4

An Introduction to VoIP: End-to-End Elements and QoS Parameters

H. Toral-Cruz[1], J. Argaez-Xool[2], L. Estrada-Vargas[2] and D. Torres-Roman[2]
[1]*University of Quintana Roo (UQROO)*
[2]*Center of Research and Advanced Studies (CINVESTAV-IPN)*
Mexico

1. Introduction

In this chapter, two of the existing communication networks are studied: voice and data networks. Each network was created with the simple goal of transporting a specific type of information. For instance, the Public Switched Telephone Network (PSTN) was designed to carry voice and the IP network was designed to carry data.
In the PSTN, the main terminal device is a simple telephone set, while in the network, it is more complex, and it is provided with most of the intelligence necessary for providing various types for voice services. On the other hand, in the IP network the most of intelligence was placed in the terminal device, which is typically a host computer and the network only offers the best effort service (Park, 2005).
In mid 1990's, the two separate networks started to merge. A buzz word around this time is voice and data convergence. The idea is to create a single network to carry both voice and data.
However, with this convergence, a new technical challenge has emerged. In the converged network, the best effort services that are offered by the IP network is no longer good enough to meet requirements of real-time applications, such as Voice over Internet Protocol (*VoIP*).
VoIP refers to the transmission of voice using IP technologies over packet switched networks. It consists of a set of end-to-end elements, recommendations and protocols for managing the transmission of voice packets using IP. A basic *VoIP* system consists of three main elements: the *sender*, the *IP network* and the *receiver*.
VoIP is one of the most attractive and important service nowadays in communication networks and it demands strict *QoS* levels and real-time voice packet delivery. The *QoS* level of *VoIP* applications depends on many parameters, such as: bandwidth, One Way Delay (*OWD*), jitter, Packet Loss Rate (*PLR*), codec type, voice data length, and de-jitter buffer size. In particular, *OWD*, jitter, and *PLR* have an important impact.
This chapter presents an introduction to the main concepts and mathematical background relating to *communications networks*, *VoIP networks* and *QoS parameters*.

2. Communications networks

A communications network is a collection of terminals, links, and nodes which connect together to enable communication between users via their terminals. The network sets up a

connection between two or more terminals by making use of their source and destination addresses (Fiche & Hébuterne, 2004).
Switched networks are divided into circuit-switched and packet-switched networks. The packet-switched networks are further divided into connection-oriented and connectionless packet networks (Kurose & Ross, 2003; Tanenbaum, 2003; Stallings, 1997). Figure 1 shows this classification.

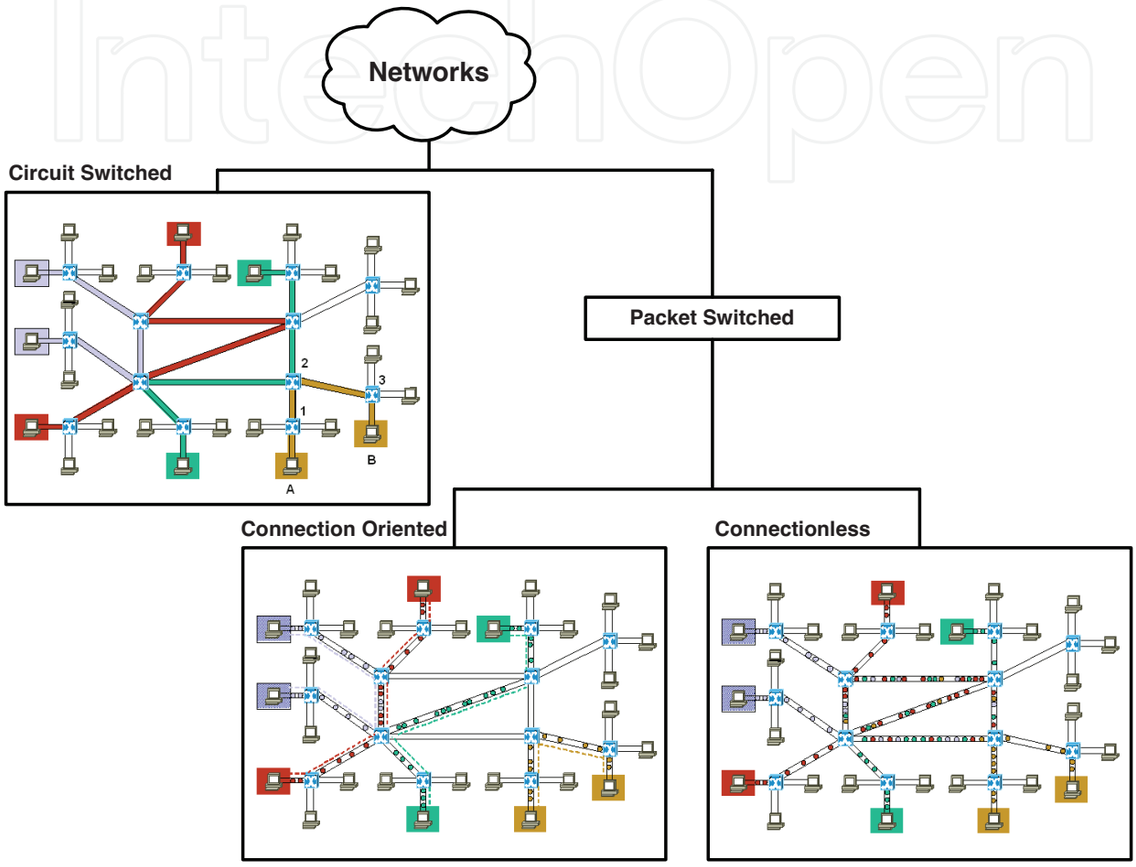


Fig. 1. Networks types

2.1 Circuit-switched network

Besides voice transport, circuit-switched networks are regularly used to transport different traffic types, such as data and control signals between computers and terminals, respectively. However, no matter which traffic type is transported, the user equipment and the set of nodes are called terminal and network, respectively. The network establishes the communication path between the terminals. The path is a connected sequence of links between nodes.
The communication via circuit-switched networks implies that there is a dedicated communication path between two or more terminals all through the communication session. Therefore, the resources (links and nodes) are reserved exclusively for information exchanges between origin and destination terminals. This communication involves three phases: circuit establishment, data transfer, and circuit disconnect. Before communication can occur between the terminals, a circuit is established between them. Thus, link capacity

