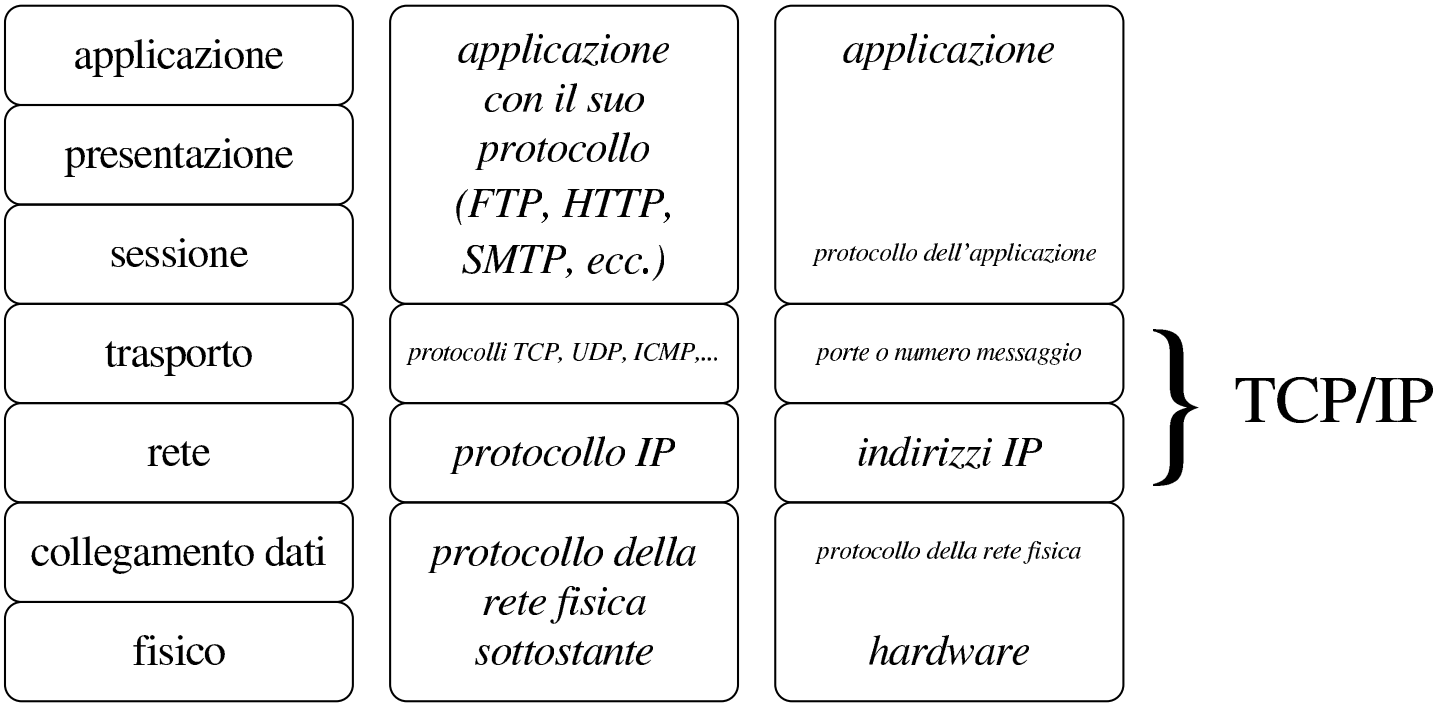
Capitolo 42

Filtri, proxy e ridirezione del traffico IP

42.1 Traffico IPv4 e filtri

Prima di poter studiare i meccanismi di filtro del traffico IP occorre conoscere alcuni concetti elementari che riguardano questi protocolli; diversamente diventa difficile comprendere il senso delle cose che si fanno. In particolare è il caso di ripetere inizialmente l'abbinamento tra il modello ISO-OSI e la realtà del TCP/IP (l'argomento è trattato approfonditamente nella sezione 32.1).

Figura 42.1. Abbinamento tra il modello ISO-OSI e la realtà dei protocolli TCP/IP.



42.1.1 Caratteristiche elementari dei protocolli fondamentali

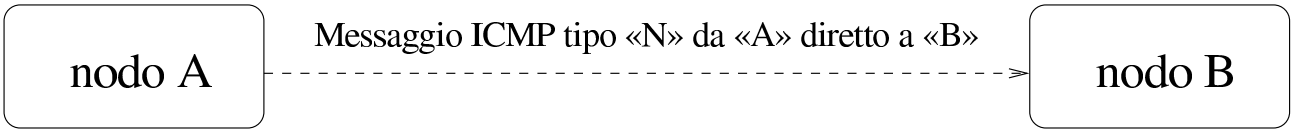
«

Sulla base del protocollo IP si utilizzano in modo particolare i protocolli ICMP, UDP e TCP. Le informazioni contenute nei pacchetti del protocollo ICMP sono diverse da quelle che riguardano UDP e TCP, principalmente per il fatto che nel primo non si utilizzano le porte. Infatti, il protocollo ICMP viene usato per l'invio di messaggi che riguardano il funzionamento della rete, distinguendoli in base a un numero. Pertanto, un pacchetto ICMP, oltre agli indirizzi IP di origine e di destinazione, contiene un numero che qualifica il tipo di messaggio (precisamente un tipo e un sottotipo).

Tabella 42.2. Alcuni tipi di messaggi ICMP.

Tipo	Nome	Chi lo utilizza
0	echo-reply	ping
3	destination-unreachable	traffico TCP e UDP
5	redirect	instradamento dei pacchetti
8	echo-request	ping
11	time-exceeded	traceroute

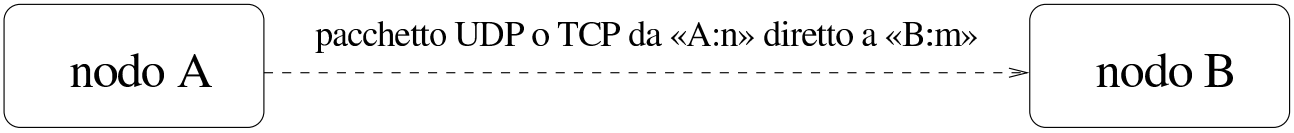
Figura 42.3. Viaggio di un messaggio ICMP.



I pacchetti dei protocolli UDP e TCP hanno la caratteristica comune di possedere, oltre all'indicazione dell'indirizzo di origine e di quello di destinazione, anche un numero di porta, sia per l'origine, sia per la destinazione. In altri termini, un pacchetto UDP o TCP è originato da un certo indirizzo IP e da una certa porta, essendo diretto

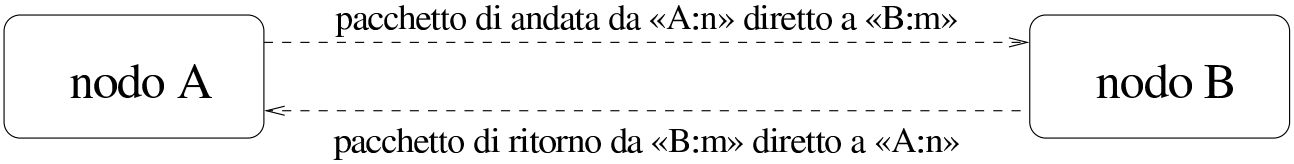
a un certo indirizzo IP e a una certa porta.

Figura 42.4. Viaggio di un pacchetto UDP o TCP.



Evidentemente, l'informazione sulla porta serve a ogni nodo per distinguere il contesto per il quale viene inviato o ricevuto un pacchetto. In particolare, se il protocollo prevede una risposta di qualche tipo, questa avviene generalmente utilizzando le stesse porte in senso inverso.

Figura 42.5. Andata e ritorno per le connessioni che prevedono l'uso delle porte.



Per quanto riguarda il caso particolare del protocollo TCP, la connessione può avvenire solo se si forma un flusso di pacchetti sia di andata, sia di ritorno, anche se uno dei due flussi serve solo per confermare gli invii dall'altra parte. In questo senso, l'interruzione della comunicazione in una direzione impedisce anche l'altra.

42.1.2 Porte

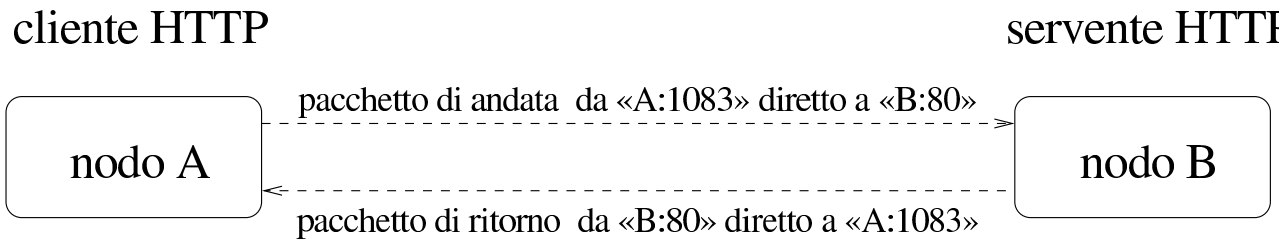
Nei sistemi Unix si distinguono due gruppi importanti di porte: quelle privilegiate, rappresentate solitamente dall'intervallo da 0 a 1023, e le altre, non privilegiate, che vanno da 1024 a 65535.

La differenza sta nel fatto che i processi possono aprire localmente una porta del gruppo da 1 a 1023 solo se funzionano con i privilegi dell'utente '**`root`**'. In questo senso, si tratta generalmente di

demoni che offrono un servizio attraverso la rete, restando in ascolto di una porta privilegiata, attraverso la quale poi rispondono quando interpellati.

Molti numeri di porta hanno un utilizzo convenzionale, specialmente per quanto riguarda il gruppo di quelle privilegiate. In questo modo si può prevedere quale sia la porta che occorre interpellare per raggiungere un certo servizio in un nodo determinato. Per converso, generalmente, il processo che inizia la comunicazione rivolgendosi a un servizio noto, apre per conto proprio una porta non privilegiata. Si può osservare a questo proposito l'esempio che appare nella figura 42.6, in cui si vede che nel nodo «A» un programma di navigazione richiede e ottiene una connessione con il nodo «B» per un servizio HTTP, offerto lì attraverso la porta 80. La porta scelta dal navigatore per questa operazione viene presa a sua discrezione tra quelle non privilegiate che non sono già allocate o riservate per qualche scopo particolare.

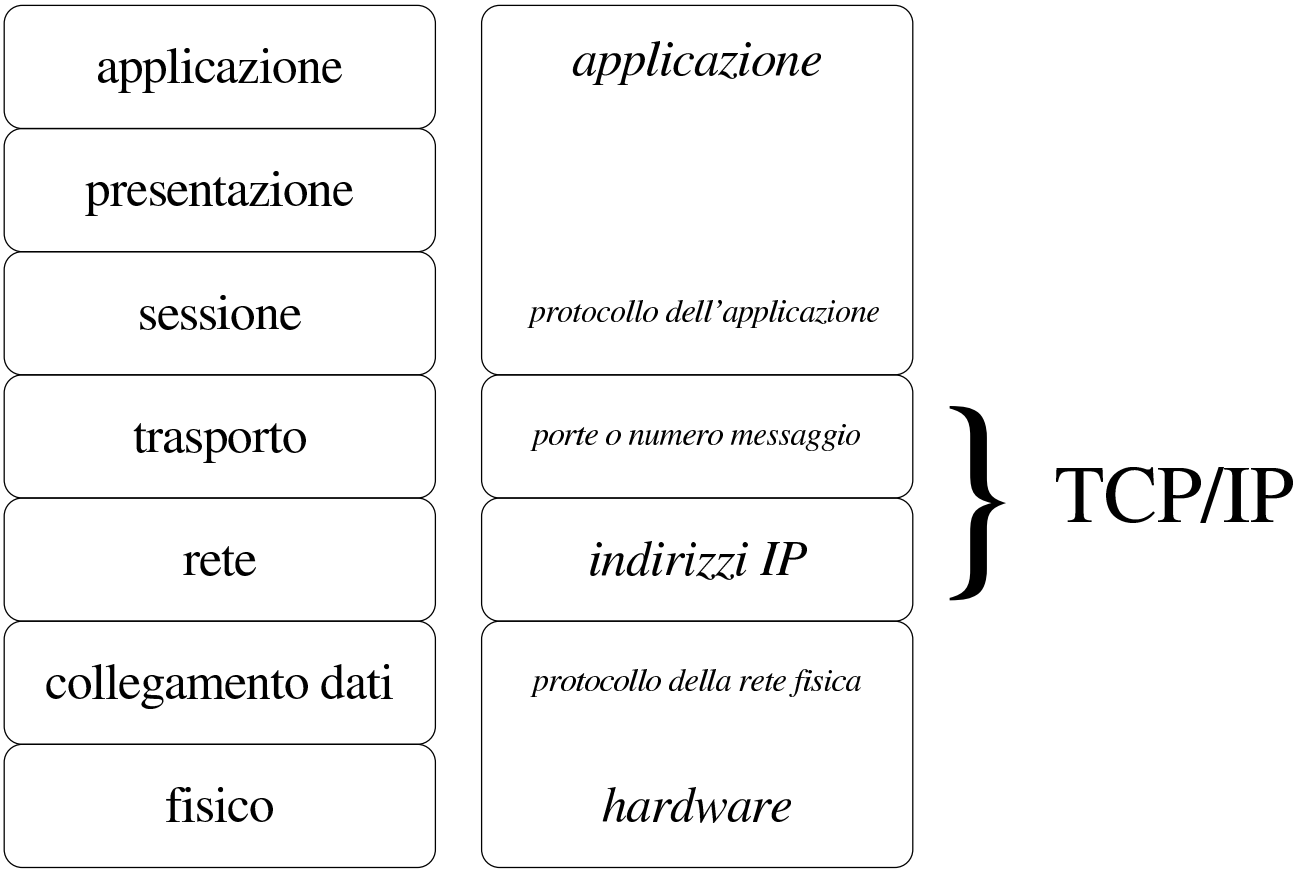
Figura 42.6. Esempio di ciò che accade quando dal nodo «A» un processo instaura una connessione HTTP con il nodo «B»; in particolare, in questo caso il processo in questione utilizza localmente la porta 1083.



42.1.3 Frammentazione IP

I pacchetti generati a livello di trasporto (TCP, UDP e ICMP) possono essere frammentati dal protocollo IP, in base alle necessità. In tal caso, i frammenti successivi al primo hanno meno informazioni a disposizione; per la precisione perdono le indicazioni salienti che permettono di identificare le loro caratteristiche in base ai protocolli del livello di trasporto. Generalmente, quando si inserisce un filtro al traffico IP si fa in modo di ricomporre i pacchetti, ammesso che sia garantito il passaggio obbligato attraverso il filtro stesso.

Figura 42.7. Informazioni essenziali nei pacchetti e livello in cui vengono inserite.



La figura 42.1 dovrebbe aiutare a capire il concetto: è il protocollo IP che si occupa di frammentare i pacchetti (al suo livello) quando il protocollo sottostante non è in grado di gestire le dimensioni che

sarebbero richieste. Pertanto, nei pacchetti frammentati è garantita soltanto la presenza dell'indicazione degli indirizzi IP del mittente e del destinatario, assieme alle informazioni necessarie a ricomporre i pacchetti. In questo modo, le informazioni relative alle porte TCP o UDP si trovano normalmente nel primo di tali frammenti, mentre gli altri ne sono sprovvisti.

Il protocollo TCP è in grado di frammentare e ricomporre i pacchetti provenienti dal livello superiore, ma questo non esclude la possibilità che debba intervenire anche una frammentazione ulteriore, a livello IP, a causa delle limitazioni della rete, di cui il protocollo TCP non può essere consapevole.

42.1.4 Pacchetti SYN

«

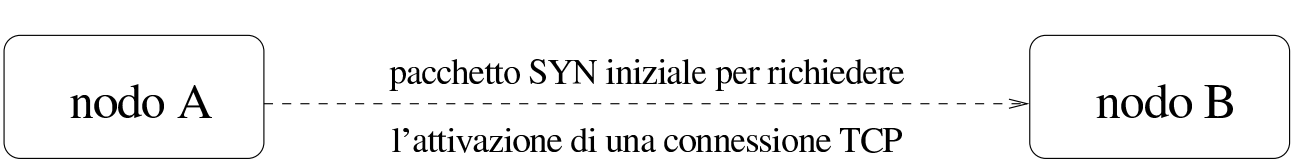
L'instaurarsi di una connessione TCP avviene attraverso fasi differenti, in cui vengono usati degli indicatori all'interno dei pacchetti per attribuire loro un significato speciale. In particolare, quando un pacchetto contiene il bit SYN attivo, si tratta di un tentativo di iniziare una nuova connessione.

L'individuazione del pacchetto SYN è importante per capire chi sia colui che inizia a fare qualcosa. Per esempio, se una connessione TCP avviene tra il nodo «A» con la porta 1083 e il nodo «B» con la porta 80, non vuol dire necessariamente che si tratti di una connessione iniziata da «A», così come non è detto che si tratti dell'utilizzo di un servizio HTTP.

Nella realizzazione di un sistema di filtri di pacchetti IP, potrebbe essere utile individuare i pacchetti SYN in modo da poter intervenire

sulle comunicazioni in base al verso che hanno.

Figura 42.8. Il pacchetto SYN rivela da quale parte ha inizio la connessione.



42.1.5 Conseguenze nell'introduzione di un filtro

Un filtro nel traffico dei pacchetti può tenere conto solo delle poche informazioni che questi portano con sé, considerando anche la possibilità che queste siano state contraffatte. In generale, diventa difficile poter dire: «voglio escludere il traffico del servizio "X"». In realtà si escludono i pacchetti che dovrebbero servire a quel tipo di servizio o che servono alla sua instaurazione.

La realizzazione di un filtro efficace per i fini che ci si aspetta di ottenere può essere realizzato solo conoscendo bene le caratteristiche dei protocolli coinvolti. In realtà, una conoscenza così approfondita è difficile da acquisire, anche quando il proprio lavoro è fare l'amministratore di rete. Infatti, una svista può causare il malfunzionamento di qualcosa, oppure, peggio, può lasciare aperto un passaggio a un aggressore o a un altro tipo di pericolo.

In generale, meno compiti si attribuiscono a un filtro, meglio si riesce a controllare la situazione. L'uso di programmi per l'analisi del traffico nella rete permette di comprendere meglio, in pratica, cosa succeda effettivamente (si veda eventualmente IPTraf descritto nella sezione 43.8.4).

42.1.5.1 Messaggi ICMP

«

In generale, bisogna fare molta attenzione se si introduce un qualche tipo di filtro ai pacchetti contenenti messaggi ICMP, dal momento che da questi dipende il funzionamento della rete. Sicuramente non si può escludere il passaggio di messaggi di tipo 3: *destination-unreachable*.

42.1.5.2 Protocolli basati su TCP

«

In linea di principio, i protocolli basati su TCP sulla base del presupposto che un servente collocato da qualche parte offra il suo servizio attraverso una porta privilegiata, mentre i clienti lo interpellano usando localmente una porta non privilegiata.

Volendo fare riferimento al caso del protocollo HTTP, si possono individuare le connessioni in uscita, verso serventi esterni, come quelle che avvengono tra il gruppo di porte locali non privilegiate e la porta 80 remota.

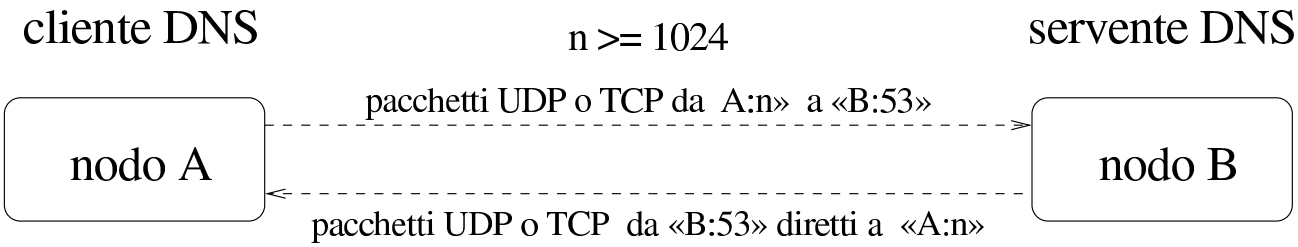
Tuttavia, non tutti i protocolli che si basano su TCP funzionano in modo così semplice. Alcuni aprono delle connessioni secondarie, utilizzando porte non privilegiate e non prestabilite, in base alle operazioni che si stanno svolgendo. In quei casi, diventa praticamente impossibile trovare un metodo per filtrare tali connessioni, allo scopo di lasciare transitare solo queste, mentre è comunque facile impedirle, perché bloccando la connessione iniziale si ottiene il risultato.

42.1.5.3 Protocolli basati su UDP

I protocolli basati su UDP possono essere ancora più articolati rispetto al TCP. Di solito vengono presi in considerazione per bloccarli semplicemente, eventualmente con l'unica eccezione di ciò che serve alla gestione del DNS.

Il servizio DNS si basa sulla porta 53, ma può usare il protocollo UDP o TCP, a seconda della necessità. Per concedere espressamente il transito ai pacchetti relativi al protocollo DNS, occorre agire su UDP e TCP.

Figura 42.9. Esempio del transito di pacchetti relativo all'utilizzo di un servizio DNS.



42.2 Cache proxy

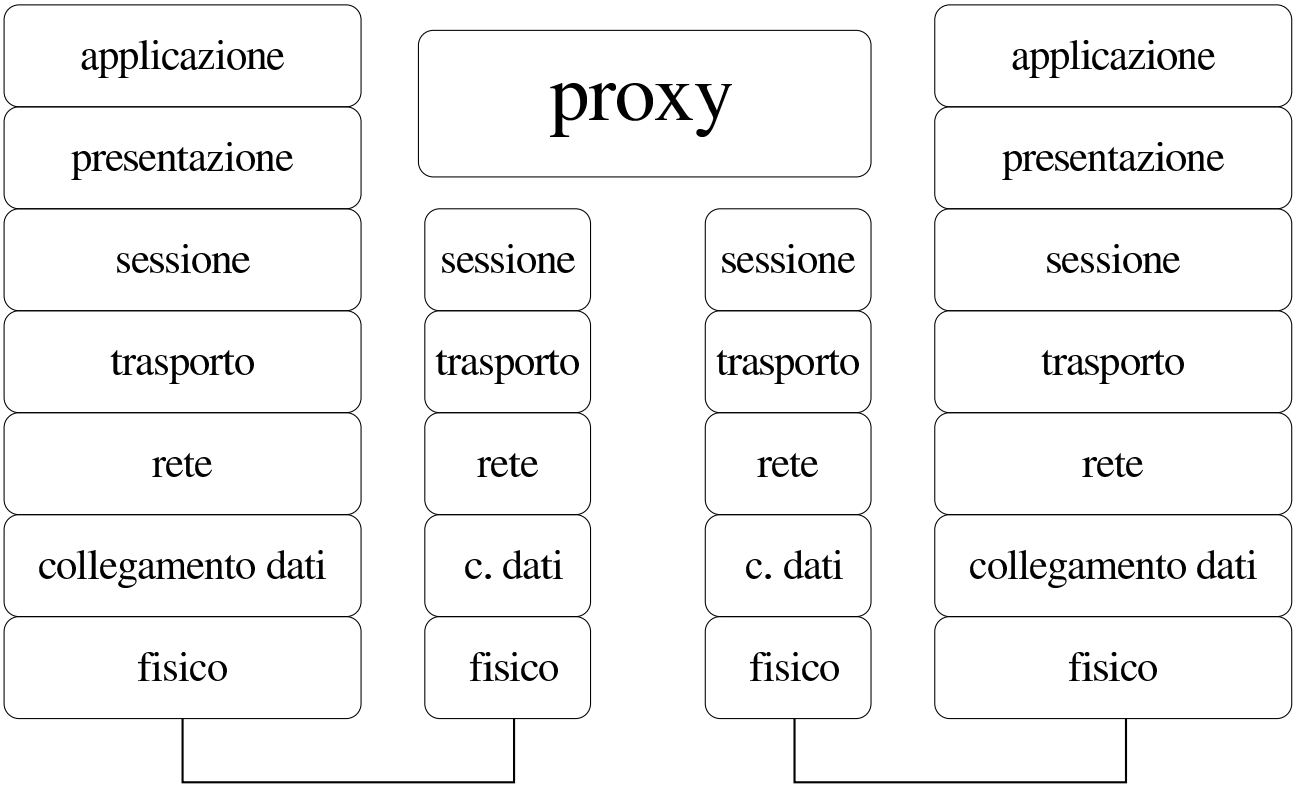
Nella terminologia utilizzata per le reti, un ***cache proxy*** è un servizio di memorizzazione locale delle risorse della rete richieste più frequentemente. Con il termine «risorsa» si deve intendere un oggetto a cui si accede attraverso un URI.

L'utilizzo di un proxy offre due vantaggi principali: l'accesso rapido a risorse già accumulate nella memoria cache e la riduzione del traffico nella rete che precede il proxy stesso.

42.2.1 Schema essenziale

Il proxy si interpone nella rete agendo, idealmente, al di sopra del quinto livello del modello ISO-OSI, come si vede nella figura 42.10. Infatti, il cliente di un proxy intrattiene normalmente una connessione HTTP o FTP; così il proxy deve intrattenere lo stesso tipo di connessione, per conto proprio, con il servente a cui il cliente avrebbe voluto rivolgersi realmente, a meno di ottenere tali risorse dalla propria memoria cache.

Figura 42.10. Il proxy trasferisce PDU al di sopra del quinto livello; in pratica gestisce direttamente i protocolli a livello di sessione.



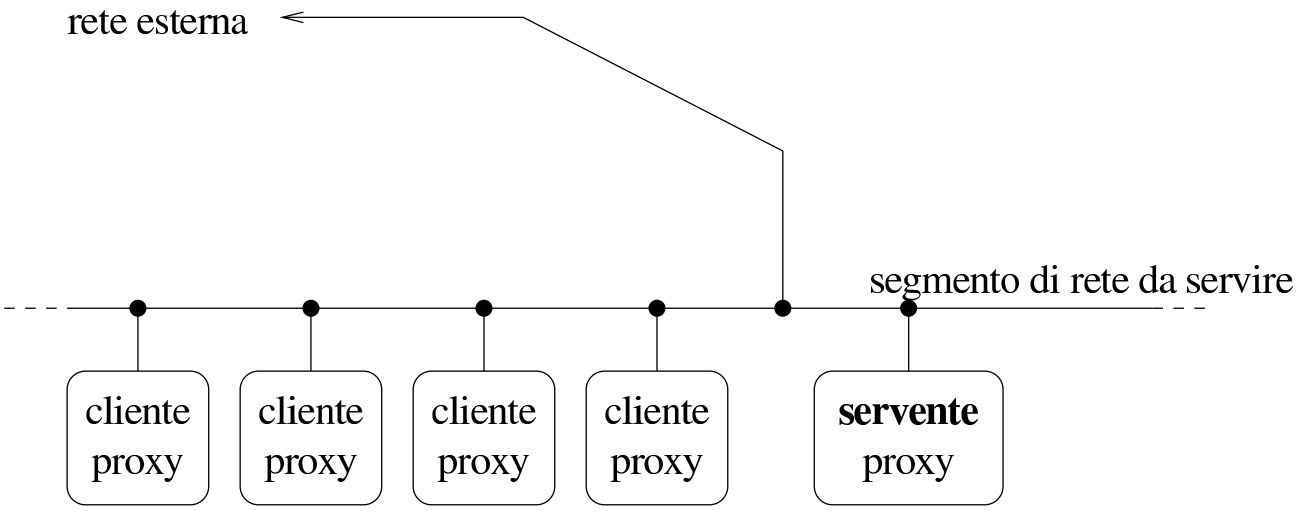
Il servizio di cache proxy può essere collocato in posizioni differenti nella rete, a seconda delle esigenze o delle particolarità delle situazioni. Generalmente, lo scopo è quello di servire un segmento di rete, indifferentemente dal fatto che questo segmento utilizzi

indirizzi privati o sia accessibile dall'esterno.

42.2.1.1 Servire un segmento di rete

Quando un proxy viene utilizzato per servire un segmento di rete rispetto alla rete esterna, senza fare altre considerazioni, è sufficiente che l'elaboratore su cui viene collocato il servizio sia accessibile da questo segmento di rete e che a sua volta sia in grado di accedere all'esterno.

