

1 Voice over Internet Protocol (VoIP)

1.1 Introduction to VoIP

VoIP (Voice over Internet Protocol) is a method of voice transport per Internet Protocol used in packet oriented networks. VoIP is defined in the recommendations ITU-T H.32x and RFC 2443. VoIP can use accelerating hardware to achieve this purpose and can also be used in a PC environment.

2 Principles of VoIP

VoIP works like that. First the A/DC (Analog to Digital Converter) to convert analog voice to digital signals. Now the bits have to be compressed in a good format for transmission. Then we have to insert our voice packets in data packets using a real-time protocol (RTP over UDP over IP). For signaling between customers terminal unit we need a signaling protocol ITU-T H.323 or SIP (Session Initiate Protocol). At Rx site we have to disassemble packets, extract data, then convert them to analog voice signals and send them to sound card (or phone). All that must be done in a real time fashion cause we cannot waiting for too long a vocal answer. On voice transport via packet oriented network could be respect requirements of QoS (Quality of Service) See below article 3.

2.1 Analog to digital conversion

For conversion analog voice to digital signal is very popular use standard PC soundcards or similar A/D converters. This cards sampling with 16bit a band of 22,050 kHz with sampling freq 44,100 kHz like that throughput of 2 bytes * 44100 (samples per second) = 88200 Bytes/s, 176.4 kBytes/s for stereo stream.

2.2 Algorithm for processing of digital signal and Compression

For VoIP we needn't such a throughput (176 kBytes/s) to send voice packet. Digitalized voice data we can compress it, route it, convert it to a new better format that could be quickly transmitted. The survey of format is in Tab. 1.

Processing of sampled signal			
Formats	Note	Speed [kbit/s]	Recommendation
PCM (mu-law, A-law)	Uncompressed data	64 kbit/s (8 kHz * 8bit)	G.711
ADPCM	Transmit of difference between samples	32	G.721 G.726 G.727
LD-CELP	linear prediction	16	G.728
CS-ACELP	linear prediction	8	G.729
MP-MLQ	True speech compression	6,3	G.723.1
LPC-10	linear prediction	2,5	-
GSM	VSELP based	9,6 – 14,4	-

Tab. 1 - Compression in VoIP

2.3 Transport and signaling protocol in VoIP network

For telephony over IP based network we need not only transmit audio data also we need transmit a drive information about connection and formation and unformation of connection. The several voice data are transmitted mostly via RTP protocol. Mostly applied signaling (driving) protocol are:

- H.323 by ITU-T
- SIP (Session Initiate Protocol) by IETF (RFC 2443)
- MGCP (Media Gateway to Media Controller Protocol) by Cisco and Bellcore

2.3.1 RTP (Real Time Transport Protocol)

The sampled and compressed data are encapsulated into TCP/IP stack. VoIP data packets live in RTP (Real-Time Transport Protocol) or RSVP packets which are inside UDP-IP packets.

RM-OSI layer model		
Num. of layer	Name of Layer	Protocol
7	Application	VoIP data, H.323, SIP or MGCP
6	Presentation	
5	Session	RTP
4	Transport	UDP
3	Network	IP
2	Link	Frame (Eth. ATM..)
1	Physical	Medium

Tab. 2 -RM-OSI reference model

Firstly, VoIP doesn't use TCP because it is too heavy for real time applications, so instead a UDP (datagram) is used.

Secondly, UDP has no control over the order in which packets arrive at the destination or how long it takes them to get there (datagram concept). Both of these are very important to overall voice quality (how well you can understand what the other person is saying) and conversation quality (how easy it is to carry out a conversation). RTP solves the problem enabling the receiver to put the packets back into the correct order and not wait too long for packets that have either lost their way or are taking too long to arrive (we don't need every single voice packet, but we need a continuous flow of many of them and ordered).

										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																		
Time stamp																																
Synchronization source (SSRC) identifier																																
Contributing source (CSRC) identifiers																																
.....																																

Tab. 3 - Payload of Real Time Transport Protocol

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

2.3.2 RSVP (Resource ReSerVation Protocol)

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.

2.3.3 ITU-T H.323 signaling protocol

H.323 protocol is not a separate protocol. H.323 covers plenty of protocol for transmit voice or video over IP on different transport networks like ATM, ISDN, PSTN. See below tab.

	H.320	H.321	H.322	H.323	H.324
Approval	1990	1995	1995	1996	1996
Network	ISDN-BRA	ISDN-PRA ATM LAN	Guaranteed Bandwidth Packet switched networks	Non- Guaranteed bandwidth packet switched networks Ethernet	PSTN or POTS analog phone system
Video	H.261 H.263	H.261 H.263	H.261 H.263	H.261 H.263	H.261 H.263
Audio	G.711 G.722 G.728	G.711 G.722 G.728	G.711 G.722 G.728	G.711 G.723 G.722 G.728 G.728	G.723
Muxing	H.221	H.221	H.221	H.225.0	H.223
Multipoint	H.231 H.243	H.231 H.243	H.242 H.243	H.323	
Control	H.230 H.242	H.242	H.231 H.243	H.245	H.245
Comm. Interface	I.400	AAL	I.400 TCP/IP	I.400 TCP/IP	V.34 modem
Data	T.120	T.120	T.120	T.120	T.120

Tab. 4 -Over view of H.32x protocol

Concerning VoIP, H.323 can carry audio codec G.711, G.722, G.723, G.728 and G.729 while for video it supports H.261 and H.263. H.323 protocol is used, for example, by Microsoft Netmeeting to make VoIP calls. The schema of H.32x network is on the figure 2.

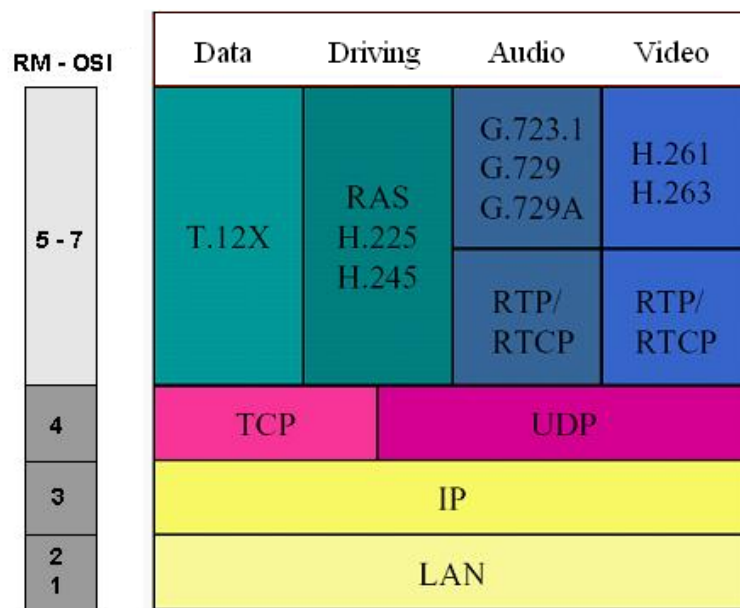


Fig. 1 - H.323 model

The H.323 encapsulated 4 primary components:

- **terminal** - Clients that initialize VoIP connection. Although terminals could talk together without anyone else, we need some additional elements for a scalable vision.
- **gateway** - points of reference for conversion TCP/IP – PSTN. Serves like as translator of different protocols.
- **MCU (Multipoint Control Units)**- to provide conference (3 or more terminals)
- **gatekeeper** - that essentially operate:
 - address translation service, to use names instead IP addresses
 - admission control, to allow or deny some hosts or some users
 - bandwidth management

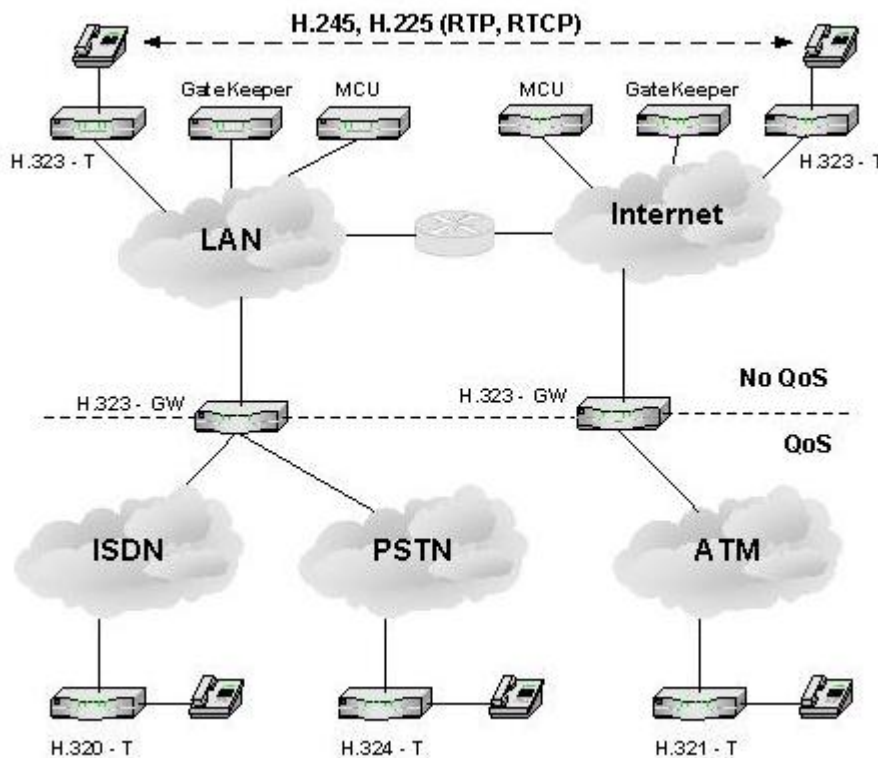


Fig. 2 -Network based on H.32x

2.3.4 SIP (Session Initiate Protocol) signaling protocol

Session Initiate Protocol specified by IETF (The Internet Engineering Task Force) in the recommendation RFC 2443, provides the necessary protocol mechanisms so that end systems and proxy servers can provide services:

- call forwarding
- callee and calling ``number" delivery, where numbers can be any (preferably unique) naming scheme;
- *personal mobility*, i.e., the ability to reach a called party under a single, location-independent address even when the user changes terminals;
- terminal-type negotiation and selection: a caller can be given a choice how to reach the party, e.g., via Internet telephony, mobile phone, an answering service, etc.;
- terminal capability negotiation;
- caller and callee authentication;
- blind and supervised call transfer;
- invitations to multicast conferences.

