



**Laurea
in
INFORMATICA**

Internet Reti Sicurezza A.A. 2025/26
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

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

protocolli applicativi

Protocollo	Porta	Descrizione	Protocollo	Porta	Descrizione
HTTP	80 TCP	Trasferimento di pagine web	NFS	2049 TCP/UDP	Condivisione file UNIX
HTTPS	443 TCP	Trasferimento web sicuro con TLS	RTP	varie UDP	Trasporto audio/video in tempo reale
WebSocket	80/443 TCP	Connessioni persistenti client-server	RTSP	554 TCP	Controllo streaming
QUIC	443 UDP	HTTP/3 basato su UDP	SIP	5060/5061 TCP/UDP	Segnalazione chiamate VoIP
SPDY	443 TCP	Versione ottimizzata di HTTP (obsoleta)	H.323	1720 TCP	Comunicazioni audio/video
SMTP	25 / 587 TCP	Invio posta elettronica	MQTT	1883/8883 TCP	Messaggistica IoT
POP3	110 / 995 TCP	Scarico e-mail da server	AMQP	5672/5671 TCP	Message broker enterprise
IMAP	143 / 993 TCP	Accesso remoto a e-mail	CoAP	5683 UDP	Protocolli IoT su dispositivi limitati
DNS	53 TCP/UDP	Risoluzione nomi di dominio	XMPP	5222 TCP	Messaggistica istantanea
DoH	443 TCP	DNS over HTTPS	IRC	6667 TCP	Chat in tempo reale
DoT	853 TCP	DNS over TLS	NTP	123 UDP	Sincronizzazione oraria
LDAP	389 TCP	Servizi di directory	iSCSI	3260 TCP	Storage su IP
LDAPS	636 TCP	LDAP sicuro (TLS)	SMTPS	465 TCP	SMTP cifrato (legacy)
FTP	21 TCP	Trasferimento file	RSync	873 TCP	Sincronizzazione file remota
FTPS	990 TCP	FTP su SSL/TLS	Kerberos	88 TCP/UDP	Autenticazione in reti AD
SFTP	22 TCP	Trasferimento file su SSH	OpenVPN	1194 UDP/TCP	VPN sicura
TFTP	69 UDP	Trasferimento file semplice	IPSec/IKEv2	500/4500 UDP	VPN basata su IPsec
SSH	22 TCP	Accesso remoto sicuro	WHOIS	43 TCP	Informazioni registrazione domini
Telnet	23 TCP	Accesso remoto non cifrato	BGP	179 TCP	Scambio rotte tra router Internet
RDP	3389 TCP	Desktop remoto Windows	RIP	520 UDP	Routing dinamico interno
VNC	5900 TCP	Desktop remoto multiplatforma	TACACS+	49 TCP	Autenticazione di rete Cisco
SNMP	161/162 UDP	Gestione e monitoraggio dispositivi	RADIUS	1812/1813 UDP	Autenticazione utenti di rete
NTP	123 UDP	Sincronizzazione oraria	LDAP	389 TCP	Directory utenti
Syslog	514 UDP	Invio log di sistema	NetBIOS	137-139 UDP/TCP	Servizi di nomi e sessioni su LAN
DHCP	67/68 UDP	Assegnazione dinamica IP	MDNS	5353 UDP	DNS locale multicast
SMB/CIFS	445 TCP	Condivisione file Windows			

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

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.5

5

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

2.6

6

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

```

7

Processi attivi - Windows - PowerShell

Get-Process

```

Microsoft 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 0.00 4632 0 AdminService
272 14 2520 10072 0.00 2880 0 alps
383 23 20048 30940 0.14 12009 1 ApplicationFrameHost
221 8 1460 520 0.00 4624 0 asanero
104 8 1200 5300 0.00 3708 0 AsLdrSvc
336 21 4188 3160 0.30 11112 1 AsaTfCenter
135 10 1600 1544 0.06 5020 1 AsaTfHelper
273 15 2492 2508 0.22 7588 1 AsaTfLoader
189 16 2860 22776 0.20 8268 1 ALDROID
197 12 10572 16788 0.77 3200 0 audiodg
444 21 34180 45168 0.33 2604 1 BPrint
456 70 44864 82136 16.13 11440 1 Brain
397 23 40284 88400 3.20 440 1 chrome
221 15 17180 28496 0.22 1284 1 chrome
262 17 23832 80852 0.38 2756 1 chrome
220 16 19440 43240 0.34 3772 1 chrome
498 38 22276 24840 0.17 6312 1 chrome
375 25 36808 130980 6.86 6684 1 chrome
444 20 12880 16040 10.34 6756 1 chrome
208 23 11840 15412 16.47 7076 1 chrome
1626 59 17292 22704 15.92 7084 1 chrome
212 14 4044 15076 0.14 9052 1 chrome
425 24 25548 43232 6.16 9604 1 chrome
250 19 47212 88100 0.50 10432 1 chrome
209 14 11992 21596 0.09 11740 1 chrome
219 9 1796 7260 0.00 12880 1 chrome
563 28 157104 169140 5.09 12436 1 chrome
227 16 7888 106344 6.38 13112 1 chrome
244 16 23876 10932 0.73 13024 1 chrome
277 18 28544 56704 0.72 14312 1 chrome
468 14 4380 18060 0.08 9432 1 cmd.exe
786 27 3596 8360 0.08 512 1 csrss
832 26 2012 5748 5.06 682 0 csrss
470 17 7148 23552 0.06 8684 1 csrss
567 20 7748 18552 0.06 5748 0 csrss
83 6 628 4596 0.00 6500 0 csrss
235 24 5680 13544 0.14 1368 1 csrss
146 11 1412 7048 0.05 8340 1 csrss
222 15 2356 1248 0.08 8200 1 csrss
1169 46 141028 215296 1264 1 csrss
505 25 4212 16364 4264 0 csrss

```

8

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]

```

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/logd
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/MobileAccessor
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/MobileSoftware
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

```

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

2.11

11

Processi Attivi - MacOS

top

```
faustomarcantoni - top - 138x41
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, 90.53% idle  SharedLibs: 1215M resident, 164M data, 359M linked.
MemRegions: 313662 total, 888M resident, 1187M private, 110 shared. PhysMem: 41G used (3951M wired, 936M compressor), 85G unused.
VM: 361T vsize, 540M framework vsize, 113246(0) swaptins, 124428(0) swaptouts. Networks: packets: 6410742/2388M in, 13263654/5625M out.
Disks: 1868277/546 read, 6325878/780 written.

  PID  COMMAND  %CPU  TIME  #TH  #WO  #PORT  MEM  PURG  CMPSR  PGPR  PPID  STATE  BOOSTS  %CPU_ME  %CPU_OTHS  UID
 641  WindowServer  0.0  01:42:12  29/1  5  4362  1317M  200M-  11M  641  1  running  *0(1)  0.01880  1.94269  88
1435  Google Drive  44.1  01:44:00  127  16  616+  381M+  844K  292M  1410  1410  sleeping  *0(3080)  1.40012  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  01:03:06  42  8  1082  581M+  1085M  0B  10737  1  sleeping  *13598+{661}  2.80733  0.00000  501
18534  VDecoderXPC  3.1  00:05:08  5  2  87  13M  0B  0B  18534  1  sleeping  *0(40+{6})  0.00000  2.75338  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  1.6  14:54:02  6  4  218  1041M  0B  896M  895  1  sleeping  *0(1)  0.05172  0.16139  0
11247  plugin-conta  1.3  00:02:40  30  1  105-  98M-  0B  0B  10444  10444  sleeping  *0(2)  0.00000  0.00000  501
10444  firefox  1.3  01:48:51  122  3  980  850M  457M  0B  10444  1  sleeping  *0(510)  0.00000  0.00000  501
10254  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  01:46:08  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  496M+  242M  0B  10391  1  sleeping  *0(515)  1.73193  0.00000  501
10350  com.apple.Dr  0.9  00:58:74  3  2  508  5616K  0B  0B  10350  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  310M  894  1  sleeping  *0(11)  0.19428  0.00000  0
8254  bluetoothd  0.7  03:36:30  11  5  414  10M  224K  0B  8254  1  sleeping  *0(11)  0.37888  0.00000  0
967  distnoted  0.7  02:10:04  4  3  710  403K  0B  848K  967  1  sleeping  *0(1)  0.00000  0.70243  501
573  logd  0.7  24:02:44  4  3  2050  8482K  0B  5504K  573  1  sleeping  *0(1)  0.00000  0.00000  0
10255  Google Chrom  0.7  01:33:15  15  2  268  157M  490M  0B  10249  10249  sleeping  *1(12)  0.00000  0.00000  501
625  distnoted  0.4  01:59:24  3  2  252  2753K  0B  1024K  625  1  sleeping  *0(1)  0.00000  0.40896  241
754  com.apple.Dr  0.4  01:40:11  8  6  1525  33M-  0B  3168K  754  1  sleeping  0(1)  0.00000  0.00211  270
11288  Terminal  0.4  00:07:34  9  4  358  416M+  314M-  0B  11288  1  sleeping  *0(71+)  0.04701  0.00785  501
687  airportd  0.3  06:30:15  8  6  363+  11M  0B  1040K  687  1  sleeping  *53(550)  0.17341  0.00000  0
974  cprfresd  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  756K  304K  1024K  691  1  sleeping  *0(88366+)  0.00000  0.06721  24
9650  com.apple.Am  0.3  00:53:38  3  1  80  380K  0B  0B  9650  1  sleeping  0(24133)  0.13689  0.00000  0
577  fsentend  0.3  04:15:57  19  1  164- 4453K- 0B  896K  577  1  sleeping  *0(1)  0.00000  0.00000  0
11252  plugin-conta  0.2  00:00:06  30  1  112  44M  0B  0B  10444  10444  sleeping  *0(2)  0.00000  0.00000  501
637  corebrihtme  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  07:43:37  7  4  639  210M+  230M  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  1440K  838  1  sleeping  *0(851410+)  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

2.12

12



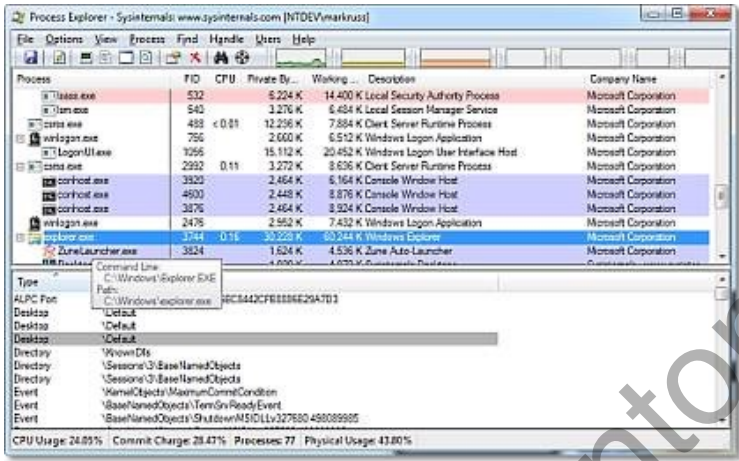
Process Monitor



Reti di Elaboratori

Process Explorer

Process Explorer



Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.15

15

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

2.16

16

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

2.17

17

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

2.18

18

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

2.19

19

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

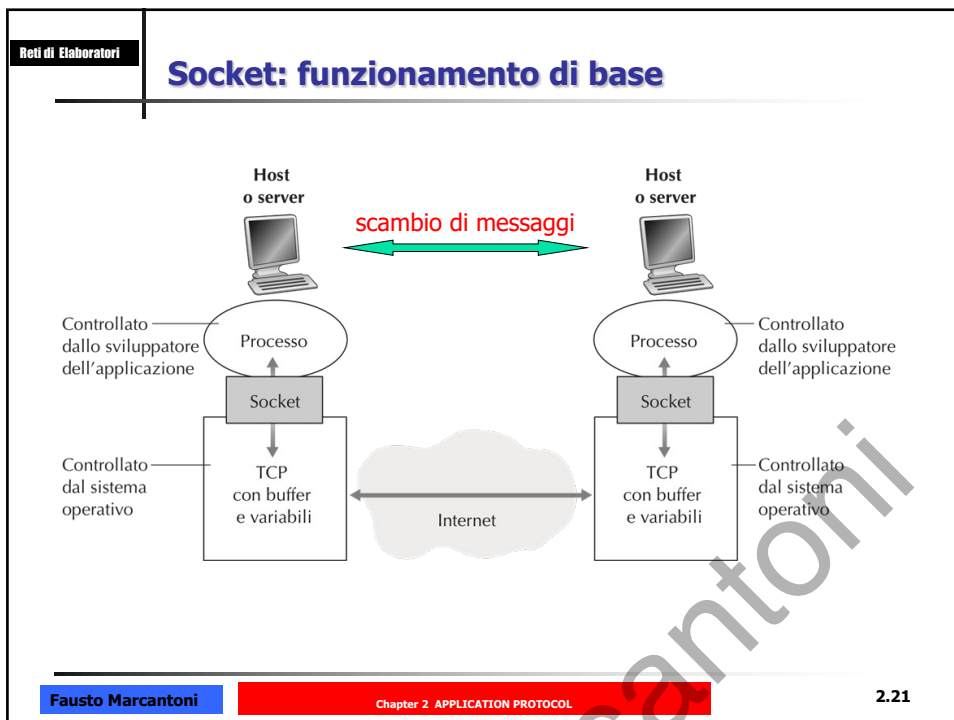
<http://www0.mi.infn.it/~cmp/CorsoReti/slides03/Lab3-sockets-java.pdf>

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.20

20



21

Reti di Elaboratori

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)

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.22

22

Reti di Elaboratori

netstat

```

C:\Users\fausto.mfausto>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    0.0.0.0:903                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    0.0.0.0:5357               0.0.0.0:0          LISTENING
TCP    127.0.0.1:1443             127.0.0.1:2794     ESTABLISHED
TCP    127.0.0.1:1443             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.mfausto>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
  opzione.
-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, TCPv6 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.
  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

```

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.23

23

Reti di Elaboratori

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.amministr:bootpc 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

2.24

24

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 statistics. It can display more TCP and UDP
  connections than netstat.

OPTIONS
  When no option is used ss displays TCP/UNIX/UDP connections that have
  established connections.

  -h, --help          Show summary of options.
  -V, --version       Output version information.
  -H, --no-header     Suppress header line.
  -O, --oneline       One line per connection.

Manual page ss(8) line 1 (press h for help)

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      root@kali:~#
ESTAB      0          0          193.205.92.191:34100   13.35.43.91:443       root@kali:~#
ESTAB      0          0          193.205.92.191:37348   99.86.154.2:443       root@kali:~#
ESTAB      0          0          193.205.92.191:40834   34.216.82.69:443      root@kali:~#

```

man ss

25

ss command

LinuxCrafts → Tutorials → Linux → ss command: Display Linux TCP / UDP Network/Socket Information

ss command: Display Linux TCP / UDP Network/Socket Information

Author: Vivek Gite • Last updated: March 1, 2024 • 13 comments

The ss command is used to show socket statistics. It can display stats for PACKET sockets, TCP sockets, UDP sockets, DCCP sockets, RAW sockets, Unix domain sockets, and more. It allows showing information similar to [netstat command](#). It can display more TCP and state information than other tools. It is a new, incredibly useful and faster (as compared to netstat) tool for tracking TCP connections and sockets. SS can provide information about:



<https://www.cyberciti.biz/tips/linux-investigate-sockets-network-connections.html>

26

NAME
lsdf - list open files

SYNOPSIS

```
ls of [-?abChlnNOPRtUvVX] [-A A] [-c c] [+c c] [+|-d d] [+|-D D] [+|-e s] [+|-E] [+|-f [cfgOn]] [-F f] [-g s]
[-s [p:s]] [-S [t]] [-T [t]] [-u s] [+|-w] [-x [fl]] [+|-X] [-z [z]] [-Z [Z]] [--] [names]
```

