Essential NETworking

reti: concetti essenziali, applicazioni in ambiente Linux

(2012.11)

Indice

Introduzione

Lo scopo di questi appunti è quello di presentare concetti teorici e applicazioni pratiche delle reti, in modo da consentire l'acquisizione delle nozioni indispensabili teoriche per comprendere i fondamenti del Networking. Accanto alla trattazione teorica, sono affrontati i principali comandi che, in ambiente Linux, consentono la configurazione e il funzionamento di una rete.

Gli appunti seguono un cammino didattico che, partendo dalle nozioni teoriche generali sulle reti, procedono ad illustrare la configurazione di una rete e, quindi, configurazione minima del web server Apache, del firewall. L'obiettivo non è solo quello di trattare la teoria delle reti, ma l'esposizione delle problematiche comuni che affronta chi vuole configurare e utilizzare una piccola rete, per esempio, per applicazioni web-based.

Parallelamente alla trattazione delle reti locali si esaminano le caratteristiche fondamentali del funzionamento e dei servizi di Internet.

Gli esempi di applicazioni concrete, laddove possibile, sono valide per qualsiasi distribuzione Linux anche se, in alcuni casi che verranno specificati, si farà riferimento a determinati software, che possono non essere installati per default in alcune distribuzioni, e a determinate versioni del software. Anche qui è possibile che versioni diverse permettano configurazioni diverse: questi appunti sono sostanzialmente uno strumento didattico che possono fare riferimento, in alcuni casi, a configurazioni specifiche anche se i concetti espressi e gli argomenti trattati vogliono essere generalmente validi.

Negli esempi di uso di comandi, riportati in questi appunti, è utilizzato il font `courier`: le righe digitate da tastiera sono scritte in **`grassetto`** e le righe di risposta sono in caratteri `normali`, il prompt riportato (`$` come utente normale, `#` per root) mostra l'utente che esegue il comando successivo, come nell'esempio:

```
$ ls
ArchivioJavascript  dispense      Internet.sxw        prove
backup              Documents     Mail                public_html
bin                 download      massime             Shared
compiti             firma         Music               sito
configurazioni      Generale.stw  mycheckbook         smb4k
dcc                 immagini      mycheckbook.backup  SwingSet.wmv
Desktop             installa      News                tmp
```

Per poter seguire in modo migliore gli esempi riportati nella parte applicativa, è richiesta una conoscenza essenziale di HTML e PHP, argomenti non trattati in questi appunti. Occorre, inoltre, conoscere il funzionamento generale di un sistema Linux. Comandi particolari, qualora necessari, verranno ricordati in sintesi all'occorrenza.

In appendice sono riportati i sorgenti delle pagine HTML utilizzate negli esempi.

Classificazioni delle reti

Si può parlare di rete quando si hanno tre o più computer collegati fra di loro allo scopo di permettere lo scambio di informazioni o la condivisione di risorse sia hardware che software. Anche due soli computer possono essere collegati per comunicare, ma in questo caso, pur rimanendo valide le configurazioni software, non esiste un hardware di rete: il collegamento viene effettuato, direttamente, utilizzando un cavo particolare; e questo è il motivo per cui, spesso, per due soli computer non si parla di rete.

Per classificare le reti è uso adottare due riferimenti: la ***tecnologia di trasmissione*** e la ***scala***.

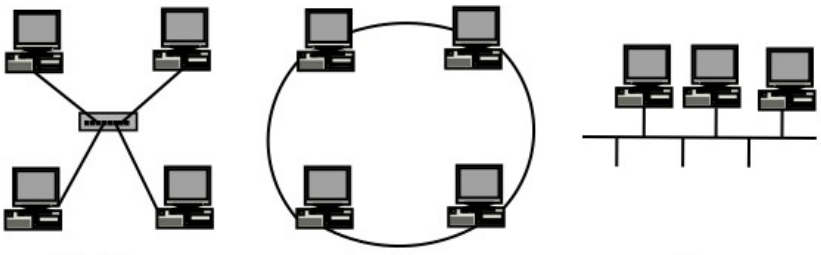
Dal punto di vista della tecnologia di trasmissione le reti si possono classificare in:

- Reti **broadcast**. Le reti di questo tipo sono dotate di un unico canale trasmissivo condiviso da tutti. I messaggi sono divisi in piccole unità (*pacchetti*), contenenti anche informazioni per individuare il destinatario (*indirizzo*), che viaggiano nel canale trasmissivo. La macchina che si riconosce destinataria di un pacchetto lo processa, le altre lo ignorano. In questi sistemi è possibile anche operare in modalità ***broadcasting***: una macchina trasmette, utilizzando un indirizzo speciale, un messaggio che dovrà essere processato da tutti i computer della rete.
- Reti **punto-punto**. In questo caso ci sono molte connessioni fra una stazione e un altra. I pacchetti, per raggiungere la destinazione, attraversano varie macchine intermedie. I cammini che congiungono chi trasmette e chi dovrà ricevere possono essere anche multipli e di diversa lunghezza ed è quindi importante, in questi casi, stabilire il migliore.

In linea generale le piccole reti tendono ad essere del primo tipo, mentre quelle di grandi dimensioni sono del secondo tipo.

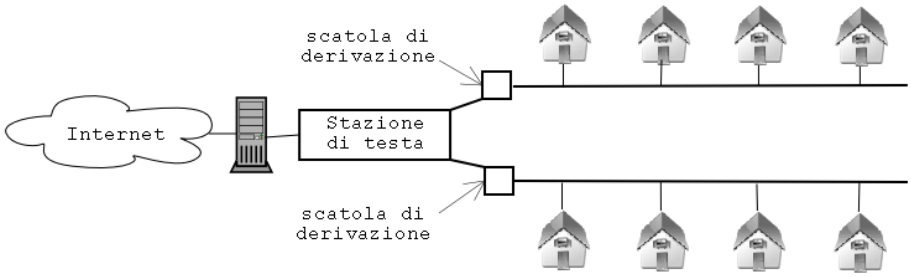
Dal punto di vista della estensione, le reti si distinguono in:

- **Reti locali** (Local Area Network – LAN). La rete copre un edificio o un gruppo di edifici vicini fra di loro. Per le LAN broadcast sono utilizzati tre schemi di collegamento dei computer fra di loro (***topologie*** di rete):



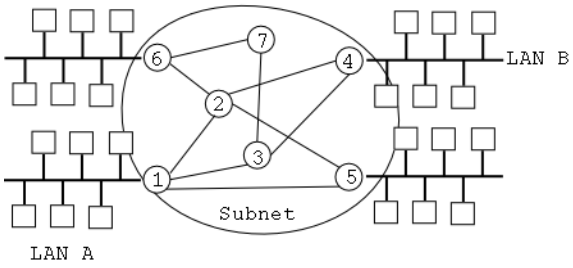
  Nella topologia a stella tutti i nodi sono collegati ad un nodo principale che gestisce le comunicazioni fra i vari computer. Nella topologia ad anello i nodi sono collegati in sequenza in modo da avere, per ognuno, un precedente e un successivo. Nella topologia a bus tutti i nodi condividono un unico canale trasmissivo.

- **Reti metropolitane** (Metropolitan Area Network – MAN). In questo caso la rete copre l'area di una intera città. Un esempio di rete di questo tipo è la rete della televisione via cavo.



la diffusione di Internet ha spinto gli operatori delle TV via cavo ad utilizzare il sistema di distribuzione del segnale video, modificato, anche per generare una MAN per l'accesso ad Internet: un singolo cavo è condiviso tra più case.

- **Reti geografiche** (Wide Area Network – WAN). Una rete di questo tipo copre un'area geografica vasta come una nazione o un continente.



le macchine degli utenti, chiamate anche ***host*** e facenti parte delle reti locali, sono collegate fra di loro da una ***communication subnet*** formata da apparecchi hardware chiamati ***router***, collegati fra di loro in una rete con *topologia a maglia*. Se, per esempio, un host della LAN A vuole comunicare con un host della LAN B, divide il messaggio in pacchetti che, attraverso il router 1 cercano di raggiungere il router 4. Ogni router sceglie in base a precisi algoritmi (*algoritmi di routing*) la strada da percorrere (1-3-4 piuttosto che 1-2-4 o, per esempio, 1-3-7-6-2-4). L'host ricevente riassembla i pacchetti e può utilizzare il messaggio.

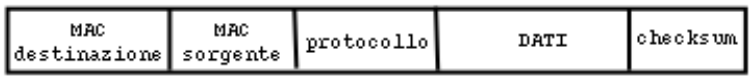
Trasmissione fisica di dati: reti Ethernet

In una rete locale di tipo broadcast, dove il canale trasmissivo è condiviso, un problema che deve essere affrontato e risolto è quello delle *collisioni*. Due stazioni occupano il canale di trasmissione iniziando a trasmettere contemporaneamente: i messaggi si sovrappongono, non sono distinguibili, ed è inutile continuare la trasmissione. In casi come questi è necessario un sistema di arbitraggio che permetta di eliminare le collisioni. Nel tempo sono state proposte diverse soluzioni: nelle reti di tipo Token ring utilizzate, ancora oggi in alcuni casi, da IBM, un piccolo pacchetto (il token) viaggia in continuazione fra una stazione e l'altra (topologia ad anello); la stazione che vuole trasmettere attacca al token il proprio messaggio; il destinatario libera il token e consuma il messaggio. In definitiva la presenza del solo token sulla rete, è indice della possibilità di trasmettere.

Lo standard attualmente più diffuso per le comunicazioni in una rete locale è Ethernet. Questa tecnologia, nata nei primi anni settanta, e aggiornata varie volte, attualmente fa riferimento alla cosiddetta versione 2.0 del 1983 detta anche Ethernet II o standard DIX (Digital, Intel, Xerox). La longevità di questa tecnologia è dovuta a diversi fattori: facilità di implementazione, basso costo, flessibilità. Lo standard ha continuato, e continua, ad evolversi proponendo versioni a 100 Mbps

(megabit al secondo) o 1.000 Mbps e diversi sistemi di collegamento fra le varie stazioni. Il nome adottato si riferisce al mezzo trasmissivo cavo (ether) che rimanda all'*etere* (il mezzo, inesistente e introdotto con una forzatura, che, si pensava nell'antichità, permettesse la trasmissione di segnali elettromagnetici).

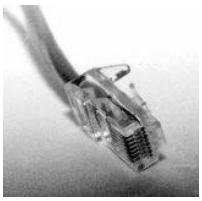
Le schede di interfaccia di una rete Ethernet si autoregolano, nel senso che sono in grado di decidere quando trasmettere e rilevare le collisioni. Inizialmente la scheda *ascolta*, verificando la presenza di un certo segnale elettrico, se il canale è libero e, in questo caso, inizia le trasmissioni. Anche durante la trasmissione la scheda ascolta il canale per rilevare modifiche anomale del segnale che indicano collisioni e, in questi casi, sospende la trasmissione che riprende dopo un intervallo di tempo casuale. Ogni scheda ha un indirizzo fisico (chiamato anche ***MAC*** Address, dove MAC sta per Media Access Control), unico al mondo, di 48 bit (24 ad indicare il costruttore e 24 utilizzati dal costruttore per distinguere una scheda dall'altra) che la individua in modo univoco, utilizzato per le comunicazioni.



frame Etehernet

I dati in una Ethernet viaggiano in unità chiamate ***frame*** ognuno contenete indirizzi MAC di destinazione e sorgente, protocollo da utilizzare per decodificare i dati, dati da trasmettere e informazioni di *checksum* per la verifica dell'integrità del pacchetto stesso.

Una rete Ethernet ha una topologia a bus: un unico cavo unisce tutte le stazioni ciascuna delle quali ascolta tutto ciò che passa nella rete, ma soltanto una per volta ha la possibilità di trasmettere. In realtà i collegamenti elettrici hanno una topologia a bus ma la topologia fisica normalmente è a stella, utilizzando speciali apparecchiature (*hub*), per aumentare l'affidabilità della rete e facilitare la ricerca di difetti nel cavo.



Il cavo utilizzato per i collegamenti in una rete Ethernet è formato da 4 doppini intrecciati (cavo *twisted pair*), 4 coppie di fili come quelli utilizzati nei cavi telefonici, terminante ai due estremi con connettori RJ45 simili a quelli delle spine telefoniche, ma più larghi per poter contenere 8 fili invece dei 2 del cavo telefonico.

Uno spinotto RJ45 va alla scheda montata nel computer che si vuole collegare, l'altro all'hub. È possibile anche collegare fra di loro due soli computer provvisti di scheda Ethernet; in questo caso si usa un *cavo incrociato*, con i collegamenti diversi rispetto a quelli di un cavo Ethernet normale.

