WHITE PAPER

Understanding VoIP

Network Characteristics, Protocols, and Test Tools

Zetron Americas
PO Box 97004, Redmond, WA USA 98073-9704
(P) 1 425 820 6363 **(F)** 1 425 820 7031 **(E)** zetron@zetron.com
Zetron EMEA
27-29 Campbell Court, Bramley, Hampshire RG26 5EG, United Kingdom
(P) +44 (0)1256 880663 **(F)** +44 1256 880491 **(E)** uk@zetron.com
Zetron Australasia
PO Box 3045, Stafford Mail Centre, Stafford QLD 4053, Australia
(P) +61 7 3856 4888 **(F)** +61 7 3356 6877 **(E)** au@zetron.com

www.zetron.com
June, 2011 005-1389D

ZETRON®

Table of Contents

1 Introduction: Why VoIP Matters

Voice-over-Internet Protocol (VoIP) is here to stay.

VoIP is a technology that transfers voice signals over packet-based Internet Protocol (IP) networks. Due to the popularity of the Internet and the widespread deployment of private IP networks, VoIP has gained tremendous attention and support from equipment developers, service providers, and customers. Indeed, it has been touted as the next generation technology to replace the current Public Switch Telephone Network (PSTN).

Many are understandably uneasy with the transition to VoIP, however. It is a new and complex technology that must be understood in order for its advantages to be fully appreciated and its disadvantages to be appropriately addressed.

The purpose of this paper is to explain and demystify VoIP technology. The paper discusses the basics of VoIP, including various VoIP protocols and related technologies. It also discusses the benefits and pitfalls of VoIP.

Armed with an understanding of VoIP, readers will be better prepared to use and fully realize the benefits of this promising and burgeoning technology.

The enthusiasm for VoIP exists because IP networks are ubiquitous and relatively inexpensive. IP networks are packet-based networks that use the IP protocol to route packets from source to destination. All data to be transferred from source to destination must be carried by IP packets. VoIP works by digitizing and encoding voice into digital frames and then sending them via IP packets at the source. At the destination, digital voice frames are reassembled from the received IP packets, then decoded and converted back into analog format. Fig. 1-1 shows a simplified VoIP voice flow diagram.

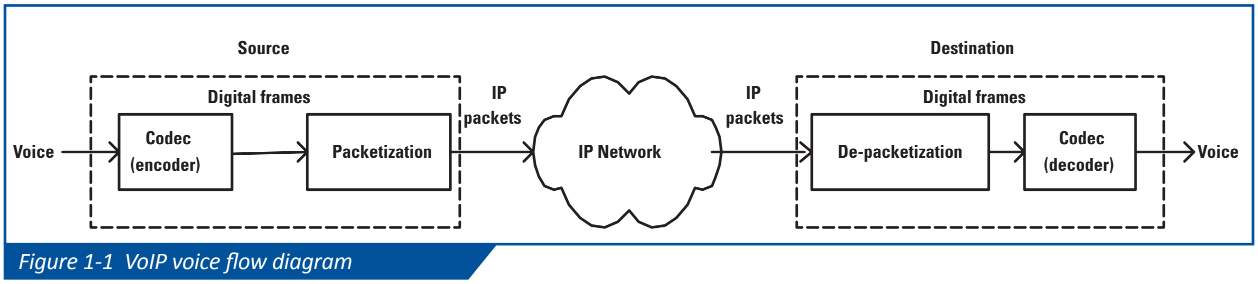


Figure 1-1 VoIP voice flow diagram

In addition to voice data, a typical voice communication requires call control and signaling (for example, to set up and tear down a call). In VoIP, call control and signaling messages are also carried by IP packets.

Like any other IP-based technology, VoIP uses layered protocols to manage and transport voice data and control messages. On top of the common IP protocol, there is TCP (Transmission Control Protocol), a reliable transport protocol; and UDP (User Datagram Protocol), an unreliable transport protocol. TCP is suitable for reliable and secure message transport, while UDP is suitable for real time data transport. On top of TCP or UDP, VoIP uses different application protocols to handle and transfer voice data and control messages. For example, RTP (Real-Time Transport Protocol) is the most used application protocol for streaming voice data, and SIP (Session Initiation Protocol) is a popular application protocol for VoIP call control and signaling.

The most important advantage of using VoIP is cost savings. Because VoIP shares the underlying network with other IP-based applications and it only uses the network when there is data to send, the cost of communication is much lower with VoIP than with traditional circuit-based voice communication. This is especially true when the underlying IP network (including the Internet) already exists for data applications. Other advantages of using VoIP include:

- Easy to add enhanced features such as caller ID, call forwarding, instant messaging, video, and multimedia conferencing
- Easy to encrypt voice data for secure communications
- Easy to add new phone lines or numbers
- Can use a low bit rate voice codec to reduce bandwidth usage
- Can use silence suppression to reduce bandwidth usage

Even though VoIP has many advantages, there are some challenges and concerns that could slow down its widespread adoption. Most of the challenges and concerns are due to the inherent nature of IP-based networks. The following lists some of them:

The quality of service (QoS) is not guaranteed

- End-to-end voice delay is long
- Echo is more noticeable (if not properly controlled) due to long network delay
- Voice quality could be degraded due to network impairments including delay, jitter, and packet loss
- Voice quality is less than toll quality if a low bit rate voice codec is used
- Vulnerable to different types of security attack

Regardless of the above challenges and concerns, the popularity of VoIP continues to grow. New network technologies are being developed to overcome some of these challenges and concerns. Today, VoIP is adopted in almost every type of communication network, including cellular networks, public safety radio networks, and Wi-Fi networks.

The purpose of this document is to introduce VoIP to people who do not have deep knowledge of VoIP. The topics covered in this document include network and Ethernet technologies, IP addresses and subnets, Internet Protocol suite, VoIP call control protocol SIP, VoIP media transport protocol RTP, voice codecs, bandwidth requirements, network impairments, voice quality measurement, quality of service, security, multicast routing, and test tools. For more detailed explanation about any topic, the readers should refer to the relevant VoIP books or tutorials available on the Web.

2 Network and Ethernet Technologies

2.1 Network Technology

Generally speaking, communication networks can be divided into two basic types: circuit-switched (or connection-oriented) and packet-switched (or connectionless). The telephone system is an example of a circuit-switched network. When you make a phone call, a dedicated connection (or circuit) is established between you and the person you called. The circuit capacity and the quality of service are guaranteed throughout the entire call. The main advantage of circuit-switching is guaranteed capacity; once a connection is established, the capacity of a circuit is guaranteed. One of the disadvantages of circuit-switching is the cost structure; cost is fixed based on connection, independent of usage.

On the other hand, the packet-switched networks are connectionless. In a packet-switched network, data to be transferred across the network is divided into small pieces called packets. A packet carries special information in the header that enables the network hardware to know how to send it to the specified destination. Because network resources are only required when there is data to send, the same network resource can be shared by many communications concurrently. This leads to more efficient bandwidth usage and therefore lower cost per communication. Because network resources are shared, the network could become overloaded, causing longer delay and packet drops. This is the main disadvantage of packet-switching; network capacity is not guaranteed. The Internet is an example of a packet-switched network. It was originally designed to transport data only. However, due to recent advances in network technology, real-time audio and video streaming over the Internet has become very common.

Packet-switched networks can be loosely divided into two broad types based on geographical distance: local area network (LAN) and wide area network (WAN). A LAN covers a small geographical area such as a home, a single building, a group of buildings, or a small campus. Most LANs connect personal computers (PCs) and servers. Because the distance is short, a LAN can operate at a very high speed with short delay. A WAN covers a relatively broad geographical area such as a city, a state, or even a country. WANs are used to connect LANs together over long distances. Compared to LANs, WANs operate at slower speeds and have much longer delay.

2.2 Ethernet Technology

Ethernet is the most popular technology for interconnecting LAN devices. Standardized as IEEE 802.3, it defines a number of wiring and signaling standards at the physical layer, the means of network access at the data link layer, and a common addressing format. Ethernet devices can operate over different types of medium such as twisted-pair cable, coaxial cable, and fiber optics. The three commonly used data rates for operation over twisted-pairs cables are:

- 10 Mbps (Mega-bits/sec): defined in 10BASE-T standard
- 100 Mbps: defined in 100BASE-TX standard under Fast Ethernet
- 1000 Mbps: defined in 1000BASE-T standard under Gigabit Ethernet

Each Ethernet device (or transceiver) is assigned a unique 48-bit address known as its Ethernet address or Media Access Control (MAC) address. The address is usually written in hexadecimal form, for example, 00A0C9357F1D. Ethernet addresses are managed by the IEEE. Companies that manufacture Ethernet devices must purchase blocks of Ethernet addresses and assign them in sequence to their Ethernet devices. An Ethernet address is permanently bound to the hardware device and should not be changed by software. In addition to the unique addresses assigned to Ethernet devices, there are many addresses reserved for multicast and one address reserved for broadcast.

Ethernet technology was originally designed to use a shared medium (coaxial cable) with a bus topology where all Ethernet devices in the network are attached to the common medium. A scheme called “carrier sense multiple access with collision detection (CSMA/CD)” was used to avoid two devices transmitting at the same time. However, this limits communication, because it can only occur in one direction at any given time (half-duplex). As Ethernet technology progressed, Ethernet hubs were introduced to connect multiple Ethernet devices together to form a single segment. An Ethernet hub is a fairly unsophisticated broadcast device with multiple ports. When it receives a signal (or packet) from one port, it simply broadcasts the signal out on every other port. It does not inspect the traffic passing through it. With Ethernet hubs, all Ethernet devices have to operate in half-duplex because signal collision could still occur (when two devices transmit at the same time into an Ethernet hub).

To eliminate the problem of signal collision and allow for full-duplex operations, Ethernet switches were created. An Ethernet switch is an intelligent network device with multiple ports. Each port in a switch has its own isolated collision domain. Therefore, an Ethernet device connected to a switch port can operate in full-duplex because the link to the switch port is a point-to-point link without other devices sharing the same link. An Ethernet switch inspects all the packets entering its ports and decides how to forward them. Each Ethernet packet carries a source Ethernet address and a destination Ethernet address. By observing the source Ethernet addresses in the packets entering its ports, a switch can learn the Ethernet address of the device connected to each of its ports. When a switch receives an Ethernet packet, it compares the destination Ethernet address in the packet to the learned device Ethernet address of each of its ports. If a match is found, the switch will forward the packet only to the port with the matched Ethernet address. If no match is found, the switch will forward the packet to every port except the port of entry. With intelligent packet forwarding and full-duplex communication, a switch offers much better performance than a hub. Therefore, switches have replaced hubs as the primary network equipment for interconnecting Ethernet devices. Due to the widespread use of switches, the modern Ethernet looks very different from the early Ethernet. Ethernet technology has transformed from half-duplex "shared Ethernet" to full-duplex "switched Ethernet".

Even though Ethernet devices can operate at different speeds and different duplex modes, two Ethernet devices (including switch ports) connected to the same link must be configured to operate at the same speed and same duplex mode. Auto-negotiation is a procedure by which two connected Ethernet devices advertise their capabilities and choose the best speed and duplex mode that both can support. With auto-negotiation, Ethernet devices can configure themselves to use common transmission parameters. However, not all Ethernet devices support auto-negotiation. Sometimes, for performance reasons, auto-negotiation is disabled on purpose by the manufacturer or user. When an Ethernet device capable of auto-negotiation is connected to a device that does not support auto-negotiation or whose auto-negotiation capability is disabled, the device capable of auto-negotiation can detect the speed of the non-auto-negotiating device and match it, but the duplex mode is always half duplex. As long as both connected devices operate with the same speed and same duplex mode, the link will work fine. However, when one device operates in full-duplex, while the other device operates in half-duplex, the link throughput will drop to a much slower rate than the link speed. This situation is called "duplex mismatch", which must be avoided. Therefore, when connecting an Ethernet device into a switch, the user should be aware of the auto-negotiation capabilities of the Ethernet device and the switch. If the Ethernet device does not support auto-negotiation or its auto-negotiation capability is disabled, the user should use a "managed switch", which allows the user to configure its port parameters such as speed and duplex mode to match the parameters used by the connected Ethernet devices.

3 IP Addresses and Subnets

An IP address (or Internet address) is a 32-bit number used to identify a host in an IP network. It is usually expressed as four numbers separated by periods such as A.B.C.D where A, B, C, and D are decimal numbers between 0 and 255. For example, 192.168.0.10 is an IP address. Within an isolated private network, the user can arbitrarily assign IP addresses to the hosts in the network as long as each address is unique. However, when a private network is connected to the Internet (a public network), the user must use registered IP addresses for all external communications in order to avoid address duplication.

An IP address has two parts: a network portion (or prefix) that identifies a particular network, and a local portion that identifies a particular host within the network. The division between the network portion and the local portion is arbitrary (depending on network size and assignment). All the hosts within a network must be assigned the IP addresses with the same network portion. A network can be further divided into several subnetworks (or subnets) for reasons such as simplifying administration, separating physical networks, or controlling network traffic. For example, a network with 150 hosts can be divided into three subnets, and each subnet, connected by a separate Ethernet network, has 50 hosts. One obvious reason for dividing the network is that an Ethernet network with 50 hosts will be much less congested than the one with 150 hosts. With subnets, the local portion of an IP address can be viewed as having two parts: a subnet portion that identifies a particular subnet within a network, and a host portion that identifies a particular host within the subnet. Again, the division between the subnet portion and the host portion is arbitrary (depending on subnet size and assignment). All the hosts within a subnet must be assigned the IP addresses with the same network portion and the same subnet portion. For example, a network with 16-bit prefix address 128.10.0.0 can have two subnets such as 128.10.1.0 and 128.10.2.0, where the subnet portion is 8 bits wide. Within the subnet 128.10.1.0, a host can have an IP address such as 128.10.1.1 or 128.10.1.2. A subnet is identified by its subnet address (the combination of network portion and subnet portion) and subnet mask. A subnet mask is a 32-bit binary number used to mask an IP address (by performing a bitwise AND operation) to generate a subnet address. Usually, a subnet mask is expressed in the form of an IP address. For example, a subnet mask 255.255.255.0 and host IP address 128.10.1.1 specify the subnet address 128.10.1.0 (with 24 effective bits).