```
DESCRIPTION
ls of revision 4.91 lists on its standard output file information about files opened by processes for the following UNIX dialects:
```

[illegible]

-i [i]

-I [i] selects the listing of files any of whose Internet address matches the address specified in i. If no address is specified, this option selects the listing of all Internet and x.25 (HP-UX) network files.

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.27

27

[Home](#) / [Tutorials](#) / [Advanced](#) / [Install Netcat on macOS](#)

How to Install Netcat on macOS [Step-by-Step Guide]



<https://eldernode.com/tutorials/install-netcat-on-macos/>

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.28

28

Reti di Elaboratori

TCPView

TCPView

Fausto Marcontoni

Chapter 2 APPLICATION PROTOCOL

2.29

29

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

Banda

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

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)

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

Fausto Marcontoni

Chapter 2 APPLICATION PROTOCOL

2.30

30

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	Sì	Audio: dal pochi Kbps a 1 Mbps Video: da 10 Kbps a 5 Mbps	Sì: centinaia di ms.
Audio/Video memorizzati	Sì	Audio: dal pochi Kbps a 1 Mbps Video: da 10 Kbps a 5 Mbps	Sì: pochi secondi
Giochi interattivi	Sì	Fino a 10 Kbps (se basta???)	Sì: centinaia di ms.
Messaggistica istantanea	No	Variabile	Sì e No

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.31

31

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

2.32

32

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
2.33

33

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
2.34

34

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

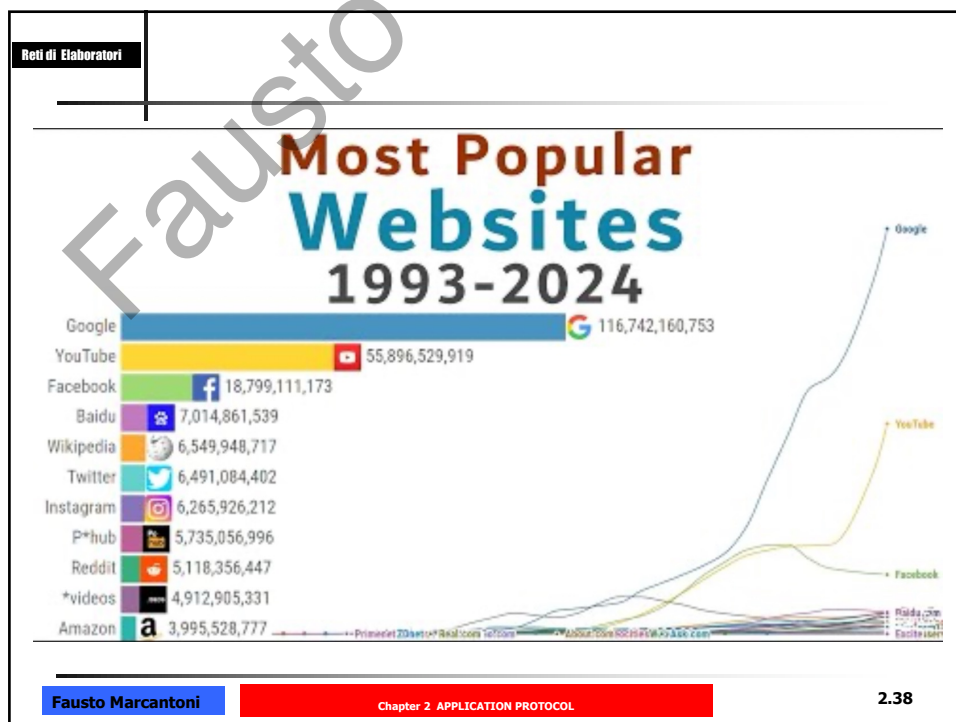
WEB - WWW - World Wide Web

WEB - WWW - World Wide Web





37



38

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

2.39

39

Reti di Elaboratori

Iper testo

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

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.40

40

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>

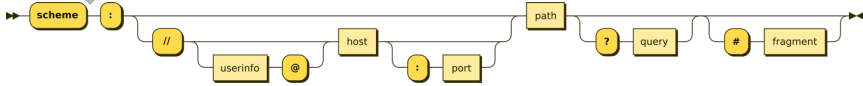
Fausto Marcantoni
Chapter 2 APPLICATION PROTOCOL
2.41

41

Reti di Elaboratori

URI

In informatica, lo **Uniform Resource Identifier** (in acronimo URI) è una sequenza di caratteri che identifica universalmente ed univocamente una risorsa.



Fausto Marcantoni
Chapter 2 APPLICATION PROTOCOL
2.42

42

URI - Uniform Resource Identifier

Un URI può essere classificato:

- ❖ come qualcosa che definisce posizioni (URL)
- ❖ nomi (URN)
- ❖ entrambi

Un **URL** (**Uniform Resource Locator**) è un URI che identifica una risorsa tramite la sua "**collocazione**" ("**location**") in un grafo.

Per esempio, l'URL `http://www.example.com/` è un URI che identifica una risorsa (l'home page di un sito web)

Un **URN** (**Uniform Resource Name**) è un URI che identifica una risorsa mediante un "**nome**" in un particolare dominio di nomi ("**namespace**"). Per esempio l'URN `urn:isbn:0-395-36341-1` è un URI che mappa universalmente e univocamente un libro mediante il suo identificativo, o nome, (0-395-36341-1) nel namespace dei codici ISBN, ma non suggerisce dove e come possiamo ottenere una copia di tale libro. Si noti che, da RFC8141, un URN è un URI assegnata sotto lo schema URI "urn".

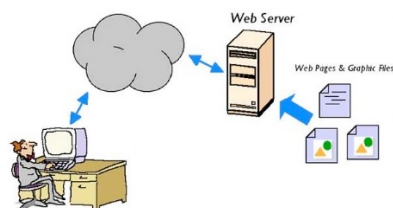
WWW: terminologia essenziale

- 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



Reti di Elaboratori

WWW: terminologia essenziale

protocol

hostname

directory

filename

query parameters

http://www.mywebsite.com/apparel/skirt.php?sku=123&lang=en§=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 querystring è una stringa di caratteri che consente di passare al server uno o più parametri.

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.45

45

Reti di Elaboratori

WWW: quali sono I browser più utilizzati

BROWSER	TOTAL MARKET SHARE
☒ Chrome	59.57%
☒ Microsoft Internet Explorer	16.50%
☒ Firefox	12.32%
☒ Microsoft Edge	5.65%
☒ Safari	3.66%
☒ Opera	1.21%
Proprietary or Undetectable	0.30%
☒ Konqueror	0.01%
☒ Android Browser	0.00%
☒ Obigo	0.00%

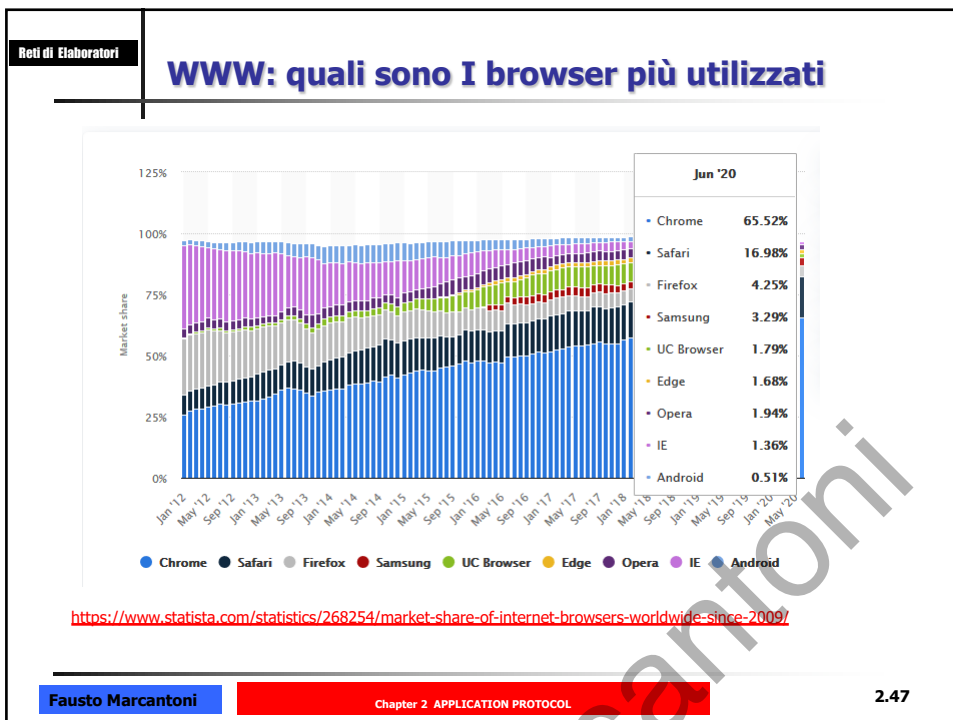
<http://marketshare.hitslink.com/report.aspx?qprid=0>

Fausto Marcantoni

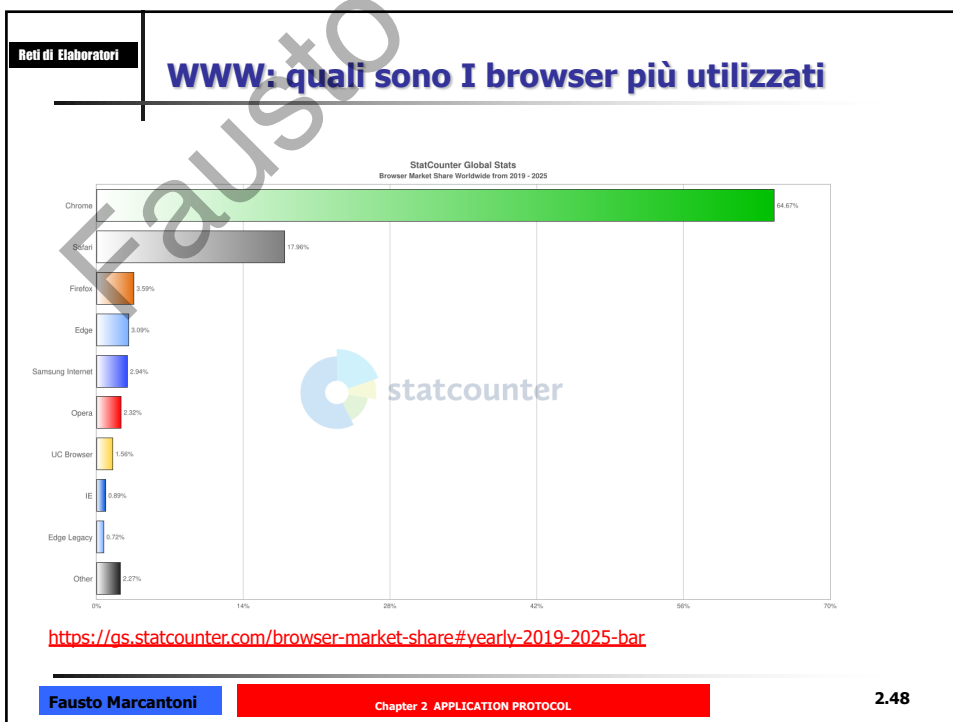
Chapter 2 APPLICATION PROTOCOL

2.46

46



47



48

Reti di Elaboratori

Mosaic



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

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.49

49

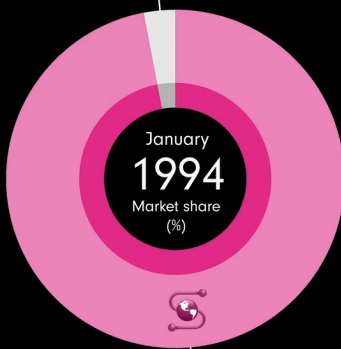
Reti di Elaboratori

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

2.50

50

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 $\neq$ HTTP

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

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.51

51

Reti di Elaboratori

HTML 5

1476 pagine

HTML

Living Standard — Last Updated 4 October 2023

One-Page Version	Multipage Version
html.spec.whatwg.org	multipage
Version for Web Devs	PDF Version
dev	/print.pdf
Translations	FAQ
on GitHub	on GitHub
Chat	Contribute on GitHub
on Matrix	whatwg/html repository
Commits	Snapshots
on GitHub	as of this commit
Twitter Updates	Open Issues
@whatwgdev	Filed on GitHub
Open an Issue	Tests
whatwg.org/newbug	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/>

<https://html.spec.whatwg.org/print.pdf>

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.52

52

Reti di Elaboratori	Storia di HTTP
Versioni HTTP: 0.9: un semplicissimo protocollo client-server di sola richiesta di risorse HTML, senza flessibilità né nella direzione, né nel formato delle risorse. Utilizzata nel primo prototipo WWW e nei primi server NCSA. 1.0 (RFC 1945): il protocollo diventa generico e definisce la statelessness (apolidia), e definisce alcuni metodi anche per l'upload di dati. Utilizzato fino al 1998-99 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. 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. 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.	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL 2.53

53

Reti di Elaboratori	Il protocollo http
Client-server 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. Protocollo generico HTTP è indipendente dal formato dati con cui vengono trasmesse le risorse. Può funzionare per documenti HTML come per binari, eseguibili, oggetti distribuiti o altre strutture dati più o meno complicate. Statelessness (apolidia) 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. Il client è tenuto a ricreare da zero il contesto necessario al server per rispondere.	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL 2.54

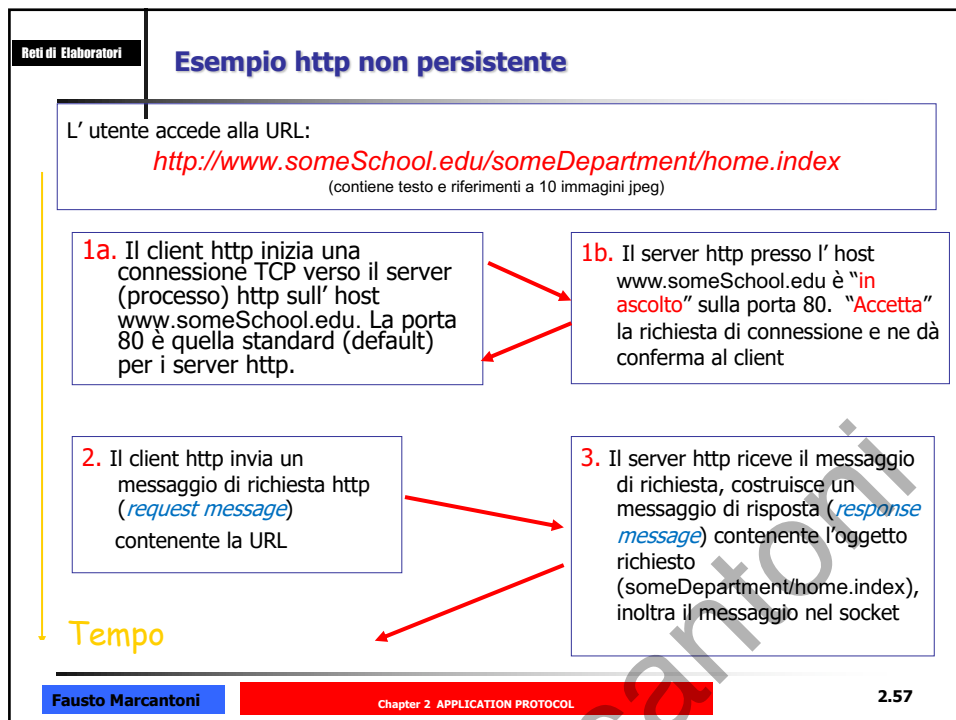
54

Reti di Elaboratori	Il protocollo http (cont.)
http: usa TCP: ■ Il client inizia una connessione TCP (socket) verso il server sulla porta 80 ■ Il server accetta la connessione TCP dal client (sempre in ascolto sulla porta 80) ■ Vengono scambiati messaggi http (messaggi del protocollo di livello applicativo) tra il browser (client http) e il Web server (server http) ■ La connessione TCP è chiusa	
http è "stateless" ■ Il server non mantiene informazione sulle richieste precedenti del client <i>Non confondere i file di log dell'applicativo</i>	
I protocolli che mantengono informazione di stato sono complessi (es. TCP) !	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL
2.55	

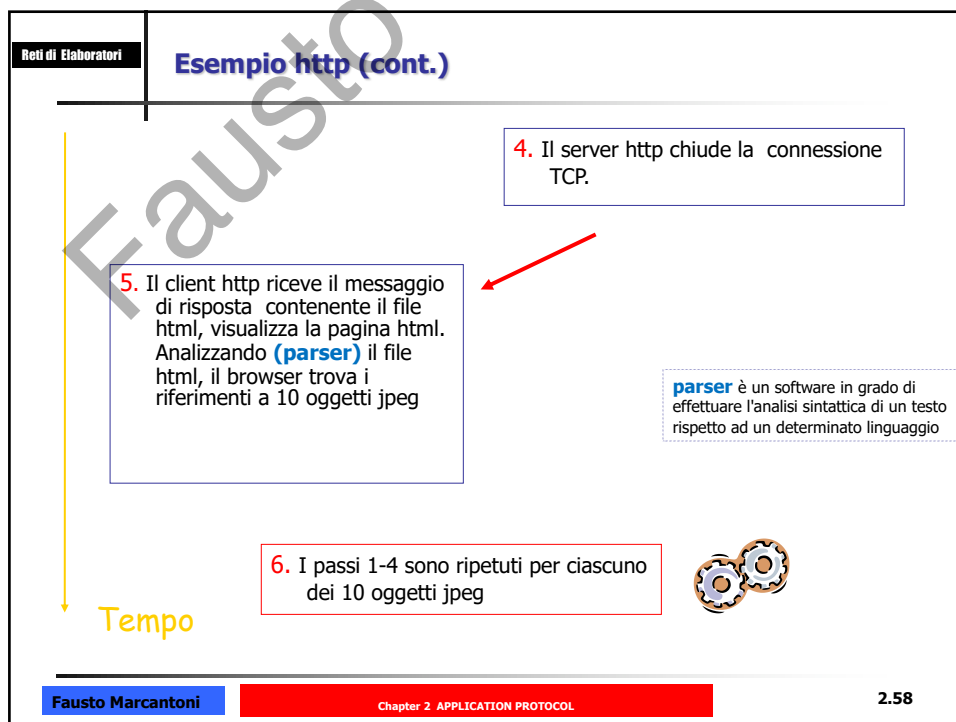
55

Reti di Elaboratori	HTTP connections
Nonpersistent HTTP ■ Al massimo un oggetto è trasmesso per ogni connessione TCP. ■ HTTP/1.0 usa nonpersistent HTTP	
Persistent HTTP ■ Oggetti multipli possono essere trasmessi in una singola connessione TCP ■ HTTP/1.1 usa persistent connections in default mode	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL
2.56	

56



57



58

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: 181]
[The RTT to ACK the segment was: 0.020975000 seconds]
[1RTT: 0.021034000 seconds]
```

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.59

59

Reti di Elaboratori

Grafico RTT

grafico RTT [grafico RTT.pcapng](https://computerscience.unica.it/marcantoni/rete_pcap/grafico%20RTT.pcapng) [https://computerscience.unica.it/marcantoni/rete_pcap/grafico RTT.pcapng](https://computerscience.unica.it/marcantoni/rete_pcap/grafico%20RTT.pcapng)

selezionare la riga 15 → Statistiche → Grafici dei flussi TCP → Round Trip Time

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.60

60

Reti di Elaboratori

Collegarsi al sito web→ analizzare RTT

Aprire wireshrk

Selezionare l'interfaccia usata

Collegarsi al sito web: <http://193.205.92.204>

Options: (12 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted, No-Operation (NOP), Window scale

[Sequence]

[SEQ/ACK analysis]

[This is an ACK to the segment in frame: 4789]

[The RTT to ACK the segment was: 0.001186000 seconds]

[iRTT: 0.001244000 seconds]

ip.addr == 193.205.92.204 and not tcp.analysis.retransmission and not tcp.analysis.duplicate_ack_frame

Fausto Marcontoni

Chapter 2 APPLICATION PROTOCOL

2.61

61

Reti di Elaboratori

consiglio ...

Preferenze per la privacy dei siti web

☐ Segnala ai siti web di non vendere o condividere i miei dati [Ulteriori informazioni](#)

Cookie e dati dei siti web

I cookie, i dati salvati dai siti web e la cache stanno utilizzando 66,2 MB di spazio su disco [Ulteriori informazioni](#)

☐ Elimina cookie e dati dei siti web alla chiusura di Firefox

Elimina dati...

