

## Fine-tuning Voice over Packet services

Written by: Yuval Boger, VP Business Development, RADCOM Ltd.

### Introduction

The transfer of voice traffic over packet networks, and especially voice over IP, is rapidly gaining acceptance. Many industry analysts estimate that the overall VoIP market will become a multi-billion dollar business within three years.

While many corporations have long been using voice over Frame Relay to save money by utilizing excess Frame Relay capacity, the dominance of IP has shifted most attention from VoFR to VoIP. Voice over packet transfer can significantly reduce the per-minute cost, resulting in reduced long-distance bills. In fact, many dial-around-calling schemes available today already rely on VoIP backbones to transfer voice, passing some of the cost savings to the customer. These high-speed backbones take advantage of the convergence of Internet and voice traffic to form a single managed network.

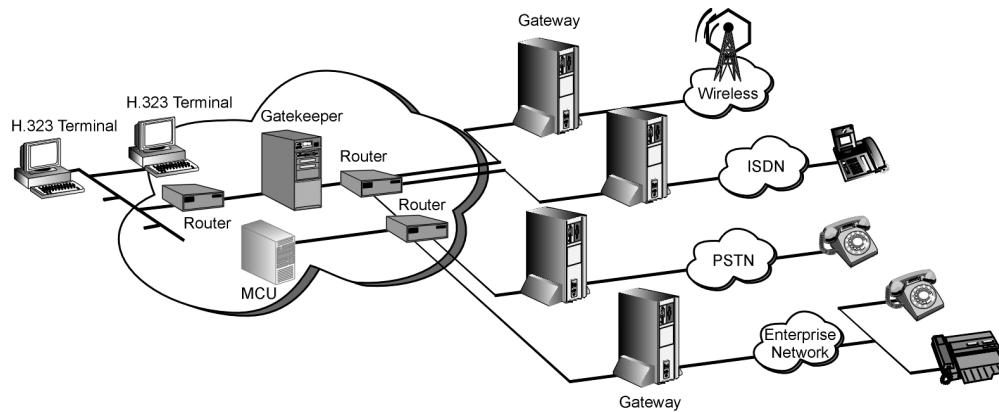
This network convergence also opens the door to novel applications. Interactive shopping (web pages incorporating a “click to talk” button) are just one example, while streaming audio, electronic white-boarding and CD-quality conference calls in stereo are other exciting applications.

But along with the initial excitement, customers are worried over possible degradation in voice quality when voice is carried over these packet networks. Whether these concerns are based on experience with the early Internet telephony applications, or whether they are based on understanding the nature of packet networks, voice quality is a critical parameter in acceptance of VoIP services. As such, it is crucial to understand the factors affecting voice over packet transmission, as well as obtain the tools to measure and optimize them.

This white paper covers the basic elements of voice over packet networks, the factors affecting voice quality and discusses techniques of optimizing voice quality as well as solving common problems in VoIP networks.

### VoIP network elements

VoIP services need to be able to connect to traditional circuit-switched voice networks. The ITU-T has addressed this goal by defining H.323, a set of standards for packet-based multimedia networks. The basic elements of the H.323 network are shown in the network diagram below where H.323 terminals such as PC-based phones (left side of drawing) connect to existing ISDN, PSTN and wireless devices (right side):



**Figure 1 - Typical H.323 network**

The H.323 components in this diagram are: H.323 terminals that are endpoints on a LAN, gateways that interface between the LAN and switched circuit network, a gatekeeper that performs admission control functions and other chores, and the MCU (Multipoint Control Unit) that offers conferences between three or more endpoints. These entities will now be discussed in more detail.

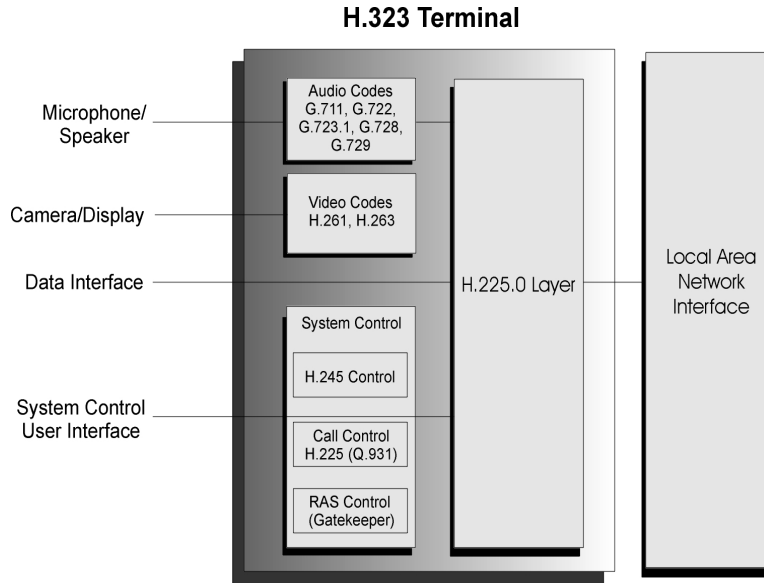
### ***H.323 Terminals***

H.323 terminals are LAN-based end points for voice transmission. Some common examples of H.323 terminals are a PC running Microsoft NetMeeting software and an Ethernet-enabled phone. All H.323 terminals support real-time, 2-way communications with other H.323 entities.

H.323 terminals implement voice transmission functions and specifically include at least one voice CODEC (Compressor / Decompressor) that sends and receives packetized voice. Common CODECs are ITU-T G.711 (PCM), G.723 (MP-MLQ), G.729A (CA-ACELP) and GSM. Codecs differ in their CPU requirements, in the resultant voice quality and in their inherent processing delay. CODECs are discussed in more detail below.

Terminals also need to support signalling functions that are used for call setup, tear down and so forth. The applicable standards here are H.225.0 signalling which is a subset of ISDN's Q.931 signalling; H.245 which is used to exchange capabilities such as compression standards between H.323 entities; and RAS (Registration, Admission, Status) that connects a terminal to a gatekeeper. Terminals may also implement video and data communication capabilities, but these are beyond the scope of this white paper.