Figura 42.11. In questa situazione, il servente proxy è collegato come tutti gli altri elaboratori al segmento di rete da servire.

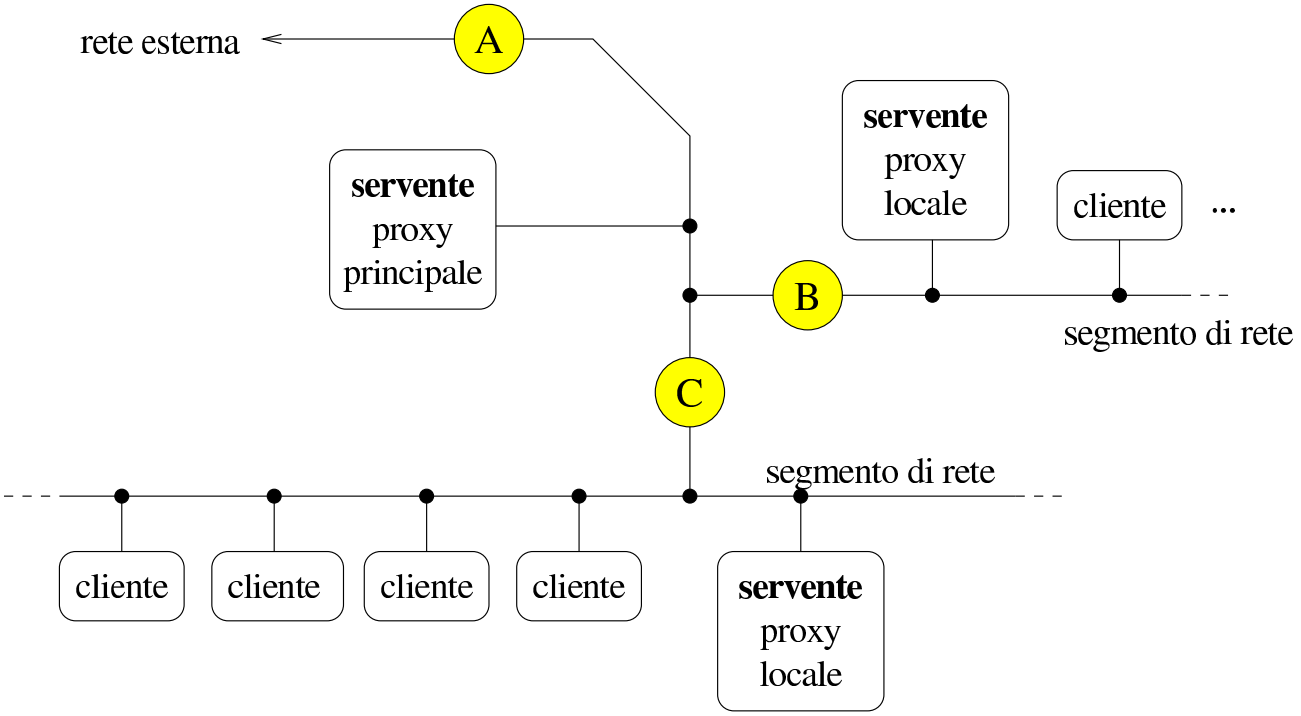


A questa situazione appartiene anche il caso limite in cui il proxy serve solo se stesso, quindi la stessa macchina è servente e anche cliente.

42.2.1.2 Proxy a più livelli

Un proxy potrebbe servirsi di altri proxy quando si tratta di accedere a reti determinate, alleggerendo in questo modo il carico della rete anche in altri punti, non solo nel tratto immediatamente precedente.

Figura 42.12. Ogni collegamento ha un proprio proxy locale che però si avvale di un proxy principale prima di raggiungere la rete esterna.



La figura 42.12 mostra il caso di un collegamento a una rete esterna, (A), condiviso da due segmenti di rete, i quali si uniscono a questa attraverso i collegamenti B e C. A valle del collegamento A si trova un proxy il cui scopo è quello di ridurre il più possibile il traffico attraverso quel tratto; a valle dei collegamenti B e C si trovano altri proxy locali il cui scopo è quello di ridurre il traffico attraverso i collegamenti rispettivi. In questa situazione, i proxy locali utilizzano a loro volta il servente principale, mentre tutto quello che viene accumulato nei proxy locali, viene conservato anche in quello principale.

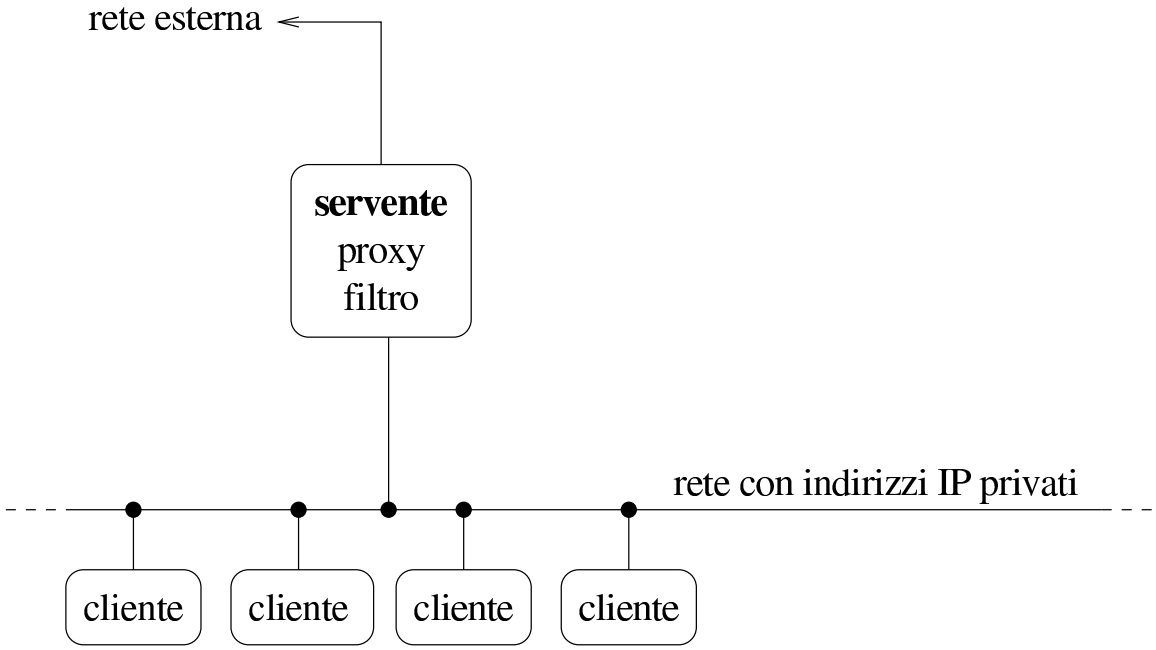
42.2.1.3 Proxy come filtro verso l'esterno

«

Il servente proxy, se si trova in un elaboratore che è connesso simultaneamente, attraverso interfacce di rete differenti, a una rete interna con indirizzi privati (cioè esclusi da Internet) e alla rete esterna, può

essere utilizzato per permettere ai clienti della rete privata di avere accesso all'esterno attraverso il proxy stesso. Ma questo accesso si limita ai protocolli gestiti dal proxy; spesso si tratta solo di HTTP e FTP.

Figura 42.13. Come caso estremo, il proxy può ricoprire anche un ruolo di filtro e inoltro di pacchetti tra una rete privata e la rete esterna.

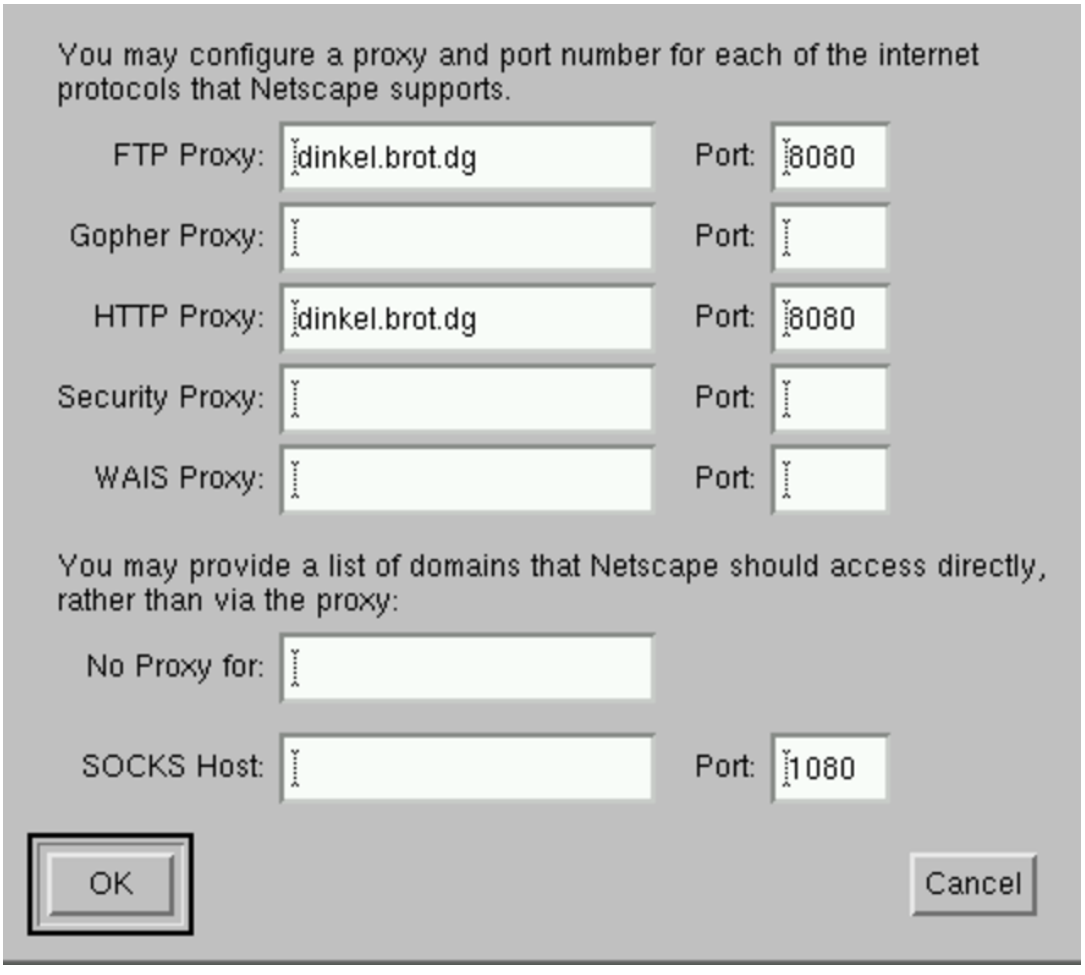


Va anche osservato che, in una condizione di questo tipo, l'elaboratore che svolge il servizio proxy potrebbe essere configurato per renderlo «trasparente». In pratica, ciò richiede che i pacchetti del protocollo TCP, destinati alle porte dei servizi di cui si occupa il proxy, vengano dirottati alla porta del proxy stesso. Ma ciò richiede anche che il proxy sia configurato per questa situazione, in quanto deve agire come se fosse un router. Per quanto riguarda gli elaboratori clienti della rete locale, questi verrebbero configurati come se il proxy fosse un router in grado di metterli in comunicazione con la rete esterna.

42.2.2 Dal lato del cliente

I clienti per la navigazione, vanno configurati per poter sfruttare il servizio del cache proxy. Per esempio, la figura 42.14 mostra la finestra di configurazione di un navigatore comune.

Figura 42.14. Esempio di configurazione di un navigatore comune per l'utilizzo del cache proxy. Si osservi il fatto che per usare la porta 8080 occorre che il servente sia in ascolto sulla stessa.



I programmi di navigazione offrono anche la possibilità di richiedere al proxy di prelevare una nuova copia della pagina, pure se non sono scaduti i tempi previsti. Nel caso di programmi grafici si tratta normalmente di selezionare pulsanti del tipo RELOAD, RICARICA o simili.

Il proxy risponde alle richieste dei programmi clienti attraverso una porta particolare, la quale dipende dalla configurazione del servizio. Apparentemente, ogni tipo di proxy ha una sua impostazione predefinita differente, mentre la tendenza generale è quella di utilizzare la porta 8080. È necessario fare attenzione a questo particolare quando si configura il proxy, per non creare confusione inutile agli utenti del servizio.

42.2.3 Caratteristiche comuni ai cache proxy da considerare

Prima di affrontare lo studio di un tipo particolare di cache proxy, vale la pena di riordinare le idee sulle esigenze tipiche di un servizio del genere, dal momento che queste si riflettono nella configurazione relativa. In breve i problemi riguardano essenzialmente i punti seguenti:

- **amministrazione della memoria cache**
  - collocazione dei file utilizzati dalla memoria cache
  - utente e gruppo proprietari di questi file
  - dimensione massima della memoria cache
  - dimensione massima di una singola risorsa accumulabile
  - scadenza massima per la validità delle informazioni accumulate nella memoria cache
  - Indirizzi esclusi dall'accumulo nella memoria (solitamente quelli che contengono le stringhe '`?`' e '`cgi-bin`', perché riguardano probabilmente delle interazioni con programmi CGI)

- **utenze**
  - individuazione degli indirizzi che possono accedere per utilizzare il servizio
  - utente fittizio mostrato all'esterno (di solito per l'accesso a un servizio FTP anonimo)
- **connessione**
  - porta o porte attraverso cui resta in ascolto per le richieste di connessione (di solito si usa la porta 8080)
  - indirizzi e porte di altri servizi del genere da interpellare se disponibili (per non sovraccaricare la rete)

42.2.4 Tinyproxy

«

Tinyproxy[1] è un programma specifico per la gestione di un cache proxy, relativamente più leggero di altri dal punto di vista elaborativo, ma in grado di fornire le funzionalità principali di questo tipo di servizio. Da un punto di vista «pratico», un aspetto importante di Tinyproxy sta nel fatto che la sua memoria cache è gestita esclusivamente in memoria centrale.

Quando si installa Tinyproxy da un pacchetto già pronto per la propria distribuzione GNU, dovrebbe essere predisposto automaticamente lo script della procedura di inizializzazione del sistema che consente di avviare e fermare il servizio in modo semplice, con un comando simile a quello seguente:

```
/etc/init.d/tinyproxy start|stop
```

Tinyproxy si compone del demone '**tinyproxy**', il quale viene avviato normalmente sullo sfondo con i privilegi di un utente di sistema specifico (potrebbe trattarsi dell'utente e del gruppo '**proxy**'). Naturalmente, la scelta dell'utenza in questione non è casuale e di conseguenza devono essere organizzati i permessi di accesso ai file che Tinyproxy deve utilizzare durante il funzionamento; pertanto, generalmente conviene affidarsi a quanto già predisposto da chi ha realizzato il pacchetto applicativo per la propria distribuzione GNU.

La configurazione è naturalmente l'aspetto più importante dell'utilizzo di Tinyproxy. Si tratta di un file principale che fa riferimento a qualche altro file esterno. Il file di configurazione potrebbe essere precisamente '`/etc/tinyproxy/tinyproxy.conf`', ma può essere cambiato utilizzando l'opzione '**-c**', come descritto nella pagina di manuale *tinyproxy(8)*.

Il file di configurazione è un file di testo, dove le righe che iniziano con il simbolo '**#**' sono ignorate, assieme a quelle bianche o vuote. Le direttive occupano una riga soltanto. Segue un esempio commentato delle direttive, escludendo quelle che hanno una definizione predefinita valida in generale. Questo esempio di configurazione si presta anche per l'utilizzo in modalità «proxy trasparente».

```
# User/Group: This allows you to set the user and group that will be
# used for tinyproxy after the initial binding to the port has been done
# as the root user. Either the user or group name or the UID or GID
# number may be used.
User proxy
Group proxy

# Port: Specify the port which tinyproxy will listen on.  Please note
# that should you choose to run on a port lower than 1024 you will need
# to start tinyproxy using root.
Port 8888
```

```
# Timeout: The maximum number of seconds of inactivity a connection is
# allowed to have before it is closed by tinyproxy.
Timeout 600

# ErrorFile: Defines the HTML file to send when a given HTTP error
# occurs.  You will probably need to customize the location to your
# particular install.  The usual locations to check are:
#   /usr/local/share/tinyproxy
#   /usr/share/tinyproxy
#   /etc/tinyproxy
#
# ErrorFile 404 "/usr/share/tinyproxy/404.html"
# ErrorFile 400 "/usr/share/tinyproxy/400.html"
# ErrorFile 503 "/usr/share/tinyproxy/503.html"
# ErrorFile 403 "/usr/share/tinyproxy/403.html"
# ErrorFile 408 "/usr/share/tinyproxy/408.html"
#
# DefaultErrorFile: The HTML file that gets sent if there is no
# HTML file defined with an ErrorFile keyword for the HTTP error
# that has occured.
DefaultErrorFile "/usr/share/tinyproxy/default.html"

# Logfile: Allows you to specify the location where information should
# be logged to.  If you would prefer to log to syslog, then disable this
# and enable the Syslog directive.  These directives are mutually
# exclusive.
Logfile "/var/log/tinyproxy/tinyproxy.log"

# LogLevel: Set the logging level. Allowed settings are:
#       Critical        (least verbose)
#       Error
#       Warning
#       Notice
#       Connect         (to log connections without Info's noise)
#       Info            (most verbose)
#
# The LogLevel logs from the set level and above. For example, if the
# LogLevel was set to Warning, then all log messages from Warning to
# Critical would be output, but Notice and below would be suppressed.
LogLevel Connect
```

```
# PidFile: Write the PID of the main tinyproxy thread to this file so it
# can be used for signalling purposes.
PidFile "/var/run/tinyproxy/tinyproxy.pid"

# MaxClients: This is the absolute highest number of threads which will
# be created. In other words, only MaxClients number of clients can be
# connected at the same time.
#
MaxClients 1024

# MinSpareServers/MaxSpareServers: These settings set the upper and
# lower limit for the number of spare servers which should be available.
#
# If the number of spare servers falls below MinSpareServers then new
# server processes will be spawned.  If the number of servers exceeds
# MaxSpareServers then the extras will be killed off.
MinSpareServers 30
MaxSpareServers 60

# StartServers: The number of servers to start initially.
#
StartServers 30

# MaxRequestsPerChild: The number of connections a thread will handle
# before it is killed. In practise this should be set to 0, which
# disables thread reaping. If you do notice problems with memory
# leakage, then set this to something like 10000.
MaxRequestsPerChild 0

# ViaProxyName: The "Via" header is required by the HTTP RFC, but using
# the real host name is a security concern.  If the following directive
# is enabled, the string supplied will be used as the host name in the
# Via header; otherwise, the server's host name will be used.
ViaProxyName "tinyproxy"

# Filter: This allows you to specify the location of the filter file.
Filter "/etc/tinyproxy/filter"

# FilterURLs: Filter based on URLs rather than domains.
```

```
FilterURLs On

# FilterExtended: Use POSIX Extended regular expressions rather than
# basic.
FilterExtended On

# FilterDefaultDeny: Change the default policy of the filtering system.
# If this directive is commented out, or is set to "No" then the default
# policy is to allow everything which is not specifically denied by the
# filter file.
#
# However, by setting this directive to "Yes" the default policy becomes
# to deny everything which is _not_ specifically allowed by the filter
# file.
FilterDefaultDeny No

# ConnectPort: This is a list of ports allowed by tinyproxy when the
# CONNECT method is used.  To disable the CONNECT method altogether, set
# the value to 0.  If no ConnectPort line is found, all ports are
# allowed (which is not very secure.)
#
# The following two ports are used by SSL.
ConnectPort 443
ConnectPort 563
```

Nella configurazione di esempio mostrata, si fa riferimento al file '`/etc/tinyproxy/filter`', contenente le regole di filtro dei siti o delle pagine. Il contenuto di questo file si intende come ciò che è concesso raggiungere, se è attiva l'opzione '**`FilterDefaultDeny Yes`**' è attiva. Diversamente, con '**`FilterDefaultDeny No`**' si intende escludere ciò che corrisponde alle regole contenute nel file '`/etc/tinyproxy/filter`'. A titolo di esempio, il contenuto del file '`/etc/tinyproxy/filter`' potrebbe essere simile a quello seguente, con lo scopo di filtrare (escludere) ciò che corrisponde alle direttive. Va tenuto conto che

il filtro si riferisce all'indirizzo URI che si intende raggiungere.

```
\.flv$
\.mp3$
\.mp4$
\.ogg$
\.ogv$
\.mpeg$
\.mpg$
\.exe$
poker
casino
jackpot
gambling
scommess
```

Si ricorda che in un sistema GNU/Linux è necessario dare un comando simile a quello seguente per ottenere in pratica la funzionalità di proxy trasparente, tenendo anche conto che ciò riguarda soltanto i nodi che si avvalgono del proxy in qualità di router:

```
# iptables -t nat -A PREROUTING -p tcp --dport 80 -i eth1 ↩
↪          -j REDIRECT --to-port 8080 [Invio]
```

In questo caso, l'interfaccia di rete '**eth1**' è quella rivolta verso la rete che si vuole controllare.

Purtroppo, però, il proxy trasparente non può filtrare una comunicazione cifrata (HTTPS), perché non è possibile ricostruirla. Pertanto, dovendo lasciare libera la comunicazione per il protocollo HTTPS, è facile raggirare un proxy trasparente, tanto che spesso i siti «delicati», come quelli di gioco d'azzardo e quelli di pornografia, utilizzano prevalentemente il protocollo HTTPS (adducendo delle discutibili motivazioni di sicurezza).

42.3 PICS: *Platform for Internet content selection*

PICS, ovvero *Platform for Internet content selection*, è un metodo per classificare, autonomamente, o attraverso l'intervento di un'autorità di classificazione esterna, i contenuti distribuiti elettronicamente attraverso Internet.

42.3.1 Come si classifica

PICS definisce i contenuti attraverso una sorta di linguaggio, nel quale però i valori delle informazioni sono da stabilirsi. Per esempio, un certo contenuto potrebbe essere classificato con il codice seguente:

```
(PICS-1.1 "http://www.weburbia.com/safe/ratings.htm"
 l r
 (s 0))
```

La classificazione si rifà a quanto definito da qualcuno; nell'esempio, si tratta di ciò che viene descritto proprio nella pagina *http://www.weburbia.com/safe/ratings.htm* . Pertanto, non esiste un metodo universale di classificazione, ma solo contestuale.

La classificazione può essere eseguita dall'autore stesso di un lavoro digitale, ma in tal caso si tratta di una semplice dichiarazione libera di ciò che questo contiene, a vantaggio del pubblico. In alternativa, la classificazione può essere eseguita da chi pubblica il materiale, anche in questo caso con lo stesso intento di agevolare il pubblico. La classificazione può avvenire anche per opera di un classificatore certificato, il quale può «firmare» la propria classificazione (in tal caso si usa un'estensione del linguaggio PICS, definita DSig). Segue un esempio di classificazione firmata, tratta da *PICS Signed Labels (DSig) 1.0 Specification* *http://www.w3.org/TR/REC-DSig-label/* :

```
(PICS-1.1 "http://www.gcf.org/v2.5"
   by "John Doe"
   labels
      for "http://www.w3.org/PICS/DSig/Overview"
      extension
        (optional "http://www.w3.org/TR/1998/REC-DSig-label/resinfo-1_0"
           ("http://www.w3.org/TR/1998/REC-DSig-label/SHA1-1_0" "aba21241241=")
           ("http://www.w3.org/TR/1998/REC-DSig-label/MD5-1_0" "cdc43463463="
            "1997-02-05T08:15-0500"))
      extension
        (optional "http://www.w3.org/TR/1998/REC-DSig-label/sigblock-1_0"
           ("AttribInfo"
              ("http://www.w3.org/PICS/DSig/X509-1_0" "efe64685685=")
              ("http://www.w3.org/PICS/DSig/X509-1_0"
               "http://SomeCA/Certs/ByDN/CN=PeterLipp,O=TU-Graz,OU=IAIK")
              ("http://www.w3.org/PICS/DSig/pgpcert-1_0" "ghg86807807=")
              ("http://www.w3.org/PICS/DSig/pgpcert-1_0"
               "http://pgp.com/certstore/plipp@iaik.tu-graz.ac.at"))
           ("Signature" "http://www.w3.org/TR/1998/REC-DSig-label/RSA-MD5-1_0"
              ("byKey" (("N" "aba212412412=")
                        ("E" "3jdg93fj")))
              ("on" "1996-12-02T22:20-0000")
              ("SigCrypto" "3j9fsaJ30SD=")))
      on "1994.11.05T08:15-0500"
      ratings (suds 0.5 density 0 color 1))
```

42.3.2 Come si pubblica la classificazione

In generale, la classificazione di un contenuto elettronico può essere fornita attraverso il protocollo di comunicazione che consente di accedervi. Nel caso più comune, dovrebbe essere inserita nel protocollo HTTP, evidentemente a opera del servizio che pubblica i contenuti (il servente HTTP). Per esempio, a seguito della richiesta da parte di un navigatore di prelevare un certo file, la risposta del servizio potrebbe contenere l'intestazione seguente:

```
HTTP/1.0 200 OK
Date: Tue, 01 Jan 2013 17:44:46 GMT
Last-Modified: Tue, 01 Jan 2012 21:07:24 GMT
PICS-Label:
 (PICS-1.1 "http://www.weburbia.com/safe/ratings.htm"
  l r
  (s 0))
Content-Type: text/html

…
```

Ciò consente di classificare tutti i tipi di file, senza doverli alterare per aggiungervi tale informazione; si pensi alle immagini, ai file audio, ai filmati. Nel caso di documenti HTML, è comunque possibile mettere la classificazione in un elemento '**META**':

```
<!DOCTYPE HTML PUBLIC "ISO/IEC 15445:2000//DTD HTML//EN">
<HTML>
<HEAD>
…
  <META http-equiv="PICS-Label" content='
   (PICS-1.1 "http://www.weburbia.com/safe/ratings.htm"
    l r
    (s 0))
  '>
…
</HEAD>
…
</HTML>
```

Evidentemente, la possibilità di inserire la classificazione in un elemento '**META**', consente all'autore di un opera di eseguire questo compito.

42.3.3 Come si sceglie e come si interpreta la classificazione

Come già accennato, il sistema PICS dà il modo di inserire delle informazioni per la classificazione di un contenuto, ma non definisce le classificazioni in sé. Per questo occorre rivolgersi a dei cataloghi noti. Per esempio, *Safe for kids* *http://www.weburbia.com/safe/ratings.htm* definisce solo tre valori:

- 0 adatto a un pubblico infantile;
- 1 adatto a un pubblico di minori, ma sotto la guida degli adulti;
- 2 adatto a un pubblico adulto.

In pratica, i tre livelli rispecchiano le classificazioni comuni usate per i programmi televisivi (bollino verde, giallo o rosso).

I tre livelli si applicano a un contenuto elettronico con i tre codici seguenti, rispettivamente:

- `(PICS-1.1 "http://www.weburbia.com/safe/ratings.htm" l r (s 0))`
- `(PICS-1.1 "http://www.weburbia.com/safe/ratings.htm" l r (s 1))`
- `(PICS-1.1 "http://www.weburbia.com/safe/ratings.htm" l r (s 2))`

L'interpretazione della classificazione e l'eventuale censura, può avvenire a opera del navigatore stesso, oppure di un programma che si interpone in qualità di «procuratore» (noto comunemente come proxy).

