



**UNICAM**  
UNIVERSITÀ DI CAMERINO

**Laurea  
in  
INFORMATICA**

Internet Reti Sicurezza A.A. 2024/2025  
Capitolo 2 – APPLICATION PROTOCOLS  
Fausto Marcantoni  
fausto.marcantoni@unicam.it

1

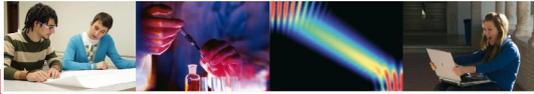


**Dichiarazione di copyright**

*L'utilizzo dei contenuti della lezione sono riservati alla fruizione personale degli studenti iscritti ai corsi dell'Università di Camerino. Sono vietate la diffusione intera o parziale di video o immagini della lezione, nonché la modifica dei contenuti senza il consenso, espresso per iscritto, del titolare o dei titolari dei diritti d'autore e di immagine.*

**Copyright notice**

*The contents of this lesson are subject to copyright and intended only for personal use by students enrolled in courses offered by the University of Camerino. For this reason, any partial or total reproduction, adaptation, modification and/or transformation of the contents of this lesson, by any means, without the prior written authorization of the copyright owner, is strictly prohibited.*



**Fausto Marcantoni** Chapter 1 INTERNET e Reti di Calcolatori 1.2

2

Reti di Elaboratori

## Livello applicativo

**Obiettivi generali:**

- Concetti dei protocolli applicativi
- Implementazioni dei protocolli applicativi
- Paradigma client server
- Modelli dei servizi

**Obiettivi specifici:**

- Protocolli specifici:
  - http
  - ftp
  - smtp
  - pop
  - dns
  - snmp

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.3

3

Reti di Elaboratori

## Applicazioni e protocolli applicativi

**Applicazione:**  
**processi distribuiti in comunicazione**

- In esecuzione su host remoti
- Si scambiano messaggi per eseguire l'applicazione
- Es., mail, FTP, WWW

**Protocolli applicativi**

- Costituiscono una parte di ogni applicazione
- Definiscono il formato dei messaggi scambiati e il loro significato (azioni)
- Usano i servizi degli strati inferiori

The diagram illustrates a multi-tier network architecture. At the top is the 'Rete mobile' (mobile network) with a table listing 'Applicazione', 'Trasporto', 'Rete', 'Collegamento', and 'Fisico'. Below it is the 'Rete domestica' (home network) with a similar table. The 'Rete aziendale' (corporate network) is at the bottom, also with a table. In the middle, there are two ISP levels: 'ISP distrettuale o regionale' (regional) and 'ISP nazionale o internazionale' (national/international), both with tables. Arrows indicate connections between these layers, showing how applications are distributed across different network types and service providers.

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.4

4

Reti di Elaboratori

## Applicazioni: terminologia essenziale

- Un **processo** è un programma in esecuzione su un host
- Sullo **stesso host** i processi comunicano mediante **meccanismi definiti dal SO**.
- Processi in esecuzione su **host diversi** comunicano mediante meccanismi definiti dal **protocollo dello strato di applicazione (application layer protocol)**
- Processi comunicanti**: programmi in esecuzione su due sistemi
- Messaggi**: vengono scambiati tra processi (processo mittente e processo destinatario) su due sistemi terminali attraverso la rete
- Architettura client-server**: per ciascuna coppia di processi comunicanti se ne etichetta uno come **client** e l'altro come **server**.
- Interfaccia tra il processo e la rete**: un processo invia messaggi nella rete e riceve messaggi dalla rete attraverso un'interfaccia software detta **socket**.

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.5

5

Reti di Elaboratori

## Processi attivi - Windows

### tasklist

```

Microsoft Windows [Versione 10.0.15063]
(c) 2017 Microsoft Corporation. Tutti i diritti sono riservati.
C:\Users\fausto.m\fausto>tasklist

```

Nome immagine	PID	Nome sessione	Sessione n.	Utilizzo mem
System Idle Process	0	Services	0	8 K
System	4	Services	0	960 K
smss.exe	444	Services	0	1.176 K
csrss.exe	628	Services	0	5.716 K
wininit.exe	756	Services	0	6.816 K
csrss.exe	764	Console	1	5.944 K
services.exe	832	Services	0	10.608 K
lsass.exe	840	Services	0	15.784 K
svchost.exe	952	Services	0	3.944 K
svchost.exe	976	Services	0	26.016 K
fontdrvhost.exe	1000	Services	0	4.768 K
svchost.exe	528	Services	0	14.384 K
svchost.exe	664	Services	0	6.916 K
winlogon.exe	1032	Console	1	9.844 K
fontdrvhost.exe	1092	Console	1	13.204 K
dwm.exe	1168	Console	1	52.368 K
svchost.exe	1260	Services	0	9.828 K
svchost.exe	1284	Services	0	15.580 K
svchost.exe	1300	Services	0	10.756 K
svchost.exe	1376	Services	0	6.164 K
svchost.exe	1476	Services	0	10.844 K

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.6

6

## Processi attivi - Windows - PowerShell

### Get-Process

```

Windows PowerShell
Copyright (C) Microsoft Corporation. Tutti i diritti riservati.

Prova la nuova PowerShell multipiattaforma https://aka.ms/powershell

PS C:\Users\fausto.m\fausto> Get-Process

Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName
-----
140 9 1752 6924 4.12 0 AdminService
272 14 2500 10072 3.80 0 algs
383 23 20668 30940 0,14 12509 1 ApplicationFrameHost
221 8 1400 920 4.04 0 asanero
104 8 1200 9300 3708 0 AsLdrSvc
339 21 4188 3160 0,30 11112 1 AsusTPCenter
139 10 1600 2544 0,06 3203 1 AsusTPDriver
273 15 2492 2508 0,22 7588 1 AsusTPLoader
389 16 2860 12776 0,30 8268 1 AZRMQ
197 12 10572 16788 0,77 3200 0 audiocd
444 21 34180 43168 0,33 2604 1 BFPinc
456 70 44644 50156 16,13 11449 1 BPrint
397 23 60284 88400 3,20 440 1 chrome
221 15 17180 28896 0,22 1256 1 chrome
262 17 23832 50852 0,38 2766 1 chrome
238 16 15460 43240 0,34 3772 1 chrome
498 38 22272 24860 15,17 6312 1 chrome
375 25 56808 130980 6,86 6684 1 chrome
444 32 128800 140400 10,34 6762 1 chrome
288 23 118408 154124 16,47 7076 1 chrome
1828 59 172920 227064 18,82 7084 1 chrome
212 14 4044 15076 0,15 8052 1 chrome
426 24 25848 43232 6,16 9604 1 chrome
250 19 47212 68100 0,50 10632 1 chrome
209 14 11992 21596 0,09 11748 1 chrome
218 9 1728 7260 0,06 12500 1 chrome
563 28 157104 149140 5,09 12436 1 chrome
227 16 78888 106344 6,38 13112 1 chrome
244 16 23676 50932 0,73 13024 1 chrome
277 18 29544 56704 0,72 14312 1 chrome
248 14 49160 140040 0,08 9432 1 conhost
786 27 3596 6360 0,12 512 1 csrss
832 26 2012 5748 0,02 680 0 csrss
470 17 7148 23552 5,06 8044 1 csrss
567 20 7768 18552 0,74 5748 0 danHost
83 6 928 4556 0,04 6548 0 danHost
235 24 5600 13544 0,14 1368 1 dillohost
146 11 1412 7068 0,05 8340 1 DMedia
222 15 2364 5248 0,08 8920 1 DropboxUpdate
1169 46 141028 215296 12,64 1264 1 dmw
305 25 4212 16364 4,64 0 serenity
    
```

7

## Processi attivi - Linux - BSD

### ps -ef ps -aux - man ps

```

root@pentest: ~
File Edit View Search Terminal Help
root@pentest:~# ps -ef
UID          PID    PPID  C  STIME TTY          TIME CMD
root         1      0  0  09:49 ?        00:00:01 /sbin/init
root         2      0  0  09:49 ?        00:00:00 [kthreadd]
root         4      2  0  09:49 ?        00:00:00 [kworker/0:0H]
root         5      2  0  09:49 ?        00:00:00 [kworker/u256:0]
root         6      2  0  09:49 ?        00:00:00 [mm_percpu_wq]
root         7      2  0  09:49 ?        00:00:00 [ksoftirqd/0]
root         8      2  0  09:49 ?        00:00:01 [rcu_sched]
root         9      2  0  09:49 ?        00:00:00 [rcu_bh]
root        10      2  0  09:49 ?        00:00:00 [migration/0]
root        11      2  0  09:49 ?        00:00:00 [watchdog/0]
root        12      2  0  09:49 ?        00:00:00 [cpuhp/0]
root        13      2  0  09:49 ?        00:00:00 [cpuhp/1]
root        14      2  0  09:49 ?        00:00:00 [watchdog/1]
root        15      2  0  09:49 ?        00:00:00 [migration/1]
root        16      2  0  09:49 ?        00:00:00 [ksoftirqd/1]
root        18      2  0  09:49 ?        00:00:00 [kworker/1:0H]
root        19      2  0  09:49 ?        00:00:00 [cpuhp/2]
root        20      2  0  09:49 ?        00:00:00 [watchdog/2]
root        21      2  0  09:49 ?        00:00:00 [migration/2]
root        22      2  0  09:49 ?        00:00:00 [ksoftirqd/2]
root        23      2  0  09:49 ?        00:00:00 [kworker/2:0]
root        24      2  0  09:49 ?        00:00:00 [kworker/2:0H]
root        25      2  0  09:49 ?        00:00:00 [cpuhp/3]
root        26      2  0  09:49 ?        00:00:00 [watchdog/3]
root        27      2  0  09:49 ?        00:00:00 [migration/3]
root        28      2  0  09:49 ?        00:00:00 [ksoftirqd/3]
    
```

8

Reti di Elaboratori

## Processi attivi - MacOS

ps -ef

```

Last login: Fri Oct 4 16:37:52 on console
faustomarcantoni@MacBook-Pro-di-Fausto ~ % ps -aux
ps: No user named 'x'
faustomarcantoni@MacBook-Pro-di-Fausto ~ % ps -ef
UID    PID  PPID  C  STIME  TTY          TIME CMD
0      1    0    0 Ven04pm ??        26:52.46 /sbin/launchd
0      573  1    0 Ven04pm ??        24:00.54 /usr/libexec/launchd
0      574  1    0 Ven04pm ??        0:00.12 /usr/libexec/smd
0      575  1    0 Ven04pm ??        5:34.13 /usr/libexec/UserEventAgent (System)
0      577  1    0 Ven04pm ??        4:15.07 /System/Library/Frameworks/CoreServices.framework
0      578  1    0 Ven04pm ??        1:12.51 /System/Library/PrivateFrameworks/MediaRemote.fr
0      581  1    0 Ven04pm ??        0:49.44 /usr/sbin/systemstats --daemon
278    583  1    0 Ven04pm ??        0:11.71 /System/Library/PrivateFrameworks/MobileAccesso
0      584  1    0 Ven04pm ??        61:37.15 /usr/libexec/configd
0      585  1    0 Ven04pm ??        0:00.02 endpointsecurityd
0      586  1    0 Ven04pm ??        29:12.05 /System/Library/CoreServices/powerd.bundle/power
0      587  1    0 Ven04pm ??        0:28.45 /usr/libexec/IOMFB_bics_daemon
0      589  1    0 Ven04pm ??        0:01.58 /usr/libexec/amfid
0      591  1    0 Ven04pm ??        0:03.25 /usr/libexec/remoted
0      593  1    0 Ven04pm ??        0:00.07 /usr/libexec/keybagd -t 15
200    594  1    0 Ven04pm ??        0:02.81 /System/Library/PrivateFrameworks/MobileSoftwa
0      596  1    0 Ven04pm ??        0:00.68 /usr/libexec/watchdogd
0      600  1    0 Ven04pm ??        10:37.66 /System/Library/Frameworks/CoreServices.framework
240    601  1    0 Ven04pm ??        0:00.91 /System/Library/CoreServices/iconservicesd
0      602  1    0 Ven04pm ??        0:00.78 /usr/libexec/kernelmanagerd
0      603  1    0 Ven04pm ??        0:12.02 /usr/libexec/diskarbitrationd
0      606  1    0 Ven04pm ??        2:13.86 /usr/libexec/coreduetd
0      607  1    0 Ven04pm ??        9:27.97 /usr/sbin/syslogd
0      610  1    0 Ven04pm ??        0:49.56 /usr/libexec/thermalmonitord
0      611  1    0 Ven04pm ??        41:57.32 /usr/libexec/opendirectoryd
0      613  1    0 Ven04pm ??        12:04.64 /System/Library/PrivateFrameworks/ApplePushServi

```

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.9

9

Reti di Elaboratori

## Processi attivi - Linux - BSD

top

```

studente@debian: ~
File Edit View Search Terminal Help
top - 15:46:17 up 2 min, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 262 total, 1 running, 261 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.2 us, 0.0 sy, 0.0 ni, 99.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 1978.0 total, 1120.0 free, 579.1 used, 365.9 buff/cache
MiB Swap: 976.0 total, 976.0 free, 0.0 used, 1398.9 avail Mem

  PID USER   PR  NI  VIRT  RES  SHR  S  %CPU  %MEM  TIME+  COMMAND
1608 studente 20   0 11680 5080 2924 R   0.3   0.3   0:00.22 top
1 root      20   0 185088 11660 8464 S   0.0   0.6   0:00.21 systemd
2 root      20   0      0      0      0 S   0.0   0.0   0:00.00 kthreadd
3 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 rcu_gp
4 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
5 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
6 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 netns
7 root      20   0      0      0      0 I   0.0   0.0   0:00.01 kworker/0:0-cgwb_release
8 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
9 root      20   0      0      0      0 I   0.0   0.0   0:00.16 kworker/u4:0-events_unbound
10 root     0 -20      0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
11 root     20   0      0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_rude_kthread
12 root     20   0      0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_trace_kthread
13 root     20   0      0      0      0 S   0.0   0.0   0:00.01 ksoftirqd/0
14 root     20   0      0      0      0 I   0.0   0.0   0:00.04 rcu_sched
15 root     rt   0      0      0      0 S   0.0   0.0   0:00.00 migration/0
16 root     20   0      0      0      0 I   0.0   0.0   0:00.00 kworker/0:1-events
17 root     20   0      0      0      0 S   0.0   0.0   0:00.00 cpuhp/0

```

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.10

10

Reti di Elaboratori

## Processi Attivi - MacOS

top

Processes: 871 total, 3 running, 868 sleeping, 4687 threads  
 Load Avg: 4.68, 4.18, 3.35 CPU usage: 5.2% user, 4.44% sys, 99.53% idle SharedLibs: 1215M resident, 164M data, 359M linkedit.  
 MemRegions: 313662 total, 8883M resident, 1187M private, 110 shared. PhysMem: 41G used (3951M wired, 936M compressor), 85G unused.  
 VM: 3617 vsize, 5486M framework vsize, 113246(0) swpans, 124428(0) swpouts. Networks: packets: 6418742/2388M in, 13263654/5625M out.  
 Disks: 1868277/54G read, 6325878/780 written.

PID	COMMAND	%CPU	TIME	#TH	#WQ	#PORT	MEM	PURG	CMPRS	PGRP	PPID	STATE	BOOSTS	%CPU_ME	%CPU_OTHRS	UID
641	WindowServer	56.0	01:42:12	29/1	5	4362	1317M	280M-	11M	641	1	running	*0(1)	0.01880	1.94269	88
1435	Google Drive	44.1	01:14:00	127	16	616+	381M+	854K	292M	1410	1410	sleeping	*0(93888)	1.60212	0.00000	501
0	kernel_task	15.6	05:01:34	851/16	0	0	12M	0B	0B	0	0	running	*0(0)	0.00000	0.00000	0
11262	top	6.3	00:01:17	1/1	0	30	18M	0B	0B	11262	11218	running	*0(1)	0.00000	0.00000	0
10737	Microsoft Po	3.4	01:03:36	42	8	1082	581M+	1005M	0B	10737	1	sleeping	*139588+{661}	2.88733	0.00000	501
18834	VTEncoderIPC	3.1	00:05:08	5	2	87	13M	0B	0B	18834	1	sleeping	*0(40+{6}	0.00000	2.78338	501
10264	Google Chrom	2.6	00:53:46	30	1	342+	176M+	0B	0B	10249	10249	sleeping	*0(3)	0.00000	0.00000	501
895	at.obdev.lit	1.5	14:54:02	6	4	218	1841M	0B	896M	895	1	sleeping	*0(1)	0.05172	0.16139	0
11247	plugin-conta	1.3	00:02:40	30	1	185-	98M-	0B	0B	10444	10444	sleeping	*0(2)	0.00000	0.00000	501
10444	firefox	1.3	01:48:51	122	3	980	858M	457M	0B	10444	1	sleeping	*0(510)	0.00000	0.00000	501
10256	Google Chrom	1.3	00:48:08	23	1	203	46M+	0B	0B	10249	10249	sleeping	*0(3)	0.00000	0.00000	501
10249	Google Chrom	1.2	01:46:68	40	1	876	166M+	1616K	0B	10249	1	sleeping	*0(791)	0.00000	0.00000	501
10391	AdobeAcrobat	1.1	02:10:53	43	8	514	490M+	242M	0B	10391	1	sleeping	*0(815)	1.72193	0.00000	501
10358	com.apple.Dr	0.9	00:58:74	3	2	588	5616K	0B	0B	10358	1	sleeping	*0(1)	0.00000	0.00000	270
10265	Google Chrom	0.9	01:25:23	27	1	313	215M+	0B	0B	10249	10249	sleeping	*0(3)	0.00000	0.00000	501
894	com.cisco.an	0.9	13:55:58	7	4	234+	393M+	0B	316M	894	1	sleeping	*0(1)	0.19428	0.00000	0
8254	bluetoothd	0.7	03:36:38	11	5	414	18M	224K	0B	8254	1	sleeping	*0(1)	0.37888	0.00000	0
967	distnoted	0.7	02:10:84	4	3	710	483K	0B	848K	967	1	sleeping	*0(1)	0.00000	0.79243	501
573	logd	0.7	24:02:44	4	3	2858	8482K	0B	5504K	573	1	sleeping	*0(1)	0.00000	0.00000	0
10255	Google Chrom	0.7	01:33:15	15	2	260	157M+	495M	0B	10249	10249	sleeping	*1(12)	0.00000	0.00000	501
425	distnoted	0.4	01:59:24	3	2	252	2753K	0B	1024K	425	1	sleeping	*0(1)	0.00000	0.40976	270
754	com.apple.Dr	0.4	01:40:11	8	6	1525	33M-	0B	3168K	754	1	sleeping	*0(1)	0.00000	0.02211	270
11208	Terminal	0.4	00:07:34	9	4	358	416M+	314M-	0B	11208	1	sleeping	*0(71+)	0.84761	0.00785	501
687	airpord	0.3	06:20:15	8	6	3634	11M	0B	1840K	687	1	sleeping	*5(1556)	0.17341	0.00000	0
974	cprefsd	0.3	00:48:75	3	2	878+	4113K+	1392K	624K	974	1	sleeping	*0(28072+)	0.00000	0.02862	501
691	symptomsd	0.3	12:33:29	4	3	182	7569K	304K	1024K	691	1	sleeping	*0(88366+)	0.00000	0.06721	24
9650	com.apple.Am	0.3	00:53:38	3	1	88	3809K	0B	0B	9650	1	sleeping	*0(24133)	0.12089	0.00000	0
577	fsventd	0.3	04:15:57	19	1	164-	4453K-	0B	894K	577	1	sleeping	*0(1)	0.00000	0.00000	0
11252	plugin-conta	0.2	00:00:66	30	1	112	44M	0B	0B	10444	10444	sleeping	*0(2)	0.00000	0.00000	501
637	corebrightne	0.2	02:00:00	4	3	142	5217K	0B	944K	637	1	sleeping	*0(1)	0.15792	0.00000	0
1353	Finder	0.2	07:14:37	7	4	639	218M+	236M	40M	1353	1	sleeping	*0(132213)	0.00000	0.00000	501
10396	AcroCEF Help	0.2	00:22:51	10	2	180	79M	370M	0B	10392	10392	sleeping	*1(5)	0.00000	0.00000	501
838	locationd	0.1	77:45:19	9	6	328	11M	768K	1448K	838	1	sleeping	*0(851418+)	0.00000	0.12520	205
1173	nearbyd	0.1	01:22:74	9	7	108	5441K	0B	1824K	1173	1	sleeping	*4(30)	0.00000	0.09585	268

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.11

11

Reti di Elaboratori

## Windows Sysinternals

Windows Sysinternals

18/08/2021 • 3 minuti per la lettura

In questo articolo

Sysinternals Live

What's New

The Sysinternals web site was created in 1996 by Mark Russinovich to host his advanced system utilities and technical information. Whether you're an IT Pro or a developer, you'll find Sysinternals utilities to help you manage, troubleshoot and diagnose your Windows systems and applications.

- Read the official guide to the Sysinternals tools, [Troubleshooting with the Windows Sysinternals Tools](#)
- Read the [Sysinternals Blog](#) for a detailed change feed of tool updates
- Watch Mark's [Sysinternals Update videos on YouTube](#)
- Watch Mark's top-rated [Case-of-the-Unexplained](#) troubleshooting presentations and other webcasts
- Read [Mark's Blog](#) which highlight use of the tools to solve real

<https://learn.microsoft.com/it-it/sysinternals/downloads/sysinternals-suite>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.12

12



Reti di Elaboratori

## Applicazioni: terminologia essenziale

- Un **agente utente (user agent)** è:
  - un'interfaccia tra l'utente e l'applicazione di rete
  - Browser Web
  - E-mail: lettore di posta
  - streaming audio/video: lettore di file audio/video



Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.15

15

Reti di Elaboratori

## Paradigma client-server

Applicazione di rete tipica consiste di due parti: **client e server**

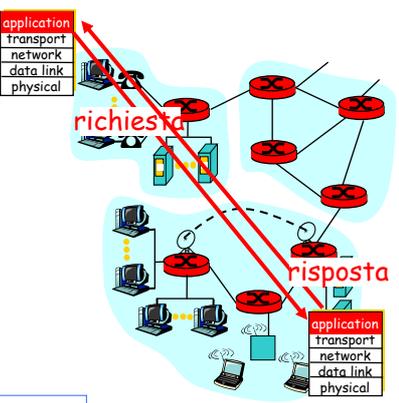
**Client:**

- Inizia il dialogo col server ("speaks first")
- Di solito richiede un servizio
- Nel caso del Web, il client è integrato nel browser

**Server:**

- Fornisce il servizio al client, su richiesta
- E' sempre pronto per le richieste

Es., un Web server invia una pagina Web richiesta, un mail server accede alla casella di posta elettronica



Esempi di sistemi client/server:

- Database server:** per la gestione di grandi moli di dati
- File server:** per la condivisione dei file
- FTP server:** per la gestione dell'upload/download dei file
- Groupware:** per la gestione d'informazioni riguardanti gruppi di lavoro
- Print server:** per la condivisione delle stampanti
- Web server:** per la gestione dell'interazione via web tra server e client

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.16

16

Reti di Elaboratori

## Protocolli di livello applicativo: servizi dagli strati inferiori e identificazione

**API: Application Programming Interface**

- Definisce l'interfaccia tra applicazione e strato di trasporto
- Socket: API Internet
  - Due processi (applicazione nel modello client server) comunicano inviando/leggendo dati nel/dal socket



come può un processo "identificare" quello con cui intende comunicare?

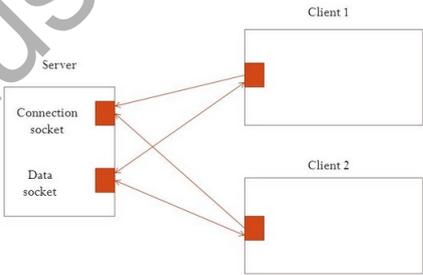
- indirizzo IP** dell'host su cui l'altro processo è in esecuzione
- numero di porta (port number)** permette all'host ricevente di identificare il processo locale destinatario del messaggio

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.17

17

Reti di Elaboratori

## Socket



```

Socket socket = getSocket(type = "TCP")
connect(socket, address = "1.2.3.4", port = "80")
send(socket, "Hello, world!")
close(socket)

```

<https://tools.ietf.org/html/rfc147>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.18

18

Reti di Elaboratori

## LA CLASSE Socket

Costruire una Socket significa aprire la comunicazione verso l'altra parte

```
public Socket(String remoteHost, int remotePort)
```

Crea una socket stream e la collega alla porta specificata della macchina remota corrispondente al nome dato

Esempio

```
Socket s = new Socket("www.nasa.gov", 8080);
```

```
public InputStream getInputStream()
- restituisce lo stream di input da cui leggere i dati (byte) che giungono dall'altra parte
public OutputStream getOutputStream()
- restituisce lo stream di output su cui scrivere i dati (byte) da inviare all'altra parte
public synchronized void close()
- chiude la connessione e libera la risorsa
```

<http://www0.mi.infn.it/~cmp/CorsoReti/slides03/Lab3-sockets-java.pdf>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.19

19

Reti di Elaboratori

## Socket: funzionamento di base

Host o server

scambio di messaggi

Host o server

Controllato dallo sviluppatore dell'applicazione

Processo

Controllato dallo sviluppatore dell'applicazione

Socket

Controllato dal sistema operativo

TCP con buffer e variabili

Internet

TCP con buffer e variabili

Controllato dal sistema operativo

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.20

20

## Socket: funzionamento di base

Un protocollo a livello di applicazione definisce come i processi di un'applicazione, in esecuzione su sistemi terminali diversi, si scambiano i messaggi.

- ❖ tipo di messaggio (di richiesta o di risposta)
- ❖ sintassi del messaggio (campi del messaggio)
- ❖ semantica del messaggio (significato delle informazioni dei campi)
- ❖ regole (come e quando avviene lo scambio dei messaggi)

## netstat

```

C:\Users\Fausto> netstat -an
Connessioni attive
Proto Indirizzo locale      Indirizzo esterno  Stato
TCP    0.0.0.0:135             0.0.0.0:0          LISTENING
TCP    0.0.0.0:443             0.0.0.0:0          LISTENING
TCP    0.0.0.0:445             0.0.0.0:0          LISTENING
TCP    0.0.0.0:803            0.0.0.0:0          LISTENING
TCP    0.0.0.0:913            0.0.0.0:0          LISTENING
TCP    0.0.0.0:1536           0.0.0.0:0          LISTENING
TCP    0.0.0.0:1537           0.0.0.0:0          LISTENING
TCP    0.0.0.0:1538           0.0.0.0:0          LISTENING
TCP    0.0.0.0:1539           0.0.0.0:0          LISTENING
TCP    0.0.0.0:1540           0.0.0.0:0          LISTENING
TCP    0.0.0.0:1544           0.0.0.0:0          LISTENING
TCP    0.0.0.0:5040           0.0.0.0:0          LISTENING
TCP    0.0.0.0:5557           0.0.0.0:0          LISTENING
TCP    127.0.0.1:443          127.0.0.1:2794     ESTABLISHED
TCP    127.0.0.1:1413        0.0.0.0:0          LISTENING
TCP    127.0.0.1:1636        127.0.0.1:1637     ESTABLISHED
TCP    127.0.0.1:1637        127.0.0.1:1636     ESTABLISHED
TCP    127.0.0.1:1638        127.0.0.1:1639     ESTABLISHED
TCP    127.0.0.1:1639        127.0.0.1:1638     ESTABLISHED
TCP    127.0.0.1:1640        127.0.0.1:1641     ESTABLISHED
TCP    127.0.0.1:1641        127.0.0.1:1640     ESTABLISHED
TCP    127.0.0.1:1642        127.0.0.1:1643     ESTABLISHED
TCP    127.0.0.1:1643        127.0.0.1:1642     ESTABLISHED
TCP    127.0.0.1:1644        127.0.0.1:1645     ESTABLISHED
TCP    127.0.0.1:1645        127.0.0.1:1644     ESTABLISHED
TCP    127.0.0.1:1646        127.0.0.1:1647     ESTABLISHED
TCP    127.0.0.1:1647        127.0.0.1:1646     ESTABLISHED
  
```

```

C:\Users\Fausto> netstat /?
Visualizza le statistiche del protocollo e le connessioni di rete TCP/IP correnti.
NETSTAT [-a] [-b] [-e] [-f] [-n] [-o] [-p proto] [-r] [-s] [-t] [-x] [-y] [interval]
-a Visualizza tutte le connessioni e le porte di ascolto.
-b Visualizza l'eseguibile coinvolto nella creazione di ogni connessione o
  porta di ascolto; in alcuni casi, host di eseguibili noti
  più componenti indipendenti e in questi casi il
  sequenza di componenti coinvolti nella creazione della connessione
  o la porta in ascolto; in questo caso, l'eseguibile
  il nome è in [] nella parte inferiore, in alto è il componente che ha chiamato
  e così via fino al raggiungimento di TCP/IP. Si noti che questa opzione
  può richiedere molto tempo e avrà esito negativo, a meno che non siano sufficienti
  autorizzazioni.
-e Visualizza le statistiche Ethernet. È possibile combinare
  opzioni.
-f Visualizza nomi di dominio completi (FQDN) per stranieri
  indirizzi.
-n Visualizza indirizzi e numeri di porta in formato numerico.
-o Visualizza l'ID del processo proprietario associato a ogni connessione.
-p proto Mostra le connessioni per il protocollo specificato da proto; proto
  può essere qualsiasi: TCP, UDP, ICMPv6 o UDPv6. Se usato con-s
  opzione per la visualizzazione delle statistiche per protocollo, Proto può essere
  qualsiasi:
  IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP o UDPv6.
-q Visualizza tutte le connessioni, le porte di ascolto e i binding
  non in ascolto di porte TCP. Le porte di nonlistening associate possono o meno
  essere
  
```

## Netstat in linux

sudo apt install net-tools

```

studente@debian:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 debian.administr:bootp MANILA2016.Amin:bootps ESTABLISHED

Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags   Type       State         I-Node    Path
unix    3      [ ]     STREAM    CONNECTED    17726
unix    3      [ ]     STREAM    CONNECTED    17478
unix    3      [ ]     STREAM    CONNECTED    17363    @/tmp/.X11-unix/X0
unix    3      [ ]     STREAM    CONNECTED    16476
unix    3      [ ]     STREAM    CONNECTED    15582    /run/dbus/system_bus_socket
unix    3      [ ]     STREAM    CONNECTED    17608
unix    3      [ ]     STREAM    CONNECTED    16156
unix    3      [ ]     STREAM    CONNECTED    18607
unix    3      [ ]     STREAM    CONNECTED    18571    @/tmp/.ICE-unix/1128
unix    3      [ ]     STREAM    CONNECTED    17599    /run/user/1000/at-spi/bus_0
unix    3      [ ]     STREAM    CONNECTED    18465
unix    3      [ ]     STREAM    CONNECTED    18518
unix    3      [ ]     STREAM    CONNECTED    17468
unix    3      [ ]     DGRAM     CONNECTED    12722
unix    3      [ ]     STREAM    CONNECTED    16469    /run/systemd/journal/stdout
   3      [ ]     STREAM    CONNECTED    16883
   3      [ ]     STREAM    CONNECTED    15585    /run/dbus/system_bus_socket
   3      [ ]     STREAM    CONNECTED    13307
   3      [ ]     STREAM    CONNECTED    17091    /run/cups/cups_sock