Subnets are interconnected by gateways (routers). To send an IP packet from one host to another host within the same subnet, the packet is delivered directly (by the physical network such as an Ethernet) to the destination host without going through the subnet gateway. However, to send an IP packet from one host in one subnet to another host in a different subnet or network, the packet must first be delivered to the subnet gateway. The gateway then forwards the packet to the destination host or another gateway that continues to route the packet toward the destination host. How does a host know an IP packet should be sent to its subnet gateway instead of to another host within the same subnet? The answer is simple with the help of a subnet mask. When an IP packet is ready to be transmitted, the host performs a bitwise AND operation using its subnet mask and the destination IP address of the packet. If the result matches its subnet address (the bitwise AND of its subnet mask and its IP address), the destination host is within the same subnet and the packet should be sent directly to the destination host. If the result does not match its subnet address, the destination host resides outside of its subnet and the packet should be sent to its subnet gateway. Subnets are invisible outside their networks; they are only known within their networks.

IP addresses can be divided into the following three types based on addressing scope:

- **Unicast:** This is the most common type. A unicast IP address is assigned to an individual host (including gateways and routers). When the destination IP address of an IP packet is a unicast address, the packet is only delivered to the host that owns the destination IP address.
- **Multicast:** A multicast IP address is associated with a group of interested hosts. The IP addresses from 224.0.0.0 to 239.255.255.255 are designated as multicast addresses. When a host sends an IP packet to a multicast address, it only sends out one copy into the network. The network will make copies and send one copy to each interested host. To express its interest in a particular multicast address, a host must join that multicast address (by sending an IGMP Join message to the network). Multicast makes it easy for a host to send the same packets to many interested hosts.
- **Broadcast:** A broadcast IP address allows a host to send IP packets to all hosts in a network or subnet. There are two types of broadcast: directed broadcast and limited broadcast. In a directed broadcast to all hosts in a network, the broadcast address is an IP address with the local portion equal to all 1's. Similarly, in a directed broadcast to all hosts in a subnet, the broadcast address is an IP address with the host portion equal to all 1's. The special IP address 255.255.255.255 is used in the limited broadcast to all hosts in a subnet (or a network without subnet). When a host does not know its IP address and subnet address at startup, it uses limited broadcast to request and learn its network parameters.

There are some IP addresses reserved for special purposes. The IP address 127.0.0.1 is designated as a loopback address (activity on this address loops back to itself). When a host sends an IP packet to 127.0.0.1, the packet will be looped back by the IP stack without being sent out. This loopback address is mainly used for testing only. The IP addresses in the following three ranges are designated as private addresses. They can only be used in private networks and are not routable.

10.0.0.0	to	10.255.255.255
172.16.0.0	to	172.31.255.255
192.168.0.0	to	192.168.255.255

The IP address of a host can be statically configured or dynamically assigned at startup (by a DHCP server). Using the statically configured IP addresses is a simpler way to set up a small network.

4 Internet Protocol Suite

4.1 Protocol Layers

Complex communication systems (like the Internet) do not use a single protocol to handle all data communication tasks. Instead, they require a set of cooperative protocols, collectively called a protocol suite. The protocols of a protocol suite usually form a layered structure. When a protocol suite is implemented in software, it is called a protocol stack. The International Organization for Standardization defines a standard Reference Model of Open System Interconnection (OSI), referred to as the OSI model, for communication protocol design. The OSI model contains seven vertical layers, as shown in Fig. 4-1, from layer 1 (the physical layer) up to layer 7 (the application layer).

The Internet protocol suite is a set of communication protocols that the Internet uses. It is also called the TCP/IP protocol suite. This protocol suite has only four layers (RFC 1122) as shown in Fig. 4-2. It does not strictly follow the OSI model because it was developed before the OSI Model. It has no session layer and presentation layer and its data link layer also contains the physical layer. Often people use the term "layer-2 device" to refer to a network device (such as a switch) that forwards Ethernet packets based on Ethernet addresses, and the term "layer-3 device" to refer to a network device (such as a router) that routes IP packets based on IP addresses.

7	application layer
6	presentation layer
5	session layer
4	transport layer
3	network layer
2	data link layer
1	physical layer

Figure 4-1 OSI model

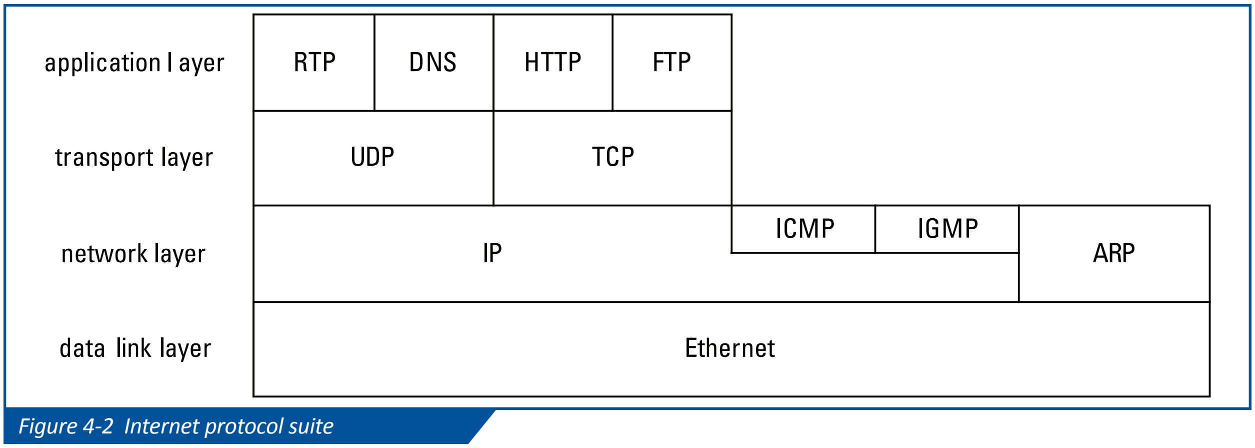


Figure 4-2 Internet protocol suite

In a layer protocol suite like TCP/IP, the entire data packet from an upper layer protocol is encapsulated inside one or more packets of a lower layer protocol. Fig. 4-3 illustrates the concept of TCP/IP encapsulation, where the application data is encapsulated inside the UDP packet as the UDP data. The UDP packet is then encapsulated inside the IP packet as the IP data. Finally, the IP packet is encapsulated inside the Ethernet packet as the Ethernet data.

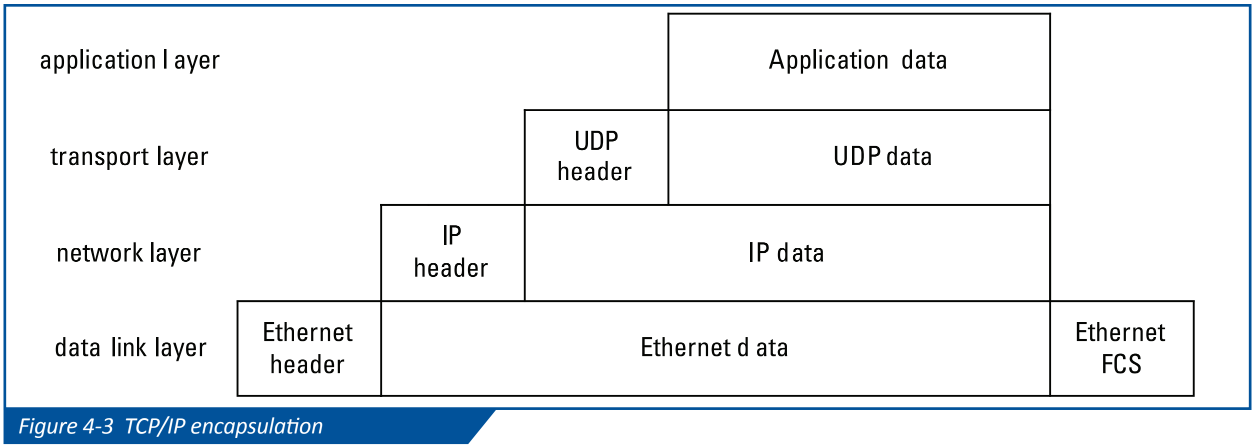
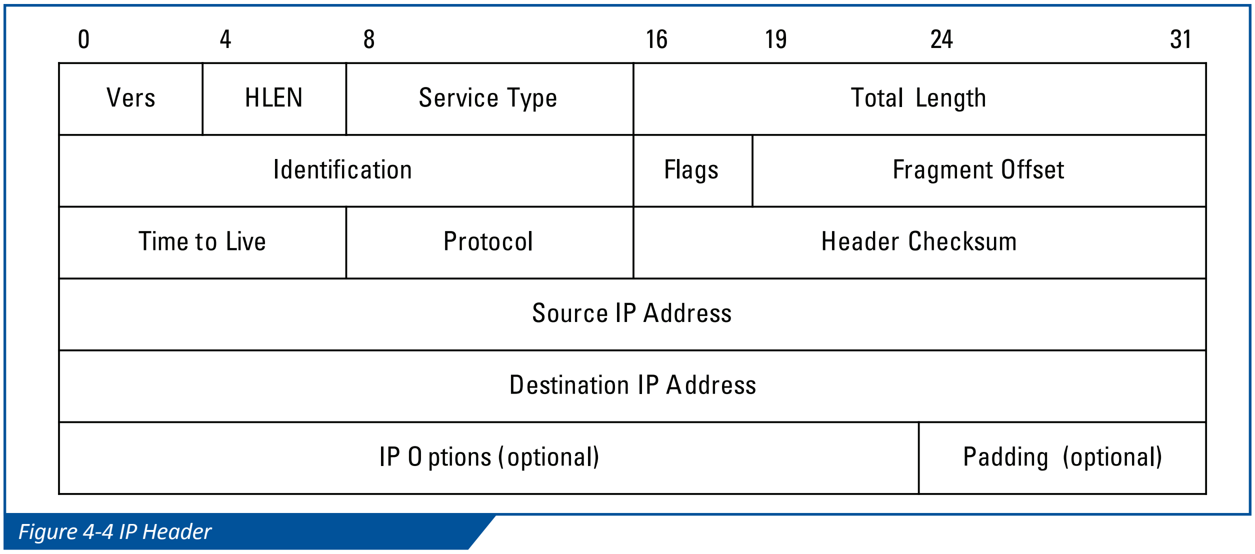


Figure 4-3 TCP/IP encapsulation

In this session, only the following six protocols in the network layer and transport layer are described: IP, ARP, UDP, TCP, ICMP, and IGMP.

4.2 IP

The Internet Protocol (IP) is the most important protocol in the TCP/IP protocol suite. It is a network layer protocol that specifies the rules and provides the necessary information for routing data in a packet-switched network. The basic data unit transferred by IP is called an IP datagram or simply a datagram. When sending a datagram to a destination, IP does not provide feedback; therefore, there is no guarantee that the datagram will reach the destination without error. To address the reliability issue, an upper layer protocol such as TCP must be used. A datagram is divided into header and data portions. The data portion holds an entire packet of an upper layer protocol such as UDP or TCP. The header portion, called IP header as shown in Fig. 4-4, carries the following information:



Vers	HLEN	Service Type	Total Length	
Identification			Flags	Fragment Offset
Time to Live		Protocol	Header Checksum	
Source IP Address				
Destination IP Address				
IP Options (optional)				Padding (optional)

Figure 4-4 IP Header

- **Version:** This identifies the version of IP protocol used.
- **Addressing:** Each IP header contains a 32-bit source address that identifies the sending host, and a 32-bit destination address that identifies the receiving host(s). The source address must be a unicast address, while the destination address can be a unicast, multicast, or broadcast address. These addresses are used by routers to select a path through the network for routing the datagram.

- **Fragmentation:** An IP datagram can be split into smaller datagrams. This permits a large datagram to be sent across a physical network that can only transfer data in smaller chunks. For example, an Ethernet packet can only carry a maximum of 1500 bytes of data. Therefore, any IP datagram with more than 1500 bytes must be fragmented in order to be sent via Ethernet. IP handles the fragmentation of a datagram at the sending host and the reassembly of the datagram at the receiving host.

- **Type of service:** This indicates the importance or priority of the datagram for supporting QoS (traffic prioritization). It was redefined as Differentiated Services Codepoint (DSCP) in the late 1990s to support the Differentiated Services.

- **Upper layer protocol:** This indicates the upper level protocol whose packet is encapsulated in the data portion.

- **Time to live**: This specifies the maximum hops that the datagram is allowed to remain in the network. The value of Time to Live (TTL) is decremented by one every time a router receives the packet. If TTL reaches zero, the packet is discarded to prevent it from staying in the network forever (in case there is a routing loop).

- **Length:** Two pieces of length information are included in the header: header length and total length of the datagram.

- Header checksum: This is a checksum covering the header portion only. IP does not provide checksum for the data portion.

The minimum length of IP header is 20 bytes (when there is no IP option and no padding).