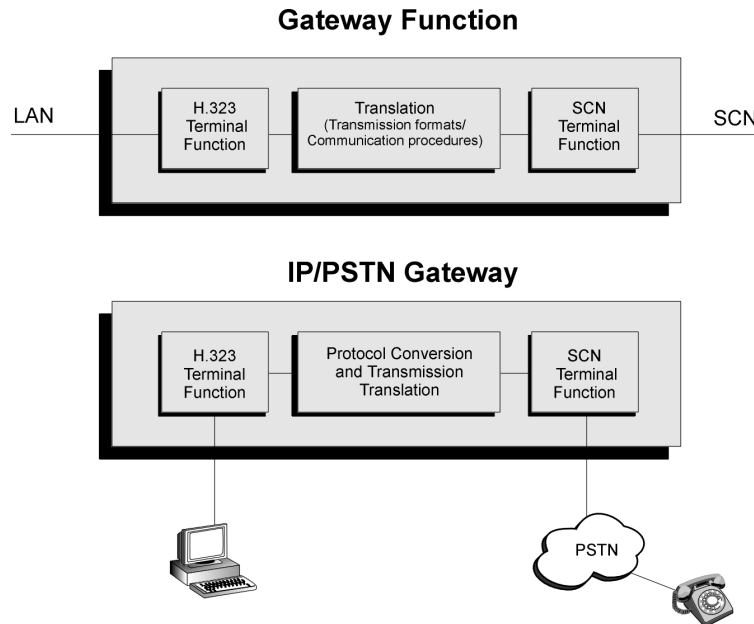
The functional block diagram of an H.323 terminal is summarized below:



**Figure 2 - Functional decomposition of an H.323 terminal**

### Gateways

The gateway serves as the interface between the H.323 and non-H.323 network. On one side, it connects to the traditional voice world, and on another side to packet-based devices. As the interface, the gateway needs to translate signalling messages between the two sides as well as compress and decompress the voice. A prime example of a gateway is the PSTN/IP gateway, connecting an H.323 terminal with the SCN (Switched Circuit Network) as shown in the following diagram:



There are many types of gateways in existence today, ranging from support of a dozen or so analog ports to high-end gateways with simultaneous support for thousands of lines.

### ***Gatekeeper***

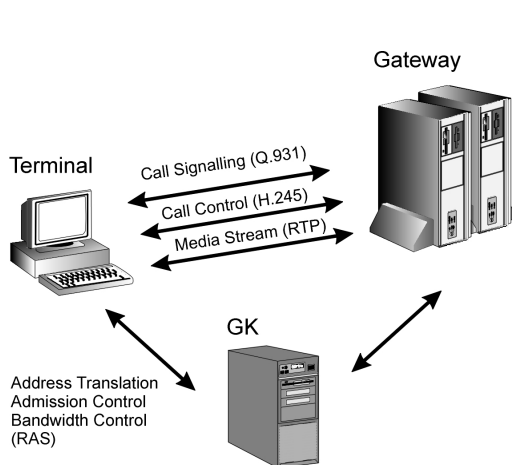
The gatekeeper is not a mandatory entity in an H.323 network. However, if a gatekeeper is present, it must perform a set of functions. Gatekeepers manage H.323 zones, logical collection of devices (for example: all H.323 devices within an IP subnet). Multiple gatekeepers may be present for load-balancing or hot-swap backup capabilities.

The philosophy behind defining the gatekeeper entity is to allow H.323 designers to separate the raw processing power of the gateway from intelligent network-control functions that can be performed in the gatekeeper. A typical gatekeeper is implemented on a PC, whereas gateways are often based on proprietary hardware platforms.

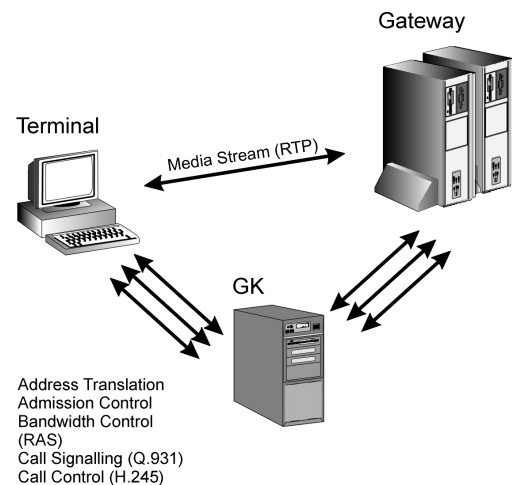
Gatekeepers provide address translation (routing) for devices in their zone. This could be, for instance, the translation between internal and external numbering systems. Another important function for gatekeepers is providing admission control, specifying what devices can call what numbers.

Among the optional control functions for gatekeepers are providing SNMP management information, offering directory and bandwidth management services.

A gatekeeper can participate in a variety of signalling models as dictated by the gatekeeper. Signalling models determine what signalling messages pass through the gatekeeper, and what can be passed directly between entities such as the terminal and the gateway. Two such signalling models are illustrated below. A direct signalling model (left diagram) calls for exchange of signalling messages without involving the gatekeeper, while in a gatekeeper routed call signalling model (right diagram), all signalling passes through the gatekeeper, and only media can pass directly between the stations.



**Figure 3 - direct signalling model**



**Figure 4 - gatekeeper routed signalling**

## ***Multipoint Control Unit (MCU)***

MCU's allow for conferencing functions between three or more terminals. Logically, an MCU contains two parts:

- Multipoint controller (MC) that handles the signalling and control messages necessary to setup and manage conferences.
- Multipoint processor (MP) that accepts streams from endpoints, replicates them and forwards them to the correct participating endpoints.

An MCU can implement both MC and MP functions, in which case it is referred to as a centralized MCU. Alternatively, a decentralized MCU handles only the MC functions, leaving the multipoint processor function to the endpoints.

It is important to note that the definition of all the H.323 network entities is purely logical. No specification has been made on the physical division of the units. MCU's, for instance, can be standalone devices, or be integrated into a terminal, a gateway or a gatekeeper.

## **Audio CODECs**

Voice channels occupy 64 Kbps using PCM (pulse code modulation) coding when carried over T1 links. Over the years, compression techniques were developed allowing a reduction in the required bandwidth while preserving voice quality. Such techniques are implemented as CODECs.

Although many proprietary compression schemes exist, most H.323 devices today use CODECs that were standardized by standards bodies such as the ITU-T for the sake of interoperability across vendors. Applications such as NetMeeting use the H.245 protocol to negotiate which CODEC to use according to user preferences and the installed CODECs. Different compression schemes can be compared using four parameters:

- Compressed voice rate – the CODEC compresses voice from 64 Kbps down to a certain bit rate. Some network designs have a big preference for low-bit-rate CODECs. Most CODECs can accommodate different target compression rates such as 8, 6.4 and even 5.3 Kbps. Note that this bit rate is for audio only. When transmitting packetized voice over the network, protocol overhead (such as RTP/UDP/IP/Ethernet) is added on top of this bit rate, resulting in a higher actual data rate.
- Complexity – the higher the complexity of implementing the CODEC, the more CPU resources are required.
- Voice quality – compressing voice in some CODECs results in very good voice quality, while others cause a significant degradation.
- Digitizing delay – Each algorithm requires that different amounts of speech be buffered prior to the compression. This delay adds to the overall end-to-end delay (see discussion below). A network with excessive end-to-end delay, often causes people to revert to a half-duplex conversation (“How are you today? over...”) instead of the normal full-duplex phone call.