```

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.23

23

## ss - another utility to investigate sockets

```

Shell No.1
File Actions Edit View Help
ss(8) System Manager's Manual ss(8)
NAME
  ss - another utility to investigate sockets
SYNOPSIS
  ss [options] [FILTER]
DESCRIPTION
  ss is used to dump socket stat:
  stat. It can display more TCP or
  UDP connections.
OPTIONS
  When no option is used ss displ:
  TCP/UNIX/UDP) that have establish:
  -h, --help          Show summary of options.
  -V, --version       Output version information
  -H, --no-header     Suppress header line.
  -O, --oneline       Manual page ss(8) line 1 (press h for

```

```

root@kali:~# ss -ant
State Recv-Q Send-Q Local Address:Port Peer Address:Port Process
root@kali:~# ss -ant
State Recv-Q Send-Q Local Address:Port Peer Address:Port Process
ESTAB 0 0 193.205.92.191:56630 93.184.220.29:80
ESTAB 0 0 193.205.92.191:21100 13.35.43.91:443
ESTAB 0 0 193.205.92.191:37348 99.86.154.2:443
ESTAB 0 0 193.205.92.191:40834 34.216.82.69:443
root@kali:~#

```

man ss

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.24

24

Reti di Elaboratori

# TCPView

TCPView

2.25

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL

25

Reti di Elaboratori

# Requisiti delle applicazioni

**Perdita dati (Data loss)**

- Alcune applicazioni (es., audio) sono tolleranti (fino a un certo punto)
- Altre (es., Web, FTP, telnet) richiedono affidabilità totale

**Ritardo**

- Alcune applicazioni (es., telefonia Internet, giochi interattivi in rete) richiedono un ritardo minimo per funzionare con qualità sufficiente
- Altre non ammettono ritardi (es., telefonia su internet, IPTV)

**Banda**

- Alcune applicazioni (soprattutto multimediali) richiedono una banda minima per poter "funzionare"
- Altre (dette "elastiche") usano la banda a disposizione

**Nota:** alcuni requisiti sono determinati da esigenze percettive umane (es. ritardo nella telefonia Internet)

2.26

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL

26

Reti di Elaboratori

## Requisiti di alcune applicazioni di rete

Applicazione	Tolleranza alla perdita di dati	Ampiezza di banda	Sensibilità al tempo
Trasferimento file	No	Variabile	No
E-mail	No	Variabile	No
Documento Web	No	Variabile (pochi Kbps)	No
Audio/Video in tempo reale	Si	Audio: dal pochi Kbps a 1 Mbps Video: da 10 Kbps a 5 Mbps	Si: centinaia di ms.
Audio/Video memorizzati	Si	Audio: dal pochi Kbps a 1 Mbps Video: da 10 Kbps a 5 Mbps	Si: pochi secondi
Giochi interattivi	Si	Fino a 10 Kbps ( <i>se basta???</i> )	Si: centinaia di ms.
Messaggistica istantanea	No	Variabile	Si e No

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.27

27

Reti di Elaboratori

## Servizi offerti dai protocolli di trasporto Internet

### Servizio TCP :

- **Orientato alla connessione:**
  - richiesto "setup" tra client e server
- **Trasporto affidabile (reliable transfer):**
  - tra processi mittente e ricevente
- **Controllo di flusso (flow control):**
  - il mittente non sommerge il ricevente
- **Controllo della congestione (congestion control):**
  - si limita il mittente quando la rete è sovraccarica
- **Non offre:**
  - garanzie di banda e ritardo minimo

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.28

28

Reti di Elaboratori

## Servizi offerti dai protocolli di trasporto Internet

### Servizio UDP :

- *Trasporto non affidabile tra processi mittente e ricevente*
- **Non offre:**
  - connessione, affidabilità, controllo di flusso, controllo di congestione, garanzie di ritardo e banda

**Perché esiste/si usa UDP?**

- Nessun ritardo dovuto al setup di connessione
- Semplicità dovuta all'assenza di stato di connessione
- Pochi dati di intestazione del segmento
- Nessun controllo di congestione
- Informazioni spedite a raffica

Può essere conveniente per le applicazioni multimediali quali audio, video, videoconferenza, telefonia internet, tutte applicazioni in cui è possibile avere una tolleranza nella perdita di dati.

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.29

29

Reti di Elaboratori

## Applicazioni Internet: loro protocolli e protocollo di trasporto usato

Applicazione	Protocollo Applicativo	Protocollo di Trasporto
e-mail	smtp [RFC 821]	TCP
remote terminal	telnet [RFC 854]	TCP
www	http [RFC 2068]	TCP
name server	dns [RFC 1034 e RFC 1035]	TCP - UDP
file transfer	ftp [RFC 959]	TCP
management protocol	snmp [RFC 1157 e RFC 3416]	UDP
streaming multimedia	proprietario (es. RealNetworks)	TCP - UDP
remote file server	nfs [RFC 1094, 1813, 3530]	TCP - UDP
internet telephony	VoIP(RTP-Real-time Transport Protocol)	UDP

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.30

30

Reti di Elaboratori

## Protocolli a livello di applicazione

Un **protocollo a livello di applicazione** definisce come i processi di un'applicazione, in esecuzione su sistemi periferici diversi, si scambiano i messaggi.

È importante distinguere tra:

- **applicazioni di rete**
- **protocolli a livello di applicazione**

applicazioni di rete	protocolli	
il Web	http	Hypertext Transfer Protocol
la posta elettronica	smtp	Simple Mail Transfer Protocol
il servizio di directory	ldap	Lightweight Directory Access Protocol
il trasferimento di file	ftp	File Transport Protocol
le applicazioni P2P	torrent	proprietario

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.31

31

Reti di Elaboratori

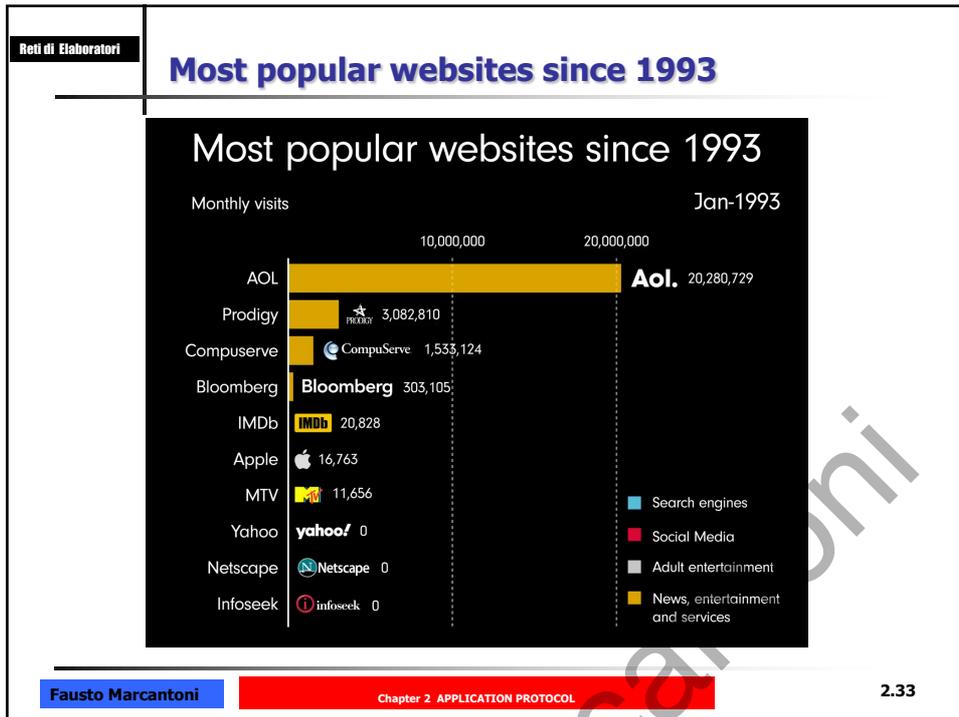
## WEB - WWW - World Wide Web

WEB - WWW - World Wide Web

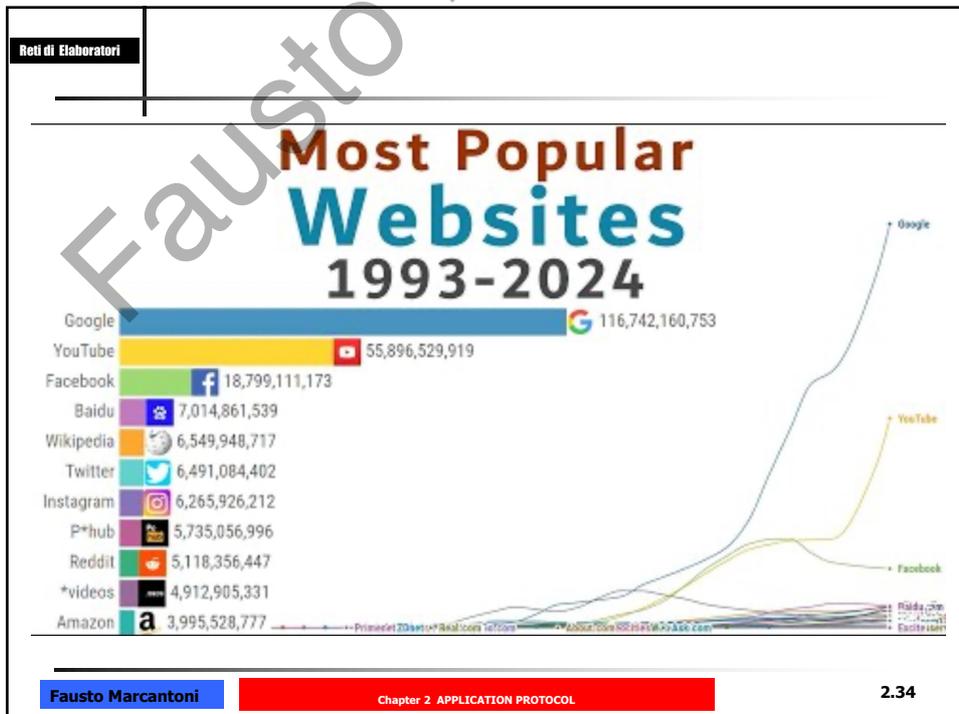


Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.32

32



33



34

Reti di Elaboratori

## Web e HTTP

World Wide Web

Il Web è implementato attraverso un insieme di standard, i principali dei quali sono i seguenti:

- HTML: il linguaggio di markup con cui sono scritte e descritte le pagine web;
- HTTP: il protocollo di rete appartenente al livello di applicazione del modello ISO/OSI su cui è basato il Web;
- URL: lo schema di identificazione, e quindi di rintracciabilità, dei contenuti e dei servizi del Web.

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.35

35

Reti di Elaboratori

## Ipertesto

Un **ipertesto** è un insieme di documenti messi in relazione tra loro per mezzo di parole chiave.

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.36

36

Reti di Elaboratori

## WWW: terminologia essenziale

- Pagina Web:
  - È costituita da "oggetti" (di solito: pagina HTML iniziale+oggetti indirizzati)
  - È indirizzata da una URL
- URL (Uniform Resource Locator)
  - Identifica un oggetto nella rete e specifica il modo per accedere ad esso
  - Ha tre componenti:
    - protocollo
    - nome dell'host
    - percorso nell'host

<http://www.someSchool.edu/someDept/pic.gif>

 <http://tools.ietf.org/html/rfc3986>

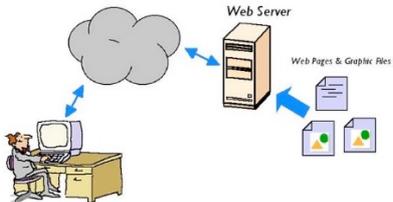
Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.37

37

Reti di Elaboratori

## WWW: terminologia essenziale

- Uno user agent per il Web è detto browser:
  - MS Internet Explorer/Edge/Cromium
  - Mozilla Firefox
  - Opera
  - Chrome
  - Safari
  - Brave
- Un server per il Web è detto Web server:
  - Apache (pubblico dominio)
  - MS Internet Information Server
  - lighttpd
  - nginx



Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.38

38

Reti di Elaboratori

## WWW: terminologia essenziale

protocol      hostname      directory      filename      query parameters

http://www.mywebsite.com/apparel/skirt.php?sku=123&lang=en&sect=silk

domain name      URI

**Protocollo:** Visualizza il protocollo utilizzato per l'accesso al server. I protocolli più comuni sono l'HTTP, HTTPS, FTP, MMS ecc. Se il protocollo non viene specificato generalmente il browser utilizza "HTTP://" come predefinito.

**Hostname:** visualizza l'indirizzo fisico del server su cui risiede la risorsa. Come detto in precedenza può essere costituito da un nome di dominio o da un Indirizzo IP.

**Directory:** è opzionale e, se presente, indica il pathname nel file system del server che identifica la risorsa, che generalmente è una pagina web, un'immagine o un file multimediale.

**Filename:** anche questo opzionale, visualizza il nome e l'estensione della risorsa che stiamo visualizzando.

**Querystring:** opzionale, se richiesto al termine dell'url è possibile aggiungere una query string separandola utilizzando il simbolo "?". La querstring è una stringa di caratteri che consente di passare al server uno o più parametri.

Fausto Marcantoni      Chapter 2 APPLICATION PROTOCOL      2.39

39

Reti di Elaboratori

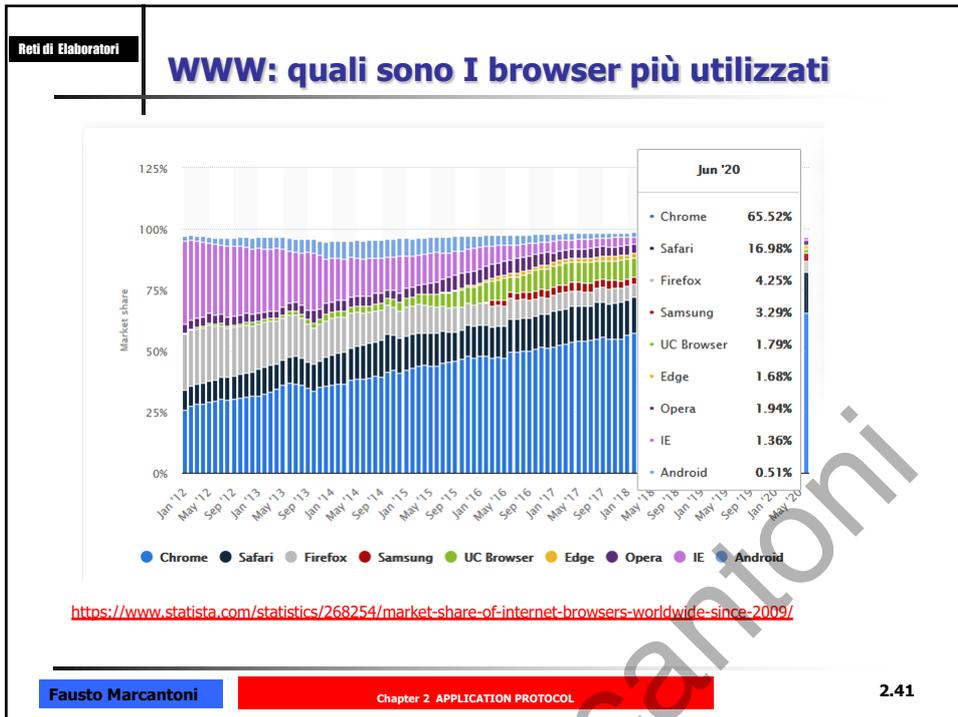
## WWW: quali sono I browser più utilizzati

BROWSER	TOTAL MARKET SHARE
<input checked="" type="checkbox"/> Chrome	59.57%
<input checked="" type="checkbox"/> Microsoft Internet Explorer	16.50%
<input checked="" type="checkbox"/> Firefox	12.32%
<input checked="" type="checkbox"/> Microsoft Edge	5.65%
<input checked="" type="checkbox"/> Safari	3.66%
<input checked="" type="checkbox"/> Opera	1.21%
Proprietary or Undetectable	0.30%
<input checked="" type="checkbox"/> Konqueror	0.01%
<input checked="" type="checkbox"/> Android Browser	0.00%
<input checked="" type="checkbox"/> Obigo	0.00%

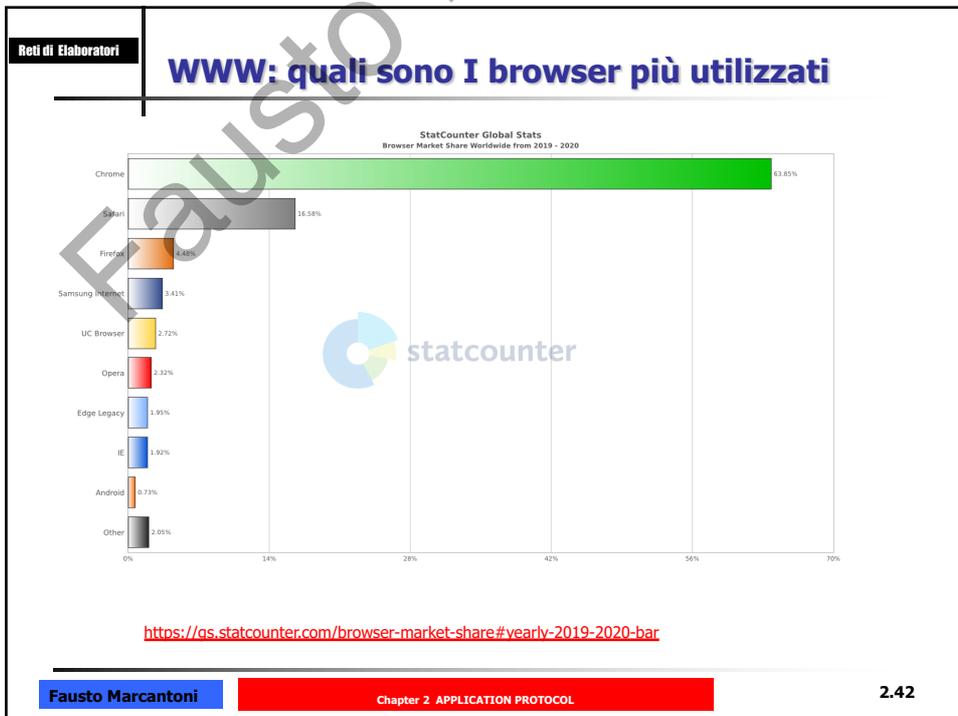
<http://marketshare.hitslink.com/report.aspx?qprid=0>

Fausto Marcantoni      Chapter 2 APPLICATION PROTOCOL      2.40

40



41



42

Reti di Elaboratori

## Mosaic

<https://www.my-internet-explorer.com/mosaic/>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.43

43

Reti di Elaboratori

## The Rise and Fall of Popular Web Browsers Since 1994

Web browsers for the last 28 years

Internet users globally:  
14,073,444

Internet user globally:

Browser	Market share (%)
Mosaic	97.0%
Other	3.0%

January 1994 Market share (%)

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.44

44

Reti di Elaboratori

## Il Web: protocollo http

**http: hypertext transfer protocol**

- Protocollo di livello applicativo per il Web
- Usa il modello client/server
  - client:** browser che richiede, riceve e "mostra" oggetti Web
  - server:** Web server che invia oggetti in risposta alle richieste
- [http1.0: RFC 1945](#)
- [http1.1: RFC 2068](#)

Server con web server Apache

PC con Internet Explorer

Linux con Firefox

Richiesta HTTP

Risposta HTTP

Richiesta HTTP

Risposta HTTP

HTML ≠ HTTP

- <http://www.w3.org/>
- <http://www.w3.org/MarkUp/>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.45

45

Reti di Elaboratori

## HTML 5

1476 pagine

HTML Living Standard — Last Updated 4 October 2023

One-Page Version html.spec.whatwg.org	Multipage Version multipage
Version for Web Devs New	PDF Version pdf
Translations 日本語 · 简体中文	FAC on GitHub
Chat on Matrix	Contribute on GitHub whatwg/html repository
Commits on GitHub	Snapshots as of this commit
Twitter Updates @htmlstandard	Open Issues filed on GitHub
Open an issue whatwg.org/newbug	Tests web-platform-tests/html
Issues for Tests ongoing work	

Table of contents

- 1 Introduction.....1
- 2 Common infrastructure.....2
- 3 Semantics, structure, and APIs of HTML documents.....3
- 4 The elements of HTML.....4
- 5 Microdata.....12
- 6 User interaction.....12
- 7 Loading web pages.....13
- 8 Web application APIs.....16
- 9 Communication.....17
- 10 Web workers.....18
- 11 Worklets.....19

<https://html.spec.whatwg.org/multipage/>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.46

46

Reti di Elaboratori	<h2 style="color: blue; margin: 0;">Storia di HTTP</h2>	
<p>Versioni HTTP:</p> <p>0.9: un semplicissimo protocollo client-server di <a href="#">sola richiesta</a> di risorse HTML, senza flessibilità né nella direzione, né nel formato delle risorse. Utilizzata nel primo prototipo WWW e nei primi server NCSA.</p> <p>1.0 (RFC 1945): il protocollo diventa generico e definisce la <a href="#">statelessness (apolidia)</a>, e definisce alcuni metodi anche per l'upload di dati. Utilizzato fino al 1998-99</p> <p>1.1 (RFC 2068, 2069 e poi 2616, 2617): la versione attuale di HTTP, specifica meglio i meccanismi di caching, permette multi-homing e connessioni persistenti.</p> <p>HTTP-NG doveva essere la naturale evoluzione di HTTP, ma il WG IETF fallì miseramente nel raggiungere l'obiettivo, e l'evoluzione del protocollo si fermò.</p> <p>2.0 (RFC 7540): ottimizzare il trasferimento delle informazioni tra server e browser, minore latenza, compressione degli header HTTP, multiplexing di richieste e risposte per ricevere più oggetti in un'unica sessione.</p>		
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL	2.47

47

Reti di Elaboratori	<h2 style="color: blue; margin: 0;">Il protocollo http</h2>	
<p><b>Client-server</b>          In HTTP esistono due ruoli specifici: il <i>client</i> attiva la connessione e richiede dei servizi. Il <i>server</i> accetta la connessione, nel caso identifica il richiedente, e risponde alla richiesta. Alla fine chiude la connessione.</p> <p><b>Protocollo generico</b>          HTTP è indipendente dal formato dati con cui vengono trasmesse le risorse. <a href="#">Può funzionare per documenti HTML come per binari, eseguibili, oggetti distribuiti o altre strutture dati più o meno complicate.</a></p> <p><b>Statelessness (apolidia)</b>  <a href="#">Il server non è tenuto a mantenere informazioni che persistano tra una connessione e la successiva sulla natura, identità e precedenti richieste di un client.</a> Il client è tenuto a ricreare da zero il contesto necessario al server per rispondere.</p>		
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL	2.48

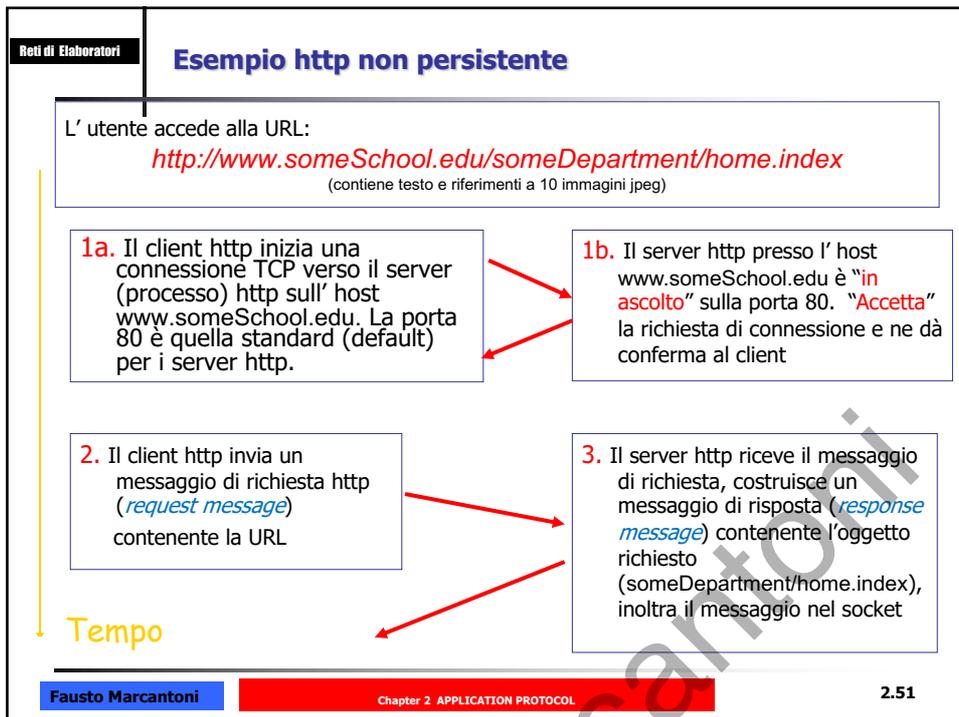
48

Reti di Elaboratori	<h2 style="margin: 0;">Il protocollo http (cont.)</h2>
<p><b>http: usa TCP:</b></p> <ul style="list-style-type: none"> <li>■ Il client inizia una connessione TCP (socket) verso il server sulla porta 80</li> <li>■ Il server accetta la connessione TCP dal client (sempre in ascolto sulla porta 80)</li> <li>■ Vengono scambiati messaggi http (messaggi del protocollo di livello applicativo) tra il browser (client http) e il Web server (server http)</li> <li>■ La connessione TCP è chiusa</li> </ul>	
<p><b>http è "stateless"</b></p> <ul style="list-style-type: none"> <li>■ Il server non mantiene informazione sulle richieste precedenti del client <i>Non confondere i file di log dell'applicativo</i></li> </ul>	
<p style="color: red;">I protocolli che mantengono informazione di stato sono complessi (es. TCP) !</p>	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL
2.49	

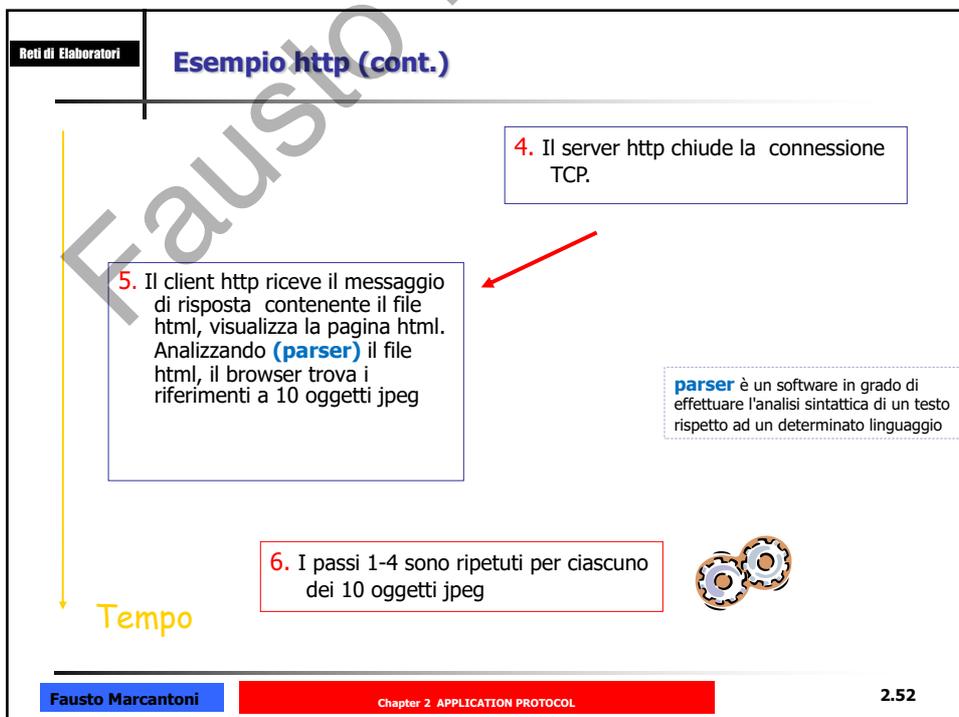
49

Reti di Elaboratori	<h2 style="margin: 0;">HTTP connections</h2>
<p style="text-align: center;"><b>Nonpersistent HTTP</b></p> <ul style="list-style-type: none"> <li>■ Al massimo un oggetto è trasmesso per ogni connessione TCP.</li> <li>■ HTTP/1.0 usa <b>nonpersistent HTTP</b></li> </ul>	
<p style="text-align: center;"><b>Persistent HTTP</b></p> <ul style="list-style-type: none"> <li>■ Oggetti multipli possono essere trasmessi in una singola connessione TCP</li> <li>■ HTTP/1.1 usa <b>persistent connections</b> in default mode</li> </ul>	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL
2.50	

50



51



52

**Reti di Elaboratori**

## Response time modeling

**Definizione di RTT (round trip time):**  
tempo occorrente ad un piccolo pacchetto per viaggiare dal client al server e ritornare al client

**Response time:**

- 1 RTT per iniziare la connessione TCP
- 1 RTT per HTTP request e pochi bytes per HTTP response di ritorno
- file transmission time

**total = 2RTT+transmit time**

```

[SEQ/ACK analysis]
[This is an ACK to the segment in frame: 18]
[The RTT to ACK the segment was: 0.020975000 seconds]
[rRTT: 0.021034000 seconds]

```

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL 2.53

53

**Reti di Elaboratori**

## Collegarsi al sito web → analizzare RTT

Aprire wireshark  
Selezionare l'interfaccia usata  
Collegarsi al sito web: <http://90.147.42.137>

```

[SEQ/ACK analysis]
[This is an ACK to the segment in frame: 4789]
[The RTT to ACK the segment was: 0.001186000 seconds]
[rRTT: 0.001244000 seconds]

```

ip.addr == 90.147.42.137 and not tcp.analysis.retransmission and not tcp.analysis.duplicate\_ack\_frame

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL 2.54

54

Reti di Elaboratori

## Response time modeling

Calcolo approssimato del tempo necessario per richiedere e ricevere un file HTML.

Diagram illustrating the response time modeling for an HTML file request. The process involves:

- Inizializzazione della connessione TCP (one RTT)
- Richiesta del file (one RTT)
- Ricezione dell'intero file (Tempo di trasmissione del file)

Time intervals are marked as 'Tempo presso il client' and 'Tempo presso il server'.