4.3 ARP

The Address Resolution Protocol (ARP) is a network layer protocol used by one host to find the hardware address (such as Ethernet address) of another host whose IP address is known. There are two types of ARP messages: request and reply. When a host is ready to transmit a unicast IP packet to a host (including the gateway) with an unknown hardware address, it simply broadcasts an ARP request message. The ARP request message contains the sender’s hardware address and IP address, and the target IP address. All hosts on a physical network will receive the broadcast ARP request message, but only the host whose IP address matches the target IP address replies. An ARP reply message is a unicast message filled with the IP address and hardware address of the replying host. When the requesting host receives the ARP reply message, it knows the physical address to use to send the pending IP packet to the replying host. The requesting host also saves both the IP address and physical address of the replying host into a table, called an ARP cache, for use in future transmission. The next time the requesting host has a unicast IP packet to send to the same replying host, it can just simply look up the ARP cache to get the physical address of the replying host. A host does not keep the binding between an IP address and a physical address in the ARP cache forever. Typically, an IP to physical address binding expires in the order of 10 minutes. The reason is to allow the host of a particular IP address to change its physical address (in the case when a backup host takes over a failed host).

A host can broadcast an ARP request message to all other hosts (in the same physical network) by filling the target IP address in the message with its own IP address and the target physical address with its own physical address. Such an ARP request, called a gratuitous ARP request, is not intended to solicit a reply. When all other hosts receive the gratuitous ARP request message, they do not reply, but simply update their ARP caches. When a host changes its physical address (in a failover situation), it can use a gratuitous ARP request to correct the binding of its IP address and its old physical address in the ARP caches of other hosts (instead of letting the binding expire, which would take much longer).

When a host is ready to send a multicast or broadcast IP packet, it does not need to use ARP to resolve the target physical address. There is a direct translation from a multicast IP address to a multicast physical address, and from a broadcast IP address to a broadcast physical address.

4.4 UDP

The User Datagram Protocol (UDP) is a transport layer protocol that provides a simple and efficient delivery service. It is stateless and connectionless. It supports IP multicasting and broadcasting. Because UDP does not provide an acknowledgment for the data it delivers, its delivery service is not guaranteed, and as a result, it is considered unreliable. If reliable delivery is required, the application layer protocols (on top of UDP) must implement acknowledgment and retransmission to guarantee data delivery. For real time applications such as VoIP, UDP is a good choice because there is no time for acknowledgment and retransmission.

A UDP packet has two parts: a header portion and a data portion. The data portion holds the data from an application layer program. The header portion, called UDP header as shown in Fig. 4-5, has a fixed length of 8 bytes. The following describes each of its fields:

0	16 31
UDP Source Port	UDP Destination Port
UDP Message Length	UDP Checksum

Figure 4-5 UDP Header

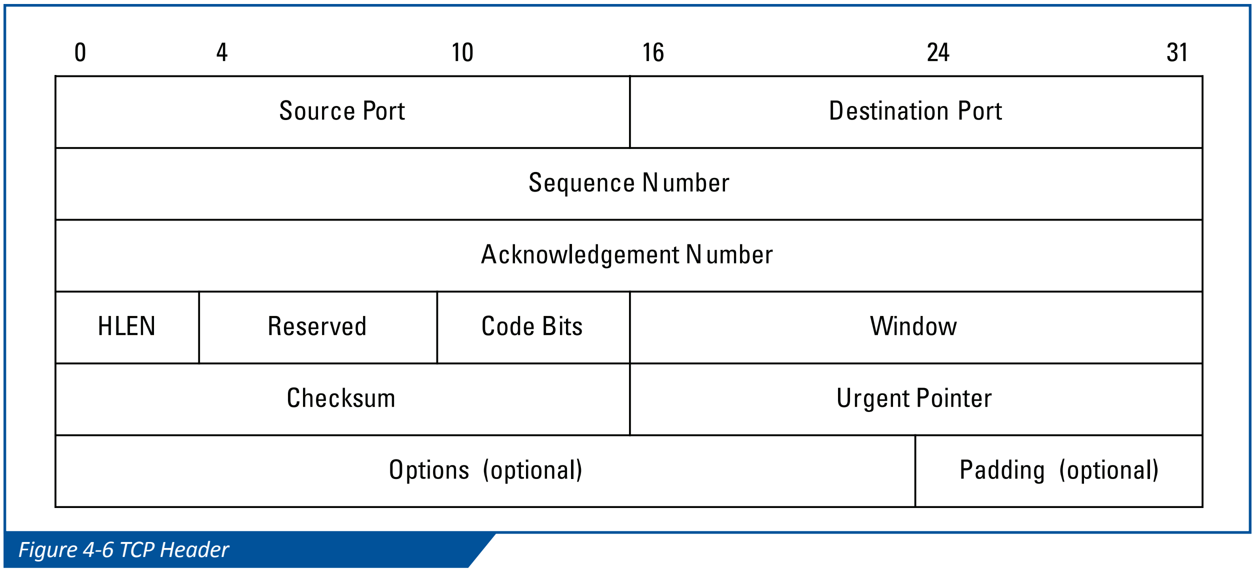
- **Source port:** Identifies the application layer program that originates the data.
- **Destination port:** Identifies the application layer program to which this data is being sent.
- **Length:** The length of data portion in bytes.
- **Checksum:** Optional field to help detect error in the received packet (header and data).

UDP ports are used to identify UDP application programs. They allow UDP to be able to multiplex and demultiplex multiple application programs that use UDP to transport their data. A UDP port is a 16-bit number, ranging from 0 to 65,535. Ports 0 through 1023 contain many "well known" ports that should not be used without IANA (Internet Assigned Numbers Authority) registration.

4.5 TCP

The Transmission Control Protocol (TCP) is a connection-based transport layer protocol that provides a reliable stream delivery service with acknowledgement and retransmission. Each TCP connection is identified by its two endpoints, where an endpoint is defined as (IP address, TCP port). Before data can be transported by TCP, a TCP connection must be established (via three-way handshake). After a TCP connection is established, data and acknowledgement can travel in both directions. To make the transfer of large amounts of data efficient, TCP uses a sliding window algorithm for sending data and acknowledging the received data. To avoid aggregate congestion, TCP controls the rate of sending data into the network. TCP provides guaranteed and ordered data delivery, retransmission of lost data, discarding of duplicate data, and congestion control. It is used extensively by many of the most popular Internet application protocols and programs, including World Wide Web, E-mail, and File Transfer Protocol. However, because TCP is optimized for guaranteed delivery with congestion control rather than timely delivery, it is not particularly suitable for real time applications such as VoIP.

The unit of information transferred between two TCP endpoints is called a segment. A TCP segment has two parts: a header portion and a data portion. The data portion holds the data from an application layer program. The header portion, called TCP header, is shown in Fig. 4-6. In a TCP header, the source port and destination port are used to identify the source and destination application programs respectively. The sequence number and the acknowledgement number are used for acknowledgement and retransmission. The 6-bit "code bits" is used to indicate the function of the segment. Similar to UDP checksum, TCP checksum is used to help detect error in the received segment (header and data). But unlike UDP checksum, TCP checksum is never optional. The sender must generate it and the receiver must check it.



<table>
<tr><th>0</th><th>4</th><th>10</th><th>16</th><th>24</th><th>31</th></tr>
<tr><td colspan="3">Source Port</td><td colspan="3">Destination Port</td></tr>
<tr><td colspan="6">Sequence Number</td></tr>
<tr><td colspan="6">Acknowledgement Number</td></tr>
<tr><td>HLEN</td><td>Reserved</td><td>Code Bits</td><td colspan="3">Window</td></tr>
<tr><td colspan="3">Checksum</td><td colspan="3">Urgent Pointer</td></tr>
<tr><td colspan="4">Options (optional)</td><td colspan="2">Padding (optional)</td></tr>
</table>

Figure 4-6 TCP Header

TCP ports allow TCP to be able to multiplex and demultiplex multiple application programs that use TCP to transport their data. A TCP port is a 16-bit number, ranging from 0 to 65,535. Ports 0 through 1023 contain many "well known" ports that should not be used without IANA registration.

4.6 ICMP

The Internet Control Message Protocol (ICMP) is a network layer protocol used to report errors, send control messages, and test network connections. Typically, ICMP messages are generated in response to errors in IP datagrams or for diagnostic or routing purposes. There are more than two dozen defined ICMP messages. Each message has its own format. When sending an ICMP message, the message is encapsulated directly within a single IP datagram. The following list shows the most common ICMP messages:

- Echo reply
- Destination unreachable
- Source quench
- Redirect
- Echo request
- Router advertisement
- Router solicitation
- Time exceeded
- Parameter problem
- Timestamp request
- Timestamp reply
- Address mask request
- Address mask reply
- Traceroute

Many commonly used network utilities are based on ICMP messages. For example, the popular PING utility command uses the ICMP "Echo request" and "Echo reply" message.

4.7 IGMP

The Internet Group Management Protocol (IGMP) is a network layer protocol for managing the membership of multicast groups. It is used by IP hosts to report their multicast group memberships to neighboring multicast routers. It is also used by multicast routers to query multicast group memberships of neighboring IP hosts. There are two versions of IGMP that are widely used: version 2 (or IGMPv2) and version 3 (or IGMPv3). IGMPv2 added the Leave message to allow group membership termination to be quickly reported to multicast routers. IGMPv3 is the latest version. It added the support for source filtering. Source filtering is the ability to specify interest in receiving packets only from some specific source addresses (inclusive), or from all but some specific source addresses (exclusive). IGMPv3 is backward compatible with IGMPv2. An IP host that implements IGMPv3 should also support multicast routers that implement only IGMPv2. A multicast router that implements IGMPv3 should also support IP hosts that implement only IGMPv2.

Before a host can receive multicast packets from a particular IP multicast group (multicast address), it must first join the multicast group. To join a multicast group, a host simply sends an IGMP Membership Report message to the multicast group address to join if IGMPv2 or to 224.0.0.22 if IGMPv3. When a host is no longer interested in receiving multicast packets from a joined multicast group, it sends an IGMP Leave message to 224.0.0.2 if IGMPv2 or an IGMP Membership Report message to 224.0.0.22 if IGMPv3. Multicast routers periodically send IGMP Membership Query messages to neighboring hosts to query their multicast group membership states. When IP hosts receive an IGMP Membership Query message, they should respond by sending back an IGMP Membership Report message if they have joined the queried multicast group.

5 Call Control

A VoIP application requires two types of protocols: call control (including call signaling) and media transport. Call control protocols are used to set up, tear down, maintain, and monitor call connections. Media transport protocols are used to transport media data. Examples of VoIP call control protocols are H.323, MGCP (Media Gateway Control Protocol), and SIP (Session Initiation Protocol). Because SIP is a simple and very popular VoIP call control protocol, it is the only call control protocol described in this session. However, SIP itself does not describe multimedia sessions and parameters. SDP (Session Description Protocol) is commonly used with SIP to fulfill these functions. Therefore, SDP is also described in this session.

5.1 SIP

SIP is an application layer control protocol for creating, modifying, and terminating multimedia sessions, include VoIP calls, multimedia distribution, and multimedia conferences. The protocol was developed by the Internet Engineering Task Force (IETF) and originally specified in RFC 2543, which was later replaced by RFC 3261 in 2002. SIP is a text-based protocol like HTTP (Hypertext Transfer Protocol). It follows the successful HTTP request/response transaction model. Because SIP is simple, open, flexible, and extensible, it has gained tremendous support from technology developers and service providers. For example, SIP was adopted by 3GPP (Third Generation Partnership Project) as the signaling protocol for the 3G cellular networks, and also adopted by APCO Project 25 (P25) as the signaling protocol for the ISSI (Inter RF Subsystem Interface). The strength of SIP lies in its simplicity, scalability, extensibility, modularity, and mobility.

5.1.1 SIP Entities

There are four types of logical SIP entities in a SIP network. The following describes each of them:

1. User Agent (UA): A UA is an endpoint entity in a SIP network. UAs are the session initiators and session terminators. There are two types of UAs:

- User Agent Client (UAC): a client application that initiates SIP requests
- User Agent Server (UAS): a server application that generates the responses to the received SIP requests

2. Proxy Server: A Proxy Server is an intermediary entity in a SIP network. It can act as both a server and a client. It is used to make requests on behalf of real clients.

3. Redirect Server: A Redirect Server is a server that when receiving a request from a client, redirects the client to contact a different server or set of servers. Unlike Proxy Servers, Redirect Servers do not pass the requests on to other servers.

4. Registrar: A Registrar is a server that accepts REGISTER requests from the UAs for the purpose of authentication and maintaining a location database.

More than one SIP entity can reside in the same SIP device. For example, a SIP phone normally contains both UAC and UAS, and a SIP server could contain both Proxy Server and Registrar.

5.1.2 SIP Operations

SIP is modeled after HTTP, the basis of the Internet World Wide Web. It is a client-server-based protocol, where the client issues requests and the server returns responses. There are two types of SIP messages: requests and responses. Requests are the messages sent from the client to the server, and responses are the messages sent from the server to the client. The following defines the five basic request methods:

1. INVITE: Used to initiate a call (or session).

2. REGISTER: Used to add, remove, or query the records of location services. The location records are added or removed by the users when they arrive or leave a certain domain.

3. ACK: Used to express a positive confirmation.

4. BYE: Used to terminate an existing call (or session).

5. CANCEL: Used to terminate a pending request.

SIP response messages contain numeric response codes. There are two types of responses and six classes. The following lists response types and classes:

Response Types:

- Provisional (1xx class): Provisional responses are used by UAS to indicate progress, but they do not terminate SIP transactions.
- Final (2xx, 3xx, 4xx, 5xx, 6xx classes): Final responses terminate SIP transactions.