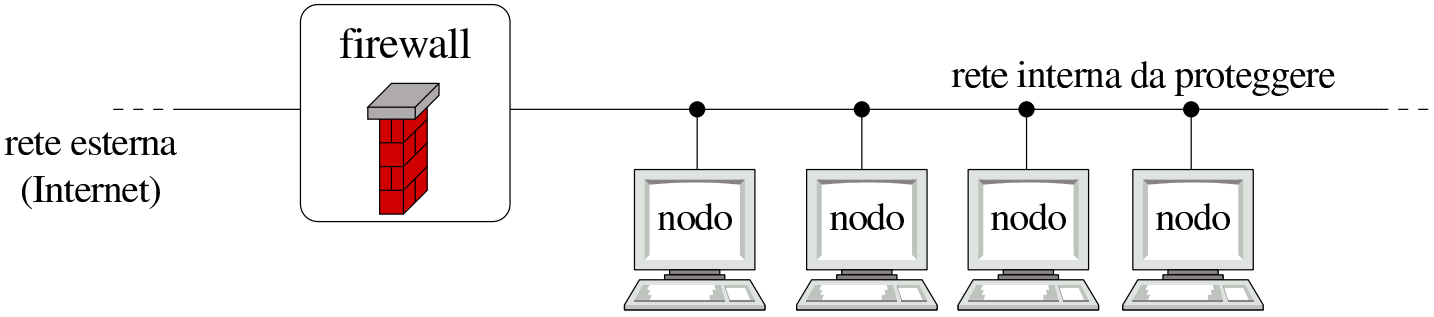
42.4 Introduzione ai concetti di firewall e di NAT/PAT

«

All'interno di una rete, il firewall è un componente che serve a proteggerne una parte rispetto al resto. Di solito, si tratta di qualcosa che si interpone tra una rete interna e una rete esterna, come Internet, per evitare un accesso indiscriminato alla rete interna da parte di nodi collocati all'esterno di questa.

Il firewall, a parte il significato letterale del nome, è una sorta di filtro (passivo o attivo) che si interpone al traffico di rete. Come tale, deve essere regolato opportunamente, in base agli obiettivi che si intendono raggiungere.

Figura 42.24. Il firewall è un filtro che si interpone tra una rete interna e una rete esterna.



Generalmente, i compiti del firewall vengono svolti da un nodo che nella rete si pone in qualità di router, munito di almeno due interfacce di rete: una per l'accesso alla rete esterna e una per la rete interna.

Si distinguono due tipi fondamentali di firewall i quali possono comunque integrarsi: filtri di pacchetto IP (a cui si aggiunge di solito la funzione di NAT[2]) e serventi proxy.

I filtri di pacchetto IP permettono di bloccare o abilitare selettiva-

mente il traffico che attraversa il firewall, definendo i protocolli (o meglio, il tipo di pacchetto), gli indirizzi IP e le porte utilizzate. Questo sistema permette al massimo di controllare i tipi di servizio che possono essere utilizzati in una direzione e nell'altra, da e verso indirizzi IP determinati, ma senza la possibilità di annotare in un registro i collegamenti che sono stati effettuati (salvo eccezioni), né di poter identificare gli utenti che li utilizzano. In un certo senso, questo genere di firewall è come un router su cui si può soltanto filtrare il tipo dei pacchetti che si vogliono lasciare transitare.

I serventi proxy rappresentano una sorta di intermediario che si occupa di intrattenere le connessioni per conto di qualcun altro nella rete interna (sezione 42.2). Dal momento che il proxy ha un ruolo attivo nelle connessioni, può tenere un registro delle azioni compiute; eventualmente può anche tentare di identificare l'utente che lo utilizza.

42.4.1 Firewall in forma di filtri di pacchetto

Il filtro di pacchetto può intervenire al terzo o al massimo al quarto livello del modello ISO-OSI. In altri termini, è in grado di identificare e filtrare i pacchetti in base agli indirizzi IP, alle porte utilizzate e a poche altre informazioni, come elencato nella tabella 42.26 a titolo di esempio.

Figura 42.25. Un firewall che funziona come filtro di pacchetto IP, può interviene al terzo e quarto livello del modello ISO-OSI.

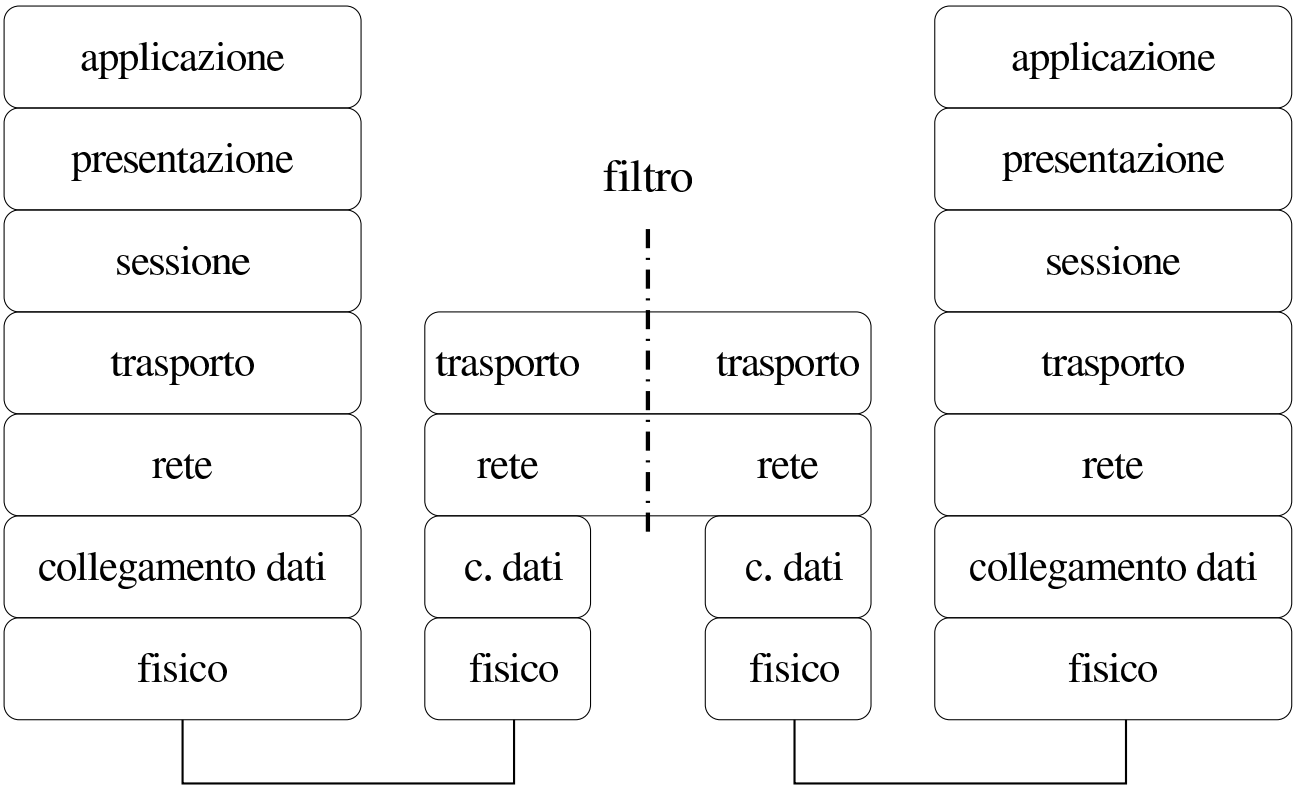


Tabella 42.26. Caratteristiche tipiche dei pacchetti che possono essere prese in considerazione per il filtro.

Caratteristica	Annotazioni
interfaccia di rete	l'interfaccia interessata nel nodo locale
indirizzo IP di origine	
indirizzo IP di destinazione	
protocollo	TCP, UDP, ICMP
porta di origine	TCP o UDP
porta di destinazione	TCP o UDP
messaggio ICMP	rappresentato da un numero
pacchetto frammentato	frammentazione a livello IP
pacchetto SYN	richiesta inizio di connessione TCP

Si tratta di una limitazione significativa che comporta i problemi maggiori nella configurazione corretta di un filtro del genere, in base ai fini che si tendono ottenere. Volendo esprimere la cosa attraverso un esempio molto semplice, un filtro di questo tipo non può intervenire esattamente ed esclusivamente sul «protocollo HTTP»; al massimo si può intercettare il transito dei pacchetti TCP in arrivo verso la porta 80, se si vuole impedire l'instaurarsi di connessioni a un servizio HTTP locale, oppure in uscita se si vuole impedire di raggiungere servizi esterni. Ma questo non vuol dire che si blocca il protocollo HTTP: è solo un intervento fatto in modo tale da arrivare a un risultato molto vicino a quello atteso.

Tabella 42.27. Messaggi ICMP.

Tipo	Codice	Nome del tipo	Nome del codice	Chi lo utilizza
0		echo-reply		risposta a un ping (pong)
1				
2				
3		destination-unreachable		traffico TCP e UDP
3	0		network-unreachable	
3	1		host-unreachable	
3	2		protocol-unreachable	
3	3		port-unreachable	
3	4		fragmentation-needed	
3	5		source-route-failed	
3	6		network-unknown	

Tipo	Codice	Nome del tipo	Nome del codice	Chi lo utilizza
3	7		host-unknown	
3	8			
3	9		network-prohibited	
3	10		host-prohibited	
3	11		TOS-network-unreachable	
3	12		TOS-host-unreachable	
3	13		communication-prohibited	
3	14		host-precedence-violation	
3	15		precedence-cutoff	
4		source-quench		
5		redirect		instradamento dei pacchetti
5	0		network-redirect	
5	1		host-redirect	
5	2		TOS-network-redirect	
5	3		TOS-host-redirect	
6				
7				
8		echo-request		ping
9		router-advertisement		
10		router-solicitation		
11		time-exceeded (ttl-exceeded)		traceroute

Tipo	Codice	Nome del tipo	Nome del codice	Chi lo utilizza
11	0		ttl-zero-during-transit	
11	1		ttl-zero-during-reassembly	
12		parameter-problem		
12	0		ip-header-bad	
12	1		required-option-missing	
13		timestamp-request		
14		timestamp-reply		
15		information-request		
16		information-reply		
17		address-mask-request		
18		address-mask-reply		

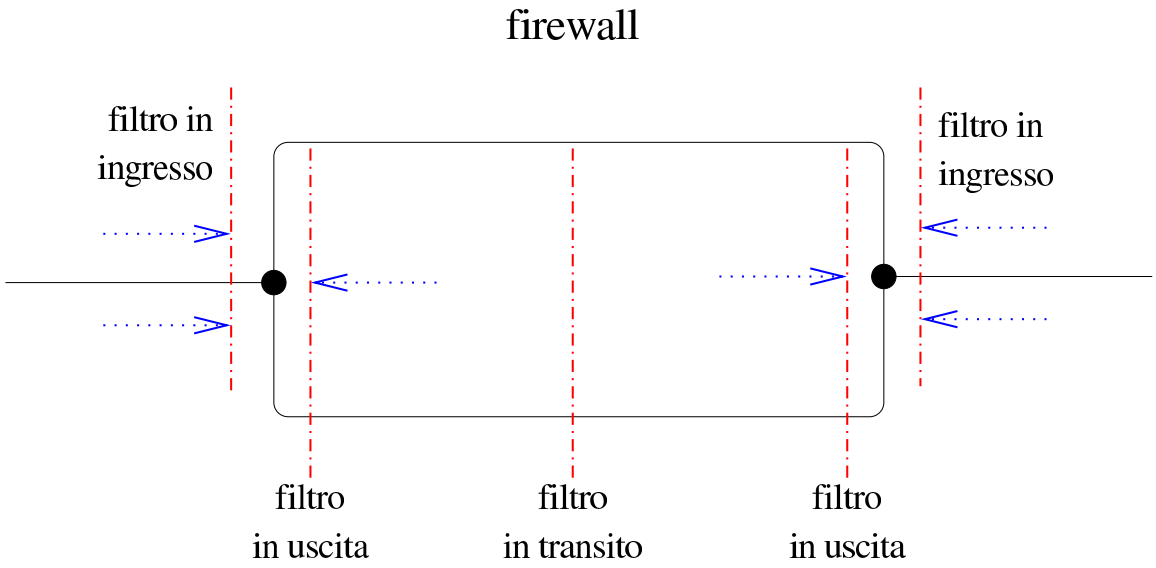
Un'altra cosa importante da considerare è il fatto che i pacchetti frammentati a livello di protocollo IP, possono essere identificati come frammenti, mentre diventa impossibile conoscere le altre caratteristiche (TCP o UDP).

42.4.1.1 Punto di applicazione e significato dell'intercettazione

Teoricamente, ammesso che l'applicazione utilizzata come filtro (assieme al kernel) sia abbastanza sofisticata da permetterlo, si può intervenire in tre punti differenti: nel transito dei pacchetti da un'in-

terfaccia a un'altra, nei pacchetti in arrivo attraverso una data interfaccia e nei pacchetti in uscita. La distinzione è importante perché i risultati pratici che si ottengono possono essere molto diversi a seconda del punto in cui si inserisce il filtro.

Figura 42.28. Punti di inserzione di un filtro di pacchetto.



Anche senza fare un riferimento preciso alle interfacce di rete coinvolte, si pensi al caso in cui si intercettano in uscita i pacchetti ICMP di tipo 8, *echo-request*, allo scopo di bloccarne il transito. In tal caso, ci si impedisce di usare il Ping verso l'esterno; al contrario, intercettando lo stesso tipo di pacchetto, ma in ingresso, il suo blocco impedisce ai nodi esterni di usare il Ping verso il proprio elaboratore. Se invece l'intercettazione avvenisse nella fase di transito, questo potrebbe servire solo a impedire il Ping che riguarda altri nodi, oppure solo l'interfaccia del lato opposto.

I pacchetti intercettati possono essere trattati in modi differenti:

- possono essere lasciati passare;
- possono essere bloccati;

- possono essere bloccati, inviando all'origine un messaggio di rifiuto attraverso un pacchetto ICMP;
- possono essere semplicemente tenuti sotto controllo (contabilizzati).

Eventualmente, la contabilizzazione del traffico può essere implicita in ogni tipo di intercettazione.

A seconda dell'organizzazione logica del firewall, può darsi che l'intercettazione di un pacchetto in ingresso, implichi la stessa cosa sia per i pacchetti destinati al firewall, sia per i pacchetti che lo attraverserebbero per raggiungere altre destinazioni, oppure le due cose potrebbero essere distinte. Nello stesso modo potrebbe esserci una differenza di funzionamento nell'intercettazione in uscita. È evidente che, a seconda del tipo di firewall utilizzato, deve essere chiarito in modo preciso il campo di azione di ogni filtro.

42.4.1.2 Ricomposizione dei pacchetti frammentati

In generale, un nodo di rete che svolge funzioni di firewall dovrebbe trovarsi in un «passaggio obbligato» della rete, per evitare che i pacchetti possano utilizzare percorsi alternativi. In questo senso, è opportuno che tale nodo possa ricomporre i pacchetti frammentati a livello IP, in modo da riunire assieme tutte le informazioni necessarie a identificare i pacchetti, proprio per poter attuare effettivamente il controllo che il firewall deve fare.

In mancanza della possibilità di ricomporre i pacchetti frammentati, il firewall può individuare nei frammenti solo gli indirizzi IP, del

mittente e del destinatario, oltre al riconoscere che si tratta di frammenti. Diventa impossibile l'identificazione delle porte, TCP o UDP, oppure i messaggi ICMP.

42.4.2 Esempi di utilizzo di firewall

È il caso di raccogliere qualche esempio schematico del modo in cui si potrebbe configurare un firewall che utilizza la tecnica del filtro di pacchetto. Le impostazioni vengono indicate in forma di tabella, secondo lo schema seguente:

Azione	Pos.	Prot.	IP srg		IP dst		ICMP	Int.	
1	2	3	4	5	6	7	8	9	10

I campi delle righe della tabella hanno il significato descritto nell'elenco che segue, tenendo conto che i valori mancanti vengono considerati indifferenti:

1. azione del filtro: blocco, rifiuto o altro;
2. posizione del filtro: in ingresso, in uscita, in transito o altro;
3. protocollo: TCP, UDP, ICMP;
4. indirizzi IP di origine;
5. porte TCP o UDP di origine;
6. indirizzi IP di destinazione;
7. porte TCP o UDP di destinazione;
8. messaggio ICMP, indicando il tipo e il codice eventuale (***tipo***[***/codice***]);
9. interfaccia di rete coinvolta;

10. altre caratteristiche.

Si osservi in particolare che gli indirizzi IP si indicano nella forma '***indirizzo / maschera***', dove la maschera si esprime attraverso un intero che rappresenta una quantità iniziale di bit da impostare a uno. Inoltre, gli indirizzi e le porte possono essere prefissati da un punto esclamativo che indica la negazione logica, ovvero tutti gli altri indirizzi o tutte le altre porte.

- Si impedisce l'ingresso a ogni pacchetto proveniente dagli indirizzi 192.168.∗.∗:

Azione	Pos.	Prot.	IP srg	IP dst	ICMP Int.
blocco	ingresso		192.168.0.0/16	0/0	

- Si impedisce l'ingresso ai pacchetti ICMP provenienti dagli indirizzi 192.168.∗.∗:

Azione	Pos.	Prot.	IP srg	IP dst	ICMP Int.
blocco	ingresso	ICMP	192.168.0.0/16	0/0	

- Si impedisce l'ingresso dei pacchetti provenienti dall'interfaccia ***x***, contenenti come mittente indirizzi tipici delle reti private. In pratica, si presume che sia impossibile ricevere pacchetti di questo tipo da tale interfaccia, perché la rete privata è connessa su un'altra; pertanto, pacchetti del genere possono essere solo contraffatti.

Azione	Pos.	Prot.	IP srg	IP dst	ICMP	Int.
blocco	ingresso		10.0.0.0/8	0/0		*x*
blocco	ingresso		172.16.0.0/12	0/0		*x*
blocco	ingresso		192.168.0.0/16	0/0		*x*

- Si impedisce l'attraversamento di pacchetti della classe D e E:

Azione	Pos.	Prot.	IP srg	IP dst	ICMP	Int.
blocco	transito		224.0.0.0/3	0/0		

- Consente l'attraversamento ai pacchetti TCP per raggiungere presumibilmente un servizio TELNET:

Azione	Pos.	Prot.	IP srg	IP dst	ICMP	Int.
consente	transito	TCP	0/0	0/0	23	

- Blocca il transito delle comunicazioni riferite alla gestione remota di applicazioni X. Si presume si possano gestire un massimo di 10 serventi grafici simultaneamente.

Azione	Pos.	Prot.	IP srg		IP dst		ICMP	Int.
blocco	transito	TCP	0/0	6000-6009	0/0			
blocco	transito	TCP	0/0		0/0	6000-6009		

- Blocca l'ingresso e l'uscita delle comunicazioni riferite alla gestione remota di applicazioni X. In questo caso, si protegge il nodo che funge da firewall.

Azione	Pos.	Prot.	IP srg		IP dst	ICMP Int.
blocco	ingresso	TCP	0/0	6000-6009	0/0	
blocco	uscita	TCP	0/0		0/0	6000-6009

42.4.3 Annotazioni finali sulla gestione di un firewall

Vanno tenute a mente alcune cose quando si configura un firewall attraverso il filtro di pacchetto, per evitare di compromettere le funzionalità che invece si vogliono mantenere.

42.4.3.1 Pacchetti ICMP

È già stato accennato il fatto che non si deve bloccare il transito dei pacchetti del protocollo ICMP. Il messaggio di tipo 3, *destination-unreachable*, è indispensabile nei protocolli TCP e UDP per sapere che un certo indirizzo non è raggiungibile; bloccandolo, si attende senza sapere il perché.

Il protocollo ICMP viene usato anche nella determinazione automatica della dimensione massima dei pacchetti (*MTU discovery*). Mancando la possibilità di ricevere questi pacchetti ICMP, il funzionamento delle comunicazioni potrebbe essere compromesso seriamente.

42.4.3.2 Pacchetti UDP

«

I protocolli che si basano su UDP sono usati frequentemente nell'ambito di servizi locali, come NIS e NFS. Tra le altre cose, questi servizi tendono a fare viaggiare informazioni particolarmente delicate che non dovrebbero essere accessibili dall'esterno. Per questa ragione, è normale che venga impedito il transito dei pacchetti UDP. Tuttavia, capita che proprio il servizio DNS (per la risoluzione dei nomi), possa averne bisogno.

Azione	Pos.	Prot.	IP srg	IP dst	ICMP Int.
blocco	transito	UDP	0/0	0/0	

Per la precisione, il servizio DNS può usare pacchetti UDP o connessioni TCP, a seconda della dimensione di questi. Così, il blocco eventuale di tale servizio si avvertirebbe solo in modo intermittente, complicando l'individuazione del problema.

Generalmente, un servizio DNS collocato in una posizione tale per cui non possa inviare o ricevere pacchetti UDP dall'esterno, si deve avvalere necessariamente di un altro collocato al di fuori di tale blocco. Infatti, in questo modo userebbe solo il protocollo TCP.

Eventualmente, il firewall potrebbe essere configurato espressamente per consentire il transito di questi pacchetti legati al servizio DNS. Nell'esempio seguente si suppone che il servizio DNS in questione sia collocato nel nodo 196.1.2.3:

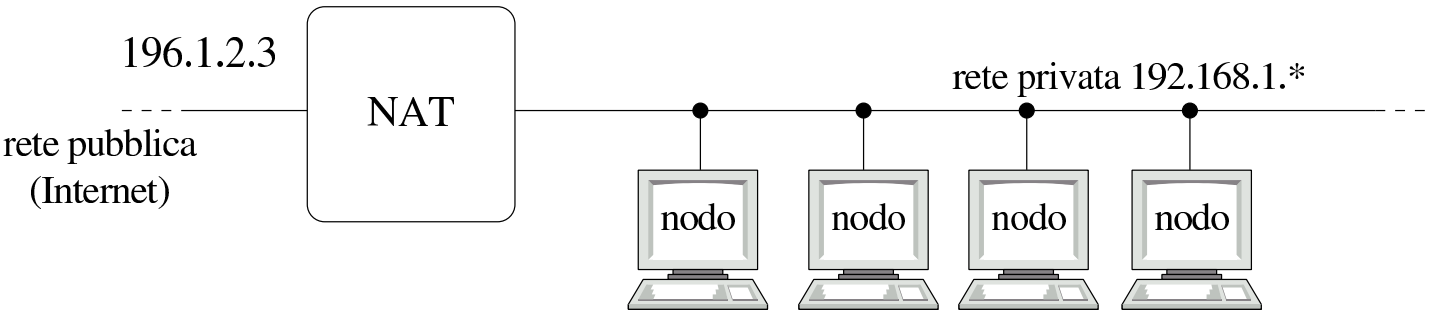
Azione	Pos.	Prot.	IP srg		IP dst		ICMP Int.
accetta	transito	UDP	0/0	53	196.1.2.3		
accetta	transito	TCP	0/0	53	196.1.2.3		
accetta	transito	UDP	196.1.2.3		0/0	53	
accetta	transito	TCP	196.1.2.3		0/0	53	

42.4.4 NAT/PAT

Il NAT, o *Network address translation*, è una tecnica descritta nell'RFC 1631, con la quale un nodo di rete speciale acquista funzionalità simili a quelle di un router, intervenendo però sui pacchetti, allo scopo di sostituire gli indirizzi IP reali con altri indirizzi più convenienti.

Il problema a cui fa riferimento l'RFC 1631 riguarda la possibilità di riutilizzare dinamicamente gli indirizzi IP riservati alle reti private, permettendo ugualmente a tali reti di accedere all'esterno, pur non essendo questi univoci a livello globale. Si osservi l'esempio della figura 42.39.

Figura 42.39. Esempio di router NAT: l'indirizzo IP 196.1.2.3 è un esempio che sta a rappresentare un indirizzo univoco riconosciuto nella rete esterna.

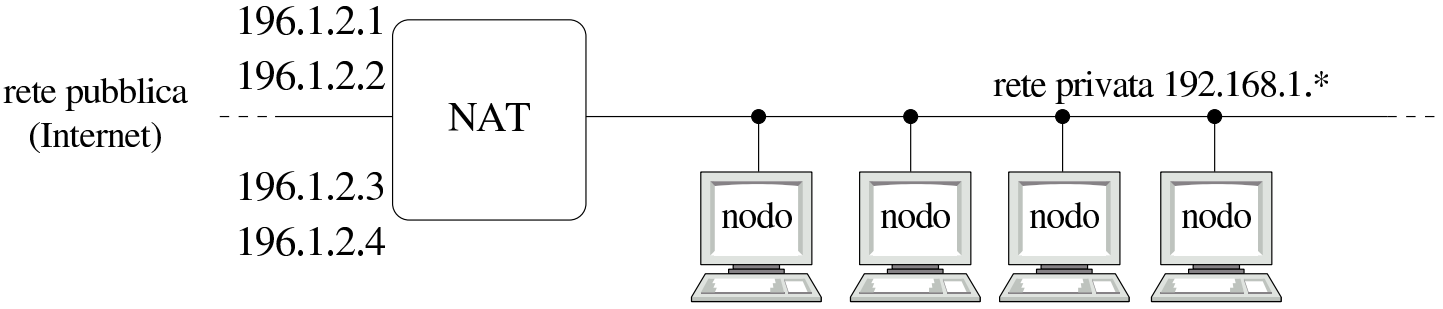


In condizioni normali, gli indirizzi IP 192.168.1.* non hanno la possibilità di essere riconosciuti univocamente nella rete globale, pertanto i nodi relativi non hanno la possibilità di accedere all'esterno. Attraverso il meccanismo NAT e le sue varianti, si può ottenere questo risultato anche se poi è necessario accettare qualche compromesso.

42.4.4.1 Conversione dinamica degli indirizzi IP

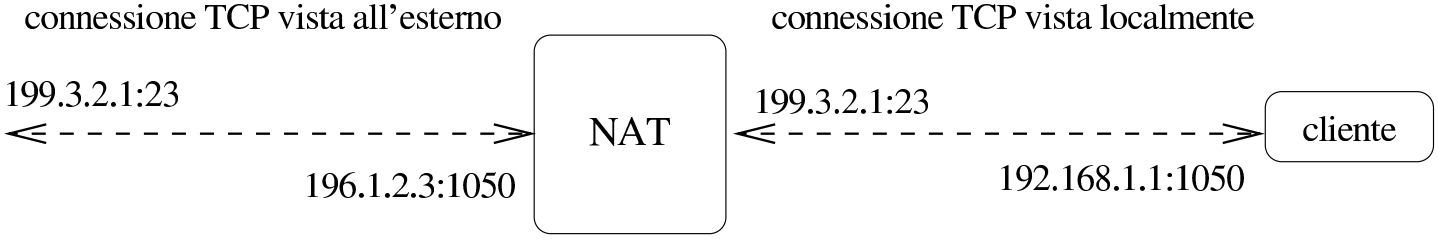
Nella sua impostazione più semplice, un router NAT può gestire un numero ristretto di indirizzi IP univoci, da abbinare dinamicamente a degli indirizzi IP locali privati.

Figura 42.40. Utilizzo dinamico di un gruppo ristretto di indirizzi IP univoci.



Osservando la figura 42.40 si può vedere che il nodo che ha il ruolo di router NAT dispone di un accesso all'esterno con quattro diversi indirizzi IP univoci. In questo modo, in base alle richieste provenienti dalla rete interna, può abbinare temporaneamente un indirizzo univoco a un indirizzo privato interno. Per esempio, in un dato momento, i pacchetti provenienti o destinati all'indirizzo 192.168.1.1 potrebbero essere modificati in modo da rimpiazzare tale indirizzo con quello univoco 196.1.2.3.

Figura 42.41. Una connessione TCP rielaborata da un router NAT.



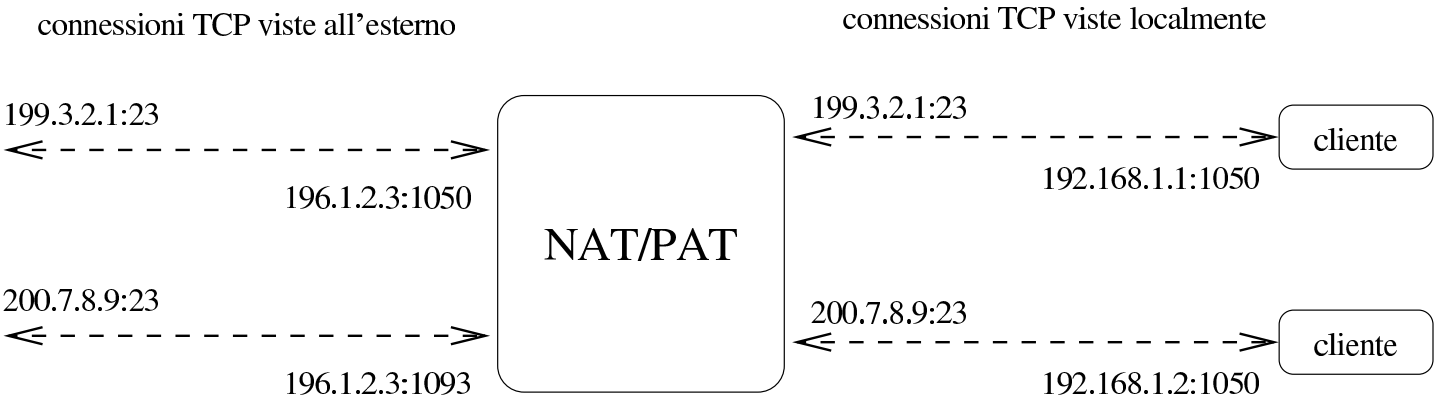
In questo caso, il router NAT si limita a sostituire ai pacchetti gli indirizzi IP di origine o di destinazione, in base all'attribuzione dinamica stabilita.

