** interfaccia usata dall'utente per leggere/inviare E-mail. Es. outlook, netscape, elm
- **mail client:** riceve messaggi dal mail reader e li invia a mail server remoti
- **mail server:** riceve messaggi da mail client remoti e li deposita nella mailbox, dalla quale il reader li legge

layering



SMTP (Simple Mail Transfer Protocol)

protocollo che definisce il formato dei messaggi scambiati tra mail client e mail server tramite una connessione TCP. Uno di questi messaggi contiene il messaggio di E-mail vero e proprio

RFC 822, MIME (Multipurpose Internet Mail Extensions)

definiscono il formato dei messaggi di E-mail

- (RFC 822) header + body in ASCII
- (MIME) separazione e codifica ASCII dei diversi pezzi (es. testo, immagini, file word, ps, ecc.)

E-mail

From: "A.P." <andreaP@hotmail.com>
To: andrea.pietracaprina@unipd.it
Subject: test
Date: Thu, 10 May 2001 15:28:43 +0200
Mime-Version: 1.0
Content-Type: multipart/mixed;
 boundary="-----=_NextPart_000_6e5b_2618_3b35"
-----=_NextPart_000_6e5b_2618_3b35
 testo del messaggio
-----=_NextPart_000_6e5b_2618_3b35
 attachment (file xyz.gif)
-----=_NextPart_000_6e5b_2618_3b35
 attachment (file wjk.doc)
-----=_NextPart_000_6e5b_2618_3b35

H
E
A
D
E
R

B
O
D
Y

E-mail

Testo del messaggio:

Content-Type: text/plain; format=flowed

.....
.....



Plain
Text
(ASCII)

Attachment (file xyz.gif):

Content-Type: image/gif; name="xyz.gif"

Content-Transfer-Encoding: base64

Content-Disposition: attachment; filename="xyz.gif"

.....
.....



Codifica
MIME del file

E-mail

Attachment (file **wjk.doc**):

Content-Type: application/msword; name="wjk.doc"

Content-Transfer-Encoding: base64

Content-Disposition: attachment; filename="wjk.doc"

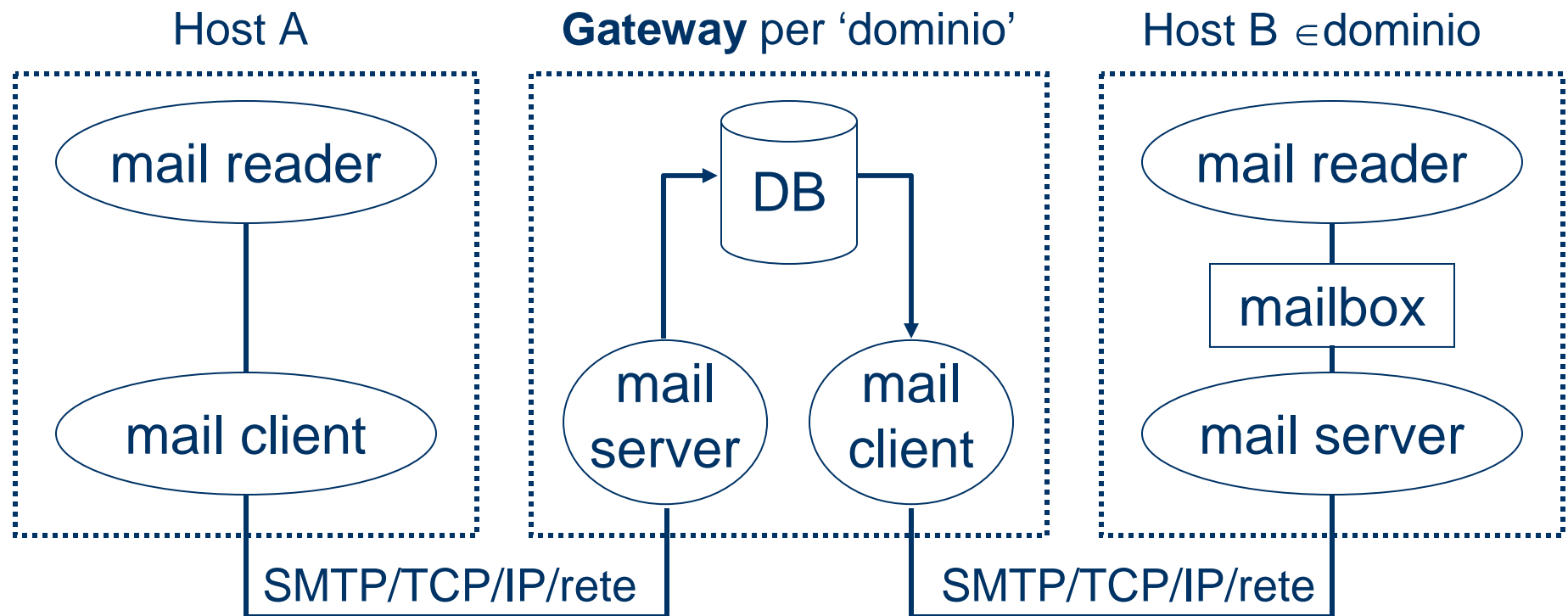
.....
.....



