VoIP−HOWTO

Table of Contents

Table of Contents

VoIP Howto

Roberto Arcomano bertolinux@galactica.it

v1.0 – 12 November 2000

Voice Over IP is a new communication means that let you telephone with Internet at almost null cost. How this is possible, what systems are used, what is the standard, all that is covered by this Howto. Web site http://web.tiscalinet.it/bertolinux contains latest version of this document.

---

6. Cards setup

7. Setup

8. Bandwidth consideration

9. Useful links

---

1. Introduction

1.1 Introduction

This document explains about VoIP systems. Recent happenings like Internet diffusion at low cost, new integration of dedicated voice compression processors, have changed common user requirements allowing VoIP standards to diffuse. This howto tries to define some basic lines of VoIP architecture.

Please send suggestions and critics to my email address

1.2 Copyright

---

2. Background

2.1 The past

20−30 years ago Internet didn't exist. Interactive communications were only made by telephone at PSTN line cost.

Data exchange was expansive (for a long distance) and no one had been thinking to video interactions (there was only television that is not interactive, as known).

2.2 Yesterday

Few years ago we saw appearing some interesting things: PCs to large masses, new technologies to communicate like cellular phones and finally the great net: Internet; people begun to communicate with new services like email, chat, etc. and business reborned with the web allowing people buy with a "click".

2.3 Today

Today we can see a real revolution in communication world: everybody begins to use PCs and Internet for job and free time to communicate each other, to exchange data (like images, sounds, documents) and, sometimes, to talk each other using applications like Netmeeting or Internet Phone. Particularly starts to diffusing a common idea that could be the future and that can allow real−time vocal communication: VoIP.

2.4 The future

We cannot know what is the future, but we can try to image it with many computers, Internet almost everywhere at high speed and people talking (audio and video) in a real time fashion. We only need to know what will be the means to do this: UMTS, VoIP (with video extension) or other? Anyway we can notice that Internet has grown very much in the last years, it is free (at least as international means) and could be the right communication media for future.

---

3. Overview

3.1 What is VoIP?

VoIP stands for 'V'oice 'o'ver 'I'nternet 'P'rotocol. As the term says VoIP tries to let go voice (mainly human) through IP packets and, in definitive through Internet. VoIP can use accelerating hardware to achieve this purpose and can also be used in a PC environment.

3.2 How does it work?

Many years ago we discovered that sending a signal to a remote destination could have be done also in a digital fashion: before sending it we have to digitalize it with an ADC (analog to digital converter), transmit it, and at the end transform it again in analog format with DAC (digital to analog converter) to use it.

VoIP works like that, digitalizing voice in data packets, sending them and reconverting them in voice at destination.

Digital format can be better controlled: we can compress it, route it, convert it to a new better format, and so on; also we saw that digital signal is more noise tolerant than the analog one (see GSM vs TACS).

TCP/IP networks are made of IP packets containing a header (to control communication) and a payload to transport data: VoIP use it to go across the network and come to destination.

```
Voice (source)  − − ADC − − − − Internet − − − DAC  − − Voice (dest)
```

3.3 What is the advantages using VoIP rather PSTN?

When you are using PSTN line, you typically pay for time used to a PSTN line manager company: more time you stay at phone and more you'll pay. In addition you couldn't talk with other that one person at a time.

In opposite with VoIP mechanism you can talk all the time with every person you want (the needed is that other person is also connected to Internet at the same time), as far as you want (money independent) and, in addition, you can talk with many people at the same time.

If you're still not persuaded you can consider that, at the same time, you can exchange data with people are you talking with, sending images, graphs and videos.

3.4 Then, why everybody doesn't use it yet?

Unfortunately we have to report some problem with the integration between VoIP architecture and Internet. As you can easy imagine, voice data communication must be a real time stream (you couldn't speak, wait for many seconds, then hear other side answering): this is in contrast with the Internet heterogeneous architecture that can be made of many routers (machines that route packets), about 20−30 or more and can have a very high round trip time (RTT), so we need to modify something to get it properly working.

In next sections we'll try to understand how to solve this great problem. In general we know that is very difficult to guarantee a bandwidth in Internet for VoIP application.