Fausto Marcontoni Chapter 2 APPLICATION PROTOCOL 2.55

55

Reti di Elaboratori

## Persistent HTTP

**Nonpersistent HTTP:**

- richiede 2 RTTs per object
- OS deve lavorare per allocare i buffer del TCP e le variabili del TCP devono essere conservate sia nel client che nel server
- In questo caso possono essere aperte connessioni TCP in parallelo

**Persistent HTTP:**

- server lascia aperta la connessione TCP dopo aver spedito la risposta
- i messaggi HTTP tra lo stesso client e server usano la medesima connessione

Diagram illustrating the difference between multiple connections and persistent connections:

- multiple connections:** Shows separate connections for each request, requiring multiple RTTs.
- persistent connection:** Shows a single connection that remains open, allowing multiple requests and responses to be sent over the same connection.

Fausto Marcontoni Chapter 2 APPLICATION PROTOCOL 2.56

56

Reti di Elaboratori

## Persistent HTTP

**Persistent without pipelining (senza parallelismo):**

- Il client emette una nuova richiesta solo quando la risposta precedente è stata ricevuta
- richiede 1 RTT per ciascun oggetto referenziato

**Persistent with pipelining (con parallelismo):**

- default in HTTP/1.1
- Il client emette una richiesta non appena incontra un riferimento
- richiede 1 RTT per tutti gli oggetti referenziati

no pipelining      pipelining

client      server      client      server

open      open      open      open

close      close      close      close

time

Fausto Marcantoni      Chapter 2 APPLICATION PROTOCOL      2.57

57

Reti di Elaboratori

## Nel nostro esempio : una pagina + 10 oggetti

Esercizio: **una pagina + 10 oggetti**

Calcolare il "response time" per le diverse tipologie di connessioni

- HTTP non persistente
  - $2RTT + 20RTT$
- HTTP persistente senza parallelismo
  - $2RTT + 10RTT$
- HTTP persistente con parallelismo
  - $2RTT + 1RTT$

Fausto Marcantoni      Chapter 2 APPLICATION PROTOCOL      2.58

58

Reti di Elaboratori

## Formato dei messaggi http

- Due tipi di messaggi http: *request*, *response*
- Messaggio http request:
  - ASCII (formato testo leggibile)

Request line (GET, POST, HEAD commands) → GET /somedir/page.html HTTP/1.1

header lines → Host: www.someschool.edu  
 Connection: close  
 User-agent: Mozilla/4.0  
 Accept: text/html, image/gif, image/jpeg  
 Accept-language: fr

Chiudi la connessione al termine della richiesta → Connection: close

Carriage return, line feed indica fine messaggio → (extra carriage return, line feed)

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.59

59

Reti di Elaboratori

## Message http request : formato generale

Riga di richiesta → metodo sp URL sp versione cr lf

Righe di intestazione → nome del campo di intestazione: sp valore cr lf

Riga vuota → cr lf

Corpo dell'entità

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.60

60

Reti di Elaboratori	<h2>Message http request : formato generale</h2>	
<ul style="list-style-type: none"> <li>• <b>HEAD</b> <ul style="list-style-type: none"> <li>- Simile a GET, ma richiede solo le informazioni dell'header.</li> <li>- Utile per controllare se un documento esiste e quanto è recente.</li> </ul> </li> <li>• <b>POST</b> <ul style="list-style-type: none"> <li>- Simile a GET, ma codifica gli input in modo diverso.</li> <li>- Utile per proporre i contenuti dei form a un programma CGI. <small>CGI: Common Gateway Interface</small></li> </ul> </li> <li>• <b>PUT</b> <ul style="list-style-type: none"> <li>- Trasferisce un documento al server.</li> <li>- Presente a partire dalla versione HTTP/1.1.</li> </ul> </li> <li>• <b>DELETE</b> <ul style="list-style-type: none"> <li>- Elimina un documento dal server.</li> <li>- Presente a partire dalla versione HTTP/1.1.</li> </ul> </li> </ul>		
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL	2.61

61

Reti di Elaboratori	<h2>Message http request : formato generale</h2>	
<ul style="list-style-type: none"> <li>• <b>TRACE</b> <ul style="list-style-type: none"> <li>- Attraverso questo metodo un client può vedere cosa viene ricevuto dall'altra parte della catena delle richieste in modo tale da eseguire delle diagnosi</li> <li>- La richiesta TRACE non deve includere un corpo</li> <li>- Se la richiesta è valida la risposta dovrebbe contenere una risposta 200 (OK) e l'intero messaggio di richiesta nel corpo dell'entità con Content-Type "message/http"</li> <li>- Le risposte non dovrebbero essere soggette a cache.</li> </ul> </li> <li>• <b>CONNECT</b> <ul style="list-style-type: none"> <li>- Usata con i proxy per tunnel (es. SSL)</li> </ul> </li> </ul>		
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL	2.62

62

Reti di Elaboratori

## The Method token

<https://www.w3.org/Protocols/rfc2616/rfc2616-sec5.html>  
<https://www.rfc-editor.org/rfc/rfc9110.html>

Method Name	Description	Section
GET	Transfer a current representation of the target resource.	9.3.1
HEAD	Same as GET, but do not transfer the response content.	9.3.2
POST	Perform resource-specific processing on the request content.	9.3.3
PUT	Replace all current representations of the target resource with the request content.	9.3.4
DELETE	Remove all current representations of the target resource.	9.3.5
CONNECT	Establish a tunnel to the server identified by the target resource.	9.3.6
OPTIONS	Describe the communication options for the target resource.	9.3.7
TRACE	Perform a message loop-back test along the path to the target resource.	9.3.8

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.63

63

Reti di Elaboratori

## Formato del messaggio http response

protocol - status code - status phrase

status line → HTTP/1.1 200 OK

header lines →

```

Connection: close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 .....
Content-Length: 6821
Content-Type: text/html

```

data, e.g., requested html file → data data data data data ...

Client HTTP 1.0: Server chiude connessione al termine della richiesta  
 Client HTTP 1.1: mantiene aperta la connessione oppure chiude se Connection: close

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.64

64

Reti di Elaboratori

## Message http response : formato generale

Riga di stato — versione sp codice di stato frase cr lf

Righe di intestazione — nome del campo di intestazione: sp valore cr lf

Riga vuota — cr lf

Corpo dell'entità

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.65

65

Reti di Elaboratori

## Hypertext Transfer Protocol (HTTP) Field Name Registry

iana  
Internet Assigned Numbers Authority

### Hypertext Transfer Protocol (HTTP) Field Name Registry

**Created**  
2021-10-01

**Last Updated**  
2022-08-25

**Available Formats**

XML HTML Plain text

**Registry included below**

- [Hypertext Transfer Protocol \(HTTP\) Field Name Registry](https://www.iana.org/assignments/http-fields/http-fields.xhtml)

<https://www.iana.org/assignments/http-fields/http-fields.xhtml>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.66

66

Reti di Elaboratori

## Risposta: codici di stato

Prima riga del messaggio di risposta server->client.  
Alcuni esempi:

**200 OK**

- Successo, oggetto richiesto più avanti nel messaggio

**301 Moved Permanently**

- L'oggetto richiesto è stato spostato. Il nuovo indirizzo è specificato più avanti (Location:)

**400 Bad Request**

- Richiesta incomprensibile al server

**404 Not Found**

- Il documento non è stato trovato sul server

**505 HTTP Version Not Supported**

- Il server non dispone della versione del protocollo

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.67

67

Reti di Elaboratori

## Risposta: codici di stato

### 15. Status Codes

The status code of a response is a three-digit integer code that describes the result of the request and the semantics of the response, including whether the request was successful and what content is enclosed (if any). All valid status codes are within the range of 100 to 599, inclusive.

The first digit of the status code defines the class of response. The last two digits do not have any categorization role. There are five values for the first digit:

- **1xx (Informational):** The request was received, continuing process
- **2xx (Successful):** The request was successfully received, understood, and accepted
- **3xx (Redirection):** Further action needs to be taken in order to complete the request
- **4xx (Client Error):** The request contains bad syntax or cannot be fulfilled
- **5xx (Server Error):** The server failed to fulfill an apparently valid request

<https://www.rfc-editor.org/rfc/rfc9110.html#name-status-codes>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.68

68

**Reti di Elaboratori**

## Risposta: codici di stato

```

Status-Code =
| "100" ; Section 10.1.1: Continue
| "101" ; Section 10.1.2: Switching Protocols
| "200" ; Section 10.2.1: OK
| "201" ; Section 10.2.2: Created
| "202" ; Section 10.2.3: Accepted
| "203" ; Section 10.2.4: Non-Authoritative Information
| "204" ; Section 10.2.5: No Content
| "205" ; Section 10.2.6: Reset Content
| "206" ; Section 10.2.7: Partial Content
| "300" ; Section 10.3.1: Multiple Choices
| "301" ; Section 10.3.2: Moved Permanently
| "302" ; Section 10.3.3: Found
| "303" ; Section 10.3.4: See Other
| "304" ; Section 10.3.5: Not Modified
| "305" ; Section 10.3.6: Use Proxy
| "307" ; Section 10.3.8: Temporary Redirect
| "400" ; Section 10.4.1: Bad Request
| "401" ; Section 10.4.2: Unauthorized
| "402" ; Section 10.4.3: Payment Required
| "403" ; Section 10.4.4: Forbidden
| "404" ; Section 10.4.5: Not Found
| "405" ; Section 10.4.6: Method Not Allowed
| "406" ; Section 10.4.7: Not Acceptable
| "407" ; Section 10.4.8: Proxy Authentication Requ
| "408" ; Section 10.4.9: Request Time-out
| "409" ; Section 10.4.10: Conflict
| "410" ; Section 10.4.11: Gone
| "411" ; Section 10.4.12: Length Required
| "412" ; Section 10.4.13: Precondition Failed
| "413" ; Section 10.4.14: Request Entity Too Large
| "414" ; Section 10.4.15: Request-URI Too Large
| "415" ; Section 10.4.16: Unsupported Media Type
| "416" ; Section 10.4.17: Requested range not sati
| "417" ; Section 10.4.18: Expectation Failed
| "500" ; Section 10.5.1: Internal Server Error
| "501" ; Section 10.5.2: Not Implemented
| "502" ; Section 10.5.3: Bad Gateway
| "503" ; Section 10.5.4: Service Unavailable
| "504" ; Section 10.5.5: Gateway Time-out
| "505" ; Section 10.5.6: HTTP Version not supporte
| extension-code

```

<https://www.rfc-editor.org/rfc/rfc9110.html#name-status-codes>

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL 2.69

69

**Reti di Elaboratori**

## Cos'è "telnet"

- Telnet è un'applicazione standard di Internet ed è disponibile nella maggior parte delle implementazioni del TCP/IP, indipendentemente dal sistema operativo host.
- Si tratta di un semplice protocollo di **login remoto**, implementato secondo un modello di tipo client-server, *che permette ad un utente attestato ad una certa macchina di stabilire una connessione TCP con un server di login che si trova su un'altra macchina.*
- La porta che utilizza è la numero 23

The diagram illustrates the Telnet protocol interaction. On the left, a 'Telnet client' is represented by a computer icon. On the right, a 'Telnet server (telnetd)' is represented by a server rack icon. A line labeled 'telnet protocol' connects the two. Below this line, a speech bubble shows the exchange of 'username' and 'password' credentials, with a small figure of a person representing the user.

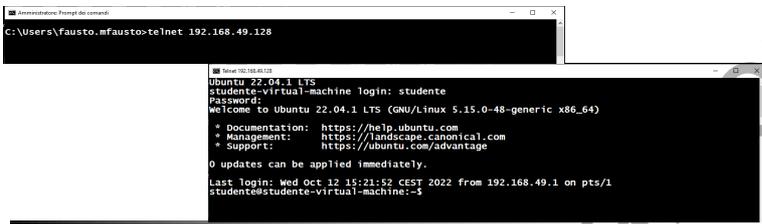
**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL 2.70

70

**Reti di Elaboratori**

## Cos'è "telnet"

- Telnet (il client) rilancia i caratteri battuti sulla tastiera dell'utente direttamente al **calcolatore remoto** come se essi fossero battuti su una tastiera direttamente connessa ad esso. Telnet (il server: telnetd) **rimanda l'output della macchina remota indietro fino allo schermo dell'utente.**
- Il servizio è definito **trasparente** perché dà l'apparenza che la tastiera e lo schermo dell'utente siano attaccati direttamente alla macchina remota.
- Telnet non molto sofisticato, ma **era** largamente diffuso.
- Il codice client di Telnet permette all'utente di specificare la macchina remota a cui ci si vuole connettere dando il suo nome di dominio oppure il suo indirizzo IP e specificando la porta

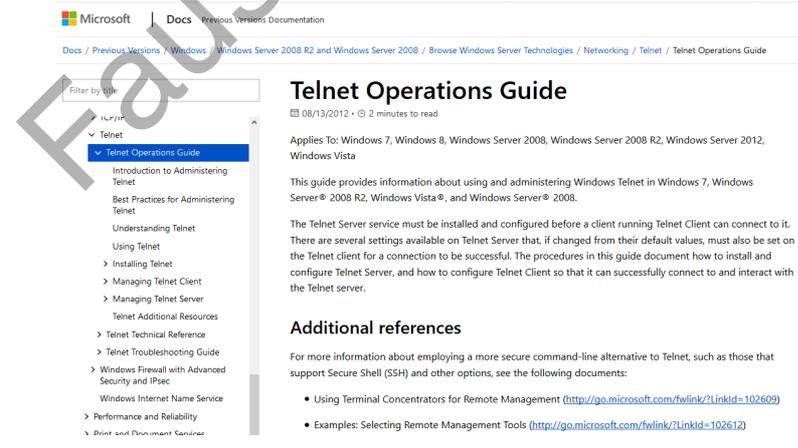


**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL **2.71**

71

**Reti di Elaboratori**

## Telnet per Windows



**Telnet Operations Guide**

08/13/2012 • 2 minutes to read

Applies To: Windows 7, Windows 8, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Vista

This guide provides information about using and administering Windows Telnet in Windows 7, Windows Server® 2008 R2, Windows Vista®, and Windows Server® 2008.

The Telnet Server service must be installed and configured before a client running Telnet Client can connect to it. There are several settings available on Telnet Server that, if changed from their default values, must also be set on the Telnet client for a connection to be successful. The procedures in this guide document how to install and configure Telnet Server, and how to configure Telnet Client so that it can successfully connect to and interact with the Telnet server.

**Additional references**

For more information about employing a more secure command-line alternative to Telnet, such as those that support Secure Shell (SSH) and other options, see the following documents:

- Using Terminal Concentrators for Remote Management (<http://go.microsoft.com/fwlink/?Linkid=102609>)
- Examples: Selecting Remote Management Tools (<http://go.microsoft.com/fwlink/?Linkid=102612>)

[https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc753164\(v%3dws.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc753164(v%3dws.10))

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL **2.72**

72

Reti di Elaboratori

## telnet - wireshark

No.	Time	Source	Destination	Protocol	Length	Info
10	0.020452	192.168.49.128	192.168.49.1	TELNET	66	Telnet Data ...
11	0.020676	192.168.49.1	192.168.49.128	TELNET	60	Telnet Data ...
14	0.029079	192.168.49.128	192.168.49.1	TELNET	57	Telnet Data ...
15	0.029241	192.168.49.1	192.168.49.128	TELNET	63	Telnet Data ...
18	0.029638	192.168.49.128	192.168.49.1	TELNET	66	Telnet Data ...
19	0.029848	192.168.49.1	192.168.49.128	TELNET	63	Telnet Data ...
22	0.030173	192.168.49.1	192.168.49.128	TELNET	70	Telnet Data ...
25	0.030947	192.168.49.128	192.168.49.1	TELNET	66	Telnet Data ...
26	0.031876	192.168.49.1	192.168.49.128	TELNET	57	Telnet Data ...
29	0.031445	192.168.49.1	192.168.49.128	TELNET	63	Telnet Data ...
32	0.031815	192.168.49.128	192.168.49.1	TELNET	60	Telnet Data ...

Frame 115: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface \Device\NPF\_{B411EEE2-1A51-4070-915B-D4511FA1E878}, id 0  
 Ethernet II, Src: VMware\_c0:00:00 (00:50:56:c0:00:00), Dst: VMware\_D2:47:ac (00:8c:29:62:47:ac)  
 Internet Protocol Version 4, Src: 192.168.49.1 (192.168.49.1), Dst: 192.168.49.128 (192.168.49.128)  
 Transmission Control Protocol, Src Port: 5813, Dst Port: 23, Seq: 75, Ack: 118, Len: 1  
 Telnet  
 Data: d

Telnet: Protocol Pacchetti: 129 - visualizzati: 45 (34.9%) - scartati: 0 (0.0%) Profilo: Default

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.73

73

Reti di Elaboratori

## telnet - wireshark

```

.....#.....].....ANSI.....Ubuntu 22.04.1 LTS
...studente-virtual-machine login: ssttuuddeenttee
Password: password
Login incorrect
studente-virtual-machine login:

```

25 pacchetti client 20 pacchetti server 32 turni

Conversazione intera (247 bytes) Mostra dati come ASCII Flusso 0

Trova: Trova successivo

Filtra questo flusso Stampa Salva come... Indietro Chiudi Aiuto

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.74

Verificare la password in chiaro

74

Reti di Elaboratori

## telnet - wireshark

Notare la lunghezza di 1 byte nel pacchetto della digitazione di username e password

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL **2.75**

75

Reti di Elaboratori

## Prova (client)

1. Telnet verso un Web server:
 

```
telnet <<mio indirizzo IP>> 80
```

Apre connessione TCP verso la porta 80 (default)
2. Si digita una richiesta http GET:
 

```
GET
```

Digitando ciò (carriage return due volte), si invia una richiesta GET al server http
3. Si osservi la risposta!

```
GET / HTTP/1.1
HOST: www.unicam.it
USER-AGENT: Fausto
```

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL **2.76**

76

Reti di Elaboratori

## Prova (client) - risposta

```

Amministratore: Prompt dei comandi

HTTP/1.1 200 OK
Date: Wed, 29 Sep 2021 08:12:12 GMT
Server: Apache/2.4.48 (Debian)
Last-Modified: Wed, 29 Sep 2021 07:33:02 GMT
ETag: "62-5cd1d57139c87"
Accept-Ranges: bytes
Content-Length: 98
Vary: Accept-Encoding
Connection: close
Content-Type: text/html

<html>
  <head>
    <title>Ciao Page: It works</title>
  </head>
  <body>
    Ciao
  </body>
</html>

Connessione all'host perduta.
C:\Users\fausto.mfausto>

```

Commentare la risposta ottenuta

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.77

77

Reti di Elaboratori

## User Agent HTTP

User-Agent: <product> / <product-version> <comment>  
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent>

- ✓ [Android User Agents](#)
- ✓ [iPhone User Agents](#)
- ✓ [MS Windows User Agents](#)
- ✓ [Tablet User Agents](#)
- ✓ [Desktop User Agents](#)
- ✓ [Set Top Box User Agents](#)
- ✓ [Games Console User Agents](#)
- ✓ [Bots and Crawlers User Agents](#)
- ✓ [E-Readers User Agents](#)

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.78

78

**Reti di Elaboratori**

## User agent generator

Google User agent generator

Circa 35.500.000 risultati (0,46 secondi)

<https://user-agents.net/random> Traduci questa pagina  
**Random User Agent**  
 A simple webpage to generate and get a random user agent list. Please note that at this moment you can generate up to 1000 random user agents

<https://generate-name.net/user-agent> Traduci questa pagina  
**Generate User Agent Online**  
 Generate Random User Agent online. Filter, sort and generate randomly.

<https://useragents.io/random> Traduci questa pagina  
**Random User Agent - UserAgents.io**  
 Generate a list of randomized user agents. Generate in fast way random user agent list. Up to 1,500 user agents can be generated at one time.

<https://generatefakename.com/user-agent-generator> Traduci questa pagina  
**Random Useragent Generator**  
 Test your software and social media. Generate random names, last names, domains, business names, addresses, passwords, mac addresses, email addresses.

[https://www.google.com/search?client=firefox-b-d&q=user-agent-generator&cs=X&ved=2ahUKEwI4wa3p2d64hVDI\\_OHHYBI8501Oj6BAoQE958biw=14738bih=886&dur=1](https://www.google.com/search?client=firefox-b-d&q=user-agent-generator&cs=X&ved=2ahUKEwI4wa3p2d64hVDI_OHHYBI8501Oj6BAoQE958biw=14738bih=886&dur=1)

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL 2.79

79

**Reti di Elaboratori**

## http matches "GET"

No.	Time	Source	Destination	Protocol	Length	Info
11710	43.266754	mfausto.local	www.unina.it	HTTP	521	GET / HTTP/1.1
11722	43.284282	mfausto.local	www.unina.it	HTTP	702	GET /home/jsessionId-D68C3328
11847	43.540722	mfausto.local	www.unina.it	HTTP	732	GET /html/css/main.css?browse
11852	43.544342	mfausto.local	www.unina.it	HTTP	756	GET /html/portlet/asset_public
11866	43.545027	mfausto.local	www.unina.it	HTTP	756	GET /html/portlet/journal_com
11868	43.545164	mfausto.local	www.unina.it	HTTP	748	GET /unina-homepage-theme/css

```

> GET / HTTP/1.1\r\n
  Host: www.unina.it\r\n
  Connection: keep-alive\r\n
  Upgrade-Insecure-Requests: 1\r\n
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36 Edg/106.0.1370.35
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: it,it;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6\r\n
  \r\n
  [Full request URI: http://www.unina.it/]
  [HTTP request 1/1]
  [Response in frame: 11716]
  
```

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL 2.80

80

Reti di Elaboratori

## Prova (client) - risposta

```
telnet <<mio indirizzo IP>> 80
```

- digitare e commentare:
  - abcdef
  - GET /index.html HTTP/1.0 (?)
  - HEAD
  - HEAD /index.html HTTP/1.0 (?)
  - POST
  - GET /index.html HTTP/1.1 (?)

Esercitazione con



internet reti sicurezza

Esercitazioni

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.81

81

Reti di Elaboratori

## I cookies

- **HTTP è stateless**: non esiste nessuna struttura ulteriore alla connessione, e il server non è tenuto a mantenere informazioni su connessioni precedenti.
- Un cookie (non è del protocollo HTTP, è un'estensione di Netscape, proposta nell'RFC 2109 e poi ancora RFC 2965 e infine in RFC 6265) è **una breve informazione scambiata tra il server ed il client**.
- Il **client mantiene lo stato di precedenti connessioni**, e lo manda al server di pertinenza ogni volta che richiede un documento.
- Il termine *cookie* (anche *magic cookie*) in informatica indica un **blocco di dati opaco** (i.e.: non interpretabile) lasciato in consegna ad un richiedente per poter ristabilire in seguito il suo diritto alla risorsa richiesta (come il tagliando di una lavanderia)

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.82

82

Reti di Elaboratori

## HTTP State Management Mechanism

---

HTTP State Management Mechanism

**Abstract**

This document defines the HTTP Cookie and Set-Cookie header fields. These header fields can be used by HTTP servers to store state (called cookies) at HTTP user agents, letting the servers maintain a stateful session over the mostly stateless HTTP protocol. Although cookies have many historical infelicities that degrade their security and privacy, the Cookie and Set-Cookie header fields are widely used on the Internet. This document obsoletes RFC 2965.

<https://www.rfc-editor.org/rfc/rfc6265>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.83

83

Reti di Elaboratori

## Cookie [RFC2109] obsoleto

4 componenti

1. Una riga di intestazione nel messaggio di risposta HTTP
2. Una riga di intestazione nel messaggio di richiesta HTTP
3. Un file cookie memorizzato sul sistema client dell'utenet e gestito dal browser
4. Un database sul sito (server)

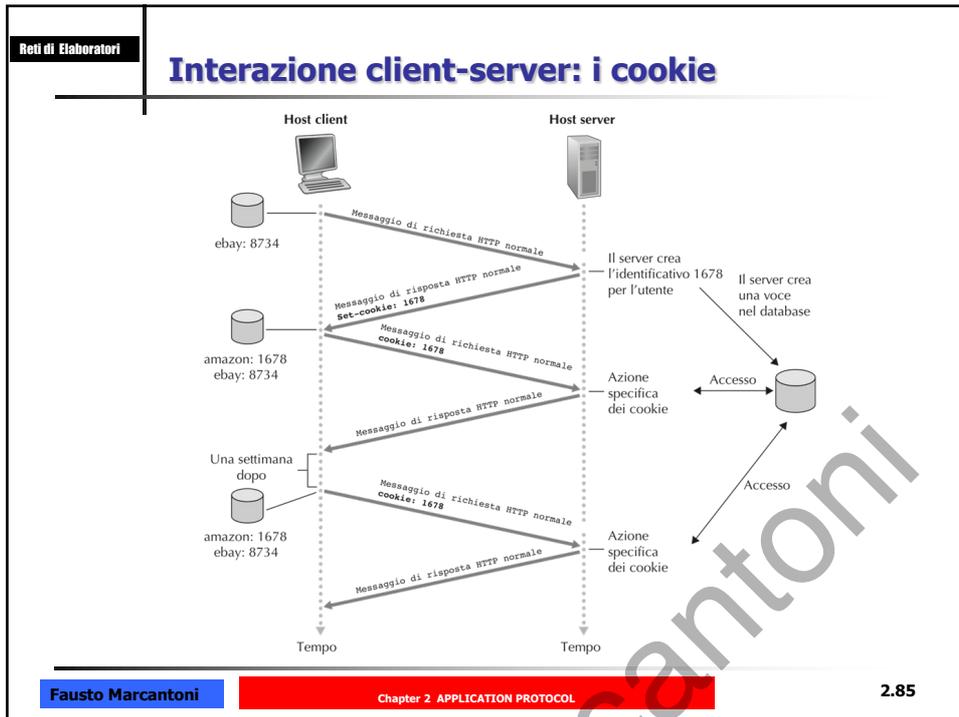
esempio

```

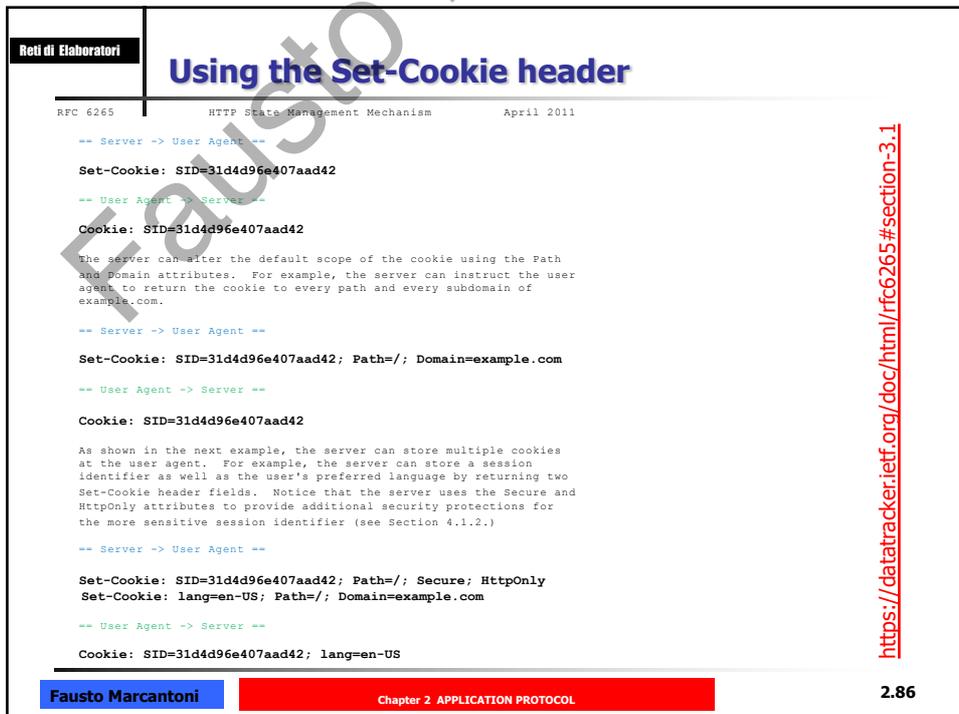
PI
P=0403710006620D11050774610068656A161D647E721804797E7A027808557F7F7C6A0205
punto-informatico.it/
1536
3164909568
29887167
1043606064
29813841
*
```

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.84

84



85



86

Reti di Elaboratori

## Cookie [RFC2109]

- Il server invia un "cookie" al client con la richiesta  
`Set-cookie: 1678`
- Il client presenta il cookie in accessi successivi  
`cookie: 1678`
- Il server controlla il cookie presentato
  - Autenticazione
  - Traccia delle preferenze dell'utente
  - Profilazione utente

client
server

usual http request msg →

← usual http response +  
`Set-cookie: #`

usual http request msg  
`cookie: #` → cookie-specific action

← usual http response msg

---

usual http request msg  
`cookie: #` → cookie-specific action

← usual http response msg

Fausto Marcantoni
Chapter 2 APPLICATION PROTOCOL
2.87

87

Reti di Elaboratori

## Come funzionano i Cookie

- Un cookie, dall'inglese "biscotto", è un piccolo file di testo che viene ricevuto dal browser durante la navigazione e viene memorizzato in un opportuno
- **repository** →  
C:\Users\fausto.mfausto\AppData\Local\Microsoft\Windows\INETCookies  
 Windows key and R together. Type *shell:cookies* and click OK.  
<http://windows.microsoft.com/it-it/windows/view-temporary-internet-files#1TC=windows-7>  
<https://www.thewindowsclub.com/cookies-folder-location-windows>

MVP | Moderatore Volontario

win+r  
Digita:  
shell:cookies  
e dai invio.

Microsoft® MVP 10th Year - Windows and Devices for IT - System Administration - Hardware/Software.  
If it ain't broke, don't fix it!  
(-\_-) Click su SI se il problema è risolto.

