
Livello Applicativo

Application Layer

Applicazioni di Rete - M. Ribaudo - DISI

Cosa vediamo ...

- **Parte 1**
Principi generali

- **Parte 2**
Il Web e il protocollo HTTP

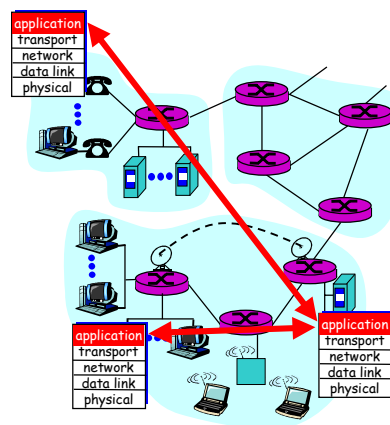
Applicazioni di Rete - M. Ribaudo - DISI

Parte 1

Principi generali

Applicazioni di Rete - M. Ribaudo - DISI

Scenario di riferimento



Applicazioni di Rete - M. Ribaudo - DISI

Principi generali: Processi

- **Processo:** programma che gira su un host
- I processi sullo stesso host comunicano usando gli strumenti di comunicazione messi a disposizione dal S.O. (**interprocess communication**)
- I processi che sono su host diversi comunicano usando **un protocollo del livello applicativo**

Applicazioni di Rete - M. Ribaudo - DISI

Principi generali: User agent

- **User agent:** interfaccia tra l'utente e la rete
- Implementa i protocolli del livello applicativo
 - ✓ Web: browser
 - ✓ E-mail: mail reader
 - ✓ Streaming audio/video: media player

Applicazioni di Rete - M. Ribaudo - DISI

Principi generali: Applicazione

- Costituita da **processi distribuiti** e comunicanti (e-mail, Web, P2P file sharing, instant messaging, ...)
- I processi "girano" sugli end-system (host)
- Si **scambiano dei messaggi** per implementare l'applicazione stessa

Applicazioni di Rete - M. Ribaudo - DISI

Principi generali: Protocollo applicativo

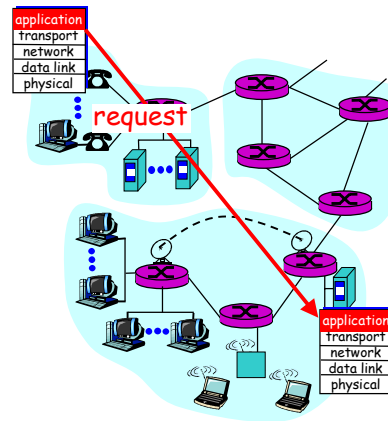
- È una parte di una applicazione e **definisce**
 - ✓ I **tipi** di messaggi che vengono inviati
 - ✓ La **sintassi** dei messaggi: che campi ci sono e come si distinguono nel messaggio
 - ✓ La **semantica** dei messaggi: il significato delle informazioni contenute nei campi
 - ✓ Le **regole** per sapere come e quando i processi inviano richieste e spediscono risposte