Hardware di rete: hub, switch, bridge e router

Le prime reti Ethernet erano cablate con una topologia a bus e usavano, come canale di comunicazione condiviso, un cavo coassiale. Le successive evoluzioni di Ethernet, principalmente orientate al cablaggio, hanno portato ad una configurazione diversa. L'impianto elettrico continua ad essere quello di prima: c'è un solo precorso comune per tutte le stazioni e quello che viene trasmesso da una stazione è ascoltato da tutte. Per quanto riguarda la comunicazione di dati, la topologia continua ad essere a bus, la topologia fisica è a stella. C'è un apparecchio, cui si è fatto cenno più volte (l'hub), a cui si collegano tutte le stazioni per mezzo di un cavo dedicato formato da un doppino telefonico.

- **Hub**: è un ripetitore-concentratore. Il suo compito è quello di ripetere il messaggio ricevuto in tutte le diramazioni. Sostanzialmente, come osservato prima, riproduce le caratteristiche del bus in una topologia a stella.
- **Switch**: è un commutatore. Il suo compito è quello di effettuare una selezione sui pacchetti in transito. Ogni stazione che vuole trasmettere, invia allo switch un particolare pacchetto; la scheda dello switch controlla se il pacchetto è destinato ad una delle stazioni collegate ad essa e, in questo caso, lo inoltra lungo il tratto interessato.
- **Bridge**: è un *ponte* fra una LAN e l'altra, interviene quando diverse LAN hanno necessità di interconnettersi fra di loro. Il suo compito è quello di duplicare i pacchetti in arrivo, verso le altre reti cui è connesso. Collega reti fisiche uguali.
- **Router**: è un ponte fra reti, anche fisicamente diverse, ma che comunichino utilizzando lo stesso protocollo. Si tratta di un apparecchio che interviene ad un livello più alto rispetto a quanto trattato finora. La comunicazione fra due stazioni non è solo un problema fisico di trasmissione, ma riguarda quelli che comunemente sono chiamati protocolli: il modo con cui sono scritti i messaggi trasmessi che deve essere lo stesso utilizzato dall'emittente e dal ricevente. Se il protocollo è lo stesso, un router può mettere in comunicazioni due reti anche se fisicamente le comunicazioni avvengono in modo diverso: per esempio una LAN e Internet.

A volte viene usato anche il termine **gateway** che è un nodo che consente di mettere in comunicazione una rete con qualcosa di altro: per esempio un router da contattare per raggiungere un'altra rete.

Utilizzando hub e switch, per esempio, una Ethernet diventa più resistente a malfunzionamenti. In una topologia a bus una interruzione nel cavo condiviso, interrompe la rete. L'utilizzo di hub o switch permette sia di isolare un singolo tratto malfunzionante sia di aggiungere, in modo semplice e poco costoso, ulteriori stazioni.

Internet e i suoi progenitori

La storia di Internet nasce nella seconda metà degli anni '50 in piena guerra fredda, quando il dipartimento della difesa degli USA si pose il problema di creare una rete di trasmissione che potesse sopravvivere ad una guerra nucleare. La spinta definitiva al progetto avvenne allorquando gli USA si resero conto di essere stati battuti nella corsa allo spazio dal lancio, ad opera dell'URSS, del primo Sputnik. Venne creata l'organizzazione ARPA (Advanced Research Project Agency) i cui studi condussero ad una subnet a commutazione di pacchetto i cui host erano, ognuno, dotati di un router che fosse in grado di decidere l'instradamento dei pacchetti verso la prossima destinazione. Tutto ciò porta alla nascita di ARPANET.

ARPA incoraggiò la ricerca di protocolli che potessero permettere una facile comunicazione fra le LAN dei centri di ricerca e ARPANET e da questi sforzi nacque il protocollo TCP/IP.

Negli anni '70 l'organismo statunitense NFS (National Science Foundation), conscio dell'importanza assunta da ARPANET per la ricerca universitaria, poiché permetteva condivisione e collaborazione a progetti di ricerca, ma anche dei limiti legati al fatto che ci si potesse collegare alla rete solo se si era in possesso di un contratto di ricerca aperto con il ministero della difesa, progetta un successore di ARPANET aperto a tutti (la rete NFSNET) cercando nel contempo di creare consorzi non profit per la commercializzazione. La rete si basava su TCP/IP.

Una forte accelerata alla espansione della rete si ebbe quando ARPANET e NFSNET furono interconnesse e fino agli anni '90 era usata principalmente da ricercatori che sostanzialmente utilizzavano servizi di e-mail, gruppi di discussione (*newsgroup*), trasferimento file (FTP).

Nel 1989 nasce presso il CERN, ad opera del fisico Tim Berners-Lee, il Web cresciuto grazie all'esigenza dei team di ricercatori di tutto il mondo, operanti presso il CERN, di rapporti collaborativi: si tratta di una rete di documenti collegati contenenti anche media diversi (foto, animazioni, filmati, suoni). Nel 1994 il CERN e il MIT firmano un accordo per creare il ***Word Wide Web Consortium*** (W3C) con Berners-Lee come direttore, organizzazione che ha lo scopo di sviluppare il Web, standardizzare i protocolli e incoraggiare l'interoperabilità fra i siti. La home page del consorzio si trova all'indirizzo *www.w3c.org*.

La crescita di Internet è stata inoltre stimolata, nel corso degli anni '90, dalla nascita degli ***ISP*** (Internet Service Provider), aziende che danno la possibilità ad un utente qualsiasi, di chiamare uno dei loro computer, collegarsi ad Internet e usufruire dei servizi disponibili (web, e-mail e quant'altro).

Internet: connessioni, PPP

Per permettere la comunicazione fra due computer situati in luoghi fisici poco distanti, il sistema più economico e rapido è quello, esaminato nelle LAN, di stendere un cavo fra di essi. Se invece le distanze cominciano a crescere, la stesura di cavi fra le varie stazioni comincia a diventare antieconomica oltre a presentare problematiche spesso non risolvibili (per esempio diritti dei proprietari dei terreni su cui devono passare i cavi per i collegamenti). È più conveniente in questi casi utilizzare ed appoggiarsi a sistemi di telecomunicazione esistenti.

La rete telefonica già esistente, ed enormemente ramificata in modo da poter raggiungere qualunque parte del mondo, può essere un appoggio per le trasmissioni di informazioni fra un computer e l'altro. Il problema principale però che si frappone è il fatto che all'inizio (la situazione odierna sta velocemente cambiando con l'introduzione di tecnologia digitale) tale rete era stata progettata per la trasmissione di voce umana. La soluzione adottata è quella di trasmettere un tono continuo (*onda portante*) e su di esso *modulare* (scrivere modificando una delle caratteristiche dell'onda) il messaggio da trasmettere. Un apparecchio che accetta in ingresso un flusso seriale di bit, produce una portante modulata che può essere inoltrata nella rete telefonica, è chiamato ***modem*** che, da come si deduce dal nome (modulatore-demodulatore), può operare anche in senso inverso riconducendo cioè i segnali analogici, provenienti dalla rete telefonica, in segnali digitali per il computer.

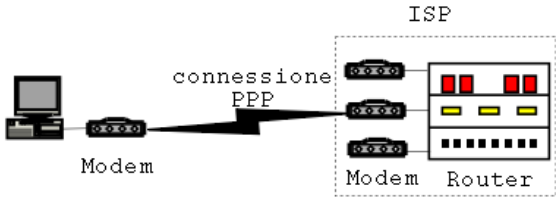
Tutti i modem moderni trasmettono in full-duplex: i dati viaggiano in contemporanea nei due sensi della connessione. La velocità di trasmissione, dipendente dal sistema di comunicazione, è anche soggetta a standard internazionali chiamati, nelle ultime versioni per linea analogica, V.90 (33,6 kbps in trasmissione e 56 kbps in ricezione) e V.92 (fino a 48kbps in trasmissione, stessa velocità di V.90 in ricezione).

L'importanza di Internet nelle attività commerciali ha portato le aziende telefoniche a proporre servizi e prodotti più competitivi dal punto di vista della velocità: nascono le linee ***DSL*** (Digital Subscriber Line) di cui ***ADSL*** (Asymmetric DSL, chiamata così per la differenza di velocità in ricezione e in trasmissione) è quella più popolare che permette, in genere e nelle configurazioni comuni, di raggiungere velocità, in ricezione, che vengono misurate in diversi Mega.

La rete telefonica, progettata per la trasmissione della voce umana, taglia per mezzo dell'uso di filtri, la capacità del collegamento. Il sistema di fondo con cui funziona il sistema DSL è quello di rendere disponibile l'intera capacità del collegamento. L'utente, abbonato al sistema e abilitato ad usare l'intera capacità disponibile, interpone un filtro fra la presa telefonica e l'apparecchio telefonico o il modem. Il filtro per il telefono taglia le frequenze oltre un certo limite, quello per il modem taglia quelle sotto un altro limite.

Comunque ci si connetta (modem analogici, linee DSL), una WAN è formata da linee punto-punto:

- Più LAN ognuna costituita da diversi host e router sono collegate in modo che esiste una connessione fra i vari router (*backbone*, dorsale). Le richieste di connessioni per il mondo esterno passano da un router che ha più linee dedicate che lo collegano ad altri router che permettono di raggiungere la destinazione. Questi router e le linee che li collegano sono le sottoreti sulle quali è costruito Internet.
- Il singolo utente che si collega ad Internet stabilisce una connessione dedicata fra il proprio PC e il router del Provider e diventa, a tutti gli effetti, un host connesso ad Internet.



La gestione del traffico fra i vari router o fra utente e ISP è governata da un protocollo punto-punto, che nel caso di Internet, è chiamato ***PPP*** (Point-to-Point Protocol) che ha i compiti di rilevazione degli errori di trasmissione, supporto per vari protocolli di comunicazione, gestire la connessione ed eventuale autenticazione. In poche parole, si potrebbe dire che, prepara la conversazione che avverrà compresa la conoscenza, fra emittente e ricevente, della lingua che si parlerà durante lo scambio di comunicazioni.

Una cosa importante che deve essere stabilita all'avvio della connessione è, inoltre, l'*assegnazione di un indirizzo* alla macchina che si collega in modo che questa possa far parte della rete. Gli ISP dispongono di un range di indirizzi fra i quali ne viene scelto uno da assegnare al computer per ogni collegamento richiesto. Tale indirizzo identificherà il computer e lo accompagnerà finché non verrà disconnesso.

ISO/OSI e TCP/IP

I dati, all'interno di una rete, viaggiano in forma di *pacchetti*. La dimensione del pacchetto dipende dalle caratteristiche fisiche del tratto di rete che devono attraversare: per attraversare un tratto di rete potrebbe essere necessario dividere, in pacchetti più piccoli, un pacchetto o, anche, effettuare l'operazione inversa.

I pacchetti vengono trasmessi in base a determinate regole chiamate ***protocolli***: per effettuare qualunque tipo di comunicazione è necessario che gli interlocutori stabiliscano la lingua, le convenzioni della conversazione.

Il punto di riferimento dei protocolli per le reti è il modello di riferimento ***OSI*** (Open System Interconnection) fondato su una proposta dell'International Standard Organization con lo scopo di standardizzare i protocolli di comunicazione e, per questo, noto con la sigla **ISO/OSI**. Anche se il

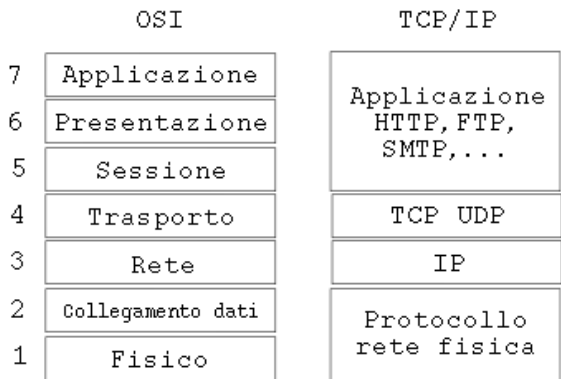
modello è in disuso, ha ancora validità di carattere generale.

Nel modello OSI la gestione della rete è scomposta in livelli o ***layer***. Il modello definisce il compito di ciascuno strato e serve da riferimento: non è necessario che, nelle implementazioni pratiche, i livelli siano tutti presenti per come definiti dallo standard.

I dati da trasmettere sono prodotti al livello più alto poi, con un processo di trasformazione che prevede di inserire il pacchetto (*imbustare* il pacchetto) in pacchetti più grandi contenenti le informazioni necessarie, il pacchetto viene passato ai livelli inferiori finché non arriva al primo che si occupa della trasmissione. Il ricevente effettua l'operazione inversa di *estrazione* dalla busta del livello inferiore.