La conversione degli indirizzi può anche essere dinamica solo in parte, in cui alcuni indirizzi univoci sono abbinati stabilmente ad altrettanti indirizzi della rete privata. Questo permette a tali nodi di essere raggiungibili anche da un accesso esterno, senza che debbano essere loro per primi a instaurare una connessione.

42.4.4.2 Conversione dinamica delle porte: PAT

Oltre alla sostituzione degli indirizzi, un router NAT più evoluto può gestire anche la sostituzione delle porte TCP e UDP; in tal caso si parla anche di PAT, ovvero di *Port address translation*. Spesso, la realtà è tale per cui diventa indispensabile questo approccio, disponendo di un solo indirizzo IP univoco.

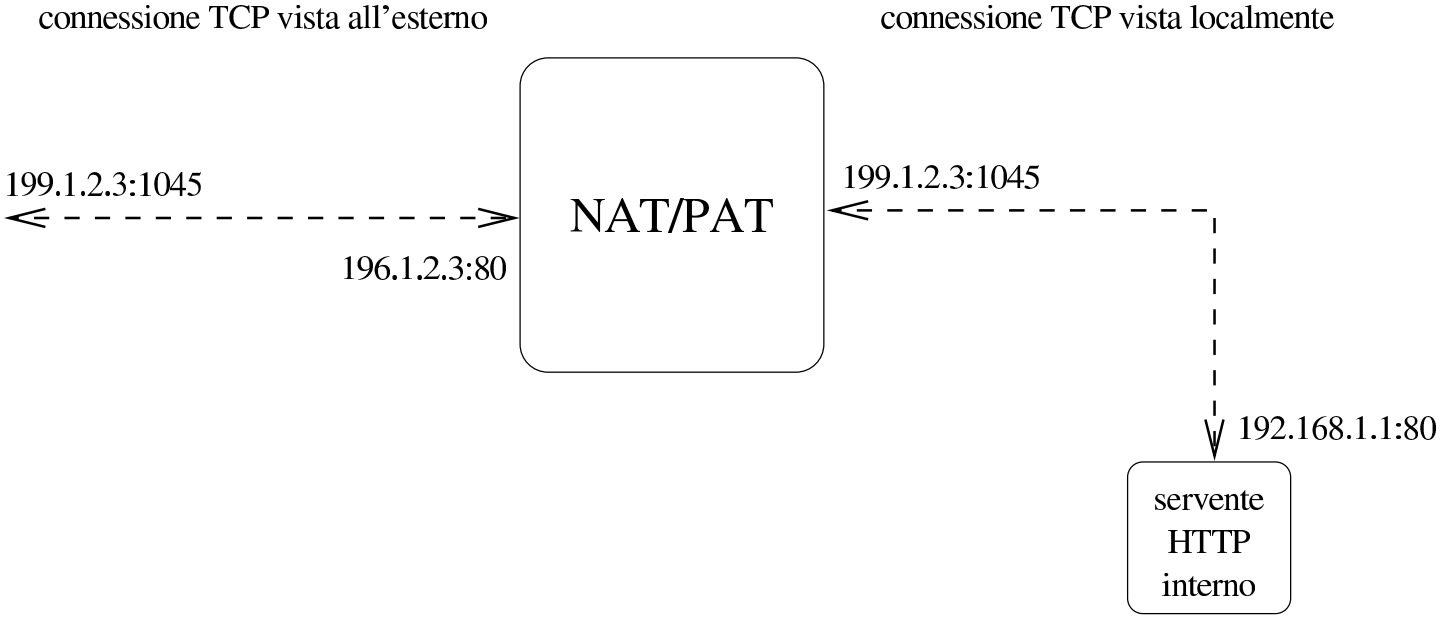
Figura 42.42. Due connessioni TCP indipendenti, rielaborate attraverso un procedimento NAT/PAT.



La figura 42.42 mostra il caso in cui i nodi 192.168.1.1 e 192.168.1.2 instaurano due connessioni TELNET indipendenti attraverso un router NAT/PAT. In questo caso, il NAT/PAT non si limita a sostituire ai pacchetti gli indirizzi IP di origine o di destinazione, intervenendo anche sui numeri di porta TCP.

Utilizzando il meccanismo NAT/PAT in questo modo, considerando che gli accessi iniziano sempre dalla parte della rete interna, per raggiungere indirizzi esterni, è normale che le porte di origine siano sempre non privilegiate, cioè siano maggiori o uguali a 1024. Il router NAT/PAT potrebbe anche essere utilizzato per dirigere le connessioni originate dall'esterno e dirette a porte determinate (probabilmente nel gruppo di porte privilegiato) a nodi ben precisi nella rete locale, solitamente per raggiungere dei servizi realizzati lì. Per fare questo occorre quindi che il router NAT/PAT annoti delle ridirezioni statiche riferite alla richiesta di porte particolari. Per esempio, la figura 42.43 mostra un router NAT/PAT che ridirige sistematicamente le connessioni provenienti dall'esterno, dirette alla porta 80, verso il nodo locale 192.168.1.1 alla stessa porta 80, dal momento che questo offre un servizio HTTP.

Figura 42.43. Ridirezione del traffico diretto a un servente HTTP interno.



42.4.4.3 Problemi

Il meccanismo NAT/PAT, come qualunque altra forma di rimaneggiamento dei pacchetti allo scopo di sostituire gli indirizzi IP o le porte TCP/UDP, funziona bene solo quando i protocolli utilizzati a livello di sessione, ovvero il quinto del modello ISO-OSI, non prendono iniziative autonome allo scopo di gestire gli indirizzi e le porte. In altri termini, tutto funziona bene se non si inseriscono informazioni sugli indirizzi e sulle porte al di sopra del livello del TCP o di UDP.

Il classico esempio problematico è dato dall'FTP che negozia con la controparte l'instaurazione di una connessione TCP aggiuntiva, attraverso informazioni contenute nell'area «dati» dei pacchetti. In questo modo, un router NAT/PAT ingenuo riuscirebbe a trasferire solo la prima connessione TCP.

Evidentemente, un router NAT/PAT evoluto dovrebbe essere consa-

pevole, non solo dei protocolli IP, TCP e UDP, ma anche di tutti i protocolli che si inseriscono al di sopra di questi, in modo da intervenire opportunamente.

Un'ultima cosa da considerare riguarda anche il problema dei pacchetti frammentati, che devono essere ricomposti quando si utilizza il meccanismo NAT/PAT.

42.5 Firewall con kernel Linux

«

Il kernel Linux può gestire direttamente il filtro dei pacchetti IP, cosa che quindi rappresenta la scelta più semplice per la realizzazione di un firewall con questo sistema operativo. A parte le limitazioni che può avere un tale tipo di firewall, il suo inserimento nella rete non genera effetti collaterali particolari, dal momento che poi non c'è bisogno di utilizzare software speciale per gli elaboratori che lo devono attraversare, come avviene invece nel caso di un firewall di tipo proxy.

Trattandosi di un'attività del kernel, è necessario che questo sia stato predisposto in fase di compilazione, oppure sia accompagnato dai moduli necessari (sezione 8.3.7). Inoltre, è opportuno aggiungere anche le funzionalità di ricomposizione dei pacchetti frammentati, oltre che le funzionalità relative al NAT (*Network address translation*).

L'attraversamento dei pacchetti tra un'interfaccia e l'altra è controllato dalla funzionalità di *forwarding-gatewaying*, che in passato andava inserita esplicitamente nel kernel. In generale, il kernel non permette questo attraversamento che deve essere abilitato attraverso un comando particolare. Per IPv4:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward [Invio]
```

Per IPv6:

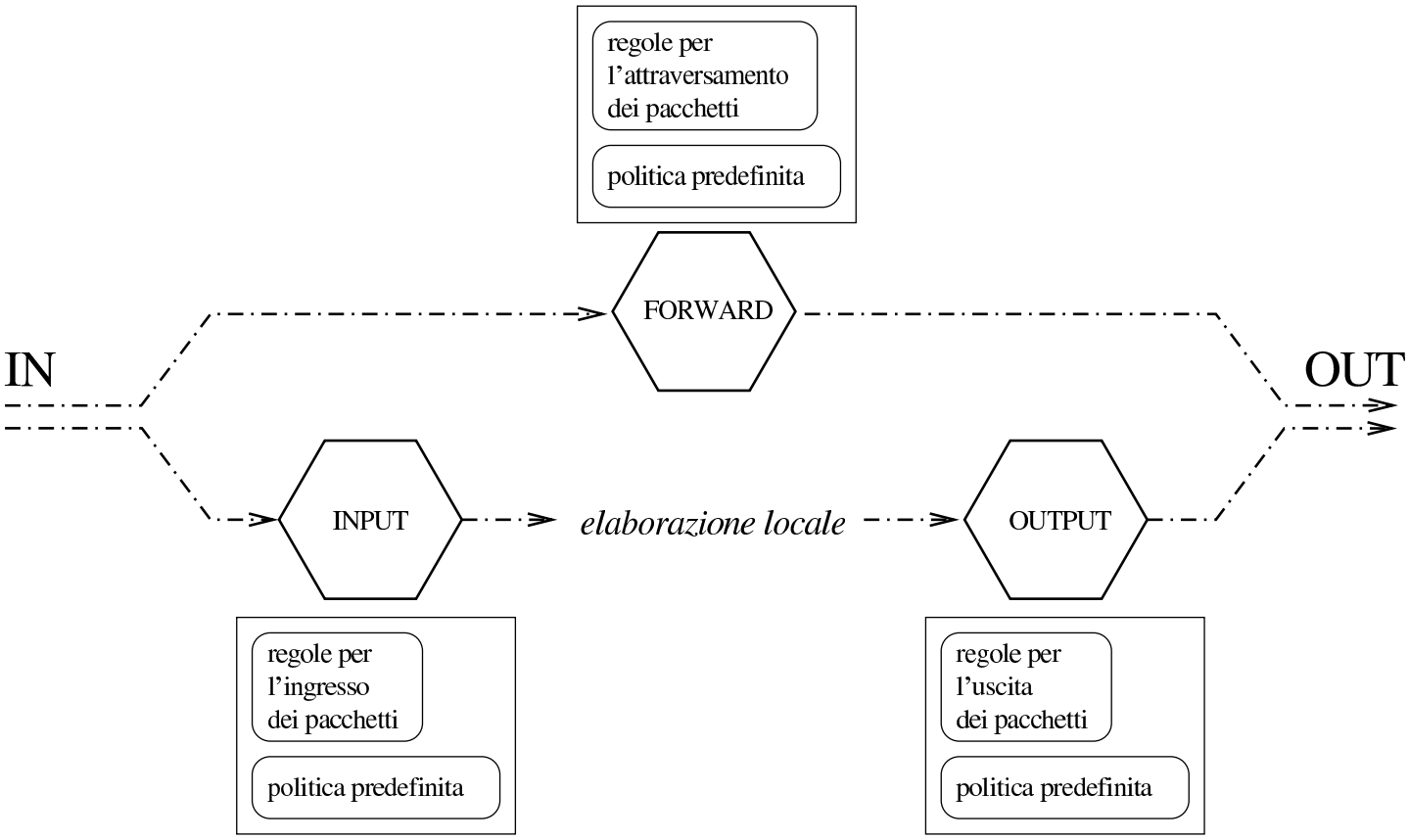
```
# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding [Invio]
```

42.5.1 Schema generale di funzionamento del kernel

I kernel Linux 2.4.* e Linux 2.6.* suddividono le funzionalità di trattamento dei pacchetti IP in «tabelle». Nell'ambito di ogni tabella ci possono essere diversi punti di controllo, denominati *chain*, i quali possono essere programmati per catturare i pacchetti IP e deciderne la loro sorte. A seconda delle circostanze, un pacchetto IP può essere sottoposto alla verifica di uno o più di questi punti di controllo, i quali vengono programmati in base a delle ***regole***. Quando un pacchetto sottoposto a controllo corrisponde a una regola, la sua sorte viene definita dall'***obiettivo*** di questa (ammesso che sia stato definito).

La tabella relativa alla gestione del firewall è denominata '**`filter`**' e si compone di tre punti di controllo, denominati '**`INPUT`**', '**`FORWARD`**' e '**`OUTPUT`**', a indicare rispettivamente i pacchetti in ingresso, quelli in transito e quelli in uscita. Gli obiettivi più frequenti sono due, '**`ACCEPT`**' e '**`DROP`**', riferiti rispettivamente al permesso di attraversamento del punto di controllo, oppure al blocco ed eliminazione del pacchetto intercettato.

Figura 42.44. Schema di intercettazione da parte dei punti di controllo relativi alla gestione del firewall.



Un pacchetto proveniente da un'interfaccia qualunque, diretto allo stesso firewall, è soggetto al controllo di ingresso; un pacchetto passante viene sottoposto al controllo di inoltro; un pacchetto che deve uscire attraverso un'interfaccia del firewall, perché generato da un processo locale, è sottoposto al controllo di uscita.

Quando un pacchetto IP viene analizzato in un punto di controllo e all'interno di questo non c'è alcuna regola che lo prenda in considerazione, la sua sorte è stabilita dalla ***politica predefinita*** per quel contesto (*policy*). Generalmente, questa politica è tale per cui gli viene concesso il transito.

42.5.2 IPTables per l'amministrazione del firewall

La gestione del filtro di pacchetto IP dei kernel 2.4.* e 2.6.* avviene per mezzo di IPTables,[3] ovvero l'eseguibile '**iptables**' per il controllo di IPv4 e '**ip6tables**' per il controllo di IPv6. Dal momento che le funzionalità di firewall del kernel sono piuttosto estese, la sintassi di questo programma è molto articolata, per cui se ne può apprendere l'utilizzo solo gradualmente.

Inoltre, è bene chiarire subito che le funzionalità di firewall del kernel non possono essere definite attraverso un file di configurazione; quindi, al massimo, tutto quello che si può fare è la realizzazione di uno script contenente una serie di comandi con IPTables.

IPTables interviene su un ***elenco di regole*** riferite alle funzionalità di controllo dei pacchetti IP del kernel, dove la gestione particolare riferita alle funzionalità di firewall riguarda la tabella '**filter**'. Il meccanismo è comunque simile a quello della gestione della tabella degli instradamenti di un router. L'ordine in cui sono elencate tali regole è importante, quindi si deve poter distinguere tra l'inserimento di una regola all'inizio, alla fine o in un'altra posizione dell'elenco esistente (elenco riferito sempre a un certo punto di controllo).

Salvo eccezioni particolari, descritte nel contesto appropriato, la sintassi di massima per l'utilizzo di IPTables è quella seguente:

```
iptables [-t tabella] opzione_di_comando punto_di_controllo ↩
↪        [regola] [obiettivo]
```

```
ip6tables [-t tabella] opzione_di_comando punto_di_controllo ↩
↪        [regola] [obiettivo]
```

La tabella serve a stabilire il contesto di intervento; il nome dell'eseguibile ('**iptables**' o '**ip6tables**') definisce il tipo di protocolli di competenza (IPv4 o IPv6). La tabella predefinita è proprio quella riferita alle funzionalità di firewall, ovvero '**filter**'.

In generale, l'utilizzo di '**iptables**' o di '**ip6tables**' è uguale, salvo le differenze che riguardano il modo di rappresentare gli indirizzi e salvo piccole eccezioni. Nel capitolo si accenna alle differenze solo quando necessario, tenendo conto che di solito basta sostituire il nome dell'eseguibile per cambiare il contesto.

L'opzione di comando serve a stabilire il tipo di intervento nel sistema di gestione del firewall. L'elenco seguente si riferisce alle opzioni che permettono la cancellazione o l'inserimento delle regole in un punto di controllo:

`-F` `--flush`	elimina tutte le regole del punto di controllo specificato, oppure di tutta la tabella;
`-D` `--delete`	elimina una o più regole dal punto di controllo specificato;
`-A` `--append`	aggiunge una regola in coda a quelle del punto di controllo selezionato;

`-I` `--insert`	inserisce una regola in una posizione stabilita del punto di controllo selezionato;
`-R` `--replace`	sostituisce una regola del punto di controllo selezionato.

Altre opzioni non modificano le regole; in particolare:

`-L` `--list`	elenca le regole di un uno o di tutti i punti di controllo della tabella;
`-P` `--policy`	cambia la politica predefinita per il punto di controllo specificato.

Altre opzioni vengono mostrate quando più opportuno.

Come già accennato, il punto di controllo viene indicato attraverso un nome. Si tratta di '**`INPUT`**', '**`FORWARD`**' e '**`OUTPUT`**', i quali intuitivamente fanno riferimento all'ingresso, al transito e all'uscita.

IPTables permette di gestire delle regole all'interno di contenitori aggiuntivi a cui si fa riferimento a partire da regole inserite nei punti di controllo normali. Nella terminologia di IPTables si parla sempre di *chain*, sia per indicare i punti di controllo standard, sia per indicare questi elenchi di regole aggiuntive.

Infine, una regola comune è conclusa con l'indicazione di un obiettivo. L'obiettivo è la definizione della sorte da dare al pacchetto intercettato, indicata attraverso una parola chiave. Le più importanti per iniziare ad apprendere la configurazione del firewall sono: '**ACCEPT**', '**DROP**' e '**REJECT**'.

ACCEPT	Consente il transito del pacchetto.
DROP	Impedisce il transito del pacchetto, limitandosi a ignorarlo.
REJECT	Impedisce il transito del pacchetto notificando all'origine il rifiuto (viene inviato un messaggio ICMP specificante che il pacchetto è stato rifiutato).

Segue la descrizione di alcuni esempi.

```
iptables [-t filter] -A INPUT regola -j DROP
```

Lo schema mostra l'aggiunta di una regola di ingresso, non meglio definita, per la quale viene applicato l'obiettivo '**DROP**'.

```
iptables [-t filter] -R INPUT 1 regola -j DROP
```

Lo schema mostra la sostituzione della prima regola di ingresso con un'altra regola non meglio definita, per la quale viene applicato l'obiettivo '**DROP**'.

```
iptables [-t filter] -I INPUT 1 regola -j ACCEPT
```

Lo schema mostra l'inserimento nella prima posizione di una rego-

la di ingresso per la quale viene consentito il transito dei pacchetti (‘**ACCEPT**’).

```
iptables [-t filter] -D INPUT 2
```

Questo schema mostra l’eliminazione della seconda regola di ingresso.

```
iptables [-t filter] -F INPUT
```

Questo schema mostra l’eliminazione di tutte le regole di ingresso.

```
iptables [-t filter] -F
```

Questo schema mostra l’eliminazione di tutte le regole di tutti i punti di controllo.

```
iptables [-t filter] -P INPUT DROP
```

Cambia la politica predefinita di ingresso specificando che, in mancanza di regole, i pacchetti devono essere bloccati.

Negli esempi è stato sottolineato l’uso facoltativo dell’opzione ‘**-t**’ per identificare precisamente la tabella su cui intervenire. Dal momento che la tabella ‘**filter**’ è quella predefinita, nel capitolo non viene più utilizzata tale opzione.

42.5.2.1 Un po' di confidenza con IPTables per la gestione del firewall

«

Data la complessità delle funzionalità di filtro di pacchetto del kernel, anche l'uso di IPTables è piuttosto articolato. Prima di iniziare a vedere come si possono definire le regole, conviene fare qualche esperimento che serva a introdurre l'uso di questo programma.

Gli esempi fanno riferimento a IPv4, ma dovrebbero andare bene anche per IPv6, salva la sostituzione degli indirizzi.

La prima cosa da sapere è il modo in cui si ottiene la visualizzazione della situazione dei punti di controllo che compongono la tabella.

```
# iptables -L[Invio]
```

In questo modo si ottiene la situazione di tutti i punti di controllo (ed eventualmente anche dei raggruppamenti di regole aggiuntivi). Inizialmente si dovrebbe osservare la situazione seguente:

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Quello che si vede è la situazione normale del sistema prima di iniziare a inserire delle regole; tutto quello che c'è sono le politiche predefinite per ogni punto di controllo.

Se si è interessati a conoscere solo la situazione di un punto di controllo particolare, basta aggiungere il nome di questo. Per esempio, per limitare il risultato al solo punto di controllo di ingresso si può usare il comando seguente:

```
# iptables -L INPUT [Invio]

  Chain INPUT (policy ACCEPT)
  target     prot opt source               destination
```

Per verificare l'effetto del blocco del traffico attraverso uno dei punti di controllo si può agire sommariamente sulla politica predefinita; per esempio si può bloccare il transito dei pacchetti in ingresso con il comando seguente:

```
# iptables -P INPUT DROP [Invio]
```

Questo tipo di blocco è totale e interviene anche nell'interfaccia virtuale che identifica il sistema locale: '**lo**'. Basta provare a fare un ping verso il nodo locale per accorgersi che non si ottiene più alcuna risposta.[4]

```
$ ping localhost [Invio]
```

Un risultato simile si potrebbe ottenere utilizzando l'obiettivo '**REJECT**'. In alternativa si può intervenire nel punto di controllo di uscita; nell'esempio seguente si ripristina prima la politica di '**ACCEPT**' per i pacchetti in ingresso.

```
# iptables -P INPUT ACCEPT [Invio]

# iptables -P OUTPUT DROP [Invio]
```

Con il ping si ottiene in pratica lo stesso risultato, con la differenza che i pacchetti trasmessi vengono bloccati prima di poter uscire dal

processo che li genera.

Se invece si interviene nel punto di controllo di inoltro (o di transito), si avverte l'effetto solo nei pacchetti che devono attraversare il firewall da un'interfaccia a un'altra. È bene ribadire che questi possono transitare solo se la cosa viene abilitata attraverso il comando:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward [Invio]
```

oppure, per IPv6:

```
# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding [Invio]
```

Il comando seguente, per quanto inutile, impedisce il transito dei pacchetti tra le interfacce, attraverso la gestione del firewall, con la modifica della politica predefinita del punto di controllo relativo:

```
# iptables -P FORWARD DROP [Invio]
```

Prima di proseguire è bene rimettere a posto le politiche predefinite dei tre punti di controllo:

```
# iptables -P INPUT ACCEPT [Invio]

# iptables -P OUTPUT ACCEPT [Invio]

# iptables -P FORWARD ACCEPT [Invio]
```

42.5.2.2 Opzioni di contorno

Prima di affrontare l'analisi delle regole che possono essere inserite nei punti di controllo riferiti alla gestione del firewall, è meglio descrivere subito l'utilizzo di alcune opzioni di contorno che hanno un'importanza minore, oppure che si possono utilizzare indipen-

dentemente dal tipo di protocollo a cui si fa riferimento con una regola.

`-v` `--verbose`	Questa opzione si utilizza generalmente assieme all'opzione di comando '**`-L`**', allo scopo di rendere più dettagliata l'informazione che si ottiene.
`-n` `--numeric`	Quando IPTables viene usato per ottenere delle informazioni, con questa opzione si fa in modo che le informazioni numeriche non siano convertite in nomi (per esempio a proposito degli indirizzi IP e delle porte TCP o UDP).

`-p [!] {tcp|udp|icmp|all}` `--protocol [!] {tcp|udp|icmp|all}`	Stabilisce il tipo di protocollo della regola che viene definita. La parola chiave '**`all`**' rappresenta qualsiasi protocollo ed è l'impostazione predefinita se questo non viene specificato. Le parole chiave che identificano i protocolli possono essere espresse anche attraverso lettere maiuscole. Il punto esclamativo, se utilizzato, serve a fare riferimento a tutti i protocolli fuorché quello indicato.

`--source-port [!] ↩`
`↪{`***porta*** `|` ***intervallo_di_porte*** `}`
`--sport [!] ↩`
`↪{`***porta*** `|` ***intervallo_di_porte*** `}`
	Stabilisce la porta o le porte di ingresso coinvolte, nel caso dei protocolli TCP o UDP.

`--destination-port [!] {`***porta*** `|`
intervallo_di_porte `}`
`--dport [!] {`***porta*** `|` ***intervallo_di_porte*** `}`
	Stabilisce la porta o le porte di destinazione coinvolte, nel caso dei protocolli TCP o UDP.

`-i [!]` ***interfaccia*** `--in-interface [!]` ***interfaccia***	Indica il nome dell'interfaccia di rete attraverso la quale sono ricevuti i pacchetti della regola che si sta definendo. Quando questa opzione non viene usata, si intende fare riferimento implicitamente a qualunque interfaccia di rete. Non è necessario che l'interfaccia indicata esista già nel momento in cui si inserisce la regola. Inoltre, è possibile indicare un gruppo di interfacce, sostituendo il numero finale con il segno '**+**'. Per esempio, '**ppp+**' rappresenta tutte le interfacce '**ppp0**', '**ppp1**', ecc. Questo comportamento riguarda anche l'opzione '**-o**', riferita all'interfaccia di uscita.

`-o` [`!`] ***interfaccia*** `--out-interface` [`!`] ***interfaccia***	Indica il nome dell'interfaccia di rete attraverso la quale sono inviati i pacchetti della regola che si sta definendo. Quando questa opzione non viene usata, si intende fare riferimento implicitamente a qualunque interfaccia di rete.
`-j` ***obiettivo*** `--jump` ***obiettivo***	Questa opzione serve a definire l'obiettivo, attraverso una parola chiave tra quelle consuete, oppure il riferimento a un gruppo di regole creato a parte, oppure ancora permette di specificare un'estensione. Un'estensione è un obiettivo speciale che può essere utilizzato in base al contesto, oppure a seguito di una richiesta esplicita di caricamento di un modulo con l'opzione '**`-m`**'. Viene chiarito in seguito di cosa si tratta.

Segue la descrizione di alcuni esempi.

- `#` **`iptables -L INPUT -v`**[*Invio*]

Elenca le regole di ingresso in modo dettagliato.

