

Laboratorio di Metodi Informazionali

Laurea in Bioinformatica

Docente: *Carlo Drioli*

Web: www.scienze.univr.it/fol/main?ent=oi&id=34118

Lucidi a cura di
Andrea Colombari, Carlo Drioli, Barbara Oliboni e
Andrea Fusiello

Lezione 5

Introduzione alle reti

Materiale tratto dai lucidi ufficiali a corredo del testo:



D. Sciuto, G. Buonanno e L. Mari
“Introduzione ai sistemi informatici”
2005 - McGrawHill

Rete di calcolatori

- Insieme di calcolatori autonomi tra loro collegati mediante una **rete di comunicazione**.
- Gli utenti sono in grado di **interagire** in modo esplicito con la rete (e in alcuni casi sono tenuti a farlo).
- I calcolatori connessi alla rete mantengono un certo grado di **indipendenza**: in caso di guasto o indisponibilità della rete ogni calcolatore continua a funzionare individualmente.

Perché una rete?

- Condividere risorse
 - utilizzo razionale di dispositivi costosi
 - modularità della struttura
 - affidabilità e disponibilità

- Comunicare tra utenti
 - scambio informazioni
 - collaborazione a distanza

La dimensione delle reti

- Reti locali (Local Area Network, LAN)
 - Di limitata estensione.
 - Collegano dispositivi collocati nello stesso edificio o in edifici adiacenti.
- Reti metropolitane (Metropolitan Area Network, MAN)
 - Collegano di dispositivi collocati nella stessa area urbana.
- Reti geografiche (Wide Area Network, WAN)
 - Collegano di dispositivi diffusi in un'ampia area geografica (nazione, continente, ...).
- “Reti di reti” (Internetwork)
 - Collegano più reti differenti (in termini sia hardware che software) mediante opportuni elementi di interfaccia, che si possono estendere su tutto il pianeta (e.g. Internet).

I protocolli di comunicazione

- Per comunicare i calcolatori debbono seguire delle le regole: i **protocolli di comunicazione**.
- I protocolli di comunicazione specificano:
 - i formati dei dati,
 - la struttura dei pacchetti (includendo la definizione delle informazioni di controllo)
 - la velocità di trasmissione
 - ...
- Definire tutte queste proprietà tramite un unico protocollo è praticamente impossibile, per questo si definisce un **insieme di protocolli**:
 - ogni protocollo gestisce univocamente una componente ben definita della comunicazione
 - ogni protocollo condivide con gli altri protocolli i dati di cui essi necessitano.

La struttura di Internet

il contenuto della
comunicazione ...

protocollo applicativo:
livello applicativo

protocolli di trasmissione:
livello di trasmissione

infrastruttura telematica:
livello di connessione fisica

Posta elettronica
Login remoto

Copia di files
World Wide Web

NNTP

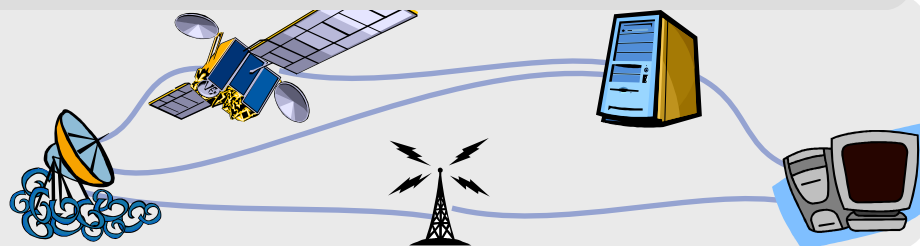
SMTP/POP

TELNET

FTP

HTTP

TCP/IP



TCP/IP: indirizzamento

- Schema di indirizzamento generale su due livelli:

Indirizzo IP + Porta TCP

- **Indirizzo IP**

- Indirizzo associato a ogni calcolatore collegato a una sottorete.
- Si tratta di un indirizzo **Internet** globale unico.

- **Porta TCP**

- Indirizzo unico all'interno dell'host che individua un processo attivo sull'host.
- Utilizzato da TCP per consegnare i dati al processo giusto.