- Lo **strato fisico** si occupa della trasmissione di bit grezzi sul canale di comunicazione.
- Lo **strato di collegamento dati** (Data Link) controlla la correttezza delle sequenze di bit trasmesse ed, eventualmente, ne richiede la ritrasmissione.
- Lo **strato di rete** (Network layer) si occupa del routing: sceglie il cammino che devono percorrere i dati.
- Lo **strato di trasporto** trasporta in modo logico i dati in pacchetti e gestisce l'invio. Dal suo punto di vista la comunicazione avviene fra due porte: una per il mittente e una per il destinatario. A questo livello l'indirizzo è identificato dal numero di porta.
- Lo **strato di sessione** si occupa di vari sistemi di sincronismo, dei turni di trasmissione e ricezione.
- Lo **strato di presentazione** si occupa della conversione dei dati in formato standard e, quando occorre, della crittazione/decrittazione dei dati nel caso di collegamenti protetti.
- Lo **strato applicazione** comprende una seri di protocolli che sono quelli più direttamente utilizzati dagli utenti.

Un ulteriore modello di rete in cui sono presenti protocolli largamente utilizzati nelle reti è il modello ***TCP/IP***. Progenitore di tutti i modelli di rete di tipo geografico, nasce con Internet con lo scopo di permettere, in maniera semplice, l'interconnessione di più reti. Questo modello è oggi affermato come standard non solo di Internet, ma anche di LAN Ethernet. Il nome deriva da due dei protocolli definiti nel modello.



- Lo **strato IP** (Internet Protocol) ha il compito di consentire all'host di trasmettere i pacchetti alla destinazione corretta. I pacchetti viaggiano in maniera indipendente uno dall'altro e potrebbero, pure, arrivare a destinazione in ordine diverso.
- Nello **strato trasporto** sono definiti due protocolli: **TCP** (Transmission Control Protocol) e **UDP** (User Datagram Protocol). Questi protocolli si occupano della suddivisione dei dati da trasmettere in parti, ricomporre le parti in ricezione e correggere gli errori. I pacchetti sono passati al livello inferiore per l'instradamento.

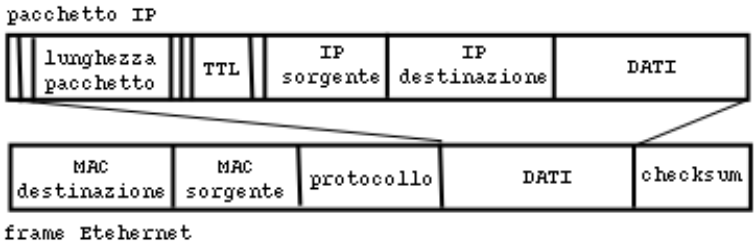
TCP, il vero motore di Internet, è stato progettato per essere un protocollo affidabile (*orientato alla connessione*): in trasmissione divide il messaggio in pacchetti (ripristinando il messaggio in ricezione) ed è in grado di rilevare se il pacchetto è arrivato a destinazione e, in caso contrario, rinvia il pacchetto alla destinazione.

UDP è un protocollo *non connesso*: non garantisce la consegna del pacchetto che, in questo protocollo, è chiamato *datagramma*. È utilizzato principalmente nelle sessioni interattive client-server o nella trasmissione di voce e filmati.

Internet Protocol: indirizzamento IPv4

Il protocollo IP si pone al livello 3 del modello ISO/OSI: a questo livello una rete è un insieme di host identificati da un indirizzo composto da 32 bit, tradizionalmente divisi in 4 ottetti divisi da punti (notazione decimale puntata), nella versione IPv4 attualmente in uso. A causa della notevole espansione di Internet, lo spazio di indirizzi utilizzabili si sta velocemente esaurendo, ed è in corso una conversione degli indirizzi alla versione IPv6 (indirizzi a 128 bit) che consentirà l'utilizzo di un più ampio margine di indirizzi.

IP riceve dal protocollo di livello superiore un pacchetto di dati, a questo aggiunge una propria intestazione, variabile, di lunghezza minima di 20 byte. Il pacchetto IP si comporta come un vagone che riceve, al suo interno, il carico da consegnare che, a sua volta, viene caricato nel vagone di livello inferiore.



Nell'intestazione sono contenuti campi che conservano informazioni diverse:

- Lunghezza totale pacchetto.
- TTL (Time To Live) tempo di vita del pacchetto. Un numero che indica per quanto tempo un pacchetto potrà rimanere in rete alla ricerca della destinazione; trascorso tale tempo, se ancora

non ha potuto raggiungere la destinazione, per evitare di intasare la rete, il pacchetto è eliminato. In realtà TTL è espresso in termini di numero di *hop*: numero di salti fra un router e l'altro che il pacchetto, al massimo, potrà fare. Il numero viene decrementato ad ogni salto.

- Indirizzi IP del sorgente e del destinatario.

Nell'indirizzo IPv4 sono contenute informazioni che riguardano l'indirizzo della rete, cui l'host appartiene, e l'indirizzo dell'host stesso. A seconda di quali convenzioni vengono utilizzate per distinguere l'indirizzo della rete dall'indirizzo dell'host, gli indirizzi IP vengono suddivisi in classi dalla A alla E. Comunemente sono utilizzati gli indirizzi delle prime tre classi:

- **Classe A**. Gli indirizzi cominciano con il primo bit della parte rete (il primo byte) 0, i restanti 3 byte individuano l'host. In definitiva gli indirizzi di questa classe vanno da 1.0.0.0 a 127.255.255.255
- **Classe B**. Per l'identificazione della LAN sono riservati due byte; i rimanenti due identificano l'host. Un indirizzo di questa classe comincia con 10 nel primo byte e gli indirizzi vanno da 128.0.0.0 a 191.255.255.255
- **Classe C**. I primi tre byte per l'indirizzo della LAN, l'ultimo per l'host. Gli indirizzi cominciano con 110 e vanno da 192.0.0.0 a 223.255.255.255. In ragione della scarsità di indirizzi disponibili, si tende ad utilizzare, anche in Internet, indirizzi di tale tipo per ottenere il maggior numero di sottoreti possibili.

Il meccanismo per distinguere la parte rete dalla parte host dell'indirizzo è quello della maschera di rete (***netmask***): un indirizzo che viene abbinato, con l'operatore booleano AND, all'indirizzo da esaminare, per isolare le singole parti. Per esempio la maschera di rete 255.255.255.0 applicata ad un indirizzo di classe C, mostra l'indirizzo della rete: infatti se per esempio si ha l'indirizzo 192.168.1.12, effettuando un AND con la maschera suddetta, si ottiene come risultato 192.168.1.0 (i bit dell'ultimo byte sono azzerati dall'AND con valori 0, gli altri restano così come sono perché, per ognuno, viene effettuato un AND con 1). L'operatore AND applicato all'indirizzo e alla maschera invertita (0.0.0.255) fornisce invece l'indirizzo relativo dell'host.

Gli indirizzi delle classi D ed E non sono sostanzialmente utilizzati perché riservati per usi futuri o particolari.

Alcuni indirizzi appartenenti a varie classi sono riservati per usi speciali:

- 0.0.0.0 *default route*: il cammino predefinito per l'instradamento dei pacchetti.
- 127.0.0.1 indirizzo di *loopback*: l'indirizzo usato, in ogni nodo, per fare riferimento a sé stesso.
- Gli indirizzi da 10.0.0.0 a 10.255.255.255 (classe A), da 172.16.0.0 a 172.31.255.255 (classe B), da 192.168.0.0 a 192.168.255.255 (classe C) sono riservati per le reti locali. Non è possibile accedere dall'esterno (Internet) ad indirizzi del genere.

A parte gli indirizzi riservati, in Internet, per poter permettere la comunicazione fra i vari host è necessario individuarli in maniera univoca. L'ente internazionale ICANN (Internet Corporation for Assigned Names and Numbers), e le sue diramazioni nazionali, hanno il compito di attribuire gli indirizzi in modo che siano univoci.

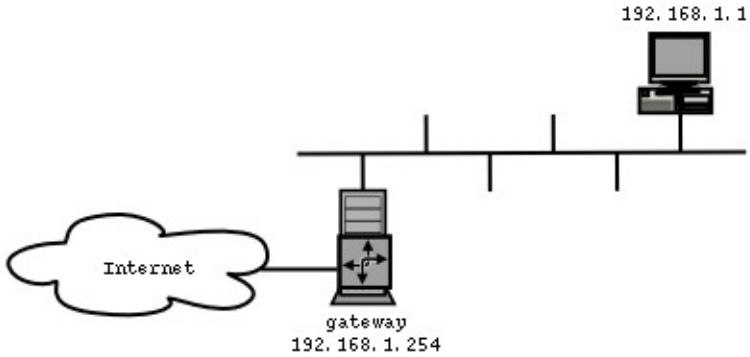
Il pacchetto generato da IP viene passato, per poter essere avviato a destinazione, al livello inferiore. Ethernet non conosce però gli indirizzi IP ma gli indirizzi MAC delle schede: è necessario avere la possibilità di associare ad un indirizzo IP il MAC di una scheda. Per trovare l'indirizzo

viene utilizzato il protocollo ***ARP*** (Address Resolution Protocol): chi vuole trasmettere manda un pacchetto broadcast (a tutti i nodi della rete) per conoscere a quale indirizzo fisico corrisponde un determinato IP; il nodo con quell'indirizzo risponde con il proprio MAC. A questo punto la macchina, che ha inoltrato la richiesta, conosce l'indirizzo del destinatario e può costruire il suo pacchetto di dati contenete il pacchetto IP. Per poter rendere più efficiente la comunicazione, la corrispondenza fra IP e MAC viene tenuta, per un certo tempo, in una cache.

Configurazione interfaccia di rete

Per configurare una scheda Ethernet in modo che il computer possa essere un nodo di una rete, è necessario attribuire un indirizzo IP e definire la tabella di routing in modo che il nodo sappia come trattare i pacchetti in transito. Gli strumenti tradizionali per la configurazione, a basso livello, sono `ifconfig` (interface configuration) e `route`. Sono disponibili, in dipendenza della particolare distribuzione di Linux utilizzata, anche comandi di livello più alto.

Per identificare le interfacce di rete, Linux utilizza un nome di device speciale che non è definito, come solitamente accade per gli altri device, nella directory `/dev`. Il nome utilizzato è `ethX` dove la `X` finale è sostituita da un numero progressivo indicante le varie schede di rete inserite nel computer (`eth0`, `eth1`, ...)



nella rete mostrata nell'immagine sono utilizzati indirizzi di classe C 192.168.x.x e al gateway per raggiungere l'esterno della rete, per esempio per collegare la rete ad Internet, verrà assegnato, come da convenzioni generalmente adottate, l'ultimo indirizzo della rete (si ricorda che 192.168.1.0 è l'indirizzo di rete e che 192.168.1.255 è l'indirizzo di broadcast a cui rispondono tutti i nodi della rete).

Un elaboratore, che sia o no connesso ad una rete, deve necessariamente avere configurata l'interfaccia di loopback:

```
# ifconfig lo 127.0.0.1
# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
...
```

`ifconfig`, seguito dal nome di interfaccia, attribuisce un IP alla stessa, senza nessun parametro mostra tutte le interfacce configurate e attive.

Il comando per la configurazione dell'interfaccia, come anche evidenziato dal prompt, può essere impartito solo dall'utente *root*. Qualsiasi utente può invece utilizzare `ifconfig` per ottenere informazioni sulle interfacce attive.

```
# ifconfig eth0 192.168.1.1 netmask 255.255.255.0
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:40:F4:6D:50:66
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
...
```

con questa sequenza di comandi prima si imposta l'IP e la netmask, poi si interroga la configurazione specifica. Nella risposta del sistema al secondo comando, oltre agli indirizzi impostati, si può notare il MAC Address della scheda: da questo momento esiste la corrispondenza fra IP e MAC.

A questo punto si può verificare che si può raggiungere l'indirizzo impostato:

```
# ping -c 5 192.168.1.1
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: icmp_seq=0 ttl=64 time=0.1 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.1 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.1 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.1 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=0.1 ms

--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.1/0.1 ms
```

il comando `ping` seguito da un IP, come evidenziato dall'output, invia 5 pacchetti verso l'indirizzo specificato e attende la risposta dal destinatario, fornendo informazioni statistiche. Se non si utilizza il parametro `-c`, la trasmissione di pacchetti continua finché non viene interrotta dalla combinazione di tasti *Crtl-c*.

La mancanza di risposta, ai pacchetti inviati, da parte del destinatario può voler dire che il destinatario non è raggiungibile o che il destinatario non risponde ai pacchetti. Si considererà in seguito, l'ultima eventualità.

