contiene la versione più recente di questo documento.

VoIP Howto

Roberto Arcomano berto@bertolinux.com v 1.8 - 26 Agosto 2002

Contents

1 Introduzione

1.1 Introduzione

Questo documento tratta di sistemi VoIP. I recenti avvenimenti come la diffusione di Internet a basso costo, la nuova integrazione di di processori dedicati per la compressione della voce (vedi DSP), hanno cambiato le esigenze del generico utente, permettendo alla tecnologia VoIP di diffondersi. Questo howto cerca di tracciare le linee guida dell'architettura VoIP.

Sono ben accolti suggerimenti e critiche alla @fatamorgana *mia Email* `mailto:berto@bertolinux.com`

1.2 Copyright

1.3 Traduzioni

Sei libero di tradurre questo documento, devi soltanto rispettare 2 regole:

1. Controllare che non esista gia' un'altra versione del documento nel tuo LDP locale

2. Mantenere la sezione 'Introduzione' (che include i paragrafi 'Introduzione', 'Copyright', 'Traduzioni', 'Ringraziamenti').

Attenzione! Non bisogna tradurre il file TXT o HTML, bensi' il file LYX, cosicche' sia possibile convertirlo negli altri formati (TXT, HTML, RIFF, ecc.): puoi utilizzare l'applicativo LyX scaricabile da *http://www.lyx.org* `http://www.lyx.org` .

Non c'e' bisogno di chiedermi l'autorizzazione! E' sufficiente comunicarmelo.

Grazie per la traduzione.

1.4 Ringraziamenti

Ringraziamenti a *Fatamorgana Computers* `http://www.fatamorgana.com` per l'equipaggiamento hardware e per la sperimentazione.

Grazie anche al *Progetto Pluto* `http://www.pluto.linux.it` per la pubblicazione.

Grazie a *David Price* `mailto:dprice@intercorp.com.au` per il suo contributo.

2 Sfondo introduttivo

2.1 Il passato

Fino a piu' di 30 anni fa Internet non esisteva. Le comunicazioni interattive si basavano sul telefono, ai costi delle linee PSTN.

Lo scambio di dati era molto costoso (soprattutto per lunghe distanze) e nessuno immaginava al video interattivo (esisteva soltanto la televisione, che, com'é noto, non é interattiva).

2.2 Ieri

Pochi anni fa abbiamo assistito ad alcuni importanti fenomeni: PCs diffusi a larga scala, nuove tecnologie per comunicare come i telefoni cellulari e, finalmente, la grande rete: Internet; la gente ha iniziato ad utilizzare i primi servizi email, chat, ecc. e il business é rinato con il web permettendo alle persone di acquistare prodotti via Internet con un "click".

2.3 Oggi

Oggi assistiamo ad una vera e propria rivoluzione nel campo della comunicazione: tutti iniziano ad usare il PC con Internet nel lavoro e nel tempo libero per scambiare dati (come immagini, suoni, documenti) e, in alcuni casi, per parlare usando applicativi come Microsoft Netmeeting o Internet Phone. In particolare inizia a diffondersi un'idea comune che potrebbe rappresentare il futuro e che permette la comunicazione vocale in tempo reale: VoIP.

2.4 Il futuro

Non possiamo prevedere il futuro, ma lo immaginiamo con molti computers, Internet quasi ovunque ad alta velocità e persone che dialogano tra loro (audio and video) in tempo reale. Dobbiamo soltanto sapere quale mezzo si utilizzerà: UMTS, VoIP (con l'estensione video) o altro ancora? Comunque sia bisogna accettare il fatto che Internet é cresciuta in una maniera impressionante negli ultimi anni, é gratis (al meno dal punto di vista di "mezzo internazionale") e potrebbe essere il giusto mezzo di comunicazione del futuro.

3 Visione generale del VoIP

3.1 Cos'é VoIP?

VoIP sta' per 'V'oice 'o'ver 'I'nternet 'P'rotocol. Come dice il termine VoIP prova a far passare la voce (prettamente quella umana) attraverso i pacchetti IP e in definitiva attraverso Internet. La tenologia VoIP può avvalersi di schede hardware acceleratrici per raggiungere tale scopo ed é possibile il suo l'utilizzo in ambiente PC.

3.2 Come funziona?

Molti anni fa é stato scoperto che per mandare un generico segnale lontano era possibile utilizzare il formato digitale: prima di mandarlo dobbiamo digitalizzarlo con un ADC (convertitore analogico-digitale), trasmetterlo, e trasformarlo di nuovo in formato analogico con un DAC (convertitore digitale-analogico) per utilizzarlo.

VoIP lavora proprio in questo modo, digitalizzano la voce i pacchetti, mandandoli in rete e riconvertendoli in voce una volta giunti a destinazione.

I vantaggi del formato digitale sono notevoli: possiamo comprimere i dati, instradarli (utilissimo su Internet), convertirli nuovamente in un formato più consono al mezzo utilizzato e così via; sappiamo anche che il segnale digitale é più "resistente" ai disturbi rispetto a quello analogico (vedi GSM contro TACS).

Le reti TCP/IP sono costituite di pacchetti IP contenenti un'intestazione (per controllare la comunicazione) e di una parte dati: VoIP utilizza questo paradigma per attraversare la rete ed arrivare a destinazione.

```
Voce (sorgente)  - - ADC - - - - Internet - - - DAC  - - Voce (dest)
```

3.3 Quali sono i vantaggi nell'usare VoIP invece delle linee PSTN?

Quando usi la linea telefonica, si paga una tariffa dipendente dal tempo utilizzato ad un gestore di telefonia fissa (o mobile): più tempo si sta' e più si paga. Inoltre é possibile parlare soltanto con una persona alla volta.

Al contrario, con la tecnologia VoIP puoi parlare tutto il tempo che vuoi (l'importante é che anche l'altra persona sia connessa ad Internet nello stesso istante), a che distanza vuoi (senza differenza di costo) e con la possibilità di parlare con più persone contemporaneamente (la cosiddetta conferenza).

Se non sei ancora convinto considera che, nello stesso istante, puoi scambiare dati con le stesse persone, mandando immagini, grafici e video e documenti.

3.4 Allora perché non lo stiamo già utilizzando tutti?

Sfortunatamente vi sono alcune problematiche di carattere tecnico nell'interazione tra l'architettura VoIP e Internet.Come si può facilmente immaginare, la comunicazione vocale richiede una sorta di tempo reale per funzionare bene (non si può parlare, aspettare alcuni secondi ed ascoltare la risposta in ritardo): questo, però va in contrasto con la filosofia architetturale di Internet che può essere composta di molti routers (computers utilizzati per instradare pacchetti IP), circa 20-30 o più e che può portare ad un tempo medio di accesso (RTT, round trip time) molto elevato rendendo necessarie delle modifiche strutturali.

Nei prossimi capitoli proveremo a capire come risolvere questo grosso problema. In generale, però dobbiamo essere consci del fatto che é molto difficile "garantire" una banda su Internet per le applicazioni multimediali.

4 Informazioni tecniche sul VoIP

Ecco alcune importanti informazioni sulla tecnologia VoIP, necessarie per comprenderla bene.

4.1 Vista d'insieme di una connessione VoIP

Per configurare una comunicazione VoIP abbiamo bisogno:

1. Prima di tutto di un ADC per convertire la voce in segnali digitali (bits)
2. I bit vanno compressi in un buon formato per la trasmissione: c'é tutta una serie di protocolli atti a tale scopo che vedremo più avanti.
3. Adesso dobbiamo inserire i pacchetti voce in pacchetti di dati standard utilizzando un protocollo real-time (tipicamente RTP su UDP su IP)
4. Abbiamo bisogno di un protocollo di segnalazione per chiamare gli utenti: ITU-T H323 fa proprio al caso nostro.
5. In ricezione dobbiamo disassemblare i pacchetti, estrarre i dati, convertirli in segnali vocali analogici e mandarli alla scheda audio (o alla cornetta).
6. Tutto questo deve essere effettuato in tempo reale poiché non possiamo permetterci ritardi eccessivi durante il dialogo vocale! (vedi sezione QoS, Qualità del Servizio)

```
                        Architettura base

Voce )) ADC - Algoritmo di compressione - Assembl. RTP in TCP/IP -----
                                                         ---->       |
                                                         <----       |
Voce (( DAC - Algoritmo di decompress.  - Disass. RTP da TCP/IP  -----
```

4.2 Conversione analogica digitale

Questo viene fatto dall'hardware, tipicamente da un ADC integrato.

Oggi pressoché ogni scheda audio permette di convertire in 16 bit una banda di 22050 Hz (per il campionamente della quale abbiamo bisogno di 44100 Hz per il teorema del campionamento) ottendendo una velocità di 2 bytes * 44100 (campioni al secondo) = 88200 Bytes/s, 176.4 kBytes/s per i flussi dati stereo.

Per il VoIP non abbiamo certamente una velocita' cosi' elevata (176.4 kBytes/s): vediamo allora quali codifiche utilizzare.

4.3 Algoritmi di compressione

Vediamo quali formati di digitalizzazione e compressione utilizziamo.

`PCM, Pulse Code Modulation, Standard ITU-T G.711`

- La banda della voce é 4 kHz, quindi abbiamo campioni alla velocità di 8 kHz (dal teorema del campionamento).
- Rappresentiamo ogni campione con 8 bit (quindi con 256 valori possibili).
- La velocità compressiva é 8000 Hz *8 bit = 64 kbit/s, la tipica velocità di un canale telefonico digitale.
- Nelle applicazioni reali si utilizzano le varianti mu-law (Nord America) e a-law (Europa) che codificano il segnale analogico su scala logaritmica sfruttando 12 o 13 bits invece di 8 bits (vedi Standard ITU-T G.711).