Extensions of SIP to allow third-party signaling (e.g., for click-to-dial services, fully meshed conferences and connections to multipoint control units (MCUs), as well as mixed modes and the transition between those) are available.

SIP addresses users by an email-like address and re-uses some of the infrastructure of electronic mail delivery such as DNS MX records or using SMTP EXPN for address expansion. SIP addresses (URLs) can also be embedded in web pages. SIP is addressing-neutral, with addresses expressed as URLs of various types such as SIP, H.323 or telephone (E.164).

SIP can also be used for signaling Internet real-time fax delivery. This requires no major changes. Fax might be carried via RTP, TCP (e.g., the protocols discussed in the Internet fax WG) or other mechanisms.

SIP is independent of the packet layer and only requires an unreliable datagram service, as it provides its own reliability mechanism. While SIP typically is used over UDP or TCP, it could, without technical changes, be run over IPX, or carrier pigeons, frame relay, ATM AAL5 or X.25, in rough order of desirability.

3 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:

- TOS field in IP protocol [3] to describe type of service: high values indicate low urgency while more and more low values bring us more and more real-time urgency
- Queuing packets methods:
 - FIFO (First in First Out), the more stupid method that allows passing packets in arrive order.
 - 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.
 - CQ (Custom Queuing), users can decide priority.
 - 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.
 - 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.
- Shaping capability, that allows to limit the source to a fixed bandwidth in:
 - download
 - upload
- Congestion Avoidance, like RED (Random Early Detection).

4 Simple solution of VoIP without especially HW for testing transmit of voice over IP based network

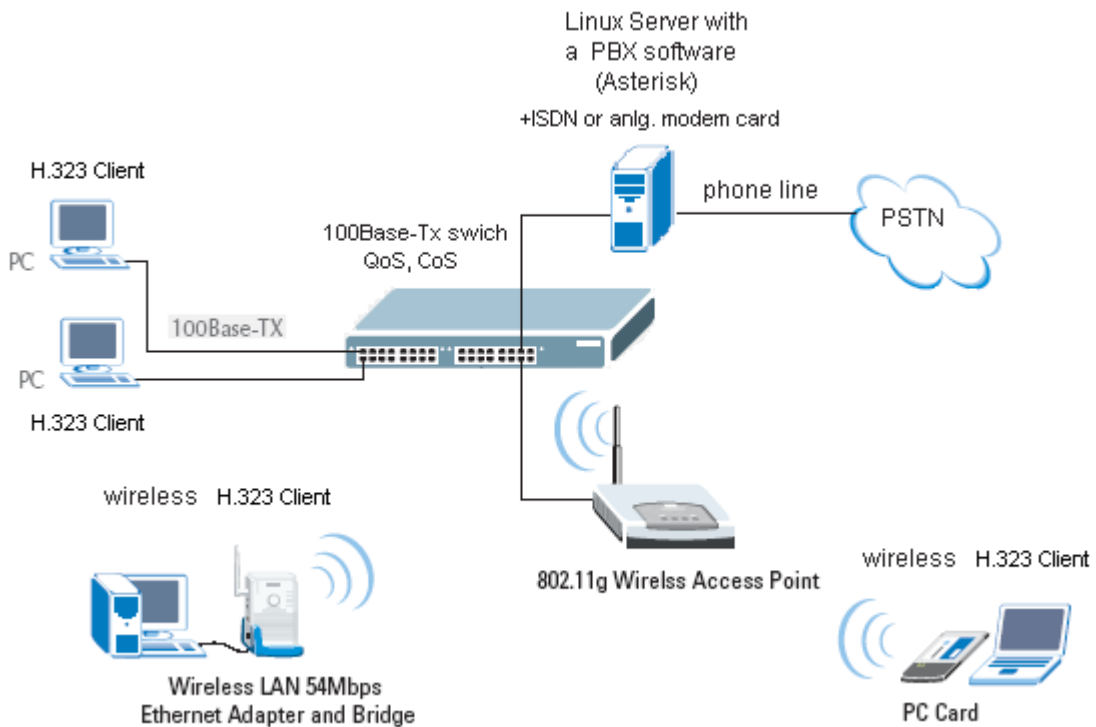


Fig. 3 -Simple solution of VoIP on switched network with H.323 clients

The ground of this solutions is server with an ISDN modem card and Open source Asterisk software. Asterisk provides all of the features you would expect from a PBX (Private Branch Exchange) and more. Asterisk does voice over IP in protocols (SIP and H.323), and can interoperate with almost all standards-based telephony equipment using relatively inexpensive hardware. Clients use a software with a protocol H.323 like a MS-Netmeeting or SIP protocol based clients like a Xlite, SjPhone. Clients are connected to Ethernet switch over 802.11x Wireless Access point or direct on switch with Ethernet card 10/100Base-Tx.

5 Introduction to measuring of QoS parameters

For prosecution of Voice over Internet protocol (VoIP) we need network which execute parameters Quality of Service (QoS). Because the IP networks are packets oriented without construction of connection like in switched oriented network, where we have guaranteed specific delay and bandwidth for services. In packet oriented networks we have not beforehand specific delay and bandwidth etc.

Hence we have to measure below stated parameters for test quality of IP networks for real-time depend services like as VoIP. QoS parameters are defined in recommendation ITU-T I.350 and ITU-T Y.1541. The parameters are:

- Loss rate
- Throughput
- Delay
- Distribution of delay
- Distribution of delay variation

First we will test parameters QoS. Second we will test himself voice transmission over IP network with different voice codec. For test of voice we use Mean Opinion Score (MOS) test.

5.1 Test method of QoS

For Test of QoS i seleceted a small GNU utilities RUDE [9] and CRUDE [9]. which can emulate different flows of packets which are similar to VoIP flows. With this utilities we can simulate different traffic on the network and watch reactions of network. For processing of product i selected utilities QoSplot [10] and GnuPlot [11].

5.2 About utilities

5.2.1 RUDE (Real-time UDP Data Emitter)

As the name says, this piece of program can generate UDP traffic to the network. In configuration file we can write definition about flow for example:

- Number of flow
- Numbers of packets per second in flow
- Numbers of Bytes in packets
- Destination Address
- Destination Ports

5.2.2 CRUDE (Collector for RUDE)

CRUDE is the receiver and logging utility for flows that are generated with the RUDE tool. In the default operation mode CRUDE prints the information to standard output, so you'll have to redirect it to file if you like to process it later on.

5.2.3 QoS-plot

Qosplot is a tool that takes as input a set of text log files created by CRUDE [9] and produces as output the data and command files for gnuplot [11], to plot diagrams depicting the QoS characteristics.

5.2.4 GNU-Plot

Gnuplot is a command-line driven interactive function plotting utility for UNIX, DOS, VMS, and many other platforms. It was originally intended as graphical program which would allow to visualize mathematical functions and data.

5.3 Measuring with GNU utilities

The principle of measuring QoS parameters is on figure 4. On the first side is PC1 with CRUDE. CRUDE emit defined flows of packets to networks. On the second side is PC2 with RUDE which receive packets from network. Information from output of RUDE are passed to Qosplot and GNU-plot programs.