Indirizzo IP (versione 4)

- 32 bit (cioè 4 byte) per un totale di 2^{32} possibili indirizzi diversi
- rappresentato in forma “dotted decimal”
 - successione di quattro numeri (uno per byte), separati da un punto (esempio 102.54.94.97)
 - ognuno dei quattro numeri della notazione dotted decimal è compreso tra 0 e 255.
- strutturato in due parti:
 - una parte che individua la rete fisica a cui la stazione è collegata,
 - l'altra che identifica la singola stazione nell'ambito della rete fisica;
 - esistono tre classi primarie, chiamate A, B e C, ognuna caratterizzata da una diversa suddivisione dei 32 bit:
 - A - un byte (8 bit) per la rete + 3 byte (24 bit) per i calcolatori; inizia per “0”;
 - B - 2 byte (16 bit) per la rete + 2 byte (16 bit) per le stazioni; inizia per “10”;
 - C - 3 byte (24 bit) per la rete + 1 byte (8 bit) per i calcolatori; inizia per “110”.

Indirizzi numerici vs indirizzi simbolici

- Gli indirizzi IP sono **machine-oriented**, quindi difficili da utilizzare per un utente “umano”.
- È stato definito un sistema per passare da indirizzi numerici (gli **indirizzi IP**) a **nomi** facilmente memorizzabili.
- **Domain Name System (DNS)**
 - Associa a ogni indirizzo IP uno o più indirizzi simbolici (*domain name*).
 - Gestisce la conversione tra indirizzi simbolici e indirizzi IP.
- Organizzato in **maniera gerarchica** (domini, sotto-domini, sotto-sotto-domini, ...) per semplificarne l'utilizzo.