`ADPCM, Adaptive differential PCM, Standard ITU-T G.726`

Converte soltanto la differenza tra il pacchetto attuale e quello precedente richiedendo 32 kbps (vedi Standard ITU-T G.726).

```
LD-CELP, Standard ITU-T G.728
CS-ACELP, Standard ITU-T G.729 and G.729a
MP-MLQ, Standard ITU-T G.723.1, 6.3kbps, Truespeech
ACELP, Standard ITU-T G.723.1, 5.3kbps, Truespeech
LPC-10, fino a 2.5 kbps!!
```

Gli ultimi protocolli sono i più significativi perché garantiscono un basso utilizzo di banda utilizzando metodi di codifica alla sorgente: inoltre le codifiche G.723.1 hanno un MOS molto elevato (Mean Opinion Score, utilizzato per misurare la fedeltà vocale) ma attenzione alle prestazioni richieste, finoa 26 MIPS (milioni di istruzioni al secondo)!

4.4 RTP Protocollo di trasporto Real Time

Adesso che abbiamo i dati "grezzi", per incapsularli nello stack TCP/IP, seguiamo la seguente struttura:

```
VoIP data packets
       RTP
       UDP
       IP
    I,II layers
```

I Pacchetti di voce risiedono in pacchetti RTP (Protocollo di trasporto Real-Time) che a loro volta giacciono su pacchetti UDP-IP.

Prima di tutto notiamo che VoIP non utilizza il protocollo TCP perche' troppo persante per le applicazioni multimediali (che sono, di per se stesse "real time"), quindi abbiamo bisogno di usare l'UDP

Secondo, con l'UDP non possiamo implicitamente controllare l'ordine di arrivo dei pacchetti o quanto impiegano ad arrivare (concetto di datagramma): entrambi sono molto importanti per ottenere una buona qualita' audio della voce (per poter distinguere le parole) e una buona qualita' di conversazione (la facilita' con cui si segue un discorso).

Il protocollo RTP risolve il problema permettendo al ricevitore di ordinare i pacchetti in base all'ordine di arrivo e di non aspettare troppo a lungo per un pacchetto perso (in effetti non abbiamo bisogno di ogni singolo pacchetto di voce, possiamo anche perderne qualcuno, l'importante e' ricevere un flusso continuo della maggiorparte dei pacchetti trasmessi e, ovviamente, in ordine).

```
                   Real Time Transport Protocol

    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |V=2|P|X|  CC   |M|     PT      |       sequence number         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                           timestamp                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           synchronization source (SSRC) identifier            |
   +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
   |            contributing source (CSRC) identifiers             |
   |                             ....                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Dove:

- V identifica la versione di RTP
- P indica il padding, un byte di riempimento.
- X indica la presenza delll'intestazione estesa.
- Il campo CC rappresenta il numero di identificatori CSRC che seguono l'header a lunghezza fissa. Per esempio il campo CSRC viene utilizzato in caso di conferenza.
- M é un bit per marcare.
- PT indica il tipo di PayLoad (dati)

Per una descrizione approfondita del protocollo RTP e di tutte le applicazioni relative si veda i relativi RFCs *1889* `http://www.ietf.org/rfc/rfc1889.txt` e *1890* `http://www.ietf.org/rfc/rfc1890.txt` .

4.5 RSVP

Nell'ambiente VoIP vengono utilizzati altri protocolli, come l'RSVP, che può gestire la Qualità del Servizio (QoS).

RSVP é un protocol di segnalazione che permette di riservare una certa quantità di banda e di latenza massima in ogni nodo (router) di rete attraversato che lo supporta.

Per informazioni dettagliate sull'RSVP si veda l' *RFC 2205* `http://www.ietf.org/rfc/rfc2205.txt?number=2205`

4.6 Qualità del servizio (QoS)

Abbiamo detto molte volte che le applicazioni VoIP richiedono un flusso di dati in tempo reale.

Sfortunatamente, TCP/IP non può garantire tale flusso (può solo effettuare il massimo sforzo per cercare di conseguirlo). Abbiamo quindi bisogno di introdurre degli artifizi e delle politiche di schedulazione che possano gestire i pacchetti in OGNI router che attraversano.

Ecco allora:

1. Il campo TOS nel protocollo IP per descrivere il tipo di servizio:valori alti indicano bassa urgenza, mentre valori via via sempre più alti segnalano urgenza crescente.
2. Metodi di accodamento dei pacchetti:
   (a) FIFO (First in First Out), Il più stupido del metodi che fa passare i pacchetti in ordine di arrivo.
   (b) WFQ (Weighted Fair Queuing), consiste nel far passare in modo equo i pacchetti (per esempio, i pacchetti FTP non possono consumare tutta la banda disponibile), a seconda del tipo di dati che contengono, tipicamente vengono fatti passare un pacchetto UDP ed uno TCP in modo paritario.
   (c) CQ (Custom Queuing), gli utenti possono decidere la loro priorità.
   (d) PQ (Priority Queuing), c'é una numbero (tipicamente 4) di code con ciascuno un livello di priorità associato: prima di tutto, vengono fatti passare i pacchetti nella prima coda, poi (una volta terminati) si passa a quelli della seconda e così via.
   (e) CB-WFQ (Class Based Weighted Fair Queuing), simile al WFQ ma, in più, abbiamo il concetto di classi (fino a 64) e la di banda associata a ciascuna classe.
3. Proprietà di shaping e policing, che permetteno di limitare la sorgente ad una ben prefissata banda in:
   (a) download
   (b) upload
4. Prevenzione della congestione di rete, come il RED (Random Early Detection).

Una esaustiva descrizione sulla QoS può essere trovata sui *Differentiated Services* `http://www.ietf.org/html.charters/diffserv-charter.html` presso lo IETF.

4.7 H323: protocollo di segnalazione

Il protocollo H323 viene utilizzato, ad esempio, dall'applicativo Microsoft Netmeeting per creare chiamate VoIP.

Questo protocollo permette ad una serie di elementi di "parlare" tra loro:

1. Terminali, clients che inizializzano le connessioni VoIP. Sebbene i terminali possano dialogare insieme senza aver bisogno di nessun altra identità, é necessario introdurre elementi addizionali per una visione più scalabile.
2. Gatekeepers, che operano essenzialmente:
   (a) Una conversione nome-indirizzo, per utilizzare i nomi al posto degli indirizzi IP
   (b) Un controllo di accesso, per abilitare o disabilitare alcune macchine o alcuni utenti
   (c) Gestione della banda.
3. Gateways, punto di riferimento per la conversione TCP/IP - PSTN.
4. Multipoint Control Units (MCUs) to provide conference.
5. Vengono anche utilizzati Servers Proxy.

H323 permette anche la gestione del video.

Per quanto riguarda la voce, l'h323 é compatibile con le codifiche G.711, G.722, G.723, G.728 e G.729, mentre per il video supporta i protocolli h261 and h263.

Maggiori informazioni sul protocollo h323 sono disponibili sugli

Standards presso Openh323 `http://www.openh323.org/standards.html` , su at *questo sito web h323* `http://www.cs.columbia.edu/~hgs/rtp/h323.html` , mentre la descrizione dello standard ufficiale é reperibile dalle *ITU H-series Recommendations* `http://www.itu.int/itudoc/itu-t/rec/h/` .

L'h323 viene implementato in molto prodotti software come *Microsoft Netmeeting* `http://www.microsoft.com` , *Net2Phone* `http://www.net2phone.com` ,

DialPad `http://www.dialpad.com` , ... e altri prodotti freeware che puoi trovare sul *sito web Openh323* `http://www.openh323.org` .

5 Requisiti

5.1 Requisiti hardware

Per creare un piccolo sistema VoIP hai bisogno del seguente hardware:

1. PC 386 o superiore
2. Scheda audio full duplex
3. una scheda di rete o una connessione ad Internet o altro tipo di interfacciamento per permettere la comunicazione TCP/IP tra 2 PCs

Tutto questo deve essere presente in 2 copie in quanto vogliamo simulare una comunicazione tra 2 entità.

Gli strumenti sopra citati sono i minimi richiesti per una connessione VoIP: più avanti vedremo che avremo bisogno di un hardware maggiore per avere successo una situazione reale su Internet

La scheda audio deve necessariamente essere full duplex, altrimenti non sarà possibile ascoltare nulla mentre si parla!

In aggiunta si possono utilizzare schede acceleratrici per permettere la gestione di un flusso dati compresso (see Par 4.3).

5.2 Schede acceleratrici

Possiamo usare schede acceleratrici per ottimizzare il protocollo di trasmissione. Due di queste (ed anche le uniche direttamente gestire dal kernel di Linux fino adesso) sono

1. Quicknet PhoneJack
2. Quicknet LineJack
3. VoiceTronix V4PCI
4. VoiceTronix VPB4
5. VoiceTronix VPB8L

La Quicknet PhoneJack é una scheda audio che utilizza algoritmi standard per la compressione di streams audio quali G.723.1 ed altri (section 4.3) fino a 4.1 kbps di velocità

Può essere connessa direttamente ad una cornetta (POTS port) o ad una coppia microfono - altoparlante.

E' disponibile con slots ISA e PCI.

La Quicknet LineJack lavora come la PhoneJack ma ha funzionalità addizionali (vedi più avanti).

Per ulteriori informazioni si veda il *sito web Quicknet* `http://www.quicknet.net` .

VoiceTronix V4PCI e' una scheda PCI simile alla Quicknet LineJack ma con 4 porte per il telefono.

VoiceTronix VPB4 e' la versione ISA equivalente alla V4PCI.

VoiceTronix VPB8L e' una scheda di registrazione chiamate con 8 porte.

Per ulteriori informazioni si veda il *sito web VoiceTronix* `http://www.voicetronix.com.au` .

5.3 Schede gateway

La Quicknet LineJack e le schede VoiceTronix possono essere connessa ad una linea PSTN permettendo di creare un gateway VoIP.

Dopodiché bisogna usare un software che ne permetta il controllo (vedi dopo).

5.4 Requisiti software

Vediamo quale software possiamo installare utilizzando rispettivamente:

1. Win9x
2. Linux

Sotto Win9x abbiamo il classico Microsoft Netmeeting oppure Internet Phone, DialPad e altri ancora mentre Quicknet ci offre il suo Internet Switchboard (scaricabile dal *sito web Quicknet* `http://www.quicknet.net`) che utilizza, appunto, le schede Quicknet.

Attenzione!!: Le ultime schede Quicknet che utilizzano Swithboard (anche versioni piu' vecchie) NECESSITANO del collegamente ad Internet per funzionanre, in quanto devono connettersi al Provider VoIP Microtelco (quindi bisogna pagare una ricarica), quindi se si prevedere di rimanere non connessi ad Internet sara' meglio utilizzare il *software OpenH323* `http://www.openh323.org`

Per le schede VoiceTronix si veda il *sito web VoiceTronix* `http://www.voicetronix.com.au`

Sotto Linux possiamo utilizzare soltanto il software libero (per il momento) sempre scaricabile da *OpenH323* `http://www.openh323.org` : simph323 o ohphone che può lavorare con l'hardware Quicknet.

Attenzione: tutto il codice sorgente Openh323 deve essere compilato in una directory home (oppure sarà necessario settare una variabile d'ambiente al valore opportuno). Il tempo di compilazione potrebbe essere molto elevato e potrebbe richiedere una quantità non indifferente di RAM (un Pentium 133 con 16 MB di ram non é consigliabile!).

5.5 Software gateway

Per gestire le caratteristiche di un gateway (far comunicare una rete VoIP con una rete PSTN) é necessario installare un qualche tipo di software:

- *Internet SwitchBoard* `http://www.quicknet.net` (solo se connesso ad Internet) per i sistemi Windows che opera anche come terminale h323;
- *PSTNGw* `http://www.openh323.org/code.html` per Linux e Windows .

5.6 Software gatekeeper

Come gatekeeper: Openh323 Gatekeeper (GK) da *qui* `http://www.gnugk.org/` .

La versione 2.0 supporta la funzione "proxy" che permette di dialogare da/verso una rete privata.

5.7 Altro software

In aggiunta riporto del software compatibile h323:

- Phonepatch, in grado di risolvere problematiche relative ad un firewall-NAT. Semplicamente permette agli utenti (esterni ed interni) di effettuare chiamate da una pagina web (raggiungibile sia dall'esterno che dall'interno della rete): quando l'applicativo web si accorge che l'utente remoto é attivo provvede a chiamare l'utente che ha effettuato la chiamata web (tramite una chiamata h323) comunicandogli che la comunicazione può avvenire. Phonepatch é un software proprietario (la versione demo permette di effettuare chiamate di non più di 3 minuti) che puoi trovare su *questo sito web* `http://www.equival.com/phonepatch` .

La stessa funzione puo' essere ottenuta con la funzionalita' di "Proxy" presente nel *Gatekeeper Gnugk* `http://www.gnugk.org/` (vedi prima).

6 Configurazione delle schede

Qui vediamo come configurare le schede hardware acceleratrici sotto Linux e Windows.

6.1 Quicknet PhoneJack

Come abbiamo visto la Quicknet Phonejack é una scheda audio con proprietà di accelerazione VoIP. Supporta:

- G.711 normale e mu/A-law, G.728-9, G.723.1 (TrueSpeech) e LPC10.
- Connettore telefonico (per l'utilizzo con una cornetta) o
- Jacks Microfono + Altoparlante.

La PhoneJack é una scheda ISA (o PCI) da installare sul Pc. Può anche lavorare senza IRQ.

6.1.1 Installazione del software

Sotto Windows bisogna installre:

1. Il driver della scheda
2. l'applicativo Internet Switchboard

tutto ciò si trova sul *sito web Quicknet* `http://www.quicknet.net`

Dopo che Internet Switchboard é stato installato, hai bisogno di registrarti alla Quicknet per poter usufruire di tutte le funzionalità della tua scheda.

Quando alzi la cornetta Switchboard si sveglia e rimane in attesa del numero di telefono da comporre (direttamente dalla cornetta). In tal caso si può:

1. Digitare asterisco, poi comporre un indirizzo IP (usando gli asterischi al posto del punto) e cancelletto per terminare la composizione.
2. Digitare direttamente un numero telefonico (completo di prefisso internazionale) per chiamare un telefono classico. In tal caso avrai bisogno della registrazione presso un gestore gateway a cui pagare a seconda del tempo che rimarrai connesso.
3. Digitare direttamente un numero di chiamata rapida che hai precedentemente memorizzato (fino a 2 cifre).

Internet Swichboard é compatibile h323, quindi puoi usare software commerciale come Microsoft Netmeeting dall'altro "capo".

Al posto di Internet Switchboard si può usare anche un applicativo free come *openphone* `http://www.openh323.org/code.html` (con interfaccia grafica) o *ohphone* `http://www.openh323.org/code.html` (linea di comando).

Sotto Linux devi installare:

1. Il driver della scheda acceleratrice, dal *sito web Quicknet* `http://www.quicknet.net` . Dopodiché dovrai compilarlo (controllando opportunamente che la directory /usr/src/linux punti alla directory dei sorgenti di Linux): digita "make" per le istruzioni.
2. L'applicativo *Openphone* `http://www.openh323.org/code.html` o *Ohphone* `http://www.openh323.org/code.html` .
3. Se sei uno sviluppatore puoi utilizzare le *SDK* `ftp://ftp.quicknet.net/Developer/Linux/Docs/` per creare le tue applicazioni personalizzate (anche sotto Windows).

6.1.2 Settaggi

Con Internet Switchboard (e anche con gli altri applicativi) puoi:

1. Cambiare l'algoritmo di compressione preferito
2. Variare il ritardo "jitter" (jitter delay)
3. Settare il volume
4. Settare il livello di cancellazione dell'eco.

6.2 Quicknet LineJack

Questa scheda é molto simile alla precedente, e, in aggiunta, supporta le funzionalità di gateway. In effetti la Linejack presenta, oltre ai connettori classici della PhoneJack, anche un connettore per la linea telefonica PSTN.

L'unico appunto che vale la pena di sottolineare é la necessità di utilizzare un applicativo in grado di gestire anche tale linea PSTN: *PSTNGw* `http://www.quicknet.net/code.html` (for Linux and Windows) e *Internet Switchboard* `http://www.quicknet.net` .

6.3 VoiceTronix prodotti

1. Scaricare il software *qui* `http://www.voicetronix.com.au/vpb-driver-2.1.8.tar.gz`
2. scompattarlo
3. modificare 'src/vpbreglinux.cpp' seguendo le istruzioni sul file README allegato
4. dare 'make'
5. dare 'make install'
6. andare nella directory src
7. digitare 'insmod vpb.o'
8. cercare (dalla console o tramite il comando 'dmesg') il major number, che chiameremo MAJOR
9. digitare 'mknod /dev/vpb0 c MAJOR 0' dove MAJOR e' il numero descritto precedentemente
10. andare nella directory 'unittest' e digitare './echo'

Per ulteriori informazioni si veda il file README.

Non ho personalmente testato i prodotti VoiceTronix, quindi si contatti direttamente il *sito web VoiceTronix* `http://www.voicetronix.com.au` per il supporto tecnico.

7 Setup

In questo capitolo vedremo come configurare una sistema VoIP, semplice al principio, via via poi più complesso.

7.1 Una configurazione semplice: IP to IP

```
A (Scheda audio)   -  -  -   B (Scheda audio)

  192.168.1.1      -  -  -     192.168.1.2


  192.168.1.1 chiama 192.168.1.2 e viceversa
```

A e B devono avere:

1. Microsoft Netmeeting, Internet Switchboard, Openh323 (sotto Windows) o Ohphone, Gnomemeeting (sotto Linux) installato e correttamente configurato.
2. una scheda di rete o un altro tipo di interfaccia compatibile TCP/IP per dialogare tra loro.

In questa prospettiva A può effettuare una chiamata h323 verso B (se B ha l'applicativo lato server attivo) usando il suo indirizzo IP 192.168.1.2. Dopodiché B, se vuole, può rispondere e i pacchetti VoIP iniziano a transitare.

7.2 Uso dei nomi

Se utilizzi Microsoft Windows in una lan, puoi chiamare una macchina utilizzando il servizio nomi NetBIOS, che può funzionare su NetBEUI o su TCP/IP. Tutto ciò di cui hai bisogno, quindi, é di chiamare il nome dell'altro computer per effettuare la connessione.

```
          A            - - -            B

     192.168.1.1       - - -       192.168.1.2

       Mario           - - -          Paola


                    Mario chiama Paola.
```

Tutto ciò é possibile in quanto la chiamata NetBIOS di Mario verso Paola viene convertita in chiamata IP.

Gli esempi sopra citati sono molto semplici da implementari ma scarsamente scalabili.

In una rete più grande quale Internet é impossibile effettuare chiamate con gli indirizzi, in quanto difficilmente conosco l'indirizzo IP del destinatario (che nella maggior parte dei casi, essendo dinamico, cambia ad ogni nuova connessione al provider). Oltretutto il servizio di nomi NetBIOS non funziona poiché i messaggi da esso usati sono di tipo broadcast, che notoriamente non attraversano i routers.

Si puo' utilizzare anche il sistema DNS per risolvere i nomi in IP address: ad esempio si puo' chiamare "box.domain.com".

7.3 Chiamata su Internet utilizzando un server WINS

L'idea dei nomi NetBIOS può essere riutilizzata anche in ambiente Internet con l'ausilio di un server WINS: le macchine possono essere configurate in modo tale da chiedere ad un unico server la risoluzione dei nomi in indirizzi.

I PCs che utilizzano lo stesso server WINS saranno in grado di comunicare fra loro.

```
A (Server WINS é S) - - - - I  - - - -  B (Server WINS é S)
                             N
                             T
                             E  - - - - -   S (Server WINS)
C (Server WINS é S) - - - - R
                             N
                             E  - - - -  D (Server WINS é S)
                             T

                   Comunicazione su Internet
```

A, B, C e D appartengono a sottoreti differenti, ma possono chiamarsi l'un l'altro utilizzando i loro nomi NetBIOS. Il requisito fondamentale, come precedentemento detto é che tutti gli hosts utilizzino la stessa macchina come Server WINS (S appunto).

Nota: I servers WINS non hanno performance eccezionali dato che si basano sul protocollo NetBIOS (che a livello locale continua a fare uso dei broadcasts) e dovrebbe essere utilizzato soltanto per unire piccole e poche sottoreti.

7.4 Server ILS

ILS e' un tipo di server che permette di risolvere i nomi in una chiamata H323: quando si lancia l'applicativo VoIP bisogna prima registrarsi presso il server ILS tramite un nome, dopodiche' chiunque potra' accedere alla nostra macchina tramite tale nome (se anch'esso specifichera' lo stesso server ILS!).

7.5 Un grosso problema: il masquering.

Un problema di carenza di indirizzi IP viene normalmente risolto utilizzando il cosiddetto masquering (anche chiamato NAT, network address translation): esiste soltanto un IP pubblico (che Internet può "vedere" direttamente), le altre macchine vengono "mascherate" utilizzando tale IP.

```
A  - - -

B  - - -   Router con NAT  - - -  Internet

C  - - -


                Questo non funziona
```

Nell'esempio A,B e C possono navigare, pingare, usare posta, news e altri servizi su Internet, ma NON possono fare una chiamata VoIP verso l'esterno. Questo perché il protocollo h323 manda l'indirizzo IP a livello applicazione, quindi la risposta non arriverà mai al chiamante (che utilizza un indirizzo IP privato).

Soluzioni:

- Esiste un modulo su Linux che modifica i pacchetti h323 evitando il problema. Lo puoi scaricare *qui* `http://www.coritel.it/coritel/ip/sofia/nat/nat2/nat2.htm` . Per installarlo devi copiarlo nella directory sorgente del kernel (specificata dalle istruzioni sul sito), modificare il Makefile e ricompilare i moduli del kernel installando poi il modulo con "modprobe ip_masq_h323". Sfortunatamente questo modulo non lavora (per il momento) con il software libero Ohphone (non ne conosco il motivo).

```
A  - - -   Router con NAT

B  - - -        +           - - -  Internet

C  - - -  modulo ip_masq_h323


                Questo funziona
```

- C'é anche un applicativo che risolve il problema del masquering: per ulteriori informazioni si veda il 5.7 (Par 5.7)

```
A  - - -

B  - - -    PhonePatch   - - -  Internet
```

```
          C  - - -


                              Questo funziona
```

7.6 Applicazioni Open Source

Con Linux (usato come terminale h323) puoi sperimentare tutte le prove fatte precedentemente eccetto quelle dietro un router-NAT con il modulo ip_masq_h323 poiché (come già detto) Ohphone non funziona con tale modulo: sarai quindi costretto ad usare Phonepatch.5.7 (Par 5.7)

7.6.1 Sintassi Ohphone

"ohphone -l|–listen [opzioni]"

"ohphone [opzioni]... indirizzo"

- "-l", ascolta sulla porta standard (1720)
- "indirizzo", significa che non vogliamo ricevere, bensi' fare una chiamata a "indirizzo"
- "-n", "–no-gatekeeper", questo dice di non usare gatekeeper
- "-q num", "–quicknet num", utilizza la scheda Quicknet, device /dev/phone(num)
- "-s device", "–sound device", utilizza il device sonoro "/dev/device"
- "-j ritardo", "–jitter ritardo", cambia il buffer di ritardo, impostandolo a "ritardo".

Inoltre, quanto lanciamo ohphone, possiamo digitare dei comandi direttamente sull'interprete (come, ad esempio, decrementare l'AEC, Automatic Echo Cancellation).

7.6.2 Gnomemeeting

Gnomemeeting e' un'applicazione che usa l'interfaccia grafica (GUI) per effettuare chiamate VoIP. E' molto semplice da usare e permette anche di specificare un server ILS, di chattare, ecc.

7.7 Configurare un gatekeeper

Puoi provare le funzionalità di gatekeeper.

```
Esempio

        (Terminale H323) A  - - -
                                 \
        (Terminale H323) B  - -  - D (Gatekeeper)
                                 /
        (Terminale H323) C  - - -

                   Configurazione del Gatekeeper
```

1. Gli hosts A,B e C hanno come gatekeeper D.
2. Alla partenza ogni host comunica al gatekeeper il proprio indirizzo e il proprio nome associato (compresi gli alias) che potrà essere utilizzato da un chiamante per raggiungerlo.
3. Quando un terminale chiede a D un determinato, D risponderà fornendo al chiamante il giusto indirizzo IP, permettendo così la comunicazione tra i 2 terminali.

Notiamo che il gatekeeper é soltanto in grado di risolvere i nomi in indirizzi, non può unire hosts che non già siano raggiungibili (a livello IP), in altre parole il gatekeeper non può operare da router NAT.

Puoi trovare il gatekeeper *qui* `http://www.opengatekeeper.org` : sono richieste le *librerie Openh323* `http://www.openh323.org/code.html` .

E' necessario lanciare il programma almeno con l'opzione -d (come demone) o -x (per eseguirlo in foreground).

E' anche disponibile un file di configurazione (.ini) *qui* `http://www.opengatekeeper.org/opengate.ini` .

7.8 Configurare un gateway

Come abbiamo già detto, un gateway é un'entità che può far comunicare una rete VoIP ad una PSTN permettendoci di effettuare una chiamata da Internet verso un comune telefono.Abbiamo quindi bisogno della scheda gateway (Quicknet LineJack) e del relativo software:

Sul *sito web OpenH323* `http://www.openh323.org` troviamo:

1. I driver per la Linejack
2. L'applicativo PSTNGw per gestire il gateway.

Se l'eseguibile non funziona avrai bisogno di scaricare i sorgenti e compilarli dopo aver opportunamente scaricato e compilato anche le *librerie Openh323* `http://www.openh323.org/code.html` , entrambi su una home directory.

8 Chiamate tramite linea telefonica PSTN

8.1 Introduzione

Il VoIP diventa interessante quando si inizia ad utilizzare le linee telefoniche (PSTN) per chiamare altre persone nel mondo, direttamente al loro telefono di casa.

8.2 Scenario

Uno schema tipico e' il seguente:

```
Telefono1 -- (PSTN) -- PC1 -- (Internet) -- PC2 -- (PSTN) -- Telefono2
```

1. Telefono1 chiama il numero di telefono di PC1 (tramite la linea telefonica PSTN, quella classica).
2. PC1 risponde.
3. Telefono1 dice a PC1 quale gateway usare (PC2 in questo caso) mandandogli l'IP address (direttamente dalla tastiera DTMF) e/o quale numero chiamare (in questo caso Telefono2).

4. Dopo di che' PC1 effettuera' una chiamata H323 verso PC2, passandogli il numero di Telefono2 che PC2 chiamera' tramite la linea telefonica.
5. Telefono2 risponde e la comunicazione tra i 2 telefoni ha inizio.

8.3 Quali modifiche si possono apportare al sistema?

1. Si puo' usare un centralino (PBX) per scegliere tra diverse linee telefoniche per raggiungere diversi gateway PC1 (ad esempio per chiamare all'interno del paese, per andare fuori Europa e cosi' via..): di solito non viene molto usato perche' il costo della chiamata non cambia.
2. Si puo' scegliere (dopo aver chiamato il gateway PC1) il PC2 che si vuole (per esempio se si chiama un PC2 che risiede in Inghilterra per chiamare un Inglese si paghera' soltanto il costo della chiamata interna all'Inghilterra).

La decisione verra' quindi presa in base al costo delle linee telefoniche. In effetti quello che il VoIP permette e' di convertire questo:

```
   Telefono1 --- (PSTN) --- Telefono2
 Costo di una chiamata ad elevata distanza
```

in questo:

```
   Telefono1 --- (PSTN) --- PC1   +
   PC2 ---- (PSTN) --- Telefono2  =
   --------------------------------
Costo di 2 chiamate a piccola distanza
```

Per risparmiare e' quindi necessario che:

```
Costo di 2 chiamate a piccola distanza < Costo di una chiamata ad elevata distanza
```

Tipicamente "chiamate a piccola distanza" si intendono chiamate urbane mentre per "chiamata ad elevata distanza" si intende un interurbana o una intercontinentale!

9 Considerazioni di banda

Da tutto quello che abbiamo detto non abbiamo ancora risolto i problemi legati alla banda, come cioé creare un flusso di dati in tempo reale. Sappiamo che non possiamo risolverlo finché non abilitiamo i protocolli per la gestione della banda su ogni router che attraversiamo, allora cosa possiamo fare?

Prima di tutto cerchiamo di utilizzare l'lagoritmo di compressione che più ci aiuta ad occupare meno banda possibile (come, ad esempio, l' LPC-10 che riesce a consumare appena 2.5 kbps, circa 313 bytes/s).

Dopo di ché iniziamo a classificare i nostri pacchetti (iniziando dal campo TOS del protocollo IP), con il valore più alto possibile coì da ottenere dai router la priorità massima possibile.

Importante: tutto questo NON ci mette al sicuro garantendoci una banda sufficiente per la nostra conversazione, perché finché non si crea un'infrastruttura in grado di gestire shaping, prenotazione di banda e così via non sarà mai possibile avere una garanzia assoluta, in effetti TCP/IP non é un protocollo real time.

A possible solution could be starts with little WAN at guaranteed bandwidth and get larger step by step.

Possiamo infine notare un'ultima cosa: anche i cosiddetti servizi garantiti come le linee PSTN non sono in grado di gestire tutti clienti che hanno nello stesso istante: per esempio una cella GSM non può sopportare più di un centinaio o un migliaio di utenti alla volta.

Comunque sia il servizio VoIP può essere considerato un buon punto di partenza, in alternativa al servizio offerto dalle linee PSTN.

10 Links utili

10.1 Free software

- *Voxilla* `http://www.voxilla.org`
- *Linux Telephony* `http://www.linuxtelephony.org`
- *sito web Open H323* `http://www.openh323.org`
- *http://www.gnomemeeting.org/* `http://www.gnomemeeting.org/`
- *Speak Freely* `http://www.speakfreely.org`
- *http://www.linphone.org* `http://www.linphone.org`
- *http://www.fsf.org/software/osip* `http://www.fsf.org/software/osip`
- *http://www.gnu.org/software/bayonne* `http://www.gnu.org/software/bayonne`

10.2 Link Commerciali

- *Fatamorgana Computers* `http://www.fatamorgana.com`
- *International Communication Union* `http://www.itu.org`
- *sito web Voicetronix* `http://www.voicetronix.com.au`
- *sito web Quicknet* `http://www.quicknet.net`
- *Cisco Systems* `http://www.cisco.com`
- *www.metropark.com* `www.metropark.com`
- *www.nbxsoftware.com* `www.nbxsoftware.com`