DA BBS A POP INTERNET: L'EVOLUZIONE POSSIBILE

Lime Pluto Meeting - Roma, 7-9 Ottobre 1998

Alessandro Falaschi - Associazione Culturale IperAudio - alef@ddns.org

INDICE

1. INTRODUZIONE

Le BBS (*Bulletin Board Systems*) hanno costituito negli ultimi venti anni il simbolo della telematica "fatta in casa" e per questo libera da condizionamenti commerciali e di altra natura. E' sufficiente configurare alcuni software sul proprio Pc, per consentire ad altri l'accesso via modem, scambiarsi messaggi, condividere files; ognuno di questi nodi fa parte di una rete (*Fidonet*) che, mediante periodici collegamenti punto-punto su linea commutata, diffonde i messaggi su scala mondiale. Oggi, questo libero universo di comunicazione elettronica ed autogestita si e' quasi del tutto estinto, schiacciato dalla evidente superiorita di internet; e sembra che l'unico mediatore delle proprie comunicazioni debba necessariamente essere un provider. Questo articolo mostra come invece, grazie alla completa

dotazione di software per *Linux*, ed all'uso oculato di alcune risorse di rete, ognuno puo' configurare il proprio computer come quello di un piccolo provider, e replicare lo spirito delle BBS adottando protocolli TCP/IP di diffusione molto piu' ampia. Tutte le interazioni tra utenti e POP (*Point of Presence*) possono avvenire mediante l'interfaccia grafica costituita dal browser internet.

2. Requisiti

Per realizzare un POP Internet, occorre chiaramente disporre di una macchina Linux su cui provvediamo a montare un modem (o due nel caso si desideri offrire possibilita' di navigazione). Nel corso dell'articolo tratteremo dei problemi di configurazione e di interdipendenza di alcuni software necessari alla realizzazione del POP, senza pretese di completezza, rimandando in ogni caso agli How-To ed alle FAQ esistenti per tutte le problematiche trattate, nonche' a qualche buon libro come ad esempio [1]. Ci riferiremo quindi a *named, pppd, mgetty, apache, ftpd, innd, pop3d, sendmail, majordomo, squid, ipfwadmin*. La distribuzione utilizzata per realizzare le configurazioni descritte e' una RedHat, ma le modalita' di configurazione esposte sono di natura generale. Inoltre, gli script realizzati per espletare particolari funzioni sono reperibili presso [20].

3. Definire l'IntraNet

Una IntraNet e' una *rete locale privata* in cui sono utilizzati i protocolli di InterNet, ma che non fa parte di InterNet, ovvero esattamente il nostro caso, in cui un utente si collega al nostro PC opera come con un provider. Il primo passo e' quello di definire una serie di indirizzi IP per la nostra macchina e per chi vi si collega, da scegliere tra quelli previsti per le reti private, cosi' identificati in quanto i router impediscono la propagazione dei pacchetti da/per questi IP, permettendone il ri-uso. Un gruppo di tali indirizzi e' quello della serie 192.168.*xxx.yyy* in cui (nel caso di rete in classe C) *xxx* e' parte del numero di rete ed *yyy* e' il numero di nodo. Non illustriamo qui come assegnare l'IP alla nostra macchina, che d'ora in poi chiameremo *host*; una buona fonte di informazioni e' [2]. Occorre ora associare ad ogni IP un mome simbolico, del tipo NODO.DOMINIO: e' consigliabile in questo caso scegliere come *dominio* un nome di fantasia che non esista anche come "vero" dominio internet, in modo da rendere esplicito che si tratta di una intranet. Il campo *nodo* corrispondera' ad uno tra diversi nomi: ne occorre infatti uno per la nostra macchina, ed uno per ogni profilo di utente che vogliamo poter attribuire a chi si collega.

Le associazioni create tra IP e nodo sono utilizzate nel processo di risoluzione dei nomi, che avviene per mezzo delle risorse indicate nel file */etc/nsswitch.conf*, ed in cui e' consigliabile definire una sequenza di interrogazione del tipo files/dns/nis/, che significa cercare prima in */etc/hosts/* [2], interrogare poi il *DNS* [3] e per

ultimo provare con *NIS* [4]. L'indirizzo del DNS utilizzato e' definito nel file */etc/resolv.conf*, e puo' vantaggiosamente indicare proprio il nostro host, che offrira' il servizio per mezzo del programma *named*. In questo caso, il nostro dominio privato "esiste", purche' si inserisca la tabella delle associazioni tra nomi ed IP dell'intranet nel file di zona */var/named/named.hosts* [5], che viene letto all'avvio da named, assieme agli altri files menzionati in */etc/named.boot*. In quest'ultimo file e' poi possibile, mediante la primitiva *forwarders*, specificare un diverso DNS (tipicamente quello presente presso il provider) da interrogare per risolvere i nomi non presenti nei files della zona locale.

E' da sottolineare il fatto che la risoluzione di nomi esterni ha senso solo quando la nostra macchina e' collegata al provider; in caso contrario infatti possono verificarsi ritardi nella esecuzione di comandi dovuto all'attesa del timeout del DNS remoto che non risponde. Per risolvere questo problema si possono prevedere due diversi files *named.boot*, ed eseguire un reload di named quando ci si connette/sconnette dal provider; in questo secondo caso, oltre a non prevedere forwarders, conviene anche indicare un file di cache vuoto, in modo da evitare qualsiasi tipo di richiesta impossibile (e relativa attesa di timeout) da parte della *resolver library*. Alcune persone utilizzano unicamente il DNS del provider, e risolvono i nomi locali mediante il file /etc/hosts. Il vantaggio di usare il proprio named risiede nella sua funzione di caching, che consiste nel "ricordare" le associazioni tra i nomi e gli IP ricevuti dal forwarder, ed utilizzarle direttamente nelle successive istanze, anziche' chiederle ogni volta, velocizzando cosi' il processo di risoluzione degli indirizzi IP. Avendo definito gli IP della nostra macchina e di chi si collega, siamo in grado di controllare l'uso dei servizi presenti nella nostra macchina mediante i files */etc/hosts.allow* ed */etc/hosts.deny* [2], che consentono di specificare quali IP possono utilizzare quali programmi. Ad esempio, potremo disabilitare la possibilita' di *telnet* (e quindi l'accesso alla shell dei comandi della nostra macchina) da parte delle richieste provenienti da IP remoti: cosa assolutamente consigliabile, altrimenti si rischia di essere "invasi" durante i collegamenti ad internet. Come vedremo nel seguito, l'uso di una autenticazione PAP o CHAP permettera' di assegnare diversi IP agli utenti, a cui potranno essere attribuiti maggiori o minori privilegi di utilizzo dei servizi locali.

Per terminare questa prima sezione, si ricorda l'utilita' dei file di log, in cui sono riportati i risultati del'esecuzione delle procedure, e la cui analisi e' fondamentale per la risoluzione dei problemi: Linux conserva queste informazioni nella directory */var/log/*, in cui si trovano i files definiti in */etc/syslog.conf*; mediante quest'ultimo, e' possibile richiedere diversi livelli di dettaglio per i log prodotti da programmi differenti. Infine, con il file */etc/logrotate.conf* si definisce la frequenza con la quale i files di log sono cancellati e/o archiviati.

4. Uscire su InterNet

Il collegamento ad internet mediante il proprio provider consiste nell'esecuzione di uno script che esegue *pppd*, con le dovute opzioni, delegando ad un ulteriore programma (*chat*) il colloquio con il modem e con l'eventuale sessione di login del provider [6]. Alcune distribuzioni di Linux forniscono una interfaccia di configurazione del ppp, per cui non ci dilunghiamo su questo argomento, se non per far notare che una particolare opzione, *defaultroute*, e' quella che permette di inoltrare tutti i pacchetti per le destinazioni esterne alla nostra intranet, verso il provider.

Un passo intermedio necessario a fornire servizi di navigazione ai nostri utenti sara' quello di predisporre due semplici script, eseguibili da linea di comando, per effettuare la connessione e la disconnessione verso internet mediante comandi creati ad hoc. In alcune circostanze puo' essere utile un programma di nome *diald*, che provvede a collegarsi ad internet in modo automatico ogni volta che se ne verifichi la necessita'; la soluzione qui proposta invece comporta l'esplicita volonta' di connettersi, e come vedremo consente il controllo dell'identita' e dello status del richiedente.

Ogni volta che viene creata/terminata una connessione di rete, Linux provvede ad eseguire rispettivamente uno dei due script *ip-up* e *ip-down*, che svolgono le operazioni di manutenzione da effettuare nei due casi. E' proprio in questi due script che, come anticipato, si effettua il restart di named con uno tra due diversi file di boot.

5. Accettare Utenti

I collegamenti via modem alla nostra macchina sono permessi dall'uso di *pppd* in qualita' di *server*. Per questo utilizziamo un secondo modem [7], ed editiamo il file */etc/inittab* in modo che il processo che gestisce la porta (ad esempio *ttySx* con x = numero della porta seriale) a cui e' connesso il modem sia in grado di eseguire le operazioni necessarie. A questo scopo si possono seguire due strade.

La prima e' quella di far gestire le chiamate da *uugetty*, a cui si comunica la velocita' della porta, la locazione di un file di configurazione (tipicamente */etc/conf.uugetty.ttySx*), ed una etichetta simbolica da usare come chiave in */etc/gettydefs*, ulteriore file in cui e' specificato il programma che effettua il dialogo di autenticazione per le connessioni sulle porte con quell'etichetta. A questo punto il chiamante puo' immettere un nome di utente e relativa password, provocando l'esecuzione della shell prevista (nel file */etc/passwd*) per quell'utente . Pertanto, sara' sufficiente prevedere l'esistenza di un utente (ad esempio l'utente ppp), che abbia *pppd* come shell, per mandare in esecuzione il protocollo (con i parametri descritti in */home/ppp/.ppprc*). Dopo l'autenticazione in una finestra terminale quindi, il chiamante lancia a sua volta il ppp e si connette.

La seconda strada prevede di configurare */etc/inittab* per l'uso, anziche' di uugetty, di *mgetty* [8], che e' espressamente concepito per la gestione dei modem, e prevede molte funzioni utili: in particolare, e' in grado di rilevare la presenza di pacchetti LCP di richiesta-configurazione usati dall'accesso remoto delle macchine MS al posto del dialogo in finestra terminale. In questo caso, mgetty stesso provvede a lanciare pppd, in modo da consentire la connessione senza necessita' di una interazione (umana o tramite script). La configurazione di mgetty avviene mediante il file */etc/mgetty/login.config*, in cui si specificano i parametri (e/o il file che li descrive) con cui eseguire pppd in risposta ai pacchetti LCP suddetti; nello stesso file si definiscono le azioni da compiere nel caso in cui la connessione provenga invece da una BBS FidoNet oppure da un nodo UUCP.

Tra i parametri [9] che descrivono il comportamento del pppd in modalita' server, e' importante inserire *proxyarp*, necessario per far rispondere il nostro host, al posto dell'utente connesso, alle richieste ARP a lui indirizzate. Inoltre, e' preferibile richiedere per l'utente una autenticazione PAP, che ha luogo durante la fase di inizializzazione del PPP, e permette (in base ad una coppia utente-password memorizzata in */etc/ppp/pap-secrets*) di assegnare al chiamante un particolare indirizzo IP. In realta', in questo caso non e' esatto dire che l'IP e' assegnato dall'host; piuttosto, viene rivendicato dall'utente, a cui viene concesso in base alla sua corretta identificazione. I diversi IP richiedibili potranno quindi essere usati per controllare i diritti di esecuzione dei servizi locali mediante i files /etc/hosts.allow e .deny.

6. Pagine locali e CGI

Ora che finalmente possiamo essere raggiunti via modem, ci si puo' dedicare alla redazione di pagine html da presentare ai nostri utenti. Perche' cio' sia possibile, questi ultimi devono configurare il proprio DNS con in nostro indirizzo IP intranet, e quindi inserire l'indirizzo *nome.dominio* che abbiamo dato al nostro host nel proprio browser. Le pagine vengono inviate mediante il server HTTP, che nella grande maggioranza dei casi e' *Apache* [10], la cui configurazione omettiamo qui di descrivere.

Nelle pagine che prepareremo, oltre ad informazioni di aiuto per l'utente, e links ai servizi come FTP e *Newsgroup* che descriviamo in seguito, occorre prevedere collegamenti a pagine interattive necessarie sia per raccogliere richieste, che per fornire informazioni sullo stato dell'host. In entrambi i casi occorre disporre di programmi chiamati CGI (*Common Gateway Interface*) [11] che non sono altro che script (ad esempio in *bash, tcl* o *perl*) che vengono eseguiti da Apache in corrispondenza a determinate URL (ad esempio *http://nome.dominio/cgi-bin/eseguimi.tcl*). L'output di tali script viene quindi inviato ad Apache e da questo presentato all'utente come una normale pagina; occorre quindi che sia conforme ai formati previsti per le pagine *html*.

Il server HTTP comunica con il CGI in diverse maniere; la piu' semplice e' quella del metodo *GET* che rende visibili al CGI un insieme di variabili d'ambiente, contenenti ad esempio i dati immessi dall'utente in una form. In particolare, occorrera' prevedere un form con cui l'utente possa richiedere un alias di posta elettronica, ed una password per guadagnare l'accesso a CGI corrispondenti a caratteristiche "speciali" come la connessione al provider. I dati inseriti nella richiesta possono essere inviati per e-mail, a chi ne mantiene traccia, dal CGI che gestisce la form.

Qualora il file */etc/httpd/conf/access.conf* prescriva per le URL associate ai CGI una forma di autenticazione, il browser dell'utente presentera' allo stesso una finestra in cui si richiede di immettere nome utente e password, la cui esattezza e' verificata da Apache in base all'esame del file di password specificato in *access.conf*; in caso negativo il CGI non e' eseguito. Il formato "piu' semplice" del file di password e' del tutto simile a quello di */etc/passwd*; la sua manutenzione avverra' solamente qualora si debba aggiungere o rimuovere un utente, in base ai dati da lui stesso inseriti. Dopo aver verificato la password, Apache controlla anche che l'utente sia presente in un secondo file (indicato ancora in access.conf), che identifica chi puo' accedere alla pagina protetta. E' quindi possibile definire gruppi differenti per funzionalita' diverse.

Nel caso in cui l'autenticazione abbia successo, l'identita' del richiedente e' passata al CGI tramite una variabile di ambiente, che puo' essere usata ad esempio come chiave in una tabella di crediti attribuiti a ciascun utente, in modo da realizzare un controllo ulteriore sull'uso di servizi da considerare a pagamento, come ad esempio la richiesta di navigazione tramite il secondo modem presente. A tale scopo sara' necessario prevedere uno script di aggiornamento della tabella dei crediti, mediante il quale decrementare i crediti (per l'utente in questione) in base ad una misura di utilizzo, come ad esempio un tempo di connessione.

Altri CGI necessari potranno essere di tipo informativo, come ad esempio il report sull'ammontare dei crediti residui, lo stato della coda di posta, o lo stato del routing, utile quest'ultimo per verificare la presenza di una connessione verso il provider mediante il secondo modem. In linea di principio, il risultato di qualsiasi comando eseguibile da una finestra terminale di Linux puo' essere inviato all'utente remoto, mediante un CGI che semplicemente compone gli header html ed invoca il comando. E' da notare che tutti i CGI eseguiti da Apache sono lanciati dall'utente *nobody*, e quindi occorre prestare attenzione ai privilegi accordati a quest'ultimo. Infine, tutti gli errori che si verificano nell'esecuzione dei CGI non compaiono a video, ma possono essere rintracciati nell'ambito del file */var/log/httpd/error_log*.

7. FTP, Newsgroup ed E-Mail

In questa sezione ci occupiamo brevemente di quei servizi del tutto analoghi a quelli offerti da una BBS "tradizionale". I programmi necessari sono tutti pienamente disponibili con Linux, e sono rispettivamente *ftpd*, *innd* e *sendmail*.

Per mettere a disposizione files per il download, occorre creare le directory sotto */home/ftp/pub* come leggibili da tutti, accertarsi che *inetd.conf* faccia partire *ftpd*, e che sia definito un utente ftp con password disabilitata [2]. Puo' essere opportuno configurare */etc/ftpaccess* per limitare i rischi di sicurezza legati all'attivazione dell'FTP anonimo.

Le *newsgroup*, accessibili con i browser web o con i reader appositi, sono gestite da *innd* [12]; durante l'editing dei files di configurazione di quest'ultimo, occorre porre attenzione a mantenerne i privilegi di accesso inalterati, in quanto innd e' eseguito dall'utente *news*, e tale dovra' mantenersi la proprieta' dei file; inoltre e' necessario re-dirigere la posta per l'utente news verso il nostro acount di manutenzione, dato che e' per questa via che innd resoconta le proprie attivita'. La manutenzione delle news avviene per mezzo del programma *ctlinnd*, che prevede comandi per la creazione e cancellazione di newsgroup, di singoli messaggi, la definizione di gruppi moderati, e svariati altri comandi che interagiscono direttamente con il server. Il browser dell'utente remoto in realta' dispone unicamente del protocollo *nnrp*, e pertanto occorre definire in */etc/news/nnrp.access* quali indirizzi possano accedere (ed in che modo) ai diversi newsgroup; queste impostazioni possono avvalersi di meccanismi di wildcard, e prevedere differenti diritti di accesso per i diversi IP assegnati mediante autenticazione PAP. Altri files di configurazione per *innd* riguardano la configurazione di altri newsserver con cui scambiare news, utile anche nel caso in cui si voglia realizzare un gateway con una BBS tradizionale (ad esempio usando GIGO [13] od *ifmail* [14]); inoltre, e' presente la definizione della durata di permanenza dei messaggi nei diversi gruppi (*/etc/news/expire.ctl*) e la descrizione dei moderatori per i gruppi moderati (nel qual caso i messaggi postati vengono inviati appunto al moderatore che ha il compito di approvarli). Quest'ultima caratteristica permette la creazione, in modo piuttosto agevole, di *gateway* tra newsgroup ed indirizzi e-mail [15], ad esempio associati a Mailing Lists. In quest'ultimo caso, alla mailing list verra' iscritto un alias di email (vedi sotto) che corrisponde al programma che effettua l'operazione di approvazione.

L'uso della *email*, lasciato per ultimo di questa sezione, risulta il piu' delicato, a causa della natura stessa di Internet. A livello locale, cioe' nell'ambito dei nostri utenti, ognuno puo' configurare i server SMTP e POP3 con l'indirizzo del nostro host per inviare e prelevare la propria posta. Agendo sul file */etc/aliases* possono essere definiti ulteriori "utenti", da associare ad altri indirizzi email (locali o remoti), ad un programma (che riceve il contenuto del messaggio) od a gruppi di

indirizzi (a tutti i componenti del quale e' inviata una copia del messaggio). Anche l'invio verso l'esterno funziona perfettamente. Il programma che se ne occupa e' *sendmail* [16], in base alle indicazioni che si trovano in */etc/sendmail.cf*; anche se questo file e' solo il prodotto finale della "compilazione" di un ulteriore file di configurazione, e' perfettamente possibile modificarlo direttamente senza troppo sforzo. In particolare, e' opportuno definire l'opzione *Delivery Mode=background* per evitare i tempi di attesa dovuti ai tentativi di risoluzione di indirizzi remoti quando non si e' connessi al provider, ed un *timeout* per i warning di *eccessiva giacenza* dei messaggi nella coda di uscita, di durata superiore all'intervallo tra connessioni con il provider. Quando ci si collega ad internet (possibilmente mediante uno script che viene eseguito periodicamente mediante cron) si provvede a svuotare la coda di posta (inserendo in *ip-up* il comando *sendmail -q*); a tale riguardo puo' essere opportuno configurare come *smart relay host* (opzione *DS* di sendmail.cf) un server SMTP situato presso il provider, in modo da consegnarli tutta la posta e delegargli cosi' l'invio agli indirizzi di destinazione, evitando i ritardi dovuti a destinatari lontani, lenti, o mal configurati.

Il problema della email nasce dal fatto che ... non puo' arrivare ! Infatti, a parte il caso banale di recuperare la nostra posta personale che giace presso un provider, il *nome.dominio* che ci siamo scelti e' valido solo nella nostra intranet (e' risolto solo dal nostro DNS), e non esiste su internet, e quindi i nostri utenti che hanno la mailbox presso di noi... non esistono. Anche se nel momento in cui ci connettiamo al provider acquisiamo un IP (e relativo indirizzo) valido, con cui in linea teorica potremmo ricevere posta, quest'IP e' differente per ogni connessione (per questo si chiama *dinamico*) e dunque serve a poco. A meno che ... non si operi come descritto all'ultima sezione !

Non resta infine che segnalare la disponibilita' di un elevato numero di software per la gestione dei servizi di comunicazione elettronica, come ad esempio *Majordomo* [17], per la gestione di mailing lists.

8. Navigazione Off-Line

Questo capitolo espone una caratteristica peculiare della configurazione che andiamo proponendo, e cioe' la possibilita' di catturare pagine od interi siti da internet, e rendere le stesse disponibili alla navigazione per i nostri utenti anche senza attivare una connessione ad internet. Questo concetto amplia e potenzia il concetto di BBS: e' possibile infatti *costruire* il proprio nodo telematico utilizzando materiale gia' perfettamente redatto ed impaginato da estensori competenti e capaci.

Per realizzare la cosa occorre installare sul proprio host un *server proxy*, ad esempio *squid* [18]. La sua configurazione e' veramente molto facile, e si basa sull'editing del file *squid.conf*, in cui si assegnano ad esempio le quantita' di disco e di memoria fisica da riservare per il suo uso. Altri parametri da modificare

sono le parti di URL che individuano pagine da non memorizzare, che nel nostro caso conviene siano in numero piu' ridotto possibile; e' inoltre opportuno lanciare squid con l'opzione -D per evitare il ritardo dovuto al DNS lookup iniziale. Infine, occorre abilitare l'accesso al proxy da parte degli IP della nostra intranet.

9. Navigazione On-Line

Disponendo di due modem, mentre un utente e' connesso sul primo, il secondo puo' connettersi al provider, consentendo cosi' all'utente di navigare su internet. Dal nostro punto di vista, occorrera' addebitare all'utente quantomeno il costo della telefonata uscente, e facendo qualche conto ci si accorge che tale costo e' superiore a quello richiesto dai "normali" provider; pertanto, la funzione di navigazione e' da ritenersi accessoria, anche se sicuramente molto interessante e valida.

Il prerequisito necessario a far navigare i nostri utenti e' l'abilitazione del *Masquerading* [19], una caratteristica di Linux ottenibile specificando le opportune opzioni nella compilazione del kernel, e che effettua una conversione tra l'indirizzo IP intranet dell'utente connesso in PPP, ed i numeri di porta utilizzati nel collegamento ad internet. Infatti, come abbiamo visto, gli IP intranet non oltrepassano i router, e dunque sarebbe impossibile accedere a servizi esterni. Specificando le opportune opzioni al comando *ipfwadm*, invece, l'host che effettua il Masquerading intercetta il pacchetto dell'utente che richiede il servizio, sostituisce il proprio IP (ossia l'IP dinamico ricevuto dal provider) al posto di quello dell'utente, e lo invia al server destinatario utilizzando un socket TCP/IP non standard. I pacchetti di ritorno fanno riferimento allo stesso socket, che l'host riconosce corrispondere alla richiesta dell'utente, al quale consegna il pacchetto ricevuto dopo avere sostituito all'IP di destinazione quello dell'utente al posto del proprio (il server remoto infatti immagina di dialogare solo con il nostro host).

La richiesta di navigazione da parte dell'utente avviene mediante accesso ad un CGI (con *autentificazione*) [10], che come gia' descritto consente di mantenere aggiornata una tabella di crediti per utente. Il CGI provvede quindi a richiedere un collegamento al provider, a memorizzare l'istante di inizio della connessione e l'identita' del richiedente; ogni volta che ci si collega ad internet, *ip-up* provvede a far partire la posta giacente e riconfigura il DNS. La sconnessione dal provider puo' essere richiesta esplicitamente dall'utente, oppure prodotta automaticamente quando quest'ultimo a sua volta si scollega. In questo secondo caso, *ip-down* rivela che l'utente e' uscito e provvede a chiudere il collegamento al provider; quest'ultimo evento causa quindi un nuovo reset del DNS e l'aggiornamento dei crediti per l'utente che ha iniziato la navigazione.

L'uso del *proxy server* da parte dell'utente anche durante la navigazione On-Line, permette di ritrovare le pagine visitate anche nei successivi collegamenti, senza bisogno di connettersi ad internet. L'invio automatico della posta in uscita ad

ogni connessione al provider costituisce un elemento di cooperazione e mutualita' tra gli utenti del nostro host, in quanto ad esempio la connessione richiesta da Tizio per far partire la propria posta urgente permette la partenza anche della posta di Caio e di Sempronio.

10. Ricezione della Posta

Quasi ovunque si puo' trovare l'affermazione che per ricevere la e-mail occorre disporre di un IP statico, ovvero il nostro provider deve assegnarci sempre lo stesso IP ad ogni collegamento, a cui associare stabilmente il dominio internet, verso cui indirizzare la posta. Se l'IP invece cambia ogni volta, questa associazione non e' possibile, a meno che... non la si effettui di volta in volta ! ... peccato pero' che nessun DNS accetti questo genere di configurazioni, tranne ... un DNS scritto apposta. Ed infatti esistono delle iniziative in rete in tal senso; in questa sede prendiamo come esempio il servizio offerto da una realta' telematica con sede a Singapore, *www.ddns.org*, anche a titolo gratuito.

Il DNS dinamico (DDNS), quando e' chiamato a rispondere alle interrogazioni relative all'IP per un host di indirizzo **.ddns.org*, non utilizza i dati letti da un file al suo avvio, ma effettua ogni volta una interrogazione ad un database. Quest'ultimo e' mantenuto aggiornato mediante un protocollo definito ad hoc, che consente ad un host remoto di specificare (con le dovute password di autentificazione) un nuovo IP per un nome appartenente ad *.ddns.org. Chiaramente, l'IP specificato e' quello appena ricevuto dal provider, ed una transazione diversa, prima di scollegarsi da internet, comunica al DDNS l'abbandono del proprio IP. Ogni volta che un host di internet richiede il nostro IP, perche' ad esempio vuole consegnarci della posta o prelevare delle pagine dal nostro Apache, il DDNS rispondera' con l'IP dinamico (e dunque l'host remoto potra' connettersi a noi) oppure con un IP non in uso, confermando la nostra esistenza ma... spenti. Gli IP restituiti dal DDNS hanno una "data di scadenza" (il *Time To Live*, TTL) di soli due minuti, trascorsi i quali la risposta ricevuta dall'host remoto non e' piu' valida; questa viene quindi rimossa dalla memoria, e viene richiesta al DDNS una nuova risoluzione, in modo che "la rete" percepisca il nostro essere On- ed Off-Line con non piu' di 2 minuti di ritardo. Il protocollo con cui comunicare il nuovo IP o la sconnessione dal provider avviene per mezzo di due programmi client da eseguire all'interno di *ip-up* e prima della sconnessione.

Sorge ora pero' un nuovo problema: la posta a noi destinata giace all'interno di qualche nodo MX (*Mail Exchanger*) che periodicamente prova ad inviarcela, riuscendovi solo quando riceve l'IP valido. Pertanto, occorrerebbe rimanere collegati per tutto il tempo che intercorre tra due tentativi, che puo' essere anche di un'ora. Sembrerebbe un problema insormontabile e costoso. Invece, e' possibile limitare quest'attesa al massimo al TTL (2 minuti), configurando come nostro MX un computer in rete il cui SMTP server sia in grado di accettare il

comando ETRN [*RFC 1985*]. Questo comando, introdotto recentemente, permette di "risvegliare" l'MX, chiedendogli di inviare al piu' presto la posta per un particolare host (il nostro), evitando cosi' il periodo di attesa.

Un ultimo dettaglio, prima di essere in grado di ricevere posta, e' di comunicare al nostro host il suo nome registrato presso il DDNS, e verso cui va indirizzata la posta che vogliamo ricevere: questo nome andra' specificato nel file *sendmail.cw*.

11. Conclusioni

Sono stati illustrati tutti i passi necessari per rendere il nostro computer Linux accessibile via modem, nel rispetto dei protocolli TCP/IP utilizzati in Internet. Il risultato di questa procedura e' l'essere in grado di offrire servizi del tutto analoghi a quanto offerto da un provider, mantenendo lo spirito e l'indipendenza che ha da sempre animato il mondo delle BBS e della telematica amatoriale. La quasi totalita' dei files di configurazione nominati ma non descritti nel dettagli nell'articolo, puo' essere reperita presso [20].

12. Riferimenti

[1] Olaf Kirch, Linux Network Administratior-s Guide, O'Reilly & Associates

[2] Terry Dawson, file:/usr/doc/HOWTO/other-formats/html/NET-3-HOWTO.html

[3] AA.VV., http://www.isc.org/bog-4.9.4/bog.html

[4] AA.VV., file:/usr/doc/HOWTO/other-formats/html/NIS-HOWTO.html

[5] Nicolai Langfeldt, file:/usr/doc/HOWTO/other-formats/html/DNS-HOWTO.html

[6] Egil Kvaleberg, file:/usr/doc/HOWTO/other-formats/html/ISP-Hookup-HOWTO.html

[7] Greg Hankins, file:/usr/doc/HOWTO/other-formats/html/Serial-HOWTO.html

[8] Gert Doering, http://www.leo.org/~doering/mgetty/index.html

[9] Robert Hart, file:/usr/doc/HOWTO/other-formats/html/PPP-HOWTO.html

[10] http://www.apache.org/

[11] http://hoohoo.ncsa.uiuc.edu/cgi/

[12] Donnie Barnes, file:/usr/doc/RedHat/INN-Tips.html

[13] http://www.gigo.com

[14] http://www.average.org/ifmail/

[15] Robert Hart, file:/usr/doc/HOWTO/mini/Mail2News

[16] http://www.sendmail.org/

[17] http://www.greatcircle.com/majordomo/

[18] http://squid.nlanr.net/Squid/

[19] Ambrose Au, file:/usr/doc/HOWTO/mini/IP-Masquerade

[20] http://comel.ing.uniroma1.it/~sandro/lime98/bbs2pop.tgz