Response Classes:

- 1xx: provisional
- 2xx: successful
- 3xx: redirection
- 4xx: request failure
- 5xx: server failure
- 6xx: global failure

The following lists some common response codes and their meanings:

- 100 Trying: this response indicates that the request has been received and it is being processed.
- 180 Ringing: this response indicates that the UA is attempting to alert the user for an incoming call.
- 200 OK: this response indicates that the request was successful.
- 401 Unauthorized: this response indicates that user authentication is required.
- 404 Not Found: this response indicates that the target user does not exist in the specified domain.
- 603 Decline: this response indicates that the user explicitly does not wish to or cannot participate.

5.1.3 SIP Addresses

In SIP, a communication resource is identified by its Uniform Resource Identifier (URI). The SIP URI addressing format is similar to the existing e-mail addresses such as user@domain. The user part of the address can be a person's username, full name, alias, or his/her phone number. The domain part of the address represents the person's ISP, employer, or organization. It can be a qualified domain name or a numeric IP address. The following lists several valid SIP URIs:

alice@zetron.com
alice@10.0.0.100
1234567@zetron.com
+1-425-123-4567@10.0.0.100

5.1.4 SIP Message Format

SIP is a text-based protocol. A SIP message is composed of the following three parts:

- **Start Line:** Every SIP message begins with a Start Line. The Start Line contains the message type (method for requests and response code for responses) and the protocol version. It may be either a Request-line for requests or a Status-line for responses.
- **Headers:** SIP header fields are used to convey message attributes and modify message meaning. They have the following format:

  <name>:<value>

  Headers can span multiple lines. Some headers can appear multiple times in a message or can take multiple comma-separated values in a single header occurrence. Some common headers are Via, From, To, Contact, Call-ID, CSeq, Subject, Record-Route, Content-Type, and Content-Length.
- **Body (Content):** A message body is used to describe the session to be initiated, or alternatively it may be used to contain opaque textual or binary data of any type that relates to the session. Message bodies are optional and can appear in both request and response messages. Possible body types include SDP, MIME, and others to be defined by the IETF or some specific implementations.

The following provides an INVITE message example. In this example, SIP user Alice at Zetron invites SIP user Bob at Microsoft to a SIP call for the purpose of discussing lunch.

```
INVITE sip: bob@microsoft.com SIP/2.0
Via: SIP/2.0/UDP client.zetron.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Bob <sip:bob@microsoft.com>
From: Alice <sip:alice@zetron.com>;tag=1928301774
Call-ID: a84b4c76e66710@client.zetron.com
CSeq: 314159 INVITE
Subject: Lunch today.
Contact: <sip:alice@client.zetron.com>
Content-Type: application/sdp
Content-Length: 142
```

(Body: Alice's SDP is not shown)

5.1.5 SIP Call Flow Example

Fig. 5-1 shows a complete call setup and teardown SIP message flow within a single SIP domain. Both SIP User A and B use the same SIP server to relay the SIP messages. In this example, SIP User A initiates a SIP call to SIP User B via the SIP Server. After some media exchange, SIP User A terminates the call.

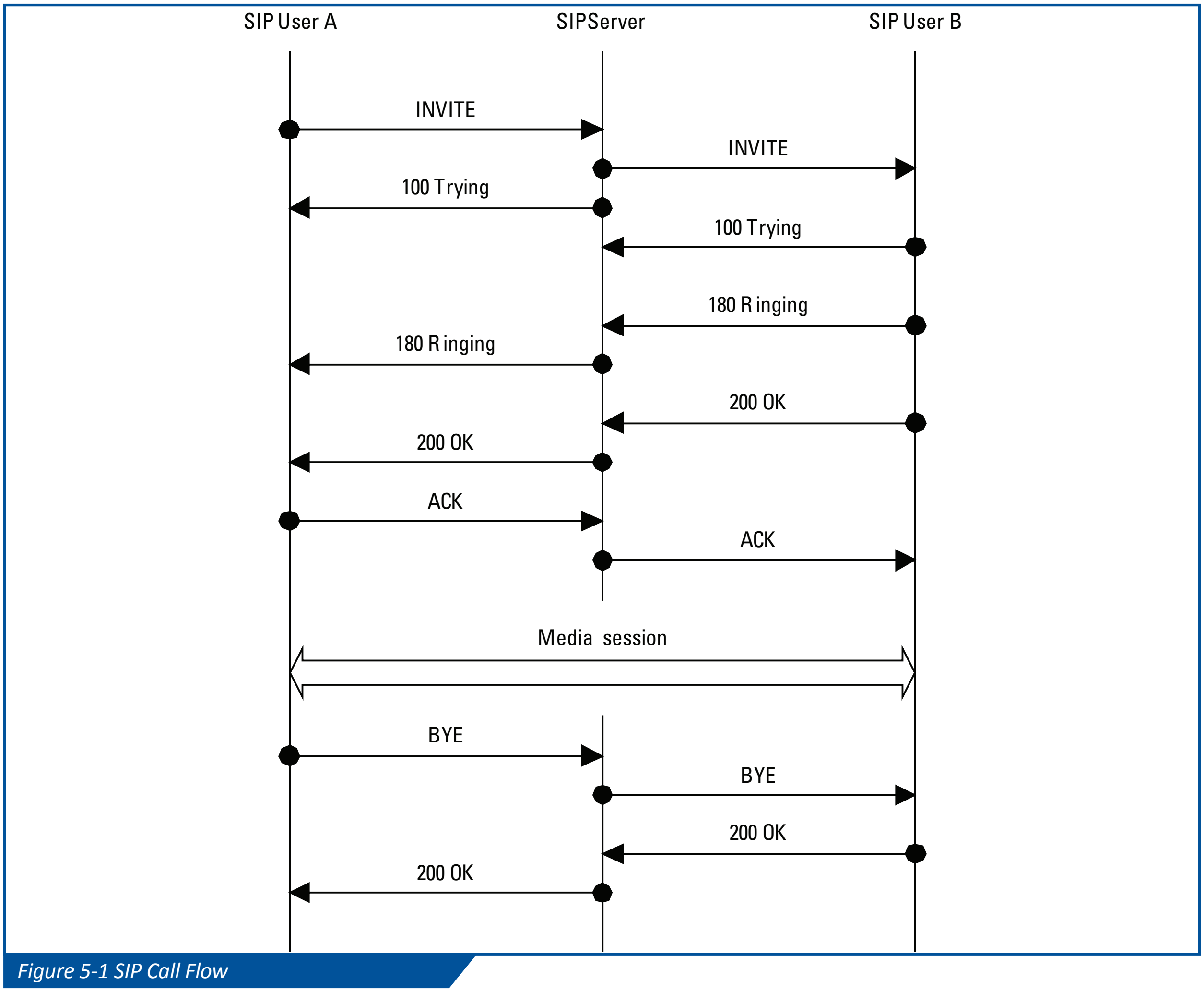


Figure 5-1 SIP Call Flow

5.2 SDP

SDP (Session Description Protocol) is used to describe multimedia session announcement and multimedia session invitation. A multimedia session is defined as a set of media streams that exist for a duration of time. When used with SIP, SDP messages are encapsulated in the body portion of SIP messages. SDP messages usually include session description, media description, and time description. An SDP description consists of a number of text lines in the following form:

<type>=<value>

where <type> must be exactly one case-significant character and

<value> is structured text whose format depends on <type>.

The following lists all the defined types under each SDP description:

- Session description:

v= (protocol version)

o= (originator and session identifier)

s= (session name)

i= (session information)

u= (URI of description)

e= (email address)

p= (phone number)

c= (connection information)

b= (zero or more bandwidth information lines)

One or more time descriptions (“t=” and “r=” lines; see below)

z= (time zone adjustments)

k= (encryption key)

Zero or more media descriptions

- Time description:

  t= (time the session is active)

  r= (zero or more repeat times)

  r= (zero or more repeat times)

- Media description:

  m= (media name and transport address)

  i= (media title)

  c= (connection information)

  b= (zero or more bandwidth information lines)

  k= (encryption key)

  a= (zero or more media attribute lines)

An SDP message example is shown below. It advertises the capability of supporting three audio codec formats: pcmu (G.711 mu-law), pcma (G.711 A-law), and G.729, with RTP as the transport protocol using port 49170. It also specifies (via “a=sendrecv”) that its application will transmit and receive audio streams in full-duplex mode.

```
v=0
o=alice 2890844526 2890842807 IN IP4 10.0.0.103
s=call
c=IN IP4 10.0.0.103
t=0 0
m=audio 49170 RTP/AVP 8 0 18
a=rtpmap:8 pcma/8000
a=rtpmap:0 pcmu/8000
a=rtpmap:18 g729/8000
a=sendrecv
```

6 Real-Time Transport Protocol

The Real-Time Transport Protocol (RTP) is the most popular media transport protocol used in VoIP applications. The Real-Time Transport Control Protocol (RTCP) is a sister protocol of RTP. RTP is used to transport real time data, while RTCP is used to provide feedback on the quality of RTP transport service. Both RTP and RTCP are described in this session.

6.1 RTP

RTP is a transport protocol implemented in the application layer. It provides end-to-end network transport functions suitable for transmitting real-time multimedia data, such as audio and video. RTP supports both unicast and multicast delivery services. The protocol was developed by the IETF and originally specified in RFC 1889, which was later replaced by RFC 3550 in 2003. Although it does not address resource reservation and does not guarantee quality of service, it does provide the necessary functionality to support real-time data streaming. Typically RTP is run on top of UDP. However, other protocols may be used to deliver RTP packets.

An RTP packet has two parts: a header portion and a payload portion. The payload portion contains the multimedia data to be transmitted. The header portion, called RTP header, is shown in Fig. 6-1. The following describes some of the important fields in RTP header:

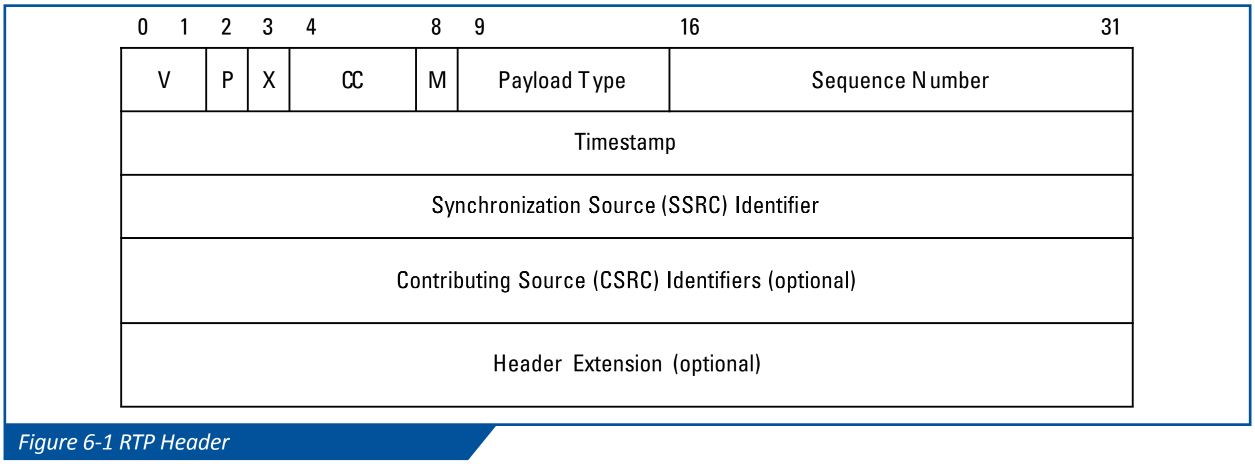

0
1
2
3
4
8
9
16
31
V
P
X
CC
M
Payload Type
Sequence Number
Timestamp
Synchronization Source (SSRC) Identifier
Contributing Source (CSRC) Identifiers (optional)
Header Extension (optional)

Figure 6-1 RTP Header

- **Payload type (7 bits):** Identifies the format of RTP payload and determines its interpretation by the application.
- **Sequence number (16 bits):** Increments by one for each RTP data packet sent, and may be used by the receiver to detect packet loss and to restore packet sequence. The initial value of sequence number should be random (unpredictable) to make known-plaintext attacks on encryption more difficult.
- **Timestamp (32 bits):** Reflects the sampling instant of the first byte in the RTP payload data. It allows the receiver to schedule the playout time for the received RTP payload. The initial value of timestamp should be a random number.
- **Synchronization Source (SSRC) Identifier (32 bits):** Identifies the source that generates the RTP payload. This identifier should be chosen randomly, with the intent that no two sources within the same RTP session will have the same SSRC identifier.

The minimum length of RTP header is 12 bytes (when there is no contributing source (CSRC) identifier and no header extension).

For fixed-rate audio streaming, one RTP packet usually carries one fixed-size audio frame with possible length of 10 – 50ms. The sender must periodically transmit RTP packets at the audio frame rate, that is, 20 – 100 RTP packets per second depending on audio frame length (10 – 50ms).

6.2 RTCP

RTCP is an application layer protocol that provides feedback on the quality of RTP data distribution service. The protocol is defined in RFC 3550. RTCP is a sister protocol of RTP and must be used together with RTP. If UDP is the underlying protocol to transport RTP and RTCP application packets, RTP should use an even port number and the corresponding RTCP should use the next higher (odd) port number. Because RTCP support is not mandatory, many VoIP applications using RTP do not implement RTCP. However, RTCP does provide valuable information to allow QoS to be monitored in real time.