must be reserved between each pair of nodes in the path, and each node must have available internal switching capacity to handle the requested connection. The nodes must have the intelligence to make these allocations and to devise a route through the network.
In circuit-switched networks, the nodes do not examine the contents of the information transmitted; the decision on where to send the information received is made just once at the beginning of the connection and remains the same for the duration of the connection. Thus, the delay introduced by a node is almost negligible. After the circuit has been established, the transmission delay is small and it is kept constant through the duration of the connection.
The circuit-switched networks can be rather inefficient. Once a circuit is established, the resources associated to it cannot be used for another connection until the circuit is disconnected. Therefore, even if at some point both terminals stop transmitting, the resources allocated to the connection remain in use.
The most common examples of circuit-switched network are the PSTN and the Integrated Services Digital Network (ISDN).

2.2 Packet-switched network

The data traffic is bursty and non-uniform. Terminals do not transmit continuously, i.e., are idle most of the time and very bursty at certain time. Data rates are not kept constant through the duration of the connection but they vary dynamically. A particular data transmission has a peak and average data rates associated to it and these are usually not the same. Therefore, employing dedicated circuits to transmit traffic with these characteristics is a waste of resources. The packet-switched network was first designed to fulfill the requirements of bursty traffic presented by data transmission.
In the packet-switched networks, the information is split up by the terminal into blocks of moderate size, called packets. These packets are autonomous, i.e., they are capable of moving on the network thanks to a header that contains the source and destination addresses. The packet is sent to the first node in this communication network.
The nodes are referred to as routers. When the router receives the packet, it examines the header and forwards the packet to the next appropriate router. This technique of inspection and retransmission is called store-and-forward, and it is accomplished in all routers of the path until the packet reaches its destination, unless the packet is lost. After reaching the destination, the destination terminal strips off the header of the packet to obtain the actual data that was originated at the source.
In this communication process, the terminal sends packets at its own rate, and the network multiplexes the packets from various origins in the same resources, to optimize their use. In this way several communications can share the same resources. The packet-switched network enables a better use of the transmission resource than circuit-switched network, in which the transmission resources are allocated without sharing. On the other hand, the multiplexing of different connections on the same resources causes delays and packet loss, which do not happen with circuit-switched network.
Finally it must be noted that in packet-switched networks a distinction is made between two modes of operation: connection-oriented mode and connectionless mode.
In connection-oriented mode, a path is established before any packets are sent; this path is called virtual circuit. There is a prior exchange of initial signaling packets to reserve resources and to establish the path. The connection-oriented mode is modeled after the telephone system. In order to talk to someone, one has to pick up the phone, dial the

number, talk, and then hang up. Similarly, in connection-oriented mode, the user establishes a connection, uses the connection, and then releases the connection. The essential aspect of a connection is that it acts like a tube, the sender pushes objects (packets) in at one end, and the receiver takes them out at the other end. In most cases the order is preserved so that the packets arrive in the order they were sent.
In connectionless mode, each packet is treated independently, with no reference to packets that have gone before and the routing decisions are taken at each node. The connectionless mode is modeled after the postal system. Each message (packet) carries the full destination address, and each one is routed through the system independently of all the others. Normally, when two packets are sent to the same destination, the first one sent will be the first one to arrive. However, it is possible that the first one sent can be delayed so that the second one arrives first.
The connectionless mode has been popularized mainly by Internet protocol. The IP networks have progressed to the point that it is now possible to support voice and multimedia applications, but does not guarantee quality of service, because are based on "best effort" services.

3. *VoIP* networks

A communications network is a collection of terminals, links, and nodes which connect *VoIP* is the real-time transmission of voice between two o more parties, by using IP technologies over packet-switched networks. It consists of a set of recommendations and protocols for managing the transmission of voice packets using the IP protocol.
Current implementations of *VoIP* have two main types of architectures, which are based on H.323 (ITU-T Recommendation H.323, 2007; Sulkin A., 2002) and Session Initiation Protocol (SIP) frameworks (Rosenberg et al 2002; Sulkin, 2002; Camarillo, 2002), respectively. H.323, which was ratified by International Telecommunication Union (ITU-T), is a set of protocols for voice, video, and data conferencing over packet-based network. SIP, which is defined in request for comments 3261 (RFC 3261) of the Multiparty Multimedia Session Control (MMUSIC) working group of Internet Engineering Task Force (IETF), is an application-layer control signaling protocol for creating, modifying, and terminating sessions with one or more participants. Regardless of their differences, the fundamental architectures of these two implementations are the same. They consist of three main logical components: terminal, signaling server, and GW. They differ in specific definitions of voice coding, transport protocols, control signaling, GW control, and call management.
The current H.323 and SIP frameworks do not provide *QoS*. *VoIP* is one of the most *QoS* sensitive and demands strict *QoS* levels. The *QoS* level of *VoIP* applications depends on many parameters, such as: bandwidth, *OWD*, jitter, PLR, codec type, voice data length, and de-jitter buffer size. In particular, *OWD*, jitter, and PLR have an important impact.