Codifica
MIME del file

E-mail

Mail Gateway: computer che riceve i messaggi indirizzati a un certo dominio e li trasferisce a host specifici del dominio, in funzione del destinatario



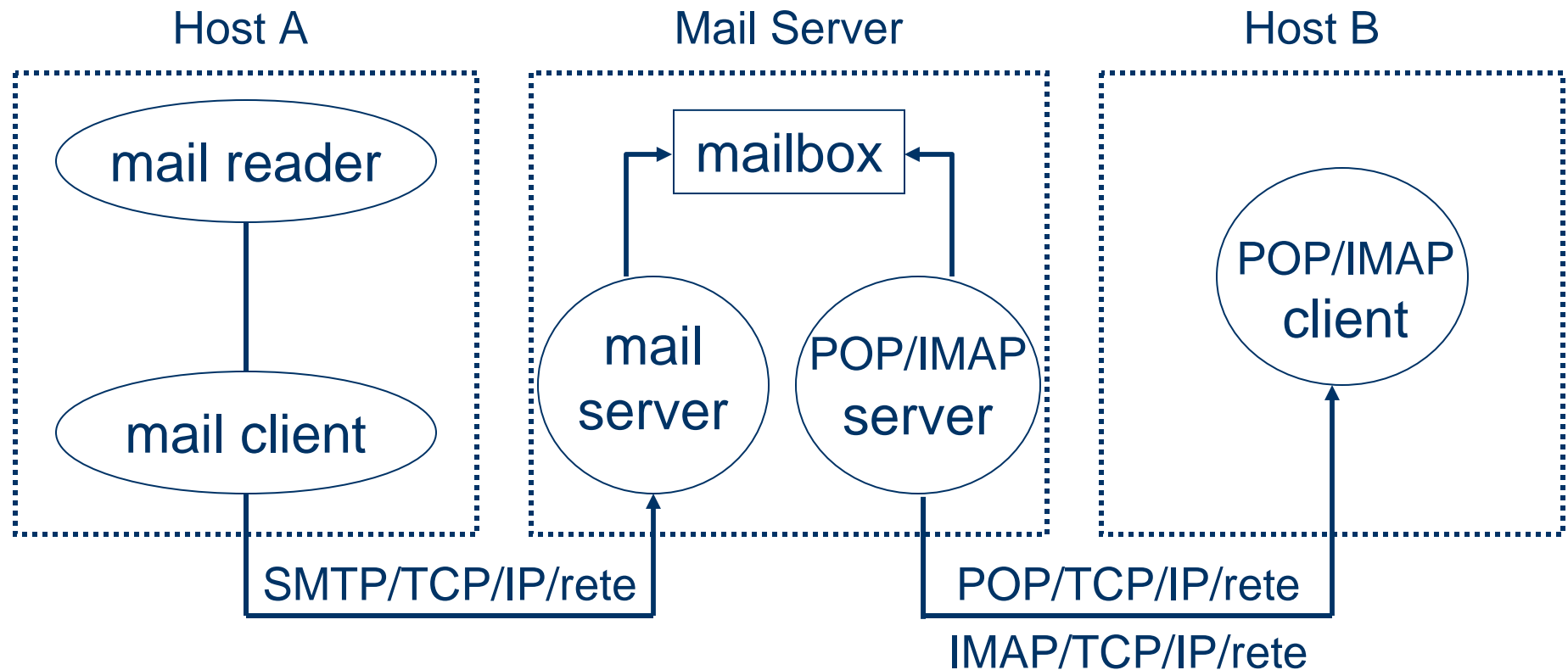
E-mail

- il software nel gateway consulta la base di dati per decidere l'host del dominio al quale trasferire il messaggio
- il gateway è sempre attivo mentre l'host può non esserlo. In questo caso il gateway mantiene il messaggio nel buffer e riprova a trasmetterlo per un certo tempo
- Nel DNS i gateway per i vari domini sono identificati dall'etichetta MX