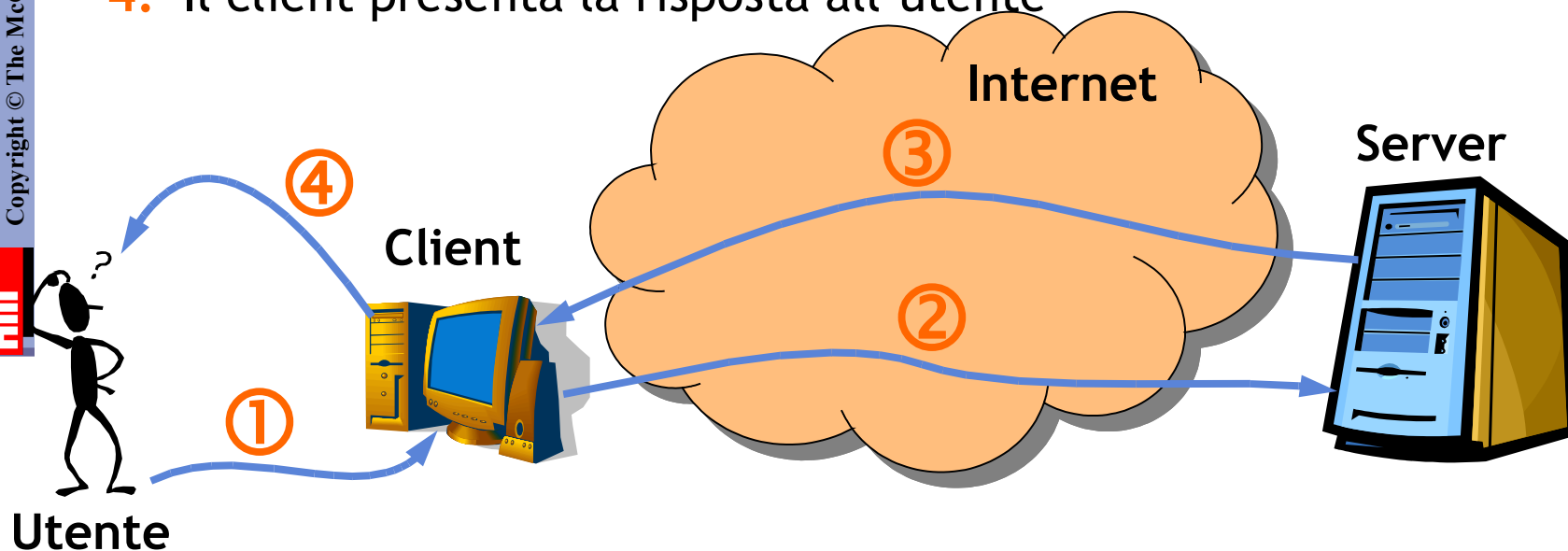


Domain Name System

- Il *domain name* di un calcolatore è costituito da una successione di stringhe alfanumeriche separate da punti (per esempio, server1.isttec.liuc.it)
- Ogni stringa identifica un “dominio”:
 - La stringa più a destra rappresenta il dominio di primo livello (detto anche dominio generale).
 - La seconda stringa, sempre proseguendo da destra verso sinistra, indica il dominio di secondo livello.
 - Le stringhe successive indicano i domini di terzo livello (sottodomini dei domini di secondo livello), quelli di quarto livello, e così via finché non si arriva a individuare un dominio che comprende il singolo host.

Il paradigma client-server

1. L'utente usa il client per esprimere le sue richieste
2. Il client si collega al server e trasmette la richiesta
3. Il server risponde al client
4. Il client presenta la risposta all'utente

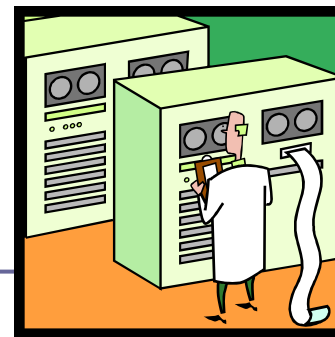


Il client



- Si preoccupa di dialogare con l'utente
- Sfrutta tutte le possibilità fornite dal calcolatore su cui viene eseguito (audio, video, ...)
- Fornisce all'utente un'interfaccia intuitiva
- Elabora le richieste dell'utente e le risposte dei server
 - la comunicazione avviene secondo un formato standard (protocollo)

Il server



- Rende disponibili delle risorse
- Accetta richieste e risponde automaticamente
 - non bada alla provenienza della richiesta
 - il processo client può trovarsi in qualsiasi punto della rete
- Si può organizzare un insieme di server in modo che siano collegati tra loro
- Potrebbe essere eseguito dallo stesso calcolatore che esegue il processo client!

Servizi

- Un server mette a disposizione dei servizi, o in generale delle *risorse*
- Ciascun servizio è identificato da un numero di *porta*
- Su ciascuna porta è in ascolto il programma (*daemon*) che esegue le operazioni necessarie per l'espletazione del servizio
- Ogni servizio usa un proprio *protocollo*

Servizi (2)

- In base a quanto appena detto, è possibile identificare una risorsa su Internet tramite:
 - Protocollo (= servizio richiesto)
 - Indirizzo del computer (IP o Domain Name)
 - Numero della porta (può mancare, i protocolli prevedono porte di default)
 - Nome della risorsa

Principali servizi e protocolli

Servizio	Descrizione	Protocollo
World Wide Web	Creazione, distribuzione e visualizzazione di ipertesti con contenuti multimediali	HTTP
E-mail	Scambio di messaggi di posta elettronica	POP e SMTP
File transfer	Copia di file da e su computer collegati a Internet	FTP
Remote login	Utilizzo delle risorse di computer remoti	Telnet, SSH

Indirizzi URL

- Ogni risorsa su Internet è identificata da un nome univoco chiamato **URL** (Uniform Resource Locator)
- L'URL è composto da tre parti: 1) il protocollo di comunicazione, 2) il nome della macchina su cui risiede la risorsa, 3) il nome della risorsa
- Esempio: l'URL della pagina che contiene la biografia di Tim Berners Lee (uno dei padri del WWW), è:

http://www.w3.org/People/Berners-Lee/Longer.html

protocollo

nome della
macchina

path e nome del
file contenente la
pagina

Comandi di rete

- **hostname** Restituisce il nome del proprio computer (host).
- **ping** Permette di diagnosticare se è raggiungibile via rete l'indirizzo specificato. Sintassi:
`ping ip_or_host_name`
Si usa CTRL-C (break) per fermare il ping.
- **traceroute** Permette di visualizzare l'elenco degli nodi che esistono fra il nostro host e l'indirizzo specificato. Sintassi:
`traceroute machine_name_or_ip`
- **host - dig** Client DNS, permettono di interrogare il server DNS per recuperare IP o domain name. Sintassi:
`host ip_address` o `host domain_name`
`dig domain_name` o `dig -x ip_address`