RTCP is based on the periodic transmission of control packets to all participants in an RTP session. The primary function of RTCP is to provide feedback on the quality of RTP data distribution, including packet loss count, packet interarrival jitter estimate, and packet round trip time. Other important RTCP functions include:

- Carry a transport-level identifier for an RTP source, called a canonical name (CNAME), which is used by receivers to synchronize audio and video
- Provide the binding between RTP timestamp and NTP (Network Time Protocol) timestamp

- Control RTCP packet sending rate, based on the number of participants in an RTP session for conserving bandwidth usage
- Convey minimal session control information

There are five RTCP packet types to carry a variety of control information. They are listed below:

- **Sender Report (SR):** for transmission and reception statistics from participants that are active senders
- **Receiver Report (RR):** for reception statistics from participants that are not active senders and in combination with SR for active senders reporting on more than 31 sources
- **Source Description (SDES):** source description items, including CNAME
- **BYE:** indicates end of participation
- **APP:** application-specific functions

The RTCP traffic should be limited to a small and known fraction of the session bandwidth. It is recommended that the fraction of the session bandwidth allocated for RTCP be fixed at 5%. There are algorithms and guidelines provided in RFC 3550 for calculating the desired RTCP packets sending interval.

7 Voice Codecs

In order to transmit voice over an IP network, voice must be digitized and encoded into digital bit streams. A device or algorithm that performs voice signal conversion or compression from one representation into another representation is called a voice codec (or speech codec). There are different types of voice codecs, each of which involves tradeoffs in terms of bit rate (degree of compression), complexity, voice quality, and robustness. In general, voice codecs can be divided into two broad categories: waveform coder and model-based coder (vocoders). In waveform coders, coding and decoding are done on a sample by sample basis with the attempt to preserve the original signal waveform. Compared to vocoders, waveform coders are simpler to implement, with low complexity and low delay. They are popular at rates over 16 kbps (kilo bits/sec). The following lists two commonly used waveform coders that are standardized by the ITU (International Telecommunication Union):

- **G.711:** The international standard for encoding telephone audio into 64 kbps bit streams. There are two variants: mu-law and A-law, where mu-law is used in North America and Japan, and A-law is used in Europe and the rest of the world. Both use a sample rate of 8000 samples per second with 8 bits per sample to produce 64 kbps bit streams. For mu-law, the input is 14-bit linear PCM (Pulse Code Modulation) samples and the output is 8-bit logarithmically compressed PCM samples. While for A-law, the input is 13-bit linear PCM samples and the output is 8-bit logarithmically compressed PCM samples (with a different compression algorithm than the one used by mu-law).
- **G.726:** A standard ADPCM (Adaptive Differential Pulse Code Modulation) voice codec. It compressed G.711 signal (mu-law or A-law) at 64 kbps into one of the following four different rates: 40 kbps, 32 kpbs, 24 kbps, and 16 kbps. The input of G.726 is 8-bit G.711 PCM samples (mu-law or A-law) and the output is 5-bit, 4-bit, 3-bit, or 2-bit ADPCM samples. With G.726, there are four different bit rates to choose from: 40 kbps, 32 kpbs, 24 kbps, and 16 kbps. Among them, the highest bit rate, 40 kbps, provides the best voice quality.

In contrast to waveform coders, vocoders use parametric models to approximate short voice frames (10 to 40 ms long). For each voice frame, a set of model parameters are estimated and converted into a bit stream. The decoder converts the bit stream back into model parameters and then uses these parameters to synthesize a speech signal that is perceptually close to the original. There is no attempt made to recreate the original speech samples. The use of parametric models allows vocoders to operate at low bit rates (under 8 kbps). However, they require accurate speech models to obtain good performance. The following lists several popular vocoders:

- **G.728:** An ITU standardized voice codec operating at 16 kbps with 0.625ms frame length. The technology used for voice compression is LD-CELP (Low Delay - Code Excited Linear Prediction). It has a very low algorithmic delay (0.625 ms).

- **G.729:** An ITU standardized voice codec operating at 8 kbps, 6.4 kbps, or 11.8 kbps with 10ms frame length. The technology used for voice compression is CS-ACELP (Conjugate Structure - Algebraic Code Excited Linear Prediction). G.729 has several variants. The two most common ones are the original G.729 and G.729A (both operate at 8 kbps). G.729A is compatible with G.729, but requires less computation and offers slightly lower voice quality.

- **G.723.1:** An ITU standardized voice codec operating at 5.3 kbps or 6.3 kbps with 30ms frame length. The higher bit rate version is based on MP-MLQ (MultiPulse - Maximum Likelihood Quantization) technology for voice compression, while the lower bit rate version is based on ACELP. As expected, the higher bit rate version provides better voice quality.

- **GSM (Global System for Mobile communications):** GSM is the most popular cellular phone standard in the world. It has used four different types of codecs for voice compression through the years: half rate, full rate, enhanced full rate, and adaptive multiple rate. The half rate codec (operating at 5.6 kbps with 20ms frame length) and the full rate codec (operating at 13 kbps with 20ms frame length) are the two original codecs used. Later in 1997, the enhanced full rate codec (operating at 12.2 kbps) was developed to improve the poor quality of the original full rate codec. In 1998, GSM adopted the Adaptive Multiple Rate (AMR) codec, which is now widely used in GSM. The AMR codec offers a choice of eight different bit rates: 4.75, 5.15, 5.90, 6.70, 7.40, 7.95, 10.2, and 12.2 kbps.

- **iLBC (Internet Low Bit-rate Codec):** A royalty free speech codec developed by Global IP Sound for robust voice communication over IP networks. There are two different bit rates: 13.33 kbps with 30ms frame length, and 15.20 kbps with 20ms frame length. The basic voice quality of iLBC codec is better than G.729A, with high robustness against packet loss. In the case of lost frames, the iLBC codec offers graceful speech quality degradation. It is being used by many PC-based VoIP applications, including Skype, MSN Messenger, Google Talk, and Yahoo! Messenger.

- **IMBE (Improved Multi-Band Excitation):** A proprietary low bit rate speech codec developed by Digital Voice Systems, Inc. (DVSI). The IMBE codec is a variable rate codec (2.4 – 9.6 kbps with 20ms frame length) designed for robustness in both background noise and channel errors. It uses the advanced Multi-Band Excitation (MBE) speech model to produce superior quality speech. The 4.4 kbps version (without forward error correction) of IMBE codec was adopted by the APCO Project 25 (P25) land mobile radio communication system as the standard P25 voice codec.

- **AMBE (Advanced Multi-Band Excitation):** A proprietary low bit rate speech codec developed by DVSI. The AMBE codec is a variable rate codec (2.0 – 9.6 kbps with 20ms frame length) designed to supercede the IMBE codec. The 2.4 kbps version (without forward error correction) of AMBE codec was adopted by M/A-COM for use in its OpenSky land mobile radio communication system.

- **AMBE+2:** A proprietary low bit rate speech codec developed by DVSI. The AMBE+2 codec is a variable rate codec (2.0 – 9.6 kbps with 20ms frame length) that outperforms the AMBE codec. The 4.4 kbps version (without forward error correction) of AMBE+2 codec (referred to as the full-rate version) is fully interoperable with the P25 version of IMBE codec. In additional to the full-rate version, the 2.45 kbps version (without forward error correction) of AMBE+2 codec is referred to as the half-rate version which will be used in the P25 phase 2 system with 6.25 kHz channel bandwidth (instead of 12.5 kHz).

8 Bandwidth Requirement

VoIP calls create two types of network traffic: voice packets and non-voice packets. Voice packets contain digitally encoded voice data, while non-voice packets contain control and status data. Normally non-voice packets constitute only a small percentage of all VoIP traffic ($\leq$ 5%). Therefore, the discussion of bandwidth requirement in this session is only focused on voice packets.

The minimum network bandwidth requirement to support a VoIP call is measured in bps (bits/sec). If the call is full-duplex, bandwidth requirement is applied to both directions at the same time. If the call is half-duplex, bandwidth requirement is only applied to one direction at a time. The formula for calculating the bandwidth required to transport the voice packets of a VoIP call (in one direction only) is very simple, as shown below:

bandwidth requirement (bps) = packet size (bits) X packet frequency (packets/sec)

In the above formula, packet frequency means how many voice packets are sent in one second, which is easy to figure out. If one voice packet carries x ms of voice data, then packet frequency = 1000/x. For example, if one voice packet carries 10 ms of voice data, then packet frequency = 100. To figure out voice packet size, the following formula can be used:

packet size (bits) = voice payload (bits) + protocol header overhead (bits)

where

voice payload (bits) = [voice codec bit rate (bps) X voice frame length per packet (ms)] / 1000

The above formula indicates using a low bit rate voice codec reduces voice packet size, which in turn reduces bandwidth use. However, even though shortening voice frame length per packet reduces voice packet size, it inversely increases packet frequency. Due to protocol header overhead, shortening voice frame length per packet actually increases bandwidth use.

To figure out protocol header overhead (per voice packet), we need to know the specific protocols used to transport voice packets. In VoIP calls, it is very common to use RTP to carry voice payload, and then use UDP to transport RTP packets. If we assume the VoIP protocol stack is RTP/UDP/IP, then protocol header overhead is the sum of RTP header length, UPD header length, and IP header length. The following shows the computation of protocol header overhead (down to the IP layer only) if RTP/UDP/IP are used:

RTP header length (without CSRC and header extension) = 96 bits

UDP header length = 64 bits

IP header length (without IP option) = 160 bits

RTP/UDP/IP protocol header overhead = 96 + 64 + 160 = 320 bits

When the layer-2 protocol is known (could be Ethernet, ATM, Frame Relay, or others), we can include the layer-2 protocol header in the protocol header overhead computation. For example, the following shows the computation of protocol header overhead (down to the layer 2) if RTP/UDP/IP/Ethernet are used:

Ethernet header length (including interframe gap, preamble, and CRC) = 304 bits

RTP/UDP/IP/Ethernet protocol header overhead = 320 + 304 = 624 bits

When bandwidth requirement is determined using the protocol header overhead down to the IP layer (but not including the layer-2 protocol), the bandwidth requirement is referred to as “IP bandwidth requirement”. If a specific layer-2 protocol header, such as the Ethernet header, is included in the computation of protocol header overhead, the resulting bandwidth requirement should be referred to as “Ethernet bandwidth requirement”.

The following table lists the IP bandwidth requirements for several different codecs with a 20ms voice frame length and the RTP/UDP/IP protocol stack.

Voice Codec	*20ms Voice Payload (bits)*	*RTP/UDP/IP Header (bits)*	*20ms Packet size (bits)*	*Packet Frequency (packets/sec)*	*IP Bandwith Requirement*
G.711 64 kbps	1280	320	1600	50	80 kbps
G.726 32 kbps	640	320	960	50	48 kbps
G.728 16 kbps	320	320	640	50	32 kbps
G.729 8 kbps	160	320	480	50	24 kbps
GSM full-rate 13 kbps	260	320	580	50	29 kbps
iLBC 15.20 kbps	304	320	624	50	31.2 kbps
IMBE 4.4 kbps	88	320	408	50	20.4 kbps
AMBE 2.4 kbps	48	320	368	50	18.4 kbps

9 Network Impairments

No network is perfect. The following four types of network impairments discussed in this session have great impacts on call quality: delay, jitter, packet loss, and echo.

9.1 Delay

Network delay is the time for a network to delivery a piece of voice data from the transmit source to the destination that receives the data. It significantly contributes to the overall mouth-to-ear delay. The shorter the network delay, the better the call quality. Network delay can be decomposed into the following four components:

- **Propagation delay:** The time that the physical signal of a packet takes to traverse a physical path. It depends on the physical medium and path length.
- **Transmission delay:** The time that a network device takes to transmit a packet (onto a physical wire). It depends on the packet size and link speed.
- **Nodal processing delay:** The time that a network node (such as switch and router) takes to process a packet, including packet header examination, route determination, and checksum calculation. It depends on the speed of the network device.
- **Queuing delay:** The time that a packet waits in a queue of a network device until it can be processed. It depends on the traffic load along the routing path.

In general, network delay is not constant; it fluctuates from time to time. When network congestion occurs, network delay could become much longer than its nominal value. In addition to network delay, there are other intrinsic delays in the transmitting and receiving devices that contribute to the overall mouth-to-ear delay (or end-to-end delay). In the transmitting device, the dominant delays are codec encoding delay and packetization delay. Codec encoding delay is the sum of codec frame size, look-ahead delay, and processing delay. Packetization delay is the time taken to fill N encoded voice frames into one RTP packet. If one RTP packet carries only one voice frame (N= 1), packetization delay is zero. In the receiving device, the dominant delays are jitter buffer delay (or playout delay) and codec decoding delay.

For full-duplex telephony communications, ITU-T G.114 defined the following three bands of one-way transmission delay (end-to-end delay), independent of other transmission impairments (such as echo):

- **0 – 150ms:** Acceptable for most applications
- **150 – 400ms**: Acceptable for some applications
- **Above 400ms:** Unacceptable for general network planning purposes; however, it is recognized that in some exceptional cases this limit will be exceeded

For half-duplex radio communications, Project 25 defines (in Project 25 Statement of Requirements) the following four throughput delay (mouth-to-ear delay or end-to-end delay) requirements depending on operation modes:

- **Direct mode throughput delay:** less than 250ms in direct radio-to-radio communications
- **Conventional repeater mode throughput delay:** less than 350ms in radio-to-radio communications through a single conventional repeater
- **Single RFSS (RF Subsystem) throughput delay:** less than 500ms in radio-to-radio communications involving a single RFSS
- **Multiple RFSS throughput delay:** less than 1 second in radio-to-radio communications involving two or more RFSS's

9.2 Jitter

Jitter, also known as delay variation, is defined as the variation in the delay of received packets. When network delay varies, it introduces jitter. Large jitter is typically caused by network congestion or route changes.