3.1 H.323 architecture

ITU-T H.323 (ITU-T Recommendation H.323, 2007; Sulkin A., 2002) is a set of protocols for voice, video, and data conferencing over packet-switched networks such as Ethernet Local Area Networks (LANs) and the Internet that do not provide a guaranteed *QoS*. The H.323 protocol stack is designed to operate above the transport layer of the underlying network. H.323 was originally developed as one of several videoconferencing recommendations issued by the ITU-T. The H.323 standard is designed to allow clients on H.323 networks to

communicate with clients on other videoconferencing networks. The first version of H.323 was issued in 1996, designed for use with Ethernet LANs and borrowed much of its multimedia conferencing aspects from other H.32.x series recommendations. H.323 is part of a large series of communications standards that enable videoconferencing across a range of networks. This series also includes H.320 and H.324, which address the ISDN and PSTN communications, respectively. H.323 is known as a broad and flexible recommendation. Although H.323 specifies protocols for real-time point-to-point communication between two terminals on a packet-switched network, also includes support of multipoint conferencing among terminals that support not only voice but also video and data communications. This recommendation describes the components of H.323 architecture. This includes terminals, Gateways (GW), Gatekeepers (GK), Multipoint Control Units (MCU), Multipoint Controller (MC), and Multipoint Processors (MP).

- *Terminal:* An H.323 terminal is an endpoint on the network which provides real-time, two-way communications with another H.323 terminal, GW, or MCU. This communication consists of control, indications, audio, moving color video pictures, and/or data between the two terminals. A terminal may provide speech only, speech and data, speech and video, or speech, data, and video.
- *Gateway:* The GW is a H.323 entity on the network which allows intercommunication between IP networks and legacy circuit-switched networks, such as ISDN and PSTN. They provide signaling mapping as well as transcoding facilities. For example, GWs receive an H.320 stream from an ISDN line, convert it to an H.323 stream, and then send it to the IP network.
- *Gatekeeper:* The GK is a H.323 entity on the network which performs the role of the central manager of *VoIP* services to the endpoints. This entity provides address translation and controls access to the network for H.323 terminals, GWs, and MCUs. The GK may also provide other services to the terminals, GWs, and MCUs such as bandwidth management and locating GWs.
- *MCU:* The MCU is an H.323 entity on the network which provides the capability for three or more terminals and GW to participate in a multipoint conference. It may also connect two terminals in a point-to-point conference which may later develop into a multipoint conference. The MCU consists of two parts, a mandatory MC, and an optional MP. In the simplest case, an MCU may consist only of an MC with no MPs. An MCU may also be brought into a conference by the GK without being explicitly called by one of the endpoints.
- *MC:* The MC is an H.323 entity on the network which controls three or more terminals participating in a multipoint conference. It may also connect two terminals in a point-to-point conference which may later develop into a multipoint conference. The MC provides the capability of negotiation with all terminals to achieve common levels of communications. It may also control conference resources such as who is multicasting video. The MC does not perform mixing or switching of audio, video, and data.
- *MP:* The MP is an H.323 entity on the network which provides for the centralized processing of audio, video and/or data streams in a multipoint conference. The MP provides for the mixing, switching, or other processing of media streams under the control of the MC. The MP may process a single media stream or multiple media streams depending on the type of conference supported.

The H.323 architecture is partitioned into zones. Each zone is comprised by the collection of all terminals, GW, and MCU managed by a single GK. H.323 is an umbrella

recommendation which depends on several other standards and recommendations to enable real-time multimedia communications. The main ones are:

- *Call Signaling and Control:* Call control protocol (H.225), media control protocol (H.245), security (H.235), digital subscriber signaling (Q.931), generic functional protocol for the support of supplementary services in H.323 (H.450.1), supplemental features (H.450.2-H.450.11).
- *H.323 Annexes:* Real-time facsimile over H.323 (Annex D), framework, and wire-protocol for multiplexed call signaling transport (Annex E), simple endpoint types - SET (Annex F), text conversation and Text SET (Annex G), Security for annex F (Annex J), hypertext transfer protocol (HTTP)-based service control transport channel (Annex K), stimulus control protocol (Annex L), and tunneling of signaling protocols (Annex M).
- *Audio Codec's*: Pulse Code Modulation (PCM) audio codec 56/64 kbps (G.711), audio codec for 7 Khz at 48/56/64 kbps (G.722), speech codec for 5.3 and 6.4kbps (G.723), speech codec for 16 kbps (G.728), and speech codec for 8/13 kbps (G.729).
- *Video Codec's:* Video codec for ≥ 64 kbps (H.261) and video codec for ≤ 64 kbps (H.263).

3.2 SIP architecture

SIP was developed by IETF in reaction to the ITU-T H.323 recommendation. The IETF believed that H.323 was inadequate for evolving IP telephony, because its command structure is complex and its architecture is centralized and monolithic. SIP is an application layer control protocol that can establish, modify, and terminate multimedia sessions or calls (Rosenberg et al 2002; Sulkin, 2002; Camarillo, 2002). SIP transparently supports name mapping and redirection services, allowing the implementation of ISDN and intelligent network telephony subscriber services. The early implementations of SIP have been in network carrier IP-Centrex trials. SIP was designed as part of the overall IETF multimedia data and control architecture that supports protocols such as Resource Reservation Protocol (RSVP), Real-time Transport Protocol (RTP), Real-time Streaming Protocol (RTSP), Session Announcement Protocol (SAP), and Session Description Protocol (SDP). SIP establishes, modifies, and terminates multimedia sessions. It can be used to invite new members to an existing session or to create new sessions. The two major components in a SIP network are User Agent (UA) and network servers (registrar server, location server, proxy server, and redirect server).

- *User Agents:* Is an application that interacts with the user and contains both a User Agent Client (UAC) and User Agent Server (UAS). A user agent client initiates SIP requests, and a user agent server receives SIP requests and returns responses on user behalf.
- *Registrar Server:* Is a SIP server that accepts only registration requests issued by user agents for the purpose of updating a location database with the contact information of the user specified in the request.
- *Proxy Server:* Is an intermediary entity that acts both as a server to user agents by forwarding SIP requests and as a client to other SIP servers by submitting the forwarded requests to them on behalf of user agents or proxy servers.
- *Redirect Server:* Is a SIP server that helps to locate UAs by providing alternative locations where the user can be reachable, i.e., provides address mapping services. It responds to a SIP request destined to an address with a list of new addresses. A redirect server does not accept calls, does not forward requests, and does not it initiate any of its own.