Osservazione: l'uso dei gateway rende uniforme l'invio di E-mail a un dominio e ne rende la gestione più flessibile

E-mail

IMAP/POP: protocolli usati per accedere alla propria mailbox su un mail server remoto, tramite autenticazione via password. IMAP permette la gestione delle cartelle sul sever, mentre POP permette solo di scaricare la posta dal server



WWW (World Wide Web)

Sistema ipermediale: collezione di documenti (testo, immagini, audio, video) collegati tra loro.

WWW (o Web): sistema ipermediale distribuito che supporta l'accesso interattivo tramite il protocollo **HTTP**

Pagina web: singolo documento della Web

Hyperlink: oggetto selezionabile che collega una pagina Web a un'altra

Homepage: pagina principale di accesso per una persona o un'organizzazione

Browser: programma client che realizza l'interfaccia usata dall'utente per accedere alla Web.

Es. Netscape, Explorer, Mozilla

Web Server: programma server che gestisce un insieme di pagine web (*sito web*) inviandole a host remoti su richiesta dei browser che girano su essi.

Es. Apache

URL (Uniform Resource Locator): usato nel browser per specificare la pagina web da aprire. Es.

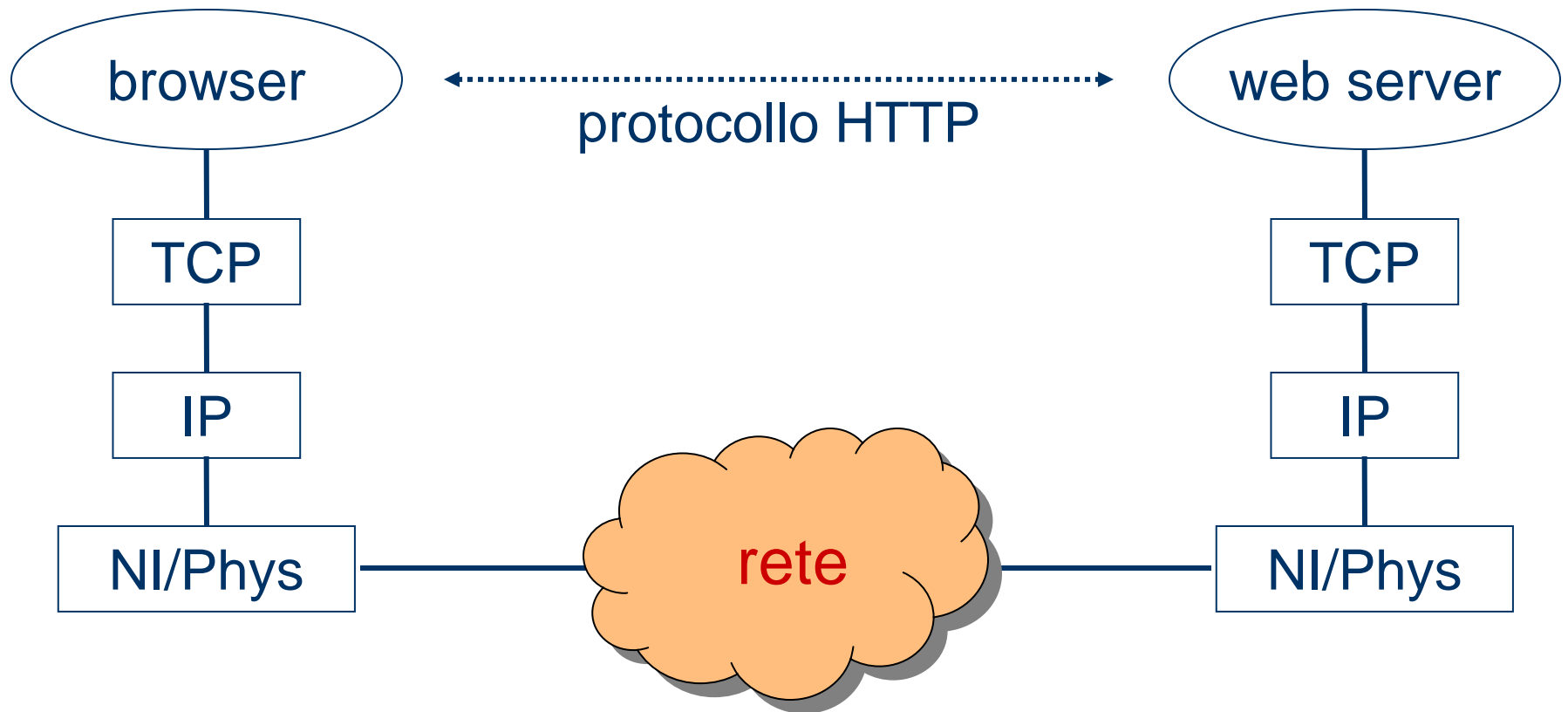
`protocol://host:port/documento` (':port' di solito omissso)

<http://www.dei.unipd.it/~capri/SI/index.htm>

HTML (HyperText Markup Language): linguaggio usato per scrivere pagine web, che il browser interpreta per visualizzare tali pagine

HTTP (HyperText Transport Protocol): protocollo che definisce il formato dei messaggi scambiati da browser e web server tramite una connessione TCP

layering



Accesso a una pagina web

1. apertura connessione TCP (browser → web server)
2. scambio di messaggi HTTP con in quali, tra le altre cose, il server invia al browser uno o più documenti che il browser interpreta e mostra sul video.
3. chiusura della connessione TCP da parte del server (ad es. dopo timeout) o del browser

Oss. Documenti binari vengono codificati in MIME

Esempio di scambio di messaggi HTTP:

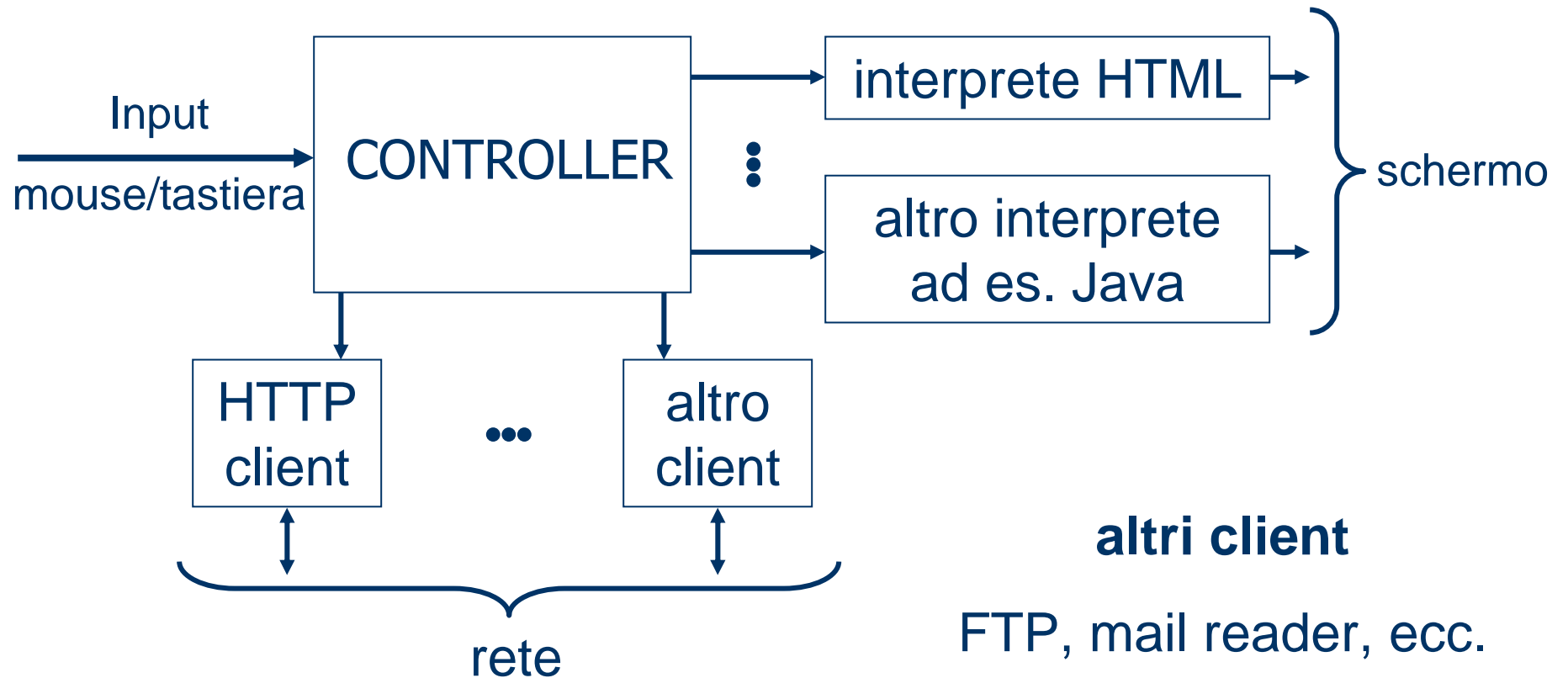
Browser: GET url HTTP/1.1

Web Server: HTTP/1.1 202 Accepted
documento

oppure

Web Server: HTTP/1.1 404 Not Found

Architettura di un browser