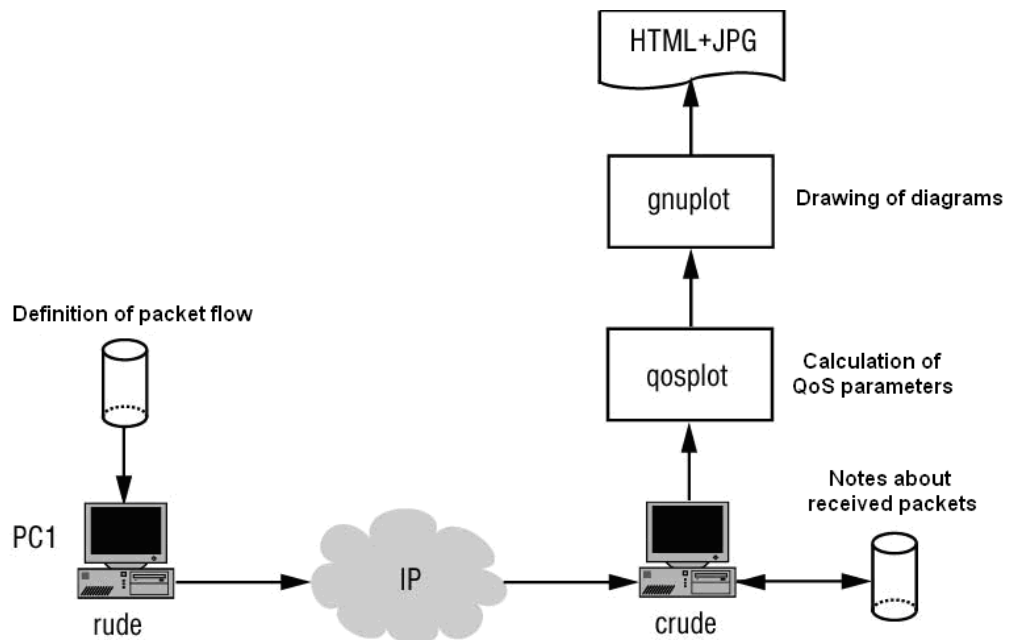


Fig. 4 -Principle of measuring QoS with GNU based program

5.3.1 The fidelity of Measuring

The fidelity of measuring depend on synchronization of time between sides of measuring [12]. The CRUDE/RUDE can use synchronization of time from:

- Autonomus celsium clock on all sides (very expensive)
- GPS receiver (available)
- LAN (simpler but less fidelity)

We will use the simplest and cheap solution of LAN synchronization via The Network Time Protocol.

5.4 Solution of QoS test

In QoS test i used a network equipment available in Lab of Signal processing on faculty of electrical engineering in Zagreb. In Test i used:

- WiFi 802.11b D-link 900+ Access point
- WiFi 802.11b D-link 122 USB network adapter
- Swich HP Procurve 2626 1G/100 with supported QoS, CoS
- Two PCs with linux OS system
- UTP Cat-5 patch cables

On the figure 5 you can see the order of lab network for test QoS parameters needy for time depend services.

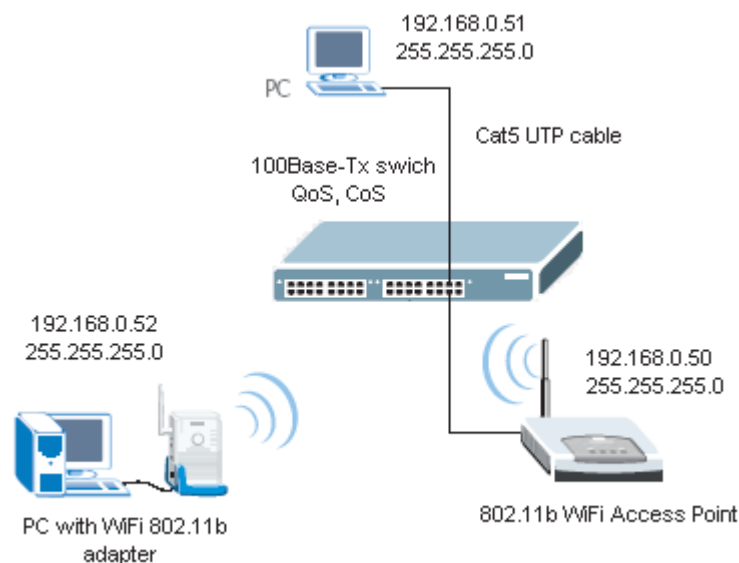


Fig. 5 - Elementary laboratory (bridged) network for test QoS parameters

5.4.1 Simulation of VoIP flows

With simulation utilities like CRUDE/RUDE we can not hollow exactly simulate VoIP flows (with signalization etc.) but we can good approximate a real traffic. In Tab. 5 you can see the most applied codec for VoIP, bitrates and number of Samples per period.

<i>Codec</i>	<i>Speed [kbit/s]</i>	<i>Delay of Algorithh [ms]</i>	<i>Alg.(modulation)</i>	<i>samples/Time</i>
G.711	64	0,125	PCM	8bits/125 s
G.723.1	5,3	37,5	MP-MLQ	20Bytes/30ms
G.723.1	6,3	37,5	ACELP	24Bytes/30ms
G.726	32	0,125	ADPCM	2-5bits/125 s
G.728	16	0,625	LD-CELP	16bits/625 s
G.729	8	15	CS-ACELP	10Bytes/10ms

Tab. 5 - Summary of codecs in VoIP

For Test i choose most apply G.723.1 codec with rate 6,3 kbit/s. This flow we can approximate with RUDE/CRUDE utilities.

5.4.2 Set-up RUDE/CRUDE for test

RUDE have a configuration file for example `config.cfg`. In this file we can specify parameters of flow. See below Tab. 6

```
## FLOW 1: (flow ID = 10 with address of flow 3002) G.723.1 6,3 kbit/s
##
## Starts 1 second after the START time with following parameters:
##   33 packets/second with 24 bytes/packet = 792bytes/sec = 6,34 kbit/s
##   from IP address 192.168.0.52 on port 10001
## Ends on 180s after start
1000 0010 ON 3002 192.168.0.51:10001 CONSTANT 24 33
180000 0030 OFF
```

Tab. 6 -Example configuration of flows in RUDE config file for G.723.1 6,3 kbit/s

The time of test we set to 3 min, while in telecommunication was measured average time of calls to 2 - 3 min. If we measured error rate of packets. We would set longer time of test. For example 1h or more. In the tab. 3 you can see the result of test for flow G.723.1 6,3 kbit/s. The higher speed of flow (round 14 kbit/s) is caused payloads of IP/UPD packets (40Bytes).

Behind the test we measured a loss of 3 packets and delay just about 180ms. This relative big delay is caused using of WiFi equipment. The WiFi use an air-interface which could be disturbanced with many of noises hence data must be very good protected with scrambling, Reed Saalamon, CRC, etc. This operations need a lot of time. The total delay is a sum of all delay. It mean sum of algorithm delay, transport delay

Filename, stream	Approximation of flow G.723.1
Packets sent	5911
Packets received	5908
Packets lost	3
Loss rate (IPLR) [-]	$0,17 \cdot 10^{-3}$
Max. throughput [b/s]	18304,00
Min. throughput [b/s]	0,00
Average throughput [b/s]	13660,98
Max. delay [ms]	183058,28
Min. delay [ms]	183057,61
Average delay (IPTD) [ms]	183057,65
Max. delay variation [ms]	0,54
Min. delay variation [ms]	-0,03
Average delay variation (AvIPDV) [ms]	0,00
IP Packet Error Ratio, IPER	Un-measured

Tab. 7 -Product of test

On the second side we receive data via CRUDE and then plot it via qosplot and gnuplot. See follow diagrams:

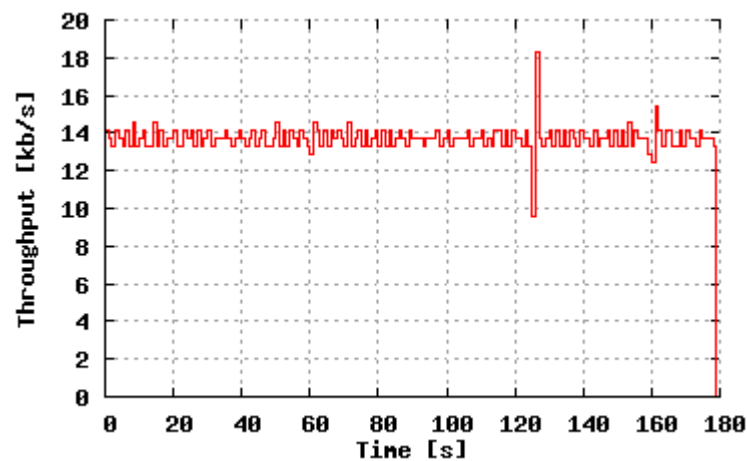


Fig. 6 - Throughput of flow

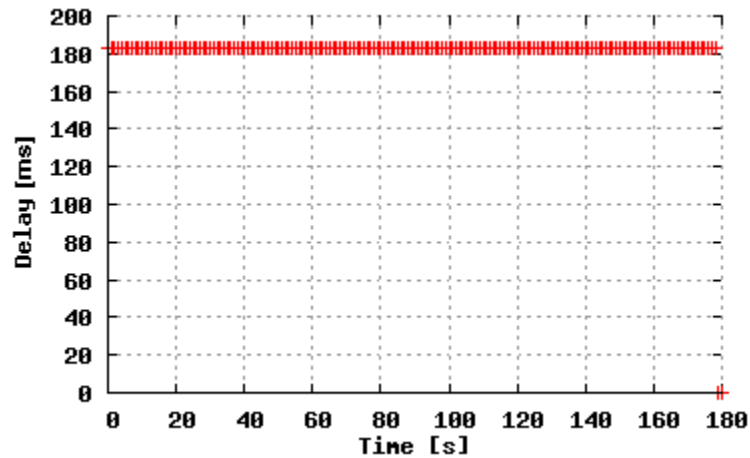


Fig. 7 - Delay of flow

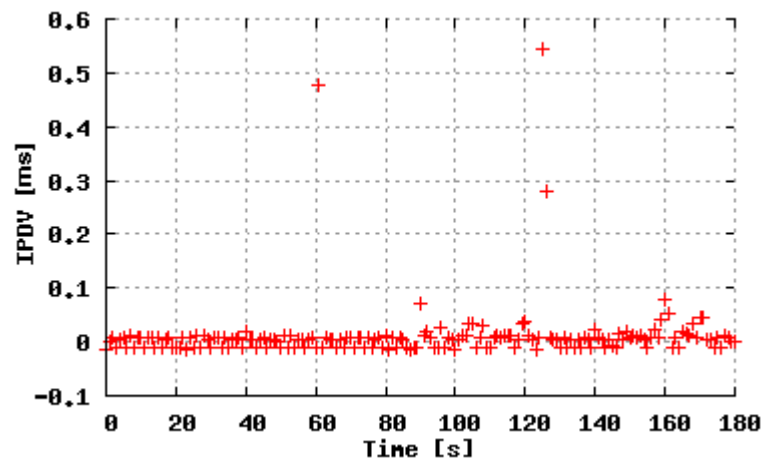


Fig. 8 - IP packet transfer delay variation (IPDV) of flow

Through recommendation ITU-T Y.1541 are defined availability of service [7]. We can compare the measured ed values in tab. 3 and diagrams 7 – 9 with the recommendation ITU-T Y.1541. (See below tab. 8). After compare we can say that our test network meets in all parameters in all classes via Recommendation of QoS ITU-T Y.1541.

Note: in the further study would be interesting to try power test with more differ flows together. And test pull on the delay and lost of packets.

Recommendation ITU-T Y.1541 IP QoS Classes and Objectives							
		QoS Classes and Objectives					
Network Performance Parameter	Nature of Objective	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5
IP packet transfer delay (IPTD)	Upper bound on the mean IPTD	100ms	400ms	100ms	400ms	1 s	Unspec.
IP packet transfer delay variation (IPDV)	Upper bound on the 1-10 ⁻³ quantile of IPTD minus the minimum IPTD	50ms	50ms	U	U	U	Unspec.
IP packet loss ratio (IPLR)	Upper bound on the packet loss probability	1.10 ⁻³	1.10 ⁻³	1.10 ⁻³	1.10 ⁻³	1.10 ⁻³	Unspec.
IP packet error ratio (IPER)	Upper bound	1.10 ⁻⁴	Unspec				

Tab. 8 - Recommendation QoS services

6 Voice network with Asterisk PBX

Now we will unveil a cheap, powerful and modular solution of local Voice network with Asterisk PBX described in chapter 4 on the figure 3. The Hw possibilities of Asterisk PBX are placed on the figure 9. The network we can portion to:

- Part of Exchange
- Part of VoIP clients (Sw, Hw phones)
- Part of NT (Network termination) ISDN connections
- Part of local interconnected ISDN phones (optional)
- Part of local POTS phones plug into FXS (Foreign Exchange Station) card (optional)

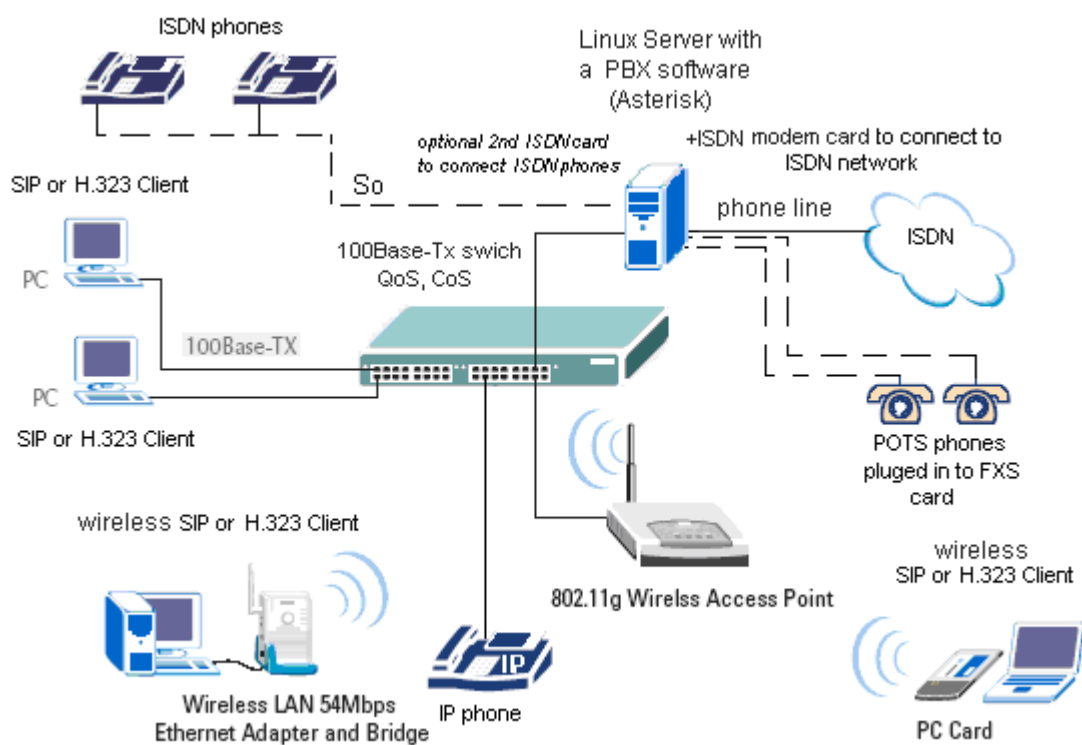


Fig. 9 -Full Asterisk PBX exchange with VoIP, ISDN and POTS clients

6.1 Part of PBX exchange configuration and functions

Asterisk is a complete PBX in software. It runs on Linux and provides all of the features you would expect from a PBX and more. Asterisk does voice over IP in three protocols, and can

interoperate with almost all standards-based telephony equipment using relatively inexpensive hardware.

Asterisk provides Voicemail services with Directory, Call Conferencing, Interactive Voice Response, Call Queuing. It has support for three-way calling, caller ID services, ADSI, SIP and H.323 (as both client and gateway). Check the Features section for a more complete list. Asterisk needs no additional hardware for Voice over IP. For interconnection with digital and analog telephony equipment, Asterisk supports a number of hardware devices, most notably all of the hardware manufactured by Asterisk's sponsors, [Digium™](#). Digium has single and quad span T1 and E1 interfaces for interconnection to PRI lines and channel banks as well as a single port FXO card and a one to four-port modular FXS and FXO card. Also supported are the Internet Line Jack and Internet Phone Jack products from Quicknet.

Asterisk supports a wide range of TDM protocols for the handling and transmission of voice over traditional telephony interfaces. Asterisk supports US and European standard signaling types used in standard business phone systems, allowing it to bridge between next generation voice-data integrated networks and existing infrastructure. Asterisk not only supports traditional phone equipment, it enhances them with additional capabilities.

Using the Inter-Asterisk eXchange (IAX™) Voice over IP protocol, Asterisk merges voice and data traffic seamlessly across disparate networks. While using Packet Voice, it is possible to send data such as URL information and images in-line with voice traffic, allowing advanced integration of information.

6.1.1 Hardware used for building of Asterisk

We can use common PC machine, but power of PC is depend on number of calls. For small area with a “few” users who have generated a small traffic we can use older inefficient hardware. For example power of machines depending on number of callers we can see below in tab. 9

<i>Machine</i>	<i>Concurrent calls</i>	<i>Loading of CPU</i>	<i>Codec</i>
Pentium II, 300MHz, 128 MB	28	30%	G.729
Xeon 2.4 GHz, 1 GB	500	100%	G.729
Pentium 4, 2.6 GHz, 1 GB RAM	200	100%	G.729

Tab. 9 - Number of concurrent calls on different machines

For building an exchange I had in hand PII machine with 256 MB RAM, 40 GB HDD with Linux Redhat 9. After Asterisk installation I tested capability of this PBX exchange which have been running on this machine. (installation of PBX is described in next subchapter). The exchanges are assessed to number of concurrent calls in time. In Tab. 10 is described number of new calls during one second, number of concurrent calls, retranslate calls, successful calls, failed calls, unaccepted calls and average response on this machine.

Power test PII 400, 256 MB RAM with average time of calls $tcs=10\text{ s}$, time of test $tst=120\text{ ms}$, G.729 codec									
New calls during 1 sec.	Concurrent calls	Total Calls created	Retranslate calls	Successfull calls	Failed calls	Timeout calls	Unaccepted calls	Average response [ms]	CPU [%]
1	11	121	0	121	0	0	0	10	1
2	21	242	0	242	0	0	0	11	2
3	31	363	0	363	0	0	0	11	3
5	52	605	0	60	0	0	0	15	32
6	62	721	0	721	0	0	0	25	82
7	210	610	2306	369	2444	64	38	200	99

Tab. 10 -Power test of asterisk PBX on PII 400 MHz machine

Note:

For measuring loading of PBX I found interesting GNU utilities SIPp. SIPp is a performance testing tool for the SIP protocol. It includes a few basic Sip Stone user agent scenarios UAC (User Agent Client) and UAS (User Agent Server) and establishes and releases multiple calls with the INVITE and BYE methods. It can also read XML scenario files describing any performance testing configuration. It features the dynamic display of statistics about running tests (call rate, round trip delay, and message statistics), periodic CSV statistics dumps, TCP and UDP over multiple sockets or multiplexed with retransmission management, regular expressions and variables in scenario files, and dynamically adjustable call rates.