---

4. Technical info about VoIP

Here we see some important info about VoIP, needed to understand it.

4.1 Overview on a VoIP connection

To setup a VoIP communication we need:

1. First the ADC to convert analog voice to digital signals (bits)
2. Now the bits have to be compressed in a good format for transmission: there is a number of protocols we'll see after.
3. Here we have to insert our voice packets in data packets using a real−time protocol (typically RTP over UDP over IP)
4. We need a signaling protocol to call users: ITU−T H323 does that.

5. At RX we have to disassemble packets, extract datas, then convert them to analog voice signals and send them to sound card (or phone)
6. All that must be done in a real time fashion cause we cannot waiting for too long for a vocal answer! (see QoS section)

```
                         Base architecture

Voice )) ADC - Compression Algorithm -  Assembling RTP in TCP/IP -----
                                                         ---->      |
                                                         <----      |
Voice (( DAC - Decompress. Algorithm -  Disass. RTP from TCP/IP  -----
```

4.2 Analog to Digital Conversion

This is made by hardware, typically by card integrated ADC.

Today every sound card allows you convert with 16 bit a band of 22050 Hz (for sampling it you need a freq of 44100 Hz for Nyquist Principle) obtaining a throughput of 2 bytes * 44100 (samples per second) = 88200 Bytes/s, 176.4 kBytes/s for stereo stream.

For VoIP we needn't a 22 kHz bandwidth (and also we needn't 16 bit!): next we'll see other coding used for it.

4.3 Compression Algorithms

Now that we have digital data we may convert it to a standard format that could be quickly transmitted.

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

- Voice bandwidth is 4 kHz, so sampling bandwidth has to be 8 kHz (for Nyquist).
- We represent each sample with 8 bit (having 256 possible values).
- Throughput is 8000 Hz *8 bit = 64 kbit/s, as a typical digital phone line.
- In real application mu−law (North America) and a−law (Europe) variants are used which code analog signal a logarithmic scale using 12 or 13 bits instead of 8 bits (see Standard ITU−T G.711).

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

It converts only the difference between the actual and the previous voice packet requiring 32 kbps (see 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, able to reach 2.5 kbps!!
```

This last protocols are the most important cause can guarantee a very low minimal band using source coding; also G.723.1 codecs have a very high MOS (Mean Opinion Score, used to measure voice fidelity) but attention to elaboration performance required by them, up to 26 MIPS!

4.4 RTP Real Time Transport Protocol

Now we have the raw data and we want to encapsulate it into TCP/IP stack. We follow the structure:

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

VoIP data packets live in RTP (Real−Time Transport Protocol) packets which are inside UDP−IP packets.

First, VoIP don't use TCP cause it is too heavy for real time application, so instead UDP (datagram) is used.

In UDP we cannot ordering packets in arrive time (which is a must in VoIP) because there isn't connection idea, each packet is independent from others (datagram concept); so we have to introduce a new protocol, such as RTP, able to manage this.

```
                    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             |
   |                             ....                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

- V indicates the version of RTP used
- P indicates the padding, a byte not used at bottom packet to reach the parity packet dimension
- X is the presence of the header extension
- CC field is the number of CSRC identifiers following the fixed header. CSRC field are used, for example, in conference case.
- M is a marker bit
- PT payload type

For a complete description of RTP protocol and all its applications see relative RFCs 1889 and 1890.

4.5 RSVP

There are also other protocols used in VoIP, like RSVP, that can manage Quality of Service (QoS).

RSVP is a signaling protocol that requests a certain amount of bandwidth and latency in every network hop that supports it.

For detailed info about RSVP see the RFC 2205

4.6 Quality of Service (QoS)

We said many times that VoIP applications require a real−time data streaming cause we expect an interactive data voice exchange.

Unfortunately, TCP/IP cannot guarantee this kind of purpose, it just make a "best effort" to do it. So we need to introduce tricks and policies that could manage the packet flow in EVERY router we cross.