Web Caching: Per rendere più efficienti accessi futuri, copie delle pagine web accedute da un browser sono mantenute su:

- web cache locale della macchina su cui gira il browser, la cui taglia è impostata dall'utente
- macchine dedicate (**PROXY**) che sono accedute esplicitamente dal browser oppure che intercettano la comunicazione tra browser e web server rispondendo alle richieste del browser nel caso abbiano una copia delle pagine richieste

Tipologie di Pagine Web

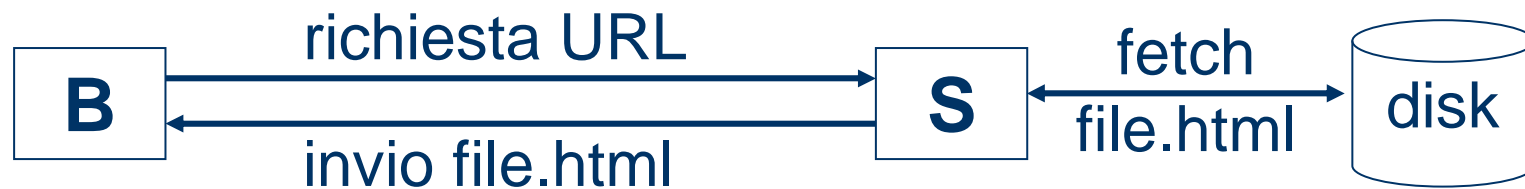
Pagina Web Statica: è scritta in linguaggio HTML e risiede nel sistema di memoria del web server. Una volta creato il suo contenuto non cambia

Pagina Web Dinamica: è scritta in linguaggio HTML ed è creata dinamicamente dal web server su richiesta del browser. I contenuti riflettono il momento della creazione

Pagina Web Attiva: contiene un programma (scritto ad es. in Java) che il browser riceve dal server ed esegue localmente.

Interazione Browser (B) - Web Server (S)