For testing I used 10 s long period of calls and number of incoming calls was from 1 to 7 new calls during 1 s. For our PBX is maximum 6 new incoming calls during 1s. This traffic generate 62 concurrent calls and PBX work without failed and retranslate calls yet. Elevation number of new call makes overload of PBX see tab. 11. The principle of test is on the figure 11. From PC are sending above defined SIP datagram to PBX on number 600. This number is reserved for demo test of PBX. (for our test we can use voice mailbox number 1000 too). The command for SIP flows is next: `$. /sipp [IP] -sn UAC -s600 -r[n] -d10`

Where IP is IP of target machine, **UAC** defined User Agent Client, **s** is a call number, **r** defined number **n** of calls during 1 s, **d** is duration of call in seconds.

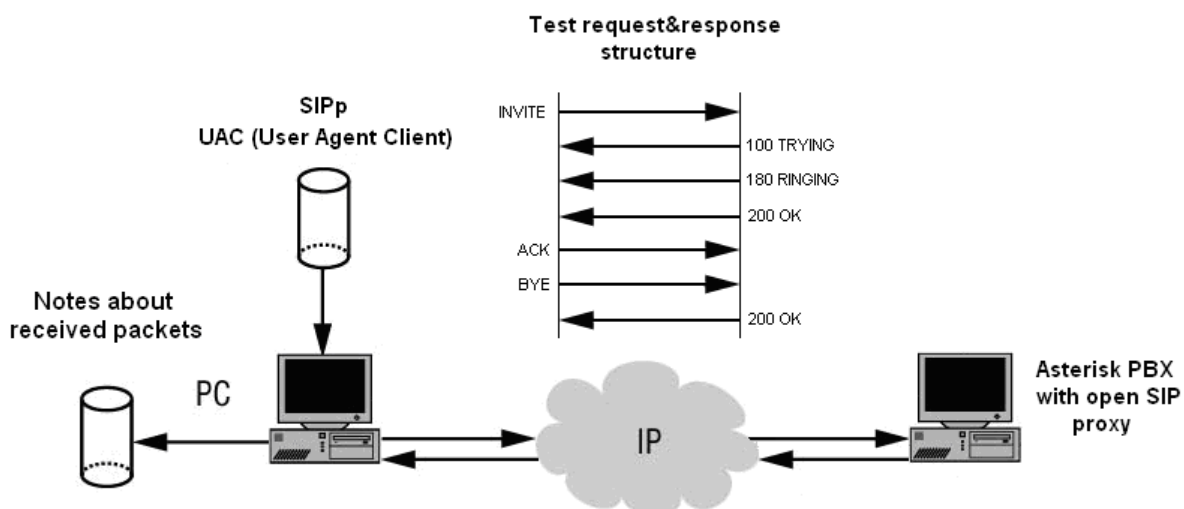


Fig. 10 -Principles of measuring PBX loading

6.1.2 Installation and configuration

The source code is available from ftp.digium.com as a tarball. After download and unpack the tarball we can run `make` to build it and then `make install`.

After installation we have to optional configure the PBX. Configuration files are stored in standard directory for configurations `/etc/asterisk/`. The detail description of configuration PBX is in next sub-chapter. When we have successful configured the PBX then we can run it in console mode with some debugging applied. If it does not start, check the hardware requirements. `$ asterisk -vvvvc`.

6.1.2.1 Configuration files

All of Asterisk is configured in text files, the Asterisk config files that are placed in the `/etc/asterisk` directory on a standard install. In the standard distribution, there are sample files with a lot of comments, explaining various configuration options. Configuration files we can portion to:

- Master configuration file
- Chanel configuration files
- Dialplan configuration files

<i>Asterisk configuration files /etc/asterisk</i>	
Master configuration file	
Asterisk.conf	
Chanel configuration files	
agents.conf	Configure agent channels
h323.conf	Configure H323 channels
iax.conf	Configure IAX channels
Modem.conf	Configure ISDN modems
mgcp.conf	Configure MGCP channels
phone.conf	Configure phone channels (Linux Telephony devices)
sip.conf	Configure SIP channels
skinny.conf	Configure Skinny channels (Cisco SCCP)
vpb.conf	Configure vpb channels (Voicetronix cards)
zapata.conf	Configure Zap channels (Digium cards)
Dialplan configuration files	
Extensions.conf	The Dialplan
parking.conf	Call Parking configuration. Note: This file has been renamed to features.conf as of Asterisk
extconfig.conf	Used by res_data to arrange external configuration (e.g. thru ODBC)
enum.conf	EnumLookup configuration
indications.conf	Playtones tone definitions
meetme.conf	MeetMe conference configuration
musiconhold.conf	MusicOnHold configuration
queues.conf	Queue configuration
voicemail.conf	VoiceMail
Uncategorized configuration files	
modules.conf	Configuration of Asterisk module loading
rtp.conf	Configuration of RTP ports for media

Tab. 11 - Important configuration files

Most important files are `extensions.conf`, `modem.conf`, and channel configuration files like `sip.conf` if we use soft sip phones.

Extensions.conf

The Dialplan is configured in the `extensions.conf` file. It is Asterisk's most important configuration file. It controls how all incoming and outgoing calls are routed and handled. The `extensions.conf` file is portion to parts:

- **General** configure a few general settings in the section headed **[general]**.
- **Globals** section, you may define global variables (or constants) and their initial values.
- **Contexts and Extensions** After the `[general]` and `[globals]` categories, the remainder of the `extensions.conf` file is taken up by the definition of the **Dialplan**. The dialplan consists of a collection of **contexts**. Each context consists of a collection of **extensions**.

```
[general]

static=yes
writeprotect=no

[globals]          ;definition of global variables

PHONE1=SIP/1234    ;Soft SIP phone with nr.1234
PHONE2=SIP/2234
TRUNK=Modem/g1     ;Out
TRUNKMSD=0         ; MSD digits to strip (usually 1 or 0)

;Contexts and Extensions
exten => 2051,1,dial(SIP/xlite,90,mTg) ;extension for SIP-Xlite client
exten => xlite,1,goto(2051,1)

exten => _1X.,1,Dial(${TRUNK}/@<myMSN1>:B${EXTEN:},90,mTg) ;trunk connection to ISDN
network via ISDN modem to nr. 1x....x
```

Tab. 12 -Example of configuration extensions.conf

Modem.conf

Configuration of modem and ISDN connections. For ISDN cards in Linux there are two drivers, either `ISDN4Linux` or `CAPI4Linux`. Which one to use depends on your ISDN card. Digium ISDN PRI cards use the `ZAPATA` drivers and are configure in the `zapata.conf` configuration file.

```

; Configuration for isdn4linux
[interfaces]
;
; By default, incoming calls should come in on the "remote" context
;
context=remote
;
; Modem Drivers to load
;
driver=i4l ;isdn4linux - an alternative to i4l is to use chan_capi
;
;Default language
;
language=en
;
; We can optionally override the auto detection. This is necessary
; particularly if asterisk does not know about our kind of modem.
;
type=autodetect
;
; We can strip a given number of digits on outgoing dialing, so, for example you can
; have it dial "8871042" when given "98871042".
;
;stripmsd=1
;
; Type of dialing
;
dialtype=tone
;dialtype=pulse
;
; Mode selection. "Immediate" means that as soon as you dial, you're
;connected and the line is considered up. "Ring" means we wait until the
;ring cadence occurs at least once. "Answer" means we wait until the
;other end picks up.
;
;mode=answer
mode=ring
;mode=immediate
;
; List all devices we can use.
; ISDN example (using i4l)
;msn=39907835 ;msn number
;device => /dev/ttyI0
;device => /dev/ttyI1

```

Tab. 13 -Example of file modem.conf

Sip.conf

In Sip.conf file we can specify each SIP client. Each SIP client and server is identified by a block of text that looks like:

```

[xxx]
type=yyy
parameter1=value
parameter2=value