So here are:

1. TOS field in IP protocol to describe type of service: high values indicate low urgency while more and more low values bring us more and more real−time urgency
2. Queuing packets methods:
    1. FIFO (First in First Out), the more stupid method that allows passing packets in arrive order.
    2. WFQ (Weighted Fair Queuing), consisting in a fair passing of packets (for example, FTP cannot consume all available bandwidth), depending on kind of data flow, typically one packet for UDP and one for TCP in a fair fashion.
    3. CQ (Custom Queuing), users can decide priority.
    4. PQ (Priority Queuing), there is a number (typically 4) of queues with a priority level each one: first, packets in the first queue are sent, then (when first queue is empty) starts sending from the second one and so on.
    5. CB−WFQ (Class Based Weighted Fair Queuing), like WFQ but, in addition, we have classes concept (up to 64) and the bandwidth value associated for each one.
3. Shaping capability, that allows to limit the source to a fixed bandwidth in:
    1. download
    2. upload
4. Congestion Avoidance, like RED (Random Early Detection).

For an exhaustive information about QoS see Differentiated Services at IETF.

4.7 H323 Signaling Protocol

H323 protocol is used, for example, by Microsoft Netmeeting to make VoIP calls.

This protocol allow a variety of elements talking each other:

1. Terminals, clients that initialize VoIP connection. Although terminals could talk together without anyone else, we need some additional elements for a scalable vision.
2. Gatekeepers, that essentially operate:
    1. address translation service, to use names instead IP addresses
    2. admission control, to allow or deny some hosts or some users
    3. bandwidth management
3. Gateways, points of reference for conversion TCP/IP − PSTN.
4. Multipoint Control Units (MCUs) to provide conference.
5. Proxies Server also are used.

h323 allows not only VoIP but also video and data communications.

Concerning VoIP, h323 can carry audio codecs G.711, G.722, G.723, G.728 and G.729 while for video it supports h261 and h263.

More info about h323 is available at Openh323 Standards, at this h323 web site and at its standard description: ITU H−series Recommendations.

You can find it implemented in various application software like Microsoft Netmeeting, Net2Phone, DialPad, ... and also in freeware products you can find at Openh323 Web Site.

---

5. Requirement

5.1 Hardware requirement

To create a little VoIP system you need the following hardware:

1. PC 386 or more
2. Sound card, full duplex capable
3. a network card or connection to internet or other kind of interface to allow communication between 2 PCs

All that has to be present twice to simulate a standard communication.

The tool above are the minimal requirement for a VoIP connection: next we'll see that we should (and in Internet we must) use more hardware to do the same in a real situation.

Sound card has be full duplex unless we couldn't hear anything while speaking!

As additional you can use hardware cards (see next) able to manage data stream in a compressed format (see Par 4.3).

5.2 Hardware accelerating cards

We can use special cards with hardware accelerating capability. Two of them (and also the only ones directly managed by the Linux kernel at this moment) are the

1. Quicknet PhoneJack
2. Quicknet LineJack

Quicknet PhoneJack is a sound card that can use standard algorithms to compress audio stream like G723.1 (section 4.3) down to 4.1 Kbps rate.

It can be connected directly to a phone (POTS port) or a couple mic−speaker.

It has a ISA or PCI connector bus.

Quicknet LineJack works like PhoneJack with some addition features (see next).

For more info see Quicknet web site.

5.3 Hardware gateway cards

Quicknet LineJack can be connected to a PSTN line allowing VoIP gateway feature.

Then you'll need a software to manage it (see after).

5.4 Software requirement

We can choose what O.S. to use:

1. Win9x
2. Linux

Under Win9x we have Microsoft Netmeeting, Internet Phone, DialPad or others or Internet Switchboard (from Quicknet web site) for Quicknet cards.

Also you can use free software you download from OpenH323.

Under Linux we only have free software from OpenH323 web site: simph323 or ohphone that can also work with Quicknet accelerating hardware.

Attention: all Openh323 source code has to be compiled in a user directory (if not it is necessary to change some environment variable). You are warned that compiling time could be very high and you could need a lot of RAM to make it in a decent time.

5.5 Gateway software

To manage gateway feature (join TCP/IP VoIP to PSTN lines) you need some kind of software like this:

- Internet SwitchBoard for Windows systems also acting as a h323 terminal;
- PSTNGw for Linux and Windows systems you download from OpenH323.