The SIP protocol follows a web-based approach to call signaling, contrary to traditional communication protocols. It resembles a client/server model; where SIP clients issue requests and SIP servers return one or more responses. The signaling protocol is built on this exchange of requests and responses, which are grouped into transactions. All the messages of a transaction share a common unique identifier and traverse the same set of hosts. There are two types of messages in SIP; requests and responses. Both of them use the textual representation of the ISO 10646 character set with UTF-8 encoding. The message syntax follows HTTP/1.1, but it should be noted that SIP is not an extension to HTTP.

- *SIP Responses:* Upon reception of a request, a server issues one or several responses. Every response has a code that indicates the status of the transaction. Status codes are integers ranging from 100 to 699 and are grouped into six classes. A response can be either final or provisional. A response with a status code from 100 to 199 is considered provisional. Responses from 200 to 699 are final responses.

1. 1xx Informational: Request received, continuing to process request. The client should wait for further responses from the server.
2. 2xx Success: The action was successfully received, understood, and accepted. The client must terminate any search.
3. 3xx Redirection: Further action must be taken in order to complete the request. The client must terminate any existing search but may initiate a new one.
4. 4xx Client Error: The request contains bad syntax or cannot be fulfilled at this server. The client should try another server or alter the request and retry with the same server.
5. 5xx Server Error: The request cannot be fulfilled at this server because of server error. The client should try with another server.
6. 6xx Global Failure: The request is invalid at any server. The client must abandon search.

The first digit of the status code defines the class of response. The last two digits do not have any categorization role. For this reason, any response with a status code between 100 and 199 is referred to as a "1xx response", any response with a status code between 200 and 299 as a "2xx response", and so on.

- *SIP Requests:* The core SIP specification defines six types of SIP requests, each of them with a different purpose. Every SIP request contains a field, called a method, which denotes its purpose.

1. INVITE: INVITE requests invite users to participate in a session. The body of INVITE requests contains the description of the session. Significantly, SIP only handles the invitation to the user and the user's acceptance of the invitation. All of the session particulars are handled by the session description protocol used. Thus, with a different session description, SIP can invite users to any type of session.
2. ACK: ACK requests are used to acknowledge the reception of a final response to an INVITE. Thus, a client originating an INVITE request issues an ACK request when it receives a final response for the INVITE.
3. CANCEL: CANCEL requests cancel pending transactions. If a SIP server has received an INVITE but has not returned a final response yet, it will stop processing the INVITE upon receipt of a CANCEL. If, however, it has already returned a final response for the INVITE, the CANCEL request will have no effect on the transaction.
4. BYE: BYE requests are used to abandon sessions. In two-party sessions, abandonment by one of the parties implies that the session is terminated.
5. REGISTER: Users send REGISTER requests to inform a server (in this case, referred to as a registrar server) about their current location.

6. OPTIONS: OPTIONS requests query a server about its capabilities, including which methods and which session description protocols it supports.

SIP is independent of the type of multimedia session handled and of the mechanism used to describe the session. Sessions consisting of RTP streams carrying audio and video are usually described using SDP, but some types of session can be described with other description protocols. In short, SIP is used to distribute session descriptions among potential participants. Once the session description is distributed, SIP can be used to negotiate and modify the parameters of the session and terminate the session.

3.3 *VoIP* system structure

VoIP is a rapidly growing technology that enables transport of voice over data networks such as Ethernet LANs. A basic *VoIP* system consists of three parts: the sender, the IP networks, and the receiver, as shown in Figure 2.

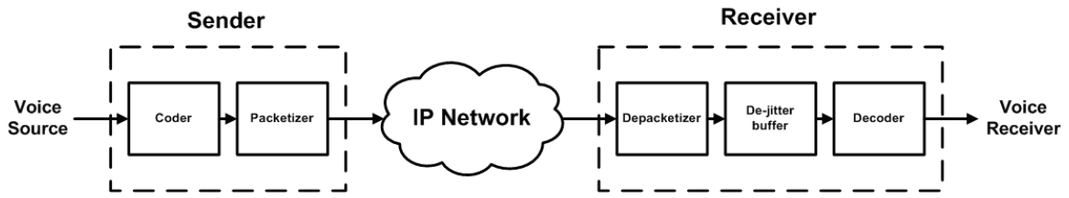


Fig. 2. *VoIP* system

Sender: The first component is the coder which periodically samples the original voice signal and assigns a fixed number of bits to each sample, creating a constant bit rate stream.

The voice stream from the voice source is first digitized and compressed by using a suitable coding algorithm such as G.711, G.729, etc. Various speech codec's differ from each other in terms of features such as coding bit-rate (kbps), algorithmic delay (ms), complexity, and speech quality (Mean Opinion Score - MOS). In order to simplify the description of speech codec's they are often broadly divided into three classes: waveform coders, parametric coders, or vocoders, and hybrid coders (as a combination thereof).

Typically waveforms codec's are used at high bit rates, and give very good quality speech. Parametric codec's operated at very low bit rates, but tend to produce speech which sounds synthetic. Hybrid codec's use techniques from both parametric and waveform coding, and give good quality speech an intermediate bit rates.

After compression and encoding into a suitable format the speech frames are packetized. The packetized process is implemented for gathering a set of voice data to be transmitted and adding the information needed for routing and handling those voice data across the IP network. The added bits are referred as the header, and the voice data to be delivered are referred as the payload. The structure of an IP packet over an Ethernet is shown in Figure 3.

The voice data length can be changed according to the *VoIP* transmission efficiency (*TE [%]*). The Media Access Control (MAC) header, IP header, User Datagram Protocol (UDP) header, RTP header, and Frame Check Sequence (FCS) are necessary for transmitting voice data over the Ethernet, while preamble and Inter Packet Gap (IPG) should be considered as occupied bandwidth on the transmission line. For instance, the total occupied bandwidth is 98 bytes including IPG, preamble, MAC header, IP header, UDP header, RTP header, and FCS when transmitting 20 byte voice data. The 78 bytes thus correspond to the overhead of IP transmission, so the ratio of voice data to the total is less than 25%, i.e. $TE\ [\%]=x/(x+y)*100$, where "x" is the voice data and "y" is the overhead.