Gestisci dati...

Gestisci eccezioni...

Password

☒ Chiedi se salvare le password

☒ Compila automaticamente nomi utente e password

☒ Suggerisci password complesse

☐ Suggerisci alias di posta elettronica Firefox Relay per proteggere il tuo [Ulteriori](#)

Eccezioni...

Password salvate

Fausto Marcontoni

Chapter 2 APPLICATION PROTOCOL

2.62

62

http su debian IRS.pcapng

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	193.205.92.98	193.205.92.204	TCP	78	53486 → 80 [SYN, ECE, CWR] Seq=376722...
2	0.000003	193.205.92.204	193.205.92.98	TCP	74	80 → 53486 [SYN, RST, ECE] Seq=462581...
3	0.000092	193.205.92.98	193.205.92.204	TCP	66	53486 → 80 [ACK] Seq=376722768 Ack=46...
4	0.001127	193.205.92.204	193.205.92.98	HTTP	435	GET / HTTP/1.1
5	0.002572	193.205.92.204	193.205.92.98	TCP	66	80 → 53486 [ACK] Seq=46258147 Ack=376...
6	0.004032	193.205.92.204	193.205.92.98	HTTP	736	HTTP/1.1 200 OK (text/html)
7	0.004213	193.205.92.98	193.205.92.204	TCP	66	53486 → 80 [ACK] Seq=376722337 Ack=46...
8	0.020003	193.205.92.98	193.205.92.204	HTTP	454	GET /image1.gif HTTP/1.1
9	0.020008	193.205.92.204	193.205.92.98	TCP	1514	80 → 53486 [ACK] Seq=46258017 Ack=376...
10	0.020009	193.205.92.204	193.205.92.98	TCP	1514	80 → 53486 [ACK] Seq=46252265 Ack=376...
11	0.020009	193.205.92.204	193.205.92.98	TCP	1514	80 → 53486 [ACK] Seq=46253713 Ack=376...
12	0.020091	193.205.92.204	193.205.92.98	TCP	1514	80 → 53486 [ACK] Seq=46255161 Ack=376...

http su debian IRS.pcapng

!(tcp.flags.ecn == 1 or tcp.flags.cwr == 1)

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.63

63

http su debian IRS.pcapng - filtro

ip.addr == 193.205.92.204

No.	Time	Source	Destination	Protocol	Length	Info
349	1.855906	193.205.92.204	193.205.92.204	TCP	66	8085 → 80 [SYN] Seq=6154240 Len=0 MSS=5400 WS=256 SACK_PERM=1
350	1.855908	193.205.92.204	193.205.92.204	TCP	66	80 → 8085 [SYN, ACK] Seq=6154240 Len=0 MSS=5400 WS=256 SACK_PERM=1
351	1.855910	193.205.92.204	193.205.92.204	TCP	54	8085 → 80 [ACK] Seq=6154240 Len=0 MSS=5400 WS=256 SACK_PERM=1
402	2.646200	193.205.92.204	193.205.92.204	HTTP	435	GET / HTTP/1.1
403	2.646208	193.205.92.204	193.205.92.204	TCP	66	80 → 8085 [ACK] Seq=6154240 Len=0 MSS=5400 WS=256 SACK_PERM=1
404	2.647153	193.205.92.204	193.205.92.204	HTTP	736	HTTP/1.1 200 OK (text/html)
405	2.648065	193.205.92.204	193.205.92.204	TCP	54	8085 → 80 [ACK] Seq=6154240 Len=0 MSS=5400 WS=256 SACK_PERM=1
406	2.648113	193.205.92.204	193.205.92.204	HTTP	435	GET /image1.gif HTTP/1.1
407	2.649167	193.205.92.204	193.205.92.204	TCP	1514	80 → 8085 [ACK] Seq=6154240 Len=0 MSS=5400 WS=256 SACK_PERM=1
408	2.649167	193.205.92.204	193.205.92.204	TCP	1514	80 → 8085 [ACK] Seq=6154240 Len=0 MSS=5400 WS=256 SACK_PERM=1
409	2.649167	193.205.92.204	193.205.92.204	TCP	1514	80 → 8085 [ACK] Seq=6154240 Len=0 MSS=5400 WS=256 SACK_PERM=1

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.64

64

Reti di Elaboratori

http su debian IRS.pcapng - flusso

selezionare la riga con GET / HTTP/1.1
Segui → Flusso HTTP

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.65

65

Reti di Elaboratori

http su debian IRS.pcapng – flusso HTTP

Fausto Marcantoni

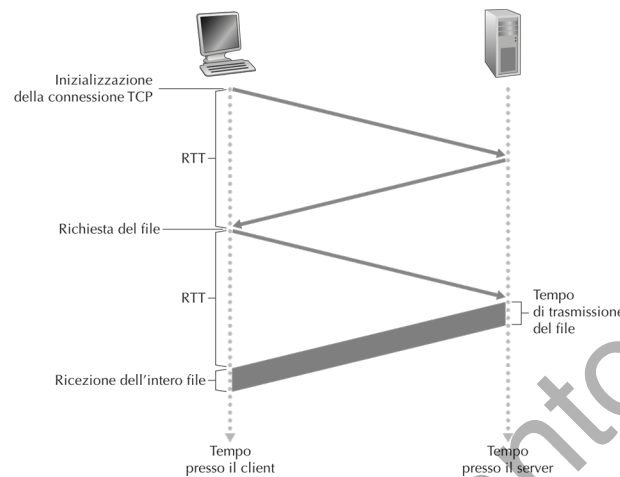
Chapter 2 APPLICATION PROTOCOL

2.66

66

Response time modeling

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



Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.67

67

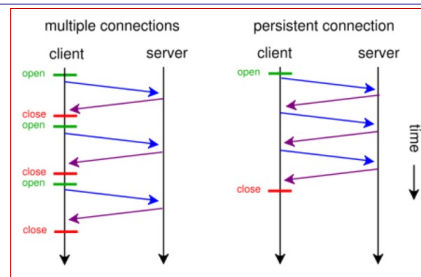
Persistent HTTP

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



Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.68

68

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

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.69

69

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

2.70

70

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)

header lines

Carriage return, line feed indica fine messaggio

Chiudi la connessione al termine della richiesta

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
Accept-language: fr
(extra carriage return, line feed)
```

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.71

71

Reti di Elaboratori

Message http request : formato generale

Riga di richiesta

Righe di intestazione

Riga vuota

Corpo dell'entità

metodo	sp	URL	sp	versione	cr	lf
nome del campo di intestazione:		sp	valore	cr	lf	
nome del campo di intestazione:		sp	valore	cr	lf	
cr	lf					

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.72

72

Reti di Elaboratori	Message http request : formato generale
• HEAD - Simile a GET, ma richiede solo le informazioni dell'header. - Utile per controllare se un documento esiste e quanto è recente. • POST - Simile a GET, ma codifica gli input in modo diverso. - Utile per proporre i contenuti dei form a un programma CGI. CGI: Common Gateway Interface • PUT - Trasferisce un documento al server. - Presente a partire dalla versione HTTP/1.1. • DELETE - Elimina un documento dal server. - Presente a partire dalla versione HTTP/1.1.	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL 2.73

73

Reti di Elaboratori	Message http request : formato generale
• TRACE - Attraverso questo metodo un client può vedere cosa viene ricevuto dall'altra parte della catena delle richieste in modo tale da eseguire delle diagnosi - La richiesta TRACE non deve includere un corpo - 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" - Le risposte non dovrebbero essere soggette a cache. • CONNECT - Usata con i proxy per tunnel (es. SSL)	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL 2.74

74

Reti di Elaboratori

The Method token

<https://www.rfc-editor.org/rfc/rfc9110.html>
<https://www.rfc-editor.org/rfc/rfc9110.html#name-method-definitions>

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

2.75

75

Reti di Elaboratori

Formato del messaggio http response

protocol - status code - status phrase

status line

header lines

data, e.g., requested html file

```

HTTP/1.1 200 OK
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 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

2.76

76

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

2.77

77

Reti di Elaboratori

Hypertext Transfer Protocol (HTTP) Field Name Registry



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>

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.78

78

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

2.79

79

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

2.80

80

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

2.81

81

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

```

graph LR
    Client[Telnet client] ---|telnet protocol| Server[Telnet server (telnetd)]
    subgraph Credentials
        direction TB
        C1["username"]
        C2["password"]
    end
    Credentials --- Client
    Credentials --- Server

```

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.82

82

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 si vuole connettere dando il suo nome di dominio oppure il suo indirizzo IP e specificando la porta

```

C:\Users\Fausto>telnet 192.168.49.128

Microsoft Windows [Version 6.0.6002.18005]
(c) 2009 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\Fausto>telnet 192.168.49.128
telnet 192.168.49.128
Ubuntu 22.04.1 LTS
studente-virtual-machine login: studente
Password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-48-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 updates can be applied immediately.
Last login: Wed Oct 12 15:21:52 CEST 2022 from 192.168.49.1 on pts/1
studente@studente-virtual-machine:~$

```

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.83

83

Reti di Elaboratori

Telnet per Windows

The screenshot shows the Windows 'Programmi e funzionalità' (Programs and Features) window. In the 'Funzionalità Windows' (Windows Features) section, the 'Client Telnet' checkbox is checked and highlighted with a red box. Other features like 'Client Cartelle di lavoro', 'Client FTP', and 'Contenitori' are also visible.

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.84

84

Reti di Elaboratori

telnet - wireshark

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.85

85

Reti di Elaboratori

telnet - wireshark

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.86

86

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

2.87

87

Reti di Elaboratori

Telnet esempio 1

1. Telnet verso un Web server:

telnet <<indirizzo IP>> **80** Apre connessione TCP verso la porta 80 (default)

digitare una richiesta http GET:

GET Digitando ciò (carriage return due volte), si invia una richiesta GET al server http

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.88

88

Telnet esempio 2

2. Telnet verso un Web server:

telnet <<indirizzo IP>> **80** Apre connessione TCP verso la porta 80 (default)

digitare una richiesta http:

3. Si osservi la risposta!

```
faustomarcantoni@MacBook-Pro-di-Fausto ~ % telnet 193.205.92.204 80
Trying 193.205.92.204...
Connected to debian-lrs.amministrazione.unicam.
Escape character is '^J'.
GET / HTTP/1.1
HOST: www.unicam.it
USER-AGENT: Fausto

HTTP/1.1 200 OK
Date: Mon, 20 Oct 2025 13:47:16 GMT
Server: Apache/2.4.65 (Debian)
Last-Modified: Tue, 08 Oct 2024 15:48:09 GMT
ETag: "6f4-623f9106fa3d8"
Accept-Ranges: bytes
Content-Length: 1780
Vary: Accept-Encoding
Content-Type: text/html

<!DOCTYPE html>
<html>
<head>
<title>Corso Internet Reti e Sicurezza</title>
</head>
<body>
```

User Agent HTTP

User-Agent: <product> / <product-version> <comment>

<https://deviceatlas.com/blog/list-of-user-agent-strings>

- ✓ [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](#)

Reti di Elaboratori

User agent generator

[Tutti](#)
[Video](#)
[Immagini](#)
[Shopping](#)
[Notizie](#)
[Altro](#)
[Strumenti](#)

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-a... 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> > u... 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&sa=X&ved=2ahUKEwIdwa3n2dz6AhVdI_0HHWBI&pg=01Q16BAQ7EAF&biw=1473&bih=886&doc=1

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.91

91

Reti di Elaboratori

Ricerca di testo

http matches "GET"

frame contains "GET"

Frame 11710: 521 bytes on wire (4168 bits), 521 bytes captured (4168 bits) on interface \Device\NPF_{1C45D58C-D191-43A8-9A6F-DFDA23ECCEA1}, id 0

Ethernet II, Src: mfausto.local (d8:9b:e6:8a:e6:1d), Dst: Cisco_E2:B4:5F (18:b3:d5:e2:b4:5f)

Internet Protocol Version 4, Src: mfausto.local (193.205.92.201), Dst: www.unina.it (143.225.19.50)

Transmission Control Protocol, Src Port: 2932, Dst Port: 80, Seq: 1, Ack: 1, Len: 467

Hypertext Transfer Protocol

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/106.0.1370.3

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

\r\n

[Full request URI: http://www.unina.it/]

[HTTP request 1/1]

[Response in frame: 11716]

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.92

92

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

WIRESHARK

internet reti sicurezza

Esercitazioni

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.93

93

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

2.94

94

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

2.95

95

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=0403710006620D11050774610068656A161D647E721804797E7A027808557F7C6A0205
punto-informatico.it/
1536
3164909568
29887167
1043606064
29813841
*

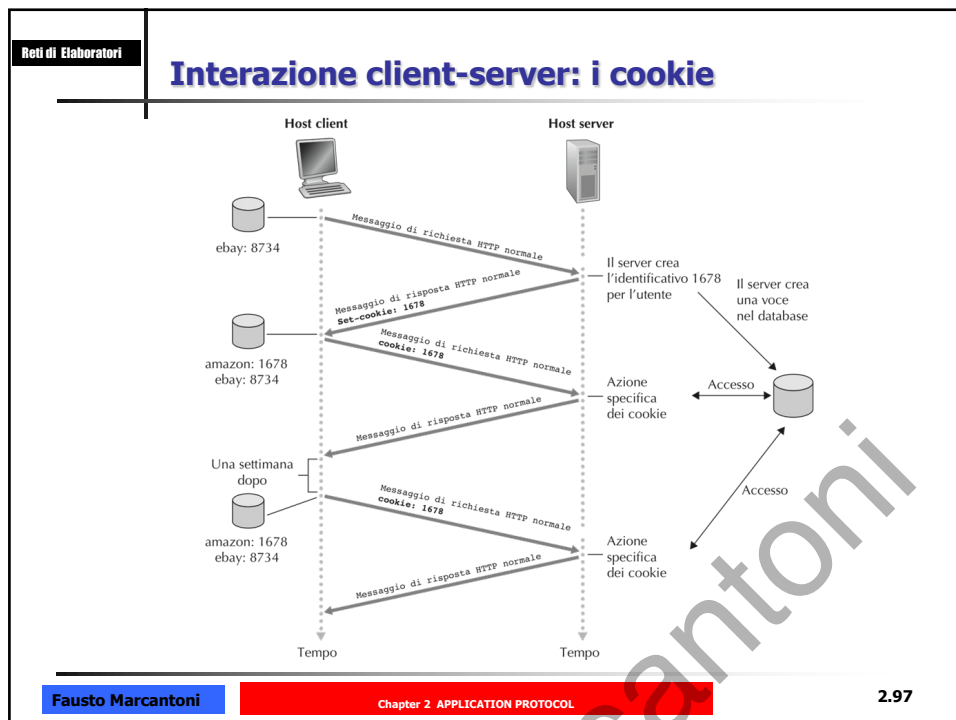
```

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.96

96



97

Reti di Elaboratori

Using the Set-Cookie header

RFC 6265 HTTP State Management Mechanism April 2011

```
-- Server -> User Agent --
Set-Cookie: SID=31d4d96e407aad42

-- User Agent -> Server --
Cookie: SID=31d4d96e407aad42

The server can alter the default scope of the cookie using the Path
and Domain attributes. For example, the server can instruct the user
agent to return the cookie to every path and every subdomain of
example.com.

-- Server -> User Agent --
Set-Cookie: SID=31d4d96e407aad42; Path=/; Domain=example.com

-- User Agent -> Server --
Cookie: SID=31d4d96e407aad42

As shown in the next example, the server can store multiple cookies
at the user agent. For example, the server can store a session
identifier as well as the user's preferred language by returning two
Set-Cookie header fields. Notice that the server uses the Secure and
HttpOnly attributes to provide additional security protections for
the more sensitive session identifier (see Section 4.1.2.)