Comandi di rete

- **mail** legge ed invia la posta elettronica. Per la lettura è obsoleto, ma per l'invio può risultare utile. Sintassi (per invio) **mail indirizzo**
prende input da stdin, termina con CTRL-D (come cat)
- **telnet** Permette un collegamento su una shell di un host remoto. La porta a cui risponde un telnet server è la 23 e viene data per sottointesa. E' comunque possibile effettuare un telnet ad altre porte (80 per HTTP, 25 per SMTP, 143 per IMAP) e digitare direttamente dei comandi validi per il protocollo utilizzato dal server a cui ci si è connessi. I dati vengono trasmesse in chiaro sulla rete. Esempio:
telnet profs.sci.univr.it o **telnet profs.sci.univr.it 80**

Comandi di rete

- **ssh** Come telnet permette l'accesso remoto via shell ad un sistema, ma i dati trasmessi vengono criptati per maggior sicurezza.
`ssh hostname`
si collega al calcolatore remoto con il nome utente corrente
`ssh username@hostname`
si collega al calcolatore remoto con il nome utente fornito
- **ftp** client dell'omonimo protocollo di trasmissione di file.
- **wget** scarica files dal WWW in modo non interattivo. Esempio:
`wget url_for_file`
scarica il file specificato dall'URL. Esempio:
`wget http://netpbm.sourceforge.net/doc/ -O- | grep Netpbm`
L'opzione -O- mette su stdout

Il Protocollo HTTP

- HTTP (Hypertext Transfer Protocol) è il “linguaggio” utilizzato per controllare l’invio di documenti HTML via Internet.
- Il protocollo HTTP prescrive le regole mediante le quali i browser effettuano le richieste e i server forniscono le relative risposte.
- Documentazione: RFC 2616
(<http://www.freesoft.org/CIE/RFC/index.htm>)
versione aggiornata delle specifiche del protocollo HTTP versione 1.1.

La richiesta HTTP

- HTTP è un protocollo *senza stati* a richieste e risposte.
- Senza stati significa che il server Web non ricorda nulla delle richieste pervenute in precedenza dallo stesso client.
Il protocollo considera semplicemente
 - la richiesta attuale di un documento e
 - la risposta costituita dal documento stesso.

La richiesta HTTP (2)

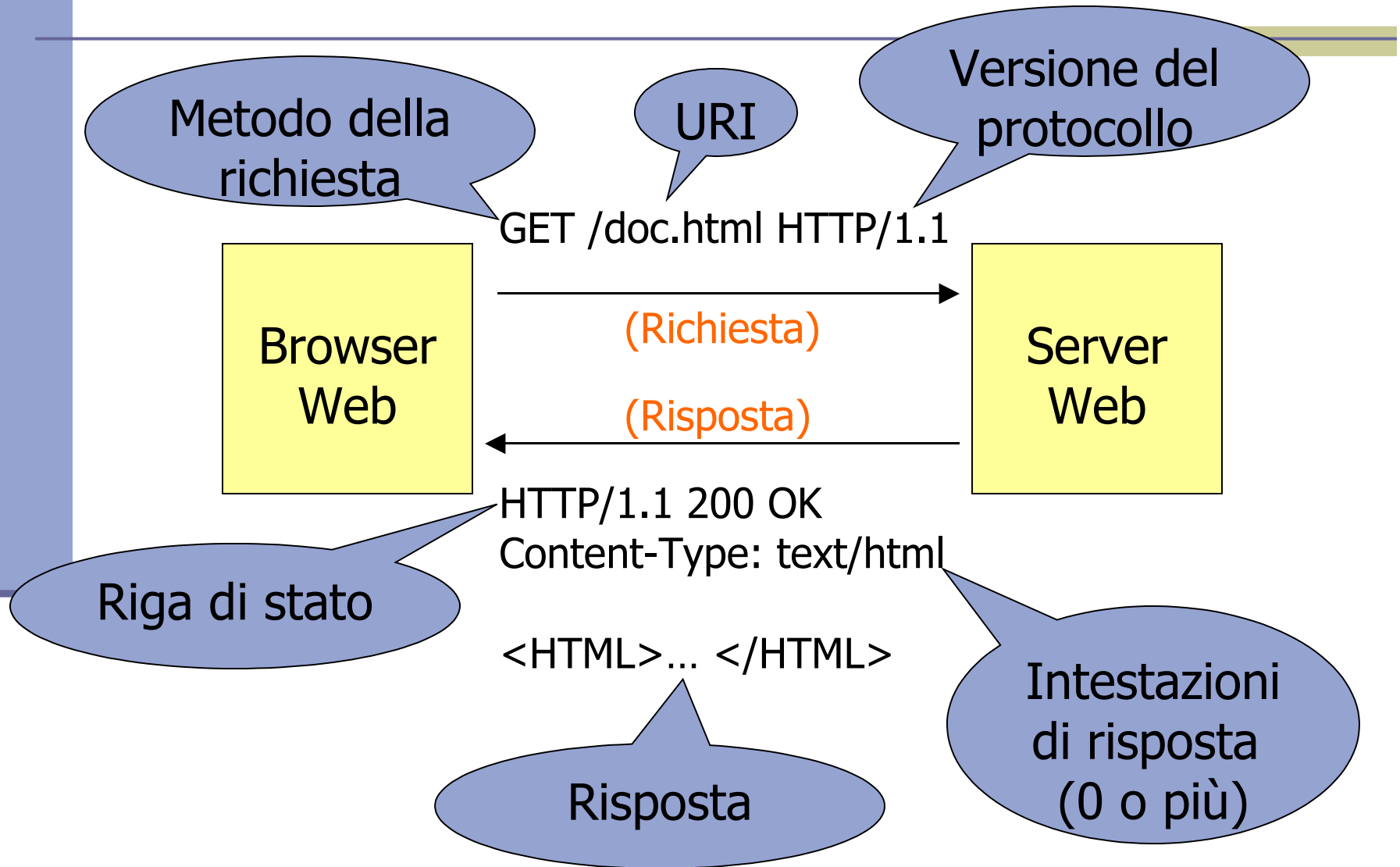
■ Operazioni di base:

- Un'applicazione client (browser Web) apre una connessione verso la porta HTTP del server Web (normalmente la porta 80).
- Il client invia una richiesta attraverso la connessione aperta.
- Il server Web analizza la richiesta ed individua la risorsa specificata.
- Il server invia una copia della risorsa.
- Il server chiude la connessione.

Connessione al Server Web

- Normalmente un server Web riceve le richieste sulla porta 80, in questo caso l'indirizzo <http://www.w3.org/People/Berners-Lee/Longer.html> fa riferimento al documento [/People/Berners-Lee/Longer.html](#) sul server Web in esecuzione sull'host www.w3.org e operante sulla porta standard 80.
- Se invece il server Web utilizzasse la porta 8080 (p.es.), l'URL dovrebbe essere: <http://morpheus.micc.unifi.it:8080/cruscle/>

Funzionamento di HTTP



Esempio

- Sulla riga di indirizzo del browser viene digitato <http://www.w3.org/People/Berners-Lee/Longer.html>
- Il browser web apre una connessione sulla porta 80 del server web www.w3.org
- Il browser web invia la richiesta [GET /People/Berners-Lee/Longer.html](http://www.w3.org/People/Berners-Lee/Longer.html) HTTP/1.0

Esempio (2)

■ Il server web restituisce:

HTTP/1.1 200 OK

Date: Mon, 16 Jul 2007 14:04:05 GMT

Server: Apache/1.3.37 (Unix) PHP/4.4.7

...

Content-Length: 13719

Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<title>Longer Bio for Tim Berners-Lee</title>

<meta http-equiv="content-type" content="text/html; charset=iso-8859-1">

<link href="general.css" rel="stylesheet" type="text/css">

</head>

...

</html>

Codice HTML

Esempio (3)

- Il browser analizza la riga di stato e trova il codice di stato **200 ok** che indica che la richiesta ha avuto successo.
- Il browser analizza le intestazioni di risposta che indicano che verranno inviati 1619 byte di codice HTML.
- Il browser legge il codice HTML e visualizza il risultato.
- Se il codice HTML contiene riferimenti ad altre risorse che devono essere caricate con il documento, allora il browser invia una richiesta per ogni risorsa necessaria.
- Si può simulare la sessione appena vista tramite telnet (provare!):
telnet www.w3.org 80