```

Where **xxx** is the username associated with the SIP client, or is an arbitrary name used by other configuration files to refer to this SIP device. Typically if a SIP phone has an extension number of 123, then its corresponding entry in this file will begin with [123]. Note that you still have to enable an extension 123 in your dial plan to reach this phone. The other way that incoming SIP requests are matched to [xxx] sections in this file, is to examine the IP address that the request is coming from, and look for a peer [xxx] section that has a matching Host=[value]. If Host = [dynamic], then no match is possible until the SIP client has registered.

```
[grandstream1]
type=friend ; either "friend" (peer+user), "peer" or "user"
context=from-sip
username=grandstream1 ; usually matches the <A&NBSP;CLASS='WIKI'&NBSP;&NBSP;H
REF='SECTION'>section title
fromuser=grandstream1 ; overrides the callerid, e.g. required by FWD
callerid=John Doe <1234>
;host=192.168.0.23 ; we have a static but private IP address
host= dynamic
nat=yes ; there is not NAT between phone and Asterisk
canreinvite=yes ; allow RTP voice traffic to bypass Asterisk
dtmfmode=info ; either RFC2833 or INFO for the BudgeTone
mailbox=1234@default ; mailbox 1234 in voicemail context "default"
disallow=all ; need to disallow=all before we can use allow=
allow=ulaw ; Note: In user sections the order of codecs
; listed with allow= does NOT matter!
;allow=alaw
;allow=g723.1 ; Asterisk only supports g723.1 pass-thru!
;allow=g729 ; Pass-thru only unless g729 license obtained

[xlite1]
;Turn off silence suppression in X-Lite ("Transmit Silence"=YES)!
;Note that Xlite sends NAT keep-alive packets, so qualify=yes is not needed
type=friend
username=xlite1
callerid="Jane Smith" <5678>
host=dynamic
nat=yes ; X-Lite is behind a NAT router
canreinvite=no ; Typically set to NO if behind NAT
disallow=all
allow=gsm ; GSM consumes far less bandwidth than ulaw
allow=ulaw
allow=alaw
```

Tab. 14 -example of sip.conf

6.2 Part of clients

In the World have made many of clients. I focused on the main on the soft clients which can be usable with the asterisk open software and have support more than one of communication protocol. Summary of software clients:

- iFon
- SJphone
- X-Lite
- KPhone
- LIPZ4

6.2.1 iFon soft-phone

iFon™ supports both audio and video with Asterisk as well as Free World Dialup. It also interoperates with all major SIP phones, including Windows Messenger (4.7 and 5.0).



Fig. 11 - iPhone client

6.2.2 xLite soft-phone

A SIP soft-phone with many PBX-like features. Open standards-based design allows for maximum network interoperation and integration. VoIP networks are expanding at a blistering pace, xLite is ready.



Fig. 12 - X-Lite softphone

The X-Lite soft-phones provide standard PBX functionality. You can Touch-tones [DTMF], Multiple Proxies, Line Hold, Inbound Call 'Ignore', Inbound Call 'Go to Voicemail', Call Forwarding URI/URL, Voicemail URL, Dial/ Redial/Hangup, Dynamic CODEC Selection, Caller ID [SIP ID], Call Timer, Silence Threshold, Backspace/Clear/Delete, Mute, Microphone & Speakers Levels, Microphone & Speakers Meters, Push-to-Talk [PocketPC], Last Caller-ID, Recent Calls Dialed, Recent Calls Received, Sound Device Selection, Direct IP to IP Calling, New Familiar-looking Menu, Phonebook [Import/Export CSV], Speed Dial.

6.2.3 Sjphone soft-phone

SJphone™ is a soft-phone which allows you to speak over Internet using any desktops, notebooks, PDAs, stand-alone IP phones, and even any traditional landline or mobile phones. It supports both SIP and H.323 industry open standards and is fully interoperable with most major Internet Telephony Service Providers (ITSP) and software and hardware manufacturers.

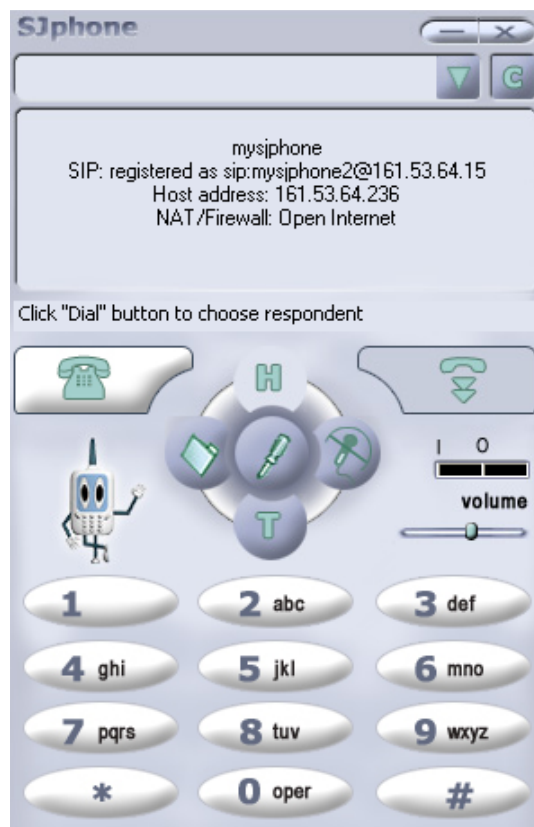


Fig. 13 - SJphone

6.2.4 Kphone softphone

KPhone is a SIP (Session Initiation Protocol) user agent for Linux, with which you can initiate VoIP (Voice over IP) connections over the Internet. It supports Presence and Instant Messaging, and to some extent also video calls between two hosts.

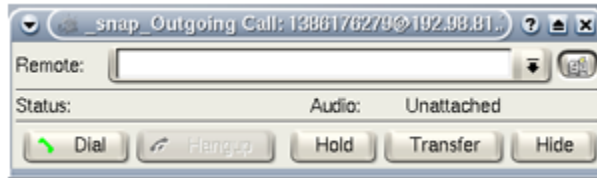


Fig. 14 - Dial screen of KPhone

6.2.5 LIPZ4 Zultys Softphone

The LIPZ4 is a soft phone that runs on the Linux operating system. By using SIP for the call control, the LIPZ4 is compatible with other soft phones, IP phones, and IP phone systems or networks. When used with a speaker or headset and a microphone on your PC, the LIPZ4 is a cost efficient and convenient way to provide telephone communication. Use it within the enterprise to turn Linux desktops into telephones. Use it over the Internet to save money on long distance calls. The LIPZ4 can communicate point to point with another SIP endpoint or with a SIP application server or gateway.

The LIPZ4 provides standard PBX functionality. You can transfer a call, put a call hold, forward all calls, or recall a number from the call log. When used within the enterprise with a SIP based system, the LIPZ4 provides you with a fully functional phone. When used to communicate point to point with other soft phones (mesh configuration) you can obtain basic PBX functionality with a community of users.



Fig. 15 -LIPZ4 Zultys Softphone

<i>Phone</i>	<i>Protocol</i>	<i>Codec</i>	<i>OS</i>	<i>PBX functions</i>
iFone	SIP/H.323	G.729A,G.723.1, GSM/AMR, G.722	Lin, Mac, Win	No
SJphone	SIP/H.323	G.711au, GSM	Lin, Mac, Win	No
X-Lite	SIP	G.711au, GSM, iLBC, SPX,	Lin, (only under Wine), Win	Yes
KPhone	SIP	Not published rec. codec	Lin.	No
LIPZ4	SIP	Not published rec. Codec	Lin.	Yes

Tab. 15 -Summary of software phones

6.3 Part of ISDN connections

To Connect of PBX to telecommunication network we have to have an ISDN access (BRI or PRI), and installed one or more ISDN cards in the machine. It is depend on traffic what we going to offer. For connection to telecommunication network we can use both active and passive ISDN cards with or without CAPI (Common ISDN Application interface). ISDN cards operated with the i4l (isdn4linux) driver.

<i>ISDN card</i>
HFC-S PCI based cards (tested)
Fritzcard PCI (tested)
Sedlbaur FAX
Winbond
HFC-4s / HFC-8s based cards (tested)
HFC-E1 based cards (tested)

Tab. 16 -Summary of ISDN cards supported by the kernel's ISDN driver

Note:

Note that there is an alternative method to use ISDN hardware within Asterisk that employs the CAPI channel driver (chan_capi); however this channel driver is not part of the standard Asterisk code distribution. Most analog modems provide half-duplex only, and Asterisk doesn't provide a simple path to integrate those into the system.

6.4 Part of local interconnected ISDN phones

At least two cards are required to interconnect internal telephones with external telephone lines. In order to connect telephones directly to an ISDN card, it must support **NT-Mode**. Then it is possible to use it as internal ISDN port. There currently there are these types of chip sets, that support NT-Mode hardware layer:

- HFC-S PCI based cards
- HFC-4s / HFC-8s chip sets
- HFC-E1

<i>NT-Mode ISDN cards</i>
Creatix ISDN-S0/PCI
Trust PCI-Modem
Acer ISDN 128 Surf PCI
Billion/Asuscom (Asuscom/Askey)
HFC cards from "Conrad Elektronik" (not available anymore)
D-Link DMI-128I+
Neolec FREEWAY ISDN ^{PCI}
ISDN 128Kbs TA Card (TAS106H)
Junghanns Asterisk boards (HFC-4S / 8S / E1)
PBX4Linux boards (HFC-4S / 8S / E1)
Scitel cards

Tab. 17 -Cards do have the required chip set capable of NT-Mode

6.5 Example of configuration Asterisk PBX in small network with SIP klientes and ISDN connection to ISDN network

In build of PBX i used a network equipment available in Lab of Signal processing on faculty of electrical engineering (FER) in Zagreb. The solution come out from figure 3 in the chapter 4. The core of PBX is a Linux machine (for detail see sub-chapter 6.1.1) with name tonka.zesoi.fer.hr connected to LAN via 100 Mbit Ethernet adapter. Connection into ISDN network is possible through ISDN card.