- `# iptables -L OUTPUT -n` [*Invio*]

  Elenca le regole di uscita senza tradurre informazioni numeriche nei nomi corrispondenti.

- ```
  iptables -A punto_di_controllo regola -i eth0 -j DROP
  ```

  Lo schema mostra l'aggiunta in coda di una regola non meglio identificata, nella quale viene specificato in particolare che deve riferirsi al traffico entrante dall'interfaccia '**eth0**'. Per i pacchetti che vengono intercettati dalla regola, viene applicato l'obiettivo '**DROP**'.

- ```
  iptables -A punto_di_controllo -p tcp regola -i eth0 -j DROP
  ```

  Lo schema mostra l'aggiunta in coda di una regola non meglio identificata, nella quale viene specificato in particolare che deve riferirsi al traffico TCP entrante dall'interfaccia '**eth0**'. Per i pacchetti che vengono intercettati dalla regola, viene applicato l'obiettivo '**DROP**'.

- ```
  iptables -A punto_di_controllo -p ! tcp regola -i ! eth0 -j DROP
  ```

  Lo schema mostra l'aggiunta in coda di una regola non meglio identificata, nella quale viene specificato in particolare che deve riferirsi a tutto il traffico che non sia TCP, entrante da un'interfaccia qualunque purché non sia '**eth0**'. Per i pacchetti che vengono intercettati dalla regola, viene applicato l'obiettivo '**DROP**'.
  ```

42.5.2.3 Regole che non fanno riferimento a un protocollo

«

Le regole che non indicano un protocollo particolare possono servire esclusivamente a individuare il traffico riferito a un'origine e a una destinazione, con l'indicazione eventuale dell'interfaccia di ingresso e di uscita:

```
[-p all] [-s [!] origine] [-i interfaccia] ↩
↪ [-d [!] destinazione] [-o interfaccia]
```

Come si vede dallo schema, si possono utilizzare le opzioni '**-s**' e '**-d**' per indicare rispettivamente l'origine e la destinazione di un pacchetto. In aggiunta, si potrebbe inserire l'indicazione di una certa interfaccia attraverso cui i pacchetti vengono ricevuti o trasmessi; inoltre, volendo indicare espressamente che non si fa riferimento a un protocollo particolare, si può aggiungere l'opzione '**-p**' con l'argomento '**all**'.

La definizione di un gruppo di indirizzi IP può essere fatta attraverso l'indicazione di una coppia ***numero_ip*/*maschera***, con una barra obliqua di separazione tra i due. La maschera può essere indicata nel modo consueto, oppure con un numero che esprime la quantità di bit iniziali da porre al valore uno. A titolo di esempio, la tabella 42.51 mostra l'equivalenza tra alcune maschere di rete tipiche e questo numero di abbreviazione.

Tabella 42.51. Maschere di rete tipiche per IPv4.

Maschera di rete	Abbreviazione	Sottorete
255.0.0.0	8	Classe A
255.255.0.0	16	Classe B
255.255.255.0	24	Classe C
255.255.255.255	32	punto-punto

Quando si vuole fare riferimento a indirizzi imprecisati, si utilizza solitamente 0.0.0.0 che può essere indicato anche con un solo zero; questo si abbina di solito alla maschera nulla: 0.0.0.0/0 o 0/0. Tuttavia, per fare riferimento a qualunque indirizzo, è sufficiente omettere la sua indicazione, in pratica basta fare a meno di indicare l'opzione '**`-s`**' o '**`-d`**'.

L'indicazione di un indirizzo può essere fatta utilizzando direttamente il nome a dominio corrispondente, ma questo richiede la disponibilità di un servizio DNS; ciò può essere conveniente quando si tratta di un firewall connesso stabilmente con la rete esterna, altrimenti si creerebbero delle attese inutili e fastidiose, nel tentativo di risolvere dei nomi che non sono di competenza delle zone locali. Pertanto, in generale è preferibile indicare indirizzi in forma numerica.

Il punto esclamativo che può essere inserito facoltativamente di fronte all'indicazione di un indirizzo IP, o di un gruppo di indirizzi, rappresenta la negazione logica e serve a fare riferimento al gruppo di indirizzi complementare.

Tabella 42.52. Rappresentazione dell'origine e della destinazione.

Opzione	Descrizione
`-s` [`!`] ***indirizzo*** [`/`***maschera***] `--source` [`!`] ***indirizzo*** [`/`***maschera***]	Permette di definire l'origine dei pacchetti. L'indirizzo viene indicato generalmente in forma numerica, anche se c'è la possibilità di usare un nome a dominio. La maschera, eventuale, serve a indicare un gruppo di indirizzi. Se questo parametro viene omesso, si intende implicitamente '**`-s 0.0.0.0/0`**', ovvero '**`-s 0/0`**', che rappresenta tutti gli indirizzi possibili.

Opzione	Descrizione
`-d` [`!`] ***indirizzo*** [/***maschera***] `--destination` [`!`] ***indirizzo*** [/***maschera***]	Permette di definire la destinazione dei pacchetti. L'indirizzo viene indicato generalmente in forma numerica, anche se c'è la possibilità di usare un nome a dominio. La maschera, eventuale, serve a indicare un gruppo di indirizzi. Se questo parametro viene omesso, si intende implicitamente '**`-d 0.0.0.0/0`**', ovvero '**`-d 0/0`**', che rappresenta tutti gli indirizzi possibili.

Segue la descrizione di alcuni esempi.

- `#` **`iptables -A INPUT -s 192.168.100.0/24 -j DROP`** [*Invio*]

  Blocca tutto il traffico in ingresso, destinato all'elaboratore locale, proveniente dalla rete 192.168.100.∗.

- `#` **`iptables -A INPUT -s 192.168.100.0/24 -d 0/0 -j DROP`** [*Invio*]

  Esattamente come nell'esempio precedente.

- `#` **`iptables -A INPUT -s 192.168.100.0/24 -d 0/0`** ↩
  ↪ **`-i eth0 -j DROP`** [*Invio*]

  Come nell'esempio precedente, specificando però che questo traffico in ingresso deve provenire dall'interfaccia '**`eth0`**' (se

provenisse da un'altra interfaccia, non verrebbe intercettato da questa regola).

- # **iptables -A FORWARD -d 192.168.100.0/24 -j DROP** [*Invio*]

  Blocca tutto il traffico in transito destinato alla rete 192.168.100.∗.

- # **iptables -A FORWARD -s 0/0 -d 192.168.100.0/24** ↩
  ↪ **-j DROP** [*Invio*]

  Esattamente come nell'esempio precedente.

- # **iptables -A FORWARD -s 0/0 -d ! 192.168.100.0/24** ↩
  ↪ **-j DROP** [*Invio*]

  Blocca tutto il traffico in transito destinato a indirizzi diversi dalla rete 192.168.100.∗.

- # **iptables -A OUTPUT -d 192.168.100.0/24 -j DROP** [*Invio*]

  Blocca tutto il traffico in uscita, generato nell'elaboratore locale, destinato alla rete 192.168.100.∗.

42.5.2.4 Utilizzo pratico di regole elementari

«

Come negli esempi mostrati in precedenza, in cui si agiva soltanto sulla politica predefinita, con la stessa semplicità si può sperimentare l'uso delle regole. Per cominciare, quando il comando '**iptables -L**' genera il risultato

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

significa che non ci sono regole per alcun punto di controllo e le politiche predefinite non oppongono resistenza al transito dei pacchetti. Con una regola molto semplice è possibile bloccare qualunque ingresso attraverso l'interfaccia virtuale corrispondente a *localhost*, cioè all'indirizzo 127.0.0.1:

```
# iptables -A INPUT -s 127.0.0.1 -j DROP [Invio]
```

Se si tenta di fare il ping verso il nodo locale, questo non genera alcuna risposta, dal momento che tutti i pacchetti in ingresso vengono eliminati. Anticipando un po' quello che viene descritto in seguito, se lo scopo fosse esclusivamente quello di impedire l'ingresso dei pacchetti del protocollo ICMP (cosa che tra l'altro impedisce il ping), si potrebbe usare un comando più specifico:

```
# iptables -A INPUT -p icmp -s 127.0.0.1 -j DROP [Invio]
```

Se sono stati eseguiti gli esempi, il comando '**`iptables -L INPUT`**' dovrebbe generare il risultato seguente:

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  localhost            anywhere
DROP       icmp --  localhost            anywhere
```

Prima di fare altre considerazioni, conviene osservare la simbologia usata nel rapporto che è stato ottenuto: la colonna '**prot**' rappresenta il protocollo di riferimento; la colonna '**opt**' rappresenta delle specificazioni opzionali delle regole che in questo caso non sono mai state utilizzate; le colonna '**source**' e '**destination**' rappresentano l'origine e la destinazione dei pacchetti, dove in particolare la parola chiave '**anywhere**' esprime in pratica ciò che altrimenti si indicherebbe con la notazione 0.0.0.0/0. Si osservi la differenza nel risultato nel caso si utilizzi l'opzione '**-n**', ovvero il comando '**iptables -L INPUT -n**', allo scopo di eliminare le rappresentazioni simboliche degli indirizzi.

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  127.0.0.1           0.0.0.0/0
DROP       icmp --  127.0.0.1           0.0.0.0/0
```

Le regole hanno una sequenza precisa; avendo utilizzato sempre l'opzione di comando '**-A**', queste sono state aggiunte di seguito. Come si può intuire, la seconda regola è inutile, dal momento che i pacchetti che potrebbero riguardarla vengono già presi in considerazione da quella precedente che li blocca completamente per conto proprio.

Le regole possono essere eliminate in modo selettivo attraverso l'opzione di comando '**-D**', oppure in modo complessivo attraverso l'opzione '**-F**'. Per eliminare la prima regola, si potrebbe utilizzare uno dei due comandi seguenti:

```
# iptables -D INPUT -s 127.0.0.1 -j DROP[Invio]

# iptables -D INPUT 1[Invio]
```

Nel primo caso viene eliminata la prima regola che corrisponde al modello, cioè la prima in assoluto, mentre il secondo comando fa riferimento direttamente al numero della regola. Naturalmente, dopo l'eliminazione della prima regola, quella che inizialmente era la seconda diventa la prima:

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  localhost            anywhere
```

Come accennato, per eliminare tutte le regole di un punto di controllo si può usare l'opzione di comando '**-F**':

```
# iptables -F INPUT[Invio]
```

L'esempio elimina tutte le regole di ingresso.

Se l'elaboratore con il quale si fanno questi esperimenti ospita un servizio si può fare qualche esperimento più interessante. Supponendo di disporre di un servente HTTP che riceve richieste attraverso la porta 80 del protocollo TCP, si potrebbe impedirne l'accesso da parte dell'utente che accede dallo stesso sistema locale attraverso il comando seguente:

```
# iptables -A INPUT -p tcp -s 127.0.0.1 -d 127.0.0.1 ↩
↪          --dport 80 -j REJECT[Invio]
```

Quando si avvia un programma di navigazione per accedere al servizio HTTP locale, questo cerca di instaurare una connessione TCP utilizzando la porta 80 nella destinazione; se il firewall dispone della regola inserita con il comando appena mostrato, intercetta il tentativo di connessione e restituisce un messaggio di rifiuto attraverso il protocollo ICMP. La scelta di utilizzare l'obiettivo '**REJECT**' è motivata da questa esigenza: evitare di fare perdere tempo a chi tenta di

accedere, perché diversamente l'obiettivo '**DROP**' renderebbe la cosa più subdola. Si osservi cosa si ottiene con l'opzione '**-L**':

```
# iptables -L INPUT[Invio]

Chain INPUT (policy ACCEPT)
target     prot opt source               destination
REJECT     tcp  --  localhost            localhost           tcp ↩
↪dpt:www reject-with icmp-port-unreachable
```

La sigla '**dpt**' sta per *Destination port*; '**www**' è evidentemente il nome della porta 80. Dal momento che è stata richiesto l'obiettivo '**REJECT**', viene mostrato esplicitamente il tipo di messaggio ICMP che viene restituito a seguito di un tentativo di accesso: '**port-unreachable**'.

Per definire delle regole corrette per i fini che ci si prefigge, occorre conoscere bene il comportamento del protocollo che si utilizza. Tornando all'esempio appena fatto, in cui lo scopo è quello di impedire all'utente del sistema locale di accedere al servizio HTTP locale, si potrebbe ottenere un risultato equivalente agendo sul punto di controllo di uscita. Per farlo occorre sapere che la connessione TCP è simmetrica e che nel flusso di ritorno il servizio HTTP utilizza ancora la stessa porta 80, già impiegata per ricevere la richiesta di connessione.

```
# iptables -F INPUT[Invio]

# iptables -A OUTPUT -p tcp -s 127.0.0.1 --sport 80 ↩
↪          -d 127.0.0.1 -j REJECT[Invio]
```

In questo caso si deve osservare comunque una cosa: il messaggio ICMP, con cui si notifica il blocco del transito del pacchetto in uscita, è diretto all'applicazione che tenta di rispondere alla richiesta del

cliente, di conseguenza il cliente ne resta all'oscuro.

42.5.2.5 Regole per i protocolli TCP e UDP

Il modo con cui si possono definire le regole necessarie a individuare i pacchetti, dipendono dal tipo di protocollo utilizzato. Generalmente si è interessati maggiormente a controllare i protocolli TCP e UDP, che hanno in comune l'utilizzo delle porte.

Dovendo fare riferimento a un protocollo TCP o UDP si utilizza l'opzione '**`-p`**', seguita dalla parola chiave '**`tcp`**' o '**`udp`**'. Dal momento che i protocolli TCP e UDP utilizzano le porte, l'origine e la destinazione possono includere questa informazione, con l'uso delle opzioni '**`--sport`**' e '**`--dport`**' rispettivamente.

Le porte possono essere indicate in modo preciso (una soltanto), oppure attraverso un intervallo. Queste porte possono essere espresse attraverso un nome, come definito nel file '`/etc/services`', oppure per numero, cosa che di solito si preferisce per evitare ambiguità o malintesi. Gli intervalli di porte, in particolare, vengono espressi nella forma seguente:

porta_iniziale **:** ***porta_finale***

Se si indica un intervallo, cosa che si determina per la presenza dei due punti, se manca l'indicazione della porta iniziale si intende in modo predefinito la numero zero, se invece manca quella finale si intende la porta 65535. Come nel caso degli indirizzi IP, l'indicazione della porta o dell'intervallo di queste può essere preceduta dal punto esclamativo in qualità di negazione logica.

Tabella 42.58. Opzioni per i protocolli TCP e UDP.

Opzione	Descrizione
`-s` [`!`] *indirizzo* [`/`*maschera*] ↩ ↪ [`!`] [`--sport` *porta* | *intervallo_di_porte*] `--source` [`!`] *indirizzo* [`/`*maschera*] ↩ ↪ [`!`] [`--source-port` ↩ ↪ *porta* | *intervallo_di_porte*] `-d` [`!`] *indirizzo* [`/`*maschera*] ↩ ↪ [`!`] [`--dport` *porta* | *intervallo_di_porte*] `--destination` [`!`] ↩ ↪ *indirizzo* [`/`*maschera*] ↩ ↪ [`!`] [`--destination-port` ↩ ↪ *porta* | *intervallo_di_porte*]	Con i protocolli TCP e UDP, l'origine e la destinazione possono includere l'indicazione delle porte.

Nel caso di protocolli TCP, è possibile analizzare i bit che qualificano lo stato della connessione. Questi bit hanno un nome simbolico, corrispondente a: '**`SYN`**', '**`ACK`**', '**`FIN`**', '**`RST`**', '**`URG`**' e '**`PSH`**'. Si può controllare lo stato di questi bit con l'opzione '**`--tcp-flags`**'. Dal momento che è comune la richiesta di individuare i pacchetti con il bit '**`SYN`**' attivo e i bit '**`RST`**' e '**`ACK`**' disattivati, si può usare per questo l'opzione '**`--syn`**'.

Tabella 42.59. Opzioni per i protocolli TCP.

Opzione	Descrizione
`--tcp-flags` ***elenco_bit_da_considerare*** ***elenco_bit_attivi***	Gli elenchi in questione si ottengono indicando i nomi dei bit separati da una virgola, senza l'aggiunta di spazi, dove in particolare, la parola chiave '**`ALL`**' fa riferimento a tutti i bit gestibili. Per esempio, '**`--tcp-flags ALL SYN,ACK`**' indica la richiesta di individuare i pacchetti TCP in cui solo i bit '**`SYN`**' e '**`ACK`**' sono attivi simultaneamente (mentre tutti gli altri sono disattivati). La stessa cosa si potrebbe esprimere in modo esteso come: '**`--tcp-flags SYN,ACK,FIN,RST,URG,PSH SYN,ACK`**'.

Opzione	Descrizione
`--syn`	Corrisponde in pratica a '**`--tcp-flags SYN,RST,ACK SYN`**'. Questi pacchetti vengono usati nel protocollo TCP per richiedere l'inizializzazione della connessione. In pratica, bloccando questi pacchetti si impedisce l'instaurarsi di una connessione TCP in un solo verso.

Segue la descrizione di alcuni esempi.

- # **`iptables -A INPUT -p tcp -s ! 192.168.0.0/16 ↩`**
  `↪ -d 192.168.0.0/16 --dport 80 -j REJECT` [*Invio*]

  Impedisce l'accesso ai servizi HTTP (protocollo TCP, porta 80) della rete 192.168.∗.∗ a tutti gli indirizzi estranei alla rete stessa.

- # **`iptables -A INPUT -p tcp -s ! 192.168.0.0/16 ↩`**
  `↪ -d 192.168.0.0/16 --dport 80 --syn -j REJECT` [*Invio*]

  Come nell'esempio precedente, limitandosi a intervenire nei pacchetti di inizializzazione delle connessioni.

42.5.2.6 Regole per il protocollo ICMP

«

Il protocollo ICMP è molto importante per il controllo del funzionamento della rete, in questo senso è rara la possibilità che sia il caso di bloccarne il transito attraverso il firewall. Tuttavia, dal momento

che i fini del firewall non si limitano al blocco del traffico, è comunque importante poter indicare una regola che sappia selezionare un tipo particolare di pacchetto ICMP. La tabella 42.27 elenca i tipi di pacchetto ICMP e il loro utilizzo.

Per indicare una regola che faccia riferimento a un tipo particolare di pacchetto ICMP, si sfruttano le opzioni che servono a specificare l'origine o la destinazione, aggiungendo il numero o il nome del tipo ICMP (il numero può essere composto da una seconda parte, denominato ***codice***). In pratica, questa informazione va a sostituire il numero di porta nel caso dei protocolli TCP e UDP.

È estremamente importante che non vengano bloccati i messaggi ICMP di tipo 3.

Il protocollo ICMP è differente tra IPv4 e IPv6, pertanto la sigla usata per farvi riferimento cambia.

Il comando '**`iptables -p icmp -h`**' genera l'elenco di tutti i messaggi ICMP gestibili con IPv4:

```
# iptables -p icmp -h[Invio]

  Valid ICMP Types:
  echo-reply (pong)
  destination-unreachable
     network-unreachable
     host-unreachable
     protocol-unreachable
     port-unreachable
     fragmentation-needed
```

```
    source-route-failed
    network-unknown
    host-unknown
    network-prohibited
    host-prohibited
    TOS-network-unreachable
    TOS-host-unreachable
    communication-prohibited
    host-precedence-violation
    precedence-cutoff
source-quench
redirect
    network-redirect
    host-redirect
    TOS-network-redirect
    TOS-host-redirect
echo-request (ping)
router-advertisement
router-solicitation
time-exceeded (ttl-exceeded)
    ttl-zero-during-transit
    ttl-zero-during-reassembly
parameter-problem
    ip-header-bad
    required-option-missing
timestamp-request
timestamp-reply
address-mask-request
address-mask-reply
```

Si può osservare che i nomi rientrati, fanno riferimento a un tipo ICMP formato anche attraverso l'indicazione di un codice. Per esempio, '**`network-unreachable`**' corrisponde a '**`3/0`**'.

Tabella 42.61. Opzioni per i protocolli ICMP.

Opzione	Descrizione
`-s` [`!`] ***indirizzo*** [`/`***maschera***] ↩ ↪ [`!`] [`--icmp-type` ***tipo*** [`/`***codice***]] `--source` [`!`] ***indirizzo*** [`/`***maschera***] ↩ ↪ [`!`] [`--icmp-type` ***tipo*** [`/`***codice***]] `-d` [`!`] ***indirizzo*** [`/`***maschera***] ↩ ↪ [`!`] [`--icmp-type` ***tipo*** [`/`***codice***]] `--destination` [`!`] ↩ ↪ ***indirizzo*** [`/`***maschera***] ↩ ↪ [`!`] [`--icmp-type` ↩ ↪ ***tipo*** [`/`***codice***]]	Come già accennato, con il protocollo ICMP l'origine e la destinazione possono includere l'indicazione del tipo di messaggio ICMP.

Segue la descrizione di alcuni esempi.

- `#` **`iptables -A INPUT -p icmp -s ! 192.168.0.0/16`** ↩
  ↪ **`--icmp-type 8 -d 192.168.0.0/16 -j DROP`** [*Invio*]

  Blocca e ignora i pacchetti ICMPv4 che contengono un messaggio di tipo 8, cioè '**`echo-request`**', proveniente da un indirizzo estraneo alla rete 192.168.∗.∗ e destinato alla rete stessa.

- `#` **`iptables -A INPUT -p icmp -s ! 192.168.0.0/16`** ↩
  ↪ **`--icmp-type echo-request`** ↩
  ↪ **`-d 192.168.0.0/16 -j DROP`** [*Invio*]

  Esattamente come nell'esempio precedente, indicando per nome il tipo ICMPv4.

- `# ip6tables -A INPUT -p icmpv6 -s ! fec0::/16 ↩`
  `↪ --icmpv6-type echo-request ↩`
  `↪ -d fec0::/16 -j DROP` [*Invio*]

  Blocca e ignora i pacchetti ICMPv6 che contengono un messaggio di tipo '**`echo-request`**', proveniente da un indirizzo estraneo alla rete fec0:∗ e destinato alla rete stessa.

42.5.2.7 Pacchetti frammentati

«

I pacchetti frammentati costituiscono un problema per la gestione del firewall. In generale ci si limita a intervenire sul primo frammento, perché questo dovrebbe contenere le informazioni necessarie a identificarlo correttamente.

Se il firewall rappresenta un passaggio obbligato per il traffico che lo attraversa, è molto importante che sia abilitata la ricomposizione dei pacchetti frammentati. Questo risolve tanti problemi e soprattutto quello del controllo dei frammenti.

Per identificare un frammento di pacchetto successivo al primo, si utilizza l'opzione '**`-f`**' nel modo seguente:

```
[!] -f | [!] --fragment
```

Il punto esclamativo permette di ottenere l'effetto contrario, cioè di fare riferimento a tutti i pacchetti che non sono frammenti. Utilizzando questa opzione non è possibile indicare delle porte TCP o UDP, né specificare il tipo di messaggio per il protocollo ICMP.

L'esempio seguente blocca l'attraversamento di frammenti dei pacchetti ICMP provenienti da un indirizzo estraneo alla rete 192.168.∗.∗ e destinati alla rete stessa.

```
# iptables -A FORWARD -p icmp -s ! 192.168.0.0/16 ↩
↪          -d 192.168.0.0/16 -f -j DROP [Invio]
```

42.5.3 Estensioni particolari

Le funzionalità di filtro del kernel sono suddivise in segmenti differenti che possono essere incluse o meno, in fase di compilazione, oppure possono essere caricate attraverso moduli esterni. Queste funzionalità particolari sono definite ***moduli***, senza per questo voler confondere il concetto con i moduli del kernel. Per utilizzare queste funzionalità si deve indicare prima il modulo, attraverso l'opzione '**-m**':

```
-m modulo
```

```
--match modulo
```

Nel seguito vengono presentati solo alcuni dei moduli disponibili.

È molto probabile che tali estensioni non siano tutte disponibili per IPv6; ma di questo ci si accorge facilmente dalle segnalazioni di errore generate da '**ip6tables**'.

42.5.3.1 Limiti

«

È possibile definire una regola che scatti fino al raggiungimento di un certo limite per un certo tipo di pacchetto. Si tratta del modulo '**limit**':

```
-m limit
```

Si distinguono due informazioni in questo contesto: la quantità di pacchetti per unità di tempo e il margine di sicurezza prima che venga preso in considerazione il raggiungimento del limite.

Tabella 42.62. Opzioni relative al modulo '**limit**'.

Opzione	Descrizione
`-m limit --limit` ***n*** [`/` ***unità_di_tempo***]	Questa opzione serve a definire la quantità di pacchetti (***n***) entro la quale scatta la regola. Se non si indica l'unità di tempo si fa riferimento implicitamente a secondi. A ogni modo, si possono usare le parole chiave seguenti, con il significato intuitivo che hanno: '**`second`**', '**`minute`**', '**`hour`**', '**`day`**'. È importante osservare che si possono usare anche solo le iniziali di questi termini. Per esempio, '**`--limit 10`**' rappresenta un limite di 10 pacchetti per secondo, cosa che si può esprimere come '**`--limit 10/second`**', oppure anche '**`--limit 10/s`**'.

Opzione	Descrizione
`-m limit --limit-burst` ***n***	Questa opzione, '**`--limit-burst`**', serve a creare un margine iniziale ulteriore, dopo il quale inizia il conteggio del limite stabilito con l'opzione '**`--limit`**'. Se non si specifica questa opzione, il margine è di 5.

Vengono riproposti gli esempi che appaiono già nel *Linux 2.4 packet filtering HOWTO* di Rusty Russell. Ovviamente, perché questi limiti abbiano un senso, dopo le regole che consentono il transito entro una certa frequenza, occorre aggiungere delle regole che blocchino lo stesso tipo di pacchetti, senza più l'indicazione di un limite.

- Protezione contro un attacco da inondazione di pacchetti «SYN»:

```
# iptables -A FORWARD -p tcp --syn -m limit ↩
↪          --limit 1/s -j ACCEPT [Invio]
```

  Consente il transito di un solo pacchetto di inizializzazione delle connessioni TCP al secondo. Per bloccare i pacchetti successivi si aggiunge il blocco degli stessi pacchetti:

```
# iptables -A FORWARD -p tcp --syn -j DROP [Invio]
```

- Protezione contro un tentativo di scansione delle porte TCP:

```
# iptables -A FORWARD -p tcp ↩
↪          --tcp-flags SYN,ACK,FIN,RST RST -m limit ↩
↪          --limit 1/s -j ACCEPT [Invio]
```

Consente il transito di un pacchetto TCP al secondo con il solo bit '**RST**' attivo, nell'ambito del gruppo di bit composto da '**SYN**', '**ACK**', '**FIN**' e '**RST**'. Per bloccare i pacchetti successivi si aggiunge il blocco degli stessi pacchetti:

```
# iptables -A FORWARD -p tcp ↩
↪          --tcp-flags SYN,ACK,FIN,RST RST -j DROP [Invio]
```

- Protezione contro un'inondazione di richieste di eco ICMP (ping):

```
# iptables -A FORWARD -p icmp --icmp-type echo-request ↩
↪          -m limit --limit 1/s -j ACCEPT [Invio]
```

Consente il transito di un pacchetto ICMP di tipo 8 (richiesta di eco) al secondo. Per bloccare i pacchetti successivi si aggiunge il blocco degli stessi pacchetti:

```
# iptables -A FORWARD -p icmp --icmp-type echo-request ↩
↪          -j DROP [Invio]
```

Gli esempi mostrano tutti un controllo applicato ai pacchetti in transito. Per proteggere anche il firewall occorre intervenire nello stesso modo sui pacchetti in ingresso.

42.5.3.2 Stato delle connessioni

Un modulo speciale, denominato '**state**', consente di analizzare le connessioni e di individuarle in base a uno status semplice da definire.

```
-m state
```

Questo modulo consente semplicemente di utilizzare l'opzione '**`--state`**', con cui si specifica lo stato di una connessione:

```
--state {NEW|ESTABLISHED|RELATED|INVALID}[,…]
```

Le varie parole chiave utilizzate per definire lo stato di una connessione hanno il significato descritto nell'elenco seguente.

Tabella 42.63. Opzioni relative al modulo '**`state`**'.

Opzione	Descrizione
`-m state --state NEW[,…]`	Si tratta di un pacchetto che crea una nuova connessione.
`-m state --state ESTABLISHED[,…]`	Si tratta di un pacchetto che appartiene a una connessione già esistente.
`-m state --state RELATED[,…]`	Si tratta di un pacchetto correlato a un'altra connessione. Per esempio, potrebbe trattarsi di un messaggio ICMP di errore, oppure di una connessione TCP generata automaticamente da una connessione FTP precedente.
`-m state --state INVALID[,…]`	Si tratta di un pacchetto che non può essere qualificato per qualche ragione e come tale viene considerato non valido.

Segue la descrizione di alcuni esempi.

- `# iptables -A FORWARD -d 192.168.0.0/16 -m state --state ESTABLISHED,RELATED -j ACCEPT` [*Invio*]

  Consente il transito verso gli indirizzi 192.168.∗.∗ quando si tratta di connessioni già realizzate o di pacchetti correlati a connessioni preesistenti.

- `# iptables -A FORWARD -d 192.168.0.0/16 -m state --state INVALID -j DROP` [*Invio*]

  Elimina i pacchetti destinati agli indirizzi 192.168.∗.∗ quando questi non sono identificabili in qualche modo, nel senso che non sembrano avere motivo di esistere.

- `# iptables -A FORWARD -m state --state NEW -i ! ppp0 -j ACCEPT` [*Invio*]

  Consente l'instaurarsi di una connessione che attraversi il nodo, purché ciò non avvenga a cominciare da un pacchetto che entri dall'interfaccia '**ppp0**' (PPP).