Risposta il marzo 21, 2018 ~

Fausto Marcantoni
Chapter 2 APPLICATION PROTOCOL
2.88

88

**Reti di Elaboratori**

## Visualizzare i cookies

**Google Chrome** stores all cookies in a single file called *Cookies*.  
The file is located at the following path:  
**C:\Users\Your User Name\AppData\Local\Google\Chrome\User Data\Default**  
<C:\Users\fausto.mfausto\AppData\Local\Google\Chrome\User Data\Default>

**Mozilla Firefox** stores all the cookies, from all the websites that you visit, in a single file called *cookies.sqlite*.  
You can find it in your Firefox profile folder, at the following path:  
**C:\Users\Your User Name\AppData\Roaming\Mozilla\Firefox\Profiles**  
<C:\Users\fausto.mfausto\AppData\Roaming\Mozilla\Firefox\Profiles>

**Opera** keeps all the cookies in a single file called *Cookies*, just like *Google Chrome*.  
The *Cookies* file is located at the following path:  
**C:\Users\Your User Name\AppData\Roaming\Opera Software\Opera Stable**  
If you are running a Windows 10 version before 1709 Fall Creators Update,  
you can find the cookies files created by **Internet Explorer** at this path:  
**C:\Users\Your User Name\AppData\Local\Microsoft\Windows\INetCookies**

In the latest versions of Windows 10, **Microsoft Edge** does not have a specific cookies file.  
If you have an older version of Windows 10, before 1709 Fall Creators Update, you might be able to find one or more cookies files in these locations:  
"C:\Users\Your User Name\AppData\Local\Packages\Microsoft.MicrosoftEdge\_8wekyb3d8bbwe\AC\MicrosoftEdge\Cookies"  
"C:\Users\Your User Name\AppData\Local\Packages\Microsoft.MicrosoftEdge\_8wekyb3d8bbwe\AC\#1001\MicrosoftEdge\Cookies"  
"C:\Users\Your User Name\AppData\Local\Packages\Microsoft.MicrosoftEdge\_8wekyb3d8bbwe\AC\#1002\MicrosoftEdge\Cookies"

<https://www.digitalcitizen.life/cookies-location-windows-10>

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL 2.89

89

**Reti di Elaboratori**

## Come funzionano i Cookie

- Non tutti i siti rilasciano cookies.
- **Facoltativo e prettamente applicativo.**
- Poterlo rileggere in un secondo tempo (comportamento applicativo di una certa complessità).
- Reale **pericolo per la privacy**

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL 2.90

90

Reti di Elaboratori

## Come funzionano i Cookie

- E' possibile rendere **facoltativa l'accettazione** dei cookie da parte dei browser web.
- All'interno di un cookie sono memorizzate **informazioni ad uso e consumo del server** web che ha chiesto al browser la memorizzazione del cookie stesso.
- La struttura di un cookie è assimilabile ad una **struttura XML**

```
<COOKIE
  NOME="name"
  VALORE="text"
  SCADENZA="period"
  SICURO="Yes/No"
  PERCORSO="urls"
  DOMINIO=".domain.com" >
```

- Notate che quasi tutte le informazioni siano state **memorizzate "criptate"**, a maggior tutela della privacy dell'utente che ha memorizzato il cookie.
- Spesso il formato SQLite

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.91

91

Reti di Elaboratori

## Set-Cookie Syntax

RFC 6265 HTTP State Management Mechanism April 2011

```

set-cookie-header = "Set-Cookie:" SP set-cookie-string
set-cookie-string = cookie-pair *( ";" SP cookie-av )
cookie-pair       = cookie-name "=" cookie-value
cookie-name       = token
cookie-value      = "cookie-octet / ( DQUOTE "cookie-octet DQUOTE )
cookie-octet      = %x21 / %x23-2B / %x2D-3A / %x3C-5B / %x5D-7E
                  ; US-ASCII characters excluding CTLs,
                  ; whitespace DQUOTE, comma, semicolon,
                  ; and backslash
token             = <token, defined in [RFC2616], Section 2.2>

cookie-av         = expires-av / max-age-av / domain-av /
                  path-av / secure-av / httponly-av /
                  extension-av
expires-av       = "Expires=" sane-cookie-date
sane-cookie-date = <rfc1123-date, defined in [RFC2616], Section 3.3.1>
max-age-av       = "Max-Age=" non-zero-digit *DIGIT
                  ; In practice, both expires-av and max-age-av
                  ; are limited to dates representable by the
                  ; user agent.
non-zero-digit   = %x31-39
                  ; digits 1 through 9
domain-av        = "Domain=" domain-value
domain-value     = <subdomain>
                  ; defined in [RFC1034], Section 3.5, as
                  ; enhanced by [RFC1123], Section 2.1
path-av          = "Path=" path-value
path-value       = <any CHAR except CTLs or ">;>
secure-av        = "Secure"
httponly-av      = "HttpOnly"
extension-av     = <any CHAR except CTLs or ">;>

```

<https://www.rfc-editor.org/rfc/rfc6265.html#page-13>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.92

92

Reti di Elaboratori

## Esempio – non funziona

C:\Users\fausto.mfausto\AppData\Local\Microsoft\Windows\INetCache

cookiefausto@google.it/

cookiefausto@microsoft.com/

cookiefausto@ieom/

cookiefausto@microsoft.com/

cookiefausto@n/

cookiefausto@seving-sys.com/

cookiefausto@tr

BVFUWLZZ.txt

```

1 SRCHD
2 AF=NOFORM
3 microsoft.com/
4 1024
5 2221714048
6 30622912
7 1778382550
8 30475860
9 *
10 SRCHUSR
11 DOB=20151014
12 microsoft.com/
13 1024
14 2221714048
15 30622912
16 1778382550
17 30475860
18 *
19

```

Contenuto

Copiato sul Desktop

BVFUWLZZ.txt

**ARCHEOLOGIA**

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.93

93

Reti di Elaboratori

## Firefox extension

Per chi fa le pagine web



Firefox Browser ADD-ONS Extensions Themes More... v

Cookie-Editor by Moustachauve

Firefox Browser ADD-ONS Extensions Themes More... v

Cookie Manager by Rob W

<https://addons.mozilla.org/en-US/firefox/search/?q=cookie>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.94

94

Reti di Elaboratori	<h2 style="color: blue; margin: 0;">Come funzionano i Cookie</h2>	
<ul style="list-style-type: none"> <li>• Un cookie è un file di testo. Non può fare nulla se non <b>trasportare informazioni</b>.</li> <li>• Il server, tendenzialmente, utilizzerà il cookie come collegamento tra il client (e quindi l'utilizzatore a tutti gli effetti) ed <b>un profilo personalizzato</b> che si sarà creato.</li> <li>• Ad ogni altro accesso il server sarà in grado di "riconoscerci" utilizzando il cookie come <i>chiave</i> per individuare il <i>profilo memorizzato</i> sul server stesso e sarà in grado, di arricchire il profilo <b>raccogliendo informazioni sulla nostra navigazione</b> all'interno del sito stesso.</li> <li>• Perché si utilizzano i cookies?             <ul style="list-style-type: none"> <li>• <b>profilo personalizzato</b> all'interno dell'applicazione, ad esempio la lingua utilizzata e le impostazioni di visualizzazione.</li> <li>• <b>autenticazione</b>, ad esempio in un'applicazione di webmail, non sarà necessario inserire username e password ogni volta che si cerca di accedere al servizio. (sicurezza=0)</li> </ul> </li> </ul>		
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL	2.95

95

Reti di Elaboratori	<h2 style="color: blue; margin: 0;">I third party cookie</h2>	
<ul style="list-style-type: none"> <li>➤ Un uso subdolo (ma alcuni lo giustificano) dei cookie è l'inserimento di <b>cookie nei banner e nelle pubblicità</b>.</li> <li>➤ Questo permette ad un'agenzia di banner di seguire la navigazione di un utente attraverso tutti i siti a cui fornisce banner, e quindi fornire una <b>profilazione</b> più precisa del navigatore, con effetti discutibili sulla sua privacy.</li> <li>➤ <b>RFC 2965 esplicitamente proibisce questo tipo di comportamento</b>, che è però largamente ignorato dai produttori di browser e dai fornitori di banner.</li> <li>➤ Si aggiunga che molte versioni di browser hanno bug che permettono a codice Javascript malevolo, nascosto dentro alle pagine dei browser, di <b>"sniffare" i contenuti dei cookie</b> destinati ad altri domini.</li> </ul>		
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL	2.96

96

## pubblicità comportamentale

la **pubblicità comportamentale** è una pratica che si basa sull'attività di navigazione in rete e permette ai brand di inviare messaggi pubblicitari agli utenti della rete in funzione dei loro interessi.



**Your Online Choices**  
a guide to online behavioural advertising

<https://www.youronlinechoices.com/it/>

Benvenuti nella guida sulla **pubblicità comportamentale** e la privacy online. In questo sito web troverete informazioni su come funziona la pubblicità comportamentale e molte informazioni **sui cookie** oltre alle tappe da seguire per proteggere la privacy su internet.

<https://www.youronlinechoices.com/it/le-tue-scelte>

## Esercitazione: cookie

- Utilizzando lo sniffer
  - Aprire una sessione http
  - Verificare che la sessione "contenga" un cookie
  - Verificare lo scambio di messaggi tra client e server
- Trovare il cookie
- Visualizzare il cookie

```

Stream Content
GET /search?hl=it&q=al+volante&btnq=Cerca+con+Google&meta= HTTP/1.1
Accept: image/gif, image/x-bitmap, image/jpeg, image/png, application/x-ms-application, application/vnd.ms-xpsdocument, application/xaml+xml, application/x-ms-xbap, application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, application/x-silverlight, */*
Referer: http://www.google.it/
Accept-Language: it
UA-CPU: x86
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; Media Center PC 5.0; .NET CLR 3.0.04506; InfoPath.2)
Host: www.google.it
Connection: Keep-Alive
Cookie: PREF=ID=8ad12785033b16b5:TM=1225122682:LM=1225122682:S=ghhNTdcby9_y2iUM;
NID=16-zt8j3k1-jx8SAZdeAcFNetgR6l1_d99AAdm2nkvjv38sv249kn30fjv8G5QvPj4ovahozlyN--
gv81wRISxAeV2p1gdf9C2B4ZILrPchrRXoYMYFCFwIbZ18G54

HTTP/1.1 200 OK
Cache-Control: private, max-age=0
Date: Mon, 03 Nov 2008 11:16:19 GMT
Expires: -1
Content-Type: text/html; charset=UTF-8
Set-Cookie: SS=Q0=Ywgdm9sYw50ZQ; path=/search
Server: gws
Transfer-Encoding: chunked
Content-Encoding: gzip
  
```

**Reti di Elaboratori**

## Filtro http

First some theoretical background:

- **Cookies:** Cookies are a way for a HTTP server to store information on the client, which will be presented back to the server in following requests. The purpose is to have a way of maintaining information between the client and the server to simulate a session (http by itself has no notion of sessions, it's just a way to exchange objects).
- **Query Strings:** A query string is a way for the client to submit (dynamic) data to the server. Mostly this is done by having a FORM on a webpage that can be filled in and when it's submitted the filled in values are transferred to the server in the "Query String". This can be done with the GET method, in which the query string will be added to the requested URL. Or it can be done with the POST method in which the query string will be sent as HTTP data, after the HTTP headers.

Both Cookies and Query Strings are completely independent of each other, but are widely used together. The way they are used depends on the way the web application has been written.

To filter all requests that contain a cookie, use:

```
http.cookie
http.cookie contains <cookieName>
```

To filter for query strings:

```
http.request.uri contains "?" or http.request.method="POST"
```

This of course only works with HTTP as HTTPS traffic is encrypted. However, if you do have access to the private key used on the HTTPS server, you are able to decrypt the HTTPS traffic which makes the HTTP traffic inside the HTTPS traffic visible.

<https://osqa-ask.wireshark.org/questions/751/cookie-and-query-strings/>

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL 2.99

99

**Reti di Elaboratori**

## Filtro per cookie in TLS

tls.handshake.extensions.cookie	Cookie	Byte sequence
tls.handshake.extensions.cookie_len	Cookie length	Unsigned integer (16 bits)

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL 2.100

100

Reti di Elaboratori

https://sqlitebrowser.org

## DB Browser for SQLite

*The Official home of the DB Browser for SQLite*

<https://sqlitebrowser.org/dl/>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.101

101

Reti di Elaboratori



Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.102

102

Reti di Elaboratori	<h2 style="margin: 0;">Modelli di sicurezza</h2>
<p>Ci sono due modi per fornire un trasporto sicuro (non intercettabile):</p> <ul style="list-style-type: none"> <li>• <b>Usare un'infrastruttura di trasporto sicura</b> <ul style="list-style-type: none"> <li>➢ Il protocollo non cambia, ma ogni pacchetto trasmesso nello scambio di informazioni viene gestito in maniera sicura dal protocollo di trasporto</li> </ul> </li> <li>• <b>Usare un protocollo sicuro a livello applicazione</b> <ul style="list-style-type: none"> <li>➢ Si usa un protocollo anche diverso, che si occupa di gestire la trasmissione delle informazioni.</li> </ul> </li> </ul> <p>HTTPS (RFC 2818)          Introdotta da Netscape, trasmette i dati in HTTP semplice su un protocollo di trasporto (SSL) che crittografa tutti i pacchetti.          Il server ascolta su una porta diversa (per default la porta 443), e si usa uno schema di URI diverso (introdotta da <b>https://</b> )</p> <p>S-HTTP (RFC 2660)          Poco diffuso, incapsula richieste e risposte HTTP in un messaggio crittografato, o secondo un formato MIME apposito (MIME Object Security Services, MOSS), o un formato terzo (Cryptographic Message Syntax, CMS).          E' più efficiente ma più complesso.</p>	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL
2.103	

103

Reti di Elaboratori	<h2 style="margin: 0;">Transport Layer Security TLS</h2>
<h3 style="margin: 0;">Transport Layer Security TLS</h3>	
<p>Transport Layer Security, o TLS, è un protocollo di sicurezza ampiamente diffuso progettato per semplificare <b>la privacy e la sicurezza dei dati per le comunicazioni su Internet.</b></p> <p>L'uso principale di TLS è <b>la crittografia della comunicazione</b> tra applicazioni Web e server</p> <p>TLS può essere utilizzato anche per crittografare altre comunicazioni come e-mail, messaggistica e Voice over IP (VoIP)</p> <p>TLS è stato proposto dalla Internet Engineering Task Force (IETF)</p> <p>La versione più recente è TLS 1.3, pubblicata nel 2016</p>	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL
2.104	

104

Reti di Elaboratori

## Cos'è un handshake TLS?

TLS è un **protocollo di crittografia** progettato per proteggere le comunicazioni Internet.  
 Un **handshake TLS** è il processo che avvia una sessione di comunicazione che utilizza la crittografia TLS.  
 Durante un handshake TLS, le due parti comunicanti si **scambiano messaggi per riconoscersi, verificarsi a vicenda, stabilire gli algoritmi di crittografia** che utilizzeranno e concordare le chiavi di sessione.

<https://www.cloudflare.com/it-it/learning/ssl/what-happens-in-a-tls-handshake/>  
<https://www.evemilano.com/tls-session-resumption/>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.105

105

Reti di Elaboratori

## handshake TLS

Un handshake TLS si verifica ogni volta che un utente accede a un sito Web tramite HTTPS e il browser inizia prima a interrogare il server di origine del sito Web.

Gli handshake TLS si verificano dopo che una connessione TCP è stata aperta tramite un handshake TCP.

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.106

106

## Quali sono le fasi di un handshake TLS

1. **Il messaggio 'client hello':** il client inizia l'handshake inviando un messaggio "hello" al server. Il messaggio includerà la versione TLS supportata dal client, le suite di crittografia supportate e una stringa di byte casuali nota come "casuale client".
2. **Il messaggio 'server hello':** in risposta al messaggio di saluto del client, il server invia un messaggio contenente il certificato SSL del server, la suite di cifratura scelta dal server e la stringa "casuale server", un'altra stringa casuale di byte generata dal server.
3. **Autenticazione:** il client verifica il certificato SSL del server con l'autorità di certificazione che lo ha emesso. Ciò conferma che il server è chi dice di essere e che il client sta interagendo con l'effettivo proprietario del dominio.
4. **Il segreto premaster:** il client invia un'altra stringa casuale di byte, il "segreto premaster". Il segreto premaster è crittografato con la chiave pubblica e può essere decrittografato solo con la chiave privata dal server. (Il client ottiene la chiave pubblica dal certificato SSL del server.)
5. **Chiave privata utilizzata:** il server decrittografa il segreto premaster.
6. **Chiavi di sessione create:** sia il client che il server generano chiavi di sessione dalla stringa casuale client, dalla stringa casuale server e dal segreto premaster. Dovrebbero arrivare agli stessi risultati.
7. **Il client è pronto:** il client invia un messaggio "terminato" che viene crittografato con una chiave di sessione.
8. **Il server è pronto:** il server invia un messaggio "terminato" crittografato con una chiave di sessione.
9. **Crittografia simmetrica sicura ottenuta:** l'handshake è stato completato e la comunicazione continua utilizzando le chiavi di sessione.

## Definizione di una sessione sicura tramite TLS

Il protocollo TLS Handshake prevede la procedura seguente:

1. Il client invia un messaggio "Client hello" al server, insieme al valore casuale del client e ai pacchetti di crittografia supportati.
2. Il server risponde inviando un messaggio "Server hello" al client, insieme al valore casuale del server.
3. Il server invia il certificato al client per l'autenticazione e può richiedere un certificato dal client. Il server invia il messaggio "Server hello done".
4. Se il server ha richiesto un certificato dal client, il client lo invia.
5. Il client crea un segreto pre-master casuale e lo crittografa con la chiave pubblica dal certificato del server, inviando il segreto pre-master crittografato al server.
6. Il server riceve il segreto pre-master. Il server e il client generano le chiavi master secret e sessione in base al segreto pre-master.
7. Il client invia una notifica "Modifica spec di crittografia" al server per indicare che il client inizierà a usare le nuove chiavi di sessione per l'hashing e la crittografia dei messaggi. Il client invia anche un messaggio "Client completato".
8. Il server riceve "Cambia spec di crittografia" e commuta lo stato di sicurezza del livello record alla crittografia simmetrica usando le chiavi di sessione. Il server invia un messaggio "Server completato" al client.
9. Il client e il server possono ora scambiare i dati dell'applicazione tramite il canale protetto stabilito. Tutti i messaggi inviati dal client al server e dal server al client vengono crittografati usando la chiave di sessione.

**Reti di Elaboratori**

## Sessione https

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	mfausto.amministrazione.unicam	antoine.unicam.it	TCP	66	2448 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000380	antoine.unicam.it	mfausto.amministrazione.unicam	TCP	66	443 → 2448 [SYN, ACK] Seq=0 Ack=1 Min=64240 Len=0 MSS=1460 SACK_PERM=1
3	0.000362	mfausto.amministrazione.unicam	antoine.unicam.it	TCP	54	2448 → 443 [ACK] Seq=1 Ack=1 Min=2102272 Len=0
4	0.000593	mfausto.amministrazione.unicam	antoine.unicam.it	TLSv1.3	571	Client Hello
5	0.000815	antoine.unicam.it	mfausto.amministrazione.unicam	TCP	60	443 → 2448 [ACK] Seq=1 Ack=518 Min=64128 Len=0
6	0.004151	antoine.unicam.it	mfausto.amministrazione.unicam	TLSv1.3	1514	Server Hello, Change Cipher Spec, Application Data
7	0.004216	mfausto.amministrazione.unicam	antoine.unicam.it	TLSv1.3	1844	Application Data, Application Data, Application Data
8	0.004216	mfausto.amministrazione.unicam	antoine.unicam.it	TCP	54	2448 → 443 [ACK] Seq=518 Ack=2451 Win=2182272 Len=0
9	0.024443	mfausto.amministrazione.unicam	antoine.unicam.it	TLSv1.3	118	Change Cipher Spec, Application Data
10	0.024994	antoine.unicam.it	mfausto.amministrazione.unicam	TLSv1.3	357	Application Data
11	0.025140	antoine.unicam.it	mfausto.amministrazione.unicam	TLSv1.3	357	Application Data
12	0.025168	mfausto.amministrazione.unicam	antoine.unicam.it	TCP	54	2448 → 443 [ACK] Seq=582 Ack=3057 Win=2181760 Len=0
13	0.025904	mfausto.amministrazione.unicam	antoine.unicam.it	TCP	66	2449 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
14	0.030182	antoine.unicam.it	mfausto.amministrazione.unicam	TCP	66	443 → 2449 [SYN, ACK] Seq=0 Ack=1 Min=64240 Len=0 MSS=1460 SACK_PERM=1
15	0.030231	mfausto.amministrazione.unicam	antoine.unicam.it	TCP	54	2449 → 443 [ACK] Seq=1 Ack=1 Min=2102272 Len=0
16	0.030447	mfausto.amministrazione.unicam	antoine.unicam.it	TLSv1.3	672	Client Hello
17	0.030715	antoine.unicam.it	mfausto.amministrazione.unicam	TCP	60	443 → 2449 [ACK] Seq=1 Ack=619 Min=64128 Len=0
18	0.031828	antoine.unicam.it	mfausto.amministrazione.unicam	TLSv1.3	290	Server Hello, Change Cipher Spec, Application Data, Application Data
19	0.032074	mfausto.amministrazione.unicam	antoine.unicam.it	TLSv1.3	118	Change Cipher Spec, Application Data
20	0.032500	antoine.unicam.it	mfausto.amministrazione.unicam	TLSv1.3	357	Application Data
21	0.084823	mfausto.amministrazione.unicam	antoine.unicam.it	TCP	54	2449 → 443 [ACK] Seq=683 Ack=548 Win=2181760 Len=0

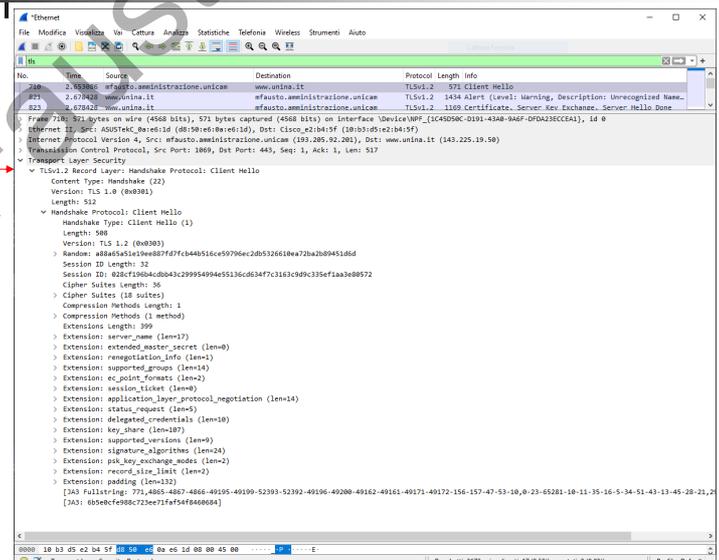
Provare con il file  antoine\_https.pcapng

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL 2.109

109

**Reti di Elaboratori**

Filtro: TLS



**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL 2.110

110

**Reti di Elaboratori**

"tls.handshake.type == 1" for Client Hello  
 "tls.handshake.type == 2" for Server Hello

tls.handshake.type==1						
No.	Time	Source	Destination	Protocol	Length	Info
138	2.398339	mfausto.amministrazione.unicam	wd-prod-ss-eu-west-2-fe.westeurope...	TLSv1.2	571	Client Hello
151	2.422994	mfausto.amministrazione.unicam	172.16.0.15	TLSv1.3	571	Client Hello
191	2.477768	mfausto.amministrazione.unicam	wd-prod-ss-eu-west-2-fe.westeurope...	TLSv1.2	571	Client Hello
249	2.547114	mfausto.amministrazione.unicam	mil04s44-in-f8.1e100.net	QUIC	1292	Initial, DCID=
254	2.547854	mfausto.amministrazione.unicam	172.16.0.15	TLSv1.3	654	Client Hello

tls.handshake.type==2						
No.	Time	Source	Destination	Protocol	Length	Info
159	2.433705	wd-prod-ss-eu-west-2-fe.westeurope...	mfausto.amministrazione.unicam	TLSv1.2	1266	Server Hello, Ce
165	2.435359	172.16.0.15	mfausto.amministrazione.unicam	TLSv1.3	1434	Server Hello, CH
237	2.513741	wd-prod-ss-eu-west-2-fe.westeurope...	mfausto.amministrazione.unicam	TLSv1.2	1266	Server Hello, Ce
264	2.561090	172.16.0.15	mfausto.amministrazione.unicam	TLSv1.3	295	Server Hello, CH
269	2.563766	mil04s44-in-f8.1e100.net	mfausto.amministrazione.unicam	QUIC	1292	Initial, SCID=b8

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL **2.111**

111

**Reti di Elaboratori**

## Certificati

Sicurezza

**Protezione contro contenuti ingannevoli e software a rischio**

- Blocca contenuti a rischio e ingannevoli [Ulteriori informazioni](#)
- Blocca download a rischio
- Avvisa in caso di software indesiderato e non scaricato abitualmente

**Certificati**

- Interroga risponditori OSCP per confermare la validità attuale dei certificati

[Mostra certificati...](#)

[Dispositivi di sicurezza...](#)

Gestione certificati

Certificati personali | Decisioni di autenticazione | Persone | Server | **Autorità**

Sono presenti certificati su file che identificano le seguenti autorità di certificazione

Nome certificato	Dispositivo di sicurezza
> AC Camerfirma S.A.	
> AC Camerfirma SA C/F A82743287	
> ACCV	
> Adialis S.p.A. 03358520967	
> AdisTrust AB	
> AffirmTrust	
> Agence Nationale de Certification Electronique	
> Agencia Catalana de Certificacio (NIF Q-0901176-I)	
> Amazon	
> ANF Autoridad de Certificacion	
> Aseco Data Systems S.A.	
> ARES	
> Autoridad de Certificacion Firmaprofesional C/F A626...	
> Arast Web/Mail Shield	

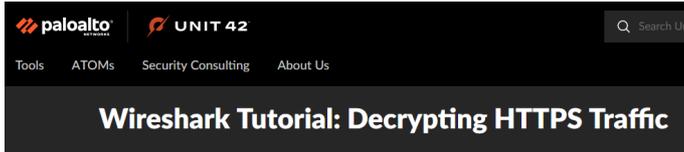
Visualizza... Modifica attendibilità... **Importa...** Esporta Elimina o considera inattendibile... **OK**

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL **2.112**

112

Reti di Elaboratori

## Wireshark Tutorial: Decrypting HTTPS Traffic



<https://unit42.paloaltonetworks.com/wireshark-tutorial-decrypting-https-traffic/>

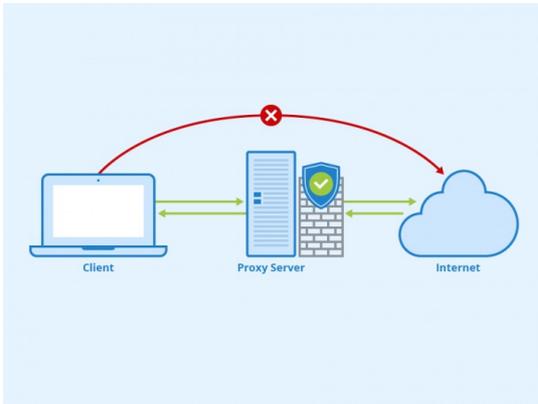
Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.113

113

Reti di Elaboratori

## Web Cache (proxy server)

Web Cache - Proxy server



Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.114

114

Reti di Elaboratori

## Web Cache (proxy server)

**Obiettivo:** rispondere alle richieste evitando di accedere al server remoto

- L'utente configura il browser: accesso attraverso web cache
- Il client invia tutte le richieste al proxy
- La cache restituisce l'oggetto se presente
  - Altrimenti l'oggetto è richiesto prima al server e poi è restituito al client

The diagram illustrates the interaction between clients, a proxy server, and origin servers. Two clients are shown on the left, and two origin servers are on the right. A central proxy server acts as an intermediary. Arrows indicate the flow of traffic: 'Richiesta HTTP' (HTTP Request) goes from clients to the proxy server, and 'Risposta HTTP' (HTTP Response) goes from the proxy server back to the clients. Simultaneously, 'Richiesta HTTP' goes from the proxy server to the origin servers, and 'Risposta HTTP' goes from the origin servers back to the proxy server.

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.115

115

Reti di Elaboratori

## Perché il Web Caching?

**Assunzione:** la cache è "vicina" al client (es., stessa rete locale)

- Tempo di risposta minore: la cache è "più vicina" al client
- Diminuisce il traffico verso server lontani
  - Il link di uscita della rete di un ISP istituzionale/locale è spesso un collo di bottiglia
- Filtraggio dei pacchetti/siti
  - sicurezza
- Accounting degli utenti
  - autenticazione (captive portal)
  - reportistica
  - monitor/controllo utenti (es. scuola)

The diagram shows a central proxy server labeled 'PROXY' connected to several desktop computers representing clients. The proxy server is also connected to a cloud labeled 'Internet', which in turn is connected to several server racks labeled 'Web Servers'.