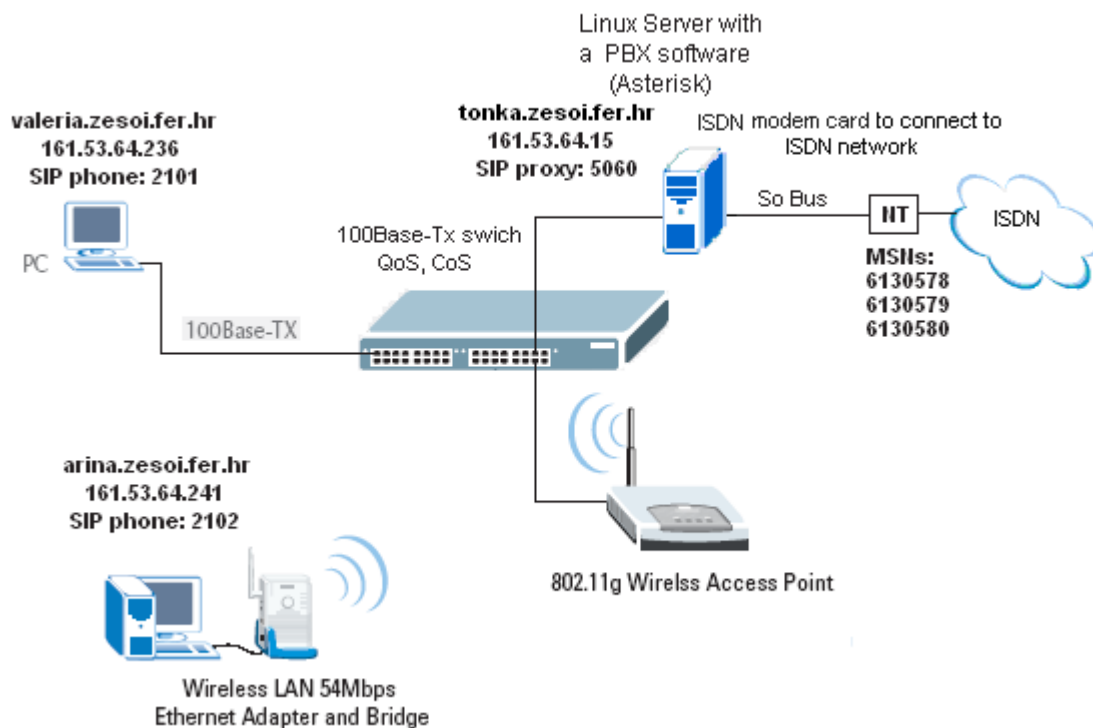


Fig. 16 - VoIP network with SIP soft-phone clients in FER network

6.5.1 Configuration of ISDN modem for outgoing calls to ISDN network

In connection to ISDN network I had in hand a cheap PCI passive ISDN card with a Winbond 6692 chipset. For parameters of this card see below.

<i>ISDN Winbond 6692 chipset</i>	
ISDN interface	S/T (ITU-T I.430)
ISDN signaling	DSS1 (EuroISDN)
B channel protocol	Multi-link PPP/128 kbit/s PPP async to sync HDLC transparent
CAPI 2.0 compatible	Yes
Computer interface	PCI 2.0

Tab. 18 -Specification of ISDN modem

6.5.1.1 Installation of ISDN modem

For ISDN cards are two drivers in Linux, either ISDN4Linux or CAPI4Linux. I selected the first driver ISDN4linux because CAPI is not included in the Asterisk distribution. ISDN4linux drivers should be supported like module or direct in Linux kernel (from ver. 2.2 and more). First we have to make sure if Linux has the isdn4linux module installed, i.e with modprobe. For ISDN card with Winbond 6692 chipset:

```
$ modprobe hisax type=13
```

If module was not install we can recompile kernel with ISDN HiSax driver for passive ISDN cards and Windbond 6692 based card. See bellow.

```
Linux Kernel v2.4.24 Configuration

Passive ISDN cards

Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted
letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help. Legend: [*] built-in
[ ] excluded <M> module < > module capable

<*> HiSax SiemensChipSet driver support
--- D-channel protocol features
[*] HiSax Support for EURO/DSS1
[ ] Support for german chargeinfo
[ ] Disable sending complete
[ ] Disable sending low layer compatibility
[ ] Disable keypad protocol optior
[ ] HiSax Support for german ITR6
[ ] HiSax Support for US NI1
(8) Maximum number of cards supported by HiSax
--- HiSax supported cards
[ ] Teles 16.0/8.0
[ ] Teles 16.3 or PNP or PCMCIA
[ ] AVM A1 (Fritz)
+((+))

<Select> < Exit > < Help >
```

Fig. 17 -Subsystem of passive ISDN card in Linux kernel configuration menu

```
Linux Kernel v2.4.24 Configuration

Passive ISDN cards

Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted
letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help. Legend: [*] built-in
[ ] excluded <M> module < > module capable

^ (-)
[ ] Elsa cards
[ ] Eicon.Diehl Diva cards
[ ] Sedlbauer cards
[ ] NETjet cards
[ ] NETspider U cards
[ ] Niccy PnP/PCI cards
[ ] Telekom A4T cards
[ ] Scitel Quadro cards
[ ] Gazel cards
[ ] HFC PCI-Bus cards
[*] Winbond W6692 based cards
[ ] HFC-S+, HFC-SP, HFC-PCMCIA cards
[ ] Formula-n enter:now PCI card
[ ] HiSax debugging
+((+))

<Select> < Exit > < Help >
```

Fig. 18 -Subsystem of passive ISDN card in Linux kernel configuration menu

6.5.1.2 Testing the card and the physical connection

For testing of physical connection we can use Minicom or another program with similar functionality. After start of Minicom isdn0 (nowadays you have to use Minicom -s isdn0) we have to setting port (ctrl-AO), /dev/ttyIO as the device. For modem we can set init string **AT&E6130578**, which define MSN, Speed to **38400** or whatever; it doesn't really matter, as the speed will be 64kb/s anyway. Last save these settings and chose exit (not exit Minicom).

Minicom will now initialize the "modem", and then connected it to the ISDN modem emulator. We can try to write **AT** commands. If we put **AT** we should get a response **OK**. The **ATI1** command return device Information ("ISDN for Linux...") or **AT&V** show current register settings.

With **ATD<number>** we can test to call specified number. If we don't see **RING**, then maybe we have to enter the number with areacode.

If in the screen was we entered **ATD** you see **BUSY** appear, then a number of things may be wrong; unfortunately most failures are simply indicated with **BUSY** (only when there is no physical connection working do you see a different message, namely **NO CARRIER**). If **BUSY** appears after 1-3 seconds, then the message may be right, and the remote number is in fact busy. However, if it takes longer, then probably something else is the cause of the problem. Look in the system log messages for hints.

6.5.1.3 Configuration ISDN connection in PBX (modem.conf)

For ISDN (BRI) we have the device: **/dev/ttyIO** so we will have to add it as well - modem.conf. Most likely you will define a group like "g1" to include all ISDN channels. In extensions.conf we probably want to define g1 as trunk: TRUNK=Modem/g1.

```
; isdn4linux
; Configuration file
;
[interfaces]
;
; By default, incoming calls should come in on the "remote" context
;
context=remote
;
; Modem Drivers to load
;
;driver=aopen; modem by AOpen
driver=i4l ;isdn4linux - an alternative to i4l is to use chan_capi
;
; Default language
language=en
;
; We can optionally override the auto detection. This is necessary
; particularly if asterisk does not know about our kind of modem.
;
type=autodetect
;
; Type of dialing
;
dialtype=tone
;dialtype=pulse
;
; Mode selection. "Immediate" means that as soon as you dial, you're connected
```

```

; and the line is considered up. "Ring" means we wait until the ring cadence
; occurs at least once. "Answer" means we wait until the other end picks up.
;
;mode=answer
mode=ring
;mode=immediate
;
; List all devices we can use.
;dtmfmode=asterisk ; Detect using Asterisk
dtmfmode=asterisk/both ; Detect using Asterisk, generate w/ both
; two other devices, which are in group '1' and are used when an
; outgoing dial used: exten => s,1,Dial,Modem/g1:1234|60|r
; (we do not need more outgoing devices, since ISDN2 has only 2 channels.)
; Lines can be in more than one group (1-31); comma separated list.
;
group=1 ; group=1,2,3,9-12
msn=6130578,6130579,6130580
outgoingmsn=6130578
device => /dev/ttyI0

```

Tab. 19 -Modem.conf with defined ISDN device like ttyI0

6.5.2 Config of SIP clients accounts

Now we have to define accounts for Sip phone clients in sip config file in our PBX. We know that configuration file for sip clients name `sip.conf`. In this file we fill the bind address and SIP proxy port too.

```

[general]
context=default ; Default context for incoming calls
; defaults to "asterisk"
; Set this to your host name or domain name
port=5060 ; UDP Port to bind to (SIP standard port is 5060)
bindaddr=161.53.64.15 ; IP address to bind to (0.0.0.0 binds to all)

```

Tab. 20 - General field in sip.conf

Every SIP client have self account with specification of name, callerid, host and codec. The host can be with dynamic or with static IP address, We allow 3 codec: iLBC, GSM and G.711a.

```

[xlite1] ;static client
type=friend
username=xlite1
secret=xlite1
callerid="2101 xlite1" <2101>
host=161.53.64.236
;nat=yes ; xLite is behind a NAT router
canreinvite=no ; Typically set to NO if behind NAT
defaultip=161.53.64.236
disallow=all
allow=gsm ; GSM consumes far less bandwidth than ulaw
;allow=ulaw
allow=alaw
allow=iLBC

[xlite2] ; xlite2-10 dynamic clients
type=friend
username=xlite2

```

```
secret=xlite2
callerid="2102 xlite2" <2102>
host=dynamic
nat=yes
canreinvite=no
disallow=all
allow=gsm          ; GSM consumes far less bandwidth than ulaw
;allow=ulaw
allow=alaw
allow=iLBC
```

Tab. 21 -Configuration of SIP clients in sip.conf in PBX

6.5.3 Dial plan configuration

For configuration of dial plan we use file `extension.conf`. Via this file we have to allocate to each SIP clients and any other clients number. For SIP xlite clients we use local number from 2101 to 2101. The dial plan is in next table.

```
[globals]
TRUNK=>Modem/g1
TRUNKMSD=0                ; MSD digits to strip (usually 1 or 0)

[default]
exten => 2101,1,dial(SIP/xlite1,90,mTg) ; xlite SIP phone_1, with nr. 2101
exten => xlite1,1,goto(2101,1)        ; 90s ringing with music in background

exten => 2102,1,dial(SIP/xlite2,90,mTg) ;xlite SIP phone_2 with nr. 2102
exten => xlite2,1,goto(2102,1)

.....
```

Tab. 22 -Dial plan configuration for Sip clients in extension conf

For connection to ISDN network we have defined global variable TRUNK like TRUNK=>Modem/g1. With special chars X, N or Z we can create mask of international number who can users call. X substitute number from interval: [0 – 9], N - substitute number from interval: [1 – 9], Z - substitute number from interval: [2 – 9].

```
[trunks]

exten => _961XXXXX,1,Dial(${TRUNK}/${EXTEN:1}+msn6130578) ;call to anlg
                                                ;loc. PBX 612xxxx

exten =>_961XXXXX,2,Congestion

exten => _900385X.,1,Dial(${TRUNK}/${EXTEN:1}+msn6130578) ;Croatia calls
exten =>_9612XXXX,2,Congestion

exten => _9112,1,Dial(${TRUNK}/${EXTEN:1}+msn6130578) ;emergency calls
exten =>_9112,2,Congestion
```

Tab. 23 -Masks for outgoing calls

6.5.4 Configuration of SIP client

From compare of soft VoIP clients we can see that the best solution is take the Xlite. First is a Freeware. Second have a lot PBX functions and supported codec for communication via lowband (iLBC, GSM) and broadband (G.711a,u-law) connection. See chapter 6.2.2. For successful set of connection with our PBX we have to set IP of SIP proxy and port of SIP proxy, because we going to use SIP protocol for calls. PBX with **proxy 161.53.64.15** have listened on **port 5060**. Client machine listen in this example on IP **161.53.64.236**.



Fig. 19 -xlite client

Configuration of xlite is available under menu button. See on the figure 19. In rolled menu we have to fill a user name and password of SIP client. In PBX we have to allocated accounts **xlite1** to **xlite10** with number of dialplan from **2101** to **2110** and passwords **xlite1** to **xlite10**.



Fig. 20 -example of network setup



Fig. 21 -user name and password setup

Appendix A

D-Link *AirPlus* DWL-900 AP+

The D-Link *AirPlus* DWL-900 AP+ is an enhanced 802.11b+ Wireless Network Access Point featuring advanced silicon chip design from Texas Instruments, utilizing their patented Digital Signal Processing-technology.

The DWL-900AP+ can be configured to perform in any one of five modes — as a wireless access point, as a point-to-point bridge with another access point, as a point-to-multi-point wireless bridge, as a wireless client, or as a wireless repeater. It also includes an embedded DHCP server that once enabled will automatically assign IP addresses to wireless clients. This unique feature along

with the repeater capability makes the DWL-900AP+ an ideal solution for quickly creating and extending a wireless local area network in offices or other workplaces or even at trade shows or other special events.

The DWL-900AP+ features 256-bit WEP encryption for a higher level of security for your data and wireless-communications.

The DWL-900AP+ is fully compatible with the IEEE 802.11b standard, making it interoperable with all existing 802.11b compliant devices. But unlike standard 802.11b access points, its data transfer rate can be up to 20% faster. When your wireless network is comprised of other D-Link *AirPlus* IEEE 802.11b+ products such as the DWL-520+ Wireless PCI Adapter, DWL-650+ Wireless Cardbus Adapter, and the DI-614+ Wireless Router — transfer rates can be as high as 22Mbps!



D-Link 11b D-link 122 USB network adapter

The DWL-122 is Wi-Fi compliant, meaning that it can connect and interoperate with other 802.11b Wi-Fi-compliant wireless routers, access points, and adapters. The DWL-122 connects to any available USB port on a laptop or desktop Windows and Macintosh-based computer and also includes a standard USB extension cable. With its USB extension cable the DWL-122 can be positioned virtually anywhere at a workstation to achieve the best available wireless signal reception - whether that workstation is in your office, on your deck or patio, by the pool, at the local coffee shop,



or in an airport terminal while waiting to board your next flight.

The adapter works with Apple Mac OS X (10.2.x) Jaguar, Microsoft Windows 98 Second Edition, ME, 2000, and XP to ensure that you'll be up and running on a wireless network in just a matter of seconds. The DWL-122 also features 64 or 128-bit WEP encryption providing a level of security for the data that you wirelessly transmit.

HP ProCurve Switch 2626 Switch

Designed for mid-size enterprises that require a smaller, more cost effective switch without sacrificing performance, the Switch 2626 broadens the scope of HP's Ethernet switch offerings, providing a lower cost option as a part of the HP ProCurve Switch 2600 series. The HP ProCurve Switch 2626 supports advanced network applications, including advanced network security and converged data and voice networks. With minimal configuration needs, the switch offers easy, plug-n-play enterprise deployment. The Switch 2626 provides the stackable 10/100 port density needed in smaller wiring closets today for aggregation at the edge of the network. With wire-speed performance and scalable port density, enterprises can utilize this lower cost option while still satisfying the performance demands of corporate environments.



Reference:

- [1] Voice over IP: Seminar [cit. 24.8.2004]. Available <URL: <http://artax.karlin.mff.cuni.cz/~brain/voip.html>>
- [2] Svet Siti: Internet jurnal [cit. 24.8.2004]. Available <URL: <http://www.svetsiti.cz>>
- [3] The TCP/IP Guide: Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) [cit. 25.8.2004]. Available <URL: <http://www.tcpipguide.com>>
- [4] Open H.323 project: H.323 Standards [cit. 25.8.2004]. Available <URL: <http://www.openh323.org/standards.html>>
- [5] International Telecommunication Union: Normalization Institute. Available <URL: <http://www.itut.int>>
- [6] IETF-The Internet Engineering Task Force: A Transport Protocol for Real-Time. Available <URL: <http://www.ietf.org/rfc/rfc1889.txt>>
- [7] Asterisk The open source Linux PBX: Open software for Private Branch Exchange [cit. 24.8.2004]. Available <URL: <http://www.asterisk.org>>
- [8] TLPD-The Linux Documentation Project: VoIP Howto [cit. 25.8.2004]. Available <URL: <http://www.tldp.org/HOWTO/VoIP-HOWTO-4.html#ss4.7>>
- [9] Juha L., Sampo S., Rui P.; RUDE & CRUDE: Real-time UDP Data Emitter and Collector, Available <URL: <http://www.atm.tut.fi/rude/>>

- [10] QoS-plot a Cesnet project: Tool for computing QoS characteristics, Available <URL: <http://www.ces.net/project/qosip/>>
- [11] GNU-Plot: Tool for plotting, Available <URL: <http://www.gnuplot.info>>
- [12] Ubik S.; Přesné a jednoduché měření kvalitativních parametrů sítě, Available <URL: <http://www.ces.net>>
- [13] Soumar M.; Signalizační protokol pro přenos hlasu přes datové sítě – SIP, Available <<http://www.elektrorevue.cz/clanky/03003/iso-8859-1>>
- [14] Komosný S.; Kvalita služeb IP, Available <http://www.elektrorevue.cz/clanky/04006/iso-8859-1>>
- [15] ITU-T Standard activities; ITU-T- Y.1541, Available <<http://www.its.bldrdoc.gov/tpr/2001/itu/itu.html>>
- [16] Voice over IP: Seminar [cit. 24.8.2004]. Available <URL: <http://artax.karlin.mff.cuni.cz/~brain/voip.html>>
- [17] Projekt QoS v IP, Available <URL: <http://www.cesnet.cz/english/project/qosip>>
- [18] Netperf A.: Network Performance Benchmark“, Available <URL: <http://www.netperf.org>>
- [19] Asterisk: The open source PBX, Available <URL: <http://www.asterisk.org>>