42.5.4 Strategie

In generale, quando si predispone uno script con tutte le regole di firewall che si vogliono applicare ai pacchetti in ingresso, in uscita e in transito, si inizia dall'azzeramento di quelle eventualmente esistenti, esattamente nel modo seguente:

```
#!/bin/sh

/sbin/iptables -F
…
```

Dal momento che le funzionalità di filtro del kernel Linux non devono interferire con quelle di instradamento (*routing*), nel caso le prime

non siano state definite, è necessario che la politica predefinita sia sempre '**ACCEPT**'. In generale, se si vuole configurare il proprio elaboratore come firewall la situazione cambia e dovrebbe essere conveniente il contrario, in modo da poter controllare la situazione. In pratica, ancora prima dell'azzeramento delle regole delle varie categorie, è solitamente opportuno modificare le politiche predefinite, in modo da bloccare gli accessi e il transito dei pacchetti.

```
/sbin/iptables -P INPUT DROP
/sbin/iptables -P OUTPUT DROP
/sbin/iptables -P FORWARD DROP
```

La definizione delle regole di firewall deve tenere conto dell'ordine in cui appaiono nell'elenco gestito all'interno del kernel, quindi, la scelta tra le opzioni di comando '**-A**' (aggiunta in coda) e '**-I**' (inserimento all'inizio o in un'altra posizione) deve essere fatta in modo consapevole. A seconda della propria filosofia personale, si potrebbe scegliere di utilizzare sempre solo lo stesso tipo.

Se si sceglie di «aggiungere» le regole, dovrebbe essere conveniente iniziare da quelle di eliminazione o rifiuto ('**DROP**' o '**REJECT**'), per finire con quelle di accettazione ('**ACCEPT**').

Se si preferisce lasciare che la politica predefinita sia '**ACCEPT**', è importante ricordare di aggiungere una regola che impedisca l'accesso in modo generalizzato alla fine di tutte le regole di un punto di controllo, come mostrato nell'esempio seguente:

```
# In coda a tutte le regole
/sbin/iptables -A INPUT -j DROP
/sbin/iptables -A OUTPUT -j DROP
/sbin/iptables -A FORWARD -j DROP
```

Nell'esempio, non avendo fatto riferimento ad alcun protocollo, né

ad alcun indirizzo sorgente o di destinazione, si intendono implicitamente tutti i tipi di pacchetto. Questo tipo di strategia è comunque applicabile con qualunque tipo di politica predefinita, dal momento che con questa regola si catturano tutti i pacchetti rimanenti.

Quando lo scopo di un firewall è solo quello di proteggere una rete interna da quella esterna, si potrebbe pensare che l'uso di regole per il solo attraversamento dovrebbe bastare. In effetti, dal momento che i pacchetti devono attraversare il firewall per raggiungere la rete interna, il ragionamento è corretto; tuttavia, bisogna pensare anche a proteggere il firewall e in tal senso si comprende l'utilità di disporre di un punto di controllo in ingresso. Infatti, se un aggressore riesce a ottenere accesso nel firewall, da lì può entrare nella rete interna che invece si considera protetta. Il punto di controllo in uscita è una possibilità in più per completare le cose ed è un bene che ci siano tante possibilità.

Naturalmente, le funzionalità di filtro dei pacchetti sono utili anche per gli elaboratori che devono difendersi da soli, perché si trovano in un ambiente ostile, o perché semplicemente non ci si può fidare. È evidente in questi casi che diventa importantissima la possibilità di intervenire nelle regole del punto di controllo di ingresso ed eventualmente anche in quelle del punto di controllo in uscita, mentre il controllo dell'attraversamento dovrebbe risultare semplicemente inutile.

42.5.4.1 UDP e DNS

Una delle politiche normali nella configurazione di un firewall che deve proteggere una rete interna è quella di non lasciare che i pacchetti del protocollo UDP possano attraversarlo. In linea di principio

questo atteggiamento è ragionevole, dal momento che con il protocollo UDP si gestiscono spesso informazioni delicate e aggredibili con facilità (NFS e NIS sono gli esempi più importanti).

```
# iptables -A FORWARD -p udp -j DROP [Invio]
```

Quello che si vede è il comando molto semplice che permette di ottenere questo risultato, intervenendo necessariamente in fase di attraversamento.

Il sistema DNS utilizza prevalentemente il protocollo UDP e a volte il protocollo TCP. In questo senso, un servizio DNS collocato all'interno di una rete protetta che abbia bisogno di risolvere nomi della rete esterna, deve necessariamente avvalersi di un altro servizio DNS posto nel firewall o anche al di fuori di questo.

```
options {
    forwarders {
        123.123.123.123;
    };
};
```

L'esempio che si vede rappresenta una parte del file '`/etc/named.conf`' (o '`/etc/bind/named.conf`') dove si indica l'indirizzo 123.123.123.123 da utilizzare per inoltrare le richieste che non possono essere risolte in base alla definizione delle zone locali. La comunicazione con il servizio presso 123.123.123.123 avviene con il protocollo TCP, permettendo di superare il problema del blocco al transito dei pacchetti UDP.

Il fatto che il sistema DNS utilizzi a volte il protocollo TCP per le comunicazioni normali deve servire a capire che un blocco del protocollo UDP può creare problemi intermittenti alla risoluzione dei nomi e degli indirizzi IP.

42.5.4.2 Contraffazione dell'origine: IP spoof

Uno dei riferimenti importanti su cui si basa il controllo da parte del firewall è l'indirizzo di origine dei pacchetti. Spesso, chi attacca un sistema altera i pacchetti che invia modificando l'origine, per non essere individuato. Il firewall non è in grado di sapere se l'origine è veritiera o contraffatta.

Per risolvere questo problema con IPv4 si utilizza la gestione dell'instradamento attraverso la procedura denominata «Source Address Verification». Per prima cosa ci si deve accertare che esista il file virtuale '`/proc/sys/net/ipv4/conf/all/rp_filter`', quindi si possono sovrascrivere tutti i file '`/proc/sys/net/ipv4/conf/*/rp_filter`' con il valore uno. In pratica:

```
if [ -e /proc/sys/net/ipv4/conf/all/rp_filter ]
then
    for f in /proc/sys/net/ipv4/conf/*/rp_filter
    do
        echo 1 > $f
    done
fi
```

In modo più grossolano è possibile eliminare i pacchetti che sono «evidentemente» contraffatti. Per esempio, se l'interfaccia di rete '**ppp0**' è quella che si rivolge verso la rete esterna, si possono bloccare tranquillamente i pacchetti che provengono da questa con l'in-

dicazione di un'origine appartenente a uno degli indirizzi riservati per le reti private.

```
/sbin/iptables -A INPUT   -s 127.0.0.0/8     -i ! lo -j DROP
/sbin/iptables -A FORWARD -s 127.0.0.0/8     -i ! lo -j DROP
/sbin/iptables -A INPUT   -s 192.168.0.0/16 -i ppp0 -j DROP
/sbin/iptables -A FORWARD -s 192.168.0.0/16 -i ppp0 -j DROP
/sbin/iptables -A INPUT   -s 172.16.0.0/12  -i ppp0 -j DROP
/sbin/iptables -A FORWARD -s 172.16.0.0/12  -i ppp0 -j DROP
/sbin/iptables -A INPUT   -s 10.0.0.0/8     -i ppp0 -j DROP
/sbin/iptables -A FORWARD -s 10.0.0.0/8     -i ppp0 -j DROP
```

Nel fare questo, tuttavia, bisogna tenere in considerazione che a volte, alcuni fornitori di accesso a Internet utilizzano degli indirizzi riservati alle reti private per le connessioni PPP; generalmente si tratta del gruppo 10.*.*.*.

42.5.4.3 Esempi

Di seguito vengono mostrati altri esempi che dovrebbero aiutare a comprendere ancora meglio il funzionamento di un firewall realizzato con un sistema GNU/Linux.

- `/sbin/iptables -A FORWARD -s 224.0.0.0/3 -d 0/0 -j DROP`

  Questa regola impedisce il transito di tutti quei pacchetti che provengono da un'origine in cui l'indirizzo IP sia composto in modo da avere i primi tre bit a uno. Infatti, 224_{10} si traduce nel numero binario 11100000_2, che esclude tutta la classe D e la classe E degli indirizzi IPv4. Segue la visualizzazione della regola attraverso '**`iptables -L FORWARD -n`**'.

```
target     prot opt source               destination
DROP       all  --  224.0.0.0/3          0.0.0.0/0
```

- ```
  /sbin/iptables -A FORWARD -s 224.0.0.0/3 -j DROP
  ```

  Questo esempio è esattamente identico a quello precedente, perché la destinazione predefinita è proprio quella riferita a qualunque indirizzo.

- ```
  /sbin/iptables -A FORWARD -p tcp -s 192.168.1.0/24 -d 0/0 23 -j ACCEPT
  ```

  Consente ai pacchetti TCP provenienti dalla rete 192.168.1.∗ di attraversare il firewall per raggiungere qualunque indirizzo, ma solo alla porta 23. In pratica concede di raggiungere un servizio TELNET. Segue la visualizzazione della regola attraverso '**`iptables -L FORWARD -n`**'.

```
target  prot opt source           destination
ACCEPT  tcp  --  192.168.1.0/24   0.0.0.0/0       tcp dpt:23
```

- ```
  /sbin/iptables -A FORWARD -p tcp -s 0/0 --sport 6000:6009 ↩
  ↪              -d 0/0 -j DROP
  /sbin/iptables -A FORWARD -p tcp -s 0/0 -d 0/0 ↩
  ↪              --dport 6000:6009 -j DROP
  ```

  Blocca il transito delle comunicazioni riferite alla gestione remota di applicazioni per X. In questo caso, si presume di poter avere a che fare con sistemi che gestiscono fino a 10 serventi grafici contemporaneamente.

- ```
  /sbin/iptables -A INPUT  -p tcp -s 0/0 --sport 6000:6009 ↩
  ↪              -d 0/0 -j DROP
  /sbin/iptables -A OUTPUT -p tcp -s 0/0 -d 0/0 ↩
  ↪              --dport 6000:6009 -j DROP
  ```

  Blocca l'ingresso e l'uscita di comunicazioni riferite alla gestione remota di applicazioni per X. Questo potrebbe essere utile

per proteggere un sistema che non si avvale di un firewall o che semplicemente non si fida della rete circostante.

```
/sbin/iptables -A INPUT -m state ↩
↪              --state ESTABLISHED,RELATED -j ACCEPT
/sbin/iptables -A INPUT -m state --state NEW ↩
↪              -i ! ppp0 -j ACCEPT
/sbin/iptables -A INPUT -j DROP
/sbin/iptables -A FORWARD -m state ↩
↪              --state ESTABLISHED,RELATED -j ACCEPT
/sbin/iptables -A FORWARD -m state --state NEW ↩
↪              -i ! ppp0 -j ACCEPT
/sbin/iptables -A FORWARD -j DROP
```

- Si consente l'ingresso e il transito di pacchetti relativi a connessioni già esistenti e di pacchetti correlati a connessioni già esistenti; si consente l'instaurazione di connessioni nuove, purché non provengano dall'interfaccia '**ppp0**'; si bloccano tutti gli altri pacchetti.

42.5.5 Contabilizzazione del traffico

«

Con i kernel Linux 2.4.∗ e 2.6.∗, la contabilizzazione del traffico è implicita nel sistema di filtro del firewall: ogni regola che venga inserita in un punto di controllo accumula i propri contatori. In questo senso possono essere opportune anche regole che non hanno l'indicazione di alcun obiettivo, in quanto utili solo per selezionare una parte del traffico ai fini contabili.

Con l'opzione '**-v**' si può osservare il valore raggiunto dai vari contatori. Per esempio, disponendo di un'unica regola che cattura tutto il traffico in ingresso,

```
# iptables -F INPUT [Invio]

# iptables -A INPUT [Invio]
```

il comando

```
# iptables -L INPUT -v -n[Invio]
```

potrebbe generare un rapporto simile a quello seguente:

```
Chain INPUT (policy ACCEPT 57716 packets, 4848K bytes)
 pkts bytes target  prot opt in  out  source     destination
57716 4848K        all  --  *   *    0.0.0.0/0  0.0.0.0/0
```

Si possono notare in particolare le colonne '**pkts**' e '**bytes**' che si riferiscono rispettivamente al numero di pacchetti IP e alla loro dimensione complessiva in byte. A fianco dei numeri che esprimono queste quantità potrebbero essere aggiunte delle lettere che rappresentano dei multipli: '**K**', '**M**' e '**G**'. È importante osservare che questi esprimono multipli del sistema di numerazione decimale: 1000, 1000000 e 1000000000.[5]

L'azzeramento dei conteggi si ottiene con l'opzione di comando '**-Z**' ('**--zero**') che interviene in tutte le regole dei punti di controllo indicati. Questa può essere utilizzata anche assieme all'opzione '**-L**', in modo da non perdere informazioni.

Segue la descrizione di alcuni esempi.

- `# iptables -L INPUT -v -n[Invio]`

  Mostra tutte le informazioni disponibili sulle regole di ingresso, senza tradurre i dati numerici in nome. Tra le altre cose mostra anche i contatori del traffico.

- `# iptables -Z INPUT[Invio]`

  Azzera i conteggi riferiti alle regole di ingresso.

- `# iptables -L -Z -v -n[Invio]`

Mostra tutte le informazioni disponibili di tutti i punti di controllo (ed eventualmente anche di altri raggruppamenti di regole), compresi i conteggi che vengono azzerati immediatamente dopo.

42.5.6 Registrazione del traffico

Esiste un obiettivo speciale, denominato '**LOG**', con il quale si ottiene l'annotazione nel registro del sistema sull'intestazione del pacchetto, ogni volta che la regola ne intercetta uno. Tuttavia, in questo caso, quando un pacchetto viene intercettato da una regola del genere, questo continua poi a essere analizzato dalle regole successive, per poterlo utilizzare anche in modo differente.

```
/sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
/sbin/iptables -A INPUT -m state --state NEW -i ! ppp0       -j ACCEPT
/sbin/iptables -A INPUT -m state --state NEW -i ppp0         -j LOG
/sbin/iptables -A INPUT -j DROP
```

L'esempio che si vede è abbastanza articolato, per farne comprendere il senso. Lo scopo è quello di annotare nel registro le connessioni in ingresso, attraverso l'interfaccia '**ppp0**', che non siano autorizzabili a seguito di qualche correlazione con connessioni preesistenti.

La registrazione può avvenire anche indicando una sigla come prefisso, attraverso l'opzione '**--log-prefix**', per distinguere facilmente le annotazioni. L'esempio seguente ripete quanto già mostrato in precedenza, con l'aggiunta del prefisso '**XXX**' iniziale:

```
/sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
/sbin/iptables -A INPUT -m state --state NEW -i ! ppp0       -j ACCEPT
/sbin/iptables -A INPUT -m state --state NEW -i ppp0         -j LOG ↩
↪--log-prefix "XXX"
/sbin/iptables -A INPUT -j DROP
```

Per controllare le segnalazioni che si ottengono in questo modo nel registro del sistema, si può fare riferimento alla voce '**kern.info**'. Per esempio, se nel file '`/etc/syslog.conf`' si inserisce la direttiva seguente, si ottiene una copia di questi messaggi nella console '`/dev/tty11`':

```
kern.info                       /dev/tty11
```

Si osservi che in condizioni normali, tutti i messaggi di tipo '***.info**' vengono inviati anche alla console attiva, contribuendo a disturbare il lavoro che lì vi viene svolto.

42.5.7 Raggruppamenti di regole al di fuori dei punti di controllo standard

Oltre ai punti di controllo normali, è possibile definire delle raccolte di regole aggiuntive, a cui si può fare riferimento quasi come se fossero delle subroutine di un linguaggio di programmazione. Queste raccolte vengono identificate da un nome, al quale si può fare riferimento attraverso altre regole in qualità di obiettivo. In pratica, una regola posta in un punto di controllo può indicare un obiettivo corrispondente al nome di un altro raggruppamento di regole, che viene così a essere incorporato idealmente in quella posizione.

Per comprendere il meccanismo, si supponga di avere creato la raccolta di regole (*chain*) denominata '**prova**', con una regola all'interno del punto di controllo di ingresso che vi faccia riferimento. Per cominciare, le regole contenute all'interno di '**prova**' potrebbero essere:

```
target     prot opt source               destination
           all  --  192.168.1.0/24       0.0.0.0/0
           all  --  0.0.0.0/0            192.168.1.0/24
           all  --  127.0.0.1            0.0.0.0/0
```

Come si può osservare in questo caso, si tratta di regole che servono solo alla contabilizzazione del traffico, dal momento che non sono stati indicati degli obiettivi.

Le regole di ingresso potrebbero essere quelle seguenti:

```
target     prot opt source               destination
…
prova      tcp  --  0.0.0.0/0            0.0.0.0/0
…
```

Si può osservare una regola il cui scopo è quello di individuare tutto il traffico TCP. Dal momento che l'obiettivo di questa è il raggruppamento '**prova**', i pacchetti che rientrano nella selezione di questa regola vengono scomposti ulteriormente attraverso le regole del raggruppamento '**prova**'. I pacchetti che non vengono «catturati» da alcuna regola del raggruppamento '**prova**' tornano a essere presi in considerazione dalle regole successive nel punto di controllo di ingresso.

La creazione di un raggruppamento di regole si ottiene con l'opzione di comando '**-N**' ('**--new-chain**') e la sua eliminazione con '**-X**' ('**--delete-chain**'). Per esempio, il comando

```
# iptables -N prova [Invio]
```

serve a creare il raggruppamento '**prova**' a cui si accennava in precedenza. L'inserimento di regole avviene nel modo normale; per continuare a seguire gli esempi fatti, i comandi dovrebbero essere

i seguenti:

```
# iptables -A prova -s 192.168.1.0/24 [Invio]

# iptables -A prova -d 192.168.1.0/24 [Invio]

# iptables -A prova -s 127.0.0.1 [Invio]
```

Così, l'inserimento della regola nel punto di controllo di ingresso che fa riferimento a questo raggruppamento, come mostrato dagli esempi in precedenza, si indica semplicemente con il comando seguente:

```
# iptables -A INPUT -p tcp -j prova [Invio]
```

L'eliminazione di un raggruppamento di regole è ammissibile solo quando questo è vuoto e quando non esistono più riferimenti da parte di altre regole nei punti di controllo normali.

```
# iptables -D INPUT -p tcp -j prova [Invio]

# iptables -F prova [Invio]

# iptables -X prova [Invio]
```

I comandi mostrati sopra servono rispettivamente a eliminare la regola di ingresso che faceva riferimento al raggruppamento '**prova**', a svuotare il raggruppamento e infine a eliminarlo.

42.6 NAT/PAT con kernel Linux

I kernel Linux 2.4.* e 2.6.*, assieme alla gestione del filtro dei pacchetti IP, possono occuparsi anche della trasformazione degli indirizzi e delle porte, ovvero del NAT/PAT. Ciò consente, tra le al-

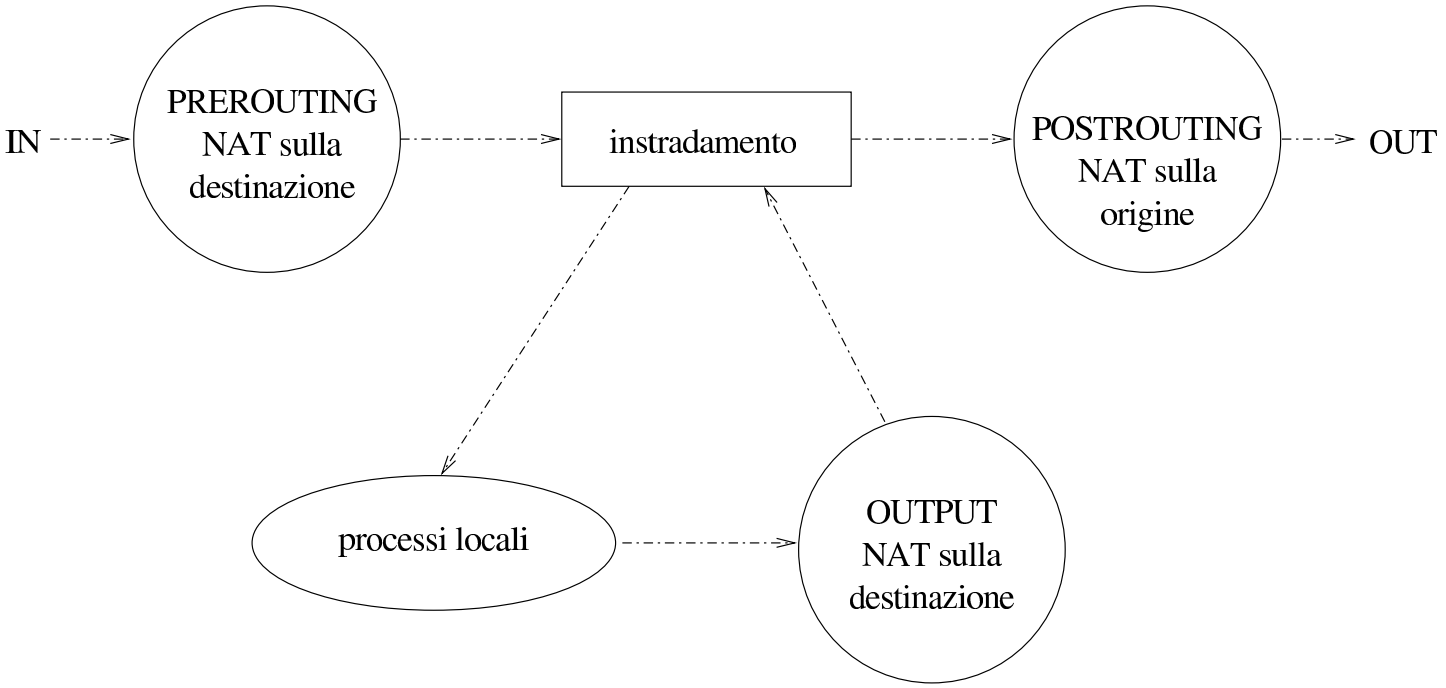
tre cose, di ottenere il mascheramento IP e la gestione del proxy trasparente.

Va però tenuto conto che queste funzionalità sono disponibili generalmente per i protocolli IPv4, ma non per IPv6.

42.6.1 Struttura e punti di intervento

La gestione NAT/PAT può essere applicata in tre punti, denominati '**PREROUTING**', '**POSTROUTING**' e '**OUTPUT**'.

Figura 42.84. Punti di intervento per la gestione del NAT/PAT e influenza relativa.



Il '**PREROUTING**' si riferisce a una posizione ideale che precede l'instradamento da parte dell'elaboratore. In questa posizione è possibile modificare gli indirizzi di destinazione, in modo che l'instradamento possa avvenire correttamente in base a tali trasformazioni.

Il '**POSTROUTING**' si riferisce a una posizione ideale successiva all'instradamento da parte dell'elaboratore. In questa posizione è

possibile modificare gli indirizzi di origine.

Il punto denominato '**OUTPUT**' si riferisce ai pacchetti generati da un processo locale. Questi vengono vagliati successivamente anche dal punto '**POSTROUTING**'; a ogni modo si può gestire solo la trasformazione degli indirizzi di destinazione.

42.6.2 Gestione con IPTables

La configurazione della trasformazione degli indirizzi avviene per mezzo di IPTables, intervenendo nella tabella '**nat**':

```
iptables -t nat opzione_di_comando punto_di_intervento regola ↩
↪ obiettivo_di_trasformazione
```

Le opzioni di comando sono le stesse che si utilizzano per la gestione del filtro dei pacchetti IP. Anche in questo caso è prevista la presenza di una politica predefinita, dove la parola chiave '**ACCEPT**' serve a specificare l'assenza di trasformazioni. In condizioni normali, la tabella risulta vuota, come si vede nell'esempio seguente:

```
# iptables -t nat -L[Invio]

Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Sono disponibili le opzioni che identificano il protocollo, gli indirizzi, le porte e le interfacce di rete, come già avviene nell'utilizzo di

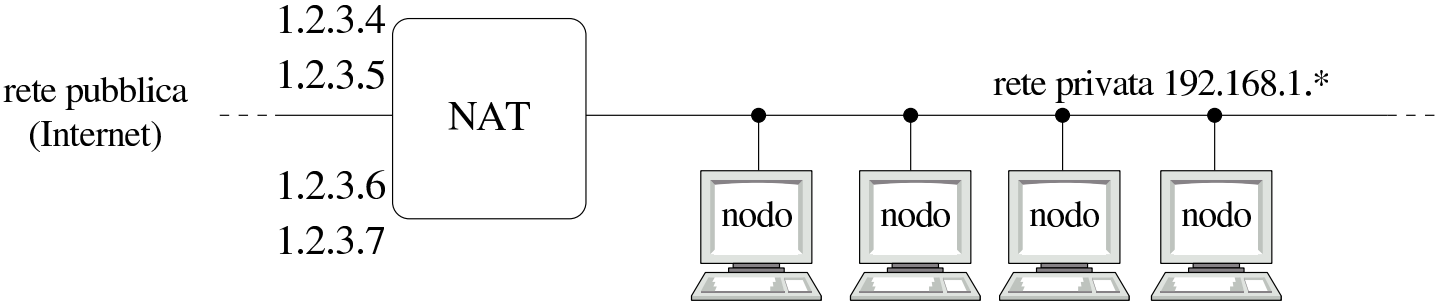
IPTables per la gestione del filtro IP.

42.6.3 Modifica dell'origine

«

Per comprendere il significato della trasformazione degli indirizzi di origine, conviene fare riferimento a un esempio, come si vede nella figura 42.86. In questo caso, il NAT si trova collegato a una rete privata, in cui si usano indirizzi 192.168.1.∗, mentre dalla parte connessa alla rete esterna, dispone di quattro indirizzi validi: 1.2.3.4, 1.2.3.5, 1.2.3.6, 1.2.3.7. Per consentire i collegamenti che partono dalla rete interna a quella esterna, il NAT deve sostituire gli indirizzi di origine utilizzando convenientemente i quattro indirizzi di cui dispone. Naturalmente, i quattro indirizzi in questione corrispondono tutti alla stessa interfaccia ed esistono gli instradamenti necessari dalla rete esterna a questi indirizzi.

Figura 42.86. Modifica degli indirizzi di origine.



Per raggiungere questo risultato, si può utilizzare il comando seguente, supponendo che '**eth0**' sia l'interfaccia a cui fanno riferimento i quattro indirizzi IP validi per la rete esterna:

```
# iptables -t nat -A POSTROUTING -o eth0 -j SNAT ↩
↪          --to-source 1.2.3.4-1.2.3.7 [Invio]

# iptables -t nat -L POSTROUTING [Invio]
```

```
Chain POSTROUTING (policy ACCEPT)
target     prot opt source     destination
SNAT       all  --  anywhere   anywhere     to:1.2.3.4-1.2.3.7
```

Come si può osservare, per ottenere la trasformazione degli indirizzi di origine viene utilizzato l'obiettivo di trasformazione '**SNAT**', il quale implica l'uso di un'opzione aggiuntiva:

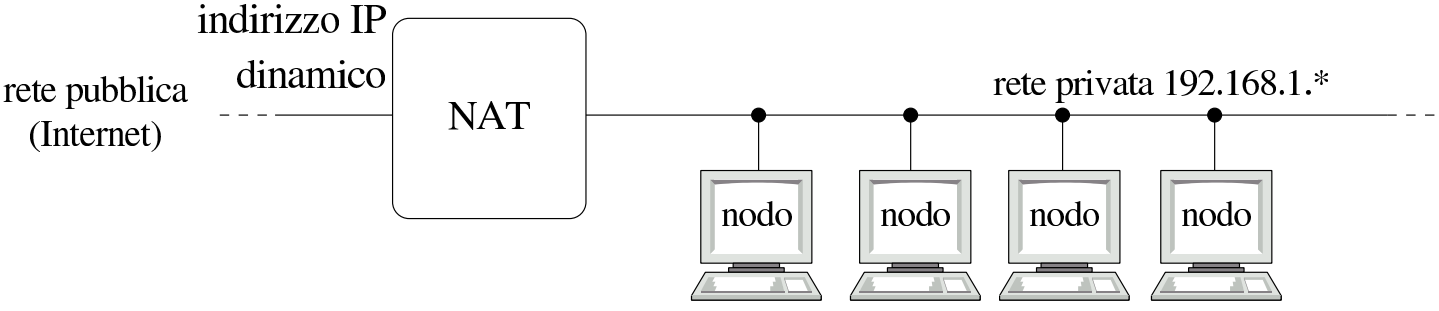
```
--to-source indirizzo_ip [-indirizzo_finale] [:porta_iniziale-porta_finale]
```

```
--to indirizzo_ip [-indirizzo_finale] [:porta_iniziale-porta_finale]
```

Come si intende dal modello sintattico, è possibile aggiungere l'indicazione di un intervallo di porte da utilizzare per la trasformazione. In generale, non mettendo questa informazione, la trasformazione delle porte avviene in modo corretto.

Questo tipo di trasformazione precisa degli indirizzi di origine si presta per le situazioni in cui l'interfaccia di rete collegata alla rete esterna ha uno o più indirizzi IP statici da poter mostrare. In alternativa, quando si può disporre soltanto di un indirizzo dinamico, come avviene nelle connessioni PPP comuni, conviene usare l'obiettivo '**MASQUERADE**'.

Figura 42.88. Mascheramento IP.



Seguendo l'esempio della figura 42.88, supponendo che l'interfaccia di rete collegata all'esterno sia '**ppp0**', si procede nel modo seguente:

```
# iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE [Invio]

# iptables -t nat -L POSTROUTING [Invio]

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE all  --  anywhere             anywhere
```

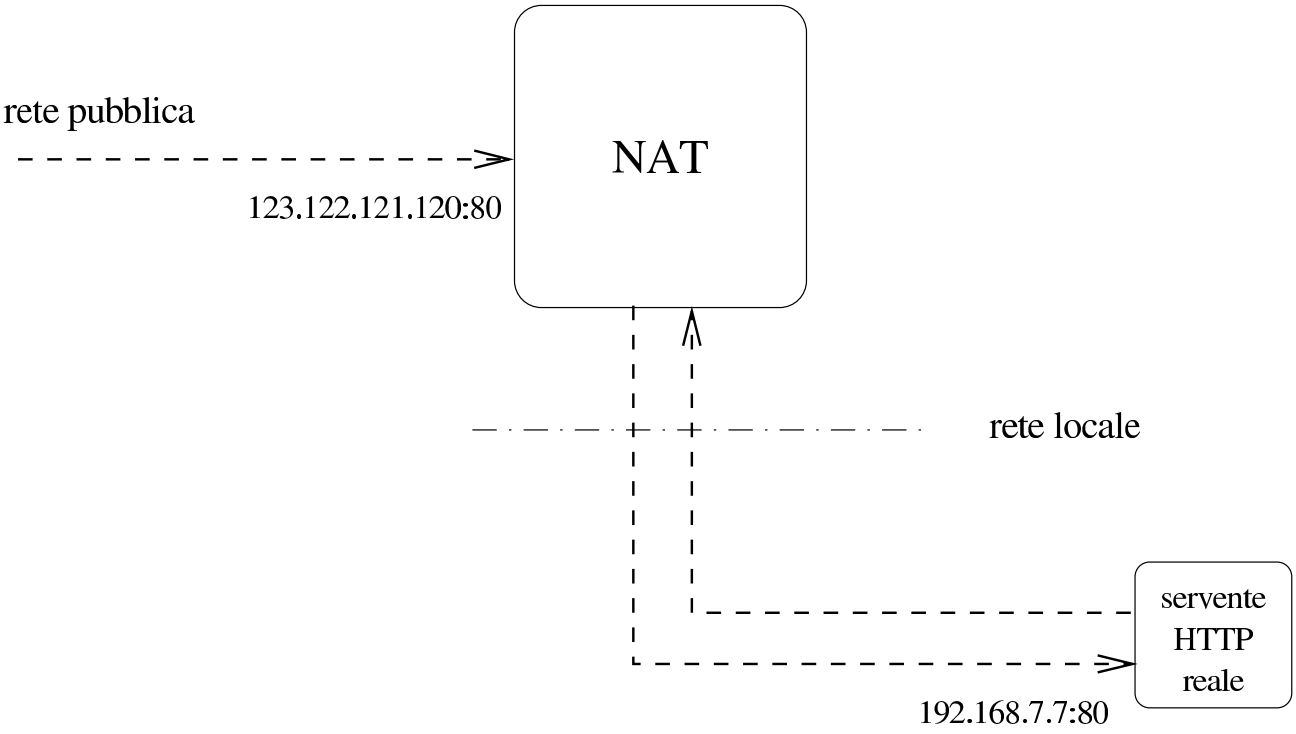
Si intende che la sostituzione dell'origine si gioca su un indirizzo IP unico, gestendo convenientemente le porte TCP e UDP. Pertanto, l'indirizzo in questione è implicitamente quello di cui dispone l'interfaccia di rete, che così può essere dinamico.

42.6.4 Modifica della destinazione

«

La modifica della destinazione si definisce con l'obiettivo '**DNAT**', che può intervenire nel punto '**PREROUTING**', oppure nei pacchetti generati localmente. Questo tipo di sostituzione serve per dirottare i pacchetti, per qualche motivo.

Figura 42.90. Il NAT/PAT trasferisce le connessioni dirette a 123.122.121.120:80 a 192.168.7.7:80.



La figura 42.90 mostra una situazione in cui viene collocato un servente HTTP in una rete locale con indirizzi privati, mentre si vuole fare in modo che all'esterno appaia collocato all'interno del router che svolge il ruolo di NAT. Per realizzare in pratica questa cosa, si può usare il comando seguente:

```
# iptables -t nat -A PREROUTING -p tcp --dport 80 -i eth0 ↩
↪          -j DNAT --to-destination 192.168.7.7 [Invio]

# iptables -t nat -L PREROUTING [Invio]

Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
DNAT       tcp  --  anywhere             anywhere        tcp ↩
↪dpt:www to:192.168.1.7
```

Come si può vedere dall'esempio, l'obiettivo di trasformazione '**DNAT**' implica l'uso di un'opzione aggiuntiva:

`--to-destination` ***indirizzo_ip*** [**-*indirizzo_finale***] [**:*porta_iniziale*-*porta_finale***]

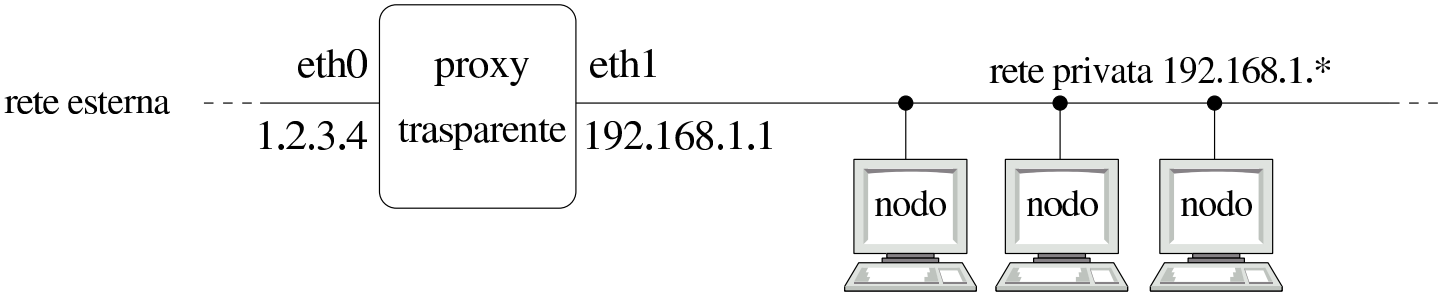
`--to` ***indirizzo_ip*** [**-*indirizzo_finale***] [**:*porta_iniziale*-*porta_finale***]

Come si intende dal modello sintattico, è possibile aggiungere l'indicazione di un intervallo di porte da utilizzare per la trasformazione. In generale, non mettendo questa informazione, la trasformazione delle porte avviene in modo corretto.

Nelle situazioni più comuni, modificando la destinazione si indica un solo indirizzo ed eventualmente una sola porta.

Un'altra situazione tipica è quella rappresentata dall'esigenza di ridirigere il traffico diretto a una certa porta, verso una porta differente di un certo nodo, nel quale esiste probabilmente un cache proxy (che ovviamente deve essere configurato correttamente per gestire tale situazione).

Figura 42.92. Realizzazione di un proxy trasparente per una rete locale.



Supponendo di gestire una rete locale simile a quella che si vede nella figura 42.92, si vuole fare in modo che tutte le richieste di accesso a servizi HTTP, da parte della rete locale, siano dirottati verso

il proxy, collocato nello stesso elaboratore che ospita il NAT, alla porta 8080 (si parla in questo caso di proxy trasparente).

```
# iptables -t nat -A PREROUTING -p tcp --dport 80 -i eth1 ↩
↪          -j DNAT --to-destination 192.168.1.1:8080 [Invio]
```

In questo caso particolare, dal momento che si vuole intervenire nello stesso elaboratore che ospita sia il NAT, sia il servizio proxy, è possibile utilizzare l'obiettivo speciale '**REDIRECT**' che richiede l'indicazione dell'opzione '**--to-port**':

```
--to-port porta
```

```
--to porta
```

L'esempio precedente potrebbe quindi essere semplificato nel modo seguente:

```
# iptables -t nat -A PREROUTING -p tcp --dport 80 -i eth1 ↩
↪          -j REDIRECT --to-port 8080 [Invio]

# iptables -t nat -L PREROUTING [Invio]

Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
REDIRECT   tcp  --  anywhere             anywhere            tcp ↩
↪dpt:www redir ports 8080
```

Il cambiamento della destinazione per quanto riguarda i pacchetti generati dalle applicazioni locali (interne al NAT), funziona nello stesso modo, ma è meno probabile la necessità di intervenire in questo modo.

L'allestimento di un proxy trasparente non si esaurisce con la ridirezione del traffico verso la porta del proxy; quasi sempre è necessario occuparsi anche della configurazione appropriata di questo.

Altri programmi affini.

netstat-nat(1)[6]	Si tratta di un programma simile a '**`netstat`**', con lo scopo di visualizzare le connessioni modificate da un kernel Linux per le funzionalità NAT.

42.7 Annotazioni sull'uso di un router ADSL per le utenze comuni

«

L'accesso a una linea ADSL (*Asymmetric digital subscriber line*) implica l'utilizzo di un «modem ADSL», oppure di un router ADSL. In generale, le opzioni proposte dai fornitori per le utenze private tendono a offrire l'uso di modem ADSL, pronti per l'utilizzo con sistemi operativi proprietari, mentre ci possono essere delle difficoltà nell'utilizzo di questi componenti se si dispone solo di software libero. Se nel contratto che viene sottoscritto non ci sono clausole che impediscono espressamente l'utilizzo di un router, a patto di assumersi comunque tutte le responsabilità per l'utilizzo del proprio accesso, vale forse la pena di acquistare un router ADSL, semplificando così molte cose.

42.7.1 Protocolli di comunicazione

Il modem o il router ADSL deve interagire con la controparte presso il fornitore di accesso attraverso un protocollo. Questo protocollo di comunicazione serve inizialmente per l'identificazione dell'utente che accede alla rete e poi per ottenere l'indirizzo IPv4, salvo il caso in cui questo sia stabilito dal contratto (indirizzo statico) e quindi già noto. Esistono due protocolli: *PPP over ethernet* e *PPP over ATM*. Questi protocolli vengono spesso abbreviati con nomi del tipo '**PPPoE**' e '**PPPoA**' rispettivamente.

Se si decide di acquistare un router ADSL, per utilizzarlo con software libero, cioè generalmente al di fuori di qualunque supporto possibile da parte del fornitore di accesso, bisogna essere sicuri, nella fase di sottoscrizione del contratto, di scegliere il protocollo «giusto».

In generale, la scelta che dovrebbe offrire più possibilità a un utilizzatore di software libero dovrebbe essere quella del protocollo *PPP over ethernet*, dal momento che con questo è possibile, teoricamente, utilizzare anche un qualunque modem ADSL (si tratta però di una procedura che qui non viene descritta, ma è disponibile molta documentazione al riguardo). Tuttavia, è bene acquistare un router ADSL che possa essere configurato per gestire indifferentemente entrambi i protocolli.

Ogni fornitore di accesso ha la propria politica nel modo di presentare l'offerta al pubblico; in questo senso, l'esigenza di semplificare al massimo la terminologia può rendere difficile a un utente più preparato il significato di certi termini. Per esempio, può capitare di dover scegliere la tipologia di collegamento usando come riferimento so-

lo la caratteristica esteriore di un modem che in quel contesto viene proposto: se il modem è di tipo *ethernet*, vuole dire che si fa riferimento a un protocollo *PPP over ethernet*, mentre altre tipologie sono riferite probabilmente al protocollo *PPP over ATM*.

42.7.2 Comunicazione e configurazione con il router ADSL

«

Normalmente, un router ADSL è un piccolo elaboratore senza tastiera e senza schermo, a cui si accede tramite un terminale seriale (attraverso una porta seriale standard), oppure attraverso un piccolo servente HTTP munito di un programma CGI adeguato.

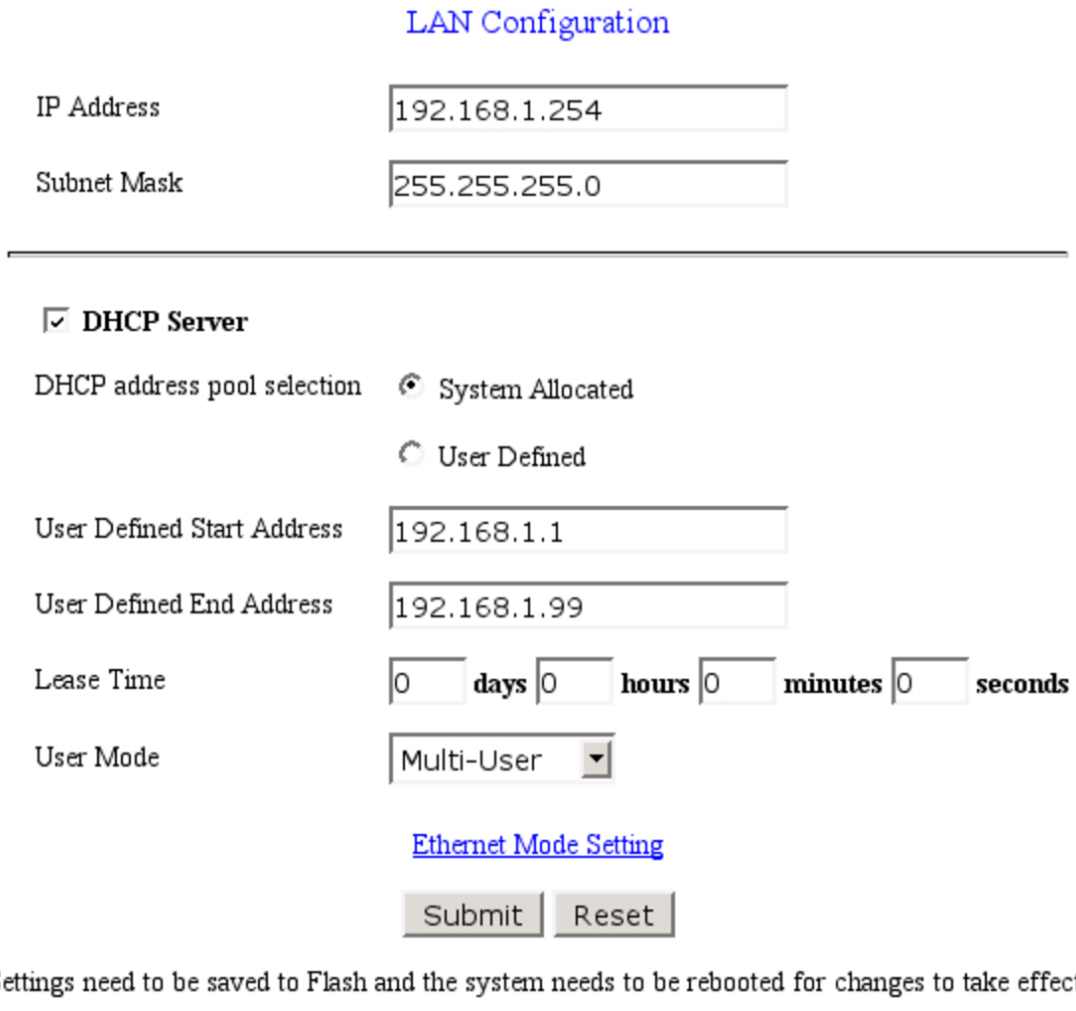
L'accesso è controllato normalmente attraverso una parola d'ordine e potrebbero essere previste due utenze: una amministrativa e una comune, dove la seconda consente la consultazione dello stato di funzionamento.

È bene iniziare a configurare il router ADSL prima di collegarlo alla linea esterna, per definire una parola d'ordine di accesso all'amministrazione differente da quella predefinita e per organizzare la rete locale. Di norma il router dovrebbe essere già impostato con un indirizzo IPv4 privato, associato all'interfaccia rivolta verso la rete interna (LAN); bisogna leggere la documentazione per determinare questo indirizzo e la sua maschera di rete; quindi, coerentemente con questi dati si configura il proprio elaboratore per accedere al router. Per qualche motivo, capita spesso che questo indirizzo sia in classe A, per esempio 10.0.0.2, con maschera di rete 255.0.0.0; di conseguenza, si deve configurare l'interfaccia di rete del proprio elaboratore in modo da poter comunicare con questo, per esempio con l'indirizzo 10.0.0.3, impostando anche l'instradamento predefinito verso il router, cioè verso l'indirizzo 10.0.0.2; quindi, con un navi-

gatore comune si dovrebbe accedere al servente HTTP del router: *http://10.0.0.2*.

Dopo l'autenticazione, con un po' di prudenza si può passare alla modifica della parola d'ordine per l'amministratore e probabilmente anche alla definizione di una rete interna con indirizzi più «ragionevoli».

Figura 42.95. Un esempio di pagina di configurazione della rete interna con indirizzi 192.168.1.∗, dove vengono riservati alcuni di questi per l'assegnazione automatica tramite protocollo DHCP.



Una volta risolto questo, occorre controllare di avere attivato la gestione del NAT, ovvero della traduzione degli indirizzi IPv4 della

rete interna nell'indirizzo valido ottenuto dal router. Probabilmente occorre verificare di utilizzare il tipo corretto di NAT, che in questo caso deve intervenire modificando anche le porte dei protocolli TCP e UDP.

Figura 42.96. Un esempio di pagina di attivazione del NAT. In questo caso è sufficiente selezionare il tipo NATP.

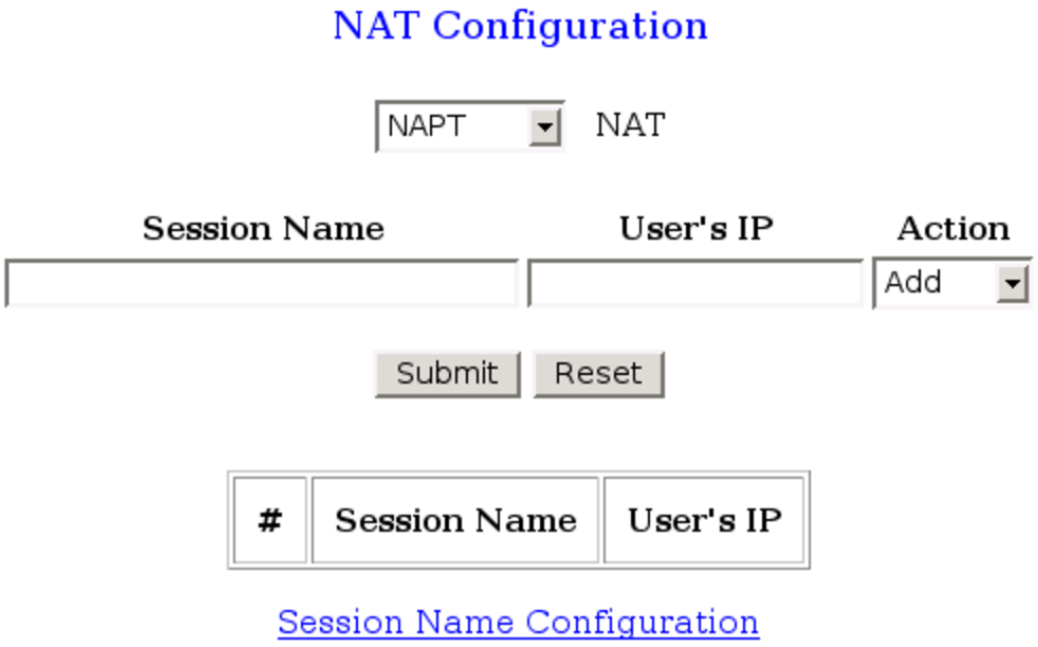
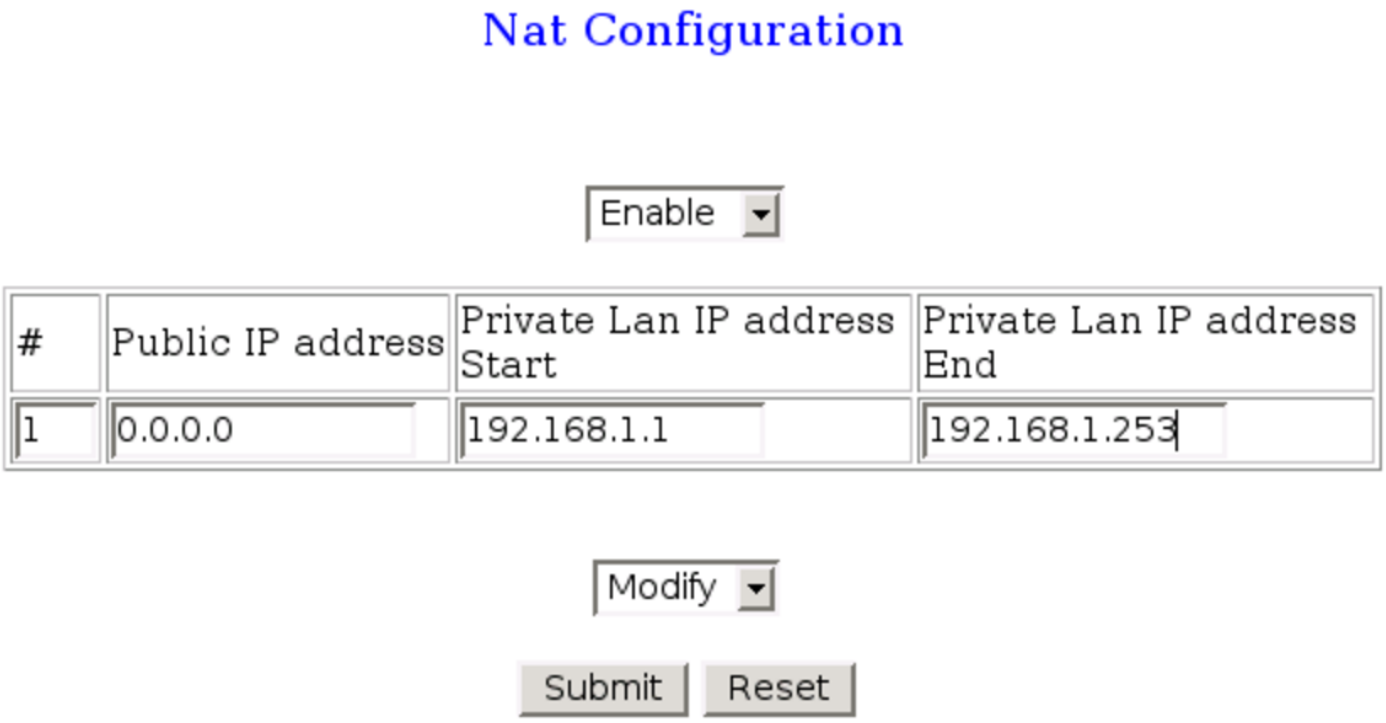


Figura 42.97. Un esempio in cui occorre specificare espressamente l'intervallo di indirizzi a cui applicare il NAT.



Quando è accertato che il collegamento della rete locale funziona correttamente, secondo le impostazioni definite, si può passare alla configurazione del lato esterno (WAN). È qui che si deve definire il protocollo di comunicazione. La figura 42.98 dà un'idea di questa configurazione per quanto riguarda *PPP over ethernet*. Si osservi che il nominativo utente e la parola d'ordine sono riferiti all'utenza presso il fornitore di accesso alla linea ADSL.

Figura 42.98. La pagina di configurazione del collegamento ADSL, con il protocollo *PPP over ethernet*, utilizzando un router CNet.

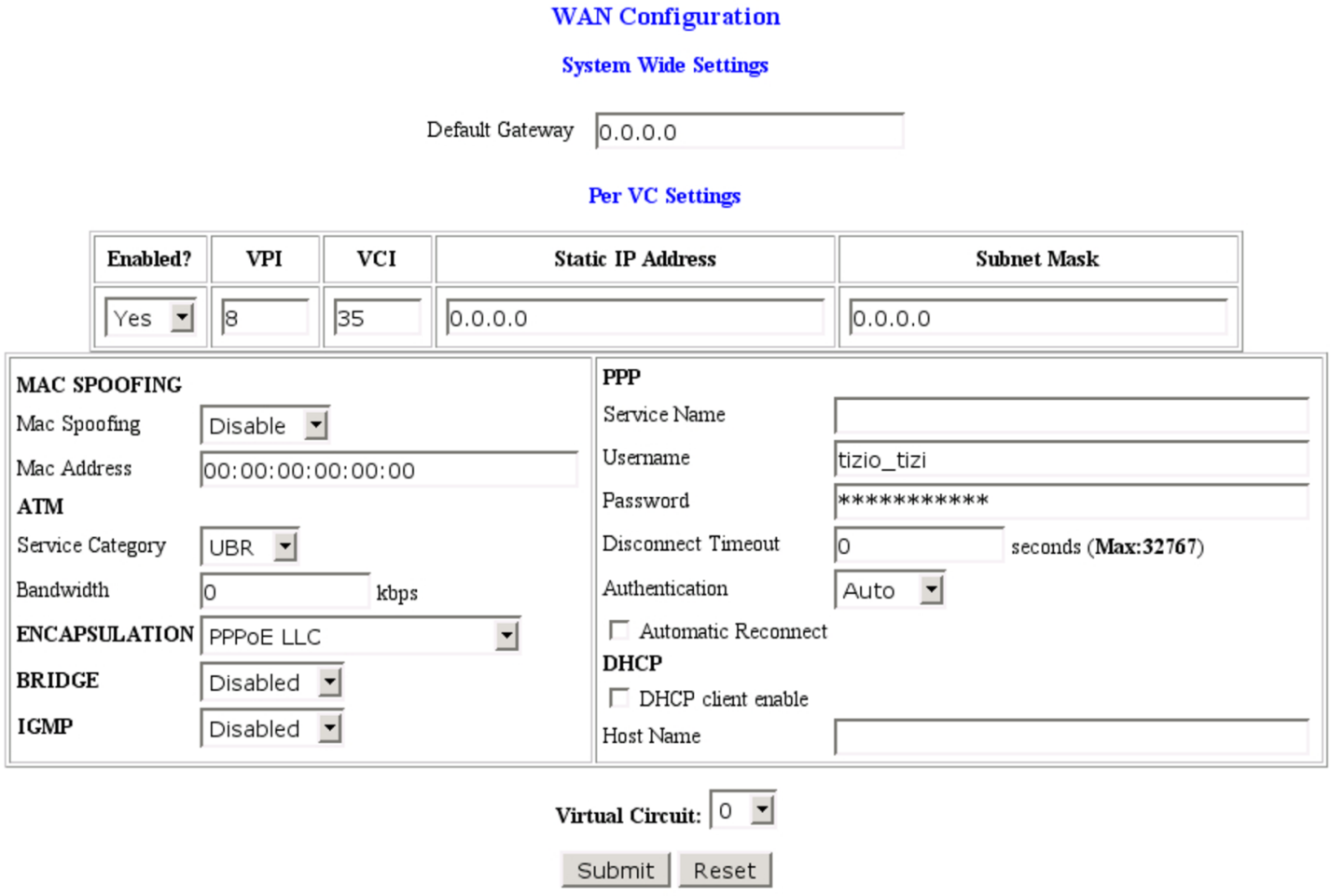
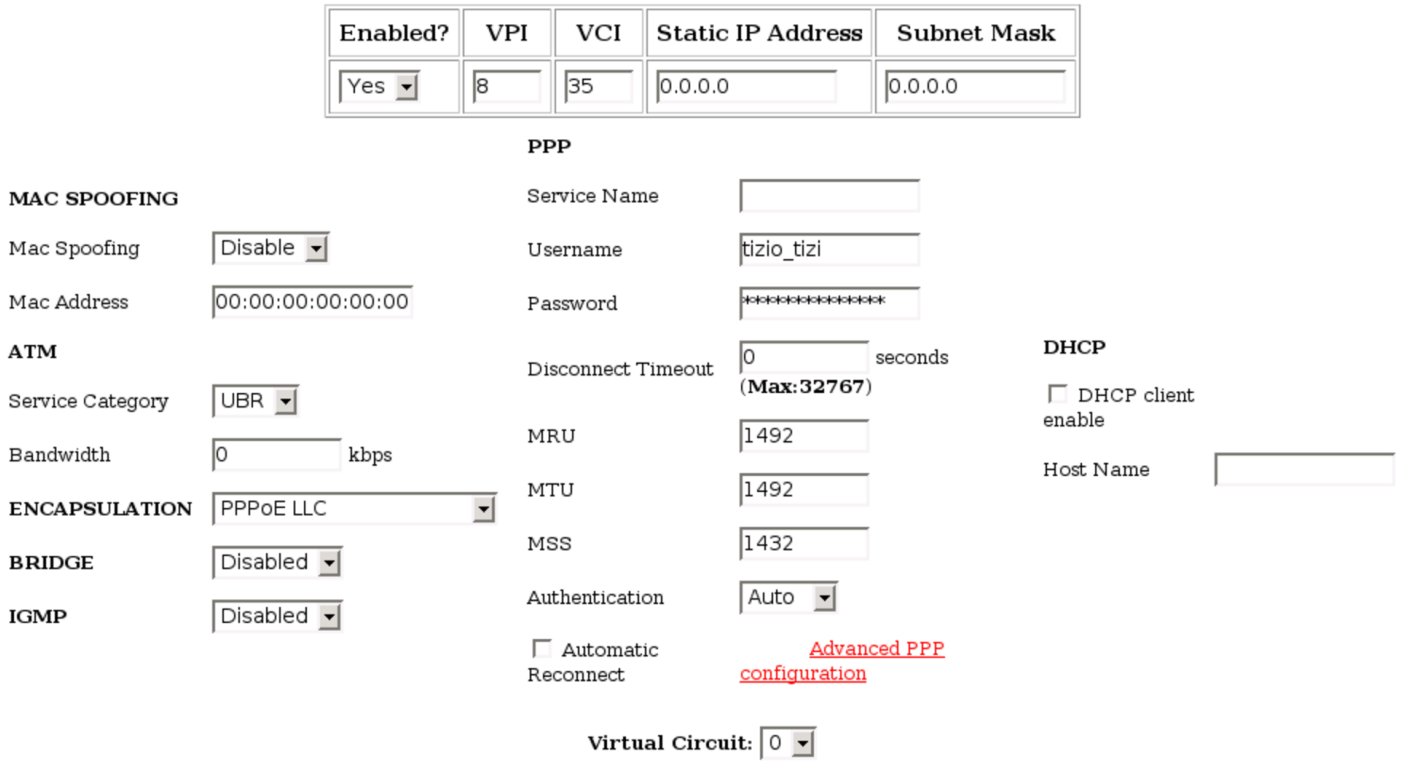


Figura 42.99. La pagina di configurazione del collegamento ADSL, con il protocollo *PPP over ethernet*, utilizzando un router Pirelli.



In questa fase è importante anche definire due parametri: VPI e VCI. Nelle reti italiane, solitamente, sono corretti i valori 8 e 35 rispettivamente.

42.7.3 Controllo

La fase successiva è quella del controllo di cosa accade collegando il router alla linea esterna. Dovrebbero essere disponibili della pagine che mostrano lo stato della connessione; se è presente una specie di registro (*log*) è questo il modo migliore per comprendere ciò che accade:

```
1/1/1970 0:0:0> Ethernet Device 0 Detected
1/1/1970 0:0:0> ATM: Detected
1/1/1970 0:0:0> ATM: Setting up vcc0, VPI=8, VCI=35
1/1/1970 0:0:0> NAPT is enabled
1/1/1970 0:0:0> Initialized NAPT.
1/1/1970 0:0:11> ATM Connected
1/1/1970 0:0:11> ATM layer is up, cell delineation achieved
1/1/1970 0:0:11> ADSL connected
1/1/1970 0:0:15> PPP1 PPPoE Session is established.
1/1/1970 0:0:35> PPP PAP Authentication success
1/1/1970 0:0:35> PPP1: PPP IP address is 80.180.115.7
1/1/1970 0:0:35> PPP1: PPP Gateway IP address is ↩
↪192.168.100.1
1/1/1970 0:0:35> PPP1: DNS Primary IP address is ↩
↪81.74.224.227
1/1/1970 0:0:35> PPP1: DNS Secondary IP address is ↩
↪212.216.112.112
1/1/1970 0:0:35> NAT/NAPT Session Start: VC# 0, WAN IP is ↩
↪80.180.115.7
1/1/1970 0:0:35> Initialized DMZ host.
1/1/1970 0:0:35> NAPT: many-to-one default session is up.
1/1/1970 0:0:36> PPP1 Session is up.
5/31/2003 22:42:35> Received time from Time Server ↩
↪128.138.140.44
```

In questo caso, si può verificare che tutto è andato a buon fine, dal momento che l'indirizzo IPv4 esterno è stato acquisito regolarmente, ma si può osservare una cosa imprevista:

```
1/1/1970 0:0:35> PPP1: PPP Gateway IP address is 192.168.100.1
```

Si intuisce che il router abbia la necessità di attribuire questo indirizzo per qualche ragione e probabilmente non c'è modo di modificarlo. Se si scopre una cosa del genere, è bene tenerne conto nella configurazione della rete locale, in modo da non interferire.

Purtroppo può succedere che le cose siano più complesse di così,

a causa delle procedure utilizzate dal fornitore. Tanto per fare un esempio comune, il fornitore potrebbe concedere l'accesso in modo preliminare utilizzando un nominativo utente e una parola d'ordine standard, per tutti gli utenti (una cosa del tipo: utente '**pippoadsl**' e parola d'ordine '**pippoadsl**'). In questo modo, gli utenti che accedono con tale identificazione possono raggiungere solo a servizi determinati, con lo scopo di completare la procedura di registrazione, ottenendo alla fine il nominativo e la parola d'ordine corretti.

In queste situazioni, occorre considerare un fatto importante: non è possibile fare nulla che non sia stato previsto in anticipo; per esempio non è possibile risolvere i nomi a dominio in proprio, perché l'accesso ai serventi DNS principali risulterebbe impedito. È proprio dalla lettura delle informazioni ottenute dal router che si può sapere come modificare, forse solo temporaneamente, il file '`/etc/resolv.conf`', per poter poi accedere al sito da cui si può completare la registrazione e ottenere i dati mancanti:

```
1/1/1970 0:0:35> PPP1: DNS Primary IP address is 81.74.224.227
1/1/1970 0:0:35> PPP1: DNS Secondary IP address is 212.216.112.112
```

42.7.4 DNS

Un router ADSL, come si vede dalla sezione precedente, dovrebbe essere in grado di ottenere dalla controparte l'informazione sui serventi DNS che possono essere utilizzati. Di solito, una volta ottenute queste informazioni, il router dovrebbe da solo gestire un servizio DNS, che in pratica rinvia semplicemente le richieste ai serventi esterni. Pertanto, la configurazione del DNS nella rete locale, potrebbe prevedere semplicemente l'accesso al router ADSL come se contenesse un servente DNS vero e proprio.

Se il router ADSL non fornisce un registro per vedere ciò che accade nella connessione con l'esterno, diventa indispensabile utilizzare il router stesso come servente DNS.

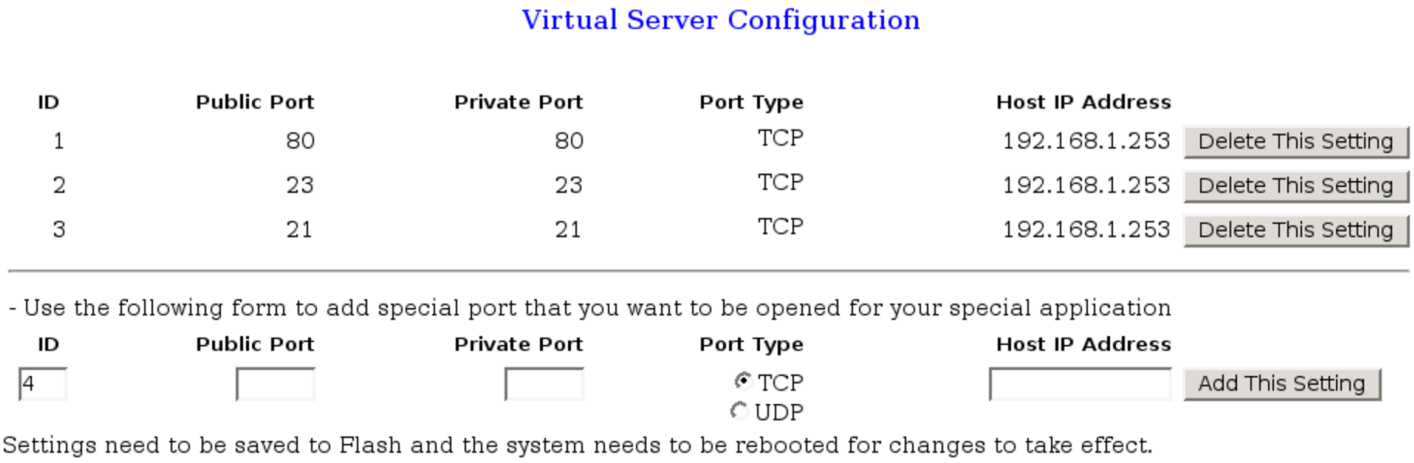
42.7.5 Protezione e accesso dall'esterno

«

In condizioni normali, un router NAT di questo tipo consente tutte le comunicazioni che hanno origine dall'interno, bloccando probabilmente tutti i pacchetti provenienti dall'esterno che non sono riferiti ad alcuna comunicazione preesistente. Questa può essere una soluzione molto semplice ai problemi di sicurezza, ma non consente di ricevere accessi dall'esterno.

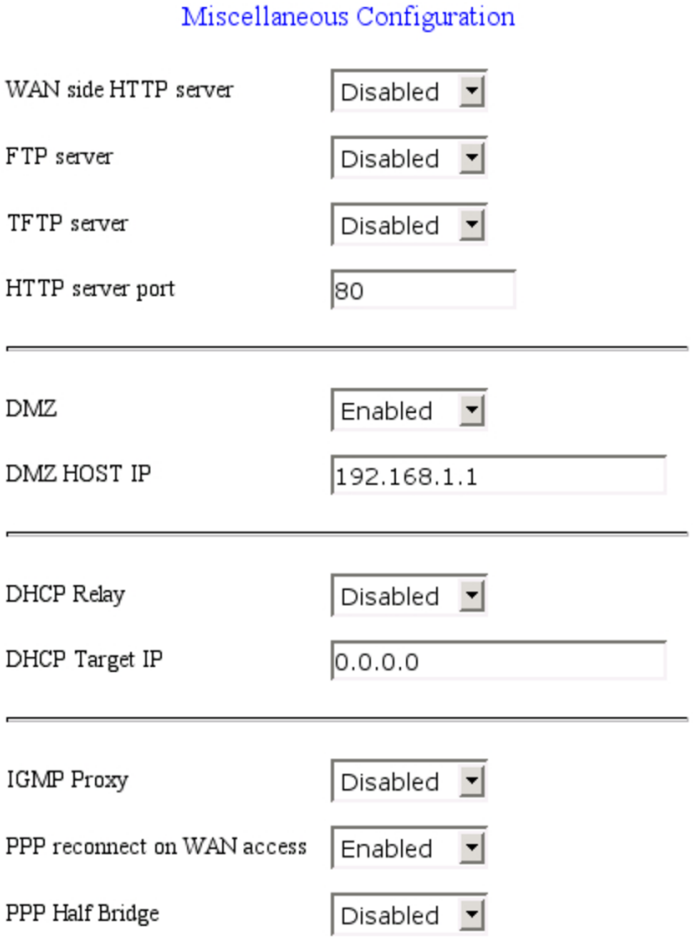
Un router più evoluto potrebbe consentire di dichiarare delle ridirezioni precise per connessioni TCP e UDP che vengono tentate dall'esterno verso porte determinate. Per esempio potrebbe essere utile definire una ridirezione del genere per le richieste che riguardano la porta 80 verso l'elaboratore della rete locale che ospita un servente HTTP (anche se un indirizzo IPv4 dinamico offre poche possibilità di utilizzare un servizio del genere).

Figura 42.103. Ridirezione di alcune porte verso un elaboratore della rete locale (indirizzo 192.168.1.253), con un router Pirelli.



Quando il router non è in grado di ridirigere un traffico particolare verso un elaboratore della rete interna, dovrebbe essere possibile almeno inviare tutti i pacchetti che non sono associati a comunicazioni preesistenti verso un indirizzo che potrebbe essere indicato come «zona demilitarizzata». Naturalmente, l'elaboratore che si trova a ricevere questi pacchetti risulta completamente accessibile dall'esterno, come se avesse l'indirizzo IP pubblico ottenuto dal router stesso e deve essere difeso in qualche modo (per esempio configurando la gestione del filtro dei pacchetti IP).

Figura 42.104. In questa pagina si vede in particolare la ridirezione di tutto il traffico che ha inizio dall'esterno verso l'indirizzo 192.168.1.1. La sigla «DMZ» sta per *demilitarized zone*, ovvero, zona demilitarizzata. L'esempio si riferisce a un router CNet.



Quando si vuole realizzare un tunnel IPv6 (sezione 32.15) è praticamente indispensabile agire in questo modo, facendo sì che poi il nodo esposto diventi anche un router IPv6.

42.7.5.1 Firewall

«

Quando il router consente la configurazione come firewall, le cose si complicano ed è molto probabile che sia consentito l'accesso dall'esterno in modo predefinito.

Per motivi di sicurezza è bene evitare che sia concessa la configurazione del router dall'esterno, ovvero al di fuori della rete locale.

Qualunque sia la configurazione del firewall che si intende applicare, occorre verificare con programmi di scansione (come Nmap), dall'esterno della propria rete locale (si veda la sezione 43.7).

Figura 42.105. Configurazione di un firewall che dovrebbe bloccare tutto il traffico diretto verso l'interfaccia esterna (non correlato alle comunicazioni interne).

Note:
If Ip = 0.0.0.0, addresses are ignored
If Wan = alias wan IP address
If From = 0 and To = 0, ports are ignored
If Protocol = IP, ports are ignored

Operation

Submit Reset

#	Source	Destination	Action	Protocol	Direction
1	IP 0.0.0.0 Mask 0.0.0.0 Port From 0 To 0	IP wan Mask 255.255.255.255 Port From 0 To 0	DENY	IP (ALL)	INPUT

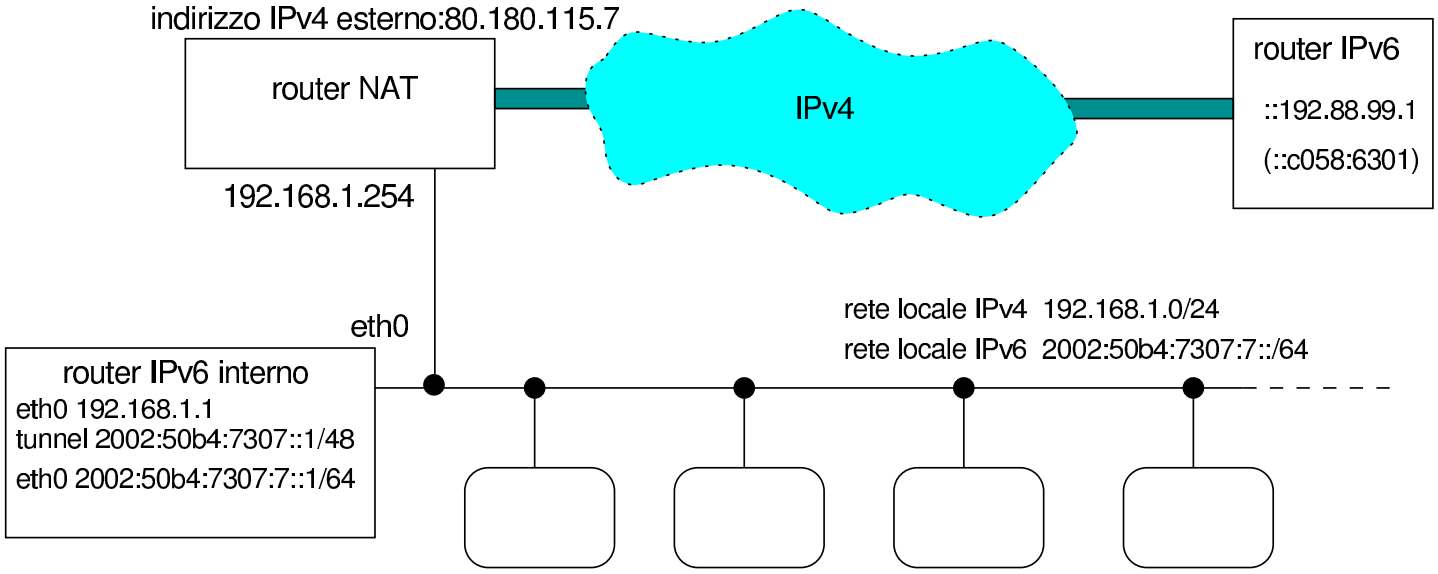
Può anche darsi che non si riesca o non ci sia il modo di disabilitare qualunque risposta dalle porte che di solito servono ad accedere dall'esterno per configurare il router; in questi casi, si può tentare di ridirigere quelle porte (o tutto il traffico non correlato a quello ge-

nerato dall'interno) verso un indirizzo inutilizzato della rete locale, oppure, addirittura verso macchine esterne di fantasia.

42.7.5.2 Tunnel 6to4

Per completezza, viene mostrato in breve come configurare un sistema GNU/Linux in modo da attraversare un router ADSL con un tunnel 6to4. I dati riportati nell'esempio sono coerenti con gli altri esempi del capitolo.

Figura 42.106. Rete locale con indirizzi IPv4 privati, che accede alla rete esterna attraverso un router che non riconosce i tunnel 6to4.



I comandi seguenti realizzano il tunnel nel nodo che deve svolgere il ruolo di router IPv6 con un sistema GNU/Linux; si osservi che l'indirizzo IPv4 80.180.117.7 si traduce in esadecimale come $50B47307_{16}$:

```
# ip tunnel add name t6to4 mode sit remote any local
192.168.1.1 [Invio]

# ip link set dev t6to4 up [Invio]
```

```
# ip -6 address add local 2002:50b4:7307::1/48 scope global ↩
↪       dev t6to4 [Invio]

# ip -6 route add to 2000::/3 via ::192.88.99.1 dev t6to4
metric 1 [Invio]

# ip -6 address add local 2002:50b4:7307:7::1/64 ↩
↪       scope global dev eth0 [Invio]

# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding [Invio]
```

Si osservi che questa tecnica è spiegata con maggiore dettaglio nella sezione 32.15.

42.7.6 Configurazione con indirizzi statici

Si suppone di avere ottenuto un pacchetto di otto indirizzi IPv4 statici, secondo le modalità seguenti:

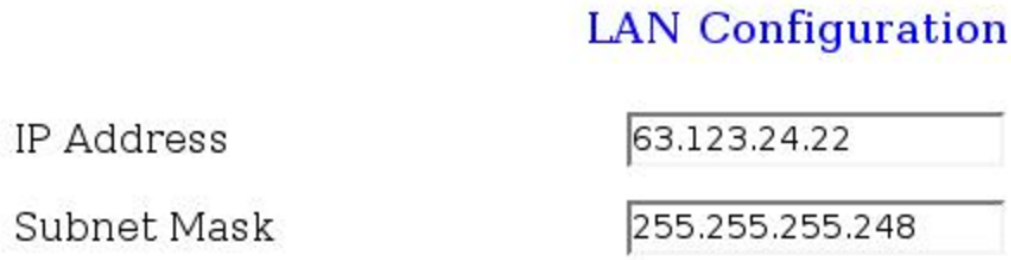
Indirizzo punto-punto assegnato al router:	194.152.059.045
Maschera di rete per l'indirizzo punto-punto:	255.255.255.252
Indirizzo di rete degli indirizzi statici assegnati per la rete locale:	63.123.24.16
Maschera di rete per gli indirizzi statici assegnati alla rete locale:	255.255.255.248
Protocollo per il collegamento punto-punto:	RFC 1483 LLC

Partendo dall'indirizzo di rete 63.123.24.16, conoscendo la maschera di rete, 255.255.255.248, si determina che si possono utilizzare gli indirizzi da 63.123.24.17 a 63.123.24.22 per i nodi.

Si assegna inizialmente un indirizzo IPv4 statico all'interfaccia interna del router, evitando di attivare un eventuale servizio DHCP, che

in questo caso sarebbe poco appropriato.

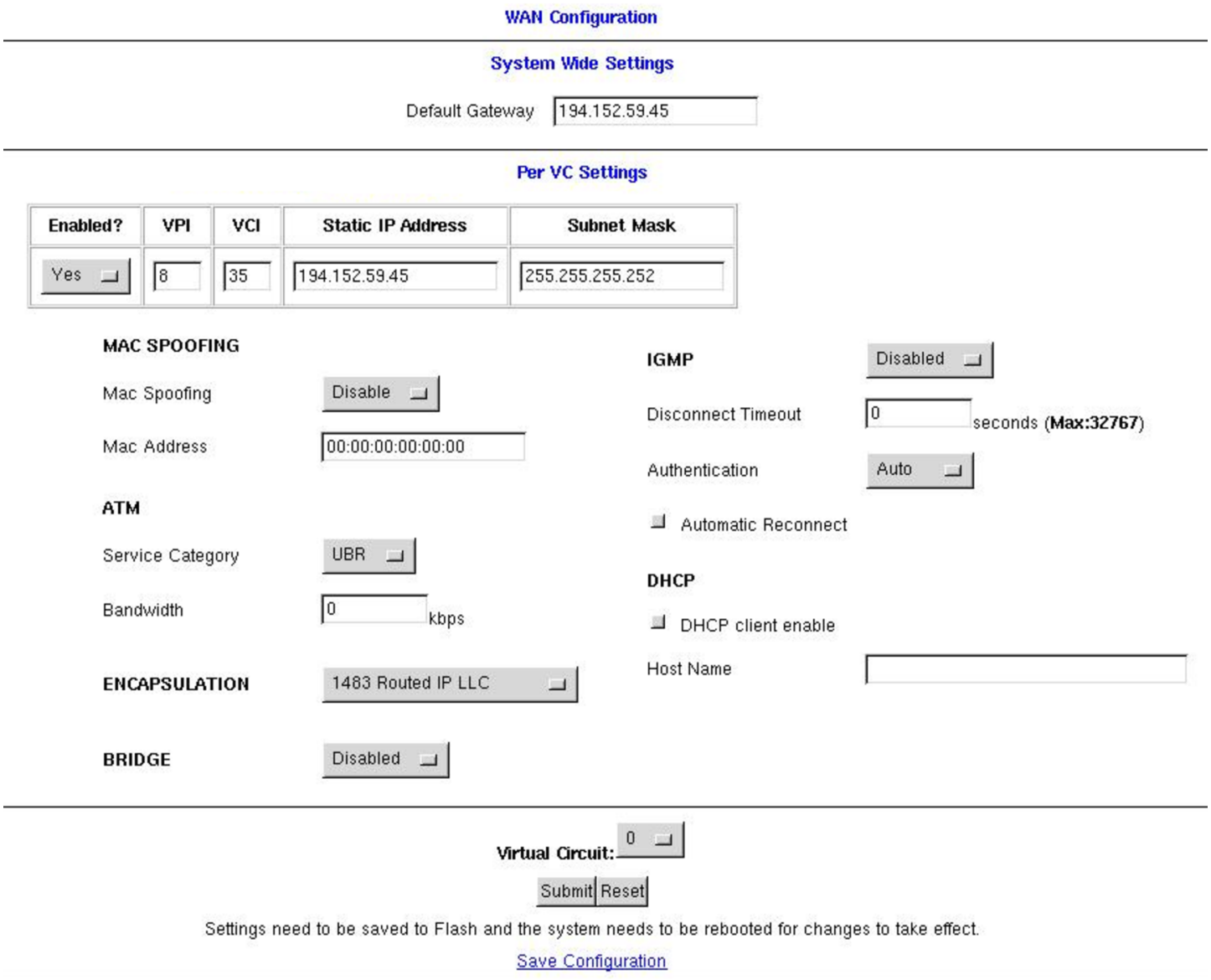
Figura 42.108. Un esempio di pagina di configurazione della rete interna con indirizzi statici. All'interfaccia del router collegata alla rete interna, si assegna l'indirizzo 63.123.24.22.



La gestione del NAT viene disabilitata, perché i nodi locali possono disporre di indirizzi IPv4 pubblici.

Quando è accertato che il collegamento della rete locale funziona correttamente (utilizzando gli indirizzi ottenuti), si può passare alla configurazione del lato esterno (WAN). È qui che si deve definire il protocollo di comunicazione.

Figura 42.109. La pagina di configurazione del collegamento ADSL, con il protocollo RFC 1483 LLC, utilizzando un router Pirelli.



Per le reti italiane, i parametri VPI e VCI corretti sono solitamente 8 e 35 rispettivamente.

Il registro del router potrebbe risultare contenere le informazioni seguenti:

```
1/1/1970 0:0:0> Ethernet Device 0 Detected
1/1/1970 0:0:0> ATM: Detected
1/1/1970 0:0:0> ATM: Setting up vcc0, VPI=8, VCI=35
1/1/1970 0:7:13> ATM Connected
1/1/1970 0:7:13> ATM layer is up, cell delineation achieved
1/1/1970 0:7:13> ADSL connected
```

42.8 Riferimenti

«

- Terry Dawson, *Linux NET-3-HOWTO, Linux Networking*, http://tldp.org/HOWTO/NET3-4-HOWTO.html
- Mark Grennan, *Firewalling and Proxy Server HOWTO*, http://tldp.org/HOWTO/Firewall-HOWTO.html
- *Squid Web Proxy Cache*, http://www.squid-cache.org/
- W3C, *Platform for Internet Content Selection (PICS)*, http://www.w3.org/PICS/
- W3C, *PICS Self-Rating Services List*, http://www.w3.org/PICS/raters.htm#self
- W3C, *Resource Description Framework (RDF)*, http://www.w3.org/RDF/
- *Safe For Kids rating description*, http://www.weburbia.com/safe/ratings.htm
- *The SafeSurf Internet Rating Standard*, http://www.safesurf.com/ssplan.htm
- K. Egevang, P. Francis, *RFC 1631, The IP Network Address Translator (NAT)*, 1994, http://www.ietf.org/rfc/rfc1631.txt

- Rusty Russell, *Linux 2.4 packet filtering HOWTO*, *http://netfilter.org/documentation/HOWTO/packet-filtering-HOWTO.html*
- Mark Grennan, *Firewalling and Proxy Server HOWTO*, *http://tldp.org/HOWTO/Firewall-HOWTO.html*
- Peter Bieringer, *Linux IPv6 HOWTO*, *http://www.tldp.org/HOWTO/Linux+IPv6-HOWTO/*
- Rusty Russell, *Linux 2.4 NAT HOWTO*, *http://www.netfilter.org/documentation/HOWTO/NAT-HOWTO.html*
- Hal Burgiss, *DSL HOWTO for Linux*, *http://tldp.org/HOWTO/pdf/DSL-HOWTO.pdf*

[1] **Tinyproxy** GNU GPL

[2] Il NAT (*Network address translation*) è un procedimento attraverso cui si modificano gli indirizzi IP, di solito allo scopo di consentire a una rete privata di accedere all'esterno.

[3] **Iptables** GNU GPL

[4] In questo caso, viene bloccato il pacchetto ICMP di richiesta di eco, quando tenta di «entrare» attraverso l'interfaccia '`lo`'.

[5] Bisogna ricordare comunque che il SI specifica la lettera «k» minuscola come prefisso moltiplicatore che esprime il valore 10^3.

[6] **netstat-nat** GNU GPL