The voice data length of an IP packet usually depends on the coding algorithm used. Eighty byte voice data is often used for G.711, whereas 20 byte voice data is used for G.729 in conventional *VoIP* communication. Table 1 shows the relationship between the voice data length in milliseconds and the voice data length in bytes, and Figure 4 shows the relationship between the voice data length and the bandwidth occupied by the *VoIP* frames of an Ethernet.

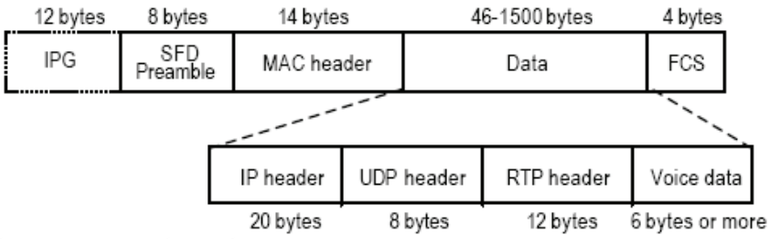


Fig. 3. *VoIP* packet structure for Ethernet

Voice Data Length (ms)	Voice Data Length (Bytes)	
	G.711	G.729
10	80	10
20	160	20
30	240	30
40	320	40
50	400	50
60	480	60
70	560	70
80	640	80
90	720	90
100	800	100

Table 1. Voice data length of *VoIP* packets

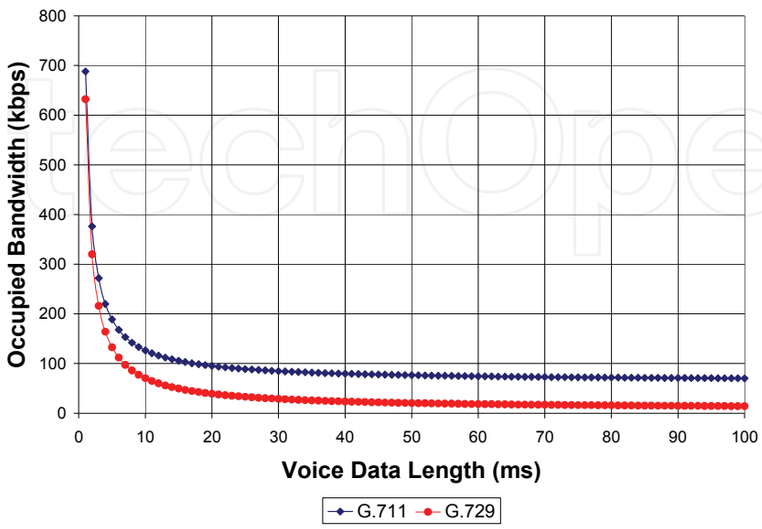


Fig. 4. Bandwidth occupied by *VoIP* frames

The longer the voice data length in a voice packet becomes, the more the transmission efficiency increases because the *VoIP* packet has overheads for the MAC header (in the case of the Ethernet), IP header, UDP header, and RTP header (see Figure 5). However, the longer one packet becomes, the more packet errors are likely to occur, so it is important to evaluate how the network traffic conditions affect the packet behavior and *QoS* in *VoIP* systems.

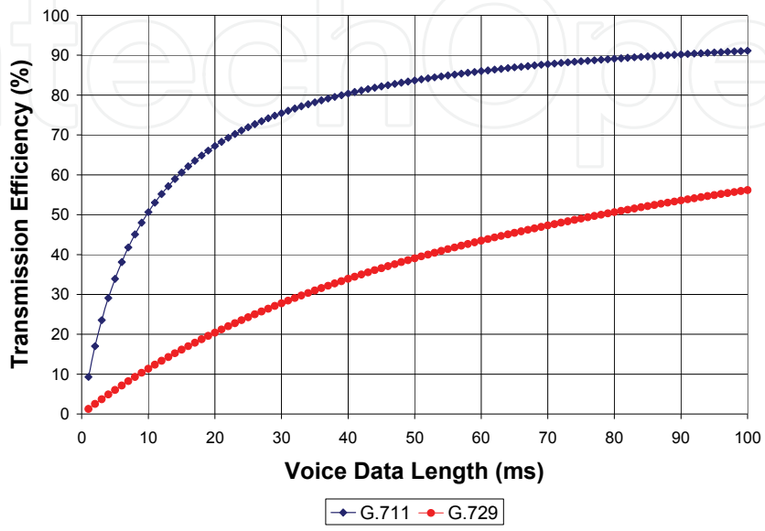


Fig. 5. Transmission efficiency

IP Network: Due to the shared nature of IP network, guaranteeing the quality of service of Internet applications from end-to-end is difficult. Since current IP networks are based on best-effort services, the packet may suffer different network impairments (e.g. packet loss, delay, and jitter), which directly impact the quality of *VoIP* applications.

Receiver: The packet headers are stripped off and voice samples are extracted from the payload by depacketizer. The voice samples must be presented to the decoder in such a way that the next sample is present for processing when the decoder has finished with its immediate predecessor. Such a requirement severely constrains the amount of jitter that can be tolerated in a *VoIP* system without having to gap the samples. When jitter results in an Inter-arrival Time (*IAT*) that is greater than the time required to re-create the waveform from a sample, the decoder has no option but to continue to function without the next sample information. Therefore, the effects of jitter will be manifested as an increase in the packet loss rate.

The buffer that holds the queued segments is called de-jitter buffer. The employment of such de-jitter buffers defines the relationship between jitter and packet loss rate in the receiver side. The delay variation that can be tolerated becomes therefore the essential descriptor of intrinsic quality that supplants jitter.

Therefore an important design parameter at the receiver side is the de-jitter buffer size or playout delay of a de-jitter buffer. Since, de-jitter buffer is used to compensate for network jitter at the cost of further delay (buffer delay) and loss (late arrival loss). Finally, the de-jittered speech frames are decoded to recover the original voice signal.