Applicazioni di Rete - M. Ribaudo - DISI

Principi generali: Client / Server

Client

- Contatta il server ("parla per primo")
- Richiede un servizio

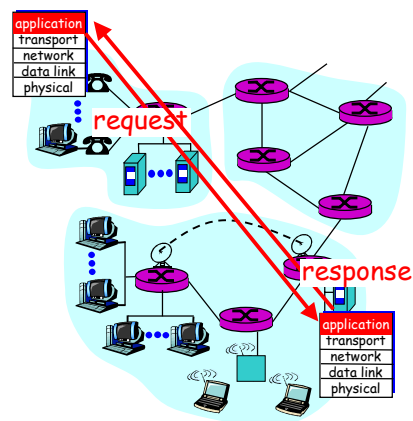


Applicazioni di Rete - M. Ribaudò - DISI

Principi generali: Client / Server

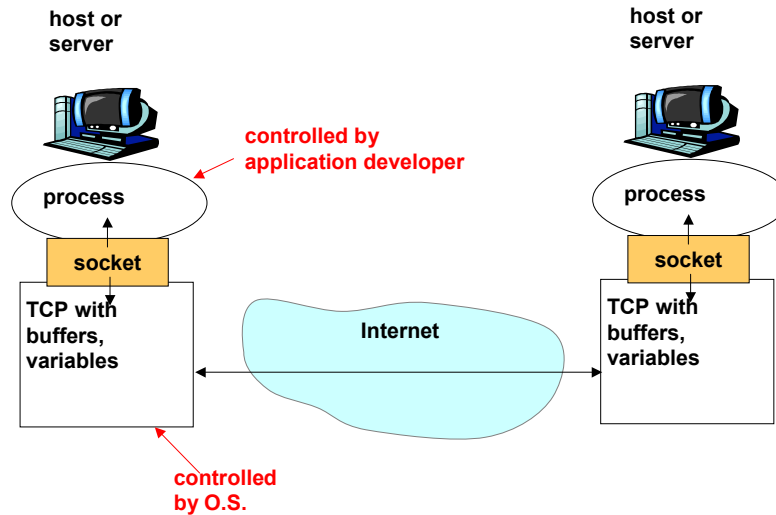
Server

- Fornisce il servizio richiesto al client
- Ad es. un Web server spedisce le pagine web richieste, un mail server inoltra le e-mail



Applicazioni di Rete - M. Ribaudò - DISI

Principi generali: Come comunicano i processi?



Applicazioni di Rete - M. Ribaudo - DISI

Principi generali: Indirizzamento

- I processi che ricevono i messaggi devono avere un **indirizzo univoco**
- Ogni host ha un indirizzo univoco a 32-bit: l'indirizzo **IP**
- **Domanda:** l'indirizzo IP dell'host è sufficiente per identificare un processo che gira sull'host?
- **Risposta:** no, ci possono essere più processi attivi sullo stesso host

Applicazioni di Rete - M. Ribaudo - DISI

Principi generali: Indirizzamento

- L'indirizzo univoco di un processo deve includere sia l'**indirizzo IP** che il **numero di porta** associato al processo sull'host

- **Esempi**
 - ✓ HTTP server: 80
 - ✓ Mail server: 25

- Rivedremo questi concetti più avanti

Applicazioni di Rete - M. Ribaudo - DISI

Principi generali:

Servizi necessari per un'applicazione

- **Perdita di dati (Data loss)**
 - ✓ Alcune applicazioni (es. audio) possono tollerare delle perdite di dati, altre (es. file transfer, telnet) richiedono un servizio affidabile al 100%

- **Dipendenza dal tempo (Time sensitive)**
 - ✓ Alcune applicazioni (es. Internet telephony, interactive games) non possono funzionare in presenza di ritardi di trasmissione

Applicazioni di Rete - M. Ribaudo - DISI

Principi generali:

Servizi necessari per un'applicazione

▪ Banda (Bandwidth)

- ✓ Alcune applicazioni (es. multimedia) richiedono una quantità minima di banda per essere efficaci
- ✓ Altre applicazioni ("elastic") possono funzionare in presenza di qualunque banda

Applicazioni di Rete - M. Ribaud - DISI

Principi generali:

Servizi necessari per un'applicazione

Application	Data loss	Bandwidth	Time Sensitive
file transfer	no loss	elastic	no
e-mail	no loss	elastic	no
Web documents	no loss	elastic	no
real-time audio/video	loss-tolerant	audio: 5kbps-1Mbps video: 10kbps-5Mbps	yes, 100's msec
stored audio/video	loss-tolerant	same as above	yes, few secs
interactive games	loss-tolerant	few kbps up	yes, 100's msec
instant messaging	no loss	elastic	yes and no

Applicazioni di Rete - M. Ribaud - DISI

Principi generali: Trasporto

- A seconda dell'applicazione ci si deve affidare ad un servizio di trasporto orientato alla connessione come quello offerto da **TCP** oppure può bastare un servizio di tipo "datagram" come quello fornito da **UDP**
- Rivedremo questi concetti più avanti

Applicazioni di Rete - M. Ribaudo - DISI

Principi generali: Trasporto

Application	Application layer protocol	Underlying transport protocol
e-mail	SMTP [RFC 2821]	TCP
remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
file transfer	FTP [RFC 959]	TCP
streaming multimedia	proprietary (e.g. RealNetworks)	TCP or UDP
Internet telephony	proprietary (e.g., Dialpad)	typically UDP

Applicazioni di Rete - M. Ribaudo - DISI

Parte 2

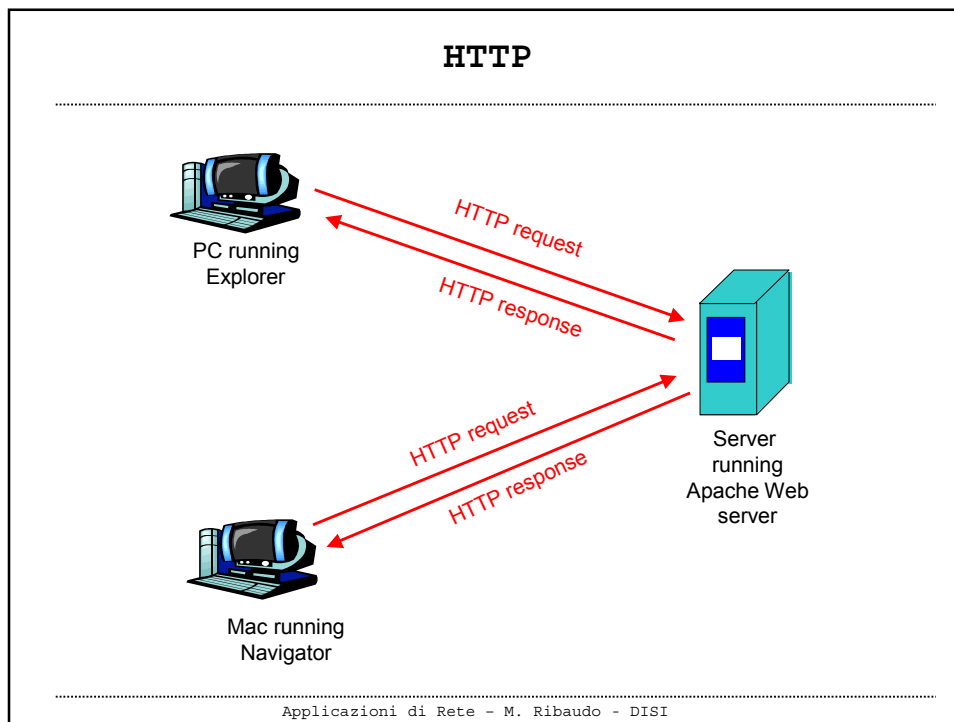
Il Web e il protocollo HTTP

Applicazioni di Rete - M. Ribaudo - DISI

Web

- Una pagina Web è formata da **oggetti**
- Gli oggetti possono essere file HTML, immagini (JPEG,GIF,PNG), applet Java, file audio,...
- Ogni oggetto è identificato da un **URL**

Applicazioni di Rete - M. Ribaudo - DISI



- ### HTTP
-
- HTTP si poggia su **TCP**
 - Il **client** HTTP (il browser) **inizia una connessione** TCP sulla porta 80 del server
 - Il **server** HTTP (Web server) **accetta la connessione** TCP del client
 - Client e server si **scambiano messaggi** HTTP (attraverso i socket)
 - Si **chiude** la connessione TCP
-
- Applicazioni di Rete - M. Ribaudo - DISI

HTTP

- **HTTP è "stateless"**: il server non mantiene alcuna informazione sulle richieste dei vari client

I protocolli che mantengono lo stato sono complessi. Si deve mantenere traccia della storia passata perchè, se il server o il client si guastano, la loro visione dello stato può diventare inconsistente e deve essere ripristinata

Applicazioni di Rete - M. Ribaud - DISI

Connessioni HTTP

HTTP 1.0 [RFC 1945]

Su ogni
connessione TCP
si invia al più
un oggetto

Connessione NON
persistente

HTTP 1.1 [RFC 2616]

Più oggetti
possono essere
inviati sulla
stessa
connessione TCP

Permette una
connessione
persistente

Applicazioni di Rete - M. Ribaud - DISI

HTTP non persistente

Supponiamo che l'utente digiti l'indirizzo www.nomedip.nomeuni.it/index.html e che il file contenga 10 immagini

1. Il client HTTP inizia una connessione TCP sulla porta 80 del server (il processo server) all'indirizzo www.nomedip.nomeuni.it

2. Il server HTTP, in attesa di connessioni TCP sulla porta 80, "accetta" la connessione e manda una notifica al client

Applicazioni di Rete - M. Ribaudò - DISI

HTTP non persistente

3. Il client HTTP manda una richiesta HTTP (request message) che contiene un URL

Il messaggio indica che il client richiede la pagina `index.html`

4. Il server HTTP riceve la richiesta, forma un messaggio di risposta (response message) che contiene l'oggetto richiesto, e lo inoltra al client

Applicazioni di Rete - M. Ribaudò - DISI

HTTP non persistente

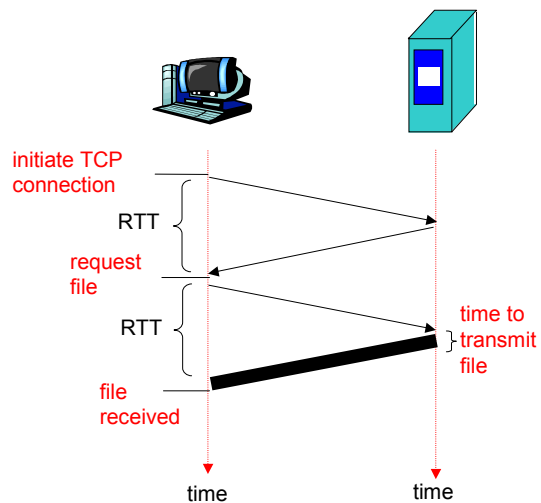
6. Il client HTTP riceve il messaggio di risposta che contiene il file index.html. Comincia a visualizzarlo e durante la parsificazione si accorge che ci sono 10 riferimenti ad oggetti JPEG

6. I passi 1-6 vengono ripetuti per ogni oggetto JPEG

5. Il server HTTP chiede la chiusura della connessione TCP

Applicazioni di Rete - M. Ribaudo - DISI

HTTP non persistente



RTT: tempo richiesto per spedire un pacchetto dal client al server e ritorno

Applicazioni di Rete - M. Ribaudo - DISI

HTTP non persistente

- Richiede 2 RTT per trasferire ogni oggetto (più il tempo per il trasferimento effettivo del file)
- Il S.O. deve allocare le risorse per ogni connessione TCP
- I browser spesso possono aprire più connessioni TCP in parallelo per caricare gli oggetti inclusi in un file HTML

Applicazioni di Rete - M. Ribaudò - DISI

HTTP persistente

- Il server non chiude la connessione TCP dopo aver inviato una risposta
- I messaggi HTTP successivi (tra lo stesso client e lo stesso server) vengono inoltrati sulla connessione aperta
- Il tempo di trasmissione diminuisce

Applicazioni di Rete - M. Ribaudò - DISI

HTTP persistente

▪ Senza pipelining

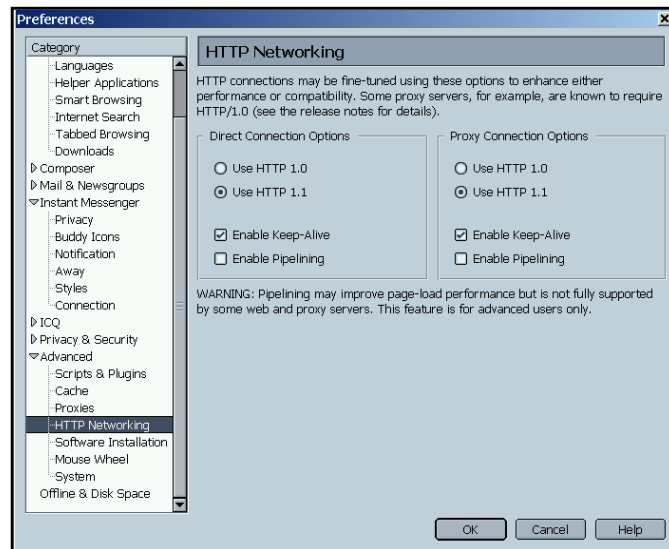
- ✓ Il client invia una nuova richiesta solo quando ha ricevuto la risposta precedente
- ✓ Tempo richiesto: un RTT per ogni oggetto

▪ Con pipelining

- ✓ Default nella versione HTTP/1.1
- ✓ Il client invia una nuova richiesta non appena incontra un riferimento ad un oggetto durante la parsificazione di un file HTML

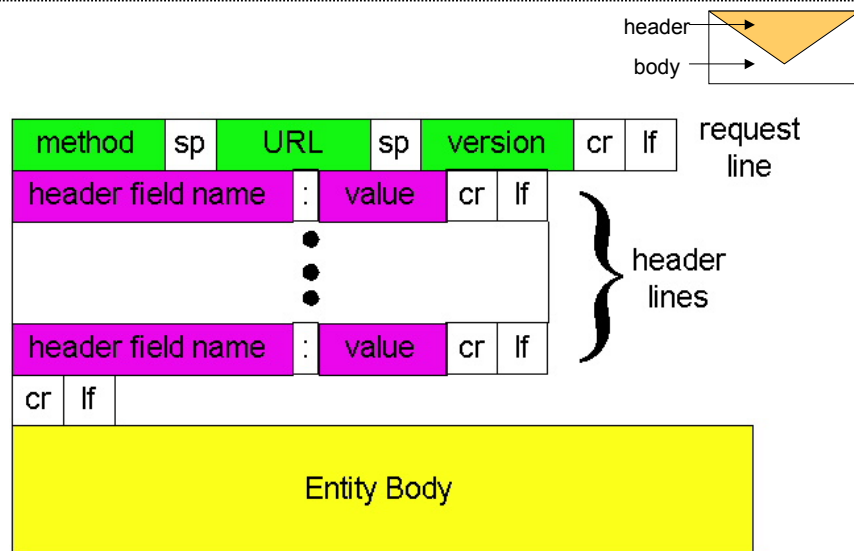
Applicazioni di Rete - M. Ribaudo - DISI

HTTP persistente



Applicazioni di Rete - M. Ribaudo - DISI

HTTP request: formato



Applicazioni di Rete - M. Ribaudo - DISI

HTTP request: formato

```
request line  GET /somedir/page.html HTTP/1.1
header lines  Host: www.someuniv.it
              User-agent: Mozilla/4.0
              Connection: close
              Accept-language: fr
              ...
              carriage return, line feed: indica la fine dell'header
              body, facoltativo nel caso di request
```

Applicazioni di Rete - M. Ribaudo - DISI

Metodi

HTTP/1.0

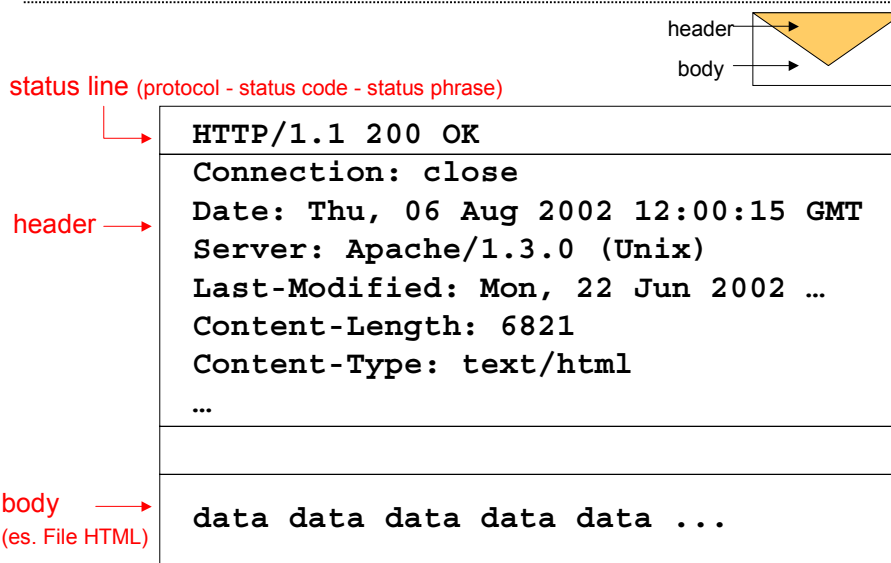
- GET
- POST
- HEAD
 - ✓ Restituisce solo l'header

HTTP/1.1

- GET, POST, HEAD
- PUT
 - ✓ File upload sul server
- DELETE
 - ✓ Per cancellare i file sul server

Applicazioni di Rete - M. Ribaudò - DISI

HTTP response: formato



Applicazioni di Rete - M. Ribaudò - DISI

HTTP response: status code

- 200 OK
- 301 Moved Permanently
- 400 Bad Request
- 404 Not Found
- 505 HTTP Version Not Supported
- ...

Applicazioni di Rete - M. Ribaudò - DISI

HTTP response ??

```
telnet www.disi.unige.it 80
GET /person/RibaudòM/applrete/index.html HTTP/1.1
Host: www.disi.unige.it
```

Dopo aver digitato 2 volte il tasto Enter (Invio) si apre una connessione TCP alla porta 80 del server www.disi.unige.it

Poi viene inviato il comando GET e il server risponde ...

Applicazioni di Rete - M. Ribaudò - DISI

HTTP response !!

```
HTTP/1.1 404 Not Found
Date: Mon, 07 Apr 2003 14:58:05 GMT
Server: Apache/1.3.26 ...
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>404 Not Found</TITLE>
</HEAD><BODY>
<H1>Not Found</H1>
The requested URL /person/Ribaudom/apprete/index.html was not found
on this server.<P>
<HR>
<ADDRESS>Apache/1.3.26 Ben-SSL/1.48 Server at anakin.disi.unige.it Port
80</ADDRESS>
</BODY></HTML>
Connection closed by foreign host.
pianeta 31 ;-)
```

Applicazioni di Rete - M. Ribaudo - DISI

HTTP response !!

GET /person/Ribaudom/ HTTP1.1

.....

```
HTTP/1.1 400 Bad Request
Date: Mon, 07 Apr 2003 14:49:24 GMT
Server: Apache/1.3.26 ...
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>400 Bad Request</TITLE>
</HEAD><BODY>
<H1>Bad Request</H1>
Your browser sent a request that this server could not understand.<P>
The request line contained invalid characters following the protocol string.<P>
<P>
<HR>
<ADDRESS>Apache/1.3.26 Ben-SSL/1.48 Server at anakin.disi.unige.it Port 80</ADDRESS>
</BODY></HTML>
Connection closed by foreign host.
pianeta 26 ;-)
```

Applicazioni di Rete - M. Ribaudo - DISI

HTTP response !!

GET /person/Ribaudom/ HTTP/1.1

.....

HTTP/1.1 400 Bad Request

Date: Mon, 07 Apr 2003 14:55:48 GMT

Server: Apache/1.3.26 ...

Connection: close

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

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
```

```
<HTML><HEAD>
```

```
<TITLE>400 Bad Request</TITLE>
```

```
</HEAD><BODY>
```

```
<H1>Bad Request</H1>
```

```
Your browser sent a request that this server could not understand.<P>
```

```
Invalid URI in request GET person/Ribaudom/ HTTP/1.0<P>
```

```
<HR>
```

```
<ADDRESS>Apache/1.3.26 Ben-SSL/1.48 Server at anakin.disi.unige.it Port 80</ADDRESS>
```

```
</BODY></HTML>
```

Connection closed by foreign host.

pianeta 30 ;-)

Applicazioni di Rete - M. Ribaudo - DISI

HTTP response !!

GET /person/Ribaudom/index.html HTTP/1.1

.....

HTTP/1.1 200 OK

Date: Mon, 07 Apr 2003 14:53:55 GMT

Server: Apache/1.3.26 ...

Connection: close

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

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>
```

```
My home page - DISI
```

```
</TITLE>
```

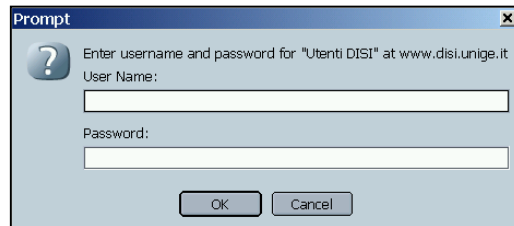
```
...
```

```
...
```

Applicazioni di Rete - M. Ribaudo - DISI

Accesso con autorizzazione

- Spesso quando ci si collega ad un sito web vengono richieste login e password



- Questo accesso si ottiene mediante speciali header HTTP

Applicazioni di Rete - M. Ribaudò - DISI

Accesso con autorizzazione

```
telnet www.someuni.it 80
Trying 130.xxx.yyy.zzz...
Connected to www.somehost.it .
Escape character is '^]'.
HEAD /intranet/index.html HTTP/1.1
Host: www.somehost.it
} request

HTTP/1.1 401 Authorization Required
Date: Tue, 08 Apr 2003 08:38:48 GMT
Server: Apache/1.3.26 Ben-SSL/1.48 (Unix) Debian GNU/Linux
PHP/4.1.2
WWW-Authenticate: Basic realm="A Password is Required to
access these informations"
Content-Type: text/html; charset=iso-8859-1
} response
```

Applicazioni di Rete - M. Ribaudò - DISI

Accesso con autorizzazione

```
telnet www.somehost.it 80
Trying 130.xxx.yyy.zzz...
Connected to www.somehost.it .
Escape character is '^]'.

HEAD /intranet/index.html HTTP/1.1
Host: www.somehost.it
Authorization: Basic VoZcXdEX5s5cVt$=
```

request

codifica di login e password

Dopo aver ricevuto il primo oggetto il **client continua ad inviare login e password al server**. Questi valori sono nella cache del browser e quindi non viene richiesta ogni volta l'autenticazione

Applicazioni di Rete - M. Ribaudò - DISI

Conditional GET

- Il browser salva gli **oggetti** che visualizza nella sua **memoria cache**
- Quando un utente richiede un oggetto che è già nella cache, questo viene visualizzato in modo immediato (es. pulsante Back)
- **Problema:** cosa succede se la copia originale sul server è stata modificata?

Applicazioni di Rete - M. Ribaudò - DISI

Conditional GET

- Il client, ogni volta che memorizza un oggetto nelle cache, tiene traccia anche della data dell'ultima modifica

Last-Modified: <data>

- Ad ogni richiesta successiva il client può specificare la data della copia in cache nella richiesta HTTP

```
GET /dir/nomefile.gif HTTP/1.1
Host: www.somehost.it
If-modified-since: <data>
```

Applicazioni di Rete - M. Ribaudo - DISI

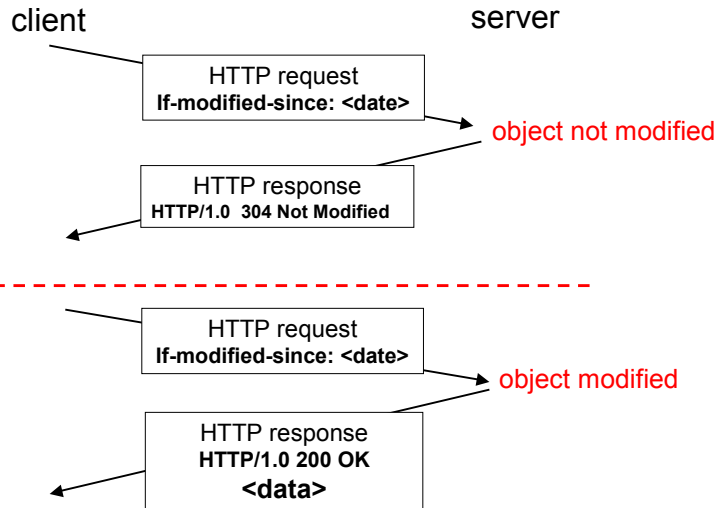
Conditional GET

- Il server invia l'oggetto richiesto solo se è diverso da quello che è già memorizzato nella cache del browser
- Altrimenti restituisce

```
HTTP/1.1 304 Not Modified
Date: Tue, 08 Apr 2003 08:38:48 GMT
Server: Apache/1.3.26 ...
```

Applicazioni di Rete - M. Ribaudo - DISI

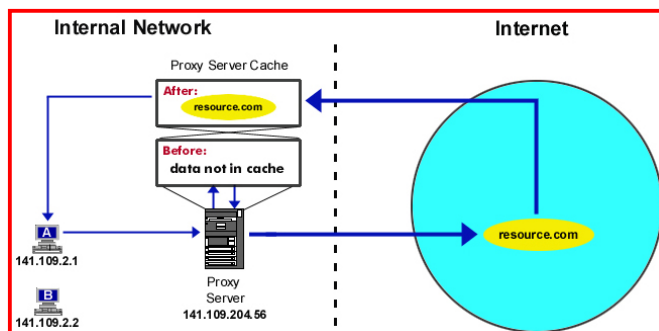
Conditional GET



Applicazioni di Rete - M. Ribauda - DISI

Proxy server: web cache

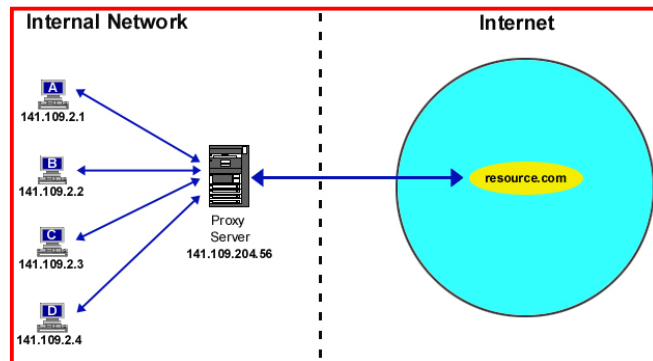
- Permette di **migliorare le prestazioni** dell'accesso ad Internet soddisfacendo, quando possibile, le richieste dei vari client al posto dei server remoti



Applicazioni di Rete - M. Ribauda - DISI

Proxy server: IP masking

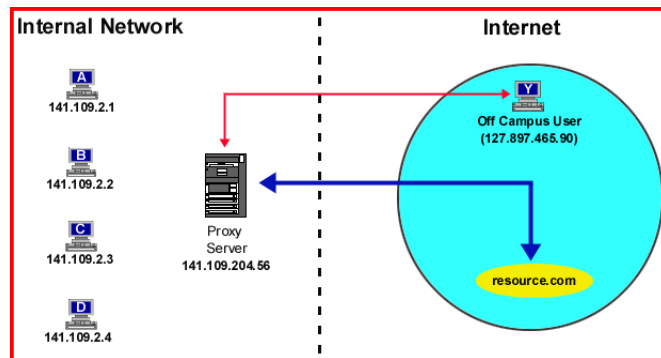
- Inoltre, permette di rendere pubblico all'esterno **un solo indirizzo IP**, quello del proxy



Applicazioni di Rete - M. Ribaud - DISI

Proxy server: access control

- Infine, può anche essere usato per **controllare l'accesso alle risorse**



Applicazioni di Rete - M. Ribaud - DISI