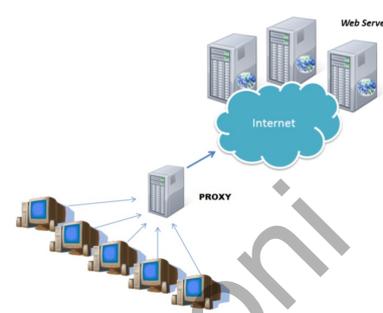
Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.116

116

**Reti di Elaboratori**

## Perché il Web Caching?

- Ritardo TOTALE = Ritardo sulla LAN + Ritardo sul LINK di accesso + Ritardo di Internet
- Esempio
  - dimensione media oggetti = 100 Kbit
  - richieste al secondo sulla rete Aziendale = 15 (in media)
  - ritardo medio di Internet = 2 sec
- Intensità del traffico sulla LAN =  $(15 \text{ richieste/sec.}) \times (100 \text{ Kbit/richiesta}) / (10 \text{ Mbit/sec.}) = 0,15$
- Intensità del traffico sul LINK =  $(15 \text{ richieste/sec.}) \times (100 \text{ Kbit/richiesta}) / (1,5 \text{ Mbit/sec.}) = 1$
- COSA FARE ?
  - Aumentare la velocità del LINK sino a 100 Mbit/sec (COSTOSA)
  - WEB CACHE →
  - 40% di richieste in media soddisfatte dalla cache
  - Con il 60% delle richieste l'intensità del traffico sul link passa da 1 a 0,6



**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL 2.117

117

**Reti di Elaboratori**

## GET condizionale (conditional GET)

HTTP presenta un meccanismo che permette alla cache di verificare se i dati sono aggiornati

- **Obiettivo:** non inviare oggetti che il client ha già in cache
- client: data dell'oggetto memorizzato in cache  
`If-modified-since: <date>`  
(questa è la data della precedente GET)
- server: la risposta è vuota se l'oggetto in cache è aggiornato:  
`HTTP/1.0 304 Not Modified`

<u>client</u>		<u>server</u>
http request msg If-modified-since: <date>	→	object not modified
	←	http response HTTP/1.0 304 Not Modified
-----		
http request msg If-modified-since: <date>	→	object modified
	←	http response HTTP/1.1 200 OK ... <data>

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL 2.118

118

Reti di Elaboratori

## Esercitazione

Esercitazione con



No.	Time	Source	Destination	Protocol	Info
10	1.985466	192.168.1.2	128.119.245.12	HTTP	GET
13	2.368092	128.119.245.12	192.168.1.2	HTTP	HTTP
21	7.705285	192.168.1.2	128.119.245.12	HTTP	GET
22	8.081586	128.119.245.12	192.168.1.2	HTTP	HTTP

```

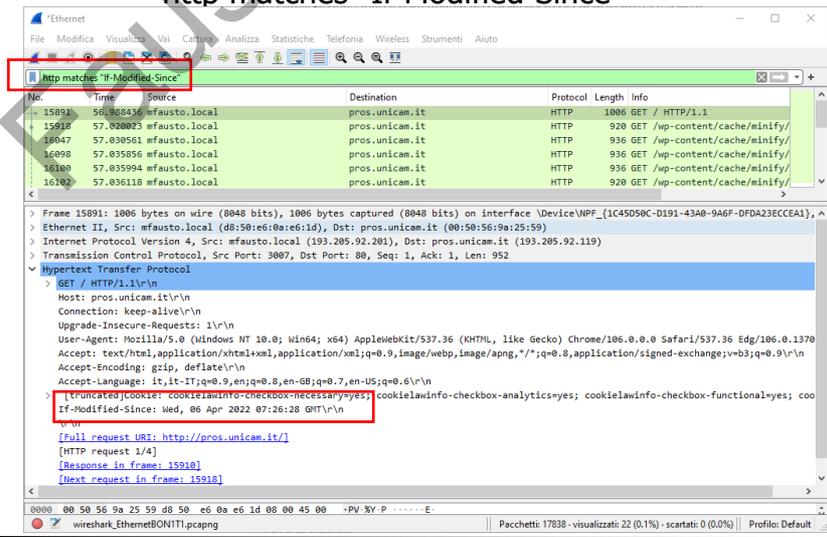
Frame 21 (403 bytes on wire, 403 bytes captured)
Ethernet II, Src: CompalE1_80:aa:38 (00:0f:b0:80:aa:38)
Internet Protocol, Src: 192.168.1.2 (192.168.1.2), Dst:
Transmission Control Protocol, Src Port: lupa (1212), D
Hypertext Transfer Protocol
  GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
  Accept: */*\r\n
  Accept-Language: th\r\n
  Accept-Encoding: gzip, deflate\r\n
  If-Modified-Since: Fri, 13 Jun 2008 04:45:01 GMT\r\n
  If-None-Match: "d6c96-173-f0cbad40"\r\n
  
```

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.119

119

Reti di Elaboratori

## http matches "If-Modified-Since"



No.	Time	Source	Destination	Protocol	Length	Info
15891	56.868436	mfausto.local	pros.unicam.it	HTTP	1006	GET / HTTP/1.1
15918	57.920823	mfausto.local	pros.unicam.it	HTTP	920	GET /wp-content/cache/minify/
16047	57.930561	mfausto.local	pros.unicam.it	HTTP	936	GET /wp-content/cache/minify/
16098	57.935856	mfausto.local	pros.unicam.it	HTTP	936	GET /wp-content/cache/minify/
16180	57.935994	mfausto.local	pros.unicam.it	HTTP	936	GET /wp-content/cache/minify/
16182	57.936118	mfausto.local	pros.unicam.it	HTTP	920	GET /wp-content/cache/minify/

```

Frame 15891: 1006 bytes on wire (8048 bits), 1006 bytes captured (8048 bits) on interface Device\WPF_{1C45058C-D191-43A0-9A6F-DFDA23ECC6A1},
Ethernet II, Src: mfausto.local (d8:59:e6:0a:e6:1d), Dst: pros.unicam.it (00:50:56:9a:25:59)
Internet Protocol Version 4, Src: mfausto.local (193.205.92.201), Dst: pros.unicam.it (193.205.92.119)
Transmission Control Protocol, Src Port: 3007, Dst Port: 80, Seq: 1, Ack: 1, Len: 952
Hypertext Transfer Protocol
  GET / HTTP/1.1\r\n
  Host: pros.unicam.it\r\n
  Connection: keep-alive\r\n
  Upgrade-Insecure-Requests: 1\r\n
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36 Edg/106.0.1370
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: it,it-IT;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6\r\n
  If-Modified-Since: Wed, 06 Apr 2022 07:26:28 GMT\r\n
  [Truncated cookie: cookieLawInfo-checkbox-necessary=yes; cookieLawInfo-checkbox-analytics=yes; cookieLawInfo-checkbox-functional=yes; coo
  \r\n
  Full request URI: http://pros.unicam.it/
  [HTTP request 1/4]
  [Response in frame: 15918]
  [Next request in frame: 15918]
  
```

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.120

120

**Reti di Elaboratori**

## Cercare una stringa con wireshark

The screenshot shows the Wireshark interface with the search menu open. The menu options are:

- Trova pacchetto... (Ctrl+F)
- Trova successivo (Ctrl+N)
- Trova precedente (Ctrl+B)
- Marca/Deseleziona pacchetto (Ctrl+M)
- Marca tutti i visualizzati (Ctrl+Shift+M)
- Rimuovi la selezione da tutti i visualizzati (Ctrl+Alt+M)
- Marchio successivo (Ctrl+Shift+N)
- Marchio precedente (Ctrl+Shift+B)
- Ignora/Considera pacchetto (Ctrl+D)
- Ignora tutti i visualizzati (Ctrl+Shift+D)
- Considera tutti i visualizzati (Ctrl+Alt+D)
- Imposta/Rimuovi il riferimento temporale (Ctrl+T)
- Rimuovi tutti i riferimenti temporali (Ctrl+Alt+T)

The packet list shows the following entries:

No.	Time	Source	Destination	Protocol	Length	Info
				Stringa		www.gazzeta.it
				inistraz.. QUIC	70	Protected Payload (KP0),
				inistraz.. TCP	60	443 → 16856 [ACK] Seq=27
				inistraz.. QUIC	72	Protected Payload (KP0),
				inistraz.. TLSv1.3	189	Application Data
				inistraz.. QUIC	70	Protected Payload (KP0),
				inistraz.. TCP	60	443 → 16870 [ACK] Seq=1
				inistraz.. TLSv1.2	1354	Server Hello
				inistraz.. TCP	1354	443 → 16870 [ACK] Seq=13

At the bottom of the screenshot, there is a footer with the text: **Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL 2.121

121

**Reti di Elaboratori**

## Esercitazioni

internet reti sicurezza

**Esercitazioni**

At the bottom of the slide, there is a footer with the text: **Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL 2.122

122

Reti di Elaboratori

## Protocolli di livello applicativo

- ✓ **FTP: file transfer protocol**
- ✓ **TFTP: Trivial File Transfer Protocol**
- ✓ **Posta elettronica**
  - **smtp**
  - **pop3 - imap**
- ✓ **DNS: Domain Name System**
- ✓ **SNMP: Simple Network Management Protocol**
- ✓ **Remote login: telnet - ssh**

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.123

123

Reti di Elaboratori

## ftp: File transfer protocol

```

graph LR
    Utente[Utente] --> UI[Interfaccia utente FTP]
    UI <--> Client[Client FTP]
    Client <--> FS_L[File system locale]
    Client <--> Server[Server FTP]
    Server <--> FS_R[File system remoto]
    Client <--> |Trasferimento file| Server
  
```

- Trasferimento file da/verso un host remoto
- Usa il modello client/server
  - *client*: parte che richiede il trasferimento (da/verso l' host remoto)
  - *server*: host remoto
- ftp: RFC 959
- ftp server: porta 21/20

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.124

124

Reti di Elaboratori

### ftp: connessioni controllo e dati separate

- Il client contatta il server sulla porta 21, specificando TCP come protocollo di trasporto
- due connessioni TCP parallele:
  - controllo**: scambio di messaggi di controllo tra client e server.  
"controllo fuori banda"
  - dati**: trasferimento dati da/verso il server
- Entrambi le connessioni aperte dal client
- Il server ftp mantiene info di "stato": directory corrente, autenticazione
- Una nuova connessione per ogni file trasferito

Client FTP      Server FTP

Porta 21 per la connessione di controllo TCP

Porta 20 per la connessione dati TCP

Fausto Marcantoni      Chapter 2 APPLICATION PROTOCOL      2.125

125

Reti di Elaboratori

### ftp comandi, risposte

Esempi di comandi:

- Inviati come testo ASCII mediante il canale di controllo
- USER** *username*
- PASS** *password*
- LIST** richiede la lista dei file nella directory corrente (ls)
- RETR** <file> richiede (get) un file
- STOR** <file> scarica (put) un file sull' host remoto

Esempi di codici

- Codice di stato e frase (come in http)
- 331 Username OK, password required**
- 125 data connection already open; transfer starting**
- 425 Can't open data connection**
- 452 Error writing file**

Fausto Marcantoni      Chapter 2 APPLICATION PROTOCOL      2.126

126

Reti di Elaboratori

## Esempio ftp 1/2

```

C:\Documents and Settings\mf>ftp stesi.cs.unicam.it
Connesso a stesi.cs.unicam.it.
220 FTP Server ready.
Utente (stesi.cs.unicam.it:(non)): anonymous
331 Anonymous login ok, send your complete email address as your password.
Password:
230-
*** Welcome to this anonymous ftp server! ***

You are user 1 out of a maximum of 10 authorized anonymous logins.
The current time here is Wed Oct 18 16:31:41 2006.
If you experience any problems here, contact : root@localhost

230 Anonymous login ok, restrictions apply.
ftp> dir
200 PORT command successful
150 Opening ASCII mode data connection for file list
drwxr-xr-x  4 ftp      ftp      4096 Sep 14 14:42 .
drwxr-xr-x  4 ftp      ftp      4096 Sep 14 14:42 ..
drwxr-xr-x  2 ftp      ftp      4096 Aug 23 14:52 pub
d-wx-wx-x  2 ftp      ftp      4096 Aug 23 14:52 uploads
-rw-r--r--  1 ftp      ftp      224 Nov  9  2004 welcome.msg
226 Transfer complete.
ftp: 309 byte ricevuti in 0,00secondi 309000,00Kbyte/sec)
ftp>

```

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.127

127

Reti di Elaboratori

## Esempio ftp 2/2

```

ftp> dir
200 PORT command successful
150 Opening ASCII mode data connection for file list
drwxr-xr-x  4 ftp      ftp      4096 Sep 14 14:42 .
drwxr-xr-x  4 ftp      ftp      4096 Sep 14 14:42 ..
drwxr-xr-x  2 ftp      ftp      4096 Aug 23 14:52 pub
d-wx-wx-x  2 ftp      ftp      4096 Aug 23 14:52 uploads
-rw-r--r--  1 ftp      ftp      224 Nov  9  2004 welcome.msg
226 Transfer complete.
ftp: 309 byte ricevuti in 0,00secondi 309000,00Kbyte/sec)
ftp> cd pub
200 CWD command successful
ftp> dir
200 PORT command successful
150 Opening ASCII mode data connection for file list
drwxr-xr-x  2 ftp      ftp      4096 Aug 23 14:52 .
drwxr-xr-x  4 ftp      ftp      4096 Sep 14 14:42 ..
226 Transfer complete.
ftp: 117 byte ricevuti in 0,00secondi 117000,00Kbyte/sec)
ftp> help
I comandi possono essere abbreviati. I comandi sono:

!          delete      literal      prompt      send
?          debug        ls           put         status
append    dir             mdelete     pwd         trace
ascii     disconnect     mdir        quit        type
bell      get            mget        quote       user
binary    glob           mkdir       recv        verbose
bye       hash           mls         remotehelp
cd        help           mput        rename
close    lcd            open        rmdir
ftp> bye
C:\Documents and Settings\mf>

```

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.128

128

Reti di Elaboratori

## Esempio ftp via browser

<ftp://ftp.funet.fi/>

Indice di ftp://ftp.funet.fi/

Nome	Dimensione	Ultima modifica
File: README	18 KB	18/06/2019 21:14:00
dev		07/12/2015 01:00:00
File: favicon.ico	2 KB	07/04/2006 02:00:00
index		07/09/2017 02:00:00
pub		18/06/2019 20:25:00
rfc		06/02/2018 01:00:00

ftp://user:password@ftp.funet.fi/

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.129

129

Reti di Elaboratori

## Esempio ftp via terminale

```

Microsoft Windows [Versione 10.0.18362.418]
(c) 2019 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\fausto_mfausto>ftp ftp.funet.fi
Connesso a ftp.funet.fi.
220----- Welcome to Pure-FTPd [privsep] -----
220-You are user number 56 of 1000 allowed.
220-Local time is now 11:17. Server port: 21.
220-Only anonymous FTP is allowed here
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 30 minutes of inactivity.
504 Unknown command
Utente (ftp.funet.fi:(none)): anonymous
331 Any password will work
Password:
230 Any password will work
ftp> dir
200 PORT command successful
150 Connecting to port 3445
-rw-rw-r-- 1 108 42 18211 Jun 18 19:14 README
drwxr-xr-x 2 0 0 125 Dec 7 2015 dev
-rwxr-xr-x 1 0 0 1078 Apr 7 2006 favicon.ico
drwxrwxr-x 3 50028 200 12288 Sep 7 2017 index
drwxr-xr-x 3 0 0 4096 Jun 18 18:25 pub
drwxrwxr-x 9 0 0 1835008 Feb 6 2018 rfc
226-Options: -l
226 6 matches total
ftp: 418 bytes received in 0.02secondi 26.13kbyte/sec)
ftp>
  
```

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.130

130

**Reti di Elaboratori**

## Windows Firewall

Avviso di Sicurezza di Windows

Windows Defender Firewall ha bloccato alcune funzionalità di questa app

Windows Defender Firewall ha bloccato alcune funzionalità di WinSCP: SFTP, FTP and SCP client in tutte le reti pubbliche, private e di dominio.

Nome: WinSCP: SFTP, FTP and SCP client  
Autore: Martin Prikyl  
Percorso: C:\users\fausto.mfausto\desktop\tools\network\winscp.exe

Consenti a WinSCP: SFTP, FTP and SCP client di comunicare su queste reti:

Reti private, ad esempio una rete domestica o aziendale

Reti pubbliche, ad esempio in aeroporti e Internet café (scelta non consigliata perché il livello di sicurezza di queste reti è spesso insufficiente o del tutto assente)

[Dettagli dell'autorizzazione di app attraverso un firewall](#)

Consenti accesso Annulla

Windows Defender Firewall con sicurezza avanzata su

- Regole connessioni in entrata
- Regole connessioni in uscita
- Regole di sicurezza delle connessioni
- Monitoraggio

Regole connessioni in entrata

Nome	Gruppo	Profilo	Abilitata	Oper
fiddler.exe		Pubblico	SI	Blocc
fiddler.exe		Pubblico	SI	Blocc
FiddlerProxy		Tutti	SI	Cons
Firefox (C:\Program Files\Mozilla Firefox)		Privato	SI	Cons
Firefox (C:\Program Files\Mozilla Firefox)		Privato	SI	Cons
Java(TM) Platform SE binary		Pubblico	SI	Cons
Java(TM) Platform SE binary		Pubblico	SI	Cons
Programma di trasferimento file (FTP)		Pubblico	SI	Cons
Programma di trasferimento file (FTP)		Privato	SI	Cons
Programma di trasferimento file (FTP)		Privato	SI	Cons
Programma di trasferimento file (FTP)		Pubblico	SI	Cons
winscp.exe		Privato	SI	Blocc
winscp.exe		Privato	SI	Blocc

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.131

131

**Reti di Elaboratori**

## How to configure Windows firewall from the command line

**Start / Stop firewall service**

```
netsh firewall set opmode DISABLE
netsh firewall set opmode ENABLE
```

**Allow/Deny Ports**

```
netsh firewall add portopening TCP _port_number_ _name_ DISABLE ALL
netsh firewall add portopening TCP 3264 CCMail DISABLE ALL
```

```
netsh firewall add portopening TCP _port_number_ _name_ ENABLE ALL
netsh firewall add portopening TCP 8443 PLESK-ADMIN ENABLE ALL
```

**Programs to not allow TCP/UDP Socket Connections**

```
netsh firewall add allowedprogram _path_ _name_ DISABLE ALL
netsh firewall add allowedprogram C:\Windows\System32\ftp.exe FTP DISABLE ALL
```

**Allow/Deny Desktop Popup**

```
netsh firewall set notifications DISABLE
```

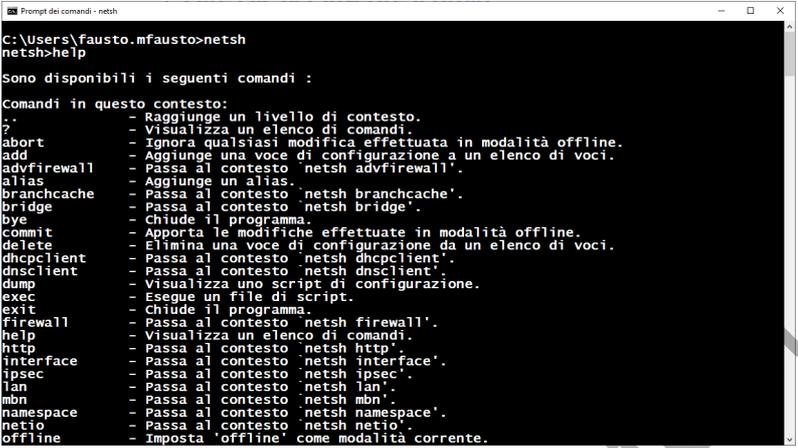
<https://www.simplified.guide/windows/firewall-command>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.132

132

Reti di Elaboratori

## netsh



<https://docs.microsoft.com/it-it/windows-server/networking/technologies/netsh/netsh-contexts>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.133

133

Reti di Elaboratori

## Altri sistemi client ftp



SSH Secure File Transfer Client  
Collegamento  
2 KB

<https://shareware.unc.edu/>



PuTTY

<http://en.wikipedia.org/wiki/PuTTY>  
<http://www.chiark.greenend.org.uk/~sgtatham/putty/>



**WinSCP**

[http://winscp.net/eng/docs/free\\_ssh\\_client\\_for\\_windows](http://winscp.net/eng/docs/free_ssh_client_for_windows)



**FileZilla**

<https://filezilla-project.org/>

**Install**

Just install tnftp with brew.

```
$ brew install tnftp
```

<https://osxdaily.com/2018/08/07/get-install-ftp-mac-os/>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.134

134

Reti di Elaboratori

## ftp su Windows 10 - problem



I have setup FTP server in Ubuntu 12.04 LTS.

Now when I try to connect to FTP server from Windows 7 through command-line `ftp.exe`, I get successfully connected but I cannot get the list of directory. I get error

```
200 PORT command successful. Consider using PASV.
425 Failed to establish connection.
```

<https://stackoverflow.com/questions/19516263/200-port-command-successful-consider-using-pasv-425-failed-to-establish-connec>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.135

135

Reti di Elaboratori

## PASV o PORT

### D : PASV o PORT , cosa sono e quale usare?

R : Molti degli utenti ftp non sanno cosa siano i comandi PASV (Passive) e PORT, ma naturalmente questo non è un difetto. Innanzi tutto occorre sapere **che ogni sessione FTP utilizza 2 connessioni TCP**, e quando ci si connette ad un server FTP quest'ultimo utilizza la prima, e quando si richiede una LISTA DIRECTORY o un TRASFERIMENTO FILE è richiesta la seconda connessione TCP, e questi 2 comandi sono utilizzati per COSTRUIRE (negoziare) la seconda connessione TCP.

Potrà sembrare strano il fatto che esistano 2 comandi per la connessione DATI, e non uno solo, e questa potrebbe essere una domanda interessante. Qualche tempo fa, il comando PORT era probabilmente sufficiente, tuttavia con l'introduzione in quantità di software FIREWALL e dispositivi NAT, il comando PORT è diventato inutile, così fu introdotto il **comando PASV, anche conosciuto come trasferimento dati compatibile FIREWALL**.

Vediamo in dettaglio qual è la differenza tra questi 2 comandi:

PORT : il cliente ftp CHIEDE al server ftp di collegarsi all'indirizzo IP del cliente ftp (con un numero di porta)

PASV : il server ftp CHIEDE al cliente ftp di collegarsi all'indirizzo IP del server ftp (con un numero di porta)

Così non è difficile realizzare quanto segue:

PORT fallirà : se il cliente ftp non conosce il suo reale indirizzo (succede quando il cliente è dietro ad un NAT)

PASV fallirà : se il server ftp non conosce il suo reale indirizzo (succede quando il server è dietro ad un NAT)

Dove è possibile vedere i comandi PORT/PASV e relative risposte?! Bene, procuratevi un buon cliente FTP con una finestra che mostri i comandi :), per esempio : FlashFXP, CuteFTP, Windows Commander e molti altri che non elenchiamo .. ma naturalmente, esistono alcuni clienti FTP che nascondono questi comandi/risposte, in modo che non si vedano affatto.

Adesso dovrebbe essere chiaro cosa configurare nel caso si incontrino problemi con PORT and PASV ..

<http://www.raidenftpd.com/kb/it-kb000000010.html>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.136

136

Reti di Elaboratori

## ftp - wireshark

No.	Time	Source	Destination	Protocol	Length	Info
6	0.005864	192.168.49.128	192.168.49.1	FTP	74	Response: 220 (vsFTPD 3.0.5)
7	0.010615	192.168.49.1	192.168.49.128	FTP	68	Request: OPTS UTF8 ON
10	0.011071	192.168.49.128	192.168.49.1	FTP	80	Response: 200 Always in UTF8 mode.
16	6.329249	192.168.49.1	192.168.49.128	FTP	70	Request: USER anonymous
19	6.330019	192.168.49.128	192.168.49.1	FTP	88	Response: 331 Please specify the pas
22	9.405190	192.168.49.1	192.168.49.128	FTP	69	Request: PASS password
25	9.410733	192.168.49.128	192.168.49.1	FTP	77	Response: 230 Login successful.
28	11.432820	192.168.49.1	192.168.49.128	FTP	60	Request: QUIT
30	11.433847	192.168.49.128	192.168.49.1	FTP	68	Response: 221 Goodbye.

Frame 6: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF\_{0411EE02-1A51-4070-9150-04511FA1E878}, id 0  
 Ethernet II, Src: VMware\_E1:47:ac (00:0c:29:62:47:ac), Dst: VMware\_c8:00:08 (00:50:56:c8:00:08)  
 Internet Protocol Version 4, Src: 192.168.49.128 (192.168.49.128), Dst: 192.168.49.1 (192.168.49.1)  
 Transmission Control Protocol, Src Port: 21, Dst Port: 5709, Seq: 1, Ack: 1, Len: 20  
 File Transfer Protocol (FTP)  
 [Current working directory: ]

0000 00 50 56 c8 00 08 00 0c 29 62 47 ac 08 00 45 00 PV.....)b6...E

VMware Network Adapter VMnet8: <live capture in progress> | Pacchetto: 52 - visualizzati: 9 (17.5%) | Profilo: Default

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.137

137

Reti di Elaboratori

## Protocolli di livello applicativo

- ✓ **FTP: file transfer protocol**
- ✓ **TFTP: Trivial File Transfer Protocol**
- ✓ **Posta elettronica**
  - **smtp**
  - **pop3 - imap**
- ✓ **DNS: Domain Name System**
- ✓ **SNMP: Simple Network Management Protocol**

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.138

138

Reti di Elaboratori	<h2 style="margin: 0;">TFTP (Trivial File Transfer Protocol)</h2>
<p>TFTP è un protocollo molto semplice utilizzato per trasferire file tra host utilizzando UDP.</p> <p>definito nella <b>RFC 1350</b></p> <p>protocollo di trasporto <b>UDP</b> (<i>User Datagram Protocol</i>).</p> <p>La porta sulla quale è in ascolto il server TFTP server è la <b>69</b>.</p> <ul style="list-style-type: none"> <li>❖ molto semplice</li> <li>❖ meno funzioni rispetto a FTP</li> <li>❖ <b>non</b> può <b>leggere directory</b></li> <li>❖ <b>non</b> è provvisto di <b>autenticazione</b></li> <li>❖ utilizzo è limitato</li> </ul> <p>TFTP viene solitamente usato per trasferire file tra un computer ed un altro dispositivo come router o switch in ambito LAN.</p>	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL
2.139	

139

Reti di Elaboratori	<h2 style="margin: 0;">TFTP (Trivial File Transfer Protocol)</h2>
<ul style="list-style-type: none"> <li>• TFTP trasmette pacchetti con una <b>lunghezza fissa di 512 byte</b>.</li> <li>• Un pacchetto avente una dimensione inferiore rappresenta l'<b>ultimo pacchetto trasmesso</b>.</li> <li>• I pacchetti dati inviati vengono memorizzati in un buffer fino alla ricezione della avvenuta accettazione da parte dell'host remoto.</li> <li>• In caso di mancata conferma della ricezione entro un determinato tempo di un pacchetto, quest'ultimo viene ritrasmesso.</li> </ul> <p>La modalità di trasferimento dati di TFTP è di due tipi:</p> <ul style="list-style-type: none"> <li>- <b>NETASCII</b> per i file di testo;</li> <li>- <b>OCTET</b> per i file binari.</li> </ul> <p>I pacchetti utilizzati durante una sessione TFTP sono di cinque tipi:</p> <ul style="list-style-type: none"> <li>- <b>RR</b>: Read Request (<i>Richiesta di lettura</i>);</li> <li>- <b>WR</b>: Write Request (<i>Richiesta di scrittura</i>);</li> <li>- <b>DATA</b>: Dati;</li> <li>- <b>ACK</b>: Acknowledgment (<i>Accettazione</i>);</li> <li>- <b>ERR</b>: Errore;</li> </ul> <p>Le fasi di una sessione TFTP:</p> <ol style="list-style-type: none"> <li>1. Il client contatta il server inviando un pacchetto di tipo <b>RR</b> (<i>richiesta di lettura</i>) o <b>WR</b> (<i>richiesta di scrittura</i>);</li> <li>2. Il server, se accetta la connessione, risponde inviando/ricevendo pacchetti <b>DATA</b> di 512 byte. Per ogni pacchetto inviato/ricevuto regolarmente viene inviato/ricevuto un <b>ACK</b> altrimenti un <b>ERROR</b>;</li> <li>3. I pacchetti vengono trasferiti finché la loro lunghezza non è inferiore a 512 byte;</li> <li>4. Termine della connessione;</li> </ol>	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL
2.140	

140

Reti di Elaboratori

## esercitazioni

internet reti sicurezza

Esercitazioni

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.141

141

Reti di Elaboratori

## Protocolli di livello applicativo

- ✓ **FTP: file transfer protocol**
- ✓ **TFTP: Trivial File Transfer Protocol**
- ✓ **Posta elettronica**
  - **smtp**
  - **pop3 - imap**
- ✓ **DNS: Domain Name System**
- ✓ **SNMP: Simple Network Management Protocol**
- ✓ **Remote login: telnet - ssh**

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.142

142

Reti di Elaboratori

2.143

Chapter 2 APPLICATION PROTOCOL

Fausto Marcantoni

143

Reti di Elaboratori

### Alice, Bob e Trudy

- ✓ Bob e Alice (amanti!) vogliono comunicare “in modo sicuro”
- ✓ Trudy (l'intrusa) potrebbe intercettare, cancellare, aggiungere messaggi

[https://it.wikipedia.org/wiki/Alice\\_e\\_Bob](https://it.wikipedia.org/wiki/Alice_e_Bob)

2.144

Fausto Marcantoni

144

Reti di Elaboratori

## Posta elettronica

**Tre componenti principali:**

- User agent
- Server di posta
- Simple Mail Transfer Protocol: SMTP

**User Agent**

- "Lettore di posta"
- Composizione e lettura di messaggi di posta
- Es.: [Outlook](#), [Thunderbird](#), [Mailbird](#), [Newton Mail](#), [eM Client](#), [Zimbra Desktop](#), [Claws Mail](#)
- Gestione dei messaggi in ingresso/uscita memorizzati sul server

Legenda:

- ▨ Coda di messaggi
- ▣ Casella di posta

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.145

145

Reti di Elaboratori

## Posta elettronica: mail server

**Mail Server**

- Mailbox contenente messaggi dell'utente (*non ancora letti*)
- Coda di messaggi in uscita (*non ancora spediti*)
- Protocol SMTP tra i mail server per il recapito dei messaggi
  - client: il server che invia il messaggio
  - server: server che riceve il messaggio

**Nota: solo una distinzione di ruoli**

Legenda:

- ▨ Coda di messaggi
- ▣ Casella di posta

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.146

146

Reti di Elaboratori

### Recapito di un messaggio di posta elettronica

1. Alice compone un messaggio e lo inoltra al suo Mail Server
2. Mail Server dispone il messaggio nella coda di messaggi in uscita
3. Mail Server di Alice apre una connessione SMTP con il Mail Server di Bob ed inoltra il messaggio
4. Se il contatto fallisce, l'invio è ripetuto ogni "n"  $\Delta$ time (minuti)
5. Se l'invio fallisce dopo "m"  $\Delta$ time (giorni), una mail di notifica inviata ad Alice
6. Mail Server di Bob riceve il messaggio dal Mail Server di Alice e lo salva nella Mailbox di Bob
7. Bob accede la propria Mailbox specificando Username e Password
8. Messaggi possono essere trasferiti dalla Mailbox all'host da cui Bob ha acceduto e/o lasciati sul server
9. Bob legge il messaggio di Alice

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.147

147

Reti di Elaboratori

### Posta elettronica: SMTP [RFC 821] (1982!)

- Usa TCP per il trasferimento affidabile dei messaggi da client a server, porta 25
- Trasferimento diretto: da server a server, non si usano server intermedi di posta
- Tre fasi
  - Handshaking (saluto)
    - Trasferimento di uno o più messaggi (connessione permanente)
    - Chiusura
  - Interazione mediante comandi/risposte
    - **Comando:** testo ASCII
    - **Risposta:** codice di stato e frase
- **Attenzione:** I messaggi devono essere comunque riportati in formato ASCII a 7 bit, **anche dati multimediali**

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.148

148

Reti di Elaboratori	<h2>Esempio di interazione SMTP</h2>	
<pre> S: 220 hamburger.edu C: HELO crepes.fr S: 250 Hello crepes.fr, pleased to meet you C: MAIL FROM: &lt;alice@crepes.fr&gt; S: 250 alice@crepes.fr... Sender ok C: RCPT TO: &lt;bob@hamburger.edu&gt; S: 250 bob@hamburger.edu ... Recipient ok C: DATA S: 354 Enter mail, end with "." on a line by itself C: Do you like ketchup? C:   How about pickles? C: . S: 250 Message accepted for delivery C: QUIT S: 221 hamburger.edu closing connection </pre>		
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL	2.149