To mitigate the impact of packet arrival jitter on received voice playout, a jitter buffer (sometimes called de-jitter buffer) is used in the receiving device to temporarily hold the received voice packets for a certain amount of time called jitter buffer delay or playout delay. When playout delay expires, the receiving device starts to play out the received voice packets. With a jitter buffer, a smooth playout can be achieved. However, an extra delay (playout delay) is added into the overall mouth-to-ear delay. When network jitter is large, a large playout delay should be used, which results in longer mouth-to-ear delay.

There are two types of jitter buffers: static and adaptive. A static jitter buffer uses a fixed playout delay which is normally configurable by the user, while an adaptive jitter buffer automatically sets the playout delay according to the measured network jitter in real time. The playout delay of an adaptive jitter buffer is not configurable by the user. When network jitter is static, a static jitter buffer is preferred in order to guarantee the constant delay performance. When network jitter is dynamic, an adaptive jitter buffer is preferred in order to achieve the balance between playout delay and late packet discard. The longer the playout delay, the less often the late arrived packets will be discarded, but the call quality may become worse due to excessive delay.

9.3 Packet Loss

Packet loss can occur in a packet-switched network. It is caused by a variety of reasons, including link failure, traffic congestion that leads to buffer overflow in network devices, random early detection in routers, corrupted Ethernet packets, and packet mis-routing. In addition to network caused reasons, packet loss can occur inside the receiving device. For example, when a voice packet arrives too late for playout, it will be discarded by the receiving device.

When voice packets are lost, there will be gaps in the played out voice stream and voice quality will be degraded. The technique that masks the gaps due to packet loss is called Packet Loss Concealment (PLC). The following lists some possible PLC techniques:

Silence insertion: This technique uses silence frames (all zeroes) to substitute the lost speech frames. It is a very simple technique, but does not work well.

- **Waveform substitution:** This technique uses the previously received speech frames to substitute the lost speech frames. There are different ways of doing it. The simplest way is just to repeat the last received frame for the lost frames. A more sophisticated way is to measure the pitch length and location in the last received few frames, and then repeat the estimated pitch waveform for the lost frames.
- **Model-based regeneration:** This technique uses the speech model parameters from the previously received frames to predict the model parameters in the lost frames. After the model parameters are estimated, synthesized frames are generated to replace to the lost frames.

PLC is effective only for a small numbers of consecutive lost packets that contain a total of up to 40ms of speech.

9.4 Echo

Echo is a major concern in full-duplex VoIP calls. There are two types of echoes: acoustic echo and line echo. The acoustic echo is caused by the acoustic feedback path going from the user's speaker to the same user's microphone. The line echo is caused by 2-to-4-wire hybrids used in the analog part of Public Switched Telephone Network (PSTN). Even though a pure VoIP network does not use hybrids, when it is interconnected into PSTN, the line echo could appear in VoIP to PSTN calls.

The degree of annoyance that echo causes depends both on the amount of delay and on the level difference between the original voice and the received echo signal. For echo to become a problem, it has to be loud enough and delayed enough. The longer the delay, the more annoying the echo becomes. Because of the inherent delay induced by IP networks, echo is much more noticeable in VoIP calls if it is not properly controlled.

The solution for eliminating the echo problem, either acoustic echo or line echo, is to install "echo canceler(s)" in the network or in the end devices. According to ITU-T G.131, it was recommended that echo cancelers be deployed on all telephone connections that use 600 Ω hybrids and exceed one-way echo transmission delay of 25ms. Echo cancelers are limited by the total amount of time they wait for the echo signal to be received, a property known as echo tail length. If echo delay is longer than the echo tail length, echo cancelers will become ineffective.

For half-duplex VoIP calls, echo is not an issue because users do not talk and listen at the same time.

10 Voice Quality Measurement

Voice quality of a VoIP call is determined by several factors, including voice codec, delay, jitter, packet loss, and echo. It can be measured using either subjective or objective methods. Subjective measurement is performed by human listeners. It is an expensive, time consuming, and non-repeatable process. The most commonly used subjective method is the Absolute Category Rating (ACR) test which produces the Mean Opinion Score (MOS). The ACR test is a listening only test carried out in a laboratory under controlled conditions. In the test, a pool of listeners rate the quality of a series of speech recordings using a five-point scale ranging from 1 to 5 as shown below:

5: Excellent
4: Good
3: Fair
2: Poor
1: Bad

After obtaining individual scores, the scores are averaged to produce the MOS score. The ACR test, also known as the MOS test, is specified by ITU-T P.800. The following table lists the MOS scores of several popular voice codecs. Although MOS scores are widely used for comparison, they do vary from test to test, and they are actually relative scores.

Voice Codec	*Bit Rate*	*Mean Opinion Score*
G.711	64 kbps	4.1
G.729	8 kbps	3.92
G.723.1	6.3 kbps	3.9
iLBC	15.2 kbps	3.9
G.726	32 kbps	3.85
G.729A	8 kbps	3.7
G.723.1	5.3 kbps	3.65
G.728	16 bkps	3.61
GSM Full-Rate	13 bkps	3.5

Source:

1. For iLBC codec: Wenyu Jiang and Henning Schulzrinne, "COMPARISONS OF FEC AND CODEC ROBUSTNESS ON VOIP QUALITY AND BANDWIDTH EFFICIENCY", Columbia University, USA

2. For GSM Full-Rate codec: "Mean Opinion Score", wikipedia.org

3. For other codecs: Cisco Tech Note document ID: 14069

Objective measurement is performed by machines using mathematical algorithms. It can be intrusive or non-intrusive. Intrusive methods normally use two input signals: a reference (or original) signal and the degraded signal measured at the output of the network or system under test. They are more accurate than non-intrusive methods, but are not suitable for monitoring live traffic because of the need to use a reference signal and the network. The following briefly describes several popular intrusive objective measurement methods:

- **PSQM (Perceptual Speech Quality Measure):** PSQM is the first international standard for the perceptual quality measurement of telephone-band (300 – 3400Hz) speech signals. It was adopted as ITU-T P.861, but was replaced by ITU-T P.862 (PESQ, see below) in 2001. PSQM uses a psycho-acoustical modeling algorithm to analyze and compare the original and the degraded speech signals. It then produces a PSQM value, which is a measure of signal quality degradation. A PSQM value ranges from 0 (no degradation) to 6.5 (highest degradation), and can be translated into an equivalent MOS score. One of the limitations of PSQM is it is unable to produce appropriate results to account for the network impairments such as delay jitter and packet loss.

- **PSQM+:** This was proposed by the KPN Research of the Netherlands to improve the performance of PSQM. It uses the same perceptual transformation module as PSQM. PSQM+ in general produces results that seem to more accurately reflect the actual performance of voice codecs under realistic network load conditions.

- **PAMS (Perceptual Analysis Measurement System):** This was developed by Antony Rix and Mike Hollier at British Telecommunications (BT) to evaluate the perceived speech quality of telephone networks. It addresses two key network properties, linear filtering and variable delay, that made previous methods unsuitable for end-to-end quality measurement. The PAMS process compares the original and the received signals to determine the MOS prediction, on a scale of 1 to 5, for the listening quality and listening effort.

- **PESQ (Perceptual Evaluation of Speech Quality):** This was developed by the KPN Research of the Netherlands and BT by integrating PSQM+ and PAMS together. It takes into account the following sources of signal degradation: coding distortion, error, packet loss, delay, jitter, and filtering in analog network components. Therefore, it is suitable for end-to-end VoIP voice quality measurement. PESQ was adopted as the ITU-T P.862. It is the latest ITU standard for objective speech quality assessment for narrow-band telephony network and codecs.

Unlike intrusive methods, non-intrusive methods do not need the injection of a reference signal. Therefore, they are suitable for use to monitor live traffic. There are two types of non-intrusive methods: signal-based and parameter-based. The signal-based non-intrusive methods predict speech quality directly from the degraded speech signal (the in-service signal) using signal processing algorithms. Examples of the signal-based, non-intrusive methods are INMD (In-service Non-intrusive Measurement Device) and CCI (Call Clarity Index). The parameter-based, non-intrusive methods predict speech quality directly from network impairment parameters (such as delay, jitter, packet loss, and echo) and non-network parameters (such as codec, language, and expectation factor). They use models and equations to establish the relationship between the perceived speech quality and network/non-network parameters. Normally the cost of parameter-based methods is cheaper than the cost of signal-based methods.

The E-model, specified in ITU-T G.107, is the most widely used parameter-based, non-intrusive method. It was originally designed as a computational model for use in telephone network planning. However, it is now being used to predict the end-to-end conversational quality of VoIP calls non-intrusively. The fundamental principle of E-model is based on a concept that all impairments are “additive” and thus independent of one another. The E-model combines all transmission parameters relevant for the considered connection into a rating factor, called transmission rating factor R. This rating factor R is composed of:

$$R = Ro - Is - Id - Ie + A$$

where

Ro: signal to noise ratio

Is: impairments that occur simultaneously with speech (for example, quantization noise and received speech level)

Id: impairments caused by delay (for example, absolute delay and echo)

Ie: impairments caused by low bit rate codecs and packet loss

A: advantage factor (for example, 0 for wireline and 10 for wide area cellular network)

The range of R is between 0 and 100, where 0 represents an extremely bad quality and 100 represents a very high quality. A particular R value can be mapped to an equivalent MOS score using a formula. To help interpret a calculated R value, ITU-T G.107 provides a guide for the relation between R value and user satisfaction, as shown in the following table:

R values	*MOS scores*	*User Satisfaction*
90 – 100	4.34 – 4.50	Very satisfied
80 – 90	4.03 – 4.34	Satisfied
70 – 80	3.60 – 4.03	Some users dissatisfied
60 – 70	3.10 – 3.60	Many users dissatisfied
50 – 60	2.58 – 3.10	Nearly all users dissatisfied
0 – 50	1.00 – 2.58	Not recommended

For all input parameters used in the E-model calculation, G.107 provides the recommended default values to use. Using the G.107 recommended default values except for Id and Ie, the E-model can be simplified as:

$$R = 93.2 - Id - Ie$$

Assuming there is no echo and no packet loss, the following figure shows the relationship between R value and one-way delay for three different codecs: G.711, G.729A, and G.723.1. This figure clearly demonstrates the effect of one-way delay on call quality. For example, if G.711 codec is used and one-way delay exceeds 250ms, at least some users will be dissatisfied with the call quality.

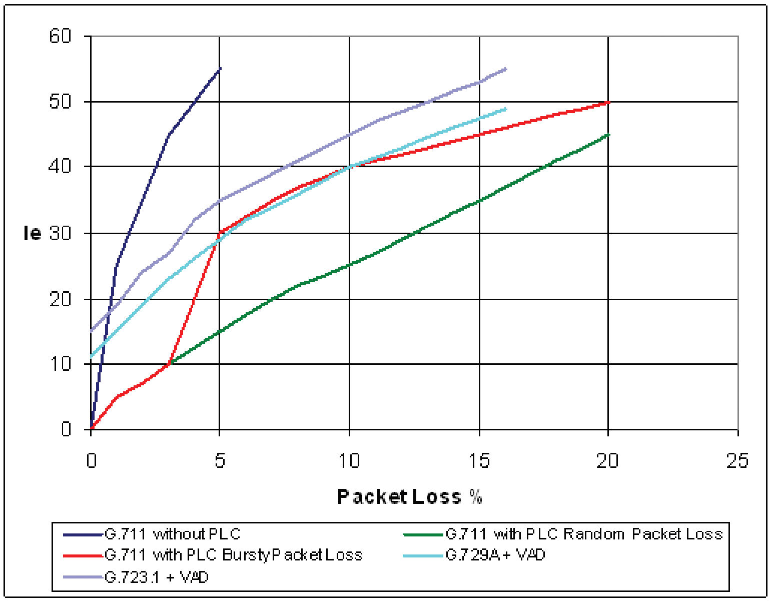


Figure 10-1 Effect of one-way delay on call quality

Assuming G.711 codec with PLC is used and there is no echo, the following figure shows the relationship between R value and one-way delay for nine different packet loss (PL) rates from 0% to 20%. This figure demonstrates the effect of packet loss on call quality when PLC is used.

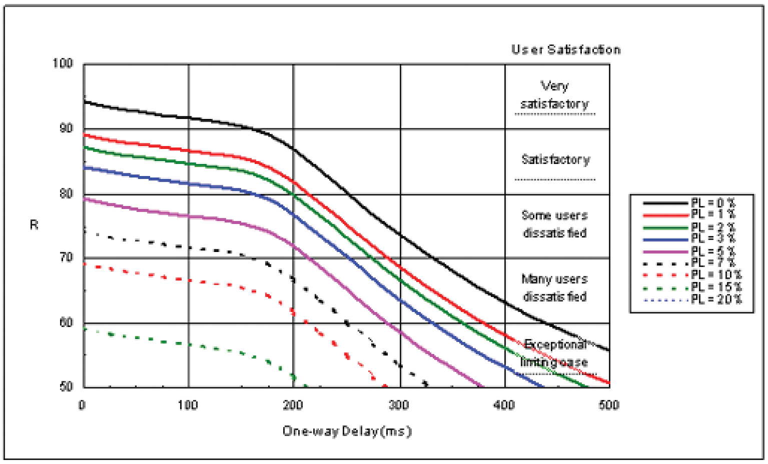


Figure 10-2 Effect of packet loss on call quality

Both PESQ and E-model were designed to evaluate the voice quality of full-duplex telephony conversations. So far, there is no similar measuring method for evaluating the voice quality of half-duplex radio communications.