The following table compares popular CODECs according to these parameters:

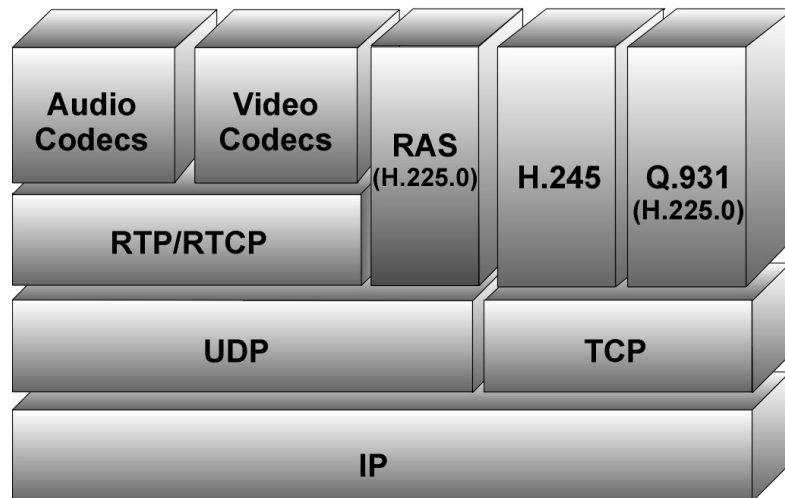
Compression scheme	Compressed rate (Kbps)	Required CPU resources	Resultant voice quality	Added delay
G.711 PCM	64 (no compression)	Non required	Excellent	N/A
G.723 MP-MLQ	6.4/5.3	Moderate	Good (6.4) Fair (5.3)	High
G.726 ADPCM	40/32/24	Low	Good (40) Fair (24)	Very low
G.728 LD-CELP	16	Very high	Good	Low
G.729 CS-ACELP	8	High	Good	Low

There is no “right CODEC”. The choice of what compression scheme to use depends on what parameters are more important for a specific installation. In practice, G.723 and G.729 are more popular than G.726 and G.728.

### H.323 Protocol Stack

The H.323 protocol stack is shown in the following diagram:

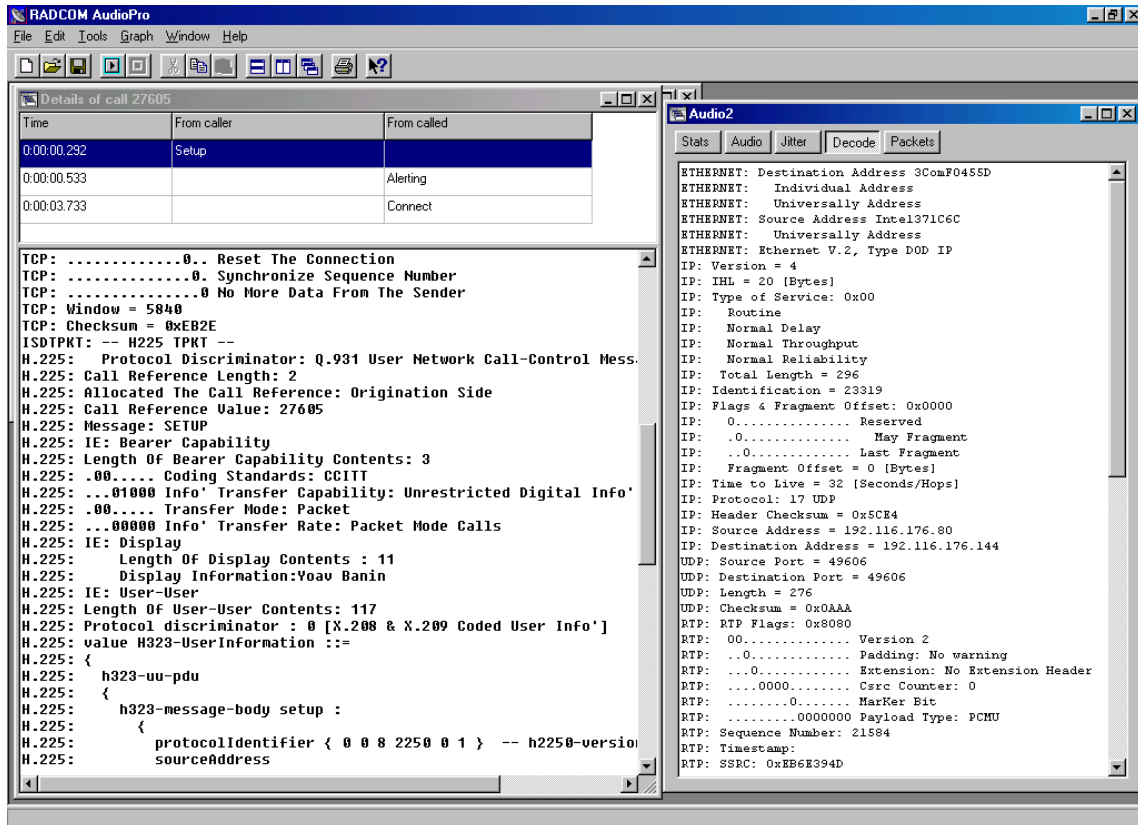
#### Protocol Stack



**Figure 5 - H.323 protocol stack**

Control messages (Q.931 signalling, H.245 capability exchange and the RAS protocol) are carried over the reliable TCP layer. This ensures that important messages get retransmitted if necessary so they can make it to the other side. Media traffic is transported over the unreliable UDP layer and includes two protocols as defined in IETF RFC 1889: RTP (Real-Time Protocol) that carries the actual media and RTCP (Real-Time Control Protocol) that includes periodic status and control messages. Media is carried over

UDP because it would not make sense for it to be retransmitted: should a lost sound fragment be retransmitted, it would most probably arrive too late to be of any use in voice reconstruction. RTP messages are typically carried on even-numbered UDP ports, whereas RTCP messages are carried on the adjacent odd-numbered ports. The following figure illustrates the different encapsulations by showing a side-by-side display of actual RTP and H.225 messages:



## Understanding and measuring factors affecting voice quality

In the traditional circuit-switched network, each voice channel occupied a unique T1 timeslot with fixed 64 Kbps bandwidth. When travelling over the packet network, voice packets must contend with new phenomena that may affect the overall voice quality as perceived by the end-customer. The premier factors that determine voice quality are choice of CODEC that we already discussed, as well as latency, jitter and packet loss.