Configurazione tabella di instradamento

Attribuito l'indirizzo IP al computer, per permettere la comunicazione è necessario stabilire la strada che devono seguire i pacchetti in transito nel nodo (instradamento, routing).

- L'instradamento dei pacchetti attraverso l'interfaccia di loopback, `lo`, è già predisposto dalla `ifconfig` sull'interfaccia. Tutti i pacchetti, in generale diretti alla rete 127.x.x.x, sono trattati dalla macchina stessa.
- Anche la configurazione dell'instradamento attraverso la scheda Ethernet è fatta da `ifconfig`. Per visualizzarla si può utilizzare il comando `route`:

  ```
  # route -n
  Kernel IP routing table
  Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
  192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
  ```

  nella riga della tabella visualizzata è espresso quanto segue: tutto ciò che è destinato alla rete 192.168.1.0 è instradato, attraverso `eth0` (Iface), al gateway di default (l'indirizzo 0.0.0.0 della colonna Gateway). La presenza di `U` nella colonna Flag indica, inoltre, che l'instradamento è attivo. Altri valori che possono essere visualizzati possono essere: `H` (la destinazione è un host), `G`

(destinazione gateway), `D` o `M` (instradamento gestito/modificato da un servizio), `!` (instradamento impedito).

- L'ultima cosa da specificare è il gateway:

```
# route add default gw 192.168.1.254
# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0        0 eth0
```

Per verificare il corretto funzionamento del routing, si può utilizzare il `ping` ad indirizzi della stessa rete e ad indirizzi che dovrebbero essere raggiungibili *passando* dal gateway.

Un altro strumento interessante per le informazioni di routing è `traceroute`:

```
$ traceroute 216.239.59.99
traceroute to 216.239.59.99 (216.239.59.99), 30 hops max, 38 byte packets
 1  192.168.100.1 (192.168.100.1)  52.502 ms  54.000 ms  51.677 ms
 2  host132-250.pool217141.interbusiness.it (217.141.250.132)  51.917 ms  50.699
ms  51.880 ms
 3  host229-8.pool8020.interbusiness.it (80.20.8.229)  68.351 ms  67.133 ms
67.765 ms
 4  r-rm180-vl4.opb.interbusiness.it (151.99.29.214)  67.778 ms  67.203 ms  67.762
ms
 5  rom4-ibs-resid-3-it.rom.seabone.net (213.144.177.177)  66.195 ms  68.913 ms
67.683 ms
 6  linx-lon1-racc1.lon.seabone.net (195.22.209.109)  101.705 ms  100.621 ms
100.185 ms
 7  195.66.226.125 (195.66.226.125)  103.727 ms  104.854 ms  105.925 ms
 8  72.14.238.242 (72.14.238.242)  105.938 ms  105.835 ms 72.14.238.246
(72.14.238.246)  106.028 ms
 9  216.239.49.254 (216.239.49.254)  115.938 ms 216.239.43.91 (216.239.43.91)
116.554 ms  114.479 ms
10  216.239.49.114 (216.239.49.114)  120.711 ms  120.746 ms  120.287 ms
11  216.239.49.126 (216.239.49.126)  125.517 ms  125.038 ms 216.239.59.99
(216.239.59.99)  115.988 ms
```

Nell'esempio si richiede di tracciare il cammino dei pacchetti per arrivare ad uno degli indirizzi di *www.google.it*. Come evidenziato nella prima riga dell'output, vengono inviati verso la destinazione, pacchetti di 38 byte e con un tempo di vita di 30 hop (salti fra router). In realtà i pacchetti sono riusciti a raggiungere la destinazione in 11 salti.

Poiché l'output del comando, per ogni salto, fornisce anche una indicazione sul tempo, è utilizzato nella realtà anche per controllare i colli di bottiglia di una rete: le tratte che pongono più problemi in termini di velocità.

Configurazioni permanenti

La configurazione delle interfacce di rete e della tabella di routing si perdono se si riavvia il sistema o se si disabilita (*si butta giù*) la scheda:

```
# ifconfig eth0 down
# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
```

```
# ifconfig eth0 up
# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
```

Come evidenziato dagli esempi riportati, anche se si riabilita (*si ritira su*) la scheda, il sistema ha *dimenticato* le configurazioni.

Per fare in modo, verificato il corretto funzionamento delle configurazioni, di rendere automatiche le configurazioni, si possono inserire le istruzioni in un file di configurazione eseguito all'avvio del sistema.

Nei sistemi derivati-Debian la configurazione delle interfacce di rete è gestita a più alto livello rispetto a quello della coppia di comandi `ifconfig/route` (validi qualunque sia la distribuzione installata). Il sistema utilizzato prevede:

- Il file di configurazione `/etc/network/interfaces`:

```
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)

auto lo
iface lo inet loopback

# DO NOT EDIT BELOW THIS LINE
#auto eth0
iface eth0 inet static
  address 192.168.1.1
  netmask 255.255.255.0
  broadcast 192.168.1.255
  gateway 192.168.1.254
...
```

  a parte la riga di commento, iniziata come al solito dal carattere `#`, le prime due righe configurano l'interfaccia di loopback e, inoltre, con `auto lo` si ordina di abilitare in automatico (negli script di avvio) l'interfaccia.

  Il successivo blocco configura la scheda `eth0` con un indirizzamento statico: l'indirizzo assegnato alla scheda sarà sempre lo stesso. La scheda non è abilitata all'accensione della macchina. Talvolta nelle reti è installato un server ***DHCP*** (Dynamic Host Configuration Protocol) che configura il nodo assegnando un indirizzo dinamico diverso di volta in volta. L'interfaccia quando si attiva, per prima cosa, invia un pacchetto per vedere se c'è un server DHCP che può configurarla, in caso contrario, si configura in base all'indirizzo statico definito. Per quanto riguarda il gateway, per esempio, potrebbero esserci due schede: una configurata per interrogare un server DHCP, l'altra per comunicare con la LAN. La configurazione delle interfacce potrebbe essere:

```
iface eth0 inet dhcp

iface eth1 inet static
  address 192.168.1.254
  netmask 255.255.255.0
```

- Due comandi `ifup` e `ifdown` per abilitare o disabilitare le interfacce:

```
# ifup eth0
# route -n
Kernel IP routing table
```

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0        0 eth0
```

Lo script di inizio `/etc/rcS.d/S40networking` avvia le interfacce definite nelle righe che cominciano con `auto` (come minimo deve esserci l'interfaccia di loopback `lo`) del file `/etc/network/interfaces`. Le altre interfacce, non avviate in automatico, si avviano o fermano con `ifup/ifdown` seguiti dal nome dell'interfaccia.

La risoluzione dei nomi

Utilizzare un IP per comunicare con un nodo non è comodo per gli utenti per cui, fin dall'inizio, si sono studiati sistemi per permettere di utilizzare nomi per individuare un nodo di una rete. Ovviamente, dal punto di vista del computer, si deve ragionare solo in termini di IP. Si è reso necessario, quindi, costruire un sistema che, partendo da un nome specificato dall'utente, possa permettere di risalire, attraverso un processo qualsiasi, all'IP ad esso associato: questo processo viene chiamato risoluzione dei nomi.

Inizialmente, anche in Internet, per la risoluzione dei nomi veniva utilizzato il file `/etc/hosts` che necessita però di essere ricopiato in tutti i nodi della rete. Finché la rete è di piccole dimensioni, questa soluzione può essere accettabile, ma nella misura in cui la rete assume dimensioni come quelle di Internet attuale, si ricorre ad un sistema diverso chiamato ***DNS*** (Domain Name System) gestito da server DNS (stavolta S sta per Server) che forniscono il servizio. Quando l'utente, per mezzo di un applicativo, specifica un nome, viene inviata al server DNS una interrogazione per conoscere l'IP del nodo associato a quel nome, ricevuta tale informazione, si può contattare la destinazione.

Il primo problema da risolvere è stabilire in che modo deve essere effettuata la risoluzione dei nomi:

```
$ more /etc/host.conf
order hosts,bind
multi on
```

La prima riga specifica l'ordine dei servizi per mezzo dei quali effettuare la risoluzione dei nomi. Nel caso specifico prima si interroga il file `/etc/hosts` e dopo si interroga il servizio DNS. La seconda riga specifica semplicemente che possono esserci più indirizzi IP associati allo stesso nome e viceversa.

```
$ more /etc/hosts
127.0.0.1       localhost.localdomain  localhost

# Indirizzi IPv4

192.168.1.1     pc1.rete        pc1
192.168.1.2     pc2.rete        pc2
...
192.168.1.254   server.rete     server

# The following lines are desirable for IPv6 capable hosts
# (added automatically by netbase upgrade)

::1     ip6-localhost ip6-loopback
...
```

Ogni riga del file, a parte le righe vuote e le onnipresenti righe di commento, conserva l'associazione fra un IP e i nomi con i quali si vuole identificare il nodo (nome di dominio completo seguito da abbreviazioni).

Replicando il file `/etc/hosts` in ogni nodo della LAN, si possono contattare i nodi specificandone i nomi, oltre che gli IP. Per esempio il comando `ping`, digitando il nome al posto dell'IP, può essere utilizzato per verificare la correttezza della risoluzione dei nomi.

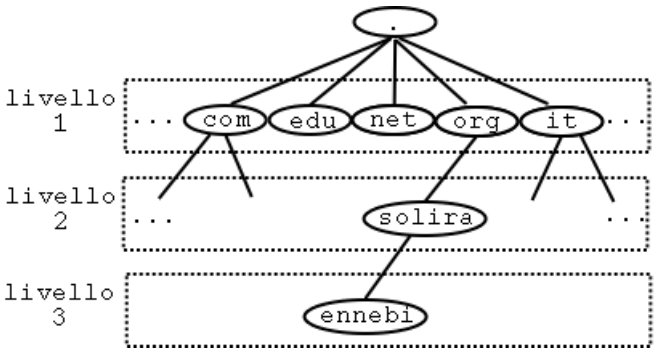
Se la risoluzione dei nomi è affidata anche a server DNS, allora è necessario predisporre il file `/etc/resolv.conf`:

```
$ cat /etc/resolv.conf
nameserver 208.67.222.222
nameserver 208.67.220.220
```

ogni riga del file che comincia con l'opzione `nameserver`, specifica l'IP dell'host da interrogare per risolvere un nome.

DNS database gerarchico e distribuito

La gestione di un insieme di nomi di notevole dimensione e in continua modifica, come è quello associato agli host di Internet, presenta problemi non indifferenti.



Il sistema DNS si basa sul ***nome di dominio*** che è la forma con cui si rappresenta, in maniera strutturata, un indirizzo. Internet è stata suddivisa in oltre 200 domini secondo un criterio che copre nomi generici (`com` per i siti commerciali, `edu` per le istituzioni educative, `org` per le organizzazioni senza scopo di lucro, `net` per i provider, ...) e nazioni (`it` per l'Italia, `uk` per l'Inghilterra, `us` Stati Uniti, ...).

I nomi di dominio sono rappresentati in una struttura ad albero dove la radice (dominio principale) è identificata da un punto, o, più spesso, sottintesa. Ogni nodo è un dominio, raggiungibile a partire dalla radice, che si rappresenta leggendolo in modo inverso. La lunghezza di un nome di dominio (quantità di nodi che si devono attraversare) si esprime in ***livelli***. Es: `ennebi.solira.org` è un dominio di terzo livello.

In linea teorica l'intero database dei domini potrebbe essere gestito da un singolo server, ma, in pratica, detto server dovrebbe rispondere a tutte le interrogazioni con un evidente sovraccarico e inefficienza del sistema stesso. Nella realtà lo spazio dei nomi è diviso in ***zone***, contenenti una parte della struttura, e a cui viene demandata la competenza per quel ramo.

I nomi di dominio si richiedono rivolgendosi ad una ***autorità di registrazione*** (RA Registration

Authority). Per l'Italia l'autorità di registrazione è *http://www.nic.it* a cui ci si rivolge per una registrazione di un dominio di secondo livello del tipo `miaditta.it`.

Il dominio di primo livello, per esempio `it`, è chiamato ***TLD*** (Top Level Domain).

L'utente inserisce in un'applicazione un nome di dominio; sarà l'applicazione utilizzata che inoltrerà una richiesta, ad uno dei server specificati in `/etc/resolv.conf`, per ottenere l'IP corrispondente e quindi poter comunicare con il destinatario.

Si può interrogare un server DNS, direttamente, utilizzando appositi comandi:

```
$ host google.it
google.it               A       216.239.59.104
google.it               A       216.239.39.104
google.it               A       216.239.57.104
$ host 64.236.24.20
Name: www5.cnn.com
Address: 64.236.24.20
```

al comando `host` si può passare come parametro un nome di dominio e, in questo caso, si avrà come ritorno il o gli indirizzi IP a cui corrisponde quel nome, oppure un indirizzo IP, per conoscere il nome di dominio associato.

Il comando, se non specificato altrimenti nell'invocazione stessa, interroga i soliti server DNS specificati in `/etc/resolv.conf`.

```
$ whois solira.org
...
Domain Name:SOLIRA.ORG
Created On:09-Nov-2003 22:39:43 UTC
Last Updated On:28-Oct-2005 08:28:05 UTC
Expiration Date:09-Nov-2006 22:39:43 UTC
...
Registrant Name:Associazione Software Libero Ragusa
Registrant Organization:Associazione Software Libero Ragusa
...
```

`whois` restituisce, invece, informazioni sull'utente che ha registrato il dominio.

TCP: socket, porte e port mapper

Il protocollo IP si occupa della trasmissione di pacchetti di dati fra un nodo e l'altro di una rete. Tuttavia quando due stazioni vogliono instaurare una connessione, si richiede che questa avvenga, per esempio, fra un processo che richiede un determinato servizio (un client) e un processo che è in grado di fornirlo (un server) e quindi la trasmissione deve poter specificare a quale processo è rivolta la connessione: in uno stesso host possono essere attivi più servizi. Di questo si occupa il protocollo TCP, che si pone al livello di Trasporto (4) di ISO/OSI, e il cui scopo è quello di fornire una connessione affidabile fra due nodi. TCP si pone quindi ad un livello superiore rispetto a IP: il pacchetto TCP viene passato al livello inferiore dove viene *inbustato* in un pacchetto IP e inviato sulla rete.

Il concetto fondamentale su cui si basa TCP è quello del ***socket*** (presa). Un socket è l'insieme formato da un indirizzo IP e un numero di ***porta***: il punto in cui un processo è in ascolto per una determinata connessione. Un socket può gestire più connessioni contemporaneamente. Un processo apre un socket in ascolto di connessioni ad esso, un secondo processo che vuole comunicare col

primo, apre anch'esso un socket e poi chiama il primo processo; chiaramente per poter comunicare con il primo processo deve conoscere la porta su cui questo è in ascolto.

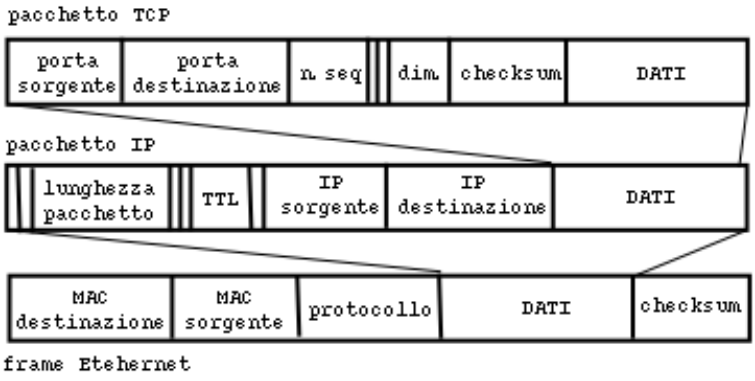
I numeri di porta inferiori a 1024, sono chiamati ***well-known number ports*** (numeri di porta ben noti) e identificano i servizi standard elencati nel sito *www.iana,org* (Internet Assigned Numbers Authority). La lista è riportata nel file `/etc/services`:

```
$ less /etc/services
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, officially ports have two entries
# even if the protocol doesn't support UDP operations.
...
smtp            25/tcp          mail
...
www             80/tcp          http            # WorldWideWeb HTTP
www             80/udp                          # HyperText Transfer Protocol
...
pop3            110/tcp         pop-3           # POP version 3
pop3            110/udp         pop-3
...
```

nelle righe riportate sono evidenziate le porte su cui sono in ascolto i servizi per la gestione di pagine web e della posta elettronica in entrata (pop3) e in uscita (smtp).

Le porte di numero superiore (fino a quella di numero massimo 65535) sono riservate ai socket della comunicazione dopo l'avvenuta connessione.

Stabilita la connessione, il mittente, invia un pacchetto al destinatario e, nel frattempo, avvia un timer. Il destinatario, all'arrivo del pacchetto, invia al mittente un pacchetto di *acknowledgement* (riconoscimento) per il pacchetto successivo. Se il pacchetto non dovesse arrivare al mittente prima della scadenza del timer, il mittente invia nuovamente il pacchetto.



Il pacchetto formato dal TCP, oltre che dai dati, è costituito anche da una intestazione di 20 byte che contiene:

- Numeri di porta sorgente e porta destinazione che identificano i due estremi del segmento di comunicazione.
- Numero di sequenza del pacchetto e numero di riconoscimento. I dati da trasmettere sono infatti frammentati in più pacchetti e, questo campo, viene utilizzato per ricostruire i dati di origine. Le funzioni del numero di riconoscimento sono già state evidenziate.

- Caratteri di controllo vari per la gestione della comunicazione.
- Dimensione dei dati che il ricevente è pronto ad accettare.
- Checksum per il controllo della correttezza dei dati stessi.

Un modo di controllare i servizi disponibili in un host, e le porte in cui sono in ascolto, è quello di usare un *port mapper*, per esempio ***Nmap*** (Network Mapper) uno dei tools fondamentali utilizzati da amministratori di reti e da chi si occupa di sicurezza:

```
# nmap -sV 127.0.0.1

Starting nmap 3.48 ( http://www.insecure.org/nmap/ ) at 2006-03-08 12:42 CET
Interesting ports on mepis1 (127.0.0.1):
(The 1654 ports scanned but not shown below are in state: closed)
PORT    STATE SERVICE VERSION
80/tcp  open  http    Apache httpd 1.3.27 ((Unix) Debian GNU/Linux PHP/4.1.2)
515/tcp open  printer
631/tcp open  ipp     CUPS 1.1

Nmap run completed -- 1 IP address (1 host up) scanned in 6.213 seconds
```

nell'esempio proposto si effettua una scansione della macchina locale specificando, con l'opzione `-sV`, la volontà di avere anche informazioni sulla versione del software che fornisce il servizio. Il risultato dell'esempio mostra che c'è, fra gli altri, il web server Apache in ascolto sulla porta 80.

Nmap può essere utilizzato anche per effettuare una scansione di una rete, effettuando ping multipli alla ricerca di host attivi (`nmap -sP 192.168.1.*`) o per tentare di rilevare il sistema operativo che gira in un host (`nmap -O 192.168.1.2`).

Protocolli livello Applicazione: HTTP e Telnet

Uno dei servizi di rete che si sono maggiormente diffusi, a partire da Internet, è senza dubbio il Web. Le tecnologie legate ad esso hanno avuto negli anni una evoluzione notevole che si è dimostrata *invasiva* anche in campi che prima si pensava impermeabili a tali tecnologie: si fa riferimento, principalmente, al cosiddetto web programming nella cui definizione si comprende un insieme di tecnologie che consentono di sviluppare applicativi che girano all'interno delle pagine web. La facilità con cui si possono sviluppare applicazioni e la potenza disponibile hanno fatto sì che è sempre più frequente trovare applicazioni che utilizzano tali tecnologie anche fuori da Internet, per esempio, nelle reti locali. È questo il motivo della scelta di trattare in dettaglio, in questi appunti, il funzionamento del www in generale e del protocollo HTTP che ne sta alla base, tralasciando altri protocolli anche enormemente diffusi e utilizzati (per esempio POP3 o SMTP).

Dal punto di vista dell'utente, il Web è un insieme di documenti (le ***pagine web***), prodotte in un determinato formato (HTML, Hyper Text Markup Language) e disponibili in host sparsi in tutto il mondo. Le pagine sono visualizzate utilizzando un programma chiamato ***browser*** e contengono riferimenti (***collegamenti ipertestuali***) che, se attivati, consentono la visualizzazione della pagina collegata. Il browser rileva, per esempio, il clic del mouse su un collegamento e per mezzo dell'***URL*** (Uniform Resource Locator) è in grado di richiedere la pagina. L'URL di una pagina assume una forma del tipo *http://ennebi.solira.org/index.htm* dove *http* specifica il protocollo da utilizzare e il seguito indica la pagina richiesta con il nome di dominio e quello della pagina specifica.

Il browser (client) apre un socket su una porta non privilegiata e tenta una connessione TCP ad un determinato host, specificando l'IP e la porta 80 di ascolto del server web. Il server web accetta la connessione TCP, riceve il nome del file richiesto, recupera il file, lo spedisce al client che ne ha fatto richiesta e chiude la connessione.

Quello esposto è in sintesi ***HTTP*** (Hyper Text Transfer Protocol), il protocollo di comunicazione fra il browser e il server web. Le operazioni previste dal protocollo vengono chiamate ***metodi*** e i più comuni sono il metodo ***GET*** e il metodo ***POST***.

- **Il metodo GET** è utilizzato nella richiesta delle pagine. Il traffico di rete in questo caso va dal server al client.
- **Il metodo POST** è utilizzato per accodare alla richiesta di una pagina, dati prodotti dal client. Il traffico di dati, in questo caso, va dal client al server. È utilizzato, per esempio, per aggiungere un messaggio in un newsgroup o per inviare i dati per una interrogazione ad un database.

Si propone ora un esempio di richiesta, di pagina web, inoltrata utilizzando il programma ***Telnet***. Si tratta di un programma nato per effettuare login ad una macchina remota e che può essere utilizzato per gestire *a mano* qualsiasi protocollo di livello applicazione: basta conoscerne i metodi.

Nel login ad una macchina remota, per motivi di sicurezza (visto che viaggia, in chiaro, in rete anche la password per il login stesso), sono stati introdotti successivamente protocolli che consentono la trasmissione di dati in formato crittografato, ma Telnet rimane, quando non ci sono problemi di sicurezza, un programma ancora valido.

```
$ telnet 127.0.0.1 80
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
GET /~nunzio/richiesta.html HTTP/1.1
Host: localhost

HTTP/1.1 200 OK
Date: Sun, 12 Mar 2006 10:35:26 GMT
Server: Apache/1.3.27 (Unix) Debian GNU/Linux PHP/4.1.2
Last-Modified: Fri, 10 Mar 2006 07:54:31 GMT
ETag: "604d7-1b8-441130b7"
Accept-Ranges: bytes
Content-Length: 440
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

<head>
  <title>richiesta</title>
</head>

<body>
<p align="center"><h1>Richiesta completamento nome</h1></p>
<br><br>
<FORM action="risposta.php" method="POST">
  Inserire nome: <INPUT type="text" name="nome">
        <br><br>
        <INPUT type="submit" value="Invia">
```

```
</FORM>
</body>
</html>

Connection closed by foreign host.
```

Il programma viene lanciato specificando l'IP dell'host con cui ci si vuole collegare e il numero di porta.

Subito dopo la risposta di connessione effettuata da parte del server, si specificano le richieste:

```
GET /~nunzio/richiesta.html HTTP/1.1
Host: localhost
```

la richiesta, per mezzo del metodo GET, specifica il nome della pagina e la versione del protocollo da utilizzare (nell'esempio la 1.1). Nelle righe successive si possono inserire altre richieste: per esempio di invio di immagini da visualizzare nella pagina. In ogni caso è necessario specificare l'host. La sequenza di righe di richieste termina con una riga vuota. Dopo l'invio della pagina richiesta, l'host, chiude la connessione.

Si sarebbe potuto specificare, dopo GET, il nome della sola pagina seguito da *Invio*. In questo caso si sarebbe avuto la risposta immediata dal server. Si tratta di HTTP/1.0 (la prima versione del protocollo) in cui, per ogni singola richiesta, si instaura una connessione. Questo modo di procedere può andare bene se la pagina è costituita da solo testo. In presenza di immagini si sarebbe dovuto effettuare una connessione per ogni richiesta di immagine contenuta nella pagina. Se si adopera invece HTTP/1.1 si possono effettuare più richieste comprese in una singola connessione (***connessioni persistenti***): è questo il modo di operare corrente.

La risposta del server comincia con un codice, nel caso specifico 200, che indica lo stato della risposta e che precede l'intestazione contente informazioni varie sul server e la pagina. Dopo una riga vuota comincia la pagina vera e propria.

Il codice 200 indica che la richiesta è stata eseguita con successo. I codici che iniziano con 3 indicano pagine spostate, quelli che iniziano con 4 indicano errori nel client (400 pagina vietata o 404 non trovata), quelli che iniziano con 5 indicano errori nel server.

Il dialogo esposto nell'esempio è sostanzialmente quello che si instaura tra il browser e il web server. Il browser, che agisce al livello Applicazione del TCP/IP, da una parte presenta all'utente una interfaccia che gli consente di introdurre facilmente la richiesta della pagina html, dall'altra *traduce* detta richiesta, utilizzando i metodi del protocollo al proprio livello (HTTP), in modo da essere comprensibile al web server. Il browser si comporta, in sostanza, da **front-end** per l'inserimento delle richieste effettive da inviare al web server (quelle espresse nell'esempio di uso di Telnet).

Wireshark e il traffico di rete

Per poter osservare lo scambio di pacchetti fra un client e un server web, si può utilizzare un programma di scansione del traffico di rete, in gergo uno ***sniffer***. In questi appunti si farà riferimento a ***Wireshark*** che ha il pregio di mostrare in chiaro ed evidenziare le parti che compongono il frame.

Il programma utilizza una interfaccia grafica e deve essere lanciato dall'utente root per potere

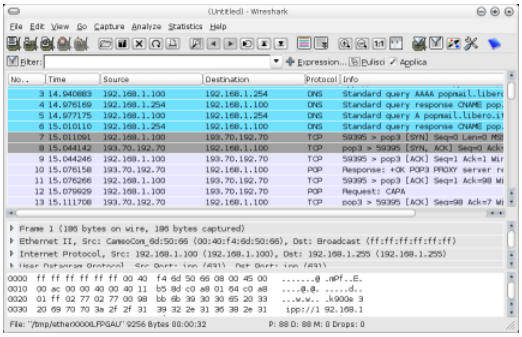
accedere alle porte e ai dispositivi di interfaccia, normalmente non accessibili da un utente con normali diritti.

La finestra dentro la quale gira il programma presenta, come solito, la barra dei menù delle opzioni disponibili e, immediatamente sotto, una barra con i pulsanti che avviano le opzioni più comuni. La prima cosa da fare è definire l'interfaccia attraverso la quale passa il traffico da esaminare. La selezione del primo, cominciando da sinistra, della barra dei pulsanti mostra una finestra di dialogo per mezzo della quale si può scegliere l'interfaccia. La stessa finestra è accessibile selezionando dal menù *Capture* la voce *Interfaces...*

La pressione del tasto *Start* corrispondente all'interfaccia desiderata (per esempio la `eth0` associata, nell'immagine riportata, all'IP 192.168.1.100) avvia la cattura dei pacchetti in transito.

La finestra di dialogo visualizzata successivamente mostra la quantità e il tipo di pacchetti in transito.

La pressione del tasto *Stop* termina l'intercettamento dei pacchetti in transito. La finestra di dialogo viene chiusa e viene mostrata l'area di lavoro di Wireshark.



L'area di lavoro del programma è suddivisa, nella sua configurazione di base, orizzontalmente in tre parti che, dall'alto verso il basso, visualizzeranno:

- **Packet List**: l'elenco dei pacchetti transitati nella interfaccia osservata.
- **Packet Details**: il dettaglio delle parti che compongono il frame. Molto utile per rintracciare e individuare le singole componenti, dei protocolli, che costituiscono il pacchetto.
- **Packet Bytes**: il contenuto del frame in esadecimale e in ASCII.

Sui pacchetti catturati possono essere applicati anche vari filtri e modalità di visualizzazione.

Wireshark: esame dei pacchetti

Al fine di evidenziare il contenuto dei frame scambiati, si propone un esempio esemplificativo.

In una rete locale (192.168.1.0) c'è un server web che gira nella macchina con IP 192.168.1.100. Il

client 192.168.1.101, per mezzo di un browser richiede le pagine web presenti nel server. Nell'esempio proposto, nel server, sono disponibili due pagine:

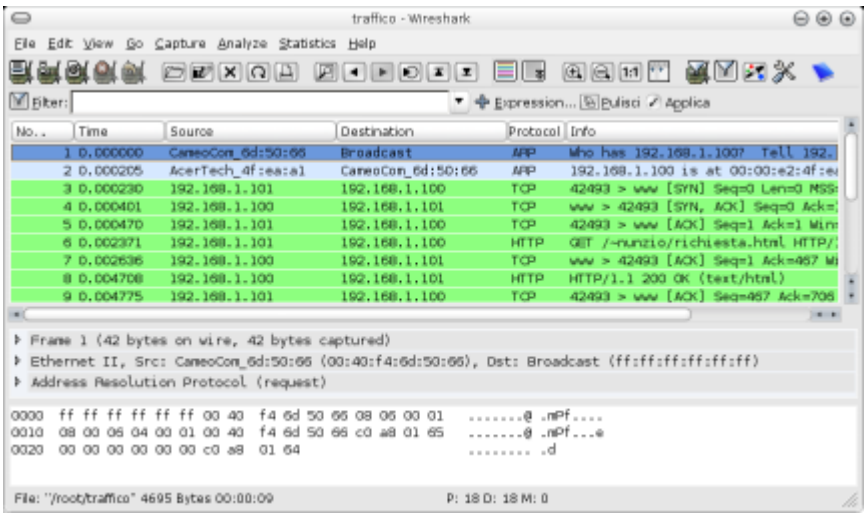
La pagina `~/nunzio/richiesta.html` contiene un form con una casella di testo, per inserire un nome, e un pulsante.

Al pulsante *Invia* è associato un metodo POST che permette l'invio del nome inserito nel form, ad una pagina php successiva che elaborerà tale dato.

La pagina `~/nunzio/risposta.php`, effettua una interrogazione ad un database, estrae il cognome e il nickname associati al nome inviato, costruisce una pagina con i dati trovati e la invia al client.

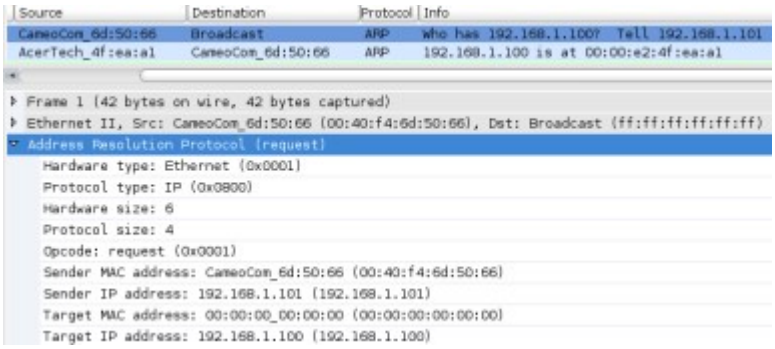
La sequenza delle azioni esposte nell'esempio è quella che avviene in un ambiente di web dinamico, quando il server web, come in questo caso, interagisce per mezzo del linguaggio di scripting PHP, con un database e costruisce la pagina da restituire al client, in base alle risposte ricevute dal server DB.

Avviata, per esempio premendo il primo a sinistra della barra dei pulsanti, la cattura dei pacchetti, viene mostrata la finestra di dialogo con le statistiche dei pacchetti e il pulsante per fermare la cattura. Dopo aver richiesto la prima pagina, inserito il nome da cercare e aver ricevuto, e visualizzato nel browser, la seconda pagina di risposta, si può fermare la cattura dei frame.



Nell'esempio riportato Wireshark ha catturato 18 frame dal dialogo fra il browser e il server web.

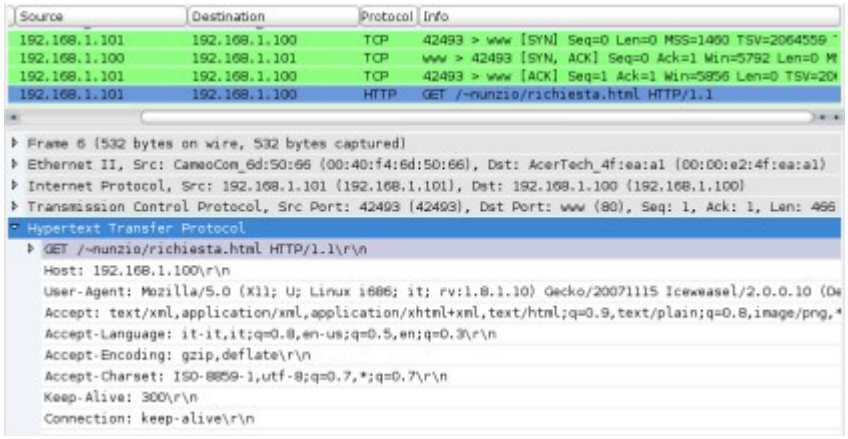
La comunicazione inizia con due pacchetti che trasportano richieste che utilizzano il protocollo ARP. Nel primo pacchetto la scheda di rete installata nel PC con IP 192.168.1.101 chiede, con un pacchetto inviato in modalità broadcasting, a che indirizzo MAC corrisponde l'IP 192.168.1.100.



Nel secondo pacchetto è contenuta la risposta dell'host, cui corrisponde l'IP specificato, che invia il proprio MAC. A questo punto la conversazione può essere avviata.

Prima di iniziare la reale conversazione fra client e server è necessario che i due interlocutori si sincronizzino.

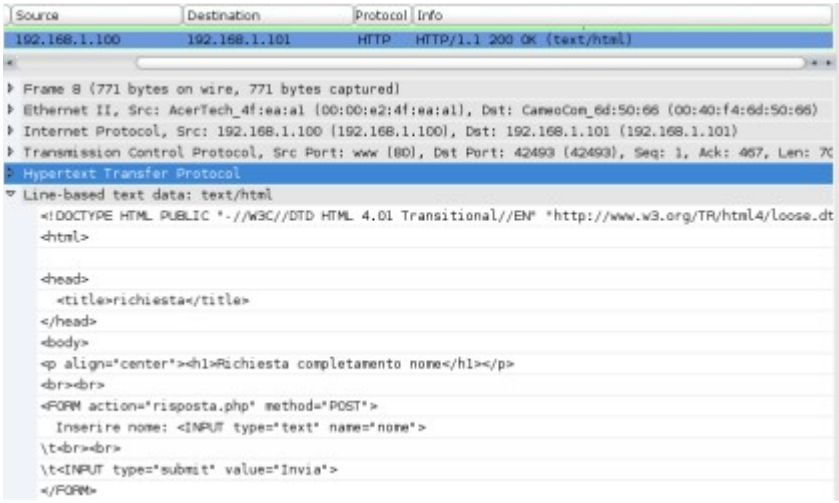
Dopo tre pacchetti di sincronizzazione (SYN del richiedente, SYN e ACK del destinatario e successivo ACK del richiedente) la conversazione può essere avviata. Un pacchetto di tipo **SYN** (SYNchronization) *chiama* l'interlocutore che, se accetta la conversazione, ritorna un pacchetto **ACK** (ACKnowledge). A questo punto anche il richiedente invia un ACK e la conversazione vera e propria può essere avviata. Quella descritta è comunemente nota come ***negoziazione dei pacchetti***. I pacchetti successivi contenenti i dati della conversazione fra i due host saranno seguiti da pacchetti di tipo ACK per attestare l'avvenuta ricezione.



Nel frame 6 viene utilizzato dal client il metodo GET per la richiesta della prima pagina. Il frame riporta nella parte riservata al protocollo HTTP, un insieme di informazioni come quelle già esaminate nell'esempio di uso di Telnet.

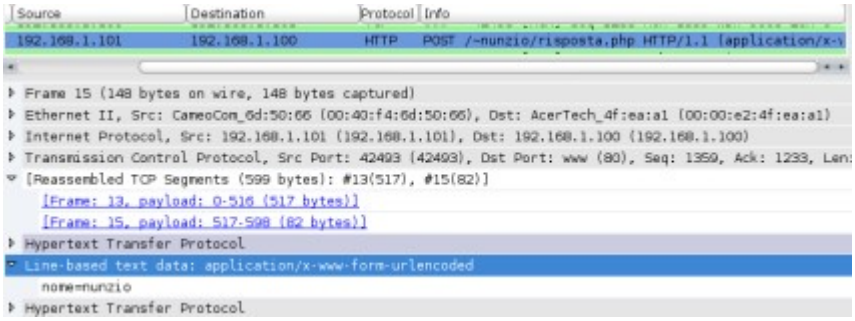
La comunicazione avviene, in questo caso, nel verso che va dalla porta, non privilegiata, 42493 aperta dal client, alla porta privilegiata 80 sulla quale risponde il server.

Il frame 8 riporta la risposta del server web: la richiesta è stata soddisfatta (la parte superiore della finestra riporta il codice 200); dopo essersi presentato (nella parte dedicata al protocollo HTTP) mostrando il proprio nome e la versione, invia in ASCII il sorgente della pagina richiesta.



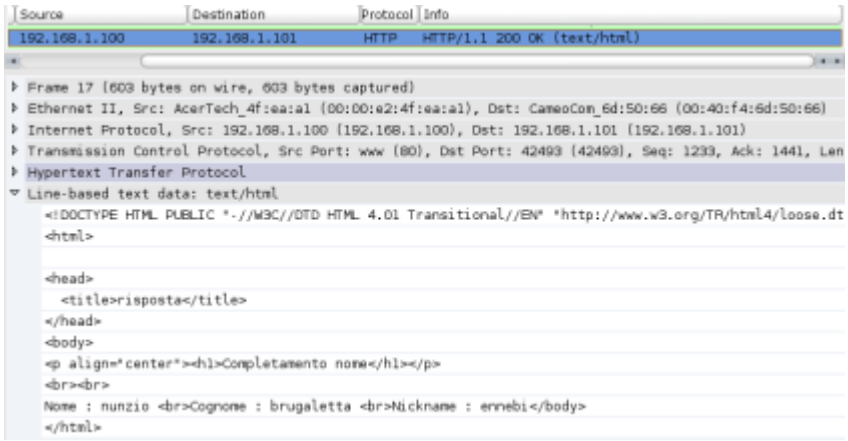
In questo caso, come evidenziato nella parte TCP, il pacchetto viaggia dalla porta 80 del web server alla porta 42493 del client.

Il comando POST per l'invio, del dato inserito, al server è suddiviso nei due frame 13 e 15.



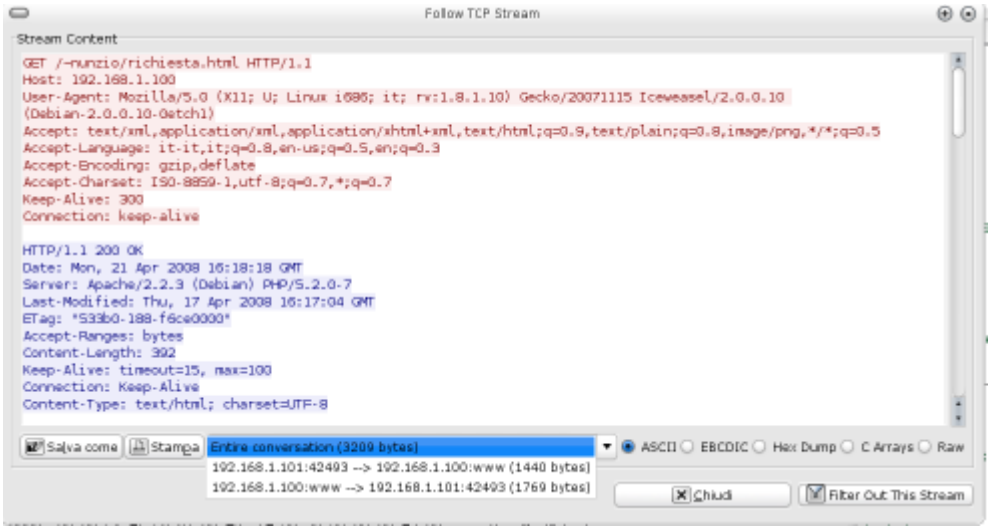
Si fa notare che accodato alla richiesta stessa si trova il nome della variabile utilizzata seguito dal valore, in chiaro, introdotto dall'utente.

Nel frame 17, infine, il server invia al client la pagina HTML con i dati richiesti.



L'interrogazione al database non genera traffico nella rete e, quindi, non ci sono frame che testimoniano lo scambio con il server DB. Nei frame c'è solo la pagina dinamica generata dall'interprete PHP.

Wireshark offre anche la possibilità, al di là della sequenza dei pacchetti, di seguire lo scambio di dati.



Selezionando uno qualsiasi dei pacchetti scambiati dagli interlocutori, scegliendo *Follow TCP Stream* dal menù *Analyze* viene visualizzata una finestra di dialogo con il contenuto della conversazione fra client e server così come, per esempio, avviene in un protocollo di alto livello (HTTP). È possibile scegliere se la visualizzazione deve riguardare l'intero traffico di pacchetti, come anche solamente quello verso una sola direzione.

Come rilevato più volte, nel protocollo HTTP i dati inseriti dagli utenti vengono trasmessi in chiaro in ASCII. È questo il motivo per cui, nel caso la procedura utilizzata richiedesse l'inserimento di dati sensibili come password o altro, sarebbe necessario utilizzare protocolli diversi rispetto ad HTTP.

Per rispondere ad esigenze di trasmissione riservata è nato ***SSL*** (Secure Socket Layer) che si occupa dell'autenticazione del client e del server e delle comunicazioni criptate per garantire la protezione dei dati. HTTP, quando usato con SSL prende il nome di HTTPS (HTTP Secure): l'uso di questo protocollo, nei browser, è evidenziato da un lucchetto chiuso visualizzato nella barra di stato della finestra del browser, e dall'URL visualizzato, nella barra dell'indirizzo, su fondo colorato.

Il file di configurazione di Apache: apache2.conf

Si è già avuto modo di osservare l'importanza e la diffusione delle tecnologie del Web anche oltre l'originario ambiente in cui sono nate. Il software che sta alla base delle applicazioni web-based è il web server. In questo paragrafo si vedranno i parametri di uso comune, utilizzati dal web server Apache (versione 2) per poter gestire un'applicazione basata sulle tecnologie web (HTML e web programming, per esempio, con PHP).

Il file di configurazione di Apache è `/etc/apache2/apache2.conf` e ogni opzione (***direttiva*** nella terminologia del file) è preceduta, come solito nei file di configurazione di Linux, da righe di commento che documentano in maniera esaustiva il significato dell'opzione: se la direttiva non è attivata, la riga è commentata; se si vuole attivare l'opzione basta togliere il commento o, viceversa, se si vuole disattivare una opzione, generalmente, non si cancella la riga ma la si commenta.

Il file è idealmente composto da 3 sezioni: *Global environment*, *Main server configuration*, *Virtual Hosts*. La configurazione in realtà è *spalmata* in una serie di file, raggruppati anche in directory, che vengono richiamati, per mezzo della direttiva `include`, nel file `apache2.conf`.

- **Section 1: Global environment**. Contiene direttive che controllano il processo Apache quando avviato:

```
StartServers 5
...
MaxClients 150
```

Un server, per soddisfare le richieste dei client, può duplicarsi in modo da rendere più efficiente il servizio stesso. La direttiva `StartServers` indica quante istanze del server vengono avviate all'inizio, la `MaxClients` indica invece il numero totale massimo di duplicazioni del processo server, ovvero, quante richieste di client il server può soddisfare. Chiaramente questi due parametri influenzano in modo notevole le prestazioni del server: più processi ci sono, più richieste si possono accogliere in contemporanea, ma più elaborazioni deve svolgere la macchina.

```
Include /etc/apache2/mods-enabled/*.load
Include /etc/apache2/mods-enabled/*.conf
```

Sono direttive che permettono il caricamento di moduli che estendono le funzionalità del server. I vari moduli installati e disponibili sono conservati nella directory `/etc/apache2/mods-available`. Nella directory `/etc/apache2/mods-enabled` si creano i link simbolici ai moduli che si vogliono caricare, con il server, e rendere disponibili.

Il link al modulo `dir.conf`, per esempio, permette di specificare quali sono i nomi di default della pagina da inviare al client in mancanza di specifica ulteriore:

```
DirectoryIndex index.html index.cgi index.pl index.php index.xhtml
```

I comandi di shell eseguiti nella directory `/etc/apache2/mods-enabled`:

```
# cd /etc/apache2/mods-enabled
# ln -s /etc/apache2/mods-available/userdir.conf userdir.conf
# ln -s /etc/apache2/mods-available/userdir.load userdir.load
# ln -s /etc/apache2/mods-available/php5.conf    php5.conf
# ln -s /etc/apache2/mods-available/php5.load    php5.load
```

generano quattro link simbolici (riferimenti ai file: esattamente come se i file fossero ricopiati nella directory) ai moduli che consentono: i primi due, rispettivamente il file di configurazione e quello contenete le direttive di caricamento del modulo stesso, per ogni utente, di gestire un sito web nella propria home e di renderlo disponibile per mezzo del server, gli ultimi due per permettere l'interazione del web server con l'interprete php.

Il file `userdir.conf` contiene righe che permettono all'utente di specificare la directory in cui inserire le proprie pagine web:

```
UserDir public_html
...
<Directory /home/*/public_html>
```

In questo modo, per esempio, l'utente tux può inserire le proprie pagine nella directory `/home/tux/public_html` e, se presente la home page con nome `index.html`, questa sarà inviata al client che effettua una richiesta del tipo `http://nomeserver/~tux/`.

Gli altri due link simbolici consentono di utilizzare nelle proprie pagine il linguaggio di scripting PHP caricando il modulo per far interagire il web server con l'interprete PHP e consentendo la gestione, da parte del web server stesso, delle pagine di tipo `.php`.

- **Section 2: 'Main' server configuration**. Con la direttiva:

```
Include /etc/apache2/ports.conf
```

si include, nella configurazione, il file in cui si definiscono le porte su cui è in ascolto il server web.

```
# cd /etc/apache2
# cat ports.conf
Listen 80
```

in questo caso si tratta della porta standard 80.

```
User www-data
Group www-data
```

Indicano il nome dell'utente, e del gruppo cui appartiene l'utente, che lancerà il processo server. Questa è una prassi comune nei server: al momento dell'installazione viene generato un utente e un gruppo cui appartiene l'utente, con diritti limitati alla singola azione di lancio del servizio associato. In questo modo, a meno di bug del programma, un eventuale malintenzionato che riuscisse a bloccare il server potrebbe agire solo come utente con limitati permessi.

- **Section 3: Virtual Hosts**. La directory `/etc/apache2/sites-enabled` conserva le configurazioni dei nomi di dominio, e delle directory corrispondenti, accessibili nella macchina nella quale è in esecuzione il server.

```
Include /etc/apache2/sites-enabled/
```

con la direttiva si includono nella configurazione del server, i file che specificano le configurazioni dei singoli domini accessibili. Il link a `/etc/apache2/sites-available/default` specifica le caratteristiche del sito di default:

```
DocumentRoot /var/www/
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>
<Directory /var/www/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
</Directory>
```

La `DocumentRoot` specifica la directory che contiene le pagine del sito. Racchiuso dai tag, in stile HTML, `<Directory /var/www>` sono specificate le caratteristiche della directory. In modo particolare la `Options` specifica la possibilità di visualizzare l'elenco dei file contenuti se non c'è index.html (`Indexes`), o la possibilità di utilizzare link simbolici (`FollowSymlinks`). La `AllowOverride None` indica l'impossibilità esterna (nella directory specificata) di utilizzare file di configurazioni (`.htaccess`) che modificano queste specifiche. `Order` specifica l'ordine delle `Allow` (permessi agli IP che possono accedere al sito. In questo caso tutti) o `Deny` (elenco IP che non hanno accesso al sito).

È possibile abilitare il supporto di Apache per i Name-Based Vritual Host. Si tratta in particolare di gestire più siti web collegati ad un singolo IP. Questa esigenza si è sviluppata principalmente

negli ultimi anni: per rispondere all'esigenza di presenza in Internet, sono nate società che forniscono la possibilità di ospitare, per chi non possa provvedere in proprio, le pagine web di diverse aziende in un unica macchina fisica. Se si vogliono gestire più domini sulla stessa macchina basta creare un nuovo file, per esempio `mieisiti`, con le proprie configurazioni in `/etc/apache2/sites-available` e creare un link simbolico ad esso nella `/etc/apache2/sites-enabled`. Il file `mieisiti` potrebbe, per esempio e nell'ipotesi che la macchina con IP 192.168.1.100 sia quella che deve ospitare più host virtuali, contenere:

```
NameVirtualHost 192.168.1.100

<VirtualHost 192.168.1.100>
     ServerName prove.php
     DocumentRoot /home/tux/public_html/prove
</VirtualHost>

<VirtualHost 192.168.1.100>
     ServerName principale
     DocumentRoot /var/www
</VirtualHost>
```

Nell'esempio proposto sono specificati due host virtuali raggiungibili, rispettivamente, con gli URL `http://prove.php` e `http://principale`. Nei tag relativi ai due host è specificato il nome e la `DocumentRoot`. Per poter risolvere i nomi è necessario che questi siano specificati, per esempio, in `/etc/hosts`:

```
####### /etc/hosts ######
192.168.1.100  principale  prove.php
```

Il traffico di rete: firewall, IPTables e Shorewall

Quando un server rende disponibili dei servizi deve poter permettere l'accesso di questi ai client della rete, ma questi ultimi non devono, per motivi di sicurezza, poter avere accesso a porte diverse da quelle permesse. Un ***firewall*** è un filtro che permette di accettare o rifiutare il traffico di rete in entrata, ma anche in uscita, in un computer o in una rete. Quando, per esempio, si usa il comando ping per controllare se un host è raggiungibile specificandone l'indirizzo, la mancata risposta da parte del destinatario dei pacchetti, può voler dire che l'host non è raggiungibile, ma anche che l'host non risponde ai pacchetti ping. Il firewall potrebbe, infatti, non permettere questo tipo di comunicazione.

Il filtraggio dei pacchetti in transito solitamente si fa sulle macchine che funzionano da gateway fra una rete e l'altra.

Il firewall di Linux è basato su ***Netfilter*** che è uno strato software a livello kernel e l'opera di filtro sui pacchetti è fatta in base a regole che si compongono in liste chiamate ***catene***. Esistono tre catene definite: INPUT (per i pacchetti in entrata), OUTPUT (per i pacchetti in uscita), FORWARD (per i pacchetti che transitano per il computer ma non sono destinati a questo e devono essere instradati). Le catene di regole si impostano usando il comando ***iptables***, con cui si può impostare la catena a cui applicare la regola, il tipo di protocollo, la porta, l'interfaccia e come trattare il pacchetto.

Impostare le regole con iptables non è sicuramente una operazione semplicissima, visto le possibilità che offre il programma e la quantità di protocolli e possibilità. È possibile utilizzare parecchi ***front end***, interfacce, verso iptables che permettono settaggi semplificati delle regole.

Esistono front end grafici per semplici configurazioni e altri che permettono personalizzazioni maggiori. In questi appunti verrà trattato **Shorewall**. Si tratta di un sistema che si occupa, utilizzando una serie di file di configurazione, di impostare le regole per Netfilter. Nonostante non sia dotato di una interfaccia grafica, i file di configurazione da utilizzare sono abbondantemente autodocumentati e, inoltre, per ognuno esiste una pagina del manuale on-line, visualizzabile tramite il comando `man`, che spiega in estremo dettaglio tutte le impostazioni.

L'installazione del software genera, oltre all'eseguibile, le directory:

- `/etc/shorewall` In questa directory si copiano i file con le impostazioni da passare all'eseguibile shorewall che si occupa di impostare le regole per Netfilter.
- `/usr/share/doc/shorewall` o `/usr/share/doc/shorewall-common`. In questa directory sono contenute le sottodirectory `default-config`, con i file di configurazione pronti da personalizzare e copiare in `/etc/shorewall`, e `examples` con esempi di configurazione già pronti da usare.
- `/usr/share/shorewall` In questa directory sono conservati una serie di file contenenti impostazioni già pronte per essere utilizzate nei file di configurazione e riguardati la maggior parte di servizi cui si può voler accedere (http, smtp, pop, ftp ecc...).

In definitiva la possibilità di utilizzare file di configurazione già predisposti per le varie combinazioni, la presenza di abbondanti commenti nei vari file di configurazione stessi, la disponibilità di pagine di manuale per tutte le operazioni possibili e il fatto stesso di distribuire le impostazioni in vari file, offrono un ambiente altamente personalizzabile e, nonostante l'assenza di interfacce grafiche, intuitivo e semplice da utilizzare.

Configurazione di Shorewall

Come esempio di configurazione del firewall si farà riferimento al seguente scenario: la macchina, con IP 192.168.1.100, sulla quale impostare il firewall mette a disposizione degli altri host della rete locale, cui è connessa per mezzo dell'interfaccia `eth1`, una applicazione web gestita dal web server. Inoltre la macchina, attraverso l'interfaccia `eth0` accede, per mezzo di un router, ad Internet.

Come accennato prima le operazioni da compiere sono il prelievo degli opportuni file da `/usr/share/doc/shorewall/default-config`, la loro personalizzazione e la copia, del file personalizzato, nella directory di configurazione `/etc/shorewall`.

- **`zones`**. La prima cosa da fare è definire le zone in modo tale da fare riferimento ad esse per stabilire le possibilità di comunicazione fra di esse. Per zona si intende un insieme di computer, definiti per mezzo degli IP o dei nomi, a cui applicare le stesse regole. Caricando in un editor il file `/usr/share/doc/shorewall/default-config/zones`, si possono aggiungere alla fine le personalizzazioni:

```
################################################################
#ZONE        TYPE     OPTIONS     IN        OUT
#                                 OPTIONS   OPTIONS
fw          firewall
net         ipv4
loc         ipv4
#LAST LINE - ADD YOUR ENTRIES ABOVE THIS ONE - DO NOT REMOVE
```

  oltre alla zona `fw` configurata di tipo `firewall`, e che rappresenta la macchina stessa nella quale

configurare il firewall, sono state aggiunte le zone: `loc` per indicare la rete locale, verso la quale verrà fornito il servizio web, e `net` per Internet. Saranno, quindi, impostate regole diverse per gli host della rete locale e per gli altri host della rete esterna Internet.

Nel file di configurazione delle zone, come negli altri file, sono presenti molte righe di commento con la spiegazione delle varie opzioni che possono essere utilizzate. Nel caso specifico i nomi delle zone possono essere a scelta, escludendo `all` (indica nei file di configurazione tutte le zone) e `none` (indica nessuna zona) che hanno significati particolari. Il tipo `firewall` identifica il firewall stesso e `ipv4` indica il tipo generale di zona.

Le opzioni si specificano separate da spazi. L'ordine è quello evidenziato nella riga di commento. Se una opzione non è espressa, e devono essere specificate le successive, occorre inserire un carattere −.

Il file così modificato viene copiato, come tutti gli altri file di configurazione, in `/etc/shorewall`.

- **`interfaces`**. In questo file di configurazione vanno definite le corrispondenze fra le zone e le interfacce hardware verso le zone:

```
###############################################################################
#ZONE    INTERFACE    BROADCAST      OPTIONS
net      eth0         detect         dhcp
loc      eth1         192.168.1.255
#LAST LINE -- ADD YOUR ENTRIES BEFORE THIS ONE -- DO NOT REMOVE
```

  in questo caso, come specificato in precedenza, `eth0` è collegato al router, riceve l'indirizzo, che viene rilevato in automatico (`detect` nella colonna `BROADCAST`), da un server dhcp. L'interfaccia `eth1` invece collega il computer con la zona `loc` identificata dall'indirizzo di broadcast 192.168.1.255.

- **`policy`**. Nel file è indicato il comportamento predefinito del traffico per le zone interessate. I pacchetti possono essere soggetti a tre tipi di trattamenti:
  - `ACCEPT` i pacchetti vengono accettati
  - `REJECT` i pacchetti vengono rifiutati e viene notificato il rifiuto al mittente
  - `DROP` i pacchetti vengono rifiutati ma, a differenza del caso precedente, non viene inviata al mittente alcuna notifica.

  La configurazione, in conseguenza delle ipotesi enunciate, potrebbe essere:

```
#############################################################
#SOURCE   DEST    POLICY    LOG LEVEL    LIMIT:BURST
fw        net     ACCEPT
fw        loc     ACCEPT
net       fw      DROP
loc       fw      REJECT
# THE FOLLOWING POLICY MUST BE LAST
all       all     REJECT    info

#LAST LINE -- ADD YOUR ENTRIES ABOVE THIS LINE -- DO NOT REMOVE
```

  Le configurazioni esprimono il fatto che, in linea generale, il traffico che viene avviato dal firewall, sia verso la rete locale che verso Internet, è accettato e che, invece, sempre in linea

generale, il traffico in entrata verso il firewall non viene accettato.

L'ultima linea, cautelativa, imposta una politica di rifiuto generale: quanto non esplicitamente permesso, viene rifiutato.

- **`rules`**. Al di là delle impostazioni generali, riportate in `policy`, in questo file si dettagliano le regole relative al trattamento di un tipo specifico di traffico.

```
#############################################################################
#ACTION  SOURCE  DEST  PROTO  DEST    SOURCE  ORIGINAL  RATE  USER/
#                           PORT(S) PORT(S) DEST      LIMIT GROUP
#SECTION ESTABLISHED
#SECTION RELATED
SECTION NEW

ACCEPT   loc     fw    tcp   80
#LAST LINE -- ADD YOUR ENTRIES BEFORE THIS ONE -- DO NOT REMOVE
```

poiché nel computer, dove si sta configurando il firewall, è in esecuzione il web server e si vuole rendere accessibile tale servizio alla rete locale, si stabilisce la regola di accettare il traffico, di tipo TCP verso la porta 80, proveniente dalla rete locale e diretto alla zona `firewall`. In definitiva, considerando le politiche generali e l'impostazione dei pacchetti di tipo tcp qui specificate, il firewall accetta, in entrata dalla zona `loc` (la rete locale), solo il traffico di tipo tcp, rifiutando tutto il resto.

La configurazione delle regole può essere dettagliata per ogni tipo di servizio disponibile in rete. L'installazione di Shorewall crea la directory `/usr/share/shorewall` contenete una serie di file `macro.*` con le configurazioni già pronte per i servizi più comuni. Le configurazioni sono conservate come macro parametrizzate in modo da essere utilizzate specificando il tipo di trattamento e le zone interessate. Per esempio per abilitare il traffico per il server web, si possono inserire nel file le righe:

```
#############################################################################
#ACTION        SOURCE  DEST  PROTO  DEST    SOURCE  ORIGINAL  RATE  USER/
#                                   PORT(S) PORT(S) DEST      LIMIT GROUP
#SECTION ESTABLISHED
#SECTION RELATED
SECTION NEW
HTTP/ACCEPT    loc    fw
HTTPS/ACCEPT   loc    fw
```

nelle due righe specificate si stanno usando le due macro `macro.HTTP` e `macro.HTTPS` per il traffico http e https.

Gestione di Shorewall

L'avvio o il fermo del firewall, così come la verifica delle regole attive in un dato momento, possono essere monitorati per mezzo dei parametri associati al comando `shorewall`.

Comando	*Effetto prodotto*
`shorewall start`	Avvia il firewall. L'eseguibile legge le impostazioni contenute nei file di configurazione e le passa ad iptables. Da questo momento le impostazioni sono attive.

Comando	*Effetto prodotto*
`shorewall stop`	Ferma il firewall, disabilita le impostazioni.
`shorewall restart`	Riavvia il firewall in modo da rendere attive le nuove impostazioni se i file di configurazione sono modificati,
`shorewall clear`	Rimuove le regole impostate dal firewall
`Shorewall show`	Mostra le regole attive.

Se si desidera che le impostazioni siano attivate ogni volta che si accende il computer è necessario modificare il file `/etc/default/shorewall` in modo che sia impostato:

```
startup=1
```

Appendice

Sorgenti dei file utilizzati negli esempi relativi al protocollo HTTP:

richiesta.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

<head>
  <title>richiesta</title>
</head>

<body>
<p align="center"><h1>Richiesta completamento nome</h1></p>
<br><br>
<FORM action="risposta.php" method="POST">
  Inserire nome: <INPUT type="text" name="nome">
  <br><br>
  <INPUT type="submit" value="Invia">
</FORM>
</body>

</html>
```

risposta.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

<head>
  <title>risposta</title>
</head>

<body>
<p align="center"><h1>Completamento nome</h1></p>
<br><br>
<?php
  $connessione=mysql_connect("localhost","root","");
  mysql_select_db("prove");
  $q  ="SELECT utenti.cognome,utenti.nick ";
  $q .="FROM utenti ";
  $q .="WHERE utenti.nome = '$_POST[nome]'";
  $recset = mysql_query($q);
  $tr = mysql_fetch_array($recset);
  echo "Nome : $_POST[nome] <br>";
  echo "Cognome : $tr[cognome] <br>";
  echo "Nickname : $tr[nick]";
?>
</body>

</html>
```

Riferimenti bibliografici

Per la realizzazione di questi appunti sono state consultate diverse fonti:

- I libri:
  - *Reti di calcolatori* di Andrew S. Tanenbaum. Il punto di riferimento assoluto sulla teoria e il funzionamento delle reti.
  - *Appunti di Informatica Libera*, l'*opera omnia* di Daniele Giacomini. Testo da cui non si può assolutamente prescindere se si parla di Linux. Disponibile in download presso *http://a2.swlibero.org* a cui si può fare riferimento per la versione più aggiornata.
  - *La guida Debian* di Osamu Aoki. Le specifiche e le personalizzazioni della distribuzione Debian e di tutte le distribuzioni Debian-based. Disponibile in download da *http://qref.sourceforge.net*.
  - *Lnux e la sicurezza* di Roberto Butti. Uno sguardo generale agli strumenti dedicati alla protezione di un sistema.
- le pagine `man` dei comandi. La prima fonte di autodocumentazione sull'uso dei comandi.