11 Quality of Service

Quality of service (QoS) refers to the capability of a network to prioritize network traffic in order to provide better service for certain applications. It requires the network being able to identify traffic types and treat them differently. For different applications, the requirements regarding how their traffic is handled by the network could be different. For example, VoIP applications can not tolerate a certain degree of packet loss, and they are also more sensitive to traffic delay and jitter. The following lists four QoS parameters that all applications should be more or less concerned with:

- **Bandwidth:** the rate at which an application's traffic must be carried by the network
- **Latency (or delay):** the time that the network takes to deliver an application's data packets
- **Jitter:** the variation in delay
- **Loss:** the percentage of lost data

If network resources were infinite, then all application traffic could be carried at the required bandwidth, with minimum delay, zero jitter, and zero loss. However, network resources are not infinite. There could be some parts of the network where resources are unable to meet traffic demand. QoS mechanisms work by controlling the allocation of network resources to application traffic to meet the application's service requirements.

QoS involves prioritization of network traffic. The following describes four well-known QoS technologies.

11.1 802.1p

The IEEE 802.1p is a standard that provides traffic prioritization and dynamic multicast filtering at the MAC level (layer 2). It is an extension of the IEEE 802.1q standard that specifies a tag, the VLAN (Virtual LAN) tag, that appends to the header of an Ethernet frame (as the result, the original Ethernet frame is increased by 4 bytes). The VLAN tag has two important fields: VLAN ID (12 bits) and user priority (3 bits). The 802.1q does not define the user priority field. It is defined by the 802.1p standard.

The 802.1p specifies the use of the 3-bit user priority field in the 802.1q VLAN tag to establish eight class levels (0 to 7) for prioritizing layer-2 traffic. How traffic is treated and assigned to any particular class is undefined and left to the user's implementation. However, the IEEE has made broad recommendations. The highest priority is level 7, which might be assigned to network critical traffic for implementing routing protocols. Levels 6 and 5 might be assigned to delay-sensitive applications such as interactive video and voice. Th lowest priority is level 0, which is used as a best-effort default, invoked automatically when no other value has been set.

Typically when the 802.1p enabled hosts or routers send traffic into a LAN, they mark each transmitted packet with an appropriate priority level in the VLAN tag. When the 802.1p enabled layer-2 devices (such as switches) receive an Ethernet packet, they use the marked priority level to handle and forward the packet accordingly. The scope of the 802.1p priority marking is limited to the LAN. Once packets are carried off the LAN, through a layer-3 device (such as a router), the 802.1p priority information is removed and lost.

11.2 Differentiated Services

The Differentiated Services (DiffServ) is a layer-3 Qos technology that defines a field in the IP header, called the DiffServ codepoint (DSCP), for prioritizing layer-3 traffic. Originally the same field in the IP header was called Type of Service (TOS). In the late 1990s, it was redefined as DSCP for use to support DiffServ.

DSCP is a 6-bit field in the IP header. When the DiffServ enabled hosts send traffic into a DiffServ network, they mark each transmitted packet with an appropriate DSCP value. When the DiffServ enabled routers receive an IP packet, they use the marked DSCP value to classify the packet and apply specific queuing or scheduling behavior, known as per-hop behavior (PHB), for routing the packet accordingly.

In theory, there could be up to 64 different classes based on different values in DSCP. However, in practice, most networks use the following four commonly-defined PHBs:

- **Default PHB:** This is assigned to any traffic that does not meet the requirements of any other defined classes. The forwarding of the default PHB traffic is based on best effort delivery. The recommended DSCP for the default PHB is “000000”.
- **Expedited Forwarding (EF) PHB:** This is assigned to the traffic that requires low loss, low delay, and low jitter, such as interactive voice and video. The recommended DSCP for the EF PHB is “101110”.
- **Assured Forwarding (AF) PHB:** This is used to provide assured and reliable services even in times of network congestion. The AF PHB is divided into four classes (1 to 4) with three levels (low drop, medium drop, and high drop) in each class. Between classes, the traffic in the higher class is given higher priority. Within a class, the traffic with low drop precedence is given higher priority over medium or high drop precedence. There are 12 different DSCP values recommended for the AF PHB (because there are four classes and three levels).
- **Class Selector PHBs:** These are defined to maintain backward compatibility with the Precedence field (3-bit) of TOS in the IP header. The Class Selector codepoints are in the form of “xxx000”, where the first three bits represent the TOS Precedence bits. Each TOS Precedence value can be mapped into a DiffServ class.

PHBs are individual behaviors applied at each router. They alone do not guarantee end-to-end QoS. However, by concatenating routers with the same PHBs, it is possible to construct an end-to-end QoS service.

11.3 Integrated Services

The Integrated Services (IntServ) is a service framework that specifies the elements to guarantee network QoS. The idea of IntServ is that every router in the network implements IntServ, and every application that requires a QoS guarantee has to make an individual reservation with the routers. The Resource Reservation Protocol (RSVP) is the primary protocol used for resource reservation. It defines how applications place reservations and how they can release the reserved resources.

The problem with IntServ is that many states must be stored in each router. Therefore, IntServ does not scale up well. It is difficult to keep track of all of the reservations in a large network. As a result, IntServ is not popular today; it has been largely replaced by DiffServ.

11.4 Multi-Protocol Label Switching

Multi-Protocol Label Switching (MPLS) is a hybrid technology that combines the advantages of both circuit switching and packet routing. It enables fast forwarding in the network core and allows conventional routing at the network edges. MPLS lies between layer 2 and layer 3 in the OSI model layer. The main concept of MPLS is to assign a label to each IP packet when it enters an MPLS network. Within the MPLS network, packet routing is based on label lookup, which is faster than IP table lookup. The MPLS labels assigned to each packet contain the following information about how to route the packet: destination, route, VPN membership, and QoS information.

The entry and exit points of an MPLS network are called label edge routers (LERs). When receiving an IP packet, the entry LER maps the IP header into a fixed-length label and then pushes the label onto the IP packet. The complete analysis of IP header is performed only once by the entry LER. After the packet enters the MPLS network, only the label is used by label switch routers (LSRs) to make routing decisions. At the exit LER, the label is popped off from the IP packet, which then might be routed in the conventional fashion thereafter.

MPLS allows network administrators to define routes, called label switched paths (LSPs), from entry LERs to exit LERs, through a series of LSRs, across the MPLS network. These LSPs are pre-assigned and pre-engineered paths which packets with a certain label should follow. MPLS can offer guaranteed QoS without requiring the use of any dedicated lines. For example, an administrator can define an LSP that ensures VoIP traffic is routed through the most reliable and fastest sections of the network, while less critical traffic is routed across the slower sections.

12 Security

Without proper security mechanisms in place, VoIP applications are vulnerable to many different types of attacks. The following lists a few examples of VoIP attacks:

- **Denial of service:** an attack that causes a loss of network connectivity or service
- **Eavesdropping:** the intercepting and reading of messages and conversations by the attacker
- **Man in the middle:** the attacker is able to read, insert, and modify at will the messages between two intended parties
- **Call hijack:** an attack in which one of the intended parties is exchanged with the attacker
- **Spoofing:** the attacker is able to masquerade successfully as another

VoIP traffic can be divided into two categories: call control (including signaling) and media. No matter what VoIP protocols are used, both types of traffic must be secured, and each may require a different security mechanism. In general, security implementation includes authorization, authentication, encryption, and key exchange. From the protocol-layer point of view, security can be implemented at the network layer, transport layer, or application layer.

At the network layer, IPsec (IP Security) is the most popular protocol used to authenticate and encrypt IP packets. It can be used to create VPNs (Virtual Private Networks). It has two modes of operation: transport mode and tunnel mode. In transport mode, only the payload of the IP packet is encrypted and/or authenticated. In tunnel mode, the entire IP packet, including the header, is encrypted and/or authenticated. As a result, the entire encrypted IP packet must be encapsulated into a new IP packet for IP routing to work.

At the transport layer, Transport Layer Security (TLS) is a commonly used protocol for establishing a secure connection between a client and a server. It provides the capability of authenticating both the client and the server and creating an encrypted connection between the two parties. In SIP's RFC (RFC 3261), TLS is the recommended security protocol for use to transport SIP messages. However, TLS can only be used with TCP, not UDP.

At the application layer, each application protocol may employ a different security mechanism. For end-to-end security, SIP uses an addressing scheme called SIPS, which is similar to HTTPS. A SIPS URI specifies that the resource be contacted securely. An example of a SIPS URI is "sips:bob@zetron.com". A call to a SIPS URI guarantees that a secure protocol (TLS) is used to transport all SIP messages from the caller to the callee.

To protect the real time media, RTP defines a secure profile called SRTP (Secure RTP). SRTP provides encryption, message authentication, and replay protection to the RTP data. The primary security goals for SRTP are to ensure the confidentiality of the RTP payload, the integrity of the entire RTP packets, and the protection against replayed packets.

The common requirement for secure communications is encryption. Encryption is a process that transforms data to make it unusable to anyone who does not possess special knowledge called a key. This following lists three well-known encryption algorithms used in VoIP applications:

- **DES:** The Data Encryption Standard (DES) was selected as an official Federal Information Processing Standard (FIPS) for the United States in 1976. It uses a 56-bit key to encrypt data. Because the key length is not long enough, it is now considered insecure for many applications.
- **TDES:** The Triple DES (TDES or TDEA) is an improvement over DES by concatenating 3 DES blocks. It uses a 192-bit key to provide better security, but the computation time is much longer.
- **AES**: The Advanced Encryption Standard (AES) is a new federal encryption standard to replace DES. It uses a key length of 128, 192, or 256 bits. AES provides higher throughput and lower computational complexity than TDES. It has become one of the most popular symmetric-key algorithms used today.

13 Multicast Routing

In IP networks, there are three schemes to deliver (or route) packets: unicast, multicast, and broadcast. In the unicast scheme, a packet is delivered only to a specific single host. In the multicast scheme, a packet is delivered to a group of hosts that have expressed interest in receiving the packet. In the broadcast scheme, a packet is delivered to all hosts in a network. Unicast and broadcast are easier to implement than multicast. With multicast, a host can send only one copy of a packet into the network, and the network will make copies of the packet and deliver one copy to each interested host. For multicast delivery to work, the following three issues must be solved:

- **Group addressing:** There must be special addresses defined to represent multicast group memberships. One address should be used to represent one multicast group.
- **Dynamic registration:** There must be ways defined for hosts to signal interests in multicast groups. Hosts should be able to join or leave a particular multicast group at any given time. The network must keep track of the multicast group membership states of all hosts.
- **Multicast routing:** The network must be able to build multicast distribution trees in order to distribute packets to all interested hosts. Because hosts can join or leave multicast groups at any given time, multicast distribution trees must be maintained and updated dynamically.

To solve the issue of group addressing, the IETF has designated the IP addresses from 224.0.0.0 to 239.255.255.255 as multicast addresses. The entire chunk of multicast addresses is divided into the following four ranges:

224.0.0.0 – 224.0.0.255: reserved for routing protocols

224.0.1.0 – 224.0.1.255: Internet control block

224.0.2.0 – 238.255.255.255: globally scoped addresses

239.0.0.0 – 239.255.255.255: limited scope addresses

The globally scoped addresses are available for general multicast applications. The limited scope addresses are designated for use within private networks. These are not routable outside their intended networks.

To solve the issue of dynamic registration, the IETF has defined the Internet Group Management Protocol (IGMP) for managing the host's memberships of multicast groups. To join a multicast group, a host just simply sends an IGMP Join (Membership Report) message into the network. When a host is no longer interested in a joined multicast group, it just sends an IGMP Leave message into the network. The multicast routers in the network must listen to the IGMP Join and Leave messages sent by the hosts in order to keep track of their multicast group memberships. The multicast routers also periodically send IGMP Membership Query messages to the hosts to query their multicast group membership states.

To solve the issue of multicast routing, the IETF has defined the following multicast routing protocols (used only by multicast routers):

- **Distance Vector Multicast Routing Protocol (DVMRP):** This is used to share information between multicast routers for distributing multicast packets throughout the network. DVMRP is the earliest protocol designed for multicast routing.
- **Multicast Open Shortest Path First (MOSPF):** This is an extension to OSPF to allow it to support IP multicast. MOSPF works only in the networks that are using OSPF. It does not work well in the networks that have many active hosts. Therefore, MOSPF is only used in some specialized applications.
- **Protocol Independent Multicast (PIM):** This is a family of multicast routing protocols, each optimized for a different environment. There are several different PIM protocols. The main two ones are: PIM Sparse Mode (PIM-SM) and PIM Dense Mode (PIM-DM). Because PIM-SM and PIM-DM are two of the most popular multicast routing protocols today, they are described in more detail below.

PIM-SM is the most commonly implemented PIM protocol. Its philosophy is based on the assumption that the members (interested hosts) of any particular multicast group are sparsely distributed throughout the network. Therefore, it assumes most subnets in the network are not interested in receiving any given multicast packet. The goal of this protocol is not to flood unwanted multicast packets to every subnet. In order to receive multicast data from a particular group, PIM-SM routers must explicitly tell their upstream routers about their interest in this particular group. PIM-SM routers use PIM Join and Prune messages to join and leave multicast distribution trees.