5.6 Gatekeeper software

You can choose as gatekeeper:

1. Opengatekeeper, you can download from opengatekeeper web site for Linux and Win9x.
2. Openh323 Gatekeeper (GK) from here.

5.7 Other software

In addition I report some useful software h323 compliant:

- Phonepatch, able to solve problems behind a NAT firewall. It simply allows users (external or internal) calling from a web page (which is reachable from even external and internal users): when web application understands the remote host is ready, it calls (h323) the source telling it all is ok and communication can be established. Phonepatch is a proprietary software (with also a demo version for no more than 3 minutes long conversations) you download from here.

6. Cards setup

Here we see how to configure special hardware card in Linux and Windows environment.

6.1 Quicknet PhoneJack

As we saw, Quicknet Phonejack is a sound card with VoIP accelerating capability. It supports:

- G.711 normal and mu/A−law, G.728−9, G.723.1 (TrueSpeech) and LPC10.
- Phone connector (to allow calling directly from your phone) or
- Mic & speaker jacks.

Quicknet PhoneJack is a ISA (or PCI) card to install into your Pc box. It can work without an IRQ.

Software installation

Under Windows you have to install:

1. Card driver
2. Internet Switchboard application

all downloadable from Quicknet web site

After Switchboard has been installed, you need to register to Quicknet to obtain full capability of your card.

When you pick up the phone Internet Switchboard wakes up and waits for your calling number (directly entered from your phone), you can:

1. enter an asterisk, then type an IP number (with asterisks in place of dot) with a # in the end
2. type directly a PSTN phone number (with international prefix) to call a classic phone user. In this case you need a registration to a gateway manager to which pay for time.
3. enter directly a quick dial number (up to 2 digits) you have previously stored which make a call (IP or PSTN).

Internet Swichboard is h323 compatible, so if you can use, for example, Microsoft Netmeeting at the other end to talk.

In place of Internet Switchboard you can use openh323 application openphone (using GUI) or ohphone (command line).

Under Linux you have to install:

1. Card driver, from Quicknet web site. After downloaded you have to compile it (you must have a /usr/src/linux soft or hard link to your Linux source directory): type make for instructions.
2. Application openphone or ohphone.
3. If you are a developer you can use SDK to create your own application (also for Windows).

Settings

With Internet Switchboard (and with other application) you can:

1. Change compression algorithm preferred
2. Tune jitter delay
3. Adjust volume
4. Adjust echo cancellation level.

6.2 Quicknet LineJack

This card is very similar to the previous, it supports also gateway feature.

We only notice that we have to download PSTNGx application (for Linux and Windows) or we use Internet Switchboard to gateway feature.

---

7. Setup

In this chapter we try to setup VoIP system, simple at first, then more and more complex.

7.1 Simple communication: IP to IP

```
A (Win9x+Sound card)   − − −   B (Win9x+Sound card)

     192.168.1.1       − − −        192.168.1.2


             192.168.1.1 calls 192.168.1.2.
```

A and B should:

1. have Microsoft Netmeeting (or other software) installed and properly configured.
2. have a network card or other kind of TCP/IP interface to talk each other.

In this kind of view A can make a H323 call to B (if B has Netmeeting active) using B IP address. Then B can answer to it if it wants. After accepting call, VoIP data packets start to pass.

7.2 Using names

If you use Microsoft Windows in a lan you can call the other side using NetBIOS name. NetBIOS is a protocol that can work (stand over) with NetBEUI low level protocol and also with TCP/IP. It is only need to call the "computer name" on the other side to make a connection.

```
          A            − − −            B

     192.168.1.1       − − −       192.168.1.2

        John           − − −          Alice
```

```
                    John calls Alice.
```

This is possible cause John call request to Alice is converted to IP calling by the NetBIOS protocol.

The above 2 examples are very easy to implement but aren't scalable.

In a more big view such as Internet it is impossible to use direct calling cause, usually, the callers don't know the destination IP address. Furthermore NetBIOS naming feature cannot work cause it uses broadcast messages, which typically don't pass ISP routers .

7.3 Internet calling using a WINS server

The NetBIOS name calling idea can be implemented also in a Internet environment, using a WINS server: NetBIOS clients can be configured to use a WINS server to resolve names.

PCs using the same WINS server will be able to make direct calling between them.

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

                 Internet communication
```

A, B, C and D are in different subnets, but they can call each other in a NetBIOS name calling fashion. The needed is that all are using S as WINS Server.

Note: WINS server hasn't very high performance cause it use NetBIOS feature and should only be used for joining few subnets.

7.4 A big problem: the masquering.

A problem of few IPs is commonly solved using the so called masquering (also NAT, network address translation): there is only 1 IP public address (that Internet can directly "see"), the others machines are "masqueraded" using all this IP.

```
          A  - - -

          B  - - -   Router with NAT  - - -  Internet

          C  - - -


                      This doesn't work
```

In the example A,B and C can navigate, pinging, using mail and news services with Internet people, but they

CANNOT make a VoIP call. This because H323 protocol send IP address at application level, so the answer will never arrive to source (that is using a private IP address).

Solutions:

- there is a Linux module that modifies H323 packets avoiding this problem. You can download the module here. To install it you have to copy it to source directory specified, modify Makefile and go compiling and installing module with "modprobe ip_masq_h323". Unfortunately this module cannot work with ohphone software at this moment (I don't know why).

```
A  - - -   Router with NAT

B  - - -         +           - - -  Internet

C  - - -  ip_masq_h323 module


                This works
```

- There is a application program that also solves this problem: for more see Par 5.7

```
A  - - -

B  - - -    PhonePatch   - - -  Internet

C  - - -


                This works
```

7.5 Using Linux

With Linux (as an h323 terminal) you can experiment everything done before except standing behind a router NAT with ip_masq_h323 module cause ohphone doesn't work with it at this moment: you have to use phonepatch application instead.

7.6 Setting up a gatekeeper

You can also experiment gatekeeper feature

```
Example

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

                   Gatekeeper configuration
```

1. Hosts A,B and C have gatekeeper setting to point to D.
2. At start time each host tells D own address and own name (also with aliases) which could be used by a caller to reach it.

3. When a terminal asks D for an host, D answers with right IP address, so communication can be established.

We have to notice that the Gatekeeper is able only to solve name in IP address, it couldn't join hosts that aren't reachable each other (at IP level), in other words it couldn't act as a NAT router.

You can find gatekeeper code here: openh323 library is also required.

Program has only to be launch with −d (as daemon) or −x (execute) parameter.

In addition you can use a config file (.ini) you find here.

7.7 Setting up a gateway

As we said, gateway is an entity that can join VoIP to PSTN lines allowing us to made call from Internet to a classic telephone. So, in addition, we need a card that could manage PSTN lines: Quicknet LineJack does it.

>From OpenH323 web site we download:

1. driver for Linejack
2. PSTNGw application to create our gateway.

If executable doesn't work you need to download source code and openh323 library, then install all in a home user directory.

After that you only need to launch PSTNGw to start your H323 gateway.

---

8. Bandwidth consideration

>From all we said before we noticed that we still have not solved problems about bandwidth, how to create a real time streaming of data.

We know we couldn't find a solution unless we enable a right real−time manager protocol in each router we cross, so what do we can do?

First we try to use a very (as more as possible) high rate compression algorithms (like LPC10 which only consumes a 2.5 kbps bandwidth, about 313 bytes/s).

Then we starts classify our packets, in TOS field, with the most high priority level, so every router help us having urgently.

Important: all that is not sufficient to guarantee our conversation would always be ok, but without an great infrastructure managing shaping, bandwidth reservation and so on, it is not possible to do it, TCP/IP is not a real time protocol.

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

We finally have to notice a thing: also the so called guaranteed services like PSTN line could not manage all clients they have: for example a GSM call is not able to manage more that some hundred or some thousand of clients.

Anyway for a starting service, limited to few users, VoIP can be a valid alternative to classic PSTN service.

9. Useful links

- Voxilla
- Linux Telephony
- International Communication Union
- Quicknet Web site
- Open H323 web site
- Speak Freely
- Cisco Systems