-- Server -> User Agent --
Set-Cookie: SID=31d4d96e407aad42; Path=/; Secure; HttpOnly
Set-Cookie: lang=en-US; Path=/; Domain=example.com

-- User Agent -> Server --
Cookie: SID=31d4d96e407aad42; lang=en-US
```

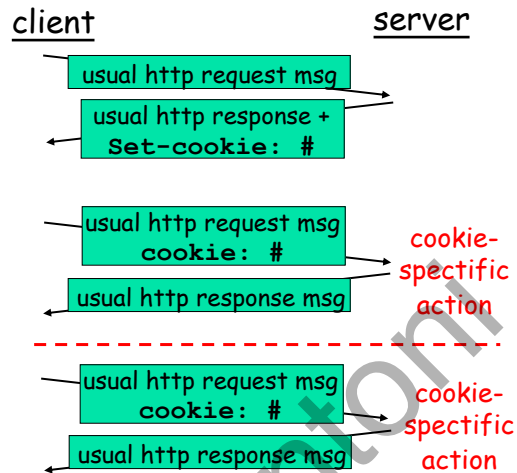
<https://data.ietf.org/doc/html/rfc6265#section-3.1>

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.98

98

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



Come funzionano i Cookie

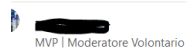
- Un cookie, dall'inglese "biscotto", è un piccolo file di testo che viene ricevuto dal browser durante la navigazione e viene memorizzato in un opportuno repository →

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>



Risposta il marzo 21, 2018

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.



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

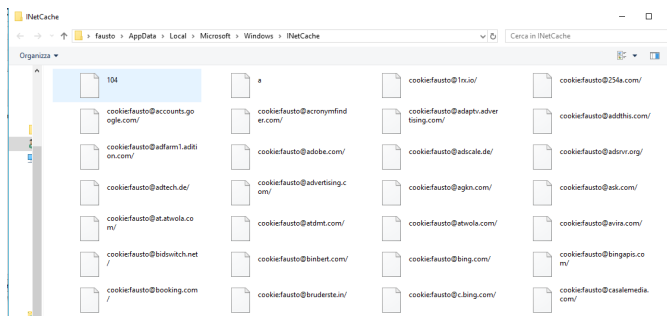
2.101

101

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

2.102

102

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

2.103

103

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

2.104

104

Reti di Elaboratori

Esempio – non funziona

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

BVFWLZZ.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

cookiefausto@google.it/

cookiefausto@microsoft.com/

cookiefausto@erving-sys.com/

cookiefausto@ieom/

cookiefausto@n/

cookiefausto@tr

Copiato sul Desktop

BVFWLZZ.txt

contenuto

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

3

Reti di Elaboratori	Come funzionano i Cookie • Un cookie è un file di testo. Non può fare nulla se non trasportare informazioni. • Il server, tendenzialmente, utilizzerà il cookie come collegamento tra il client (e quindi l'utilizzatore a tutti gli effetti) ed un profilo personalizzato che si sarà creato. • 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 raccogliendo informazioni sulla nostra navigazione all'interno del sito stesso. • Perché si utilizzano i cookies? • profilo personalizzato all'interno dell'applicazione, ad esempio la lingua utilizzata e le impostazioni di visualizzazione. • autenticazione, 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)
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL
2.107	

107

Reti di Elaboratori	I third party cookie ➤ Un uso subdolo (ma alcuni lo giustificano) dei cookie è l'inserimento di cookie nei banner e nelle pubblicità. ➤ 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 profilazione più precisa del navigatore, con effetti discutibili sulla sua privacy. ➤ RFC 2965 esplicitamente proibisce questo tipo di comportamento, che è però largamente ignorato dai produttori di browser e dai fornitori di banner. ➤ Si aggiunga che molte versioni di browser hanno bug che permettono a codice Javascript malevolo, nascosto dentro alle pagine dei browser, di "sniffare" i contenuti dei cookie destinati ad altri domini.
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL
2.108	

108

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

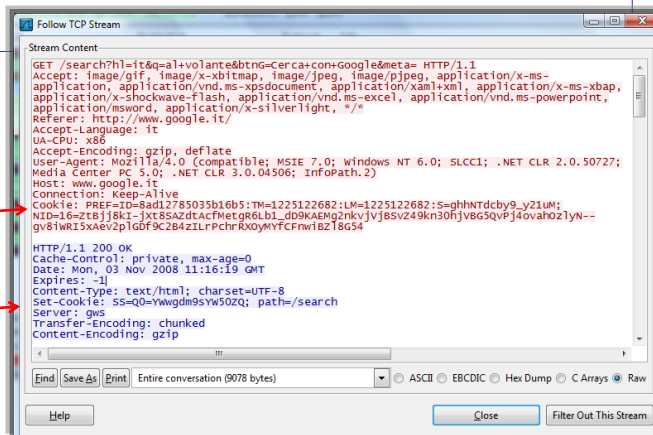
<https://www.youronlinechoices.com/it/>

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



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

2.111

111

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

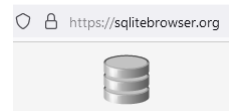
Chapter 2 APPLICATION PROTOCOL

2.112

112

DB Browser for SQLite

The Official home of the DB Browser for SQLite

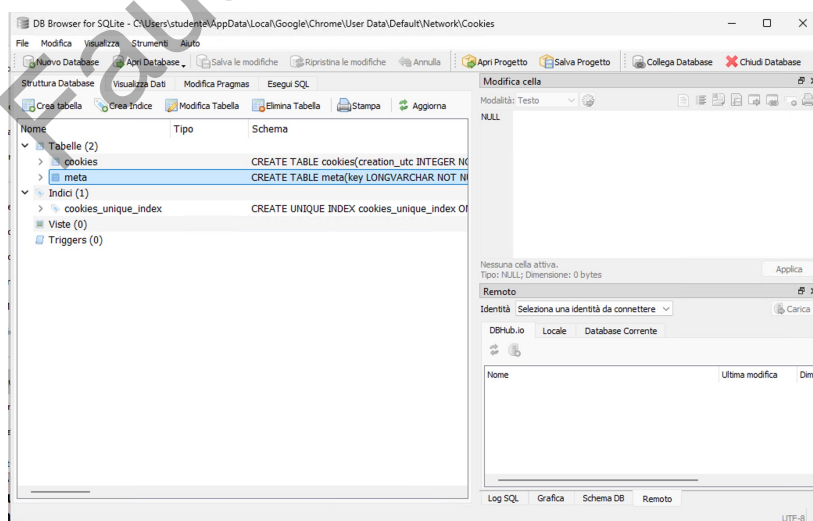


<https://sqlitebrowser.org/dl/>

113

SQLite browser cookies

C:\Users\studente\AppData\Local\Google\Chrome\User Data\Default\Network\Cookies



114

Reti di Elaboratori

SQLite browser cookies

id	name	host_key	top_frame_key	value	encrypted_value	flags
1	__utms	google.com				
2	ANCV_AA673AE12E15B730A49D4C460A60...	ga-zetta.it				
3	DV	www.google.com				
4	IDB	doubleclick.net				
5	IDBID	doubleclick.net				
6	ipflash	ga-zetta.it				
7	PermutiveRandomInt	ga-zetta.it				
8	SOC3	google.com				
9	TSTop	ga-zetta.it				
10	_CFga-zetta_it	ga-zetta.it				
11	_GDFga-zetta_it	ga-zetta.it				
12	_Secure-ENID	google.com				
13	_chKVLST	ga-zetta.it				
14	_cmpAcceptedPurposes	ga-zetta.it				
15	_cpmtClicked	ga-zetta.it				
16	_pio	www.ga-zetta.it				
17	_gcl_ku	ga-zetta.it				
18	_is_get_location	www.ga-zetta.it				
19	_is_get_location_state	www.ga-zetta.it				
20	_js_cmp_mainline	ga-zetta.it				
21	_js_cmpall	ga-zetta.it				
22	api_cache	ga-zetta.it				
23	cacheid	ga-zetta.it				
24	cpmt_ku	ga-zetta.it				
25	eucosment-v2	ga-zetta.it				
26	gptTest	ga-zetta.it				

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.115

115

Reti di Elaboratori

HTTP vs HTTPS

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.116

116

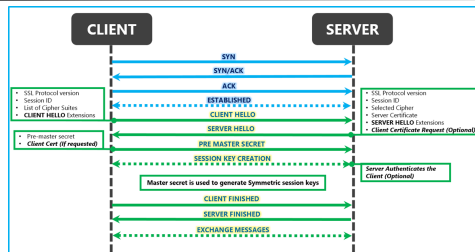
Reti di Elaboratori	Modelli di sicurezza
Ci sono due modi per fornire un trasporto sicuro (non intercettabile): • Usare un'infrastruttura di trasporto sicura ➤ Il protocollo non cambia, ma ogni pacchetto trasmesso nello scambio di informazioni viene gestito in maniera sicura dal protocollo di trasporto • Usare un protocollo sicuro a livello applicazione ➤ Si usa un protocollo anche diverso, che si occupa di gestire la trasmissione delle informazioni. 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 https://) 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.	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL 2.117

117

Reti di Elaboratori	Transport Layer Security TLS
Transport Layer Security TLS Transport Layer Security, o TLS, è un protocollo di sicurezza ampiamente diffuso progettato per semplificare la privacy e la sicurezza dei dati per le comunicazioni su Internet. L'uso principale di TLS è la crittografia della comunicazione tra applicazioni Web e server TLS può essere utilizzato anche per crittografare altre comunicazioni come e-mail, messaggistica e Voice over IP (VoIP) TLS è stato proposto dalla Internet Engineering Task Force (IETF) La versione più recente è TLS 1.3, pubblicata nel 20	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL 2.118

118

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.

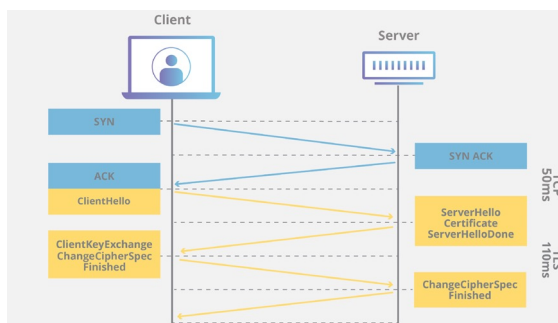
<https://www.cloudflare.com/it-it/learning/ssl/what-happens-in-a-tls-handshake/>

<https://www.evemilano.com/tls-session-resumption/>

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.



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.

l'handshake SSL

Fondamentalmente, l'handshake SSL non è altro che una conversazione tra due parti (**client** e **server**) che vogliono raggiungere lo stesso scopo: proteggere la comunicazione con l'aiuto della crittografia simmetrica. Immagina questa handshake.

Client: Ciao. Voglio stabilire una comunicazione sicura tra noi due. Ecco i miei cipher suits* e la versione compatibile SSL/TLS. Una cipher suite – suite di crittografia è un insieme di algoritmi che aiutano a proteggere una connessione di rete che utilizza Transport Layer Security (TLS) o Secure Socket Layer (SSL). L'insieme di algoritmi che le suite di crittografia contengono di solito include: un algoritmo di scambio di chiavi, un algoritmo di crittografia di massa e un algoritmo di codice di autenticazione dei messaggi (MAC).

Server: Ciao Client. Ho controllato i tuoi cipher suits e la versione SSL/TLS. Possiamo procedere, ecco il mio certificato e la mia chiave pubblica. Controllali.

Client: Fammi verificare il tuo certificato. (Dopo un po' ...) Ok, sembra a posto, ma ho bisogno di verificare la tua chiave privata. Quello che farò sarà generare e cifrare una chiave pre-master (chiave segreta condivisa) usando la tua chiave pubblica. Decifralo usando la tua chiave privata e useremo la chiave master per cifrare e decifrare le informazioni.

Server: Fatto.

[Ora che entrambe le parti sanno con chi stanno parlando, le informazioni trasferite tra di loro saranno protette usando la chiave principale. Tieni presente che una volta terminata la parte di verifica, la crittografia viene eseguita solo tramite la chiave master. Questa è la crittografia simmetrica.]

Client: Ti sto inviando questo messaggio di esempio per verificare che la nostra chiave principale funzioni. Inviarmi la versione decrittografata di questo messaggio. Se funziona, i nostri dati sono in buone mani.

Server: Sì, funziona. Penso che abbiamo raggiunto quello che stavamo cercando.

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.123

123

Sessione https

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	mfausto.amministrazione.unica	antoine.unica.it	TCP	66	2448 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PER
2	0.000380	antoine.unica.it	mfausto.amministrazione.unica	TCP	66	443 → 2448 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK
3	0.000362	mfausto.amministrazione.unica	antoine.unica.it	TCP	54	2448 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
4	0.000893	mfausto.amministrazione.unica	antoine.unica.it	TLSv1.3	571	Client Hello
5	0.000815	antoine.unica.it	mfausto.amministrazione.unica	TCP	60	443 → 2448 [ACK] Seq=1 Ack=518 Win=64128 Len=0
6	0.004151	antoine.unica.it	mfausto.amministrazione.unica	TLSv1.3	1514	Server Hello, Change Cipher Spec, Application Data
7	0.004154	antoine.unica.it	mfausto.amministrazione.unica	TLSv1.3	1844	Application Data, Application Data, Application Data
8	0.004216	mfausto.amministrazione.unica	antoine.unica.it	TCP	54	2448 → 443 [ACK] Seq=518 Ack=2451 Win=2102272 Len=0
9	0.024443	mfausto.amministrazione.unica	antoine.unica.it	TLSv1.3	118	Change Cipher Spec, Application Data
10	0.024994	antoine.unica.it	mfausto.amministrazione.unica	TLSv1.3	357	Application Data
11	0.025140	antoine.unica.it	mfausto.amministrazione.unica	TLSv1.3	357	Application Data
12	0.025169	mfausto.amministrazione.unica	antoine.unica.it	TCP	54	2448 → 443 [ACK] Seq=582 Ack=3057 Win=2101760 Len=0
13	0.025904	mfausto.amministrazione.unica	antoine.unica.it	TCP	66	2449 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PER
14	0.030182	antoine.unica.it	mfausto.amministrazione.unica	TCP	66	443 → 2449 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK
15	0.030231	mfausto.amministrazione.unica	antoine.unica.it	TCP	54	2449 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
16	0.030447	mfausto.amministrazione.unica	antoine.unica.it	TLSv1.3	672	Client Hello
17	0.030713	antoine.unica.it	mfausto.amministrazione.unica	TCP	60	443 → 2449 [ACK] Seq=1 Ack=619 Win=64128 Len=0
18	0.031028	antoine.unica.it	mfausto.amministrazione.unica	TLSv1.3	290	Server Hello, Change Cipher Spec, Application Data, Application
19	0.032074	mfausto.amministrazione.unica	antoine.unica.it	TLSv1.3	118	Change Cipher Spec, Application Data
20	0.032500	antoine.unica.it	mfausto.amministrazione.unica	TLSv1.3	357	Application Data
21	0.084823	mfausto.amministrazione.unica	antoine.unica.it	TCP	54	2449 → 443 [ACK] Seq=683 Ack=548 Win=2101760 Len=0

Provare con il file



antoine_https.pcapng

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.124

124

Reti di Elaboratori

Filtro: TLS

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.125

125

Reti di Elaboratori

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

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

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

2.126

126

Reti di Elaboratori

filtri tss.handshake

Usa tls nei filtri di visualizzazione:

Tutti i messaggi di handshake:
`tls.handshake`

Certificato (contiene la chiave pubblica del certificato del server):
`tls.handshake.type == 11`

Server Key Exchange (server ephemeral public key per DHE/ECDHE, TLS ≤1.2):
`tls.handshake.type == 12`

Client Key Exchange (client ephemeral public key in alcuni scambi, o premaster criptato con la chiave pubblica del server per RSA):
`tls.handshake.type == 16`

Filtrare più tipi insieme:
`tls.handshake.type in {1, 2, 11, 12, 16}`
`tls.handshake.type == 11 || tls.handshake.type == 12 || tls.handshake.type == 16`

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.127

127

Reti di Elaboratori

per verificare lo scambio di certificati

The image shows a Wireshark capture of a TLS handshake. The packet list at the top shows a Certificate (type 11) and a Server Key Exchange (type 12). The packet details pane shows the structure of the Certificate, including the public key and signature. The packet bytes pane shows the raw data of the certificate.

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.128

128

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 OCSP per confermare la validità attuale dei certificati

[Mostra certificati...](#)

[Dispositivi di sicurezza...](#)

about:preferences#privacy

Gestione certificati

Certificati personali Decisioni di autenticazione Persone Server **Autore**

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

Nome certificato	Dispositivo di sicurezza
> AC Camerfirma S.A.	
> AC Camerfirma SA CIF A82743287	
> ACIV	
> Adialis S.p.A./03358520967	
> AdTrust AB	
> AffirmTrust	
> Agence Nationale de Certification Electronique	
> Agencia Catalana de Certificacio (NIF Q-8901176-I)	
> Amazon	
> ANF Autoridad de Certificacion	
> Aspeco Data Systems S.A.	
> Atos	
> Autoridad de Certificacion Firmaprofesional CIF A626...	
> Arxist Web/Mail Shield	

Visualizza... Modifica attendibilità... **Importa...** Esporta Elimina o considera inutilizzabile...

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.129

129

Reti di Elaboratori

Chrome – certificati locali

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.130

130

Reti di Elaboratori

Chrome – Certification Authority

Gestione certificati

Chrome Root Store

Chrome Root Store contiene certificati di autorità di certificazione attendibili per Chrome Root Program ed è sottoposto a revisione continua. [Scopri di più](#)

Certificati attendibili

Actalis Authentication Root CA

Amazon Root CA 3

Amazon Root CA 2

Amazon Root CA 1

Amazon Root CA 4

Certum Trusted Network CA

Certum Trusted Network CA 2

Atos TrustedRoot 2011

Autoridad de Certificación Firmaprofesional CIF A62634068

Esporta

Fausto Marcantoni
Chapter 2 APPLICATION PROTOCOL
2.131

131

Reti di Elaboratori

Wireshark Tutorial: Decrypting HTTPS Traffic

Wireshark Tutorial: Decrypting HTTPS Traffic

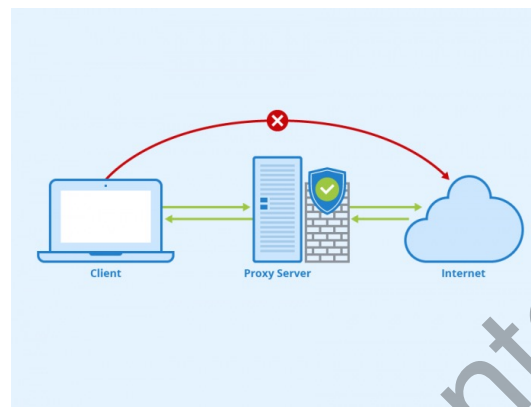
<https://unit42.paloaltonetworks.com/wireshark-tutorial-decrypting-https-traffic/>

Fausto Marcantoni
Chapter 2 APPLICATION PROTOCOL
2.132

132

Web Cache (proxy server)

Web Cache - Proxy server



Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

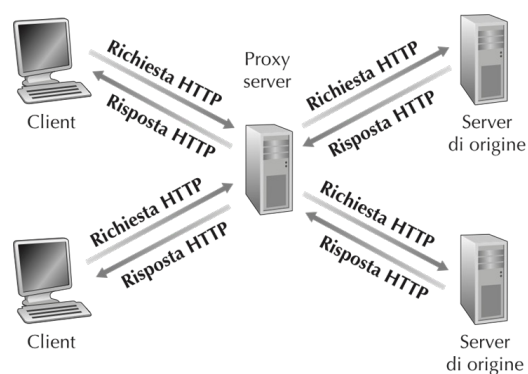
2.133

133

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



Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.134

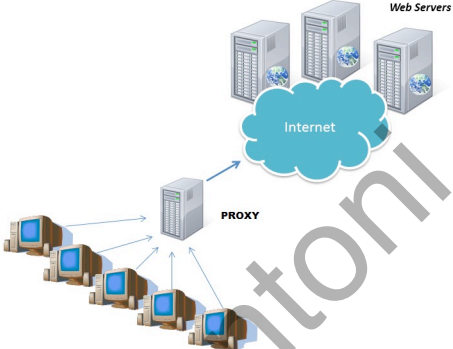
134

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)



Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.135

135

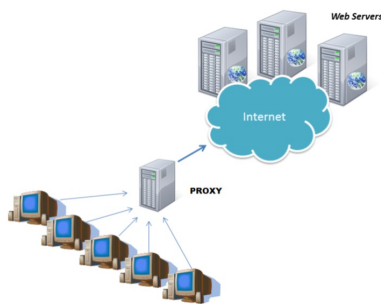
Reti di Elaboratori

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

2.136

136

Reti di Elaboratori

GET condizionale (conditional GET)

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

client

server

http request msg

If-modified-since: <date>

http response

HTTP/1.0

304 Not Modified

object not modified

http request msg

If-modified-since: <date>

http response

HTTP/1.1 200 OK

...

<data>

object modified

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

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.137

137

Reti di Elaboratori

Esercitazione

Esercitazione con

WIRESHARK

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: 128.119.245.12 (128.119.245.12)

Transmission Control Protocol, Src Port: lupa (1212), Dst Port: 80

Hypertext Transfer Protocol

GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1

Accept: */*

Accept-Language: th

Accept-Encoding: gzip, deflate

If-Modified-Since: Fri, 13 Jun 2008 04:45:01 GMT

If-None-Match: "d6c96-173-f0cbad40"

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.138

138

Reti di Elaboratori

http matches "If-Modified-Since"

The screenshot shows a Wireshark packet capture of an HTTP GET request. The packet list pane shows several packets, with packet 15918 selected. The packet details pane shows the HTTP request structure, including the 'If-Modified-Since' header. The packet bytes pane shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
15891	56.988436	mfausto.local	pros.unica.m.it	HTTP	1006	GET / HTTP/1.1
15918	57.020023	mfausto.local	pros.unica.m.it	HTTP	920	GET /wp-content/cache/minify/
16047	57.030561	mfausto.local	pros.unica.m.it	HTTP	936	GET /wp-content/cache/minify/
16098	57.035856	mfausto.local	pros.unica.m.it	HTTP	936	GET /wp-content/cache/minify/
16100	57.035994	mfausto.local	pros.unica.m.it	HTTP	936	GET /wp-content/cache/minify/
16102	57.036118	mfausto.local	pros.unica.m.it	HTTP	920	GET /wp-content/cache/minify/

Frame 15918: 920 bytes on wire (7360 bits), 920 bytes captured (7360 bits) on interface \Device\NPF_{1C45D50C-D191-43A0-9A6F-DFDA23ECCEA1}, Ethernet II, Src: mfausto.local (d8:50:e6:0a:e6:1d), Dst: pros.unica.m.it (00:50:56:9a:25:59), Internet Protocol Version 4, Src: mfausto.local (193.205.92.201), Dst: pros.unica.m.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\r\nHost: pros.unica.m.it\r\n\r\nConnection: keep-alive\r\n\r\nUpgrade-Insecure-Requests: 1\r\n\r\nUser-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/106.0.0.0\r\n\r\nAccept: 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\r\nAccept-Encoding: gzip, deflate\r\n\r\nAccept-Language: it,it-IT;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6\r\n\r\n[Truncated] cookie: cookieLawInfo-checkbox-necessary=yes; cookieLawInfo-checkbox-analytics=yes; cookieLawInfo-checkbox-functional=yes; cookieLawInfo-checkbox-essential=yes\r\n\r\nIf-Modified-Since: Wed, 06 Apr 2022 07:26:28 GMT\r\n\r\n\r\n[Full request URI: http://pros.unica.m.it/]\r\n\r\n[HTTP request 1/4]\r\n\r\n[Response in frame: 15910]\r\n\r\n[Next request in frame: 15918]

0000 00 50 56 9a 25 59 d8 50 e6 0a e6 1d 08 00 45 00 .PV.XY.P.....E:

wireshark_EthernetBONITI.pcapng

Pacchetti: 17838 - visualizzati: 22 (0.1%) - scartati: 0 (0.0%) - Profilo: Default

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.139

139

Reti di Elaboratori

Cercare una stringa con wireshark

The screenshot shows the Wireshark interface with the 'Cerca' (Search) menu open. The search string 'www.gazzeta.it' is entered in the search bar. The search results show several packets containing the string.

No.	Time	Source	Destination	Protocol	Length	Info
15891	56.988436	mfausto.local	pros.unica.m.it	HTTP	1006	GET / HTTP/1.1
15918	57.020023	mfausto.local	pros.unica.m.it	HTTP	920	GET /wp-content/cache/minify/
16047	57.030561	mfausto.local	pros.unica.m.it	HTTP	936	GET /wp-content/cache/minify/
16098	57.035856	mfausto.local	pros.unica.m.it	HTTP	936	GET /wp-content/cache/minify/
16100	57.035994	mfausto.local	pros.unica.m.it	HTTP	936	GET /wp-content/cache/minify/
16102	57.036118	mfausto.local	pros.unica.m.it	HTTP	920	GET /wp-content/cache/minify/

Frame 15918: 920 bytes on wire (7360 bits), 920 bytes captured (7360 bits) on interface \Device\NPF_{1C45D50C-D191-43A0-9A6F-DFDA23ECCEA1}, Ethernet II, Src: mfausto.local (d8:50:e6:0a:e6:1d), Dst: pros.unica.m.it (00:50:56:9a:25:59), Internet Protocol Version 4, Src: mfausto.local (193.205.92.201), Dst: pros.unica.m.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\r\nHost: pros.unica.m.it\r\n\r\nConnection: keep-alive\r\n\r\nUpgrade-Insecure-Requests: 1\r\n\r\nUser-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/106.0.0.0\r\n\r\nAccept: 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\r\nAccept-Encoding: gzip, deflate\r\n\r\nAccept-Language: it,it-IT;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6\r\n\r\n[Truncated] cookie: cookieLawInfo-checkbox-necessary=yes; cookieLawInfo-checkbox-analytics=yes; cookieLawInfo-checkbox-functional=yes; cookieLawInfo-checkbox-essential=yes\r\n\r\nIf-Modified-Since: Wed, 06 Apr 2022 07:26:28 GMT\r\n\r\n\r\n[Full request URI: http://pros.unica.m.it/]\r\n\r\n[HTTP request 1/4]\r\n\r\n[Response in frame: 15910]\r\n\r\n[Next request in frame: 15918]

0000 00 50 56 9a 25 59 d8 50 e6 0a e6 1d 08 00 45 00 .PV.XY.P.....E:

wireshark_EthernetBONITI.pcapng

Pacchetti: 17838 - visualizzati: 22 (0.1%) - scartati: 0 (0.0%) - Profilo: Default

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

ftp: File transfer protocol

- 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

2.143

143

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

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.144

144

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

2.145

145

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 (stes.cs.unicam.it:(none)): 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

2.146

146

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
250 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

2.147

147

Reti di Elaboratori

Esempio ftp via browser

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

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

Vai alla cartella superiore

Nome	Dimensione	Ultima modifica
File: README	18 KB	18/06/2019 21:14:00
dev	2 KB	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

2.148

148

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

2.149

149

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 su 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)

[Richiedi l'autorizzazione di app attraverso un firewall](#)

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

2.150

150

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>

netsh

```
C:\Users\fausto.m\fausto>netsh
netsh>help

Sono disponibili i seguenti comandi :

Comandi in questo contesto:
..          - Raggiunge un livello di contesto.
?          - Visualizza un elenco di comandi.
abort      - Ignora qualsiasi modifica effettuata in modalità offline.
add        - Aggiunge una voce di configurazione a un elenco di voci.
advfirewall - Passa al contesto 'netsh advfirewall'.
alias      - Aggiunge un alias.
branchcache - Passa al contesto 'netsh branchcache'.
bridge     - Passa al contesto 'netsh bridge'.
bye        - Chiude il programma.
commit     - Apporta le modifiche effettuate in modalità offline.
delete     - Elimina una voce di configurazione da un elenco di voci.
dhcpcclient - Passa al contesto 'netsh dhcpcclient'.
dnsclient  - Passa al contesto 'netsh dnsclient'.
dump       - Visualizza uno script di configurazione.
exec       - Esegue un file di script.
exit       - Chiude il programma.
firewall   - Passa al contesto 'netsh firewall'.
help       - Visualizza un elenco di comandi.
http       - Passa al contesto 'netsh http'.
interface - Passa al contesto 'netsh interface'.
ipsec      - Passa al contesto 'netsh ipsec'.
lan        - Passa al contesto 'netsh lan'.
mbn        - Passa al contesto 'netsh mbn'.
namespace - Passa al contesto 'netsh namespace'.
netio      - Passa al contesto 'netsh netio'.
offline    - Imposta 'offline' come modalità corrente.
```

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

Reti di Elaboratori

Altri sistemi client ftp

 SSH Secure File Transfer Client
Collegamento
2 KB

 SSH Secure Shell Client
Collegamento
1 KB

PuTTY



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

<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



<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

2.153

153

Reti di Elaboratori

ftp su Windows 10 - problem



```

C:\WINDOWS\system32>ftp 193.205.92.195
Connesso a 193.205.92.195.
220 (vsFTPd 3.0.3)
200 Always in UTF8 mode.
Utente (193.205.92.195:(none)): anonymous
331 Please specify the password.
Password:
230 Login successful.
ftp> dir
200 PORT command successful. Consider using PASV.
425 Failed to establish connection.
ftp> bye
221 Goodbye.
C:\WINDOWS\system32>
    
```

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

2.154

154

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>

ftp - wireshark

No.	Time	Source	Destination	Protocol	Length	Info
6	0.005564	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	192.168.49.1	192.168.49.128	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	192.168.49.1	192.168.49.128	FTP	88	Response: 331 Please specify the password
22	9.405190	192.168.49.1	192.168.49.128	FTP	69	Request: PASS password
25	9.410733	192.168.49.1	192.168.49.128	FTP	77	Response: 230 Login successful.
28	11.432620	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_{B411EE2-1A51-4D70-9158-D4511FA1E878}, id 0
 > Ethernet II, Src: VMware_52:47: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.....)bG...E
 VMware Network Adapter VMnet8: <live capture in progress> Pacchetti: 52 - visualizzati: 9 (17.3%) Profilo: Default

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
2.157	

157

Reti di Elaboratori	TFTP (Trivial File Transfer Protocol)
TFTP è un protocollo molto semplice utilizzato per trasferire file tra host utilizzando UDP. definito nella RFC 1350 protocollo di trasporto UDP (<i>User Datagram Protocol</i>). La porta sulla quale è in ascolto il server TFTP server è la 69. ❖ molto semplice ❖ meno funzioni rispetto a FTP ❖ non può leggere directory ❖ non è provvisto di autenticazione ❖ utilizzo è limitato TFTP viene solitamente usato per trasferire file tra un computer ed un altro dispositivo come router o switch in ambito LAN.	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL
2.158	

158

Reti di Elaboratori

TFTP (Trivial File Transfer Protocol)

- TFTP trasmette pacchetti con una **lunghezza fissa di 512 byte**.
- Un pacchetto avente una dimensione inferiore rappresenta l'**ultimo pacchetto trasmesso**.
- I pacchetti dati inviati vengono memorizzati in un buffer fino alla ricezione della avvenuta accettazione da parte dell'host remoto.
- In caso di mancata conferma della ricezione entro un determinato tempo di un pacchetto, quest'ultimo viene ritrasmesso.

La modalità di trasferimento dati di TFTP è di due tipi:

- **NETASCII** per i file di testo;
- **OCTET** per i file binari.

I pacchetti utilizzati durante una sessione TFTP sono di cinque tipi:

- **RR**: Read Request (*Richiesta di lettura*);
- **WR**: Write Request (*Richiesta di scrittura*);
- **DATA**: Dati;
- **ACK**: Acknowledgment (*Accettazione*);
- **ERR**: Errore;

Le fasi di una sessione TFTP:

1. Il client contatta il server inviando un pacchetto di tipo **RR** (*richiesta di lettura*) o **WR** (*richiesta di scrittura*);
2. Il server, se accetta la connessione, risponde inviando/ricevendo pacchetti **DATA** di 512 byte. Per ogni pacchetto inviato/ricevuto regolarmente viene inviato/ricevuto un **ACK** altrimenti un **ERROR**;
3. I pacchetti vengono trasferiti finché la loro lunghezza non è inferiore a 512 byte;
4. Termine della connessione;

Fausto Marcantoni
Chapter 2 APPLICATION PROTOCOL
2.159

159

Reti di Elaboratori

esercitazioni

internet reti sicurezza

Esercitazioni

Fausto Marcantoni
Chapter 2 APPLICATION PROTOCOL
2.160

160

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
2.161	

161

Reti di Elaboratori	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL
2.162	

162

Reti di Elaboratori

Un po' di storia ... di Outlook

Nel 1997, Microsoft ha acquisito Hotmail, uno dei primi servizi di posta elettronica gratuiti basati sul web. Hotmail divenne MSN Hotmail, poi Windows Live Hotmail e infine Outlook.com. Outlook.com non è un client di posta elettronica desktop ma un servizio di posta elettronica gratuito.

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.163

163

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

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.164

164

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

2.165

165

Reti di Elaboratori

Migliori Client di Posta Elettronica Gratuiti

Migliori Client di Posta Elettronica Gratuiti 2024

Thunderbird.

Gmail/Google WorkspaceWorkspace.

eM Client.

Mailbird.

Spark.

Mailspring.

Shift.

Spike.

<https://canarymail.io/blog/it/clienti-email-gratuiti/>

<https://kinsta.com/it/blog/alternative-ad-outlook/>

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.166

166

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

2.167

167

Reti di Elaboratori

I 15 migliori server di posta elettronica open source per Web nel 2024

Nome	Descrizione
Forward Email	E-mail crittografate incentrate sulla privacy per te. Siamo il servizio di posta elettronica di riferimento per centinaia di migliaia di creatori, sviluppatori e aziende. Invia e ricevi email come tu@tuodominio.com con il tuo dominio personalizzato o utilizza uno dei nostri.
Docker Mailserver	docker-mailserver, o DMS in breve, è un server di posta fullstack pronto per la produzione ma semplice (SMTP, IMAP, LDAP, antispam, antivirus, ecc.).
Maddy Mail Server	Maddy Mail Server implementa tutte le funzionalità necessarie per eseguire un server di posta elettronica.
Mail-in-a-Box	Riprendi il controllo della tua posta elettronica con questo server di posta in una scatola facile da implementare.
mailcow	mailcow: dockerized è una suite groupware/e-mail open source basata su docker.
Dovecot	Dovecot è una scelta eccellente sia per installazioni piccole che grandi.
WildDuck	WildDuck è un moderno software server di posta per IMAP e POP3.
Exim	Exim è un agente di trasferimento messaggi (MTA) originariamente sviluppato presso l'Università di Cambridge per l'utilizzo su sistemi Unix connessi a Internet.
Postfix	Il server di posta di Wietse Venema è nato presso la ricerca IBM come alternativa al diffusissimo programma Sendmail.
chasquid	chasquid è un server SMTP (e-mail) incentrato sulla semplicità, sicurezza e facilità d'uso.
Haraka	Un server SMTP moderno, performante e flessibile.
Postal	Postal è un server di posta completo e ricco di funzionalità per l'utilizzo da parte di siti Web e server Web.
ZoneMTA	ZoneMTA fornisce un controllo granulare sull'instradamento di messaggi diversi.
kaila	Strumento di newsletter open source.
listmonk	Gestore di newsletter e mailing list self-hosted.

<https://forwardemail.net/it/blog/open-source/web-email-server>

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.168

168

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" Δtime (minuti)
5. Se l'invio fallisce dopo "m" Δtime (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

2.169

169

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

2.170

170

Reti di Elaboratori

Esempio di interazione SMTP

S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
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

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.171

171

Reti di Elaboratori

SMTP: considerazioni

- Connessioni TCP persistenti
- Richiede che il messaggio (header & corpo) sia in **formato ascii 7-bit**
- Alcune sequenze di caratteri non consentite (es., CRLF.CRLF).
Conseguenza: il messaggio deve essere codificato
- Il server smtp usa CRLF.CRLF per determinare la fine del messaggio

Confronto con http

- http: pull protocol
- email: push protocol
- Entrambi usano un'interazione mediante comandi/risposta in testo ASCII e codici di stato
- http: ogni oggetto incapsulato nel messaggio di risposta
- smtp: tutti gli oggetti sono inviati mediante un unico messaggio (eventualmente in più parti)

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.172

172

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
2.173

173

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.mrwebmaster.it/dev/base64-encoder-decoder.html>

Fausto Marcantoni
Chapter 2 APPLICATION PROTOCOL
2.174

174

Reti di Elaboratori

Image <-> base64

base64

```

jVBRwKGGgAAAAANSUeUgAAAZAAAGQCA
VAAAGAvz2MAAAgEIEQvRkUv9B5gJR5n2
qRzJOC/ObJrMedFzsrT0S5wx2p1s
/2GCDdjCEcSwP3DwG7jAgU0wR7rjTLCxwQ
bsdV7r3FUGmS53+ccN8qNpP
/5qrw6q1staaX5C57H42Jquqr1r99pfe
z9BR8xuG1SwJTFH3jPg0RrWvzQegm8hP
1Nr6FQcRqkdPthdPtgcV1Af+b3tts9lnz
Rmshlg5SkYDmAp8kxGalsSwJjAwPwe8bwM
JQAPz0g8B4+j7
/udxP8KrmwZue8Bc3H0DZkus1nEtCSQ1A
QTAkhkBNM117BDw+yTARA1hZhgQoeDoR1
6Qo9j59KYmHjSPmYuaHQzGZyIPLwAQK
U5Bz215AXAw409BAbRatJmqdnYHc4K3ge
ZBHjT4U7LVKYVq1k5m6o5SABku31LF0Pg

```

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 Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.175

175

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)
32kadtcm (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 Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.176

176

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

2.177

177

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

2.178

178

Reti di Elaboratori

Esempio opzioni messaggio

Opzioni messaggio

Impostazioni messaggio

Priorità: Normale

Riservatezza: Normale

Protezione

☐ Crittografia contenuto e allegati del messaggio
☐ Aggiungi la firma digitale al messaggio in uscita
☐ Richiedi conferma S/MIME per il messaggio

Opzioni di verifica

☐ Richiedi conferma di recapito per il messaggio
☐ Richiedi conferma di lettura per il messaggio

Opzioni di recapito

Invia risposte a: Amedeo Belfiore; Lista del LUG di Camerino (MC)

Scadenza: Nessuna

0.00

Contatti...

Categorie...

Intestazioni Internet

Received: from picard.linux.it ([213.254.12.146] verified by uncam.it (CommuniGate Pro SMTP 5.1c.4) with ESMTPS id 12548883 for fausto.marcantoni@unicam.it; Tue, 17 Oct 2006 13:43:11 +0200

Received: from picard.linux.it (localhost [127.0.0.1]) by picard.linux.it (Postfix) with ESMTP id 2627C4903 for <fausto.marcantoni@unicam.it>; Tue, 17 Oct 2006 13:37:55

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.179

179

Reti di Elaboratori

Gmail - Opzioni - Mostra originale

oggetto della mail

Posta in arrivo x

Fausto Marcantoni a me

TESTO DELLA MAIL

Fai clic qui per rispondere o inoltrare il messaggio

11:56 (4 minuti fa)

Rispondi

Inoltra

Apri chat con Fausto Marcantoni

Filtra i messaggi di questo tipo

Stampa

Elimina questo messaggio

Blocca "Fausto Marcantoni"

Segnala spam

Segnala phishing

Mostra originale

Il testo del messaggio è incomprensibile?

Traduci messaggio

Segna come da leggere

9CHKiw81TyVqoQEtfQ==

2016 02:56:34 -0700 (PDT)

--001a11c151a0001e8f053f34d323

Content-Type: text/html; charset=UTF-8

<div dir="ltr">TESTO DELLA MAIL
</div>

--001a11c151a0001e8f053f34d323--

Fausto Marcantoni

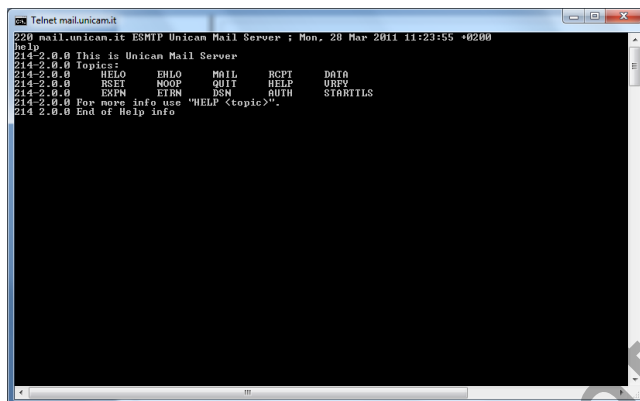
Chapter 2 APPLICATION PROTOCOL

2.180

180

Esempio mail

telnet 90.147.42.137 25



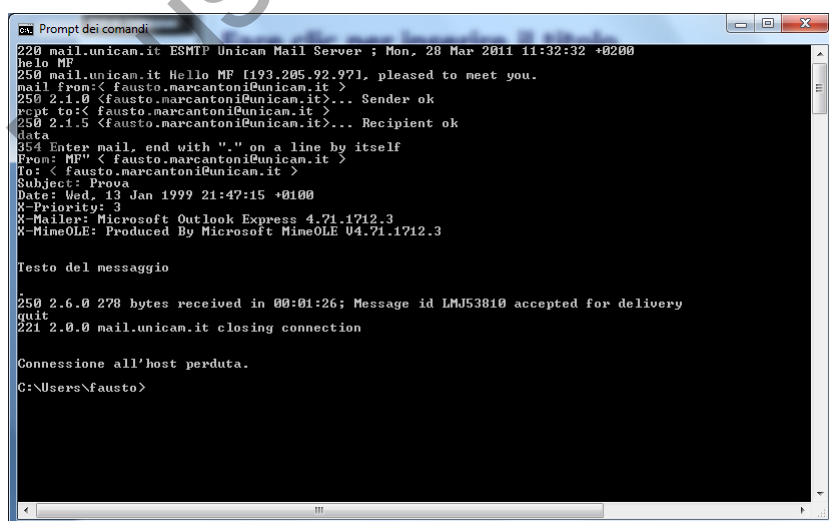
```

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 Topical:
214-2.0.0      EHLO      MAIL      RCPT      DATA
214-2.0.0      RSET      NOOP      QUIT      HELP      URFBY
214-2.0.0      EXPN      ETIM      RSN      AUTH      STARTTLS
214-2.0.0 For more info use "HELP <topic>".
214 2.0.0 End of Help info

```

181

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 V4.71.1712.3

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>

```

182

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

quit
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

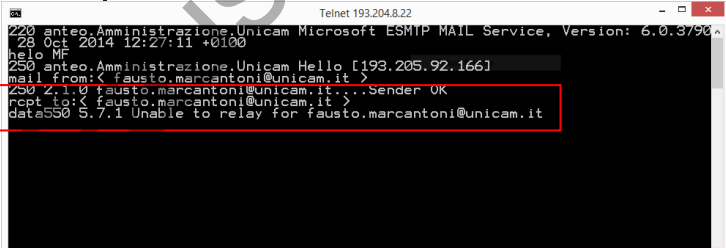
Chapter 2 APPLICATION PROTOCOL

2.183

183

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".

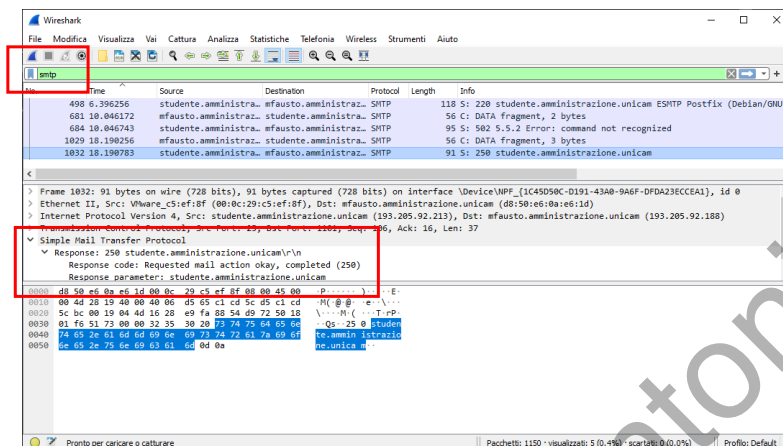
Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.184

184

smtp <-> wireshark



Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.185

185

ESMTP command

```

faustomarcantoni@MacBook-Pro-di-Fausto ~ % telnet smtp.libero.it 25
Trying 213.209.1.146...
Connected to smtp.libero.it.
Escape character is '^['.
220 smtp-17.iol.local smtp-17.iol.local ESMTP server ready
HELP
214-This is smtp-17.iol.local ESMTP service help
214-To contact postmaster send email to postmaster@italianonline.it.
214-For local information send email to Postmaster at your site.
214 end of HELP info
quit
221 smtp-17.iol.local smtp-17.iol.local closing connection
Connection closed by foreign host.
faustomarcantoni@MacBook-Pro-di-Fausto ~ %

```

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

2.186

186

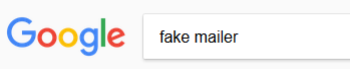
Reti di Elaboratori

Fake Mailer


<https://emkei.cz/>


<http://anonymailer.net/>


mailsend
 Send mail via SMTP protocol from command line
 <https://code.google.com/p/mailsend/>
<https://github.com/muquit/mailsend>


 fake mailer

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.187

187

Reti di Elaboratori

Test <http://anonymailer.net>

test zorro Posta in arrivo x
 zorro <zorro@yyyy.com>
 s me

Inglese Italiano Traduci messaggio


 Free anonymous email service!
Anonymailer
Anonymailer.net

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

2.188

188

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 Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.189

189

Reti di Elaboratori

Protocolli di accesso alla posta

```

graph LR
    Alice[Agente di Alice] -- SMTP --> AliceServer[Server di posta di Alice]
    AliceServer -- SMTP --> BobServer[Server di posta di Bob]
    BobServer -- "POP3, IMAP o HTTP" --> Bob[Agente di Bob]
  
```

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 Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.190

190

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): 995 Use Defaults

☒ This server requires an encrypted connection (SSL)

Outgoing server (SMTP): 587

Use the following type of encrypted connection: Auto

```

[gnail-pop3]
client = yes
accept = 127.0.0.1:110
connect = pop.gmail.com:995

[gnail-imap]
client = yes
accept = 127.0.0.1:143
connect = imap.gmail.com:993

[gnail-smtp]
client = yes
accept = 127.0.0.1:25
connect = smtp.gmail.com:465

```

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.191

191

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

2.192

192

Reti di Elaboratori	Il protocollo IMAP
IMAP <i>Internet Message Access Protocol</i> <i>Interactive Mail Access Protocol</i> è un protocollo di comunicazione per la ricezione di e-mail. Il significato "Interactive Mail Access Protocol" è stato valido fino alla versione 3, dalla quarta in poi è cambiato in "Internet Message Access Protocol" Il protocollo è stato inventato da Mark Crispin nel 1986 come alternativa più moderna all'utilizzatissimo POP.	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL
2.193	

193

Reti di Elaboratori	Il protocollo IMAP
Accesso alla posta sia online che off-line 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. Più utenti possono utilizzare la stessa casella di posta 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. Accesso a molteplici caselle di posta sul server 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. Possibilità di fare ricerche sul server 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. Password criptate 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.	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL
2.194	

194

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

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.

Reti di Elaboratori	Esercitazione
---------------------	----------------------

internet reti sicurezza

Esercitazioni

Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL	2.197
-------------------	--------------------------------	-------

197

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	2.198
-------------------	--------------------------------	-------

198

Reti di Elaboratori	DNS: Domain Name System
Come identifico le persone: - molte mezzi di identificazione: CF, nome, # Passaporto Come identifico computer, host, router, ... : - Indirizzi IP (32 bit) – usati per indirizzare i datagrammi IP "Nome", es., dida.cs.unicam.it – usati dagli utenti <u>Risposta:</u> corrispondenza tra indirizzo IP e nome?	
Domain Name System Database distribuito implementato come una gerarchia di molti <i>name server</i> Protocollo applicativo usato da host, router, name server per comunicare allo scopo di risolvere (tradurre) i nomi in indirizzi IP - Funzione di base di Internet implementata come protocollo applicativo - La complessità trasferita al "bordo" della rete	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL 2.199

199

Reti di Elaboratori	Name server DNS
Perché non un server DNS centralizzato? ★ Minor tolleranza ai guasti ★ Traffico eccessivo ★ Database centrale troppo distante in molti casi ★ Scarsa scalabilità!	
■ Nessun name server contiene tutte le associazioni nome simbolico/indirizzo IP Name server locali : - Ogni ISP o compagnia ha un <i>name server locale (default)</i> - La richiesta di traduzione (mapping) di un host è prima rivolta al name server locale Name server di riferimento (assoluto authoritative): - Per un host: per definizione è quello che è sempre in grado di eseguire la traduzione (mapping) nome simbolico/indirizzo IP dell' host	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL 2.200

200

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.

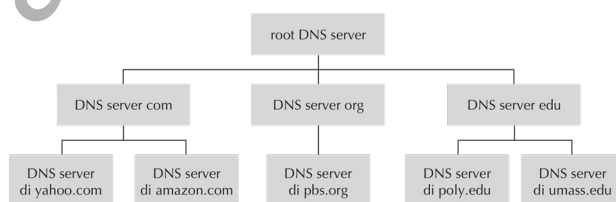
Il Primario ha i dati effettivi, mentre i **Secondari ne ha una copia** (Transfer Zone) ricevuta dal Primario.

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**

Gerarchia di server DNS



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

2.203

203

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

2.204

204

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

2.205

205

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

2.206

206

Reti di Elaboratori

Whois - italia

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

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.207

207

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

```

graph TD
    Root[root name server]
    Local[local name server  
dns.eurecom.fr]
    Auth[authoritative name server  
dns.umass.edu]
    Host1[Host che inoltra la richiesta  
surf.eurecom.fr]
    Host2[gaia.cs.umass.edu]

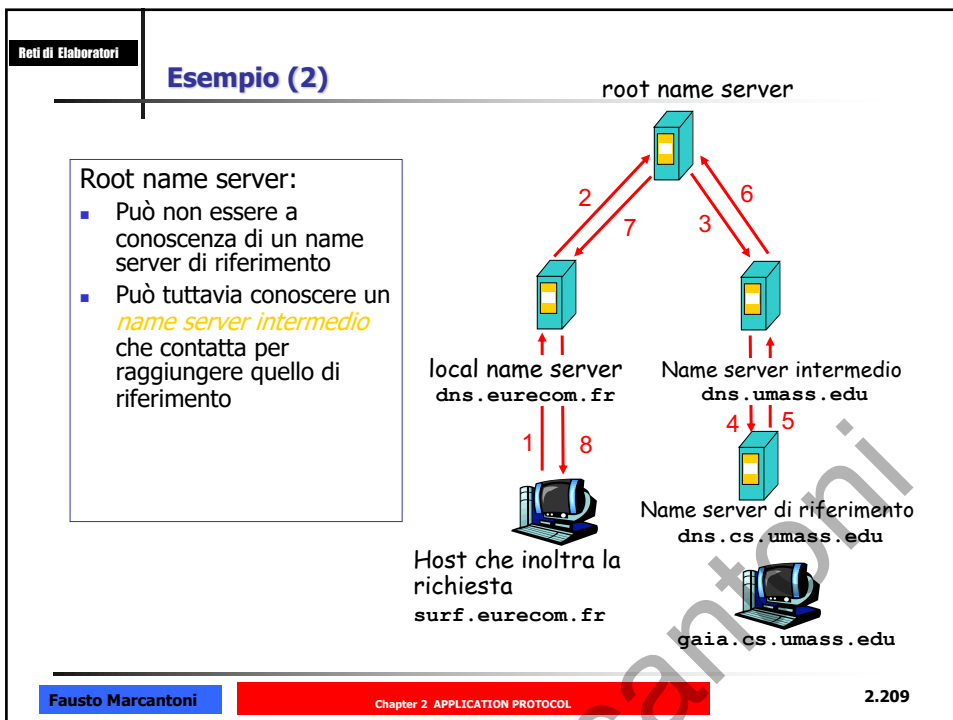
    Host1 -- 1 --> Local
    Local -- 2 --> Root
    Root -- 3 --> Auth
    Auth -- 4 --> Root
    Root -- 5 --> Local
    Local -- 6 --> Host1
  
```

Fausto Marcantoni

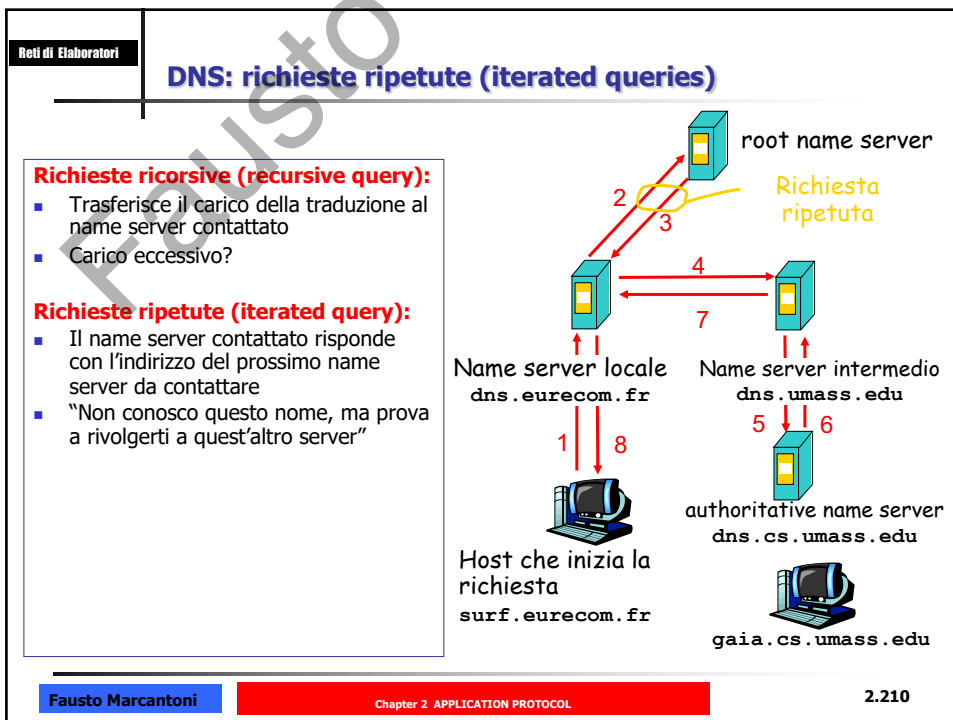
Chapter 2 APPLICATION PROTOCOL

2.208

208



209

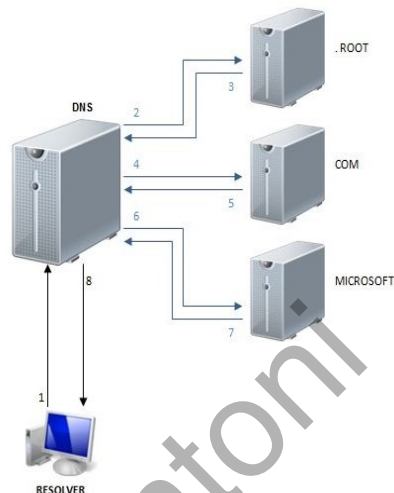


210

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	DNS riepilogo
Quando un server DNS riceve un'interrogazione, verifica se il nome è relativo alla propria zona di autorità: 1. se sì, converte il nome in un indirizzo in base alle informazioni del proprio database, inserisce l'indirizzo nella risposta e la spedisce al client 2. se non può risolvere completamente il nome, esamina la modalità specificata nell'interrogazione: A. Se è ricorsiva, contatta un server di livello superiore che possa risolvere il nome, e riporta al client la risposta ottenuta B. Se è iterativa, si limita a comunicare al client il nome del server a cui rivolgersi	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL
2.213	

213

Reti di Elaboratori	DNS: caching e aggiornamento
■ Quando un qualsiasi name server apprende una traduzione la memorizza localmente (caching) ■ Le traduzioni memorizzate nella cache (cache entries) scadono (timeout) dopo un certo tempo (di solito un paio di giorni) ■ Se possibile, richieste successive vengono servite usando la traduzione presente in cache ■ I meccanismi di aggiornamento/modifica in studio da parte dell' IETF ■ RFC 2136 ■ http://www.ietf.org/html.charters/dnsind-charter.html	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL
2.214	

214

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 AerioConnect		✗
	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 Irpinia 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

2.215

215

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

2.216

216

Reti di Elaboratori

Recursive/Iterative Queries in DNS



Recursive/Iterative Queries in DNS

https://media.pearsoncmg.com/curriculum/intl/01ab/9788891902559_KUROSU_BOSS/studente/interactive%20animations/recursive-iterative-queries-in-dns/index.html

Fausto Marcantoni
Chapter 2 APPLICATION PROTOCOL
2.217

217

Reti di Elaboratori

Record DNS

DNS: database distribuito che memorizza Resource Record (RR)

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

- Tipo=A**
 - nome è il nome dell' host
 - valore è l' indirizzo IP
- Tipo=CNAME**
 - nome è un alias di qualche nome reale ("canonico")
 - valore è il nome canonico
- Tipo=NS**
 - nome è il dominio (es. foo.com)
 - valore è l'indirizzo IP del name server di riferimento per questo dominio
- Tipo=MX**
 - valore è il nome di un mailserver associato a nome

Fausto Marcantoni
Chapter 2 APPLICATION PROTOCOL
2.218

218

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 CNAME

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

2.219

219

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	sicetsimpliciter.cs.unicam.it.
28.92.205.193.in-addr.arpa.	PTR	Default	cvs.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

2.220

220

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	geronaccs.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	sicetsimpliciter.cs.unicam.it
193.205.92.5	Default	www.cs.unicam.it	193.205.92.59	Default	shab.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
2.221

221

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
2.222

222

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 aggiuntivi	
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

2.223

223

Reti di Elaboratori

DNS – Esempio query

```

61 8.697688 193.205.92.161 193.205.92.4 DNS Standard query A www.unina.it
62 8.700564 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
65 9.084314 143.225.172.36 193.205.92.161 TCP http > 2599 [SYN, ACK] Seq=0 Ack=1 win=1654
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)
      ....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....Di...E.
0010 00 2a 42 6f 00 00 80 11 bc 03 c1 cd 5c a1 c1 c6 .: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 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

2.224

224

Reti di Elaboratori

DNS – Esempio response

```

62 8.700564 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 MSS=
64 8.856788 3comeuro.04:09:69 spanning-tree-(for STP RST. Root = 32768/00:06:53:11:60:00 Cost = 4. Port
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 9.084549 143.205.92.161 143.225.172.36 HTTP GET / HTTP/1.1

```

Flags: 0x8180 (Standard query response, No error)
. = Response: Message is a response
.000 0... .. = Opcode: Standard query (0)
.... 0... .. = Authoritative: Server is not an authority for domain
.... .0... .. = Truncated: Message is not truncated
.... .1... .. = Recursion desired: Do query recursively
.... .1... .. = Recursion available: Server can do recursive queries
.... .0... .. = 2: reserved (0)
.... .0... .. = Answer authenticated: Answer/authority portion was not authenticated by the server
.... .0000 = 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 10 c0 01 00 01 00 00 00 f3 d9 00 04 c0 83 ..KX.....
0090 1c 01 c0 3a 00 01 00 01 00 00 f3 d9 00 04 c0 83 ..KX.....
00a0 1c 07

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.225

225

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

2.226

226

nslookup - esempio

```

C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\mf>nslookup www.unicam.it
Server: winserv2.informatica.unicam.it
Address: 193.205.92.4

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

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

Nome:      web.unicam.it
Address:   193.204.8.30

C:\Documents and Settings\mf>_

```

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

nslookup - esempio

```

C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\mf>nslookup -ds www.unicam.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.unicam.it
ttl = 1200 (20 mins)

Server: winserv2.informatica.unicam.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.unicam.it, type = A, class = IN
ANSWERS:
-> www.unicam.it
canonical name = srv1web.unicam.it
ttl = 82501 (22 hours 56 mins 21 secs)
-> srv1web.unicam.it
internet address = 193.204.8.30
ttl = 82501 (22 hours 56 mins 21 secs)

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

C:\Documents and Settings\mf>

```

modalità debug (-ds)

Reti di Elaboratori

dig - esempio

```

studente@server-IRS: ~
;; MSG SIZE rcvd: 299
studente@server-IRS: $ dig
; <<> DIG 9.18.1-1ubuntu1.2-Ubuntu <<>
;; global options: +cnd
;; Got answer:
;; -->HEADER<-- opcode: QUERY, status: NOERROR, id: 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      m.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    2801:5b3:b3e:2:3b
a.root-servers.net. 3146    IN      A       198.41.0.4
m.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

2.229

229

Reti di Elaboratori

Il file hosts (windows)

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

Nome	Ultima modifica	Tipo	Dimensione
hosts	01/12/2015 12:34	File	1 KB
hosts.ics	23/03/2017 12:35	File iCalendar	1 KB
lmhosts.sam	29/09/2017		
networks	22/08/2013		
protocol	22/08/2013		
services	22/08/2013		

```

# 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

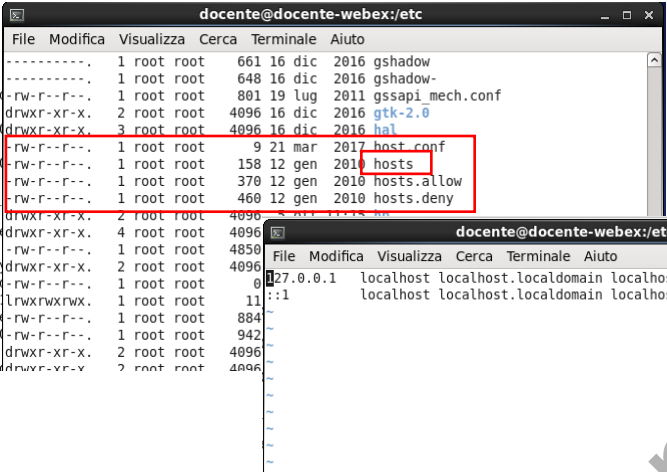
2.230

230

Reti di Elaboratori

Il file hosts (linux)

/etc/hosts




Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.231

231

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

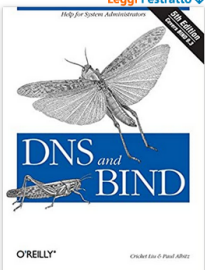
Chapter 2 APPLICATION PROTOCOL

2.232

232

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 41,60 €
---------------------------	---------------------------------

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 that maps names like www into addresses that computers can understand.

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

2.233

233

Reti di Elaboratori

Enumerate DNS

<https://dnstoolkit.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

2.234

234

Reti di Elaboratori	Laboratorio
---------------------	--------------------

internet reti sicurezza

Esercitazioni

Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL	2.235
-------------------	--------------------------------	-------

235

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	2.236
-------------------	--------------------------------	-------

236

Reti di Elaboratori	Simple Network Management Proocol
Gestione delle reti: SNMP RFC 1157 https://datatracker.ietf.org/doc/html/rfc1157 ✓ Simple Network Management Proocol ✓ 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. ✓ Semantica del protocollo SNMP: ➤ GET, GETNEXT ("WALK"), GETBULK, SET, TRAP SNMP Agent SNMP Manager	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL 2.237

237

Reti di Elaboratori	SNMP
SNMP è usato diffusamente per monitorare e configurare dispositivi di rete e anche computer o software ☐ SNMP version 1, 2c e 3 ☐ Uso di ACL ☐ Specificare SNMP per porta UDP port 161 ☐ Blocco di SNMP al firewall Ogni stazione di gestione o agente in una rete gestita da SNMP mantiene un database locale di informazioni rilevanti per la gestione della rete, noto come Management Information Base (MIB)	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL 2.238

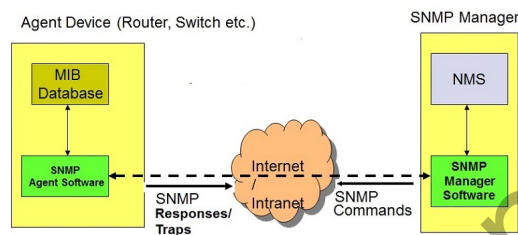
238

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



Fausto Marcantoni

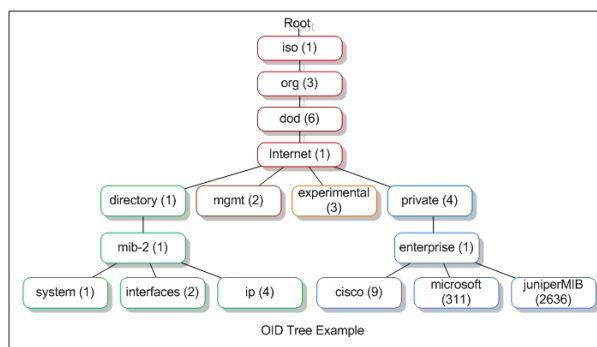
Chapter 2 APPLICATION PROTOCOL

2.239

239

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.



OID Tree Example

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.240

240

identificatore di oggetto (OID)

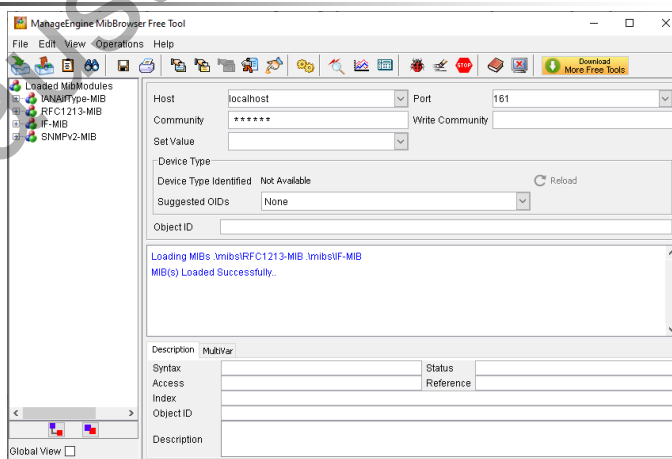
Per ottenere il tempo di accensione del sistema di un dispositivo gestito, è possibile eseguire il polling di questo

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)
```

Tools SNMP



ManageEngine

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

Reti di Elaboratori

Tools SNMP

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

The screenshot shows the iReasoning MIB Browser interface. On the left, the MIB tree is expanded to show the 'iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable' entry. The main window displays a 'Result Table' with the following data:

Name/OID	Value	Type	IP Port
sysDescr.0	RouterOS RB4011IGS+	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...

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.243

243

Reti di Elaboratori

Come installare e configurare SNMP su Windows 10

The screenshot shows the Windows 10 Settings application, specifically the 'Optional features' section. Under 'Features that can be turned on or off', the 'Simple Network Management Protocol (SNMP)' feature is listed with an 'Install' button next to it.

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.tecnobabele.com/come-installare-e-configurare-snmp-su-windows-10/2021-08-27/>

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.244

244

Reti di Elaboratori

snmp <-> wireshark

UDP

Request

OID

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.245

245

Reti di Elaboratori

snmp <-> wireshark

Response

OID

valore

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.246

246

Reti di Elaboratori	creare grafici basati su dati SNMP creare grafici basati su dati SNMP (cioè raccolti da dispositivi di rete via <i>Simple Network Management Protocol</i>) https://www.cacti.net https://www.zabbix.com https://www.librenms.org https://oss.oetiker.ch/rrdtool/
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL

2.247

247

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

2.248

248

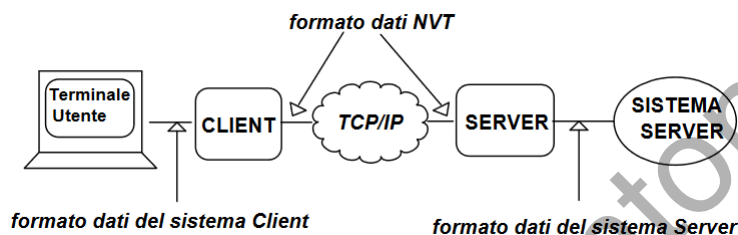
Remote Login - telnet

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.

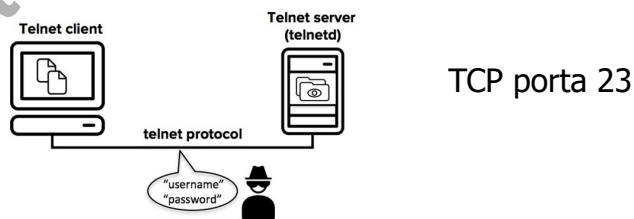
Più precisamente, ad entrambi i capi della comunicazione, il protocollo telnet prevede istanze paritetiche di un **NVT (Network Virtual Terminal)**.



Remote Login - telnet

telnetd

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



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>/
```

Reti di Elaboratori

Wireshark <-> telnet

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.251

251

Reti di Elaboratori

Wireshark <-> telnet

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.252

252

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 2.253

253

Reti di Elaboratori	Remote Login - ssh
Sshd è il daemon che sta in attesa per le connessioni. Normalmente resta in ascolto su una porta 22 TCP fornisce un canale criptato sicuro su di una rete insicura. Effettua: ✓ Autenticazione del server ✓ Scambio di chiavi ✓ Cifratura ✓ Protezione d'integrità	
Fausto Marcantoni	Chapter 2 APPLICATION PROTOCOL 2.254

254

Reti di Elaboratori

Remote Login - tools




PuTTY

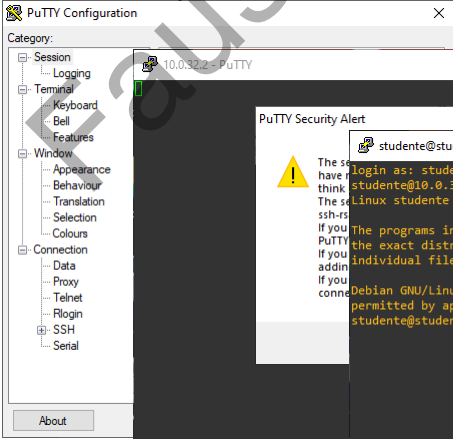
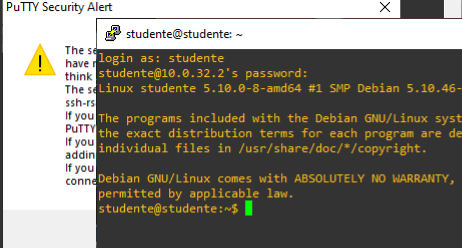
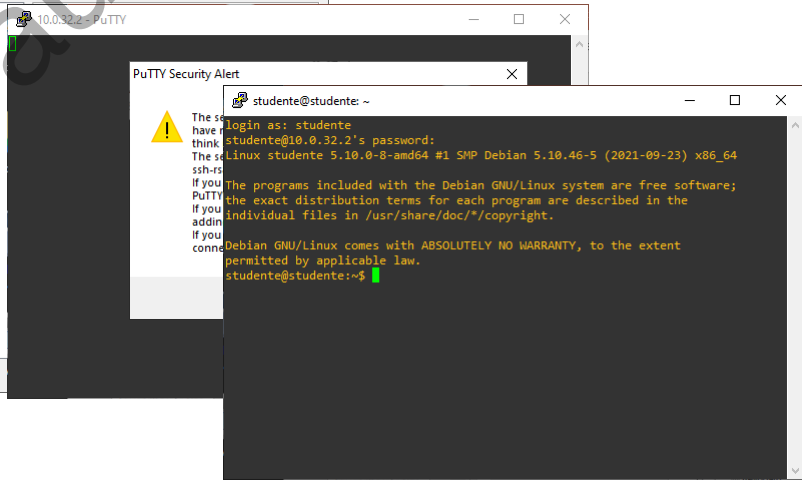
<https://www.putty.org/>
<https://winscp.net>

Fausto Marcantoni
Chapter 2 APPLICATION PROTOCOL
2.255

255

Reti di Elaboratori

Connessione ssh

Fausto Marcantoni
Chapter 2 APPLICATION PROTOCOL
2.256

256

Reti di Elaboratori

Wireshark <-> ssh

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.257

257

Reti di Elaboratori

Modello client - server

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.258

258

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

2.259

259

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

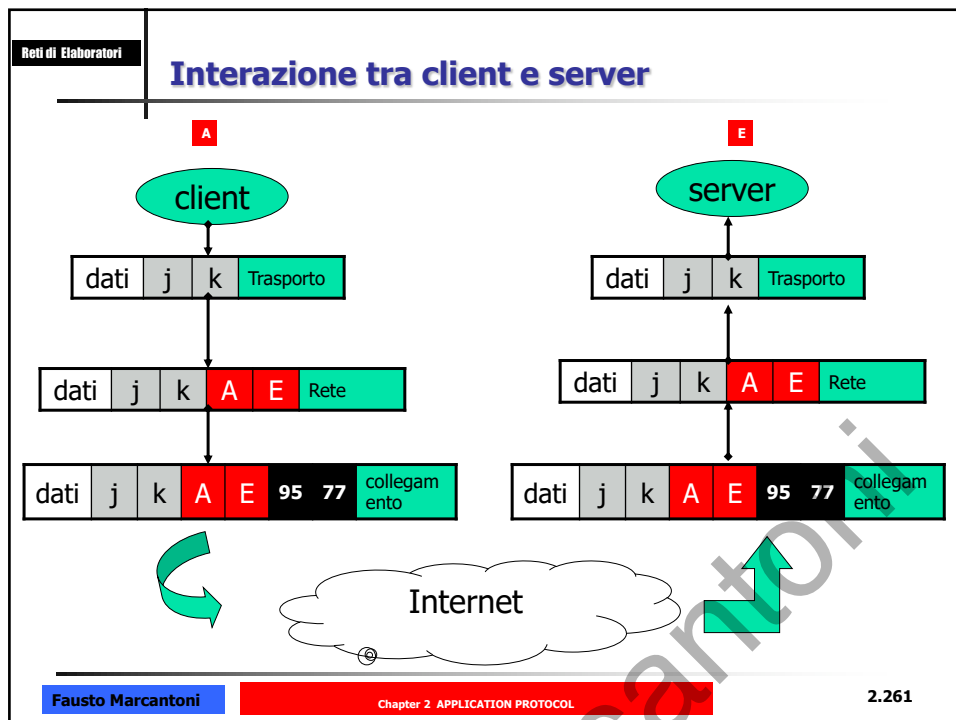
- Programma specializzato
 - Dedicato all'erogazione di un servizio
 - Può servire più di un client alla volta
- Automaticamente avviato all'avvio del sistema
- Attende passivamente di essere contattato dai client
- Accetta richieste da client arbitrari, ma offre un solo servizio
- Calcolatore condiviso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

Fausto Marcantoni

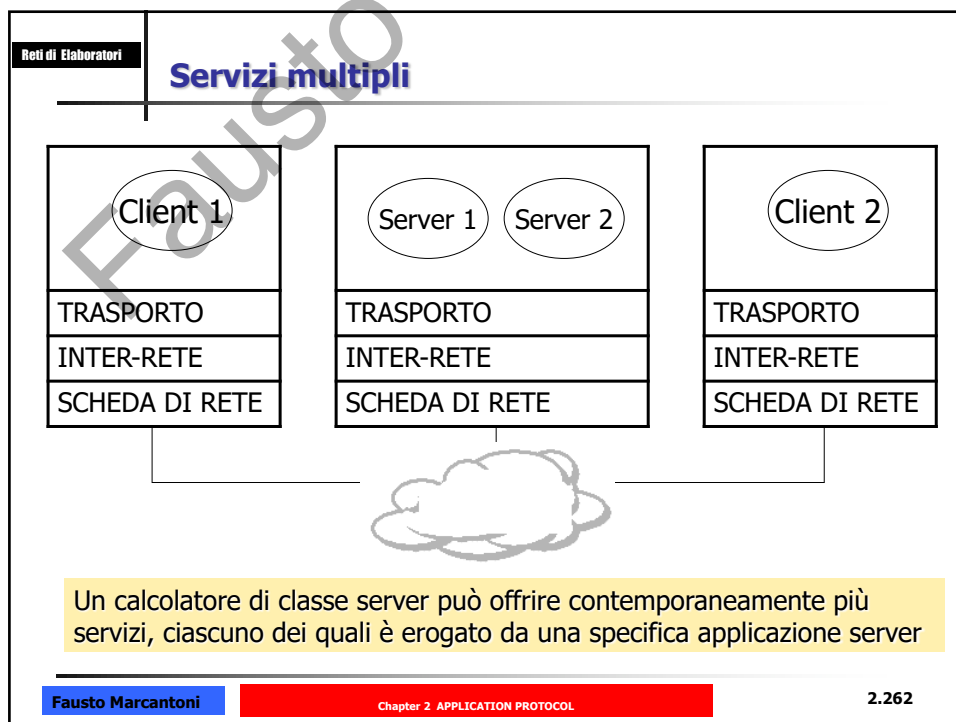
Chapter 2 APPLICATION PROTOCOL

2.260

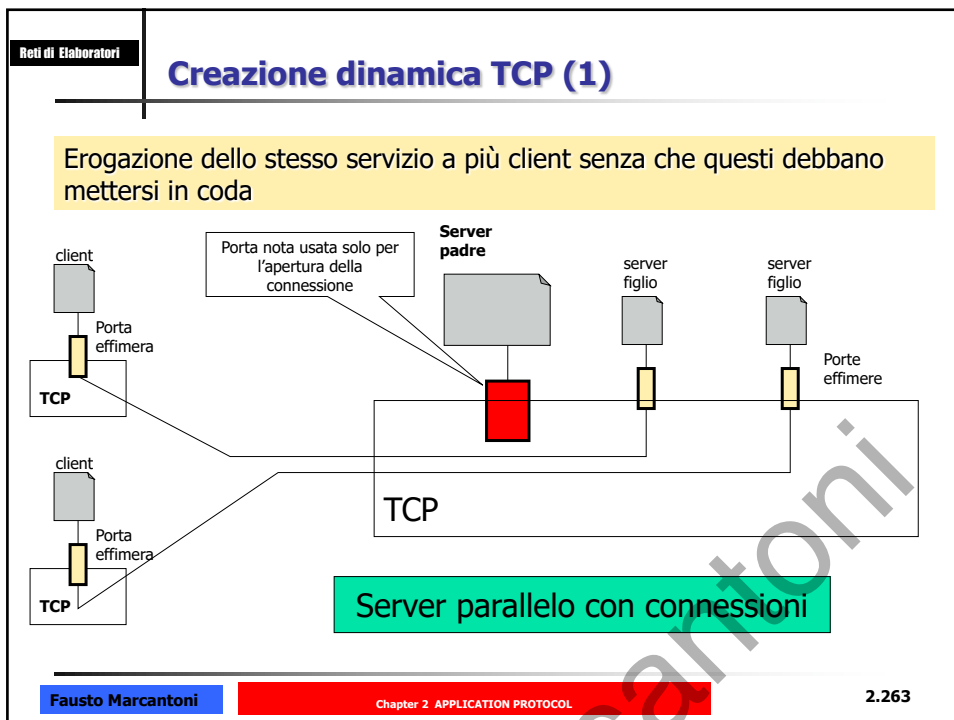
260



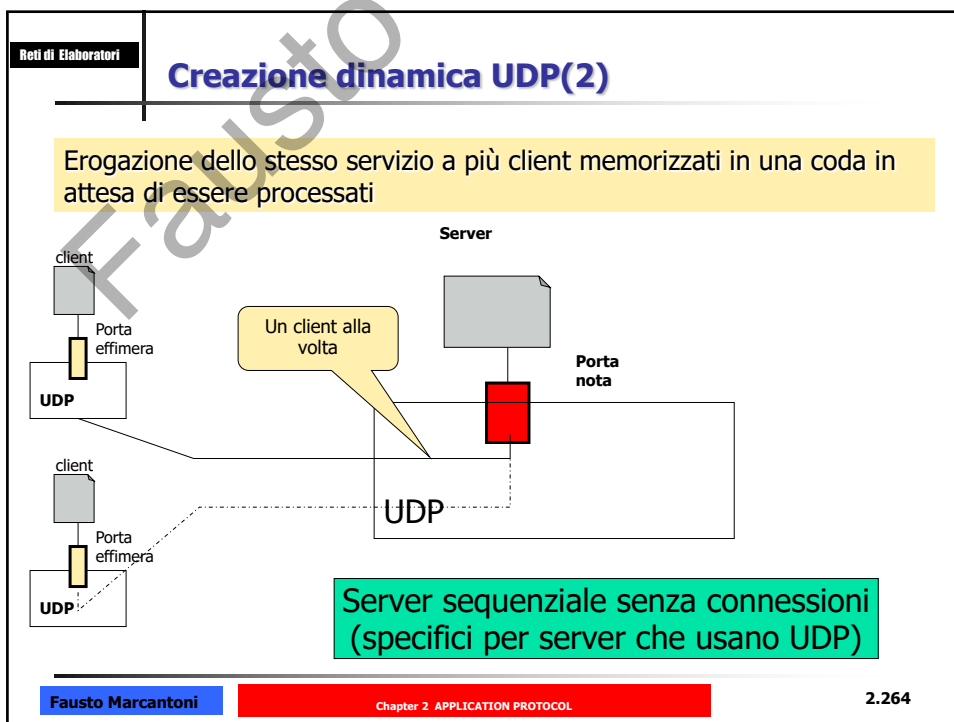
261



262



263



264

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

2.265

265

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))
```

The screenshot shows a Google search interface with the query 'socket programming howto'. Below the search bar, there are tabs for 'Web', 'Immagini', 'Maps', and 'Shopping'. The 'Web' tab is selected, and the results show 'Circa 1.910.000 risultati (0,22 secondi)'.

Fausto Marcantoni

Chapter 2 APPLICATION PROTOCOL

2.266

266

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
2.267

267

Reti di Elaboratori

Socket programming with TCP

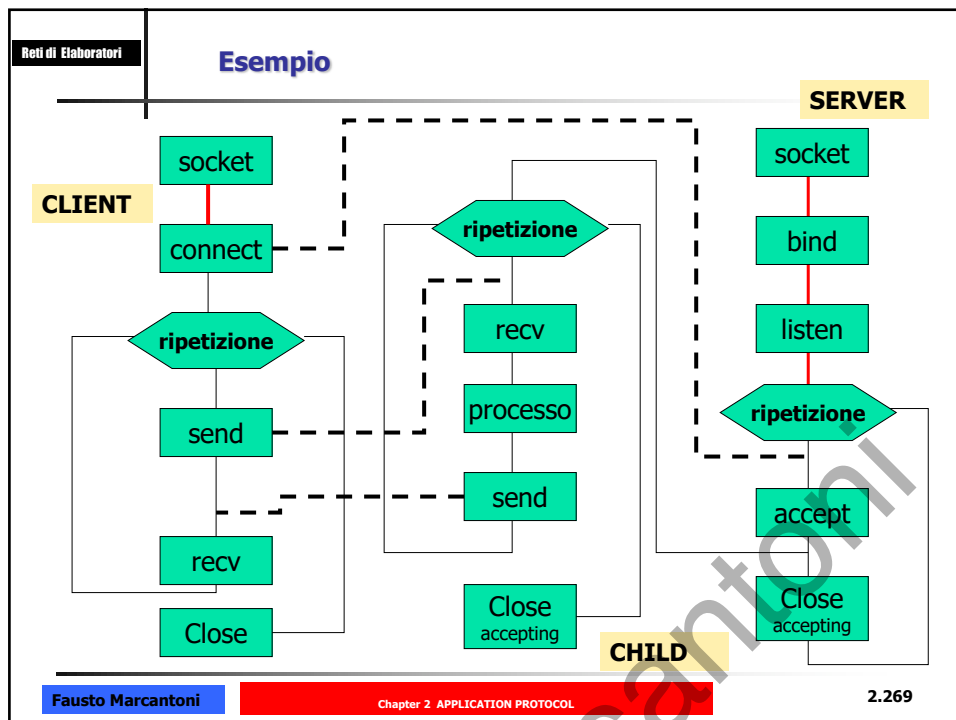
Client deve contattare il server

- 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
2.268

268



269

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

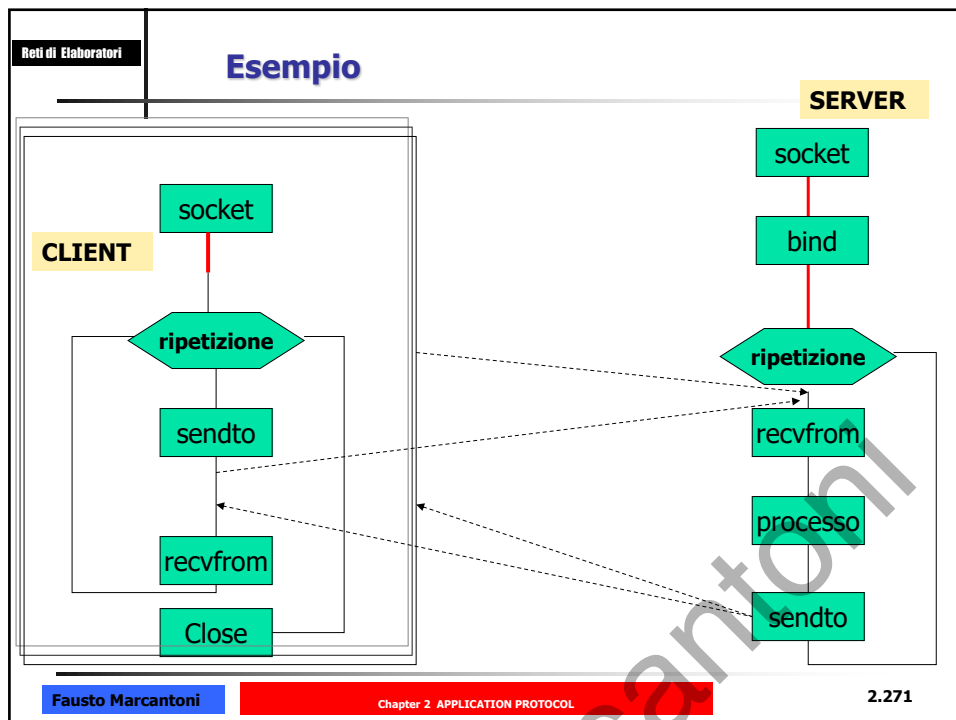
UDP: transmitted data may be received out of order, or lost

application viewpoint

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

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.270

270



271

Reti di Elaboratori

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.

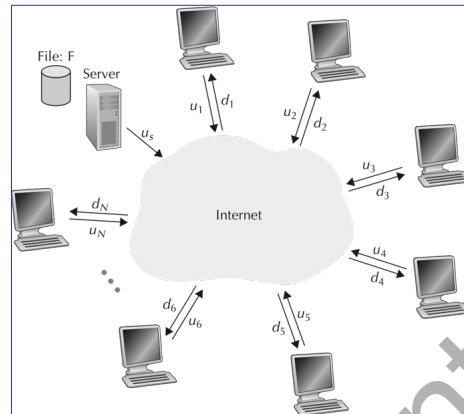
BitTorrent: uno tra i più diffusi protocolli di distribuzione di file tramite P2P.

Scalabilità dell'architettura P2P

Fausto Marcantoni Chapter 2 APPLICATION PROTOCOL 2.272

272

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



BitTorrent

BitTorrent



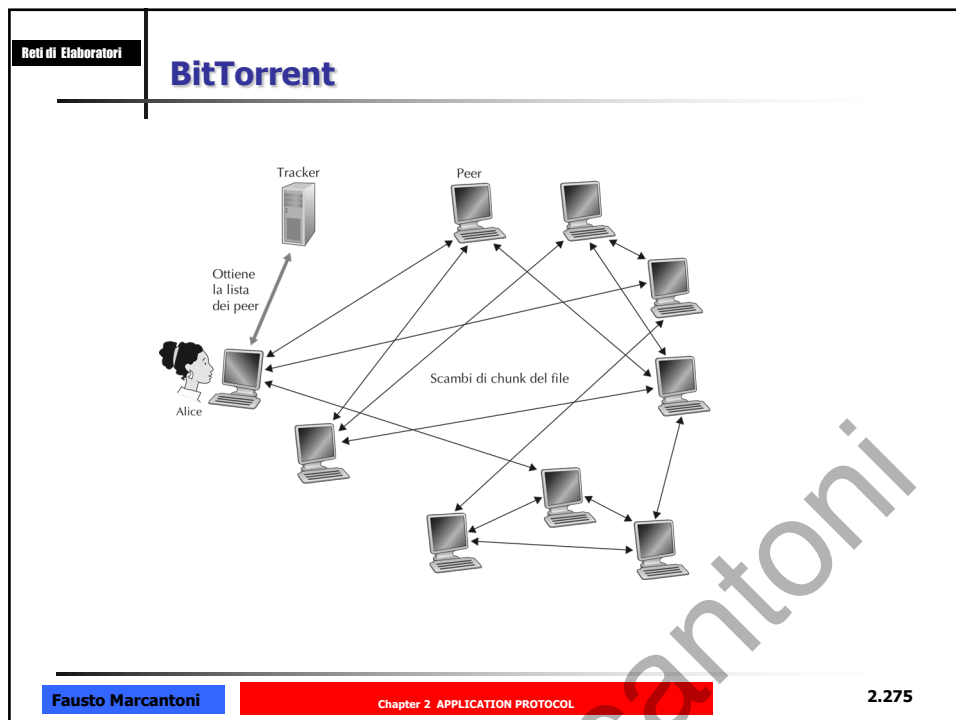
L'insieme di tutti i peer che partecipano alla distribuzione di un particolare file è chiamato **torrent** (**torrente**).

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

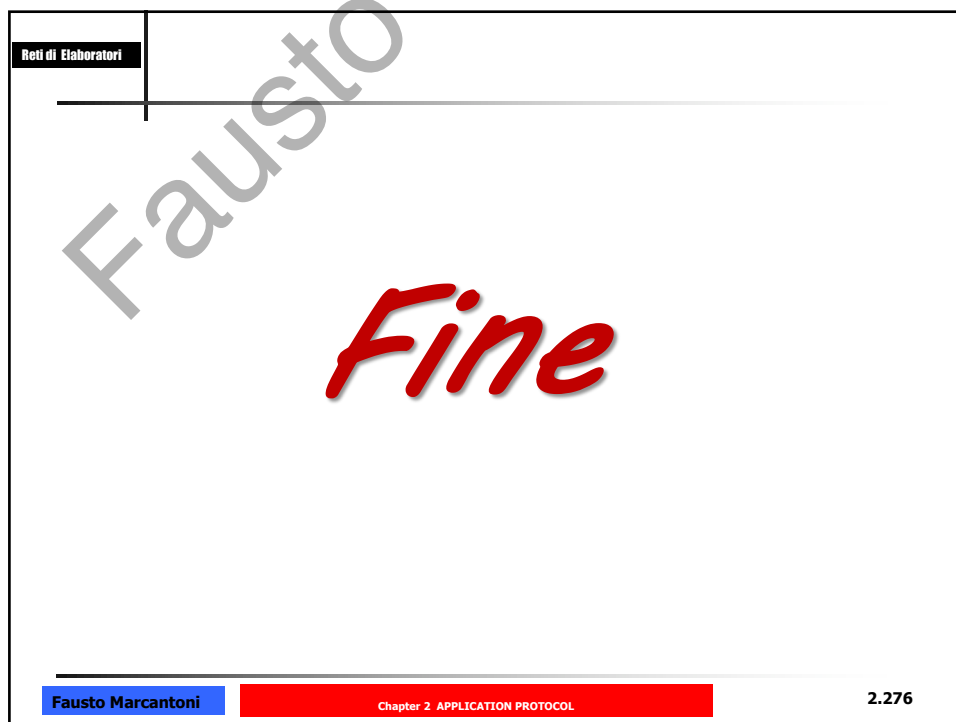
Ciascun torrent ha un nodo di infrastruttura chiamato: **tracker**.

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.



275



276