A PIM-SM multicast distribution tree for a particular group is rooted at some selected router called the Rendezvous Point (RP). When a multicast router wants to receive multicast data from a particular group (because of receiving an IGMP Join message from one of its neighboring hosts), it registers with the RP by sending a PIM Join message to the RP. After receiving the PIM Join message, the RP adds the multicast router to its multicast distribution tree of a particular group, called the Rendezvous Point Tree (RPT). When a source router wants to send multicast data, it first sends the data to the RP. After receiving the multicast data, the RP forwards the data to all of the registered routers along the RPT of a particular group. Once the multicast data stream begins to flow, the routers in the data-flow path will automatically optimize the path (based on the shortest path) to remove any unnecessary hops, including to the RP. The optimized path results in a change from RTP to Shortest Path Tree (SPT) to make multicast routing more efficient. When a registered router is no longer interested in receiving multicast data from a particular group, it de-registers with the RP by sending a PIM Prune message to the RP.

In contrast to PIM-SM, the philosophy of PIM-DM is based on the assumption that the members of any particular multicast group are densely distributed throughout the network. Therefore, it assumes most subnets in the network would want to receive any given multicast packet. In a PIM-DM network, multicast data is initially sent from the source router to all other routers in the network (similar to broadcast). After receiving the multicast data, routers that do not have any interested hosts will send PIM Prune messages upstream to remove themselves from the distribution tree. Any upstream router that has received PIM Prune messages from all of its downstream interfaces will also send a PIM Prune message further upstream to remove itself from the distribution tree. Eventually the multicast data is only sent to those routers that actually want it. PIM-DM only uses source-based distribution trees. It does not use RPs. Therefore, it is simpler to implement and deploy than PIM-SM. However, it does not scale up well in a large network where most hosts are not interested in any given multicast data.

It is possible to support both PIM-SM and PIM-DM in a network. In this case, some multicast groups will operate in PIM-SM and some in PIM-DM. The choice between PIM-SM and PIM-DM is on a per group basis, rather than on a per router interface basis. It is the responsibility of network administrator to configure the network and router interfaces to determine which protocol is used on which multicast group.

The above description on multicast routing was focused on the IP layer (layer 3) only. At the layer 2, by default, all Ethernet switches forward multicast Ethernet packets to every switch port except the port of entry. However, most of the hosts connected to switch ports may not be interested in any given multicast packet. To minimize the unwanted multicast flooding at layer 2, a feature of Ethernet switch called "IGMP snooping" was introduced. A switch that supports IGMP snooping will listen in on the IGMP messages passed between hosts and multicast routers. When the switch detects an IGMP Membership Report message from a host to join a multicast group, it adds the host's port number and the multicast group to its multicast forwarding table. When the switch detects an IGMP Leave message from a host to leave a multicast group, it removes the host's port number and the multicast group from its multicast forwarding table. When the switch receives a multicast packet from a particular group, it only forwards the packet to the ports whose port numbers and the particular multicast group are in its multicast forwarding table. Therefore, with IGMP snooping, hosts will not receive any unwanted multicast packets from their connected switches.

14 Test Tools

14.1 Wireshark

Wireshark (formerly known as Ethereal) is the most popular, free network protocol analyzer (or packet sniffer). It is an application program that runs on Windows, Unix, Linux, and Mac OS X machines. It captures, decodes, and displays Ethernet packets from a live network to allow the user to see all traffic being passed through a network device such as a switch or hub. The user can interactively browse the captured data, and view the detailed information of each packet. Wireshark understands the structures of several hundred different network protocols. Therefore, it is able to display packet encapsulation and interpret the meaning of every single field. Wireshark is a great tool for network troubleshooting, traffic analysis, and software and protocol development.

Even though Wireshark can capture and display all traffic being passed through a network device, it allows the user to specify capture filters to limit the amount of packets that it captures. It also allows the user to specify display filters to limit the amount of packets that it displays. To specify capture or display filters, the user can enter network address, IP address, protocol, and/or port number. For example, the user can specify a capture filter to capture packets only to/from IP address 192.168.1.10. Wireshark can also save the captured packets into a file for offline analysis.

When the Wireshark host machine is connected into a switch port, the port must be configured as a "monitor port" in order to capture all traffic being passed through the switch. If the switch port connected to the Wireshark host machine is a regular port instead of a monitor port, then Wireshark can only see the traffic that is forwarded to this regular port.

14.2 NIST Net

NIST Net is a popular, free network emulation software package created by the National Institute of Standards and Technology (NIST). It runs on Linux machines only. It is implemented as a kernel module extension to the Linux operating system, so it is very fast. NIST Net allows the user to emulate a wide variety of network conditions on a single machine that has been set up as a router. The emulated network conditions can be delay, jitter, packet loss, packet ordering, packet duplication, and bandwidth limitation. For VoIP applications, NIST Net can be used to test the voice quality of a VoIP system under certain degree of network impairments including delay, jitter, and packet loss. With NIST Net, the user can control each individual network impairment parameter, so testing can be done in repeatable lab settings.

Typically NIST Net is installed on a Linux machine with two network interfaces. The NIST Net machine should be configured as a router where each network interface is connected to a different IP network. The user should place some test host(s) in the network on one side of the NIST Net router, and some other test host(s) in the other network. All packets that flow between two networks will be routed through the NIST Net router which emulates the configured network conditions.

NIST Net provides two types of user interfaces: a simple command line interface, suitable for scripting, and an interactive graphical interface, allowing the user to specify and monitor a large number of emulation rules simultaneously. For each emulation rule, the user can specify the source IP address (in one network), destination IP address (in the other network), protocol (TCP or UDP), and port numbers; and can also specify the network condition to be emulated, including delay, jitter, packet loss, packet duplication, and bandwidth limitation. After emulation rules are configured, the NIST Net will apply these rules to route packets between two networks. Emulation rules are unidirectional, so to emulate bi-directional network conditions, the user must specify the emulation rule for each direction.

A. Appendix - Network Terms Glossary

Default Gateway

A router that serves as an access point (for a network or subnet) to outside networks.

Firewall

A security device that inspects and filters (denies or permits) network traffic passing through it based on a set of rules. Firewalls are used to prevent unauthorized outside users from accessing private networks, and control what outside resources their inside users have access to.

Gateway

The word “gateway” has different meanings depending on where it is used. It can be used for a network gateway, which is a router that connects an inside network or subnet to outside networks such as a WAN or the Internet. It can also be used for a device such as a PSTN or radio gateway that converts one protocol to another or one signal format to another.

Hub

A device with multiple ports for connecting multiple Ethernet devices together to form a single network segment. Hubs work at the physical layer (layer 1) of the OSI model. When a hub receives a signal (an Ethernet packet) from one port, it simply broadcasts the signal out on every other port. It does not inspect the traffic passing through it. All Ethernet devices connected by hubs must operate in half-duplex mode due to potential signal collision. Today, hubs have mostly been replaced by switches and therefore have become obsolete.

LAN (Local Area Network)

A network that covers a small geographical area such as a home, a single building, a group of buildings, or a small campus. Most LANs connect personal computers (PCs) and servers. Because the distance is short, LANs can operate at very high speeds with short delay. A LAN can contain multiple subnets. Today the dominant LAN technology is Ethernet.

Managed Switch

A network switch that allows the user to configure and control each individual port. The configuration options normally include at least the port on/off setting, link speed, duplex mode, virtual LAN, and Spanning Tree Protocol. When connecting an Ethernet device that does not support auto-negotiation or its auto-negotiation capability is disabled, the user should use a managed switch and configure the connected port to match the link speed and duplex mode used by the connected Ethernet device for the best performance. In contrast to a managed switch, an unmanaged switch has no configuration interface or options.

Router

A device with multiple interfaces for interconnecting multiple networks (or subnets). Routers work at the network layer (layer 3) of the OSI model. They route IP packets from one network to another network based on IP addresses.

Subnet (Subnetwork)

A single IP network can be divided into several subnets for reasons such as simplifying administration, separating physical networks (to reduce broadcast traffic), or controlling network traffic. A typical subnet is a physical network served by one router (the subnet gateway or default gateway). When sending an IP packet from one host in one subnet to another host in a different subnet or network, the packet first has to be delivered to the subnet gateway, then the gateway forwards the packet to the destination host or another gateway which continues to route the packet toward the destination host. Subnets are only visible within their networks. They are invisible outside their networks.

Switch

A device with multiple ports for connecting multiple Ethernet devices together. Each port in a switch has its own isolated collision domain. Therefore, an Ethernet device connected to a switch port can operate in full-duplex mode. Generally speaking, switches work at the data link layer (layer 2) of the OSI model. They inspect the source and destination Ethernet addresses of the packets that enter each port, and only forward the packets to the connected devices that the packets were destined for. With intelligent packet forwarding and allowing full-duplex communication, a switch offers much better performance than a hub. Note the definitions of managed and unmanaged switches in this glossary.

WAN (Wide Area Network)

A network that covers a relatively broad geographical area such as a city, a state, or even a country. WANs are used to connect LANs together over long distances. Compared to LANs, WANs operate at slower speeds and have much longer delay. They often use transmission facilities provided by telephone companies. Popular WAN technologies include DSL, SONET, ATM, Frame Relay, and ISDN.

B. Appendix - VoIP White Paper References

[1] IEEE Standard 802.1D, Media Access Control (MAC) Bridges, 2004.

[2] IEEE Standard 802.1Q, Virtual Bridged Local Area Networks, 2005.

[3] IEEE Standard 802.3, Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, 2005.

[4] IETF RFC 768, User Datagram Protocol, 1980.

[5] IETF RFC 791, Internet Protocol, 1981.

[6] IETF RFC 792, Internet Control Message Protocol, 1981.

[7] IETF RFC 793, Transmission Control Protocol, 1981.

[8] IETF RFC 826, Ethernet Address Resolution Protocol, 1982.

[9] IETF RFC 1075, Distance Vector Multicast Routing Protocol, 1988.

[10] IETF RFC 1584, Multicast Extensions to OSPF, 1994.

[11] IETF RFC 1633, Integrated Services in the Internet Architecture: An Overview, 1994.

[12] IETF RFC 2205, Resource ReSerVation Protocol (RSVP), 1997.

[13] IETF RFC 2236, Internet Group Management Protocol, Version 2, 1997.

[14] IETF RFC 2327, SDP: Session Description Protocol, 1998.

[15] IETF RFC 2401, Security Architecture for the Internet Protocol, 1998.

[16] IETF RFC 2474, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, 1998.

[17] IETF RFC 2475, An Architecture for Differentiated Services, 1998.

[18] IETF RFC 3031, Multiprotocol Label Switching Architecture, 2001.

[19] IETF RFC 3032, MPLS Label Stack Encoding, 2001.

[20] IETF RFC 3261, SIP: Session Initiation Protocol, 2002.

[21] IETF RFC 3376, Internet Group Management Protocol, Version 3, 2002.

[22] IETF RFC 3550, RTP: A Transport Protocol for Real-Time Applications, 2003.

[23] IETF RFC 3711, The Secure Real-Time Transport Protocol (SRTP), 2004.

[24] IETF RFC 3951, Internet Low Bit Rate Codec (iLBC), 2004.

[25] IETF RFC 3973, Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised), 2005.

[26] IETF RFC 4346, The Transport Layer Security (TLS) Protocol Version 1.1, 2006.

[27] IETF RFC 4601, Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised), 2006.

[28] ISO/IEC 7498-1, Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model, 1994.

[29] ITU-T Recommendation G.107, The E-Model, a Computational Model for Use in Transmission Planning, 2003.

[30] ITU-T Recommendation G.114, One-Way Transmission Time, 2003.

[31] ITU-T Recommendation G.131, Talker Echo and Its Control, 2003.

[32] ITU-T Recommendation G.711, Pulse Code Modulation (PCM) of Voice Frequencies, 1993.

[33] ITU-T Recommendation G.723.1, Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s, 1996.

[34] ITU-T Recommendation G.726, 40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM), 1993.

[35] ITU-T Recommendation G.728, Coding of Speech at 16 kbit/s Using Low-Delay Code Excited Linear Prediction, 1992.

[36] ITU-T Recommendation G.729, Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP), 1996.

[37] ITU-T Recommendation P.800, Methods for Subjective Determination of Transmission Quality, 1996.

[38] ITU-T Recommendation P.861, Objective Quality Measurement of Telephone-Band Specch Codecs, 1996.

[39] ITU-T Recommendation P.862, Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-End Specch Quality Assessment of Narrow-Band Telephone Networks and Specch Codecs, 2001.

[40] TIA/EIA/TSB116 Draft 5, Voice Quality Recommendations for IP Telephony, Notel Networks, 2000.

[41] NIST Special Publication 800-58, Security Considerations for Voice Over IP Systems, 2005.

[42] Project 25 Statement of Requirements, 2006.

[43] D. E. Comer, Interworking with TCP/IP: Principles, Protocols, and Architectures, 4th Edition, Prentice Hall, 2000.

[44] L. Sun, Speech Quality Prediction for Voice Over Internet Protocol Networks, Ph.D. Thesis, University of Plymouth, 2004.

[45] A. W. Rix and M. P. Hollier, The Perceptual Analysis Measurement System for Robust End-to-End Speech Quality Assessment, IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000.

[46] W. Jiang and H. Schulzrinne, Comparisons of FEC and Codec Robustness on VoIP Quality and Bandwidth Efficiency, Columbia University, 2002.

[47] A. Clark, Voice Quality Measurement: Understanding VoIP, Telchemy Inc.

[48] A. Mihai, Voice Over IP Security A Layer Approach, XMCO Partners.

[49] Cisco Tech Notes, Document ID: 14069, Understanding Codecs: Complexity, Hardware Support, MOS, and Negotiation.

[50] M. Carson and D. Santay, NIST Net - A Linux-Based Network Emulation Tool, NIST.

[51] http://en.wikipedia.org/wiki/GSM

[52] http://en.wikipedia.org/wiki/Mean_Opinion_Score

[53] http://en.wikipedia.org/wiki/Wireshark

[54] http://www.dvsinc.com/papers/iambe.htm