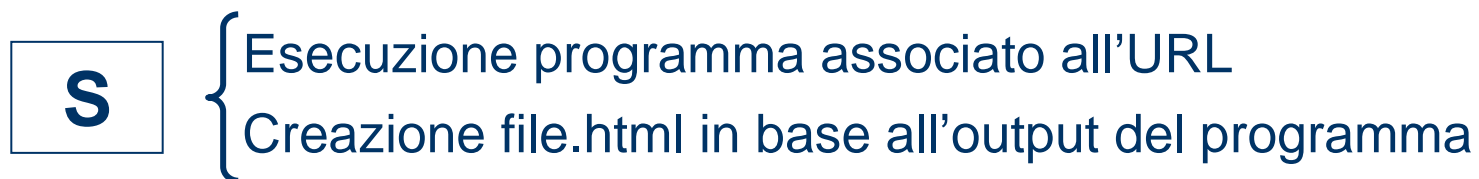
Pagina Web Statica



Vantaggi: semplicità, efficienza

Svantaggi: staticità dei contenuti

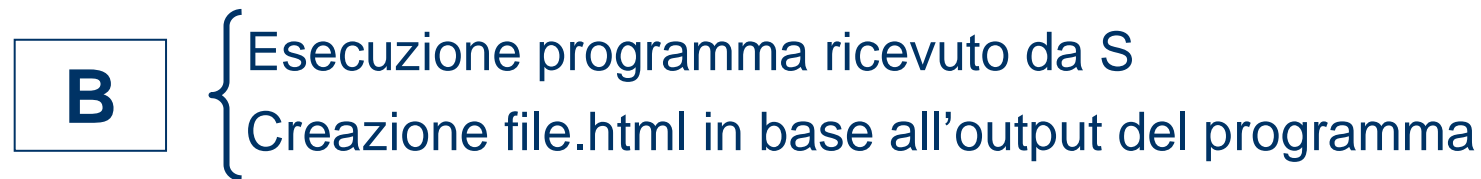
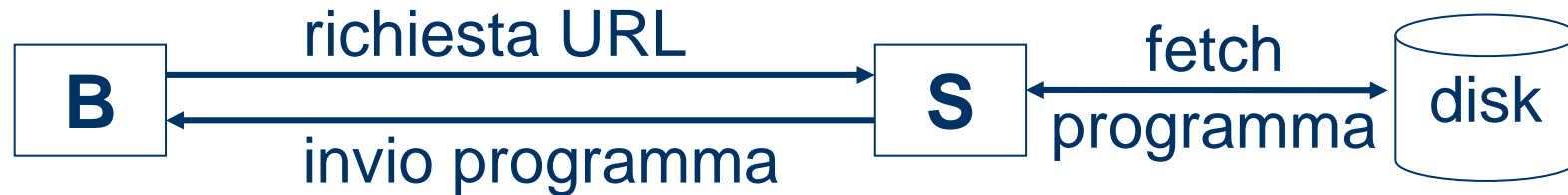
Pagina Web Dinamica



Vantaggi: contenuti aggiornati

Svantaggi: maggiore overhead per il server,
pagina non aggiornabile dopo invio

Pagina Web Attiva



Vantaggi: pagina aggiornabile dopo invio

Svantaggi: maggiore complessità del browser,
minore sicurezza

Tecnologie per pagine web dinamiche:

- CGI
- Java Servlet
- Server side scripting: ASP, JSP, PHP

Tecnologie per pagine web attive:

- Applet Java
- JavaScript

CGI (Common Gateway Interface): interfaccia che specifica come il web server interagisce con un programma

- Il browser richiede un URL associato a un **GATEWAY**, ovvero un programma scritto in linguaggio ad alto livello (ad es. C, C++, Perl) chiamato anche *programma CGI*
- il webserver invoca il gateway passandogli eventuali parametri “appesi” all’URL dal browser, ne interpreta l’output e crea la pagina web da inviare al browser

Osservazione

Ogni richiesta dell'URL comporta l'esecuzione del programma come nuovo processo

→ spreco di risorse del server

La tecnologia basata su **Java Servlet** è analoga a CGI ma risolve i problemi di performance.

Server Side Scripting:

- Nel codice HTML della pagina web è integrato un programma scritto in un *linguaggio di scripting* e delimitato da tag particolari (ad es. `<% %>`)
- il webserver esegue il codice che trova all'interno della pagina web e produce una pagina in HTML “puro” da inviare al browser. Tale pagina è ottenuta integrando il codice HTML presente nella pagina originale e l'output del programma

Tecnologie per server side scripting:

ASP (Active Server Pages): tecnologia Microsoft

- Estensione file: “.asp”
- Tag: “<% ... %>”

JSP (Java Server Pages): tecnologia Sun

- Estensione file: “.jsp”
- Tag: “<% ... %>”

PHP (PHP: Hypertext Preprocessor):

- Estensione file: “.php”
- Tag: “<?php ... ?>”

Invio di parametri dal browser B al web server S:

- B richiede a S un URL contenente un form, ovvero una pagina html in cui sono presenti sezioni delimitate da: **<FORM>** **</FORM>** per la raccolta di valori da associare a parametri specificati
- S invia a B la pagina web contenente il form
- L'utente da B compila il form inserendo i valori richiesti e richiede a S (cliccando un apposito bottone di invio) un URL associato a una pagina web dinamica, specificando i valori passati come parametri:

Esempio

Form (ad es. www.mysite.com/index.html):

```
<HTML>
```

```
.....
```

```
<FORM METHOD = "GET" ACTION="xyz.php">
```

```
Nome: <INPUT TYPE="text" NAME="nome"> <p>
```

```
Cognome: <INPUT TYPE="text" NAME="cognome">
```

```
<INPUT TYPE="submit" VALUE="invia">
```

```
</FORM>
```

```
.....
```

```
</HTML>
```

Inserendo “Mario” e “Rossi” nei due box presenti sul form e cliccando il bottone “invia” si invia al server la richiesta

`www.mysite.com/xyz.php?nome=Mario&cognome=Rossi`

Il file xyz.php conterrà (in mezzo al codice HTML):

```
<?php
```

```
    $nome=$_GET['nome'];
```

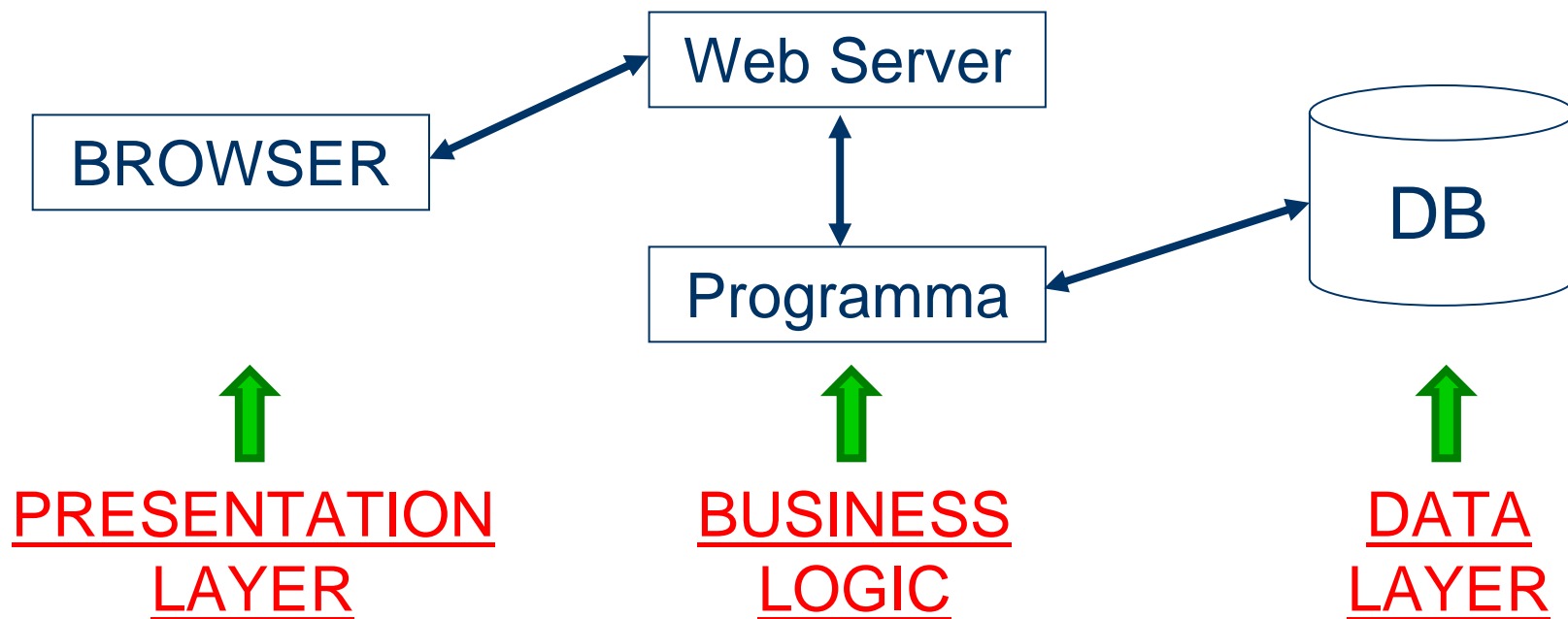
```
    $cognome=$_GET['cognome'];
```

```
    print("$nome $cognome")
```

```
?>
```

Implementazione dell'architettura a 3 livelli

Il programma associato a (o integrato in) una pagina web dinamica può interagire con un DBMS (ad es. Access, MySQL) accedendo a un DB



Tecnologie per pagine web attive

Applet Java: un programma Java compilato nel formato **bytecode** viene inviato al browser che lo esegue usando un **interprete** Java.

JavaScript: un programma scritto in JavaScript (linguaggio di scripting), inserito all'interno di una pagina HTML (tra i tag `<SCRIPT>` e `</SCRIPT>`), viene inviato al browser che lo esegue creando una pagina in HTML puro da visualizzare