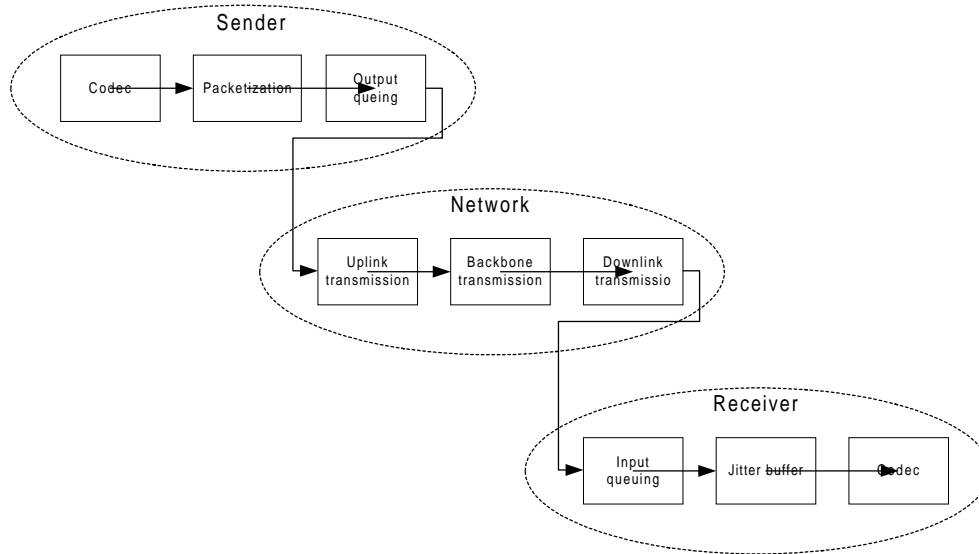
### Understanding latency

In contrast to broadcast-type media transmission (e.g., RealAudio), a two-way phone conversation is quite sensitive to latency, Most callers notice round-trip delays when they exceed 250mSec, so the one-way latency budget would typically be 150 mSec. 150 mSec is also specified in ITU-T G.114 recommendation as the maximum desired one-way latency to achieve high-quality voice. Beyond that round-trip latency, callers start feeling uneasy holding a two-way conversation and usually end up talking over each other. At 500 mSec round-trip delays and beyond, phone calls are impractical, where you can

almost tell a joke and have the other guy laugh after you've left the room. For reference, the typical delay when speaking through a geo-stationary satellite is 150-500mSec.

Data networks were not affected by delay. An additional delay of 200 mSec on an e-mail or web page goes mostly unnoticed. Yet when sharing the same network, voice callers will notice this delay.

When considering the one-way delay of voice traffic, one must take into account the delay added by the different segments and processes in the network, as shown in the following diagram:



**Figure 6 - Delay budget in a network**

Some components in the delay budget need to be broken into fixed and variable delay. For example, for the backbone transmission there is a fixed transmission delay which is dictated by the distance, plus a variable delay which is the result of changing network conditions.

The most important components of this latency are:

- **Backbone (network) latency.** This is the delay incurred when traversing the VoIP backbone. In general, to minimize this delay, try to minimize the router hops that are traversed between end-points. To find out how many router hops are used, it is possible to use the `traceroute` utility. Some service providers are capable of providing an end-to-end delay limit over their managed backbones. Alternatively, it is possible to negotiate or specify a higher priority for voice traffic than for delay-insensitive data.
- **CODEC latency.** Each compression algorithm has certain built-in delay. For example, G.723 adds a fixed 30 mSec delay. When this additional gateway overhead is added in, it is possible to end up paying 32-35 mSec for passing through the gateway. Choosing different CODECs may reduce the latency, but reduce quality or result in more bandwidth being used.

- Jitter buffer depth. To compensate for the fluctuating network conditions, many vendors implement a jitter buffer in their voice gateways. This is a packet buffer that holds incoming packets for a specified amount of time before forwarding them to decompression. This has the effect of smoothing the packet flow, increasing the resiliency of the CODEC to packet loss, delayed packets and other transmission effects. However, the downside of the jitter buffer is that it can add significant delay. The jitter buffer size is configurable, and as shown below, can be optimized for given network conditions. The jitter buffer size is usually set to be an integral multiple of the expected packet inter-arrival time in order to buffer an integral number of packets. It is not uncommon to see jitter buffer settings approaching 80 mSec for each direction.

When designing or optimizing a network, it is often useful to build a table showing the one-way delay budget as in the example below with typical values:

Parameter	Fixed delay	Variable delay
CODEC (G.729)	25 mSec	
Packetization	Included in CODEC	
Queuing delay		Depends on uplink. In the order of a few mSec.
Network delay	50mSec	Depends on network load
Jitter buffer	50mSec	
Total	125mSec	

**Figure 7 - Sample delay budget**

***Measuring latency***

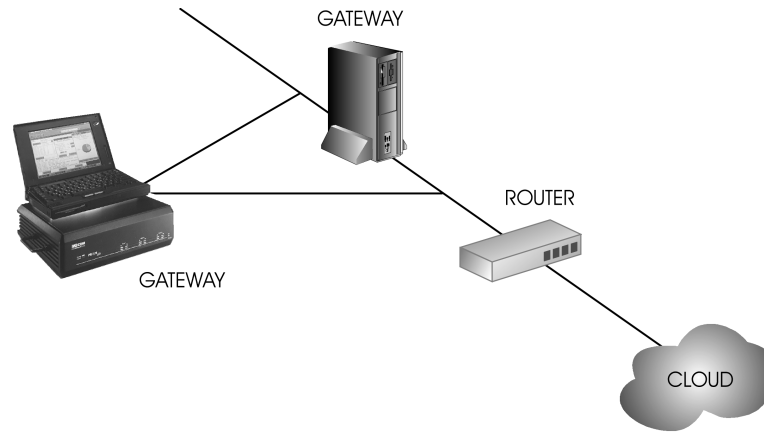
There are three interesting configurations for measuring latency: measuring latency of a device, measuring round-trip delay and measuring one-way delay.

Measuring latency of a device is important to understand how the delay budget gets spent over the network. In particular, it is interesting to measure the latency of data going through a gateway since several user-configurable parameters such as jitter-buffer size affect the latency. Thus, after configuring such parameters, it is important to be able to verify that the gateway actually behaves as expected.

RADCOM products allow measuring the latency by generating controlled data through an ingress port and capturing it off an egress port. Our protocol analyzers can operate two technologies at the same time, with a synchronized timestamp that allows inter-port or inter-technology latency measurement.