The playout delay is explained in the next few paragraphs. Figure 6 and Table 2 show a sample of packet delays. In the Figure6, the abscissa is the time at which a packet is sent, the

ordinate is the time at which the packet is received, and the time here is measured by a globally synchronized clock. In an ideal network such as circuit switched network, the delay d for a given path is constant and low, so the (x,y) points form a line $y=x+d$ (e.g. $d=5ms$). This means packets can be played out as soon as they are received without having to pause (ideal playout time, $y=x+5$). In the packet switched network such as the Internet, delays are not constant, as queuing delays can vary significantly over time.

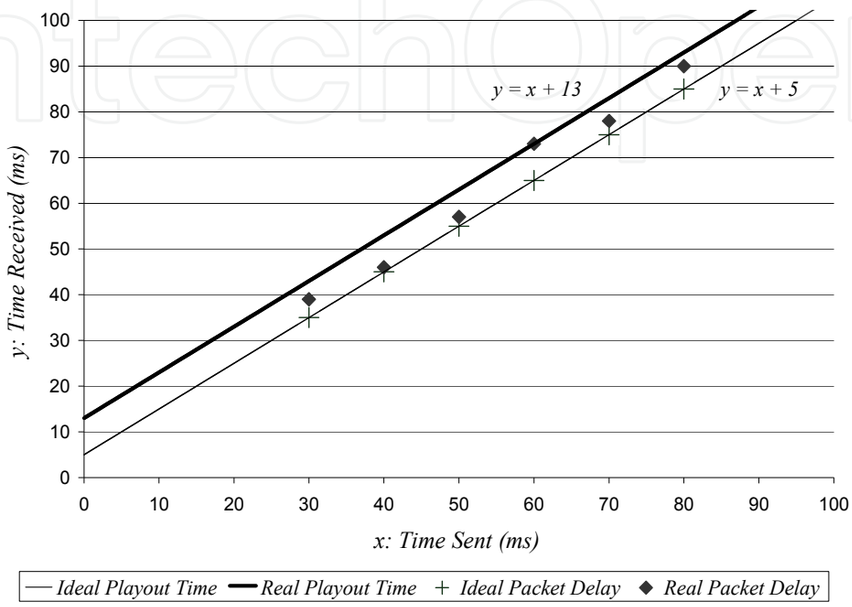


Fig. 6. Illustration of packet delay and playout delay

Time Sent (ms)	Ideal: Time Received (ms)	Real: Time Received (ms)
30	35	39
40	45	46
50	55	57
60	65	73
70	75	78
80	85	90

Table 2. Sample of packet delays

An example is the diamond-shaped plot in Figure 6. For *VoIP* applications, if the receiver plays out voice packets as they come in, it will have to generate a pause if the next packet delayed arrives. Therefore, in Figure 6, we must wait at least d_{play} time to prevent this situation. The term d_{play} is called the playout delay. Usually, d_{play} is calculated by subtracting the actual play time of the first packet from its receiving time. In this example, the first packets is sent at 30 ms, received at 39 ms, and played at 43 ms. Therefore, d_{play} is 4 ms, since the actual play times of all packets form a line $y=x+13$ (real playout time). An alternative definition of playout delay is the delay between sending time and playout time.

Many techniques have been developed for controlling the playout delay. Most existing playout adaptation algorithms work by taking some measurements on the delays experienced by packets and updating the playout delay.

4. *QoS* parameters

The voice quality of *VoIP* applications depends on many parameters, such as bandwidth, *OWD*, jitter, packet loss rate, codec, voice data length, and de-jitter buffer size. In particular, *OWD*, jitter, and packet loss have an important impact on voice quality.

4.1 One way delay

The delay experienced by a packet across a path consists of several components: propagation, processing, transmission, and queuing delays (Park, 2005). The Internet metric called one way delay (ITU-T Recommendation G.114, 2003) is the time needed for a packet to traverse the network from a source to a destination host. It is described analytically by Equation (1):

$$D^K(L)_{OWD} = \delta + \sigma + \sum_{h=1}^{s}\left(\frac{L}{C_h} + X_h^K(t)\right) \tag{1}$$

where $D^K(L)_{OWD}$ is the *OWD* of a packet K of size L, δ represents the propagation delay, σ the processing delay, s the number of hops, L/C_h the transmission delay, and $X_h^K(t)$ the queuing delay of a packet K of size L at hop h $(h=1,...,s)$ with capacity C_h. The *OWD* variation between two successive packets, K and $K-1$ is called *OWD* jitter and is given by the Equation (2):

$$J^K(L) = D^K(L)_{OWD} - D^{K-1}(L)_{OWD} \tag{2}$$

4.2 Jitter

When voice packets are transmitted from source to destination over IP networks, packets may experience variable delay, called delay jitter. The packet *IAT* on the receiver side is not constant even if the packet Inter-departure Time (*IDT*) on the sender side is constant. As a result, packets arrive at the destination with varying delays (between packets) referred to as jitter. We measure and calculate the difference between arrival times of successive voice packets that arrive on the receiver side, according to RFC 3550 (Schulzrinne et al, 2003), this is illustrated in Figure 7.

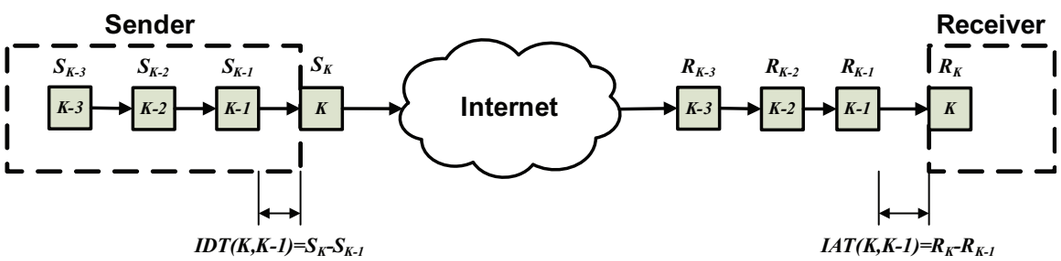


Fig. 7. Jitter experienced across Internet paths

Let S_K denote the transmission timestamp for the packet K of size L, and R_K the arrival time for packet K of size L. Then for two packets K and $K-1$, $J^K(L)$ may be expressed as:

$$J^K(L) = (R_K - S_K) - (R_{K-1} - S_{K-1}) = (R_K - R_{K-1}) - (S_K - S_{K-1}) \quad (3)$$

$$IDT(K, K-1) = (S_K - S_{K-1}) \quad (4)$$

$$IAT(K, K-1) = (R_K - R_{K-1}) \quad (5)$$

where, $IDT(K, K-1)$ is the Inter-departure Time (in our experiments, *IDT*= {10ms, 20ms, 40ms, and 60ms}) and $IAT(K, K-1)$ is the Inter-arrival Time for the packets K and $K-1$. In the current context, $IAT(K, K-1)$ is referred to as jitter. So, the *VoIP* jitter between two successive packets, i.e., packets K and $K-1$, is:

$$IAT(K, K-1) = J^K(L) + IDT(K, K-1) \quad (6)$$

4.3 Packet loss

There are two main transport protocols used in IP networks: UDP and TCP. While UDP protocol does not allow any recovery of transmission errors, TCP include an error recovery process. However, the voice transmission over TCP connections is not very realistic. This is due to the requirement for real-time operations in most voice related applications. As a result, the choice is limited to the use of UDP which involves packet loss problems.

Amongst the different quality elements, packet loss is the main impairment which makes the *VoIP* perceptually most different from the public switched telephone network. Packet loss can occur in the network or at the receiver side, for example, due to excessive network delay in case of network congestion.

Owing to the dynamic, time varying behavior of packet networks, packet loss can show a variety of distributions. The packet loss distribution most often studied in speech quality tests is random or Bernoulli-like packet loss. Uniform random loss here means independent loss, implying that the loss of a particular packet is independent of whether or not previous packets were lost. However, uniform random loss does not represent the loss distributions typically encountered in real networks. For example, losses are often related to periods of network congestion. Hence, losses may extend over several packets, showing a dependency between individual loss events. In this work, dependent packet loss is often referred to as bursty. The packet loss is bursty in nature and exhibits temporal dependency (Yajnik et al, 1999). So, if packet *n* is lost then normally there is a higher probability that packet *n* + 1 will also be lost. Consequently, there is a strong correlation between consecutive packet losses, resulting in a bursty packet loss behavior. A generalized model to capture temporal dependency is a finite Markov chain (ITU-T Recommendation G.1050, 2005).

2-state Markov Chain: Figure 8 shows the state diagram of a 2-state Markov chain.

In this model, one of the states (S_1) represents a packet loss and the other state (S_2) represents the case where packets are correctly transmitted or received. The transition probabilities in this model, as shown in Figure 8, are represented by p_{21} and p_{12}. In other words, p_{21} is the probability of going from S_2 to S_1, and p_{12} is the probability of going from S_1 to S_2. Different values of p_{21} and p_{12} define different packet loss conditions that can occur on the Internet.

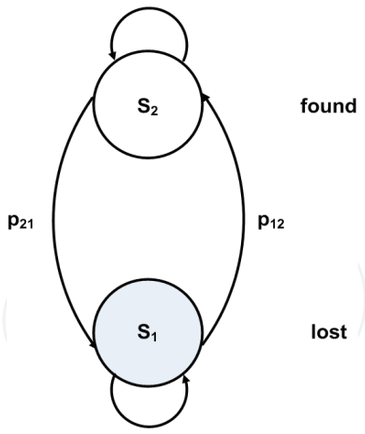


Fig. 8. 2-state Markov chain

The steady-state probability of the chain to be in the state *S1*, namely the *PLR*, is given by Equation (7):

$$PLR = S_1 = \frac{p_{21}}{p_{21} + p_{12}} \tag{7}$$

and clearly $S_2 = 1 - S_1$.

The distributions of the number of consecutive received or lost packets are called gap ($f_g(k)$) and burst ($f_b(k)$) respectively, and can be expressed in terms of p_{21} and p_{12}. The probability that the transition from S_2 to S_1 and S_1 to S_2 occurs after k steps can be expressed by Equations (8) and (9):

$$f_g(k) = p_{21}(1 - p_{21})^{k-1} \tag{8}$$

$$f_b(k) = p_{12}(1 - p_{12})^{k-1} \tag{9}$$

According to Equation (9), the number of steps k necessary to transit from S_1 to S_2, that is, the number of consecutively lost packets is a geometrically distributed random variable. This geometric distribution of consecutive loss events makes the 2-state Markov chain (and higher order Markov chains) applicable to describing loss events observed in the Internet.

The average number of consecutively lost and received packets can be calculated by $\overline{b}$ and $\overline{g}$, respectively, as shown in Equations (10) and (11).

$$\overline{b} = E\{f_b(k)\} = \frac{1}{p_{12}} \tag{10}$$

$$\overline{g} = E\{f_g(k)\} = \frac{1}{p_{21}} \tag{11}$$

4-state Markov Chain: Figure 9 shows the state diagram of this 4-state Markov chain.

In this model, a '*good*' and a '*bad*' state are distinguished, which represent periods of lower and higher packet loss, respectively. Both for the '*bad*' and the '*good*' state, an individual 2-state Markov chain represents the dependency between consecutively lost or found packets.