149

Reti di Elaboratori	<h2>SMTP: considerazioni</h2>		
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; padding: 5px; vertical-align: top;"> <ul style="list-style-type: none"> <li>■ Connessioni TCP persistenti</li> <li>■ Richiede che il messaggio (header &amp; corpo) sia in <b>formato ascii 7-bit</b></li> <li>■ Alcune sequenze di caratteri non consentite (es., CRLF.CRLF). Conseguenza: il messaggio deve essere codificato</li> <li>■ Il server smtp usa CRLF.CRLF per determinare la fine del messaggio</li> </ul> </td> <td style="width: 50%; padding: 5px; vertical-align: top;"> <p style="color: red; margin: 0;"><b>Confronto con http</b></p> <ul style="list-style-type: none"> <li>■ http: pull protocol</li> <li>■ email: push protocol</li> <li>■ Entrambi usano un'interazione mediante comandi/risposta in testo ASCII e codici di stato</li> <li>■ http: ogni oggetto incapsulato nel messaggio di risposta</li> <li>■ smtp: tutti gli oggetti sono inviati mediante un unico messaggio (eventualmente in più parti)</li> </ul> </td> </tr> </table>		<ul style="list-style-type: none"> <li>■ Connessioni TCP persistenti</li> <li>■ Richiede che il messaggio (header &amp; corpo) sia in <b>formato ascii 7-bit</b></li> <li>■ Alcune sequenze di caratteri non consentite (es., CRLF.CRLF). Conseguenza: il messaggio deve essere codificato</li> <li>■ Il server smtp usa CRLF.CRLF per determinare la fine del messaggio</li> </ul>	<p style="color: red; margin: 0;"><b>Confronto con http</b></p> <ul style="list-style-type: none"> <li>■ http: pull protocol</li> <li>■ email: push protocol</li> <li>■ Entrambi usano un'interazione mediante comandi/risposta in testo ASCII e codici di stato</li> <li>■ http: ogni oggetto incapsulato nel messaggio di risposta</li> <li>■ smtp: tutti gli oggetti sono inviati mediante un unico messaggio (eventualmente in più parti)</li> </ul>
<ul style="list-style-type: none"> <li>■ Connessioni TCP persistenti</li> <li>■ Richiede che il messaggio (header &amp; corpo) sia in <b>formato ascii 7-bit</b></li> <li>■ Alcune sequenze di caratteri non consentite (es., CRLF.CRLF). Conseguenza: il messaggio deve essere codificato</li> <li>■ Il server smtp usa CRLF.CRLF per determinare la fine del messaggio</li> </ul>	<p style="color: red; margin: 0;"><b>Confronto con http</b></p> <ul style="list-style-type: none"> <li>■ http: pull protocol</li> <li>■ email: push protocol</li> <li>■ Entrambi usano un'interazione mediante comandi/risposta in testo ASCII e codici di stato</li> <li>■ http: ogni oggetto incapsulato nel messaggio di risposta</li> <li>■ smtp: tutti gli oggetti sono inviati mediante un unico messaggio (eventualmente in più parti)</li> </ul>		
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL	2.150	

150

Reti di Elaboratori

## Formato dei messaggi

smtp: protocollo per lo scambio di messaggi di posta  
 RFC 822: standard per il formato dei messaggi inviati:

- header, es.,
  - To:
  - From:
  - Subject:*Diversi dai comandi smtp!*
- body
  - Il "messaggio" vero e proprio, solo caratteri ASCII

header

body

Linea vuota

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.151

151

Reti di Elaboratori

## Formato: estensioni multimediali

- MIME: multipurpose internet mail extension, RFC 2045, 2056.
  - Dati Multimediali e di specifiche applicazioni
- Righe aggiuntive dell'header specificano il tipo del contenuto MIME

Versione MIME

Metodo di codifica

Tipo di dato multimediale, sottotipo, dichiarazione di parametri

Dati codificati

```

From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

base64 encoded data .....
.....base64 encoded data
  
```

<http://toolset.mnwebmaster.it/dev/base64-encoder-decoder.html>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.152

152

Reti di Elaboratori

## Image <-> base64

base64

```

1VB0RwRKG0AAAANSUHEUGAAA2AAAAGQCA
YAAACAvzMAAAAgAELEQVR4Xux9B5gJR5n2
q6z3Dc/ObJMedf2srt065wx3p14
/26CDJjEE5P3DwG7JAgU6wR7r3TLcXwQ
hs9Y7r3FUG6M5S1ccN8qPp
/5qrU6q1staXR5C57M42kqurq1r99pFe
z9bR8xuG1SjTFM3JpG0RnwzQwqBghP
3W6FQcKqKdPhdPkgv12f4B3tEs9Imz
Rmsh1g5SkYDNAdBkxGAN5w2JAwPwe8bW
JQAP2Dg8B4+j7
/ubXp8RmZueBbc3HbDZ3kus1nEC5Q1A
QLAnKbNw133Bwv74BA1hZkGoeDpRB1
G9091s9KYehJSMmYUahQnGZ3Y1PLwAKQX
b5Bsz154XAYw4098ABRaE3mqdnYHE4X3Gm
Z8HjT4U7LYKYvq1k5mgoS5ABkqu31LF0Pg

```

Import from file    Save as...    Copy to clipboard

png



Chain with...    Save as...    Copy to clipboard

<https://onlinepngtools.com/convert-base64-to-png>

<https://www.base64-image.de/>

Fausto Marcontoni    Chapter 2 APPLICATION PROTOCOL    2.153

153

Reti di Elaboratori

## Tipi MIME

Content-Type: type/subtype; parameters

**Text**

- Esempi di sottotipi:  
plain, html

**Image**

- Esempi di sottotipi :  
jpeg, gif

**Audio**

- Esempi di sottotipi :  
basic (8-bit mu-law encoded)  
32kadpcm (32 kbps coding)

**Video**

- Esempi di sottotipi :  
mpeg, quicktime

**Application**

- Dati che devono essere processati da un' applicazione prima di essere "visibili"
- Esempi di sottotipi :  
mword, octet-stream

<https://tools.ietf.org/html/rfc1521>

[Elenco dei più comuni MIME \(Multipurpose Internet Mail Extensions\) types associati alle corrispondenti estensioni dei file.](#)

Fausto Marcontoni    Chapter 2 APPLICATION PROTOCOL    2.154

154

**Reti di Elaboratori** **Tipo Multipart**

---

```

From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=98766789
--98766789
Content-Transfer-Encoding: quoted-printable
Content-Type: text/plain

Dear Bob,
Please find a picture of a crepe.
--98766789
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

base64 encoded data .....
.....base64 encoded data
--98766789--

```

- E-mail contenenti più oggetti
- Boundary character: delimitano i messaggi
- per ogni oggetto
  - Content-Transfer-Encoding
  - Content-Type

**Fausto Marcantoni**
Chapter 2 APPLICATION PROTOCOL
2.155

155

**Reti di Elaboratori** **Messaggio ricevuto**

---

```

Received: from hamburger.edu by sushi.ij 6 Oct 2003
Received: from crepes.fr by hamburger.edu; 6 Oct 2003
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

```

- **Received indica i Mail Server che hanno recapitato il messaggio**
- **Più linee "Received" se il messaggio è stato inoltrato da più server SMTP lungo il percorso da mittente a destinatario**

**Fausto Marcantoni**
Chapter 2 APPLICATION PROTOCOL
2.156

156

**Reti di Elaboratori** **Esempio opzioni messaggio**

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL **2.157**

157

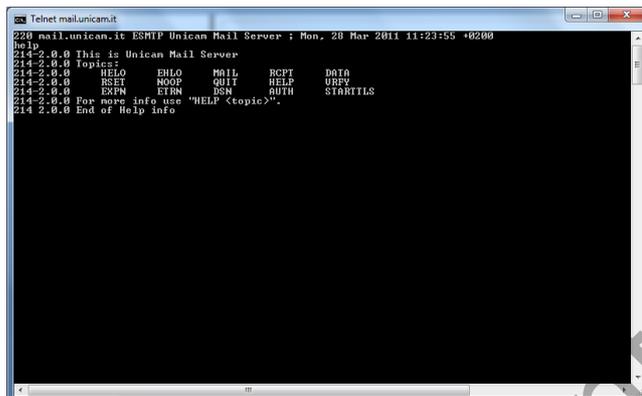
**Reti di Elaboratori** **Gmail - Opzioni - Mostra originale**

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL **2.158**

158

## Esempio mail

telnet 90.147.42.137 25

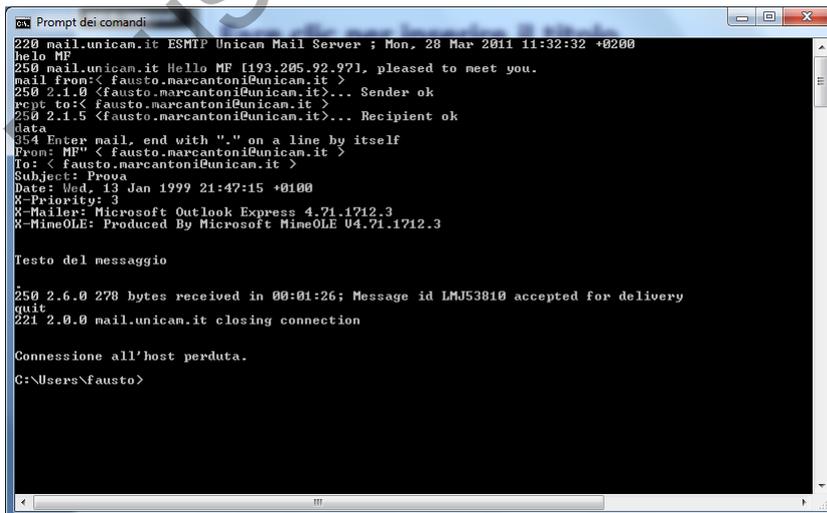


```

telnet mail.unican.it
220 mail.unican.it ESMTP Unican Mail Server ; Mon, 28 Mar 2011 11:23:55 +0200
help
214 2.0.0 This is Unican Mail Server
214 2.0.0 Topics:
214 2.0.0      HELO      EHLO      MAIL      RCPT      DATA
214 2.0.0      RSET      NOOP      QUIT      HELP      VRFY
214 2.0.0      EXPN      ETRN      DSN      AUTH      STARTTLS
214 2.0.0 For more info use "HELP <topic>".
214 2.0.0 End of Help info
  
```

159

## Esempio mail



```

Prompt dei comandi
220 mail.unican.it ESMTP Unican Mail Server ; Mon, 28 Mar 2011 11:32:32 +0200
helo MF
250 mail.unican.it Hello MF [193.205.92.97], pleased to meet you.
mail from: <fausto.marcantoni@unican.it >
250 2.1.0 <fausto.marcantoni@unican.it>... Sender ok
rcpt to: <fausto.marcantoni@unican.it >
250 2.1.5 <fausto.marcantoni@unican.it>... Recipient ok
data
354 Enter mail, end with "." on a line by itself
From: MF" <fausto.marcantoni@unican.it >
To: <fausto.marcantoni@unican.it >
Subject: Prova
Date: Wed, 13 Jan 1999 21:47:15 +0100
X-Priority: 3
X-Mailer: Microsoft Outlook Express 4.71.1712.3
X-MimeOLE: Produced By Microsoft MimeOLE 04.71.1712.3

Testo del messaggio

250 2.6.0 278 bytes received in 00:01:26; Message id LMJ53810 accepted for delivery
quit
221 2.0.0 mail.unican.it closing connection

Connessione all'host perduta.
C:\Users\fausto>
  
```

160

Reti di Elaboratori

## Fake mail

```
telnet mail.unicam.it 25
220 mail.unicam.it ESMTP Unicam Mail Server ; Mon, 28 Mar 2011 11:32:32 +0200
helo MF
250 mail.unicam.it Hello MF [193.205.92.97] pleased to meet you.
mail from:< fausto.marcantoni@unicam.it >
250 2.1.0 <fausto.marcantoni@unicam.it>... Sender ok
rcpt to:< fausto.marcantoni@unicam.it >
250 2.1.5 <fausto.marcantoni@unicam.it>... Recipient ok
data
354 Enter mail, end with "." on a line by itself
From: MF" < fausto.marcantoni@unicam.it >
To: < fausto.marcantoni@unicam.it >
Subject: Prova
Date: Mon, 28 Mar 2011 11:32:32 +0200
X-Priority: 3
X-Mailer: Microsoft Outlook Express 4.71.1712.3
X-MimeOLE: Produced By Microsoft MimeOLE V4.71.1712.3

Testo del messaggio
.
250 2.6.0 278 bytes received in 00:01:26; Message id LMJ53810 accepted for delivery
```

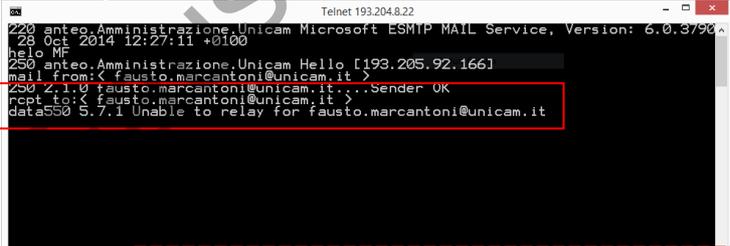


Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.161

161

Reti di Elaboratori

## Mail relay



Il modulo SMTP è stato concepito per filtrare la posta in base a determinati criteri specificati dall'utente e poi inviarla verso la destinazione desiderata. Un problema che si può verificare durante questo processo è di evitare utilizzi indesiderati da parte di estranei. Al fine di evitare che utenti esterni utilizzino il modulo SMTP per spedire messaggi indirizzati ad altri utenti esterni, è necessario configurare correttamente il modulo SMTP. L'utilizzo improprio dei **relay di posta** è il maggior veicolo di diffusione dello spam, dove il mittente cerca di nascondere la propria identità. Occorre configurare correttamente i parametri "Host Locali" e "Relay Domain".

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.162

162

**Reti di Elaboratori**

## smtp <-> wireshark

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL **2.163**

163

**Reti di Elaboratori**

## ESMTP command

**Essential SMTP commands in the order they may be used**

- HELO/EHLO
- MAIL FROM
- RCPT TO
- DATA
- NOOP
- HELP
- VRFY and EXPN
- RSET
- QUIT

Extended SMTP commands that some SMTP servers may support

- STARTTLS
- AUTH
- ATRN
- BDAT
- ETRN

<https://mailtrap.io/blog/smtp-commands-and-responses/>

<https://www.samlogic.net/articles/smtp-commands-reference.htm>

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL **2.164**

164

Reti di Elaboratori

## Fake Mailer



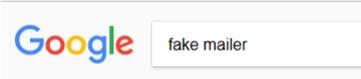
<https://emkei.cz/>



<http://anonymailer.net/>



<https://code.google.com/p/mailsend/>  
<https://github.com/muquit/mailsend>



Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.165

165

Reti di Elaboratori

## Test <http://anonymailer.net>

test zorro Posta in arrivo x

zorro <zorro@yyyy.com>  
a me

Inglese Italiano Traduci messaggio

-----  
 This email was sent via Anonymous email service for free.  
[YOU CAN REMOVE THIS TEXT MESSAGE BY BEING A PAID MEMBER FOR \\$19/year.](#)  
[CLICK HERE =>](#)  
 IP address of the sender:193.205.92.194 Message ID= 183555  
 -----

test zorro

-----  
 This email was sent via Anonymous email service for free.  
[YOU CAN REMOVE THIS TEXT MESSAGE BY BEING A PAID MEMBER FOR \\$19/year.](#)  
[CLICK HERE =>](#)  
 Message ID= 183555  
 -----

Rispondi Inoltra

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.166

166

Reti di Elaboratori

## Protocolli di accesso alla posta

- Soluzione tradizionale: utente legge direttamente la posta sul Mail Server (shell, telnet/ssh+shell, ...) **archeologia**
- L'host su cui è disposto il Mail Server deve essere sempre attivo
- Agenti di posta permettono di trasferire la posta dal Mail Server all'host locale del ricevente
- Possibilità visualizzare file multimediali e di specifiche applicazioni
- Occorre un protocollo "Push" per accedere alla Mailbox collocata sul Mail Server

Fausto Marcontoni Chapter 2 APPLICATION PROTOCOL 2.167

167

Reti di Elaboratori

## Protocolli di accesso alla posta

SMTP: consegna al/memorizza nel server di posta del ricevente  
 Protocollo di accesso: recupero della posta dal server locale

- POP: Post Office Protocol [RFC 5034] <https://tools.ietf.org/html/rfc5034>
  - Autenticazione (agent <-->server) e scaricamento
- IMAP: Internet Mail Access Protocol [RFC 3501] <https://tools.ietf.org/html/rfc3501>
  - Più possibilità (più complesso)
  - Manipolazione dei messaggi memorizzati sul server
- HTTP: Gmail, Hotmail, Yahoo! Mail, Libero, ecc.

Fausto Marcontoni Chapter 2 APPLICATION PROTOCOL 2.168

168

**Reti di Elaboratori**

## POP3 – IMAP : porte usate

By default, the **POP3** protocol works on two ports:

- 110** - this is the default POP3 non-encrypted port
- 995** - this is the port you need to use if you want to connect using POP3 securely

By default, the **IMAP** protocol works on two ports:

- 143** - this is the default IMAP non-encrypted port
- 993** - this is the port you need to use if you want to connect using IMAP securely

Type	Dest Host	Dest Port	Local Port
TCP	smtp.gmail.com	465	465
TCP	pop.gmail.com	995	995
TCP	imap.gmail.com	993	993

Secure Services Preferences Router

Server Address: smtp.gmail.com  
Port: 587

General | Outgoing Server | Connection | Advanced

Server Port Numbers

Incoming server (POP3): 996 Use Defaults

This server requires an encrypted connection (SSL)

Outgoing server (SMTP): 588

Use the following type of encrypted connection: Auto

```

[ gmail-pop3 ]
Client = yes
Accept = 127.0.0.1:110
Connect = pop.gmail.com:995

[ gmail-imap ]
Client = yes
Accept = 127.0.0.1:143
Connect = imap.gmail.com:993

[ gmail-smtp ]
Client = yes
Accept = 127.0.0.1:587
Connect = smtp.gmail.com:465

```

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL 2.169

169

**Reti di Elaboratori**

## Protocollo POP3

**Fase di autorizzazione**

- Comandi del client:
  - user**: nome utente
  - pass**: password
- Risposte del server
  - +OK**
  - ERR**

**Fase di transazione, client:**

- list**: lista numeri e dim. msg
- retr**: scarica messaggio in base al numero
- dele**: cancella
- quit**

```

S: +OK POP3 server ready
C: user alice
S: +OK
C: pass hungry
S: +OK user successfully logged on

C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off

```

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL 2.170

170

Reti di Elaboratori	<h2 style="color: blue;">Il protocollo IMAP</h2>	
<p><b>IMAP</b>  <i><b>Internet Message Access Protocol</b></i>  <i><b>Interactive Mail Access Protocol</b></i></p> <p>è un protocollo di comunicazione per la ricezione di e-mail.</p> <p>Il significato "Interactive Mail Access Protocol" è stato valido fino alla versione 3, dalla quarta in poi è cambiato in "Internet Message Access Protocol"</p> <p>Il protocollo è stato inventato da Mark Crispin nel 1986 come alternativa più moderna all'utilizatissimo POP.</p>		
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL	2.171

171

Reti di Elaboratori	<h2 style="color: blue;">Il protocollo IMAP</h2>	
<p><b>Accesso alla posta sia online che off-line</b>          Con POP3 il client si connette, scarica le email sul computer, le cancella dal server e poi si disconnette. Con l'IMAP il client rimane connesso e sostanzialmente si limita a leggere il contenuto del server senza trasferire nulla; questo permette di risparmiare tempo se ci sono messaggi di grandi dimensioni.</p> <p><b>Più utenti possono utilizzare la stessa casella di posta</b>          Il protocollo POP assume che un solo client (utente) sia connesso ad una determinata mailbox (casella di posta), quella che gli è stata assegnata. Al contrario l'IMAP permette connessioni simultanee alla stessa mailbox, fornendo meccanismi per controllare i cambiamenti apportati da ogni utente.</p> <p><b>Accesso a molteplici caselle di posta sul server</b>          Alcuni utenti, con il protocollo IMAP, possono creare, modificare o cancellare mailbox (di solito associate a cartelle) sul server. Inoltre, questa gestione delle mailbox, permette di avere cartelle condivise tra utenti diversi.</p> <p><b>Possibilità di fare ricerche sul server</b>          L'IMAP permette al client di chiedere al server quali messaggi soddisfano un certo criterio, per fare, per esempio, delle ricerche sui messaggi senza doverli scaricare tutti.</p> <p><b>Password criptate</b>          Con il protocollo POP le password vengono solitamente inviate in testo, rendendo facile, con una intercettazione, l'individuazione della password. Con l'IMAP è possibile criptare la password.</p>		
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL	2.172

172

Reti di Elaboratori

## Protocollo POP3 e IMAP

2 modalità di utilizzo

- Scarica ed elimina:
  1. User Agent **elimina** la posta dalla Mailbox dopo averla scaricata
  2. Un utente **disperde** la posta sui diversi host da cui accede la Mailbox
  3. User Agent permette di creare cartelle, spostare messaggi, effettuare ricerche nei messaggi
- Scarica e conserva
  1. User Agent **conserva** la posta sulla Mailbox
  2. Utente può **leggere i messaggi da macchine diverse**
  - Permette di gestire cartelle di posta remote come se fossero locali
  - IMAP deve mantenere una gerarchia di cartelle per ogni utente

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.173

173

Reti di Elaboratori

## Vantaggi di IMAP

**Vantaggi di IMAP**

- **Sicurezza.** I messaggi stanno sul server per cui non rischiano di essere persi se il computer si dovesse rompere.
- **Semplicità.** Se l'utente cambia il computer o cambia client di posta elettronica, non è necessario trasferire tutte le email sul nuovo computer o sul nuovo programma.
- **Velocità.** Dato che le email non devono essere trasferite dal server al singolo computer, la velocità è massima anche in presenza di messaggi con allegati di grosse dimensioni. Il server IMAP inoltre scarica in continuo le email su di lui e tutti i client accedono alle email in rete locale.
- **Accessibilità.** Se di solito scaricate le email sul vostro computer in ufficio e, da casa o da fuori ufficio, dovete controllare un'email ricevuta, dovete per forza recarvi in ufficio. Con un server IMAP invece potete accedere alla vostra casella di posta da qualsiasi computer connesso a internet senza nessun problema e con la massima velocità.

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.174

174

Reti di Elaboratori

## Esercitazione

internet reti sicurezza  
Esercitazioni

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.175

175

Reti di Elaboratori

## Protocolli di livello applicativo

- ✓ **FTP: file transfer protocol**
- ✓ **TFTP: Trivial File Transfer Protocol**
- ✓ **Posta elettronica**
  - **smtp**
  - **pop3 - imap**
- ✓ **DNS: Domain Name System**
- ✓ **SNMP: Simple Network Management Protocol**
- ✓ **Remote login: telnet - ssh**

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.176

176

Reti di Elaboratori	<h2 style="margin: 0;">DNS: Domain Name System</h2>
<p><b>Come identifico le persone:</b></p> <ul style="list-style-type: none"> <li>▪ molte mezzi di identificazione: CF, nome, # Passaporto</li> </ul> <p><b>Come identifico computer, host, router, ... :</b></p> <ul style="list-style-type: none"> <li>▪ Indirizzi IP (32 bit) – usati per indirizzare i datagrammi IP</li> <li>▪ "Nome", es., dida.cs.unicam.it – usati dagli utenti</li> </ul> <p><b>Risposta:</b> corrispondenza tra indirizzo IP e nome?</p>	
<p><b>Domain Name System</b></p> <ul style="list-style-type: none"> <li>▪ <b>Database distribuito</b> implementato come una gerarchia di molti <i>name server</i></li> <li>▪ <b>Protocollo applicativo</b> usato da host, router, name server per comunicare allo scopo di <b>risolvere</b> (tradurre) i nomi in indirizzi IP <ul style="list-style-type: none"> <li>▪ Funzione di base di Internet implementata come protocollo applicativo</li> <li>▪ La complessità trasferita al "bordo" della rete</li> </ul> </li> </ul>	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL
2.177	

177

Reti di Elaboratori	<h2 style="margin: 0;">Name server DNS</h2>
<p><b>Perché non un server DNS centralizzato?</b></p> <ul style="list-style-type: none"> <li>★ Minor tolleranza ai guasti</li> <li>★ Traffico eccessivo</li> <li>★ Database centrale troppo distante in molti casi</li> <li>★ Scarsa scalabilità!</li> </ul>	
<ul style="list-style-type: none"> <li>▪ Nessun <b>name server</b> contiene tutte le associazioni nome simbolico/indirizzo IP</li> </ul> <p><b>Name server locali :</b></p> <ul style="list-style-type: none"> <li>▪ Ogni ISP o compagnia ha un <b>name server locale (default)</b></li> <li>▪ La richiesta di traduzione (mapping) di un host è prima rivolta al name server locale</li> </ul> <p><b>Name server di riferimento (assoluto authoritative):</b></p> <ul style="list-style-type: none"> <li>▪ Per un <b>host</b>: per definizione è quello che è sempre in grado di eseguire la traduzione (mapping) nome simbolico/indirizzo IP dell' host</li> </ul>	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL
2.178	

178

Reti di Elaboratori

## DNS Primario - Secondario

Ogni dominio deve avere almeno due DNS Server per funzionare:  
**Primario (master)**  
**Secondario (slave)** e se fosse possibile un altro Secondario esterno alla rete del dominio che gestisce.

**Il Primario ha i dati effettivi**, mentre i **Secondari ne ha una copia** (Transfer Zone) ricevuta dal Primario.

Gli Amministratori di Rete, cambiano/aggiornano/modificano i dati solo nel DNS Primario, i Secondari verranno aggiornati automaticamente.

Ogni DNS contiene le informazioni relative al proprio dominio (zona) e sa come ottenere le informazioni di quelli sottostanti (in quanto ne ha delegato la gestione ad altri DNS, oppure li gestisce direttamente).

Quindi il DNS del dominio **.it** sa chi detiene le informazioni del dominio **libero.it** e chi detiene le informazioni del dominio **unicam.it** ma non sa chi detiene le informazioni per il dominio **mat.unicam.it**

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.179

179

Reti di Elaboratori

## Gerarchia di server DNS

```

graph TD
    Root[root DNS server] --- Com[DNS server com]
    Root --- Org[DNS server org]
    Root --- Edu[DNS server edu]
    Com --- Yahoo[DNS server di yahoo.com]
    Com --- Amazon[DNS server di amazon.com]
    Org --- Pbs[DNS server di pbs.org]
    Edu --- Poly[DNS server di poly.edu]
    Edu --- Umass[DNS server di umass.edu]
  
```

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.180

180

Reti di Elaboratori

## Root name server

I **root "name" servers** sono dei server DNS che hanno tutte le informazioni relative ai domini di 1 livello (top-level) ed indirizzano i vari DNS nelle traduzioni da nomi di domini in indirizzi IP e viceversa.

I DNS server di un dominio (zona) sono detti **autoritativi** per quel dominio, mentre risulteranno **non autoritativi** per tutti gli altri domini che non gestiscono direttamente.

I DNS del dominio **.it** non sono autoritativi per il dominio **libero.it**.  
 Ogni client di una rete ha installato automaticamente nel sistema operativo un **DNS client (resolver)** che manda le richieste ai DNS server configurati nelle proprietà di rete.

E' **FQDN (Full Qualified Domain Name)** un nome (di un host) comprensivo del dominio di appartenenza: **www.mat.unicam.it** è un FQDN, **www.libero.it** è un FQDN, **www.non** è un FQDN.

Il DNS server usa il protocollo **UDP ed ascolta sulla porta 53** per le richieste da parte dei client (che usano invece una porta alta, sulla quale risponderà il DNS), usa invece **protocollo TCP e porta 53** per i trasferimenti di zona con i DNS Secondari.

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.181

181

Reti di Elaboratori

## DNS: Root name servers

- Contattato dal name server locale che non riesce a risolvere un nome
- root name server:
  - Contatta il name server di riferimento (authoritative) se la traduzione non è nota
  - Ottiene la traduzione
  - Restituisce la traduzione al name server locale
- ~ una dozzina di root name server nel mondo

Verisign USC-ISI Cogent UMD NASA Ames ISC DISA DoD NIC ARL Netnod RIPE NCC ICANN WIDE



<http://www.root-servers.org/>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.182

182

Reti di Elaboratori

## Root Zone Database



Internet Assigned Numbers Authority

DOMAIN	TYPE	TLD MANAGER
.aaa	generic	American Automobile Association, Inc.
.aarp	generic	AARP
.abarth	generic	Fiat Chrysler Automobiles N.V.
.abb	generic	ABB Ltd
.abbott	generic	Abbott Laboratories, Inc.
.abbvie	generic	AbbVie Inc.
.abc	generic	Disney Enterprises, Inc.
.able	generic	Able Inc.
.abogado	generic	Minds + Machines Group Limited
.abudhabi	generic	Abu Dhabi Systems and Information Centre
.ac	country-code	Network Information Center (AC Domain Registry) c/o Cable and Wireless (Ascension Island)

<https://www.iana.org/domains/root/db>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.183

183

Reti di Elaboratori

## Root Zone Database - whois



Internet Assigned Numbers Authority

### IANA WHOIS Service

The IANA WHOIS Service is provided using the WHOIS protocol on port 43. This web gateway will query the query arguments are domain names, IP addresses and AS numbers.

<https://www.iana.org/whois>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.184

184

Reti di Elaboratori

## Whois - italia



ENMAGARE DEI DOMINI .IT

Trova il tuo .it   Gestisci il tuo .it   Valorizza il tuo .it   Il mondo di .it

Cerca il tuo .it   Come registrare   Scegli il nome

**RICERCA WHOIS**

Domínio:

Non sono un robot

<https://www.nic.it/it>  
<https://web-whois.nic.it/>

Fausto Marcantoni   Chapter 2 APPLICATION PROTOCOL   2.185

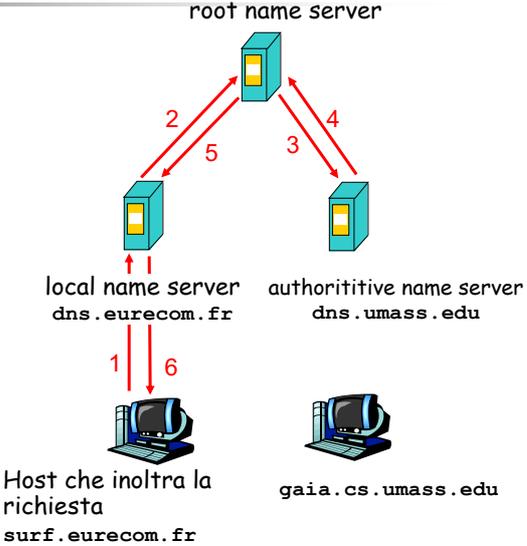
185

Reti di Elaboratori

## Esempio (1)

L' host `surf.eurecom.fr` vuole l'indirizzo IP di `gaia.cs.umass.edu`

1. Contatta il server DNS locale, `dns.eurecom.fr`
2. `dns.eurecom.fr` contatta il root name server, se necessario
3. Il root name server contatta il name server di riferimento, `dns.umass.edu`, se necessario



The diagram illustrates the DNS resolution process for the host `surf.eurecom.fr` to find the IP address of `gaia.cs.umass.edu`. It shows a hierarchy of servers: a local name server (`dns.eurecom.fr`), a root name server, and an authoritative name server (`dns.umass.edu`). The host sends a request (1) to the local name server. The local name server contacts the root name server (2). The root name server contacts the authoritative name server (3). The authoritative name server returns the IP address (4) to the root name server. The root name server returns the IP address (5) to the local name server. Finally, the local name server returns the IP address (6) to the host.

Host che inoltra la richiesta: `surf.eurecom.fr`

gaia.cs.umass.edu

Fausto Marcantoni   Chapter 2 APPLICATION PROTOCOL   2.186

186

Reti di Elaboratori

### Esempio (2)

**Root name server:**

- Può non essere a conoscenza di un name server di riferimento
- Può tuttavia conoscere un *name server intermedio* che contatta per raggiungere quello di riferimento

root name server

local name server  
dns.eurecom.fr

Name server intermedio  
dns.umass.edu

Name server di riferimento  
dns.cs.umass.edu

Host che inoltra la richiesta  
surf.eurecom.fr

gaia.cs.umass.edu

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.187

187

Reti di Elaboratori

### DNS: richieste ripetute (iterated queries)

**Richieste ricorsive (recursive query):**

- Trasferisce il carico della traduzione al name server contattato
- Carico eccessivo?

**Richieste ripetute (iterated query):**

- Il name server contattato risponde con l'indirizzo del prossimo name server da contattare
- "Non conosco questo nome, ma prova a rivolgerti a quest'altro server"

root name server

Name server locale  
dns.eurecom.fr

Name server intermedio  
dns.umass.edu

authoritative name server  
dns.cs.umass.edu

Host che inizia la richiesta  
surf.eurecom.fr

gaia.cs.umass.edu

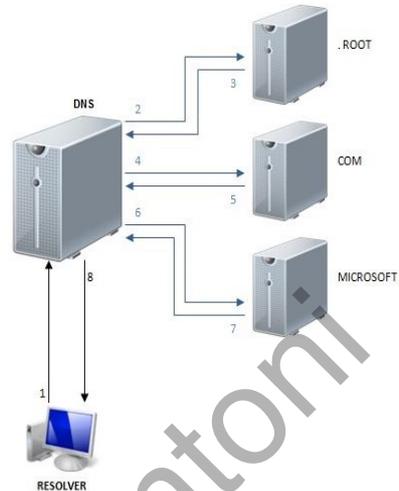
Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.188

188

## Esempio: query dns

Il resolver ha bisogno dell'indirizzo IP di microsoft.com

1. Il client contatta il DNS con una RECURSIVE QUERY : il DNS dovrà rispondere o con l'IP o con il messaggio di errore
2. Il DNS non trova il nome ne' nella propria cache ne' nelle proprie zone, quindi contatta il DNS autoritativo ROOT con una ITERATIVE QUERY
3. Il DNS non conosce microsoft.com quindi restituisce l'indirizzo del referente per il dominio COM
4. Il DNS contatta il server autoritativo per il dominio COM con una ITERATIVE QUERY
5. Il DNS autoritativo per COM non conosce microsoft.com quindi restituisce l'indirizzo del referente per il dominio Microsoft.com
6. Il DNS contatta il server autoritativo per il dominio MICROSOFT.COM con una ITERATIVE QUERY
7. Il DNS autoritativo per MICROSOFT.COM restituisce l'IP address richiesto.
8. Il DNS contattato dal resolver gli restituisce l'IP address richiesto.



## MODALITA' ITERATIVA - RICORSIVA

### MODALITA' ITERATIVA

Il server DNS invia al client DNS (Resolver) la risposta alla richiesta di risoluzione o l'indirizzo di un server che secondo lui è in grado di risolvere il nome

### MODALITA' RICORSIVA

Il client si aspetta dal server DNS la risposta alla sua richiesta. Il server DNS, se è responsabile del dominio, risolve l'indirizzo altrimenti trasmette la richiesta ad un server DNS di livello superiore e aspetta la risposta per il client

In ogni caso, il resolver genera un'interrogazione contenente:

- nome da risolvere
- dichiarazione della classe del nome
- tipo di risposta desiderata
- modalità

Reti di Elaboratori	<h2 style="color: blue; margin: 0;">DNS riepilogo</h2> <p>Quando un server DNS riceve un'interrogazione, <b>verifica se il nome è relativo alla propria zona di autorità:</b></p> <ol style="list-style-type: none"> <li>1. se sì, <b>converte il nome in un indirizzo</b> in base alle informazioni del proprio database, inserisce l'indirizzo nella risposta e la spedisce al client</li> <li>2. se non può risolvere completamente il nome, <b>esamina la modalità specificata nell'interrogazione:</b> <ol style="list-style-type: none"> <li>A. Se è <b>ricorsiva</b>, contatta un server di livello superiore che possa risolvere il nome, e riporta al client la risposta ottenuta</li> <li>B. Se è <b>iterativa</b>, si limita a comunicare al client il nome del server a cui rivolgersi</li> </ol> </li> </ol>
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL
2.191	

191

Reti di Elaboratori	<h2 style="color: blue; margin: 0;">DNS: caching e aggiornamento</h2> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <ul style="list-style-type: none"> <li>■ Quando un qualsiasi name server apprende una traduzione la memorizza localmente (<b>caching</b>) <ul style="list-style-type: none"> <li>■ Le traduzioni memorizzate nella cache (<b>cache entries</b>) scadono (<b>timeout</b>) dopo un certo tempo (<b>di solito un paio di giorni</b>)</li> </ul> </li> <li>■ Se possibile, richieste successive vengono servite usando la traduzione presente in cache</li> <li>■ I meccanismi di aggiornamento/modifica in studio da parte dell' IETF <ul style="list-style-type: none"> <li>■ RFC 2136</li> <li>■ <a href="http://www.ietf.org/html.charters/dnsind-charter.html">http://www.ietf.org/html.charters/dnsind-charter.html</a></li> </ul> </li> </ul> </div>
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL
2.192	

192

Reti di Elaboratori

## DNS Propagation Checker

Global DNS Propagation Checker

Global DNS Propagation Checker

www.unicam.it A Search

	New York, United States Internap	• 94.177.192.171	✓
	Los Angeles, United States NetScoutConnect		✗
	San Francisco, United States Digital Ocean		✗
	Jacksonville, United States Florida Technology Managed Services	• 94.177.192.171	✓
	Charlotte, United States Time Warner Cable		✗
	Toronto, Canada Cogeco Peer 1	• 94.177.192.171	✓
	London, United Kingdom Exponential-e		✗
	Paris, France SFR		✗
	Berlin, Germany Verizon Deutschland GmbH		✗
	Rome, Italy Irpina Net-Com SRL		✗
	Madrid, Spain Escuelas De Estudios Superiores Esic sacerdotes de	• 94.177.192.171	✓

<https://www.whatsmydns.net>

<https://www.gdnspc.com/>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.193

193

Reti di Elaboratori

## cache DNS

Come è possibile visualizzare il contenuto della cache DNS?

`ipconfig /displaydns`

Come è possibile cancellare il contenuto della cache DNS?

`ipconfig /flushdns`

`rndc`  
`resolvectl status`

`sudo dscacheutil -flushcache;sudo killall -HUP mDNSResponder`

La cache DNS è mantenuta dal servizio **mDNSResponder**.  
Se il tuo macOS è una versione molto recente, potrebbe non mostrare tutte le voci DNS in modo dettagliato, poiché alcune informazioni sono gestite internamente dal sistema.

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.194

194

Reti di Elaboratori

## Recursive/Iterative Queries in DNS



### Recursive/Iterative Queries in DNS

[https://media.pearsoncmg.com/curriculum/intl/it/lab/9788891902559\\_KUROSE\\_BOSS/studente/interactive%20animations/recursive-iterative-queries-in-dns/index.html](https://media.pearsoncmg.com/curriculum/intl/it/lab/9788891902559_KUROSE_BOSS/studente/interactive%20animations/recursive-iterative-queries-in-dns/index.html)

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.195

195

Reti di Elaboratori

## Record DNS

**DNS:** database distribuito che memorizza Resource Record (RR)

**Formato RR:** (nome, valore, tipo, ttl)

<ul style="list-style-type: none"><li>■ Tipo=A<ul style="list-style-type: none"><li>▪ nome è il nome dell' host</li><li>▪ valore è l' indirizzo IP</li></ul></li></ul>	<ul style="list-style-type: none"><li>■ Tipo=CNAME<ul style="list-style-type: none"><li>▪ nome è un alias di qualche nome reale ("canonico")</li><li>▪ valore è il nome canonico</li></ul></li></ul>
<ul style="list-style-type: none"><li>■ Tipo=NS<ul style="list-style-type: none"><li>▪ nome è il dominio (es. foo.com)</li><li>▪ valore è l'indirizzo IP del name server di riferimento per questo dominio</li></ul></li></ul>	<ul style="list-style-type: none"><li>■ Tipo=MX<ul style="list-style-type: none"><li>▪ valore è il nome di un mailserver associato a nome</li></ul></li></ul>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.196

196

Reti di Elaboratori

## Record DNS – Esempio Master Zone

Edit Master Zone 193.205.92



**PT**  
Reverse Address (18)



**NS**  
Name Server (1)



**CI**  
Name Alias (0)



**A MX  
NS PTR  
RR CN**  
All Record Types (19)

---



Edit Records File



**SC**  
Edit Zone Parameters



Edit Zone Options



**A host1  
A host2  
A host3  
A host4**  
Record Generators



<http://www.webmin.com/>

Fausto Marcantoni
Chapter 2 APPLICATION PROTOCOL
2.197

197

Reti di Elaboratori

## Record DNS – Esempio - All Record

All Records In 193.205.92

Name	Type	TTL	Values	Name	Type	TTL	Values
92.205.193.in-addr.arpa	NS	Default	nameserver.cs.unicam.it	48.92.205.193.in-addr.arpa	PTR	Default	carteraffaello.cs.unicam.it
1.92.205.193.in-addr.arpa	PTR	Default	nameserver.cs.unicam.it	54.92.205.193.in-addr.arpa	PTR	Default	cosy.cs.unicam.it
28.92.205.193.in-addr.arpa	PTR	Default	giasone.cs.unicam.it	53.92.205.193.in-addr.arpa	PTR	Default	geronacces.cs.unicam.it
33.92.205.193.in-addr.arpa	PTR	Default	geronaccess.cs.unicam.it	56.92.205.193.in-addr.arpa	PTR	Default	dida.cs.unicam.it
3.92.205.193.in-addr.arpa	PTR	Default	mail.cs.unicam.it	55.92.205.193.in-addr.arpa	PTR	Default	sicetsinpiciter.cs.unicam.it
28.92.205.193.in-addr.arpa	PTR	Default	cv.s.cs.unicam.it	59.92.205.193.in-addr.arpa	PTR	Default	shib.cs.unicam.it
5.92.205.193.in-addr.arpa	PTR	Default	www.cs.unicam.it	60.92.205.193.in-addr.arpa	PTR	Default	guernica.cs.unicam.it
32.92.205.193.in-addr.arpa	PTR	Default	ibfit.cs.unicam.it	42.92.205.193.in-addr.arpa	PTR	Default	usr.cs.unicam.it
5.92.205.193.in-addr.arpa	PTR	Default	intranet.cs.unicam.it	22.92.205.193.in-addr.arpa	PTR	Default	iceproject.cs.unicam.it
46.92.205.193.in-addr.arpa	PTR	Default	sibilla.cs.unicam.it				

Fausto Marcantoni
Chapter 2 APPLICATION PROTOCOL
2.198

198

Reti di Elaboratori

## Record DNS – Esempio – Reverse Address Record

Reverse Address Records

In 193.205.92

Add Reverse Address Record

Address: 193.205.92 Time-To-Live:  Default  seconds

Hostname:

Update forward?  Yes  No Create

Address	TTL	Hostname	Address	TTL	Hostname
193.205.92.1	Default	nameserver.cs.unicam.it.	193.205.92.48	Default	carteraffaello.cs.unicam.it.
193.205.92.28	Default	giasone.cs.unicam.it.	193.205.92.54	Default	cosy.cs.unicam.it.
193.205.92.33	Default	geronaccess.cs.unicam.it.	193.205.92.53	Default	geronacces.cs.unicam.it.
193.205.92.3	Default	mail.cs.unicam.it.	193.205.92.56	Default	dida.cs.unicam.it.
193.205.92.28	Default	cvs.cs.unicam.it.	193.205.92.55	Default	sicetsinpliciter.cs.unicam.it.
193.205.92.5	Default	www.cs.unicam.it.	193.205.92.59	Default	shib.cs.unicam.it.
193.205.92.32	Default	ibfit.cs.unicam.it.	193.205.92.60	Default	guernica.cs.unicam.it.
193.205.92.5	Default	intranet.cs.unicam.it.	193.205.92.42	Default	usr.cs.unicam.it.
193.205.92.46	Default	sibilla.cs.unicam.it.	193.205.92.22	Default	iceproject.cs.unicam.it.

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.199

199

Reti di Elaboratori

## Protocollo DNS, messaggi (1/2)

**Protocollo DNS**: messaggi di *richiesta (query)* e *risposta (reply)*, *client-server*

Header di messaggio

- **identification**: numero a 16 bit per la richiesta, la risposta usa lo stesso numero
- **flags**:
  - Richiesta o risposta
  - Bit di competenza
  - Richiesta di ricorsione (Q)
  - Ricorsione disponibile (R)
  - Il server che risponde è di riferimento per la richiesta (R)

**Nota**: richiesta e risposta hanno lo stesso formato

Identificazione	Flag	
Numero di domande	Numero di RR di risposta	-12 byte
Numero di RR autoritativi	Numero di RR addizionali	
Sezione delle domande (numero variabile di domande)		-Campi di nome e tipo di una query
Sezione delle risposte (numero variabile di record di risorsa)		-RR in risposta alla query
Sezione autoritativa (numero variabile di record di risorsa)		-Record per i server autoritativi
Sezione aggiuntiva (numero variabile di record di risorsa)		-Informazione aggiuntiva "d'aiuto" che può essere usata

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.200

200

**Reti di Elaboratori**

## Protocollo DNS, messaggi (2/2)

Nome, campi tipo per una richiesta

RR in risposta a una richiesta

Record per server di riferimento

Informazioni aggiuntive  
Es.: RR di tipo A contenente indirizzo IP di un mail server il cui nome canonico è contenuto nella answer section

Identificazione	Flag	
Numero di domande	Numero di RR di risposta	12 byte
Numero di RR autoritativi	Numero di RR addizionali	
Sezione delle domande (numero variabile di domande)		Campi di nome e tipo di una query
Sezione delle risposte (numero variabile di record di risorsa)		RR in risposta alla query
Sezione autoritativa (numero variabile di record di risorsa)		Record per i server autoritativi
Sezione aggiuntiva (numero variabile di record di risorsa)		Informazione aggiuntiva "d'aiuto" che può essere usata

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL **2.201**

201

**Reti di Elaboratori**

## DNS – Esempio query

```

61 8.687688 193.205.92.161 193.205.92.4 DNS standard query A www.unina.it
62 8.700584 193.205.92.4 193.205.92.161 DNS standard query response A 143.225.172.36
63 8.702404 193.205.92.161 143.225.172.36 TCP 2599 > http [SYN] Seq=0 Ack=0 Win=65535 Len=0
64 8.856788 3comEuro_04:09:69 Spanning-tree-(for STP RST. Root = 32768/00:06:53:11:80:00 Cost =
65 9.084314 143.225.172.36 193.205.92.161 TCP http > 2599 [SYN, ACK] Seq=0 Ack=1 Win=1651
66 9.084363 193.205.92.161 143.225.172.36 TCP 2599 > http [ACK] Seq=1 Ack=1 Win=65535 [C
67 9.084547 193.205.92.161 143.225.172.36 HTTP GET / HTTP/1.1
...
Frame 61 (72 bytes on wire, 72 bytes captured)
Ethernet II, Src: 00:0f:b0:44:69:10, Dst: 00:0a:5e:02:83:f2
Internet Protocol, Src Addr: 193.205.92.161 (193.205.92.161), Dst Addr: 193.205.92.4 (193.205.92.4)
User Datagram Protocol, Src Port: 1988 (1988), Dst Port: domain (53)
Domain Name System (query)
Transaction ID: 0xce9b
Flags: 0x0100 (Standard query)
0... .. = Response: Message is a query
.000 0... .. = Opcode: Standard query (0)
... .. = Truncated: Message is not truncated
... ..1 .. = Recursion desired: Do query recursively
... ..0 .. = z: reserved (0)
... ..0 .. = Non-authenticated data OK: Non-authenticated data is unacceptable
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
www.unina.it: type A, class inet
Name: www.unina.it
Type: Host address
Class: inet
0000 00 0a 5e 02 83 f2 00 0f b0 44 69 10 08 00 45 00 ..A...D1...E.
0010 00 3a 42 6f 00 00 80 11 bc 03 c1 cd 5c a1 c1 cd ..:Bo... \...
0020 5c 04 07 c4 00 35 00 26 18 32 ce 9b 01 00 00 01 \...5.&.2....
0030 00 00 00 00 00 03 77 77 77 05 75 6e 69 6e 61 .....w ww.unina
0040 02 69 74 00 00 01 00 01 .....it.....

```

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL **2.202**

202

Reti di Elaboratori

## DNS – Esempio response

```

66 8.702404 193.205.92.161 193.205.92.161 DNS Standard query response for 143.225.172.36
63 8.702404 193.205.92.161 143.225.172.36 TCP 2599 > http [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=
64 8.856788 3comEuro_04:09:69 Spanning-tree-(for STP RST. Root = 32768/00:06:53:11:80:00 cost = 4 Por
65 9.084314 143.225.172.36 193.205.92.161 TCP http > 2599 [SYN, ACK] Seq=0 Ack=1 Win=16560 Len=0
66 9.084363 193.205.92.161 143.225.172.36 TCP 2599 > http [ACK] Seq=1 Ack=1 Win=65535 [CHECKSUM
67 9.084547 193.205.92.161 143.225.172.36 HTTP GET / HTTP/1.1

Flags: 0x8180 (Standard query response, No error)
..... = Response: Message is a response
..... = opcode: standard query (0)
..... = Authoritative: Server is not an authority for domain
..... = Truncated: Message is not truncated
..... = Recursion desired: Do query recursively
..... = Recursion available: Server can do recursive queries
..... = 2: reserved (0)
..... = Answer authenticated: Answer/authority portion was not authenticated by the server
..... = Reply code: No error (0)

Questions: 1
Answer RRs: 1
Authority RRs: 2
Additional RRs: 2
Queries
  www.unina.it: type A, class inet
    name: www.unina.it
    Type: Host address
    Class: inet
Answers
  www.unina.it: type A, class inet, addr 143.225.172.36
Authoritative nameservers
  unina.it: type NS, class inet, ns dscna2.unina.it
    Name: unina.it
    Type: Authoritative name server
    Class: inet
    Time to live: 22 hours, 53 minutes, 38 seconds
    Data length: 9
    Name server: dscna2.unina.it
  unina.it: type NS, class inet, ns dscna1.unina.it
Additional records
0060 41 f2 00 09 06 64 73 63 6e 61 32 c0 10 c0 10 00 A...dsc na2....
0070 02 00 01 00 01 41 f2 00 09 06 64 73 63 6e 61 31 .....A...dscna1
0080 c0 10 c0 4f 00 01 00 01 00 00 0f fd 00 04 e0 85 ..f.o....
0090 fc 01 e0 3a 00 01 00 01 00 00 f3 69 00 04 e0 85 .....
00a0 fc 07

```

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.203

203

Reti di Elaboratori

## nslookup

- NSLookup (Name Server Lookup) è un programma che effettua ricerche all'interno dei database del DNS
- l'utente comune non ha alcun bisogno di effettuare questo tipo di ricerche, se non per curiosità, o per qualche caccia al tesoro informatica;



nslookup



nslookup



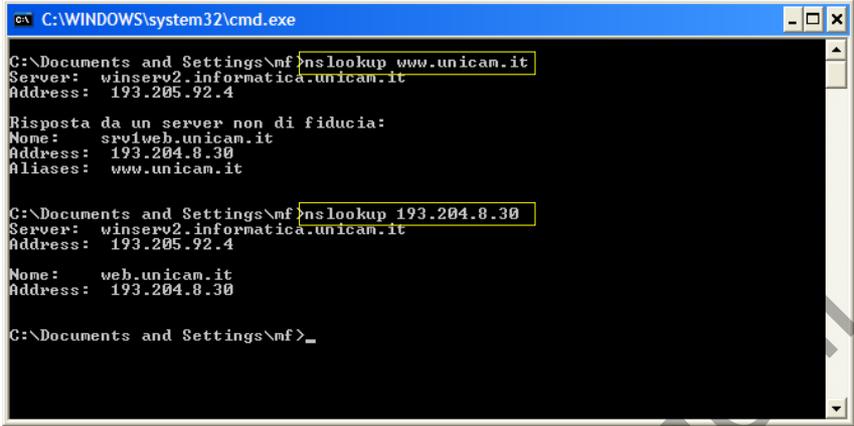
dig

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.204

204

Reti di Elaboratori

## nslookup - esempio



```

C:\WINDOWS\system32\cmd.exe

C:\Documents and Settings\mf>nslookup www.unican.it
Server: winserv2.informatica.unican.it
Address: 193.205.92.4

Risposta da un server non di fiducia:
Nome:     srv1web.unican.it
Address:  193.204.8.30
Aliases:  www.unican.it

C:\Documents and Settings\mf>nslookup 193.204.8.30
Server: winserv2.informatica.unican.it
Address: 193.205.92.4

Nome:     web.unican.it
Address:  193.204.8.30

C:\Documents and Settings\mf>_

```

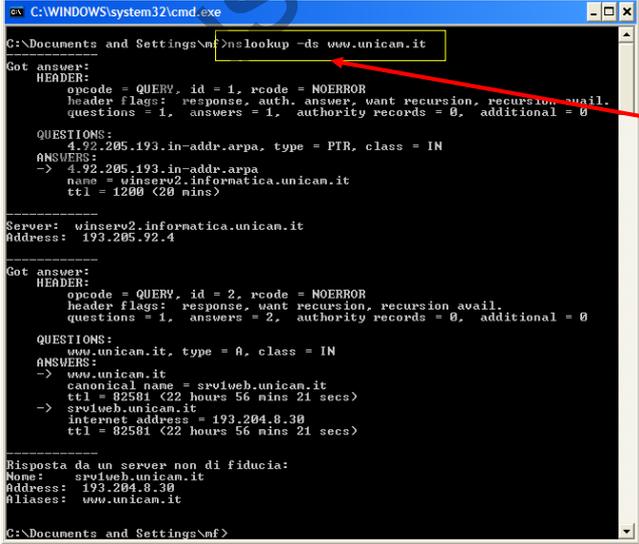
<https://technet.microsoft.com/it-it/library/cc645505.aspx>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.205

205

Reti di Elaboratori

## nslookup - esempio



```

C:\WINDOWS\system32\cmd.exe

C:\Documents and Settings\mf>nslookup -ds www.unican.it
Got answer:
HEADER:
opcode = QUERY, id = 1, rcode = NOERROR
header flags: response, auth, answer, want recursion, recursion avail.
questions = 1, answers = 1, authority records = 0, additional = 0

QUESTIONS:
  4.92.205.193.in-addr.arpa, type = PTR, class = IN
ANSWERS:
-> 4.92.205.193.in-addr.arpa
   name = winserv2.informatica.unican.it
   ttl = 1200 (20 mins)

Server: winserv2.informatica.unican.it
Address: 193.205.92.4

Got answer:
HEADER:
opcode = QUERY, id = 2, rcode = NOERROR
header flags: response, want recursion, recursion avail.
questions = 1, answers = 2, authority records = 0, additional = 0

QUESTIONS:
  www.unican.it, type = A, class = IN
ANSWERS:
-> www.unican.it
   canonical name = srv1web.unican.it
   ttl = 82581 (22 hours 56 mins 21 secs)
-> srv1web.unican.it
   internet address = 193.204.8.30
   ttl = 82581 (22 hours 56 mins 21 secs)

Risposta da un server non di fiducia:
Nome:     srv1web.unican.it
Address:  193.204.8.30
Aliases:  www.unican.it

C:\Documents and Settings\mf>

```

modalità debug (-ds)

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.206

206

Reti di Elaboratori

## dig - esempio

```

studente@server-IRS: ~
;; MSG SIZE rcvd: 299
studente@server-IRS: $ dig
; <<> DLG 9.18.1-1ubuntu1.2-Ubuntu <<>
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, ld: 63029
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 4
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags: udp: 65494
;; QUESTION SECTION:
;;      IN      NS
;;
;; ANSWER SECTION:
3146 IN NS d.root-servers.net.
3146 IN NS b.root-servers.net.
3146 IN NS a.root-servers.net.
3146 IN NS l.root-servers.net.
3146 IN NS k.root-servers.net.
3146 IN NS i.root-servers.net.
3146 IN NS f.root-servers.net.
3146 IN NS e.root-servers.net.
3146 IN NS h.root-servers.net.
3146 IN NS j.root-servers.net.
3146 IN NS n.root-servers.net.
3146 IN NS g.root-servers.net.
3146 IN NS c.root-servers.net.
;; ADDITIONAL SECTION:
a.root-servers.net. 3146 IN AAAA 2001:503:ba3e::2:30
a.root-servers.net. 3146 IN A 198.41.0.4
n.root-servers.net. 3146 IN A 202.12.27.33
;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Tue Oct 18 11:05:37 CEST 2022
;; MSG SIZE rcvd: 299
studente@server-IRS: $

```

man dig

Fausto Marcantoni
Chapter 2 APPLICATION PROTOCOL
2.207

207

Reti di Elaboratori

## Il file hosts (windows)

C:\Windows\System32\drivers\etc\hosts

Name

- hosts
- hoststics
- lmhosts.sam
- networks
- protocol
- services

Nome	Ultima modifica	Tipo	Dimensione
hosts	01/12/2015 12:34	File	1 KB
hoststics	23/03/2017 12:35	File iCalendar	1 KB

```

# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
# 102.54.94.97 rhino.acme.com # source server
# 38.25.63.10 x.acme.com # x client host
#
# localhost name resolution is handled within DNS itself.
# 127.0.0.1 localhost
# ::1 localhost
127.0.0.1 localhost

```

Fausto Marcantoni
Chapter 2 APPLICATION PROTOCOL
2.208

208

Reti di Elaboratori

## Il file hosts (linux)

/etc/hosts

```

docente@docente-webex:/etc
File Modifica Visualizza Cerca Terminale Aiuto
-----
----- 1 root root 661 16 dic 2016 gshadow
----- 1 root root 648 16 dic 2016 gshadow-
-rw-r--r-- 1 root root 801 19 lug 2011 gssapi_mech.conf
drwxr-xr-x 2 root root 4096 16 dic 2016 gtk-2.0
drwxr-xr-x 3 root root 4096 16 dic 2016 hal
-rw-r--r-- 1 root root 9 21 mar 2017 host.conf
-rw-r--r-- 1 root root 158 12 gen 2010 hosts
-rw-r--r-- 1 root root 370 12 gen 2010 hosts.allow
-rw-r--r-- 1 root root 460 12 gen 2010 hosts.deny
drwxr-xr-x 2 root root 4096
drwxr-xr-x 4 root root 4096
-rw-r--r-- 1 root root 4850
drwxr-xr-x 2 root root 4096
-rw-r--r-- 1 root root 0
lrwxrwxrwx 1 root root 11
-rw-r--r-- 1 root root 884
-rw-r--r-- 1 root root 942
drwxr-xr-x 2 root root 4096
drwxr-xr-x 2 root root 4096
File Modifica Visualizza Cerca Terminale Aiuto
-----
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

```

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.209

209

Reti di Elaboratori

## Il file hosts

quando vogliamo aprire una determinata pagina web, prima viene controllato questo file, per vedere se l'host dell'URL è contenuto al suo interno e:

- se il file hosts contiene la corrispondenza hostname -> IP, allora viene usato questo indirizzo IP per aprire la pagina desiderata.
- se il file hosts non contiene l'hostname, allora viene contattato il server DNS a cui viene chiesto di risolvere il nome dell'host dell'URL in indirizzo IP.

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.210

210

Reti di Elaboratori

## approfondimenti

**DNS and BIND** Copertina flessibile – 20 giugno 2006  
 Edizione inglese | di Cricket Liu (Autore), Paul Albitz (Autore)  
 ★★★★★ 95 voti

Visualizza tutti i formati ed edizioni

Formato Kindle 26,61 €	Copertina flessibile <b>41,60 €</b>
---------------------------	--

Leggilo con la nostra App gratuita

1 Usato da 19,03 €  
6 Nuovo da 41,60 €

**Spedizione GRATUITA** con consegna presso punti di ritiro (se disponibile per il tuo ordine)

Questo articolo è acquistabile con il Bonus Cultura e con il Bonus Carta del Docente e spedito direttamente da Amazon. Sono esclusi prodotti di Venditori terzi sul Marketplace. Verifica i termini e condizioni dell'Iniziativa Bonus Cultura 18app e di Carta del Docente.

DNS and BIND tells you everything you need to work with one of the Internet's fundamental building blocks: the distributed host information database. It's the essential for troubleshooting common DNS problems.

[https://www.mrwebmaster.it/hosting/dns-come-funziona\\_11718.html](https://www.mrwebmaster.it/hosting/dns-come-funziona_11718.html)

<https://www.extraordy.com/dns-e-bind-concetti-e-configurazione-caching-dns/>

<https://www.apogeeonline.com/contrib/uploads/linux-server-amministratore-di-rete-capitolo4.pdf>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.211

211

Reti di Elaboratori

## Enumerate DNS

<https://dnsdumpster.com/>

<https://github.com/fwaeytens/dnsenum>

```
sudo apt install dnsenum
dnsenum --enum <domain>
```

<https://github.com/darkoperator/dnsrecon>

```
sudo apt install dnsrecon
dnsrecon -d <domain>
dnsrecon -d <domain> -t axfr
dnsrecon -r <ip_range>
```

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.212

212

Reti di Elaboratori

## Laboratorio

internet reti sicurezza  
Esercitazioni

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.213

213

Reti di Elaboratori

- ✓ **FTP: file transfer protocol**
- ✓ **TFTP: Trivial File Transfer Protocol**
- ✓ **Posta elettronica**
  - **smtp**
  - **pop3 - imap**
- ✓ **DNS: Domain Name System**
- ✓ **SNMP: Simple Network Management Protocol**
- ✓ **Remote login: telnet - ssh**

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.214

214

Reti di Elaboratori	<h2 style="margin: 0;">Simple Network Management Protocol</h2>	
<p style="text-align: right;">RFC 1157  <a href="https://datatracker.ietf.org/doc/html/rfc1157">https://datatracker.ietf.org/doc/html/rfc1157</a></p> <p>Gestione delle reti: SNMP</p> <ul style="list-style-type: none"> <li>✓ Simple Network Management Protocol</li> <li>✓ disponibile in praticamente tutti i dispositivi di rete maneggiabili, comprese le stampanti, i printer server, i fax server e, spesso in modo nascosto, anche i sistemi operativi.</li> <li>✓ Semantica del protocollo SNMP:             <ul style="list-style-type: none"> <li>➤ GET, GETNEXT ("WALK"), GETBULK, SET, TRAP</li> </ul> </li> </ul> <p>SNMP Agent</p> <p>SNMP Manager</p>		
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL	2.215

215

Reti di Elaboratori	<h2 style="margin: 0;">SNMP</h2>	
<p>SNMP è usato diffusamente per monitorare e configurare dispositivi di rete e anche computer o software</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> SNMP version 1, 2c e 3</li> <li><input type="checkbox"/> Uso di ACL</li> <li><input type="checkbox"/> Specificare SNMP per porta <span style="float: right;">UDP port 161</span></li> <li><input type="checkbox"/> Blocco di SNMP al firewall</li> </ul> <p>Ogni <b>stazione di gestione</b> o <b>agente</b> in una rete gestita da SNMP mantiene un database locale di informazioni rilevanti per la gestione della rete, noto come <b>Management Information Base (MIB)</b></p>		
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL	2.216

216

Reti di Elaboratori

## SNMP Agent - Station

**Modello client-server**

- SNMP Station: modulo che emette una richiesta
  - **Get-Request** per richiedere informazioni all'Agent
  - **Get-Next-Request** per richiedere righe successive di tabelle dell'Agent (ritorna errore a fine tabella)
  - **Set-Request** per configurare parametri dell'Agent
- SNMP Agent: modulo che formula la risposta
  - **Get-Response** per fornire informazioni alla Station
  - **Trap** per riportare un evento a una Station predefinita

### SNMP Architecture

Agent Device (Router, Switch etc.)      Internet      Intranet      SNMP Manager

MIB Database      SNMP Agent Software      NMS      SNMP Manager Software

SNMP Responses/Traps      SNMP Commands

Fausto Marcantoni      Chapter 2 APPLICATION PROTOCOL      2.217

217

Reti di Elaboratori

## Albero OID

- Ogni oggetto nel MIB ha un identificatore di oggetto (OID), che la stazione di gestione usa per richiedere il valore dell'oggetto dall'agente.
- Un OID è una **sequenza di numeri interi** che identifica univocamente un oggetto gestito definendo un percorso verso quell'oggetto attraverso una struttura ad albero chiamata **albero OID** o **albero di registrazione**.
- Quando un agente SNMP ha bisogno di accedere a uno specifico oggetto gestito, attraversa l'albero OID per trovare l'oggetto.

```

graph TD
    Root[Root] --> iso["iso (1)"]
    iso --> org["org (3)"]
    org --> dod["dod (6)"]
    dod --> Internet["Internet (1)"]
    Internet --> directory["directory (1)"]
    Internet --> mgmt["mgmt (2)"]
    Internet --> experimental["experimental (3)"]
    Internet --> private["private (4)"]
    directory --> mib2["mib-2 (1)"]
    mib2 --> system["system (1)"]
    mib2 --> interfaces["interfaces (2)"]
    mib2 --> ip["ip (4)"]
    experimental --> cisco["cisco (9)"]
    private --> enterprise["enterprise (1)"]
    enterprise --> microsoft["microsoft (311)"]
    enterprise --> juniperMIB["juniperMIB (2636)"]
  
```

OID Tree Example

Fausto Marcantoni      Chapter 2 APPLICATION PROTOCOL      2.218

218

**Reti di Elaboratori**

## identificatore di oggetto (OID)

Per ottenere il tempo di accensione del sistema di un dispositivo gestito, è possibile eseguire il polling di questo

**OID -1.3.6.1.6.3.10.2.1.3**

e l'apparato restituirà il numero di secondi dall'ultima volta che il motore SNMP è stato attivato.

```

sysDescr (1.3.6.1.2.1.1.1)
sysObjectID (1.3.6.1.2.1.1.2)
sysUpTime (1.3.6.1.2.1.1.3)
sysContact (1.3.6.1.2.1.1.4)
sysName (1.3.6.1.2.1.1.5)
sysLocation (1.3.6.1.2.1.1.6)
sysServices (1.3.6.1.2.1.1.7)

```

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL **2.219**

219

**Reti di Elaboratori**

## Tools SNMP

**ManageEngine**  
<https://www.manageengine.com/products/mibbrowser-free-tool/download-confirm.html>

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL **2.220**

220

Reti di Elaboratori

## Tools SNMP

<http://www.ireasoning.com/>

**About**  
iReasoning MIB Browser  
Personal Edition  
Powered by iReasoning SNMP API  
Version: 14.0 (Build 4716)  
<http://www.ireasoning.com>  
Copyright(c) 2002-2022 iDeskCenter Inc. All rights reserved.

iReasoning MIB Browser  
File Edit Operations Tools Bookmarks Help  
Address: 193.205.92.214 Advanced... OID: 1.3.6.1.2.1.2.2.1.1.5 Operations: Get Next Go

**SNMP MIBs**

MIB Tree

- iso org dod internet mgmt
  - system
  - interfaces
    - at
    - ip
    - icmp
    - tcp
    - udp
    - egp
    - transmission
    - snmp
    - host

**Result Table**

Name/OID	Value	Type	IP-Port
sysDescr.0	RouterOS RB4011GS+	OctetString	193.205...
sysObjectID.0	1.3.6.1.4.1.14988.1	OID	193.205...
sysUpTime.0	14 minutes 42 seconds...	TimeTicks	193.205...
sysContact.0	Fausto Marcantoni	OctetString	193.205...
sysName.0	RB - Corso Reti e Sicur.	OctetString	193.205...
sysLocation.0	Informatica - Internet Re.	OctetString	193.205...
sysServices.0	78	Integer	193.205...
ifNumber.0	15	Integer	193.205...
ifIndex.1	1	Integer	193.205...
ifIndex.2	2	Integer	193.205...
ifIndex.3	3	Integer	193.205...
ifIndex.4	4	Integer	193.205...
ifIndex.5	5	Integer	193.205...

Name  
OID  
MIB  
Syntax  
Access

iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifIndex.5

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL **2.221**

221

Reti di Elaboratori

## Come installare e configurare SNMP su Windows 10

Funzionalità facoltative

Aggiungi una funzionalità

Azioni più recenti

- Simple Network Management Protocol (SNMP) Installazione
- Provider SNMP WMI Installazione

Vedi la cronologia delle funzionalità facoltative

Come abilitare SNMP utilizzando PowerShell

Per abilitare SNMP tramite PowerShell, assicurati innanzitutto che il tuo computer abbia accesso a Internet. In tal caso, esegui un PowerShell con privilegi elevati premendo Win + X e selezionando Windows PowerShell (Admin) . Eseguire il seguente comando per installare i server SNMP dai server Microsoft:

```
Add-WindowsCapability -Online -Name "SNMP.Client---0.0.1.0"
```

<https://www.tecnobabile.com/come-installare-e-configurare-snm-p-su-windows-10/2021-08-27/>

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL **2.222**

222

Reti di Elaboratori

## snmp <-> wireshark

UDP

Request

OID

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.223

223

Reti di Elaboratori

## snmp <-> wireshark

Response

OID

valore

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.224

224

Reti di Elaboratori

- ✓ **FTP: file transfer protocol**
- ✓ **TFTP: Trivial File Transfer Protocol**
- ✓ **Posta elettronica**
  - **smtp**
  - **pop3 - imap**
- ✓ **DNS: Domain Name System**
- ✓ **SNMP: Simple Network Management Protocol**
- ✓ **Remote login: telnet - ssh**

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.225

225

Reti di Elaboratori

## Remote Login - telnet

Il protocollo **telnet** permette di accedere a un servizio interattivo con un server remoto in modalità di console a caratteri.

**Ciò significa che l'host client diventa un terminale remoto del server.**

Sulla console del client si apre una **shell di comandi** per il server.  
 Più precisamente, ad entrambi i capi della comunicazione, il protocollo telnet prevede istanze paritetiche di un **NVT (Network Virtual Terminal)**.

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.226

226

**Reti di Elaboratori**

## Remote Login - telnet

**telnetd**  
 È il daemon del servizio necessario per ricevere connessioni attraverso telnet.

TCP porta 23

Client e Server negoziano le opzioni del collegamento (es., ASCII a 7 bit o a 8 bit)

**sintassi URI (Uniform Resource Identifier) per telnet (RFC4248)**

```
telnet://<user>:<password>@<host>:<port>/
```

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL **2.227**

227

**Reti di Elaboratori**

## Wireshark <-> telnet

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL **2.228**

228

Reti di Elaboratori

## Wireshark <-> telnet

26 pacchetti clienti, 20 pacchetti server, 32 turni.  
Conversazione intera (216 bytes)

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.229

229

Reti di Elaboratori

## Remote Login - ssh

### Ssh (Secure Shell)

Consente di:

- connettersi ad un altro sistema sulla rete.
- eseguire comandi su un sistema remoto.
- muovere files da un sistema all'altro sulla rete.

Ssh deve rimpiazzare completamente i comandi "r" (rlogin, rsh, rcp)

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.230

230

Reti di Elaboratori	<h2 style="margin: 0;">Remote Login - ssh</h2>	
<p>Sshd è il daemon che sta in attesa per le connessioni.</p> <p>Normalmente resta in ascolto su una porta 22 TCP</p> <p>fornisce un canale criptato sicuro su di una rete insicura.</p> <p>Effettua:</p> <ul style="list-style-type: none"><li>✓ Autenticazione del server</li><li>✓ Scambio di chiavi</li><li>✓ Cifratura</li><li>✓ Protezione d'integrità</li></ul>		
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL	2.231

231

Reti di Elaboratori	<h2 style="margin: 0;">Remote Login - tools</h2>	
 <b>PuTTY</b>  <a href="https://www.putty.org/" style="color: red; text-decoration: underline;">https://www.putty.org/</a>	  <a href="https://winscp.net" style="color: red; text-decoration: underline;">https://winscp.net</a>	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL	2.232

232

**Reti di Elaboratori** **Connessione ssh**

The screenshot shows the PuTTY Configuration window on the left, with the SSH connection type selected. The main terminal window shows the connection to 10.0.32.2. A 'PuTTY Security Alert' dialog box is displayed, warning that the user is connecting to a host that has not been added to the known hosts file. The terminal output shows the login process for the user 'studente' on a Debian system.

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL **2.233**

233

**Reti di Elaboratori** **Wireshark <-> ssh**

The screenshot displays the Wireshark interface capturing an SSH session. The packet list pane shows several SSHv2 packets between mfausto.administraz... and 10.0.32.2. The packet details pane for the selected packet shows the SSH Protocol structure, including the Protocol (SSH-2.0-PuTTY\_Release\_0.67) and the direction (client-to-server).

Time	Source	Destination	Protocol	Length	Info
14.201737	mfausto.administraz...	10.0.32.2	SSHv2	82	Client: Protocol (SSH-2.0-PuTTY_Release_0.67)
14.215165	10.0.32.2	mfausto.administraz...	SSHv2	86	Server: Protocol (SSH-2.0-OpenSSH_8.4p1 Debian-5)
14.215498	mfausto.administraz...	10.0.32.2	SSHv2	726	Client: Key Exchange Init
14.217270	10.0.32.2	mfausto.administraz...	SSHv2	1110	Server: Key Exchange Init
14.217412	mfausto.administraz...	10.0.32.2	SSHv2	78	Client: Diffie-Hellman Group Exchange Request
14.239071	10.0.32.2	mfausto.administraz...	SSHv2	590	Server: Diffie-Hellman Group Exchange Group
14.334533	mfausto.administraz...	10.0.32.2	SSHv2	582	Client: Diffie-Hellman Group Exchange Init
14.357323	10.0.32.2	mfausto.administraz...	SSHv2	1414	Server: Diffie-Hellman Group Exchange Reply, New Keys
14.470260	mfausto.administraz...	10.0.32.2	SSHv2	70	Client: New Keys
14.470489	mfausto.administraz...	10.0.32.2	SSHv2	118	Client: Encrypted packet (len=64)
14.470897	10.0.32.2	mfausto.administraz...	SSHv2	118	Server: Encrypted packet (len=64)

**Fausto Marcantoni** Chapter 2 APPLICATION PROTOCOL **2.234**

234

Reti di Elaboratori

Modello client - server

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.235

235

Reti di Elaboratori

**Modello client - server**

Il server attende passivamente di essere contattato, mentre il client avvia il processo di comunicazione



Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.236

236

Reti di Elaboratori

## Modello client - server

■ Applicazione client

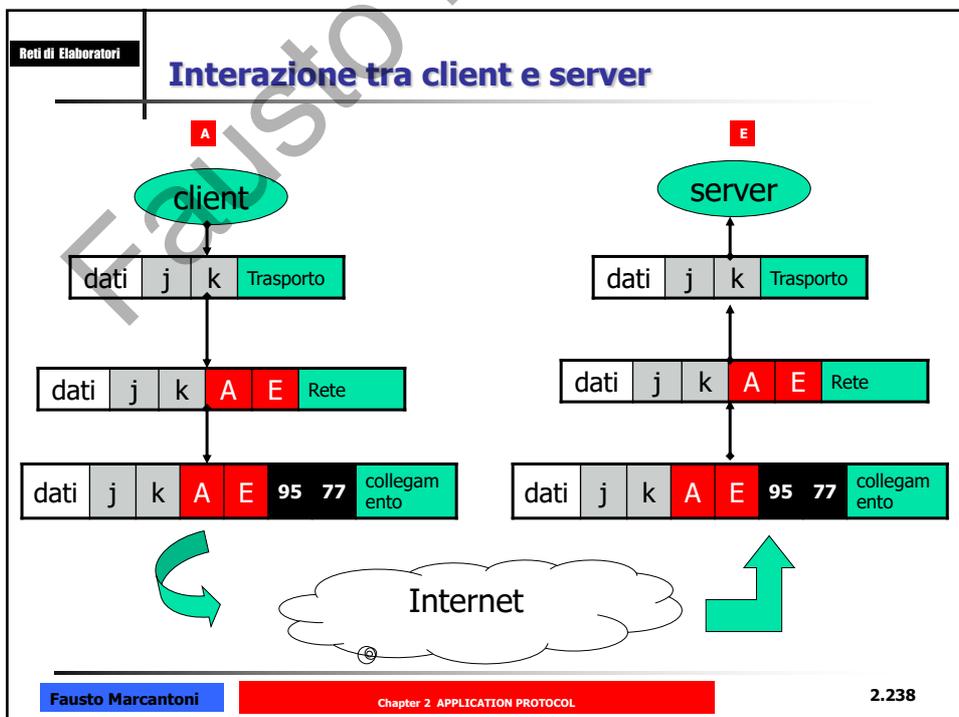
- Generico programma
  - Si comporta da client quando richiede l'uso della rete
  - Esegue anche elaborazioni in locale
- È invocata direttamente dall'utente
- È eseguita localmente sul computer dell'utente
- Contatta attivamente un solo server alla volta
- Non richiede dispositivi dedicati

■ Applicazione server

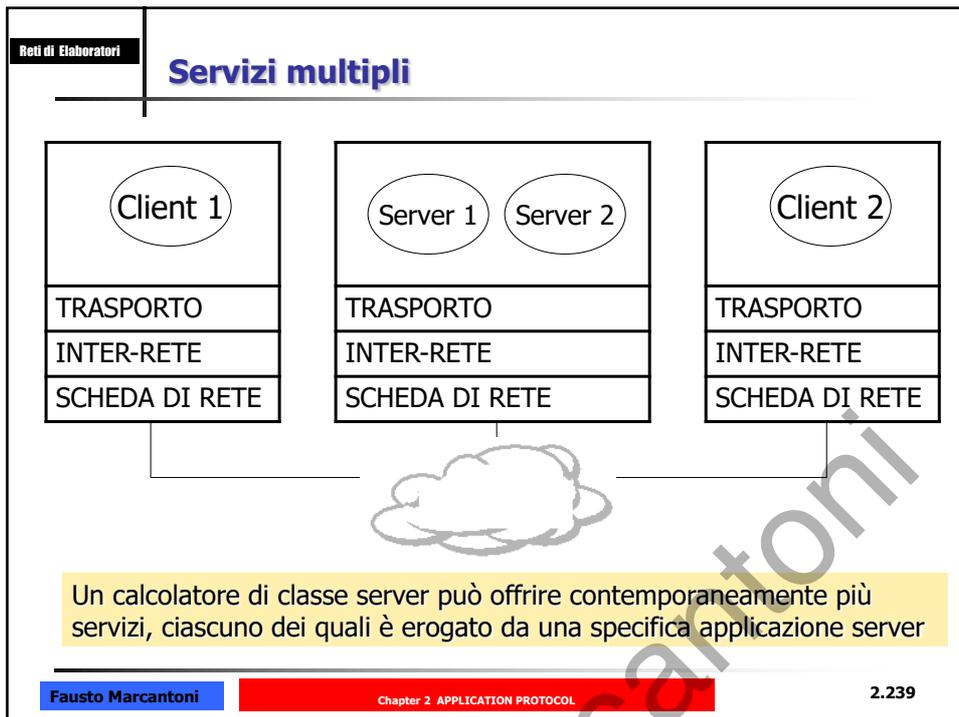
- Programma specializzato
  - Dedicato all'erogazione di un servizio
  - Può servire più di un client alla volta
- Automaticamente avviato all'avvio del sistema
- Attende passivamente di essere contattato dai client
- Accetta richieste da client arbitrari, ma offre un solo servizio
- Calcolatore condiviso e macchina potente

Fausto Marcontoni Chapter 2 APPLICATION PROTOCOL 2.237

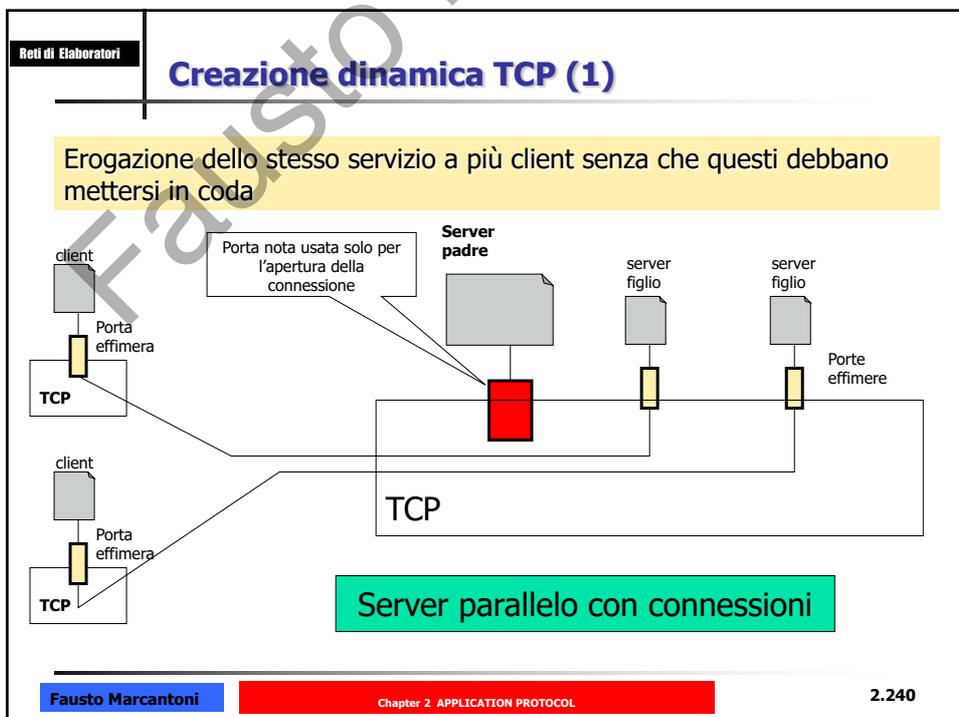
237



238



239



240

Reti di Elaboratori

## Creazione dinamica UDP(2)

Erogazione dello stesso servizio a più client memorizzati in una coda in attesa di essere processati

Server sequenziale senza connessioni  
(specifici per server che usano UDP)

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.241

241

Reti di Elaboratori

## Socket Programming

- Le applicazioni client e server comunicano tramite i protocolli di trasporto
- Le applicazioni debbono fornire al protocollo molte informazioni
  - API (Application Program Interface): insieme di procedure che le applicazioni possono invocare
- API non è definita all'interno dei protocolli di comunicazione, ma parte dei sistemi operativi residenti
- Interfaccia SOCKET standard de facto

- Introdotta UNIX BSD 1981
- Molti produttori aggiungono le socket ai loro sistemi
- Librerie di procedure (Libreria delle Socket)
  - Mettono a disposizione delle applicazioni un API Socket anche se il sistema operativo sottostante non le prevede
  - Quando le applicazioni chiamano una procedura della libreria essa invoca le funzioni del proprio sistema operativo per ottenere l'effetto desiderato

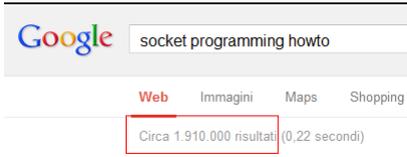
Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.242

242

Reti di Elaboratori

## Socket Programming

```
# crea un socket INET di tipo STREAM
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# ora si connette al server web sulla porta 80 la normale porta http
s.connect(("www.unicam.it", 80))
```



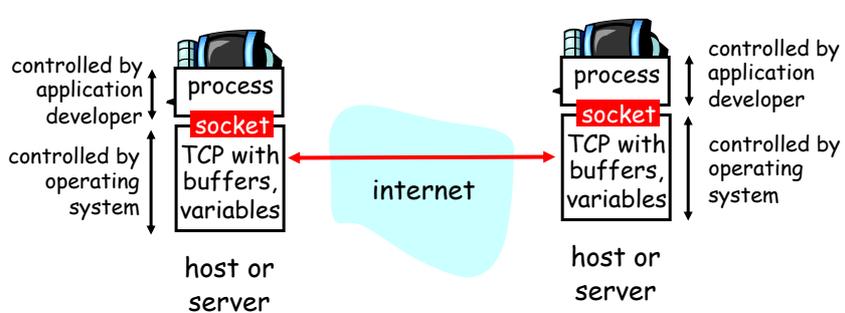
Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.243

243

Reti di Elaboratori

## Socket-programming using TCP

Socket: una interfaccia tra i processi delle applicazioni ed i protocolli di trasporto end to end  
TCP service: trasferimento affidabile di dati tra un processo e l'altro



Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.244

244

Reti di Elaboratori

## Socket programming with TCP

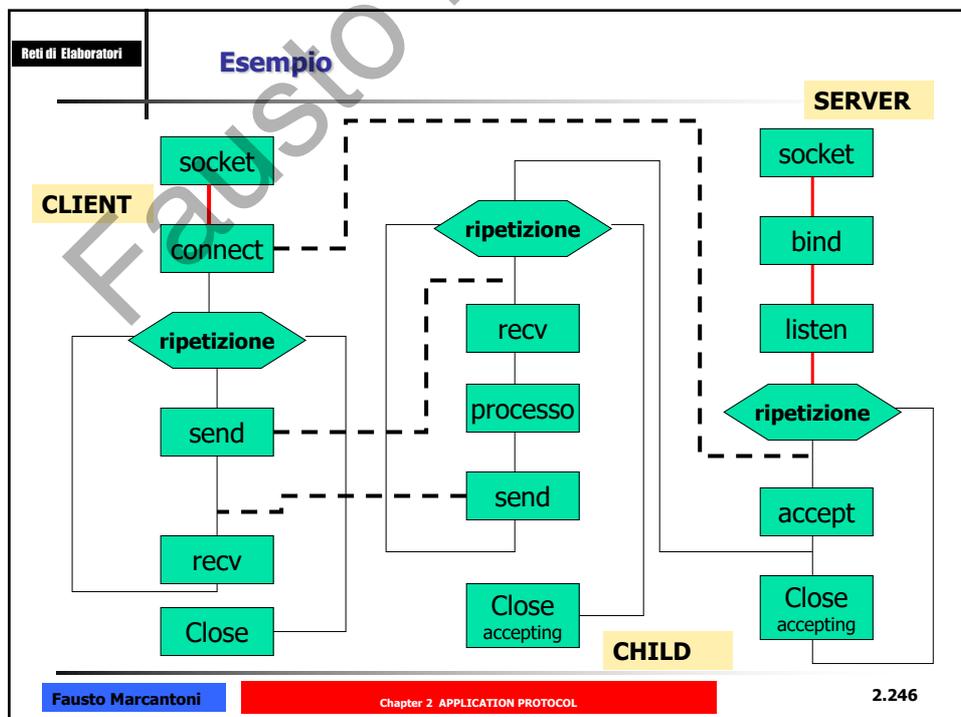
**Client deve contattare il server**

- Il processo server deve essere in esecuzione
- Il server deve avere creato il socket che è in grado di accettare il contatto del client
- Quando il client contatta il server :**
  - Crea un client-local TCP socket
  - specifica IP address e port number del processo server

- Quando **client crea il socket**: il client TCP stabilisce una connessione con il server TCP
- Quando sarà contattato da un client, **il server TCP crea un nuovo socket** affinché il processo server possa comunicare con il client
  - Questo permette al server di comunicare con multiple clients

Fausto Marcantoni
Chapter 2 APPLICATION PROTOCOL
2.245

245



246

Reti di Elaboratori

## Socket programming with UDP

UDP: connectionless tra client e server

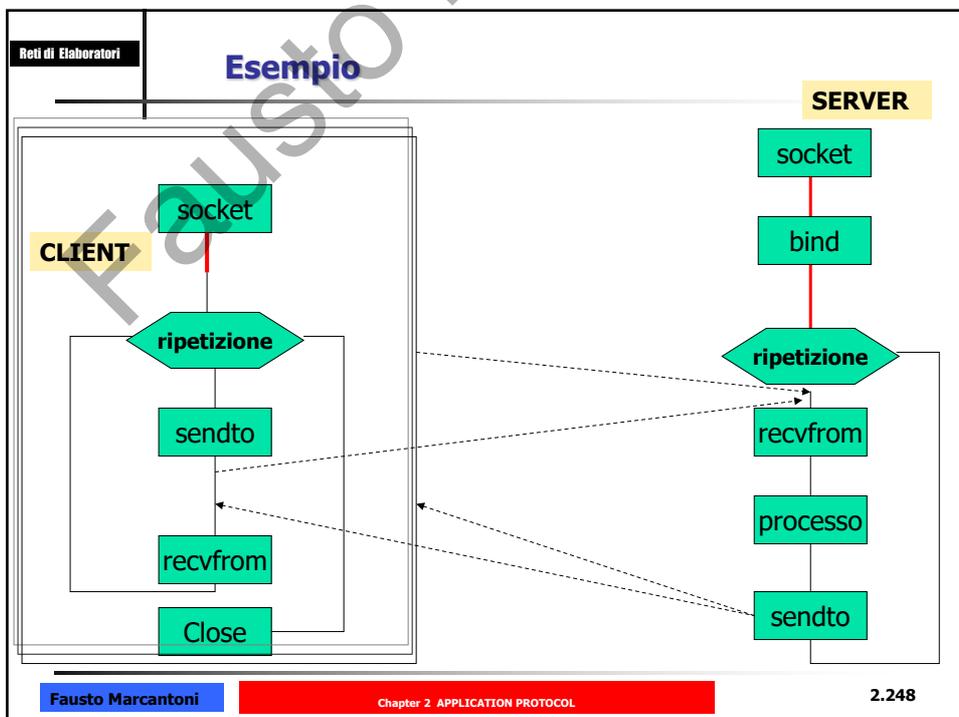
- no handshaking
- sender explicitly attaches IP address and port of destination
- server must extract IP address, port of sender from received datagram

UDP: transmitted data may be received out of order, or lost

application viewpoint  
*UDP provides **unreliable** transfer of groups of bytes ("datagrams") between client and server*

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.247

247



248

## Applicazioni peer-to-peer

**Peer:** coppie di host connessi in modo intermittente, che comunicano direttamente l'uno con l'altro.

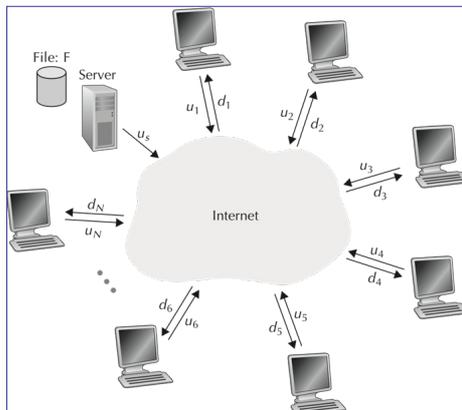
### *Distribuzione di file P2P (peer-to-peer)*

In una distribuzione di file con P2P ciascun peer può ridistribuire agli altri qualsiasi porzione del file abbia ricevuto, aiutando in questo modo il server nel processo di distribuzione.

BitTorrent: uno tra i più diffusi protocolli di distribuzione di file tramite P2P.

Scalabilità dell'architettura P2P

Server e peer sono collegati a Internet con collegamenti di accesso.



Reti di Elaboratori

## BitTorrent



**BitTorrent**

L'insieme di tutti i peer che partecipano alla distribuzione di un particolare file è chiamato **torrent (torrente)**.

I peer in un torrent scaricano **chunk (parti)** del file di uguale dimensione uno dall'altro, con una dimensione tipica di 256 kbyte.

Ciascun torrent ha un nodo di infrastruttura chiamato: **tracker**.

Il meccanismo di incentivazione degli scambi descritto precedentemente viene spesso chiamato **tit-for-tat** ("pan per focaccia").

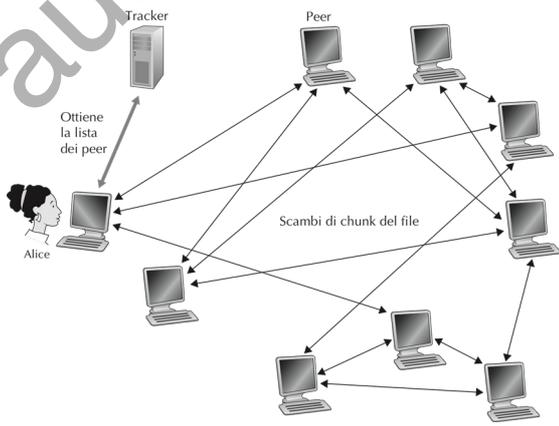
L'ecosistema BitTorrent ha un grande successo con milioni di peer che si scambiano file simultaneamente in centinaia di migliaia di torrent.

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.251

251

Reti di Elaboratori

## BitTorrent



Tracker

Peer

Alice

Ottiene la lista dei peer

Scambi di chunk del file

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.252

252

Reti di Elaboratori

*Fine*

Fausto Marcontoni Chapter 2 APPLICATION PROTOCOL 2.253

Fausto Marcontoni