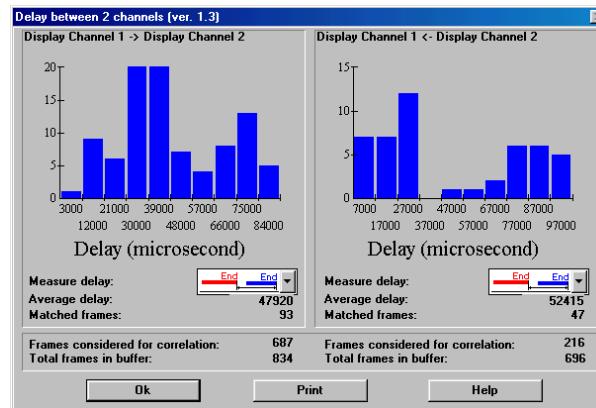
Another unique application that is extremely suitable for these types of measurement is the latency and loss application. When running this application, the analyzer is placed in non-intrusive monitor (listening) mode on the ingress and egress ports. The gateway continues to perform its role in the network, with actual packets flowing

through it. Instead of requiring the user to define test traffic for generation through the device, RADCOM analyzers perform this measurement on the actual data and on any device. The diagram below shows such test configuration



**Figure 8 - Measuring the delay across a gateway**

Once the data is captured on both sides of the device, the analyzer runs a heuristic algorithm that automatically correlates the data captured on both segments. Each frame on one side is matched with data on the other side. As a result, the analyzer displays graphical and numerical information about the latency and loss through the device. As latency may be different in each direction, two latency histograms are displayed side-by-side as shown below:

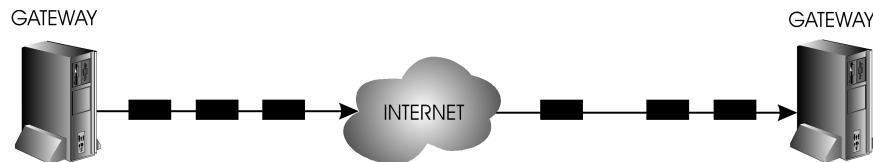


**Figure 9 - Latency and loss measurement results**

The analyzer can measure the latency through a network using a similar method. When the two end-points are geographically distant, it is often less convenient to perform one-way latency measurements because such an operation requires synchronizing the control and timestamp of two separate analyzers. Instead, many users measure the round-trip time and assume it is twice the one-way time for each direction. Round-trip measurements can be done using a protocol analyzer, or as a first approximation using the ping utility generating ICMP echo requests through the network.

## ***Understanding jitter***

While network latency effects how much time a voice packet spends in the network, jitter controls the regularity in which voice packets arrive. Typical voice sources generate voice packets at a constant rate. The matching voice decompression algorithm also expects incoming voice packets to arrive at a constant rate. However, the packet-by-packet delay inflicted by the network may be different for each packet. The result: packets that are sent in equal spacing from the left gateway arrive with irregular spacing at the right gateway, as shown in the following diagram:



**Figure 10 - Packet Jitter**

Since the receiving decompression algorithm requires fixed spacing between the packets, the typical solution is to implement a jitter buffer within the gateway. The jitter buffer deliberately delays incoming packets in order to present them to the decompression algorithm at fixed spacing. The jitter buffer will also fix any out-of-order errors by looking at the sequence number in the RTP frames. The operation of the jitter buffer is analogous to a doctor's office where patients that have appointments at fixed intervals do not arrive exactly on time and are deliberately delayed in the waiting room so they can be presented to the doctor at fixed intervals. This makes the doctor happy because as soon as he is done with a patient, another one comes in, but this is at the expense of keeping patients waiting. Similarly, while the voice decompression engine receives packets directly on time, the individual packets are delayed further in transit, increasing the overall latency.

## ***Measuring jitter***

Jitter is calculated based on the inter-arrival time of successive packets. Frequently, two numbers are given: the average inter-arrival time, and the standard deviation. On a good network, the average inter-arrival time will be the inter-arrival time of the emitted packets, and the standard deviation will be low – pointing at a consistent inter-arrival time.

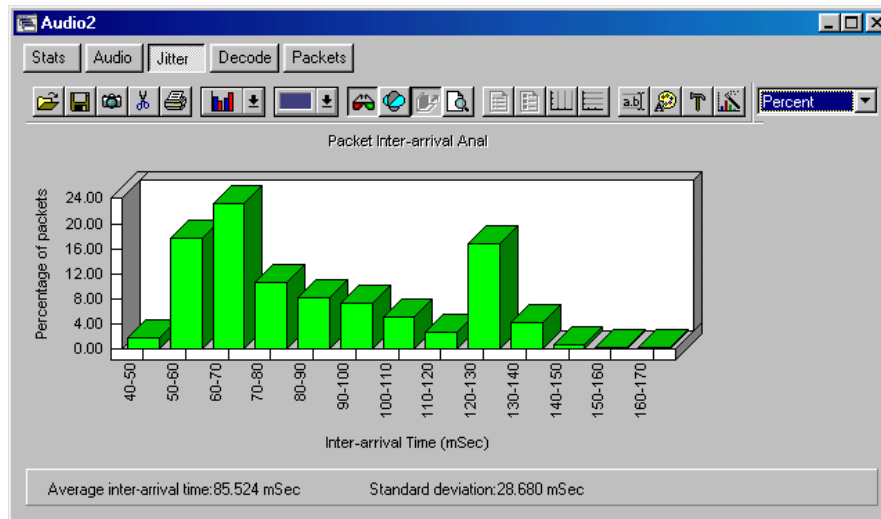
When correct jitter measurements are desired for audio streams, it is important to take into account three phenomena: silence suppression, packet loss and out of sequence errors.

CODECs take advantage of periods of silence in the conversation to reduce the number of packets being sent. Typically, up to 50% bandwidth savings can be realized in this way. The RTP packet immediately after a period of silence is marked with the silence suppression bit. Jitter calculations look at the silence suppression bit and disregard the long gap between the packet right before the silence and the packet right after the silence period.

In the event of packet loss, the inter-arrival time between two successive packets will also appear excessive. For instance, if three packets were sent at a time of 0, 20 and 40 mSec, and the second packet was lost in transit, the inter-arrival time would appear to be 40mSec even if the network induced no jitter. Correct jitter measurements would discover these cases by looking at the packet sequence number and compensate for packet loss in the jitter calculation.

Out of sequence packets may also skew jitter measurements when not taken into account. For instance, consider an example where packet 1 was sent at time 0 and arrived at time 100, packet 2 was sent at time 20 and arrived at time 140 while packet 3 was sent at time 40 and arrived at time 120. Packets arrived to the receiver at times 100, 120 and 140, so no jitter would be detected unless the analyzer also examined the sequence numbers. When doing so, the jitter would be calculated based on a 40 mSec inter-arrival between packets 1 and 2, as well as a -20mSec inter-arrival time between packets 2 and 3.

RADCOM offers solutions for measuring jitter over any physical interface. In particular, the RADCOM AudioPro is capable of tapping into a VoIP link, separating the individual audio streams and providing simultaneous jitter measurements of these streams while taking into account silence suppression, packet loss and out-of-sequence packets. Such an analysis is shown below:



**Figure 11 - Audio jitter analysis**

***Packet loss***

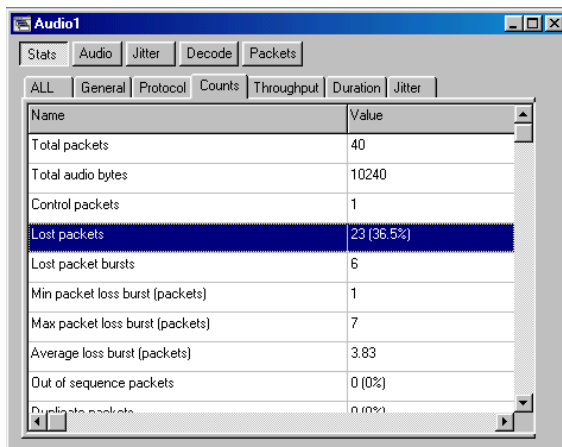
Packet loss is a normal phenomenon on packet networks. Loss can be caused by many different reasons: overloaded links, excessive collisions on a LAN, physical media errors and others. Transport layers such as TCP account for loss and allow packet recovery under reasonable loss conditions.

Audio CODECs also take into account the possibility of packet loss, especially since RTP data is transferred over the unreliable UDP layer. The typical CODEC performs

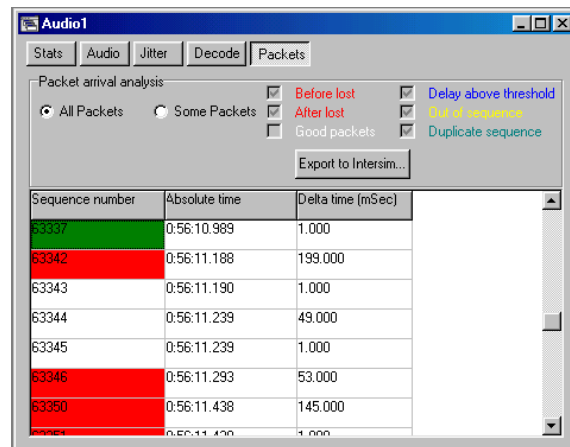
one of several functions that make an occasional packet loss unnoticeable to the user. For example, a CODEC may choose to use the packet received just before the lost packet instead of the lost one, or perform more sophisticated interpolation to eliminate any clicks or interruptions in the audio stream.

However, packet loss starts to be a real problem when the percentage of the lost packets exceeds a certain threshold (roughly 5% of the packets), or when packet losses are grouped together in large packet bursts. In those situations, even the best CODECs will be unable to hide the packet loss from the user, resulting in degraded voice quality. Thus, it is important to know both the percentage of lost packets, as well as whether these losses are grouped into packet bursts.

When the RADCOM AudioPro analyzes audio streams, it provides both top-level statistics as well as drill-down analysis of individual packet loss. The summary statistics for a sample audio stream with heavy losses are shown in Figure 12. Figure 13 shows the packet-by-packet analysis, with packets just before or after a loss clearly marked.



**Figure 12 - Summary Statistics**



**Figure 13 - Drill-down analysis**

Using this analysis, it is easily possible to verify whether the current network conditions allow quality voice communications.

## Important network parameters

Having discussed the parameters that affect voice quality, and especially jitter and loss, perhaps it is a good time to elaborate on some of network conditions affect these parameters.

A very important factor affecting voice quality is the total network load. When the network load is high, and especially for networks with statistical access such as Ethernet, jitter and frame loss typically increase. For example, when using Ethernet, higher load leads to more collisions. Even if the collided frames are eventually sent over the network, they were not sent when intended to, resulting in excess jitter. Beyond a certain level of collisions, significant frame loss occurs.

While good network design takes into account the network load, it is not always under your control. However, even in congested networks it is sometimes possible to employ packet prioritization schemes, based on port numbers or on the IP precedence field. These methods, typically built into routers and switches, allow giving timing-sensitive frames such as voice priority over data frames. There is often no perceived degradation in the quality of data service, but voice quality significantly improves. Another alternative is to use bandwidth reservation protocols such as RSVP (resource reservation protocol) to ensure that the desired class of service is available to the specific stream.

To measure the network load, as well as the number of collisions, many different tools are available, including RADCOM protocol analyzers. To gauge the effect of priorities, it is possible to use the latency and loss application to ensure that priorities are configured correctly and indeed voice is given precedence over data traffic.

## Tunable factors in VoIP equipment

### *Jitter buffer settings*

The jitter buffer can be configured in most VoIP gear. The jitter buffer size must strike a delicate balance between delay and quality. If the jitter buffer is too small, network perturbations such as loss and jitter will cause audible effects in the received voice. If the jitter buffer is too large, voice quality will be fine, but the two-way conversation might turn into a half-duplex one.

One can decide on a jitter buffer policy that specifies that a certain percentage of packets should fit in the jitter buffer, say 95%. Since the utilization of the jitter buffer depends on the arrival times of the packets, it is useful to look at the jitter buffer problem using a few calculations, as automatically performed by the AudioPro:

Sequence number	Absolute time	Delta time (mSec)	Delta-Expected IAT (mSec)	Bias (mSec)	Normalized Bias
1	0:00:05.974	16.000	-4.000	-4.000	-6.449
2	0:00:06.006	31.000	11.000	7.000	4.551
3	0:00:06.036	31.000	11.000	18.000	15.551
4	0:00:06.036	0.000	-20.000	-2.000	-4.449
5	0:00:06.068	32.000	12.000	10.000	7.551
6	0:00:06.083	15.000	-5.000	5.000	2.551
7	0:00:06.099	16.000	-4.000	1.000	-1.449
8	0:00:06.115	15.000	5.000	4.000	6.449

**Figure 14 - Jitter buffer calculations using the Jitter Expert™**

In the above table, an audio-stream with a typical inter-packet emission time of 20 mSec is analyzed. the following columns are displayed:

- Sequence number. This designates the RTP sequence number of the incoming packet.
- Absolute time – the absolute arrival time of the packet.
- Delta time – the inter-arrival time (absolute time of each packet – absolute time of previous packet).
- Delay-Expected Inter-Arrival time – since the expected inter-arrival time is 20 mSec (the inter-emission time), this column shows how much the inter-arrival time deviated from the expected inter-arrival time. If all packets arrive exactly on schedule, this column will be always 0.
- Bias – The bias is cumulative sum of the delay-expected inter-arrival times, giving a very good measure of the desired jitter buffer. If all packets arrive on schedule, the delay-expected inter-arrival times will be zero, and no delay will be accumulated in the bias. However, if packets are consistently early or consistently late, the bias will grow. This emulates the operation of the jitter buffer. If the bias exceeds the size of the jitter buffer, packets will be simply dropped.
- Normalized bias – The bias column is normalized around zero.

From the Bias column it is now possible to determine the size of the desired jitter buffer. If no packets are to be lost, set the jitter buffer size to the maximum bias value. If you want the buffer to accommodate 95% of the packets, set the jitter size to the 95-percentile value of the bias.

Although this analysis was performed on only one stream, the AudioPro collects the data on all audio streams and allows performing statistical calculations across multiple streams.

### ***Packet size***

Packet size selection is also about balance. Larger packet sizes significantly reduce the overall bandwidth but add to the packetization delay as the sender needs to wait more time to fill up the payload.

Overhead in VoIP communications is quite high. Consider a scenario where you are compressing down to 8 Kbps and sending frames every 20 mSec. This results in voice payloads of 20 bytes for each packet. However, to transfer these voice payloads over RTP, the following must be added: an Ethernet header of 14 bytes, IP header of 20 bytes, UDP header of 8 bytes and an additional 12 bytes for RTP. This is a whopping total of 54 bytes overhead to transmit a 20-byte payload.

In some cases, such an overhead is fine. In others, there are two solutions to the problem:

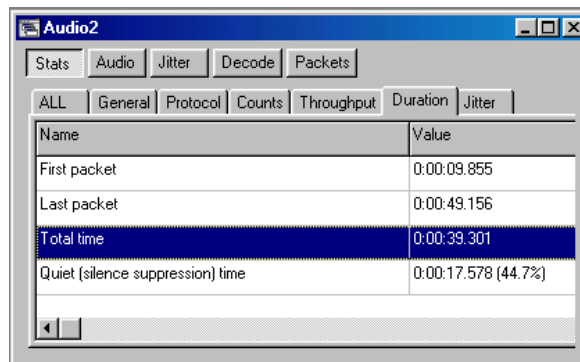
- Increase packet size. By deciding to send packets every 40 mSec, it is possible to increase the payload efficiency. Before the inter-arrival time is increased, it should be verified that the delay budget can support this.
- Employ header compression. Header compression is popular with some vendor's equipment, especially on slow links such as PPP, FR or ISDN. This is commonly called CRTP or Compressed RTP. It compresses the header down to a few bytes on a hop-by-hop basis. This can be done because the "logical channel" is determined by the FR DLCI and thus some header information is redundant.

The AudioPro displays the payload efficiency, average frame rate and average bit rate for each stream, so it is possible to make intelligent decisions on the suitability of the payload size for each network.

### ***Silence suppression***

Silence suppression takes advantage of prolonged periods of silence in conversations to reduce the number of packets. In a normal interactive conversation, each speaker typically listens for about half the time, so it is not necessary to transmit packets carrying the speaker's silence. Many vendors take advantage of this to reduce the bandwidth and number of packets on a link.

There is no discernible downside to employing silence suppression. To verify if silence suppression is turned on in various devices, and what the typical savings can be gained, the AudioPro reports the silence suppression statistics, as shown below for a NetMeeting recording:



The screenshot shows the 'Audio2' application window with a table of statistics. The table has two columns: 'Name' and 'Value'. The 'Total time' row is highlighted in blue. The 'Quiet (silence suppression) time' row shows a value of 0:00:17.578, which is 44.7% of the total time.

Name	Value
First packet	0:00:09.855
Last packet	0:00:49.156
<b>Total time</b>	<b>0:00:39.301</b>
Quiet (silence suppression) time	0:00:17.578 (44.7%)

Aside from verifying that silence suppression is turned on, these statistics allow planning for the expected utilization of a network.

### **Other performance issues in VoIP networks**

So-far this article has focused on quality and performance problems related to voice transmission once a voice call has been established. However, the call establishment process also presents a performance challenge, one that is clearly important to the customer. In particular, the following parameters should be noted:

- Call setup time, defined as the time required from the initial dialing of digits, to establishing a voice connection. Customers are accustomed to fast call setup times in the CSN world, and expect to get similar performance in the new VoIP network.
- Call success ratio, defined as the ratio of successful connects to dial attempts.

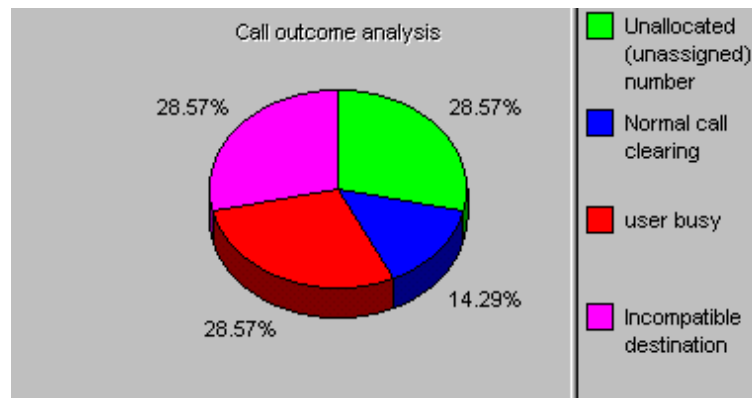
Also of importance to the service provider are:

- Call setup rate – how many calls/sec can be setup through the network. This determines the upper performance limit of the current devices.

Testing and measuring these parameters involves looking deeply into Q.931 messages, and analyzing the message sequences. Q.931 messages have a field called “call reference” that allows to distinguish one setup procedure from the other. These messages are interleaved with the normal data transfer, so it is sometimes difficult to fish for the Q.931 needle in the total packet haystack.

Fortunately, tools like the AudioPro separate the control plane message from the packet stream and present it in a call-oriented format. Using such tools, you will want to perform the following some or all of the following steps:

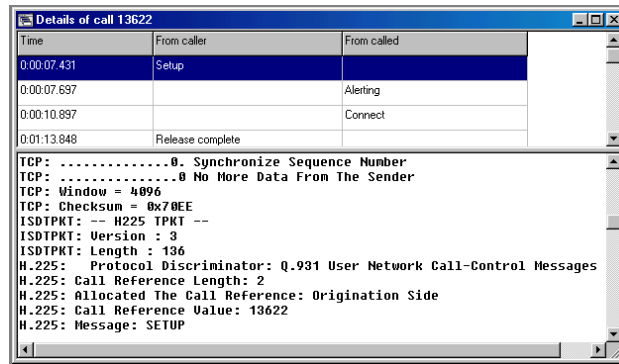
1. Analyze the call success ratio, including the ability to look at calls generated or received from specific phone numbers.



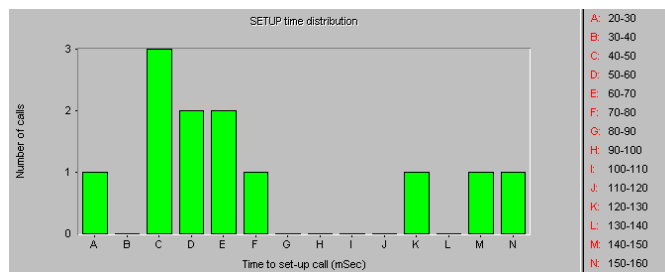
2. Look at the call list to identify individual problematic calls.

Num	To	Channel ID	Call Ref	Status
1	1234	ion:unknown	3085	Closed
2	1234	ion:unknown	13622	Closed
3	1234	ion:unknown	3086	Closed
4	1234	ion:unknown	13624	Closed
5	1234	ion:unknown	3087	Closed
6	1234	ion:unknown	13626	Closed
7	1234	ion:unknown	3088	Opened
8	1234	ion:unknown	13628	Opened

3. Drill down to specific calls to view conversation-based H.323 decoders.



4. Examine performance statistics such as call setup times.



## Can voice quality be measured?

With all the factors affecting voice quality, many people ask how one measures voice quality. Standards bodies like ITU are continuously addressing this issue, and have already derived two important recommendations: P.800 (MOS) and P.861 (PSQM). P.800 deals with defining a method to derive a Mean Opinion Score of voice quality. The test involves recording several pre-selected voice samples over the desired transmission media and then playing them back to a mixed group of men and women under controlled conditions. The scores given by this group are then weighed to give a single MOS score ranging between 1 (worst) and 5 (best). A MOS of 4 is considered “toll-quality” voice.

P.861 Perceptual Speech Quality Measurement (PSQM) tries to automate this process by defining an algorithm through which a computer can derive scores that have a close correlation to the MOS scores. While PSQM is useful, many people have voiced concerns over the suitability of this recommendation to packetized voice networks. It seems that PSQM was designed for the circuit-switched network and does not take into effect important parameters such as jitter and frame loss that are only relevant to VoIP.

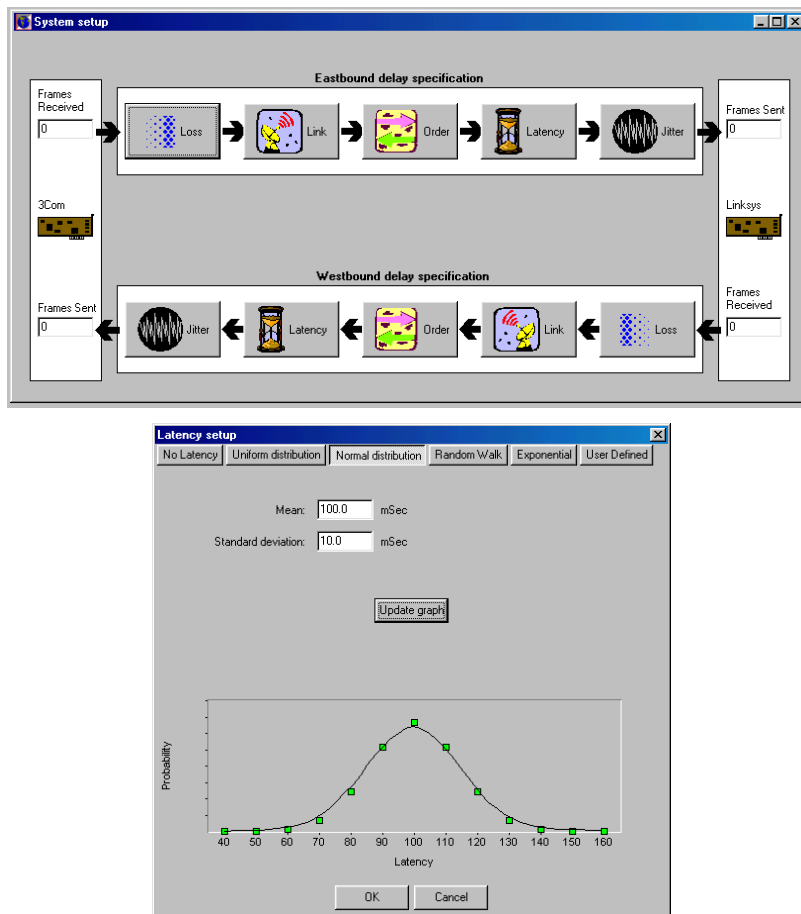
As a result of PSQM limitations, researchers are trying to come up with alternative objective ways to measure voice quality. One such proposal is the Perceptual Analysis/Masurement system (PAMS) developed by British Telecom. Tests conducted by BT have shown good correlation between automated PAMS scoring and manual MOS results.

Sometimes, you will have to be your own judge of quality. Tools like RADCOM's AudioPro extract individual voice calls and then decompress the captured data using the detected compression method to allow listening to the actual voice recording. By repeating this process at different points along the voice path it is not only possible to get a sense of quality, but also to determine at what point along the network voice quality degradation occurs.

## Simulating the effects of the network

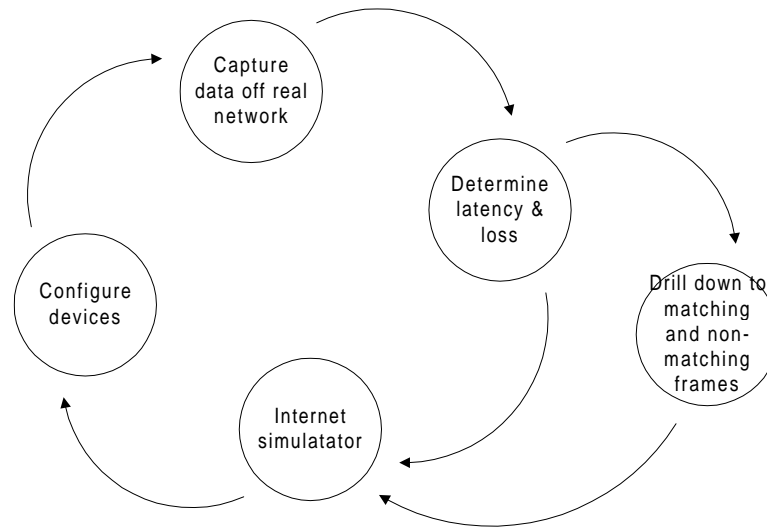
With so many potential pitfalls, it is often desirable to simulate the target network before deploying actual equipment. Such simulation allows you to gauge the effect of a sub-optimal communication link on voice quality and configure critical parameters such as jitter buffers before going into the field.

Tools such as RADCOM's Internet Simulator™ allow simulating the effect of the Internet or corporate Intranet on delay-sensitive data. Such solutions allow injection of controlled delay, jitter, out-of-sequence errors as well as frame loss.



**Figure 15 - Internet Simulator**

The Internet Simulator also works hand-in-hand with the latency and loss application and AudioPro, allowing the extraction network parameters from the customer network and their transfer into the Internet Simulator for simulation in the lab.



**Figure 16 - VoIP design and deployment planning cycle**

## Summary

Voice over IP services offer lucrative advantages to customers and service providers alike. However, as with any new technology, it brings its own sets of network design and optimization issues. By understanding the important parameters, and acquiring the proper tools, you can reap the benefits of voice over packet services.

## About RADCOM

RADCOM is a leading network test equipment manufacturer. The company specializes in the design, manufacture, marketing and support of a line of high-quality, integrated, multi-technology test solutions for LANs, WANs and ATM. RADCOM's test and analysis equipment is used in the development and manufacturing of network equipment; the installation of networks; and the ongoing maintenance of operational networks. RADCOM's sales and support network includes over 50 distributors in 35 countries worldwide and over 14 manufacturer's representatives across North America.

Further information on RADCOM's products can be obtained at the company's web-site, <http://www.radcom-inc.com> and our protocol information site <http://www.protocols.com>.