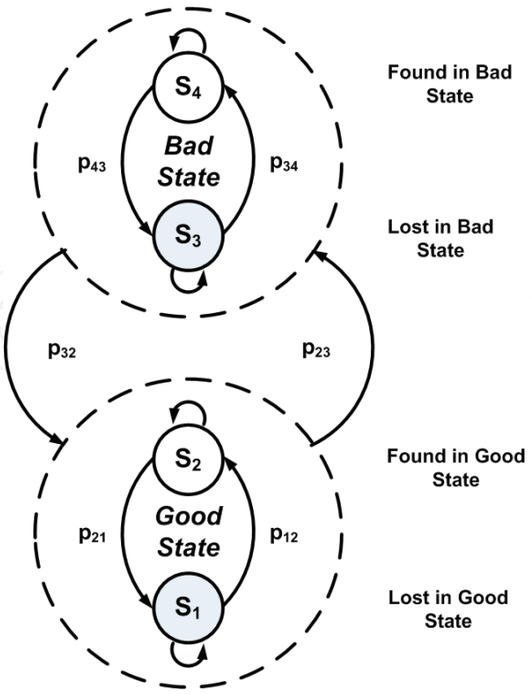


Fig. 9. 4-state Markov chain

The two 2-state chains can be described by four independent transition probabilities (two each one). Two further probabilities characterize the transitions between the two 2-state chains, leading to a total of six independent parameters for this particular 4-state Markov chain.

In the 4-state Markov chain, states S_1 and S_3 represent packets lost, S_2 and S_4 packets found and six parameters ($p_{21}, p_{12}, p_{43}, p_{34}, p_{23}, p_{32} \in (0,1)$) are necessary to define all the transition probabilities.

In the "*good state*" (*G*) packet loss occur with (*low*) probability P_G while in the "*bad state*" (*B*) they occur with (*high*) probability P_B. The occupancy times for states *B* and *G* are both geometrically distributed with respective means $\frac{1}{p_{32}}$ and $\frac{1}{p_{23}}$, respectively. The steady state probabilities of being in states *G* and *B* are $\pi_G = \frac{p_{32}}{p_{32}+p_{23}}$ and $\pi_G = \frac{p_{32}}{p_{32}+p_{23}}$, respectively.

The overall packet loss rates in the '*good*' and '*bad*' states P_G and P_B can be calculated by the following Equations:

$$P_G = \frac{p_{21}}{p_{21}+p_{12}} \tag{12}$$

$$P_B = \frac{p_{43}}{p_{43}+p_{34}} \tag{13}$$

The overall packet loss for the four-state Markov model is given by:

$$PLR = P_G \cdot \pi_G + P_B \cdot \pi_B \tag{14}$$

5. Conclusion

VoIP has emerged as an important service, poised to replace the circuit-switched telephony service in the future. However, when the voice traffic is transported over Internet, the packet based transmission may introduce degradations and have influence on the *QoS* perceived by the end users. The current Internet only offers best-effort services and was designed to support non-real-time applications. *VoIP* demands strict *QoS* levels and real-time voice packet delivery.

The voice quality of *VoIP* applications depends on many parameters, such as: bandwidth, *OWD*, jitter, *PLR*, codec, voice data length, and de-jitter buffer size. In particular, packet loss, *OWD* and jitter have an important impact on voice quality.

This chapter presents an introduction to the main concepts and mathematical background relating to communications networks, *VoIP* networks and *QoS* parameters.

6. References

Camarillo, G. (2002). *SIP Demystified*. USA: McGraw-Hill Companies, Inc.

Fiche, G., & Hébuterne, G. (2004). *Communicating Systems & Networks: Traffic & Performance*. London and Sterling, VA: Kogan Page Science.

ITU-T Recommendation G.114, (2003). *One-Way Transmission Time*. International Telecommunications Union, Geneva, Switzerland.

ITU-T Recommendation G.1050, (2005). *Network Model for Evaluating Multimedia Transmission Performance over Internet Protocol*. International Telecommunications Union, Geneva, Switzerland.

ITU-T Recommendation H.323, (2007). *Packet-Based Multimedia Communications Systems*. International Telecommunications Union, Geneva, Switzerland.

Kurose, J., & Ross, K. (2003). *Computer Networking: A Top-Down Approach Featuring the Internet*. USA: Pearson Education, Inc.

Park, K. I. (2005). *QoS in Packet Networks*. Boston, MA: Springer Science + Business Media, Inc.

Rosenberg, J., et al (2002). *SIP: Session Initiation Protocol (RFC 3261)*. Internet Engineering Task Force.

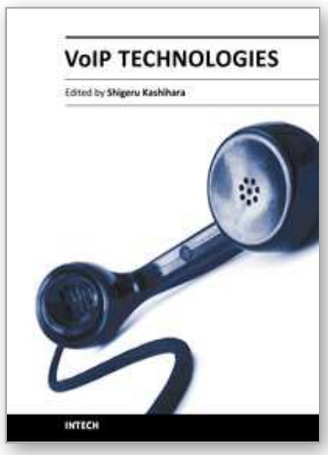
Schulzrinne, H., et al (2003). *RTP: A Transport Protocol for Real-Time Applications (RFC 3550)*. Internet Engineering Task Force.

Stallings. W. (1997). *Data and Computer Communications*. Upper Saddle River, NJ: Pearson Education, Inc.

Sulkin, A. (2002). *PBX Systems for IP Telephony: Migrating Enterprise Communications*. New York, NY: McGraw-Hill Professional.

Tanenbaum, A. S. (2003). *Computer Networks*. Upper Saddle River, NJ: Pearson Education, Inc.

Yajnik, M., Moon, S., Kursoe, J., & Towsley, D. (1999). *Measurement and Modelling of the Temporal Dependence in Packet Loss*. Paper presented at the 18th International Conference on Computer Communications (IEEE INFOCOM), New York, NY.



VoIP Technologies

Edited by Dr Shigeru Kashihara

ISBN 978-953-307-549-5
Hard cover, 336 pages
Publisher InTech
Published online 14, February, 2011
Published in print edition February, 2011

This book provides a collection of 15 excellent studies of Voice over IP (VoIP) technologies. While VoIP is undoubtedly a powerful and innovative communication tool for everyone, voice communication over the Internet is inherently less reliable than the public switched telephone network, because the Internet functions as a best-effort network without Quality of Service guarantee and voice data cannot be retransmitted. This book introduces research strategies that address various issues with the aim of enhancing VoIP quality. We hope that you will enjoy reading these diverse studies, and that the book will provide you with a lot of useful information about current VoIP technology research.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen